Human Language Technologies as a Challenge for Computer Science and Linguistics – 2023

Patrick Paroubek, Zygmunt Vetulani

WYDAWNICTWO NAUKOWE UAM

Human Language Technologies as a Challenge for Computer Science and Linguistics – 2023

Zygmunt Vetulani & Patrick Paroubek
(eds.)

UNIWERSYTET IM. ADAMA MICKIEWICZA W POZNANIU

Human Language Technologies as a Challenge for Computer Science and Linguistics – 2023

Zygmunt Vetulani
& Patrick Paroubek (eds.)

POZNAŃ 2023

This volume has been compiled from the pdf files supplied by the Authors

Front cover photo: Adrian Wykrota

Cover design: Marcin Tyma & Zygmunt Vetulani

Technical editor: Marcin Tyma & Marta Kunegunda Witkowska

ISBN 978-83-232-4176-8 (Print)
ISBN 978-83-232-4177-5 (PDF)
DOI: 10.14746/amup.9788323241775

WYDAWNICTWO NAUKOWE UNIWERSYTETU IM. ADAMA MICKIEWICZA W POZNANIU
61-701 POZNAŃ, UL. FREDRY 10
www.press.amu.edu.pl
Office: phone 61 829 46 46, fax 61 829 46 47, e-mail: wydnauk@amu.edu.pl
Marketing: phone 61 829 46 40, e-mail: press@amu.edu.pl

Contents

List of Reviewers*

Jemal Antidze, Sokhumi State University
Victoria Arranz, ELDA
Jolanta Bachan, Adam Mickiewicz University, Poznań
Brigitte Bigi, CNRS Aix-en-Province
Krzysztof Bogacki, Warsaw University
Lynne Bowker, University of Ottawa
Mathieu Dehouck, CNRS – École Normale Supérieure – Sorbonne Nouvelle
Pinar Durdu, Kocaeli University
Paweł Dybała, Jagiellonian University
Moses Ekpenyong, University of Uyo
Juuso Eronen, Prefectural University of Kumamoto
Piotr Fuglewicz, TIP Ltd.
Maria Gavrilidou, Institute for Language and Speech Processing, RC Athena
Dafydd Gibbon, University of Bielefeld
Carlos Gomez, University of A Coruña
Filip Graliński, Adam Mickiewicz University, Poznań
Magdalena Igras-Cybulska, AGH University of Science and Technology
Henryk Jankowski, Adam Mickiewicz University, Poznań
Krzysztof Jassem, Adam Mickiewicz University, Poznań
Rafał Jaworski, Adam Mickiewicz University, Poznań
Besim Kabashi, Friedrich-Alexander Universitaet Erlangen-Nuernberg
Szymon Kaczmarek, XTM International Poznań
Irakli Kardava, Adam Mikiewicz University, Poznań
Adnan Kavak, Kocaeli University
Yasutomo Kimura, Otaru University of Commerce
Katarzyna Klessa, Adam Mickiewicz University, Poznań
Grzegorz Krynicki, Adam Mickiewicz University, Poznań
Elmurod Kuriyozov, University of A Coruña
Temur Kutsia, Juhannes Kepler University, Linz
Paweł Lempa, Cracow University of Technology
Yves Lepage, Waseda University
Svetlozara Leseva, Institute for Bulgarian Language, Bulgarian Academy of Sciences
Gérard Ligozat, LIMSI/CNRS
Wiesław Lubaszewski, AGH University of Science and Technology
Khabibulla Madatov, Urgench State University
Sławomir Magala, Rotterdam School of Management, Erasmus University Rotterdam
Bernardo Magnini, Fondazione Bruno Kessler
Belinda Maia, University of Porto
Jakub Malke, Global Partnership Management Institute
Jacek Martinek, Poznań University of Technology
Corentin Masson, Autorité des Marchés Financiers (AMF)
Sanatbek Matlatipov Jr., National University of Uzbekistan
Panchanan Mohanty, GLA University
Agnieszka Mykowiecka, Institute of Computer Science Polish Academy of Sciences
Karol Nowakowski, Tohoku University of Community Service and Science
Ngoc-Thanh Nguyen, Wroclaw University of Science and Technology
David Odrakiewicz, Global Partnership Management Institute
Peter Odrakiewicz, Global Partnership Management Institute
Maciej Ogrodniczuk, Institute of Computer Science Polish Academy of Sciences
Atul Kr Ojha, Data Science Institute, National University of Ireland
Jędrzej Osiński, Adam Mickiewicz University, Poznań
Patrick Paroubek, Universite Paris Sacle – CNRS
Paweł Pawłowski, Poznan University of Technology
Maciej Piasecki, Wroclaw University of Science and Technology
Benjamin Piwowarski, CNRS – Université Paris-Sorbonne
Laurent Prevot, Aix-Marseille Université
Piotr Przybyła, Institute of Computer Science Polish Academy of Sciences
Michal Ptaszynski, Kitami Institute of Technology
Piotr Rybak, Institute of Computer Science Polish Academy of Sciences
Rafal Rzepka, Hokkaido University
Paweł Skórzewski, Adam Mickiewicz University, Poznań
Ivelina Stoyanova, Institute for Bulgarian Language, Bulgarian Academy of Sciences
Janusz Taborek, Adam Mickiewicz University, Poznań
Shiv Tripathi, Berlin School of Business and Innovation
Yuzu Uchida, Hokkai-Gakuen
Josef Van Genabith, DFKI
Jernej Vicic, University of Primorska
Dusko Vitas, University of Belgrade
Marta Kunegunda Witkowska, Adam Mickiewicz University, Poznań
Motoki Yatsu, Aoyama Gakuin
Mariusz Ziółko, AGH University of Science and Technology
Andrzej Zydroń, XTM International

* Double blind peer reviewing

Preaface

With this monograph, as well as with the LTC conferences organised since 1995 at AMU in Poznań, Poland, we continue our activity of fostering sound development of language technologies aiming at facing new challenges in the world marked by the COVID 19 pandemic, wars in Europe, mass migrations and natural disasters at the scale not seen since many years.

The hot issues of the discipline we are engaged in remain approximately the same as before: developing of language and communication technologies to ease contacts and mutual understanding between people, preventing linguistic exclusion, but also preserving language and cultural heritage.

However, the reaction to the invitation to submit papers shows that the priorities of the authors who decided to share their achievements with readers have evolved during the last four years. In particular, we can observe a strong activation of the scientific community vitally interested in languages belonging to the group of languages insufficiently equipped with language engineering tools and resources (the so called Less-Resourced Languages). We observe a particular presence of Central Asia, but also South-Africa, Balkans and Turkey. We also notice a growing interest in interdisciplinary research at the intersection of computer science and humanities, in particular in works focused on expressing emotions and opinions (Japan).

Among emerging phenomena of more general nature, we notice the new trend of business deglobalisation which seems to be a natural reaction to the so far dominating ideology of Global Village. This new trend, reflected in papers on HLTs for business management communication, might will have in the end a significant impact on the evolution of Human Language Technologies.

* * *

In this volume we present contributions by 186 authors from more than 24 countries from Africa, Asia, Australia, Europe and North America. The contributions to this book cover large area of language technologies and related fields. We present this classification in the alphabetical order below:

Computational Semantics
Emotion, Decisions, Opinion (EDO)
HLTs for Business Management Communication
HLTs for Humanities
Human Language Technologies
Language Modeling
Language Resources
Less-Resourced Languages (LRL)
Machine Translation
Non-verbal Communication
Speech Processing
Text Processing

We wish you a pleasant reading!

Poznań, April 2023

Zygmunt Vetulani and Patrick Paroubek
Editors

Turkic language stemmer python package for Natural Language Processing

Nilufar Abdurakhmonova, Alisher Ismailov[2], Ra'no Sayfullaeva[3].

[1]National University of Uzbekistan
n.abduraxmonova@nuu.uz
[2] Tashkent Institute of Finance
alisherismailov1991@gmail.com
[1]National University of Uzbekistan
abduqodir7@gmail.com

Abstract

There has been a massive growth in the volume of data produced around the world in last few decades. Preprocessing text data sets for use in Natural Language Processing chores is normally a time-consuming and expensive effort. Text data, normally obtained from sources such as, but not limited to, web, documents or PDF files, is typically unstructured and prone to artifacts and other types of noise. One of the first methods to process the unstructured text is stemming process. The stemming usually applied in information retrieval, nlp and machine learning. The main goal of information retrieval is to systematically analyze data and to extract some related data or documents that user needed or required information. In this paper, we represent development of python stemmer package named TLstemmer. The goal of the TLstemmer package is to find stem of the any Turkic family language texts. The TLstemmer package is installable through pip function. The download address is 'pip install TLstemmer'.

Keywords: stemming, information retrieval, natural language processing, machine learning, less-resourced-languages, stemming methods

1. Introduction

One of the first parts of the natural language processing pipeline is a stemming (Sharma, 2013). Using stemming, many contemporary search engines related words with prefixes and suffixes to their word stem, to make the search broader that means that it can ensure that the greatest number of relevant matches is included in search results. Stemming has also applications in machine translation, document summarization (Orasan, Pekar, and Hasler L., 2004), and text classification (Gaustad and Bouma, 2002). Popular approach to find stemming is lemmatization. In lemmatization, the developer has to have a good knowledge of the language and its grammar. Lemmatization also requires a dictionary look up, therefore, lemmatization is more complex than basic stemming. Hence, in lemmatization more accurate results are expected (Loponen, and Järvelin, 2010). For example, a word 'better' has a lemma 'good'. This kind of words cannot fix in basic stemming unless the algorithm has a look-up table.

2. Motivation and significance

There are more than 200 languages in the world. Every natural language has its unique characteristics and rules. For the developer, the main problem is that it is very difficult to apply same stemming algorithm on every natural language (Alvares, 2005). Each language has its affixes and as well as individual exceptions, which means it needs handling differently from one to another language. That means normally need have to develop a new stemming algorithm for most of the languages.

When we look at deeper on the process of developing a stemming algorithm, it is clear that each language has a different kind of difficulties for developing a stemmer. However, Turkic family languages have morphological similarity to each other. There is many research has been done for Turkic family language stemmers. However, all these stemmers developed for specific language to use. It is difficult to apply to other Turkic family language. Purpose of this paper is to propose universal stemming package for Turkic family languages. This package is called TLstemmer, it can be install using pip on any python application. (Detailed explanation given in software architecture section.) Main goal of the TLstemmer package is to give researcher readymade functionality for finding the stem of the text. Using this package young researcher especially linguists, can create their own stemmer within few minutes. This package gives researcher a programming advantage.

3. Literature review

3.1 Stemming taxonomy

Stemming taxonomy displays the general view of stemming algorithms. Stemming taxonomy is a collection of conflation methods. Conflation methods have two approaches manual approach and automatic approach. In this paper, we discuss the automatic approach that is stemmers. Stemming means, to reduce the word into its root form (Frakes, and Fox, 2003). Automatic (stemmer) approach has four methods to achieve stemming algorithm, namely affix removal algorithm, successor variety, table lookup and N-gram approaches. Next section will give a brief description of all four automatic methods.

3.1.1 Affix Removal Algorithm

Affixes are usually referred to either prefixes or suffixes. Affix stripping algorithms are based on set rules that are prepared for the algorithm. Normally each rule or affix is defined and stored in the algorithm. Affix removing algorithm finds stem by checking whether an input word has defined any affixes or not. If the defined affixes are

identified, then algorithm will strip the affix from the input word. The remaining part of input word is assumed to be root or stem (Stein and Potthas, 2007).
These are some examples of the affixes:

- if the word starts with 'in', remove the 'in'
- if the word starts with 'mis', remove the 'mis'
- if the word ends with 'ed', remove the 'ed'
- if the word ends with 'ing', remove the 'ing'
- if the word ends with 'ly', remove the 'ly'

This method's advantage is algorithm is simple. However, it has a disadvantage too where it has poor performance when handling with exceptional relations, such as 'went' and 'go'.

Input	Removed suffix	Output	Final Output
Stemming	Ing	Stemm	Stem
Stemmer	Er	Stemm	Stem

Table 1. Suffix removal example.

Table 1 shows the example of suffix removal stemmer. In the first word, "stemming", it has 'ing' suffix, therefore stemmer will remove 'ing' and the remaining part of the word 'stemm'. Usually, affix removal stemmers apply a transformation rule to this type of condition. In this case, double 'mm' on 'stemm' will be transformed to single 'm', the final output will be 'stem'.

3.1.2 Successor Variety Method

Successor variety method is that based on research in structural linguistics. This kind of stemmer determines input word and morpheme boundaries based on the distribution of phonemes in a collection of words. Successor variety method uses letters in place of phonemes (Hafer, and Weiss, 1974). Abhijit (2014) describes successor variety method as "Let α be a word of length n; αi, is a length i prefix of α. Let D be the corpus of words. D αi is defined as the subset of D containing those terms whose first i letters match αi exactly. The successor variety of αi, denoted S αi, is then defined as the number of distinct letters that occupy the i + 1st position of words in D αi. A test word of length n has n successor varieties S αi, Sα2, . . ., S αn". In other words, the successor variety of a string is the number of different characters that follows the words in some body of text.
Table 2 displays an example of the word "stemming using successor variety approach. In this example, the body of the text consists of words "Statistic, steam, star, sole, stemmer, stemming, and stem".
Body of text: Statistic, steam, star, sole, stemmer, stemming, stem
Test word: STEMMING.

Prefix	Successor Variety	Letters
S	2	T, O
ST	3	A, E, O
STE	2	A, M
STEM	1	M
STEMM	2	E,I
STEMMI	1	N
STEMMIN	1	G
STEMMING	1	BLANK

Table 2. Shows that letter variety counts for word STEMMING.

3.1.3 Table Look up Approach

Table lookup method uses a stem of the database. It stores set of stem words into a database. Table lookup approach find the stem by comparing input word with database stems. It applies several constraints, such as the short prefix "be", would not be measured as the stem of the word "behind" (Adamson, G. W., and Boreham, 1974).
For example:

Input	Removed suffix	Final Output
Stemming	ming	stem
Stemmer	mer	stem

Table 3. Table lookup stemmer example.

Table 3 displays an example of how Table lookup stemmers would operate. Normally table lookup stemmers reduce the entered word into its root form by comparing the word with stored stems. In the example, inputs are 'stemming', and 'stemmer', and the table lookup stemmer removed 'ming' from 'stemming' and 'mer' from 'stemmer' by comparing them with the word 'stem' that has been stored to database.

3.1.4 N-Gram Approach

Adamson and Boreham (1974) have designed n-gram method. An n-gram is a method of conflating terms known the shared digram method. The digram consists of a couple of consecutive letters. The n-gram method calculates between pairs of terms based on shared unique digrams.
For example, the words 'statistics' and 'statistical', they can be broken into digrams as follows.

Input	Diagram	Unique digram
Stemming	st ta at ti is st ti ic cs	st ta at is ti ic cs
Stemmer	st ta at ti is st ti ic ca al	st ta at ti is ic ca al

Table 4. Table lookup stemmer example.

In the above example, 'statistics' contains nine digrams, seven of them are unique digrams, the second word "statistical" contains ten digrams, and eight of them are unique digrams These two words share six unique digrams: at, ic, is, st, ta, ti.
After the number of unique digrams is found then a similarity measure based on the unique digrams is calculated using dice coefficient. Dice coefficient is defined as: S=2xC/((A+B)).
C – Common unique digrams
A – Number of unique digrams in the first word
B – Number of unique digrams in the second word.
For the example above, Dice's coefficient would equal
(2 x 6) / (7+ 8) = .80.
From the dice coefficient calculation, it can be concluded that the stem for these two of words lies in the first eight digrams, which are st ta at ti is st ti ic. We can conclude that stem of the 'statistical' is 'statistic'.
There are a few existing stemmers which already developed using all above-mentioned methods. Some available stemmers are Lovins stemmer, Porters stemmer, Dawson stemmer, Paice/Husk stemmer. Each of these stemmers will be discussed in the next following section.

4. Turkic family language morphology

The Turkic languages family consists of 35 languages (Jalil, 2017), spoken by the Turkic peoples of Eurasia from Eastern Europe and Southern Europe to Central Asia, East Asia, North Asia (Siberia), and Western Asia. Turkic languages are spoken by around 170 million people (Johanson, 2015).

From a sociolinguistic perspective, the Turkic languages official status in sovereign states, such as Turkish, Azeri, and Uzbek, Kazakh.

4.1. Morphology of Turkic family languages

In linguistics, morphology described as a study of the forms of words. Morphology will help to analyze the words into their elements (morphemes). In English for example, “programming”, which is composed of “program-”, and “-ming”.

4.2. Turkic family words composition

Turkic words compose by adding prefixes and suffixes to the root word (Abdurakhmonova, 2022; Mengliev, 2021).

The formula of the Turkic words are prefixes + root + suffixes. However, prefix attached words are not many. TLstemmer package’s main focus in on suffixes. Table 5 displays four Turkic language word composition.

Uzbek	Turkish	Kazakh	Tatar
Bino+lar	Bina+lar	Ғимарат+тар	Бина+лар
Maktab+ning	Okul+un	Мектеп+тің	Мәктәп+нең
Biz+ga	Biz+e	Біз+ге	Безг+ә
Ular+ga	Onlar+a	Олар+ға	Алар+га
Dars+lar+ga	Ders+ler+e	Сабақ+тар+ға	Дәрес+ләр+гә

Table 5. Uzbek, Turkish, Tatar and Kazakh language composition.

5. Software description

5.1. Software architecture

TLstemmer is written in Python 3 and provides algorithm and function to accomplish common stemming process for Turkic family language text data sets. It follows a procedural programming style, and the provided function as shown in Fig. 1. A Function is defined at a high level, enabling users to take advantage of parameters for a given Turkic family language text data set. The code is developed from scratch specifically for Turkic family language texts to use. TLstemmer package can be applied to deep learning and text operations like nltk. It consists of a function that has two parameters, which is to identify stem of the given input text. As detailed in Fig 2. A function is designed in a consistent way, offering parameters to select the desired column of the data frame.

4.2. Software functionalities

The stemming module describes the function as follows:

TLstemmer(param1, param2) this function is used to find stem of the input text. The function has two parameters:

1) First parameter should be input text. (one of the Turkic language text). Input text data type should be a dictionary.

Example: inputText = [{‘inputT’: ‘vatanim’}, {‘inputT’: ‘bolalar’}, {‘inputT’: ‘maktabim’}]

2) Second parameter is lexicon of root words. This lexicon should be in dictionary data type. Example: root = [{‘stem’: ‘bola’}, {‘stem’: ‘maktab’}, {‘stem’: ‘lingvistika’}]

TLstemmer(param1, param2) function takes the parameters then proceeds to stemming process. First parameter is input text as mentioned, algorithm will loop through this each input text, letter by letter then compare it with second parameter that is lexicon root words. If input word match with lexicon root word function, it returns it as a stem of the input text. Algorithm takes longest matching stem as an output.

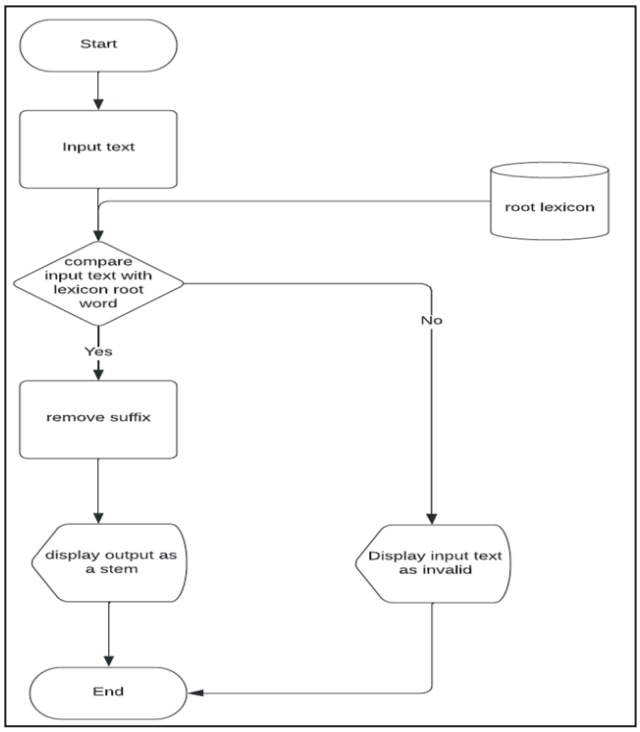


Fig. 1: Flowchart of TLstemmer package algorithm

6. Illustration

6.1. Illustration of installation and testing

In order to install TLstemmer package, go to pypi.org, then type ‘TLstemmer’ in search input. You will get the TLstemmer package.

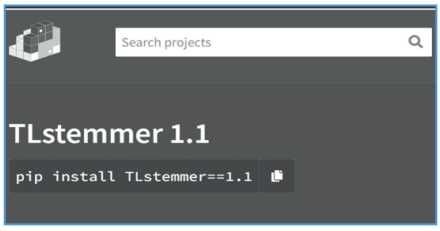


Fig. 2: TLstemmer pypi.org address

Copy the ‘pip install TLstemmer’ line, open any python language IDE, and put the following code to terminal. In this example, we use Jupyter IDE

```
pip install TLstemmer==1.1

Collecting TLstemmer==1.1
  Downloading TLstemmer-1.1-py3-none-any.whl (1.9 kB)
Installing collected packages: TLstemmer
Successfully installed TLstemmer-1.1
Note: you may need to restart the kernel to use updat
```

Fig. 3: installation of TLstemmer package through pip

Now our environment is ready to use. Next section we will create data set to test.

6.2. Collecting Data set

For data collection, we have created few words with root words and their combination with suffixes. There are four language we will test, Uzbek, Turkish, Tatar and Kazakh languages. Table 6 displays the data set

Uzbek	Turkish	Tatar	Kazakh
Men(root word)	ben(root word)	min(root word)	men(root word)
meni	beni	mine	Meni
menda	bende	mindä	Mende
mendan	benden	minnän	Menen
mening	benim	minem	Meniñ
sen(root word)	sen(root word)	sin(root word)	Sen(root word)
Seni	Seni	Sine	Seni
Senda	Sende	Sindä	Sende
sendan	senden	Sinnän	Senen
Sening	Senin	Sineñ	Seniñ
Siz(root word)	siz(root word)	Sez(root word)	Siz(root word)
Sizga	Size	Sezgä	Sizge
Sizni	Size	Sezne	Sizdi
Sizda	Sizde	Sezdä	Sizde
sizdan	Sizden	sezdän	Sizden
sizning	Sizin	sezneñ	Sizdiñ
Biz(root word)	biz(root word)	Bez(root word)	Biz(root word)
Bizga	Bize	Bezgä	Bizge
Bizni	Bizi	Bezne	Bizdi
Bizda	Bizde	Bezdä	Bizde
Bizdan	Bizden	bezdän	Bizden
bizning	Bizim	bezneñ	Bizdiñ
Uzbek	Turkish	Tatar	Kazakh

Table 6. Collection 4 language root words and their combination with suffixes.

6.3. Testing

To start testing the TLstemmer package, first import TLstemmer in Jupyter Notebook as shown in Fig 4.

```
In [4]: import TLstemmer as stem
```

Fig. 4: importing TLstemmer package into python application

Next, define inputText for each language and define lexicon root word for each language. To find the stem of the input texts, call the variable defined in fig 1, that is 'stem' then call the TLstemmer function. In addition, put parameters in function. First parameter is input text then second parameter is root lexicon.

```
putUzbek = [{'inputT':'menga'}, {'inputT':'senga'}
            {'inputT':'bizga'}, {'inputT':'ularga'

otUzbek = [{'stem':'men'},{'stem':'sen'},{'stem':'
           {'stem':'biz'},{'stem':'ular'}]

em.TLstemmer(inputUzbek, rootUzbek)
```

```
Input Word->menga; Root word->men; suffix->ga',
Input Word->senga; Root word->sen; suffix->ga',
Input Word->sizning; Root word->siz; suffix->ning'
Input Word->bizga; Root word->biz; suffix->ga',
Input Word->ularga; Root word->ular; suffix->ga']
```

Fig. 5: Defining input text and lexicon of root words, result of the stem for Uzbek language

```
otTurk = [{'stem':'ben'},{'stem':'sen'},{'stem':'s
          {'stem':'biz'},{'stem':'onlar'}]

putTurk = [{'inputT':'beni'}, {'inputT':'sende'},
           {'inputT':'bizden'}, {'inputT':'onlara'

em.TLstemmer(inputTurk, rootTurk)
```

```
Input Word->beni; Root word->ben; suffix->i',
Input Word->sende; Root word->sen; suffix->de',
Input Word->sizin; Root word->siz; suffix->in',
Input Word->bizden; Root word->biz; suffix->den',
Input Word->onlara; Root word->onlar; suffix->a']
```

Fig. 6: Defining input text and lexicon of root words, result of the stem for Turkish language

```
otTatar = [{'stem':'min'},{'stem':'sin'},{'stem':'
           {'stem':'bez'},{'stem':'alar'}]

putTatar = [{'inputT':'mindä'}, {'inputT':'sine'},
            {'inputT':'bezneñ'}, {'inputT':'alarda

em.TLstemmer(inputTatar, rootTatar)
```

```
Input Word->mindä; Root word->min; suffix->dä',
Input Word->sine; Root word->sin; suffix->e',
Input Word->sezdän; Root word->sez; suffix->dän',
Input Word->bezneñ; Root word->bez; suffix->neñ',
Input Word->alarda; Root word->alar; suffix->da']
```

Fig. 7: Defining input text and lexicon of root words, result of the stem for Tatar language

```
otKazakh = [{'stem':'men'},{'stem':'sen'},{'stem':
            {'stem':'biz'},{'stem':'olar'}]

putKazakh = [{'inputT':'menen'}, {'inputT':'sende'
             {'inputT':'bizdi'}, {'inputT':'olarğa

em.TLstemmer(inputKazakh, rootKazakh)
```

```
Input Word->menen; Root word->men; suffix->en',
Input Word->sende; Root word->sen; suffix->de',
Input Word->sizge; Root word->siz; suffix->ge',
Input Word->bizdi; Root word->biz; suffix->di',
Input Word->olarğa; Root word->olar; suffix->ğa']
```

Fig. 8: Defining input text and lexicon of root words, result of the stem for Kazakh language

7. Impact

The main impact of the created software package on the community is its simplicity and multilingual functionality. The TLstemmer package can be used many Turkic family languages. The package provides the one of the first Python implementation of the stemming algorithm for and allows to use different Turkic languages. Algorithm in the package is designed to have simple calls with flexible parameters, allowing users with minimal Python experience to use it.

8. Conclusion

In this paper, we have discussed TLstemmer package for python language. The TLstemmer package is designed to find the stem of the input text. The TLstemmer package can be applied most of the Turkic family languages. All Turkic family languages have similar morphological form. Therefore, we took advantage of this to create universal stemmer for Turkic family languages. However, depending on the language the package may require to add some changes. Main disadvantage of the TLstemmer package is it finds stem only using lexicon of root words. Hence, if input text outside lexicon it may reject valid word as an invalid word. In the future research, this disadvantage will be solved.

References

Sharma, D. (2013). Stemming Algorithms: A Comparative Study and their. International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868, 7-10.

Orasan C., Pekar V., and Hasler L. (2004): A comparison of summarization methods based on term specificity estimation. In Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC-04). May, Lisbon, Portugal pp.1037-1041.

Gaustad T. and Bouma G. (2002): Accurate Stemming of Dutch for Text Classification. In Computational Linguistics in the Netherlands 2001, pp. 104-117.

Loponen, A., & Järvelin, K. (2010). A dictionary-and corpus-independent statistical lemmatizer for information retrieval in low resource languages. In Multilingual and Multimodal Information Access Evaluation (pp. 3-14). Springer Berlin Heidelberg.

Alvares, R. V., Garcia, A. C. B., & Ferraz, I. (2005). Stembr: A stemming algorithm for the brazilian portuguese language. In Progress in Artificial Intelligence (pp. 693-701). Springer Berlin Heidelberg.

Frakes, W. B., & Fox, C. J. (2003, April). Strength and similarity of affix removal stemming algorithms. In ACM SIGIR Forum (Vol. 37, No. 1, pp. 26-30). ACM.

Stein, B., & Potthast, M. (2007). Putting successor variety stemming to work. In Advances in Data Analysis (pp. 367-374). Springer Berlin Heidelberg.

Hafer, M. A., & Weiss, S. F. (1974). Word segmentation by letter successor varieties. Information storage and retrieval, 10(11), 371-385.

Abhijit Paul, A. D. (2014). An Affix Removal Stemmer for Natural Language. International Journal of Computer Applications (0975 – 8887).

Adamson, G. W., & Boreham, J. (1974). The use of an association measure based on character structure to identify semantically related pairs of words and document titles. Information storage and retrieval, 10(7), 253-260.

Bijal, D., & Sanket, S. (2014). Overview of Stemming Algorithms for Indian and Non-Indian Languages. arXiv preprint arXiv:1404.2878.

Jivani, A. G. (2011). A comparative study of stemming algorithms. Int. J. Comp. Tech. Appl, 2(6), 1930-1938.

Karaa, W. B. A. (2013). A new stemmer to improve information retrieval. International Journal of Network Security & Its Applications, 5(4), 143.

Lennon, M., Pierce, D. S., Tarry, B. D. & Willett, P. (1988). Document retrieval systems. In P. Willett (Ed.), Document retrieval systems, (pp. 99-105). London: Taylor Graham Publishing.

Sirsat, S. R., Chavan, V., & Mahalle, H. S. (2013). Strength and accuracy analysis of affix removal stemming algorithms. International Journal of Computer Science and Information Technologies, 4(2), 265-269.

Rani, S. R., Ramesh, B., Anusha, M., & Sathiaseelan, J. G. R. (2015). Evaluation of Stemming Techniques for Text Classification. International Journal of Computer Science and Mobile Computing, 4(3), 165-171.

Adam, G., Asimakis, K., Bouras, C., & Poulopoulos, V. (2010). An efficient mechanism for stemming and tagging: the case of Greek language. In Knowledge-Based and Intelligent Information and Engineering Systems (pp. 389-397). Springer Berlin Heidelberg.

Hajeer, S. I., Ismail, R. M., Badr, N. L., & Tolba, M. F. (2014). An Adaptive Information Retrieval System for Efficient Web Searching. In Advanced Machine Learning Technologies and Applications (pp. 472-482). Springer International Publishing.

Johanson, L., & Johanson, É. Á. C. (2015). The Turkic Languages. Routledge.

Abdurakhmonova N. Z., Ismailov A. S. and Mengliev D., "Developing NLP Tool for Linguistic Analysis of Turkicn Languages," 2022 IEEE International Multi-Conference onEngineering, Computer and Information Sciences. (SIBIRCON), Yekaterinburg, Russian Federation, 2022, pp.1790-1793, doi: 10.1109/SIBIRCON56155.2022.10017049.

Abdurakhmonova, N., Alisher I. and Sayfulleyeva, R. "MorphUz: Morphological Analyzer for the Uzbek Language," 2022 7th International Conference on Computer Science and Engineering (UBMK), Diyarbakir, Turkey, 2022, pp. 61-66, doi: 10.1109/UBMK55850.2022.9919579.

Fadzli, S. A., Norsalehen, A. K., Syarilla, I. A., Hasni, H., & Dhalila, M. S. S. (2012). Simple rules malay stemmer. In The International Conference on Informatics and Applications (ICIA2012) (pp. 28-35). The Society of Digital Information and Wireless Communication.

Jalil, M. M., Ismailov, A., Abd Rahim, N. H., & Abdullah, Z. (2017). The Development of the Uzbek Stemming Algorithm. Advanced Science Letters, 23(5), 4171-4174.

Mengliev, D., Barakhnin, V.B. and Abdurakhmonova, N. (2021) 'Development of Intellectual Web System for Morph Analyzing of Uzbek Words', Applied sciences [Preprint]. Available at: https://doi.org/10.3390/app11199117. .

Towards analysis of hegemonic masculinity in the dialogues of Polish novels

Jolanta Bachan, Marek Kubis, Natalia Maria Łozińska, Marta Kunegunda Witkowska

Adam Mickiewicz University, Poznań, Poland
{jbachan, marek.kubis, natalia.lozinska, marta.witkowska}@amu.edu.pl

Abstract

Hegemonic masculinity is a set of practices based on orthodox values considered masculine since ancient times: power, dominance, activity, physical strength and readiness to use it and taking risks, agency, rationalism, control and emotional restraint. The paper analyses the manner of speech of the male characters in Polish novels. The focus is placed on the sentiment expressed by the author's commentary to the utterances spoken by the male characters in the dialogues. The analysis centers around verbs appearing in the dialogue stage directions and the structures of a verb with an adverb. For this purpose, the utterances of the male protagonists of the novel extracted from the dialogues of Polish novels collected in the corpus (Kubis, 2021) were used. To identify the problem of the occurrence of linguistic manifestations of the manner of speech of the characters in the novel, a quantitative analysis of the selected verbs and the structures formed with adverbs was performed. The analysis showed that among the types of constructs of male characters in the dialog of Polish novels, models of hegemonic and cooperating masculinity dominate.

Keywords: dialog corpus, sentiment, masculinity studies, Polish novels

1. Introduction

In gender studies, hegemonic masculinity is a set of practices which puts men in a dominant position in a social hierarchy. The aim of the current study is to analyze male characters' utterances in Polish novels, particularly the sentiment expressed by the author's commentary to the utterances spoken by the male characters. The focus is on structures with a verb in a central position in dialogue stage directions .

For this purpose, ca 516,000 male utterances were automatically indicated in a corpus of Polish novels (Kubis, 2021) and the author's commentaries were extracted. To identify the problem of the occurrence of linguistic manifestations of the manner of speech of the characters in the novel, a quantitative analysis of the selected verbs and the structures formed with adverbs was performed.

2. Masculinity studies

All gender perspectives of reading literature, including literary masculinity studies, stem from sociological concepts (Hobbs, 2013). The beginnings of this course of research on gender can be found in the 1970s. The social rebellion against the existing order, which was started by women's movements, caused gender issues from the streets of New York, London and Paris to reach the academies. While the importance of feminist research has never been questioned, scientists undertaking masculinity research have faced (and still face) the need to prove the necessity for such research. Their pioneer, Raewyn Connell, argues that femininity studies and masculinity studies (and later also queer studies) can only be fully understood and developed in relation to each other (Connell, 2005)The document size is 5 pages formatted as described above. The only accepted document type is PDF.

For centuries, patriarchal culture pushed women to the off-stream, while placing men in the mainstream, known as the male-stream (Brod, 1987). The historical and social invisibility of women caused their discursive absence for a long time, but did not the omni-visibility of men do the same? Michael Kimmel, the second leading scholar of critical studies on men and masculinities, in one of his most important publications, pointed out that it is a mistake to treat the ways of acting, the style of speaking, making decisions, choosing topics for research, or even conducting literary narratives by male protagonists as if their gender - their masculinity - had no influence on them (Kimmel, 1996). Sally Robinson added that even if men benefit from their omni-visibility, without a theoretical foundation and conceptual framework that would be able to describe their gender specificity, they will never be understood (Robinson, 2000).

This theoretical foundation, which is the starting point for the research proposed and described in this article, was created by the aforementioned Raewyn Connell, the Australian sociologist who was the first to postulate the need to develop masculinity studies. In her seminal work *Masculinities*, published in 1993, she presented a theory of hegemonic masculinity (Connell, 2005). Hegemonic masculinity is a set of practices based on orthodox values considered masculine since ancient times: power, dominance, activity, physical strength and readiness to use it and taking risks, agency, rationalism, control and emotional restraint. Functioning based on them guarantees striving towards the full fulfillment of the project of ideal masculinity (Karlsson, 2010). Many fail, but because they persevere and continue to benefit from the patriarchy, they are placed within subordinate masculinities. Each deflection from the fulfillment of the project of ideal masculinity places individuals in sets of marginalized or opposing masculinities. The aforementioned project of ideal masculinity can also be an idea with which women and people identifying their gender in the area of intersection identify themselves. The same applies to the category of hegemony. Although it was initially believed that this category could not refer to femininity, let alone other forms

of gender expression (Connell, 2005), today hegemonic femininity is an important reference when describing the cultural space of domination. Hegemonic femininity, as opposed to hegemonic masculinity, should be referred to in the plural form (hegemonic femininities), due to the multitude of possibilities of implementing this category depending on the space in relation to which it will be described.(Hamilton et al., 2019).

These terms are also used in literary studies. In the case of Polish literature at the turn of the 19th and 20th centuries, the corpus of which is the material of our study, the dominant model of masculinity adopted by the authors as a material for building male-personal literary characters is hegemonic masculinity. This is evidenced by the linguistic expressions used in the dialogues of the novel.

3. Corpus of Polish novels

For the purpose of this study we excerpted over 650,000 dialogues from 2434 Polish novels. The data collected for the task covers the period from the early nineteenth century to the mid-twentieth century and were extracted from three distinct sources:

1. Wolne Lektury (Modern Poland Foundation, 2023), a digital library oriented towards school readings that contains carefully edited and contemporized literary works in Polish.
2. The Polish edition of the Wikisource project (Wikimedia Foundation, 2023) which contains transcriptions of printed books whose copyright has expired.
3. The Polona digital library maintained by the National Library of Poland (2023) that provides digitized and OCR-ed copies of printed texts that have fallen in public domain.

Following (Kubis, 2021) and (Karlinska et al., 2022) we combined multi-volume editions of books from Polona into aggregated text files that contain the whole texts of novels. The literary works fetched from Wikisource and Polona were contemporized with the use of a diachronic normalizer (Jassem et al., 2017). From the collected dataset we selected exactly one edition of every novel for our study, giving preference to the most recent editions and favoring scrupulously revised texts from Wolne Lektury over human-made transcriptions available in Wikisource and OCR-ed volumes gathered in Polona.

The boundaries of paragraphs, sentences and tokens were identified in the collected texts and the results were stored in a common, tab-separated format modeled after CONLL-U Plus representation (CoNLL-U Plus Format, 2022).

3. 1. Automatic extraction of male turns

In order to identify dialogue turns we used shallow speech act parser described in (Kubis, 2021). The gender of the speakers was determined on the basis of morphological information provided by a part-of-speech (POS) model trained with the use of the manually annotated 1-million word subcorpus of the NKJP corpus (Przepiórkowski et al., 2012). As a result we obtained the corpus of over 516,000 utterances attributed to male speakers that we used for investigating surface indicators of hegemonic masculinity in Polish novels.

An exemplar of automatically extracted male turn from the corpus, together with its POS-tags is presented below in the format (the commentary is underlined - only the commentaries were used for the analysis:

```
masc (male identifier) \tab author's commentary with POS-tagging \tab the whole male turn with the author's commentary

masc odrzekł/verb niedbale/adv — Ja ? Stamtąd ... — odrzekł niedbale , wskazując tamten plugawy grzech tak się zemścił straszliwie.
```

English translation: *I? From there ... — he replied carelessly, pointing to that filthy sin that took such terrible revenge.*

4. Analysis of commentaries to male turns

The material of ca 516,000 male utterances was used to create frequency lists to identify the most common structures in the author's commentaries. The analysis showed that the verb played the central role in the most frequent commentaries in the novels of the male turns. Therefore, for the semantic analysis of the sentiment, the verb and the structures of the verb with an adverb were used.

4.1. General POS structures

The selected material for analysis was filtered and only POS tags were extracted and ordered according to the frequency of occurrence. Table 1 shows the most frequent structures of the author's commentaries to the male turns in Polish novels. The results show that the most frequent commentaries contain the verbs and the most common are stand-alone verbs. The next category are verbs followed by an adverb. The third category are stand-alone reflexive verbs.

Structures of commentaries	Frequency	Polish example	Translation
Verb	335984	Syknął	he hissed
verb adverb	54691	rzekł miękko	he said softly
reflexive_verb	43284	odezwał się	he spoke up
verb preposition noun	13803	splunął z gniewem	he spat in anger
adverb verb	8867	ostro napadł	he attacked sharply
reflexive_verb adverb	6390	uśmiechnął się lekceważąco	he smiled dismissively
verb noun	6345	wzruszył ramionami	he threw up his arms

Table 1: The most frequent POS-structures of author's commentaries of male turns with Polish examples from the corpus - the examples are random and only illustrate the structure of the commentary.

4.2 Semantic analysis of frequency lists

The obtained frequency results show the language exhibits of sentiment, for which verbs and verbs found in the structure with an adverb or with a preposition and a noun were used in this study.

Classification of the sentiment of the analyzed verbs and verbs with an adverb (or less frequent structures of verbs with a preposition and a noun) was based on the semantic

layer of verbs, whose sentiment was modified by an adverb co -occurring in the structure. These constructions used by the authors of the novel helped to determine the emotional state of the protagonist, its feelings and attitudes towards the interlocutor.

The exemplar results presented in Table 2 show the most common verbs used by the authors of the novel together with the sentiment marker from three categories: positive (P), negative (Neg) and neutral (Neu). At this stage of the study the sentiment markers were added manually based on the semantics of the analyzed words, but in the future the sentiment markers will be taken automatically from the plWordNet-emo sentiment lexicon for Polish (Zasko-Zielińska et al., 2015) as proposed in (Skórzewski, 2019).

Verb	English translation	Frequency
rzekł	he said	55747
zawołał	he called	25583
odparł	he replied	20051
zapytał	he asked	17096
mówił	he spoke	17088
odpowiedział	he replied	13352
odezwał się	he spoke up	12917
spytał	he asked	12270
dodał	he added	9960
odrzekł	he replied	9638
szepnął	he whispered	9707
przerwał	he interrupted	7954
krzyknął	he shouted	7589
mruknął	he murmured	6871

Table 2: Frequency list of verbs above 5000 occurrences. They are all evaluated as neutral.

Adverb following *"rzekł" / said/*	Translation	Sentiment	Frequency
cicho	quietly	Neu	561
z uśmiechem	with a smile	P	490
spokojnie	calmly	P	475
poważnie	seriously	Neu	451
krótko	briefly	Neu	311
wesoło	cheerfully	P	267
półgłosem	in a shivery	Neu	221
nagle	suddenly	Neu	218
łagodnie	gently	P	214
surowo	harshly	Neg	211
stanowczo	firmly	Neg	207
powoli	slowly	Neu	206

Table 3: Frequency list of the verb *"rzekł"* (En. *"he said"*) in a structure with an adverb above 200 occurrences.

The verbal phrase which appears most often is 'he said' (*rzekł*, Table 3), which emphasizes the tendency of the characters to express masculine assertive statements (which may indicate authority and self-confidence).

Adverb following *"zawołał"*	Translation	Sentiment	Frequency
nagle	Suddenly	Neu	356
wesoło	cheerfully	P	345
żywo	vividly	P	176
z zapałem	eagerly	P	152
głośno	out loud	Neu	140
gwałtownie	violently	Neg	125
z oburzeniem	out of indignation	Neg	103
gniewnie	angrily	Neg	103
radośnie	happily	P	102
groźnie	threatening	Neg	92
niecierpliwie	impatiently	Neg	91

Table 4: Frequency list of the verb *"zawołał"* (En. *"he called"*) in structure with an adverb or a preposition and a noun above 90 occurrences.

Adverb following "odparł"	Translation	Sentiment	Frequency
z uśmiechem	with a smile	P	214
krótko	briefly	Neu	207
wesoło	cheerfully	P	185
żywo	vividly	P	173
poważnie	seriously	Neg	173
obojętnie	indifferently	Neg	135
sucho	dry	Neu	124
zimno	cold	Neg	112
stanowczo	firmly	Neg	110
cicho	quietly	Neu	103

Table 5: Frequency list of the verb *"odparł"* (En. *"he replied"*) in structure above 100 occurrences.

The use of the verb 'he replied' (*odparł*, Table 5) may in turn indicate a willingness to confront, which is usually associated with the ability to take risks (Karlsson, 2014). It is worth noting that most of the adverbs have a negative connotation. These relate to emotional restraint and self-control. Even those adverbs that have a positive connotation show the transience of emotions and do not indicate a longer state of emotional excitement.

Adverb following *"przerwał"*	Translation	Sentiment	Frequency
niecierpliwie	impatiently	Neg	169
żywo	sividly	P	158
nagle	suddenly	Neu	103
gwałtownie	rapidly	Neg	103
znów	again	Neu	55
znowu	again	Neu	46
porywczo	gruesomely	Neg	46
wesoło	cheerfully	P	36
szybko	quickly	Neu	31

Table 6: Frequency list of the verb *"przerwał"* (En. *"he interrupted"*) in structure above 30 occurrences.

Adverb following *"krzyknął"*	Translation	Sentiment	Frequency
nagle	suddenly	Neu	139
groźnie	threatening	Neg	62
gniewnie	angrily	Neg	36
znowu	again	Neu	32
radośnie	happily	P	31
gwałtownie	violently	Neg	28
ostro	sharply	Neg	27
głośno	loudly	Neu	24
wesoło	cherfully	P	23
ze złością	angrily	Neg	22
niecierpliwe	impatiently	Neg	22

Table 7: Frequency list of the verb *"krzyknął"* (En. *"he shouted"*) in structure above 20 occurrences.

On the other hand, the verbs presented in Tables 4, 6 and 7 may refer to activities: 'he called' (*zawołał*), 'he interrupted' (*przerwał*), 'he shouted' (*krzyknął*), and the accompanying adverbs repeatedly emphasize the aspect of movement (violence, speed, frequency, intensity).

Adverb following *"mruknął"*	Translation	Sentiment	Frequency
niechętnie	reluctantly	Neg	116
ponuro	gloomily	Neg	57
półgłosem	in a low voice	Neu	40
gniewnie	angrily	Neg	39
posępnie	gloomily	Neg	28
pogardliwie	contemptuously	Neg	22
niewyraźnie	vaguely	Neu	20

Table 8: Frequency list of the verb *"mruknął"* (En. *"he murmured"*) in structure above 20 occurrences.

The verb 'he murmured' (*mruknął*, Table 8) appearing less frequently may indicate emphasis on the situation of subordination (reluctance to the recipient of the uttered sentences). Almost all adverbs appearing in presence of the verbs indicate a state of emotional discouragement and a sense of superiority over the interlocutor, which means that the male character does not have to treat the addressee with respect, seriousness, etc. However, constructions of this type are extremely rare, which suggests that the preferred way of constructing male utterances is to mark them with readiness to confront the interlocutor, regardless of the context resulting from the plot and the arrangement of the literary space (Hobbs, 2013).

5. Conclusions and future work

The analysis of the extracted data allows to conclude that among the types of constructs of male characters in the dialog of Polish novels, models of hegemonic and cooperating masculinity dominate. The majority of adverbs co-occurring in the structure of the utterances were negative or neutral. They can be assigned to specific features which, according to the theory proposed by Connell (2005), indicate the presence of hegemonic masculinity:

· power and dominance – 'he said harshly, firmly' (rzekł stanowczo, surowo), 'he called threatening' (zawołał groźnie), 'he replied seriously' (odparł poważnie), 'he interrupted sharply' (przerwał ostro), 'he shouted threatening, sharply, loudly, contemptuously' (krzyknął groźnie, ostro, głośno, pogardliwie);

· activity – 'he called suddenly, out loud' (rzekł nagle, głośno), 'he interrupted again, rapidly, quickly' (przerwał znowu, szybko), 'he shouted loudly, impatiently' (krzyknął głośno, niecierpliwie);

· physical strength and readiness to use it, tendency to violence (violent behaviours) – 'he called violently, angrily, threatening' (zawołał gwałtownie, gniewnie, groźnie), 'he interrupted gruesomely, rapidly, sharply' (przerwał porywczo, gwałtownie, ostro), 'he murmured contemptuously' (mruknął pogardliwie);

· control and emotional restraint – 'he said briefly' (powiedział krótko), 'he replied dry, cold, firmly' (odparł sucho, zimno, stanowczo), 'he murmured reluctantly, gloomily, vaguely' (mruknął niechętnie, posępnie, niewyraźnie).

It should be noted that the given phrases do not necessarily indicate negative sentiment. It can be assumed that such a shape of male dialogues in novels, in which the male protagonist of hegemonic masculinity is clearly visible, is characteristic of the period from which the studied novels come from. At the 19th and the beginning of the 20th centuries, the male protagonist of hegemonic masculinity was synonymous with the male protagonist of ideal masculinity.

Other male protagonists of masculinity appeared in the literature of that time, but most of them were built on the basis of defective male heroes, losers or cursed men.

Thus, if a literary protagonist was to represent a man (a male character) who had successfully undergone the initiation process to become a real man (Karlsson, 2014), he had to be built on the male protagonist of hegemonic masculinity, which was emphasized by his statements.

In the future we plan to check how many male protagonists of cooperating masculinity can be distinguished among the hegemonic models. However, one should then take into account a wider linguistic spectrum of novel dialogues, extended, for example, by nouns accompanying verbs, but also adjectives defining the sentiments of utterance.

Last but not least, a comparison study will be carried out using modern unsupervised machine learning methods to create clusters of collocations in male turns with those in other turns and a stylometric study.

References

Brod H. (2005). *Theorizing Masculinities*. Sage Publication, ed.. ISBN 978-0-8039-4904-1.

Connell R. (2005). *Masculinities*. Cambridge: Polity Press ISBN 978-0-7456-3427-2.

CoNLL-U Plus Format (2022) https://universaldependencies.org/ext-format.html, (accessed: 28.2.2022)

Hamilton L.T., Armstrong E.A., Lotus Seeley J., Armstrong E.M. (2019). Hegemonic Femininities and Intersectional Domiantion, *Sociological Theory*, vol. 37, no. 4.

Hobbs A.(2013). Masculinity Studies and Literature, Literature Compass vol. 10, Issue 4. https://doi.org/10.1111/lic3.12057

Jassem, K., Graliński, F., Obrębski, T. (2017). Pros and Cons of Normalizing Text with Thrax. In: *Proceedings of the 8th Language and Technology Conference*. Poznań: Fundacja Uniwersytetu im. Adama Mickiewicza, pp. 230-235.

Karlsson G. (2014). Masculinity as Project: Some Psychoanalytic Reflections, Norma: *International Journal for Masculinity Studies* 2014, vol. 9, no. 4.

Kimmel M. (Eds.). (1996). *Changing Men: New Directions in the Study of Men and Masculinity*. Newbury Park, California: SAGE Publications. ISBN 978-0-8039-2996-8.

Kubis M. (2021). Quantitative analysis of character networks in Polish 19th- and 20th-century novels. In: *Digital Scholarship in the Humanities*, 36 (Supplement_2), pp. ii175–ii181. doi:https://doi.org/10.1093/llc/fqab012

Karlińska, A., Rosiński C., Wieczorek J., Hubar P., Kocoń J., Kubis M., Woźniak S., Margraf A., and Walentynowicz W. (2022) Towards a contextualised spatial-diachronic history of literature: mapping emotional representations of the city and the country in Polish fiction from 1864 to 1939. In: *Proceedings of the 6th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*. Gyeongju: International Conference on Computational Linguistics, pp. 115–125.

Przepiórkowski, A., Bańko, M., Górski, R. L. and Lewandowska-Tomaszczyk, B. (Eds.) (2012). *Narodowy Korpus Języka Polskiego*. Warszawa: Wydawnictwo Naukowe PWN.

Robinson S.(2000). *Marked Men. White Masculinity in Crisis*. Columbia: University Press. ISBN 978-0-2311-1293-2.

Scott J. (Eds.) (2015). *Hegemonic masculinity. A Dictionary of Sociology* (4th ed.). Oxford and New York: Oxford University Press. p. 302. doi:10.1093/acref/9780199683581.001.0001. ISBN 9780191763052. LCCN 2014942679.

Skórzewski, P. (2019) Using book dialogs to extract emotions from texts in Polish. In: Vetulani Z., Paroubek P. (Eds.) *Human Language Technologies as a Challenge for Computer Science and Linguistics.* Poznań: Wydawnictwo Nauka i Innowacje. ISBN 978-83-65988-30-0, pp. 244-255.

Wolne Lektury (2023). About the project. Available at: https://wolnelektury.pl/info/o-projekcie/ (accesed: 1.3.2023).

Wikimedia Foundation. (2023). About Wikisource: https://wikisource.org/wiki/Wikisource:About_Wikisource. (accessed: 1.3. 2023)

National Library of Poland. (2023). About Polona Website: https://polona.pl/page/about-polona/.(accesed:1.3.2023).

Zaśko-Zielińska, M., Piasecki M., and Szpakowicz S. (2015). A large wordnet-based sentiment lexicon for Polish. In: Angelova G., Bontcheva K., and Mitkov R. (Eds.), *International Conference Recent Advances in Natural Language Processing. Proceedings.* Bulgaria:Hissar

DeepL and Google Translate Translating Portuguese Multi-Word Units into French: Progress, Decline and Remaining Challenges (2019-2023)

Françoise Bacquelaine[1]

[1]Centre of Linguistics of the University of Porto
franba@letras.up.pt/shirleybac@gmail.com

Abstract

The transition from statistical machine translation trained with machine learning to neural machine translation (NMT) using deep machine learning has proved successful for high-resourced languages. Researchers are exploring new avenues such as zero-shot NMT models for less-resourced languages or the use of English as a pivot language to improve NMT performance. A comparative study conducted in 2019 and 2021 on DeepL (DL) and Google Translate (GT) raw NMT output shows that the performance of GT deteriorated significantly in 2021, mainly because it seemed to use English as a pivot language between two romance languages. In 2023, the same sample of 167 instances of Portuguese multi-word units (MWU) expressing progression and proportion was translated into French by DL and GT. The output in 2019, 2021 and 2023 NMT is analyzed in terms of potential error factors in the Portuguese sample and actual error types in NMT output. The progress of DL from 2019 to 2023 is insignificant while GT exceeds its 2019 score after the 2021 decline. Stronger error factors are unusual structures, combination of potential error factors, and longer MWUs. Phraseology, calque and nonsense are the most frequent error types in this study on NMT progress, decline and remaining challenges.

Keywords: neural machine translation; pivot language; corpora; phraseology; error

1. Introduction

The rather soft transition from statistical machine translation (SMT), trained with machine learning, to neural machine translation (NMT), using ever-evolving deep machine learning, has proved successful. Apart from using the same data as SMT since the neural adventure began in the mid-2010s, researchers are exploring new avenues such as zero-shot NMT models for less-resourced languages (Zhang et al., 2020) or the use of English as a pivot language (Soler Uguet et al., 2022). After a remarkable improvement over SMT, progress has slowed down and successive attempts to improve the model may or may not be successful. A study conducted in 2021 (Bacquelaine, 2022a; idem 2022b) suggests that Google Translate (GT) sometimes uses English (EN) as a pivot language[1] to translate multi-word units (MWU) from Portuguese (PT) into French (FR). Consequently, its score drops dramatically compared to 2019 and other NMT systems (DeepL, eTranslation).

MWU translation is a challenge both for human translators and machines, mainly because ambiguity can raise "problems" at phrase, syntactic and semantic level (Koehn 2020). This study focuses on three PT MWUs. The first (*cada vez* COMP[2]) expresses quantitative or qualitative progression (PROG), the second (typically: NUM *em cada* QP[3]) indicates proportion between a set and a subset (P3S), and the third (typically: QP *por cada* QP) a proportion between two sets (P2S), as shown in examples (1) to (4) taken from the aligned corpus Europarl v7 (Tiedermann, 2012):

(1) Quantitative PROG
- *... cada vez mais mercados...*
- *... more and more markets ...*

(2) Qualitative PROG
- *... artes da pesca cada vez mais selectivas.*
- *... increasingly selective fishing gear.*

(3) P3S
- *... um em cada dois homens ...*
- *... one in two men ...*

(4) P2S
- *... uma embarcação por cada 70 cidadãos.*
- *... one boat for every 70 citizens.*

If the EN universal quantifier *every* is possible to translate P3S and P2S and the FR universal quantifier *toujours* to translate PROG, *each* and *chaque* are not, according to a human translation model obtained from several good quality aligned corpora (Bacquelaine, 2020). Hence the first criterium to assess NMT is literality or word-for-word translation. Any translation of these three MWUs in FR with *chaque* is considered as literal and therefore wrong. The second criterium is acceptability. Any translation by one of the model's solutions is acceptable. Typically, PROG translates in FR as *de* COMP *en* COMP, P3S as QP *sur* NUM, and P2S as QP *pour* QP.

In this narrow scope, this paper aims to examine the evolution of the literality and acceptability performance of DeepL (DL) and GT between 2019 and 2023, to determine a possible link between error factors in the PT sample and error types in NMT, and to identify, system by system, the progress, decline and remaining challenges according to error types detected in the output.

First, methodology, tools, corpora and analysis criteria are described in section 2. Then, results are presented and discussed in three subsections: the global evolution of DL and GT performance between August 2019 and January 2023, the assessment of the causal link between potential

[1] Markus Foti (DGT), personal communication (2021): EN is used as a pivot language in eTranslation.

[2] Comparative adjective or adverb: *mais, menos, melhor, pior, menor, maior*.

[3] QP: quantifier phrase consisting of a cardinal numeral adjective (NUM) and a noun (N), such as *três Portugueses - three Portuguese*.

error factors in the PT MWU instances and actual error types in the FR NMT output, and the evaluation of progress, decline and remaining challenges in 2023.

2. Materials and Methods

We adopt first a diachronic approach. The period covered is very short, but NMT has shown more progress in less than ten years than any other (hybrid) model before. The global evolution of GT and DL is evaluated in terms of acceptable MWU translations in FR. Then, potential error factors in the PT sample and actual errors in the NMT output by GT and DL (2019-2023) are analyzed to determine whether there is a causal link between them. Finally, the respective progress and decline of GT and DL are observed year by year, and remaining challenges in 2023 are identified.

DL and GT are two well-known NMT systems that can be used in daily life, mostly to translate general language. GT developed from statistical to neural MT and DL emerged as NMT from Linguee (dictionary and search engine for aligned bilingual segments).

For the first part of this study, a sample of 102 instances of PROG, 41 of P3S, and 24 of P2S was selected from *CETEMPúblico* (CTP), a Portuguese journalistic corpus from the end of the 20th century explored with AC/DC (Santos and Bick, 2000). PROG is much more frequent in general language than P3S and P2S, and some instances were selected because of specific translation challenges. So, the sample does not reflect general use, but we must presume the PT instances are correct. It was translated into FR by GT and DL in August 2019, September 2021, and January 2023. The raw NMT output is analyzed in terms of literality and acceptability.

For the second and third parts, PT instances that had been well translated (non-literal and acceptable FR MWUs) by GT and DL in 2019, 2021 and 2023 were excluded. The remaining corpus consists of 110 PT instances and 660 raw (mis)translations in FR by GT and DL in 2019, 2021, and 2023: 64 PT instances of PROG, 30 of P3S, and 16 of P2S.

Potential error factors in the PT sample result from the selection of specific instances to challenge the machine. Some challenges are common to PROG and proportion MWUs. They are classified into eight categories: (1) *cada vez mais/menos* as a sentence adverb of frequency quantification (AQF, Leal 2012); (2) splitting (*scission*); (3) inversion; (4) split inversion; (5) coordinated instance; (6) ellipsis in coordinated instance; (7) non-compositional sense (idiom in the broadest sense, including puns); (8) atypical preposition (PREP) in PT, i.e. other than *em* for P3S and *por* for P2S; (9) atypical structure of P3S (2 N instead of one); (10) long QP. Two or three factors may combine in a single instance.

Actual NMT errors detected in FR fall into eight types: (1) calque from PT; (2) calque from EN; (3) omission; (4) addition; (5) nonsense; (6) wrong meaning; (7) opposite meaning; (8) phraseological inadequacy. Apart from typical equivalents of PROG, P3S and P2S, other FR equivalents are attested in the model and accepted as possible translations of the PT MWUs. Omission and addition of part of the PT segment in the NMT output in FR usually result in semantic errors (5, 6 and 7) while phraseological errors at lexical or syntactic level may lead to lack of fluency (wrong collocation, wrong PREP, unusual inversion or splitting, …), agrammaticality (omission of internal argument, …) or semantic issues (5, 6, 7). So, a combination of errors is also possible.

To evaluate the causal link between potential error factors and actual errors, the approach is global, and results are presented in two steps: average number of errors per potential error factor then number of actual errors by type. So, the aim in the second subsection is not to examine the evolution of systems, but to try to establish a causal link between the error factors and the actual errors.

To conclude the analysis, the progress and decline of each system are analyzed diachronically, comparing the number of actual errors by type, to identify some remaining challenges in 2023 GT and DL output.

3. Results and Discussion

The results are divided into three subsections. The global performance of GT and DL from August 2019 to January 2023 is presented in 3.1.; the possible causal link between potential error factors and actual errors is discussed in 3.2.; the progress, decline and remaining challenges are addressed in the last subsection.

3.1. DL and GT from 2019 to 2023

Figure 1 presents the evolution of the global performance of Gt and DL in translating the 167 PT instances in French.

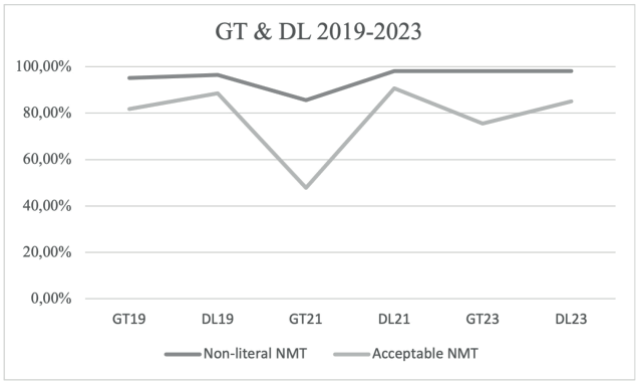


Fig. 1: Global evolution of GT and DL in terms of literality and acceptability performance

In terms of literality, DL improves less than 2% from 2019 to 2021 and stagnates after that. From 2019 to 2023, GT improves nearly 3%, after declining almost 10% from 2019 to 2021. As to acceptability, all scores are lower. The decline of GT in 2021 is much more obvious. Surprisingly, the 2019 scores (GT: 81,74% and DL: 88,62%) are slightly higher than in 2023 (GT: 75,45% and DL: 85,03%). Here are some examples illustrating these results:

(5) Literality

- *Cada vez com maior frequência …* (CTP)
- *De plus en plus souvent, …* (GT23, DL21-23)
- *?... de plus en plus …* (GT19, DL19)
- ** Chaque fois avec une plus grande fréquence …* (GT21)
- *More and more often* / *Increasingly* /... (EN)

In (5), the PT MWU is split. GT23, DL21 and DL23 produce the best output. In 2019, both systems give an acceptable output, i.e. attested in the human translation model, but GT21 proposes an unacceptable calque from PT.

(6) Acceptability

- *uma em cada 5000 gravidezes*.
- *une grossesse sur 5 000.* (GT19, GT23, DL21-23)
- **une femme sur 5 000 grossesses.* (DL19)
- **un dans tous les 5000 grossesses.* (GT21)
- *one in every 5,000 pregnancies.* (EN)

The PT instance in (6) does not present any significant challenge, and none of the proposals contains *chaque*. Nevertheless, DL19 added a second N (*femme*), which results in an unacceptable agrammatical MWU, while GT21 produces a calque from EN (NUM *in every* QP).

The next example illustrates the consistent performance of DL, the decline of GT in 2021, and the better output of GT in 2019 than 2023.

(7) Evolution of acceptability scores

- *em cada dez segundos que passam*
- *toutes les dix secondes* (GT19, DL19-23)
- ?*toutes les dix secondes qui passent* (GT23)
- **dans chaque seconde dix qui passent* (GT21)
- *every ten seconds* (EN)

In (7), GT21 produces a hybrid calque from PT and EN that results in nonsense. Surprisingly, GT NMT is more literal in 2023 than in 2019.

Literality is not a significant challenge any more for GT and DL, but acceptability still is. It is therefore necessary to go deeper into the analysis to identify some obstacles to the production of quality translations of these MWUs.

3.2. Potential error factors in PT and actual errors in FR

Potential error factors are distributed among the 110 instances of the corpus as shown in Table 2:

Potential error factor	Nr
Split instance	22
Idiom in the broadest sense	21
No particular challenge	17
Combination of 2 to 3 factors	16
Atypical PREP (P3S and P2S)	8
AQF (PROG)	7
P3S with two N instead of one	6
Coordinated instance	4
Inversion (P3S and P2S)	3
Long QP	3
Split inversion (P3S and P2S)	2
Ellipsis in coordinated instances	1

Table 1. Distribution of error factors among instances.

As to hybrid error factors, splitting combines with idiom (4 instances), coordination (1), ellipsis (2); inversion combines with atypical PREP (1); split inversion combines with long QP (5) or with two N in P3S and long QP (1); and two N in P3S combines with long QP (2).

Globally, 386 errors were identified in the output of GT19-23 and DL19-23: 155 phraseology issues, 53 calques from EN, 50 calques from PT, 45 nonsenses, 42 wrong meanings, 27 omissions, 12 additions, and only 2 opposite meanings in this corpus.

The average number of any error type per factor was calculated as the number of actual errors divided by the number of instances of each factor. The results are presented in Figure 2:

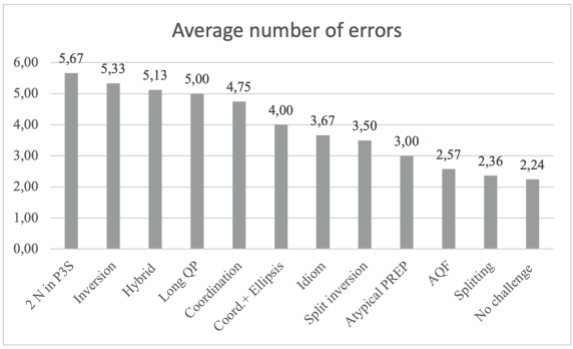


Fig. 2: Average number of errors per error factor instance

According to this chart, GT and DL's performance is higher with more usual structures (no challenge, splitting, AQF), as could be expected. The four strongest error factors provoke mostly phraseological errors, such as agrammatical use of two N in the French P3S output, but calques from PT and EN, nonsense, wrong meanings and omissions are other frequent errors in the case of factor combination (*Hybrid*). Example (8) illustrates the causal link between combination of three potential error factors and actual errors:

(8) Split inversion, 2N in P3S and long QP

- *... em cada mil habitantes há respectivamente 592 e 582 pessoas que compram pelo menos um jornal por dia.* (CTP)
- ?*...avec respectivement 592 et 582 personnes achetant au moins un journal par jour pour mille habitants.* (DL19)
- ?*... avec respectivement 592 et 582 personnes pour mille habitants achetant au moins un journal par jour.* (DL21-23)
- ?*... 592 et 582 personnes achètent respectivement au moins un journal par jour pour 1 000 habitants.* (GT23)
- **... dans tous les mille habitants il y a respectivement 592 et 582 personnes qui achètent au moins un journal par jour.* (GT19)
- **... hors de mille habitants, il y a respectivement 592 et 582 personnes qui achètent au moins un journal par jour.* (GT21)
- *... with 592 and 582 out of every thousand inhabitants who respectively buy at least one newspaper a day.* (EN)

In PT, split inversion of P3S combines with two N (*habitantes*, *pessoas*) and a long QP (*592 e 582 pessoas que compram pelo menos um jornal por dia*) resulting in an unusual structure (*em cada* QP […] QP). The two nearly synonymous N (*personnes* and *habitants*) are systematically translated in FR. GT19 and GT21 give calques from EN (*in every QP* and *out of QP*) that are not attested in the human translation model.

The coordination and ellipsis scores are in the middle. Example (9) illustrates a challenging ellipsis of the ordering element *cada vez* in coordinated instances:

(9) Ellipsis in coordination

- *cada vez mais conflitos e mais violentos", ...* (CTP)
- *des conflits toujours plus nombreux et plus violents", ...* (GT21)
- ?*des conflits de plus en plus violents", ...* (GT19)
- ?*de plus en plus de conflits et de violence", ...* (DL19-23 and GT23)
- *ever more numerous and more violent conflicts...* (EN)

In (9), the PT instance coordinates a N (*conflitos*) with an ADJ (*violentos*), which is unusual in FR. Exceptionally, GT21 gives the best output using a calque from EN attested in the human translation model (*toujours plus*) and coordinating two ADJ (*nombreux* and *violents*) Like *cada vez*, *toujours* can operate on both coordinated elements while *de plus en plus* should be repeated. GT19 keeps an N and an ADJ. It expresses the qualitative progression correctly (*de plus en plus violents*) but omits the quantitative progression of N (*conflits*). The others select two N and produce a wrong meaning.

Idioms are particularly challenging when they combine with humour, as in example (10):

(10) Idiom and splitting

- *Para os noruegueses, isto está cada vez com menos espinhas...* (CTP)
- ?*Pour les Norvégiens, cela devient de moins en moins acnéique ....* (GT19)
- ?*Pour les Norvégiens, cela fait de moins en moins de boutons...* (GT23)
- ?*Pour les Norvégiens, cela devient de moins en moins osseux...* (DL19)
- ?*Pour les Norvégiens, il s'agit de devenir de moins en moins boutonneux...* (DL21-23)
- **Pour les Norvégiens, c'est devient moins et moins de boutons ...* (GT21)
- *For Norwegians, it's getting easier and easier...* (EN)

In its literal sense, the idiom *estar com espinhas* means "to suffer from acne". In informal usage, *sem espinhas* means *easily*, *without problems*. In (10), ambiguity arises from the pun based on these two idioms. All translations are considered as nonsense. Besides, GT21 produces an agrammatical (**est devient moins et moins*) calque from EN (*is becoming less and less*).

The number of actual errors is presented by type in Figure 3:

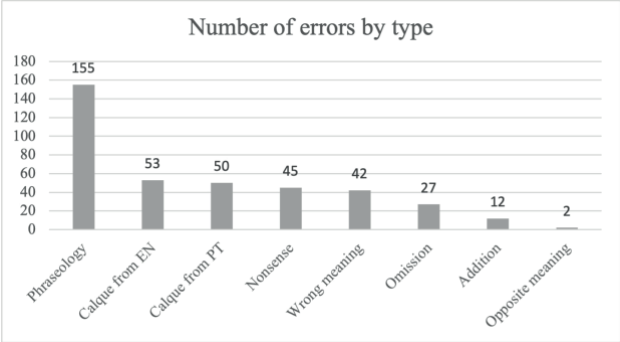


Fig. 3: Number of actual errors by type

Phraseology in the broadest sense includes agrammaticality as in example (8) in the case of DL19-23 and GT23 or GT21's output in example (10). This error type also includes lack of fluency, such as GT23 in (7) or wrong collocations illustrated in example (11):

(11) Phraseology: wrong collocation

- *... quem tem trabalho trabalha cada vez mais.* (CTP)
- ?*ceux qui ont un emploi travaillent de plus en plus fort.* (GT19)
- *... those who have jobs are working more and more.* (EN)

GT19 and DL19 added an ADJ in FR, but the collocation *travailler fort* is very odd whereas *travailler dur* proposed by DL19 is the best output, since GT21-23 and DL21-23 choose the easy correct solution without ADJ (*travailler de plus en plus*) and with a certain meaning loss.

The diversity of phraseology explains its high score. Calques from EN and from PT are on a par (53 and 50). They are illustrated in most of the above examples (5, 6, 7, 8, 10). At the semantic level, nonsense like in (7) and (10) and wrong meaning as in (9) follow with 45 and 42 errors. Omission, addition shown in (6), and opposite meaning are not very significant in this corpus. Here is the only example of opposite meaning in a segment containing two PT MWU instances:

(12) Opposite meaning and omission

- *Há quem, com um certo humor, defina como especialista aquele que sabe cada vez mais de cada vez menos.* (CTP)
- *Il y a ceux qui, avec un certain humour, définissent comme experts ceux qui en savent de moins en moins.* (GT19)
- *There are those who, with a certain humour, define a specialist as one who knows more and more about less and less...* (EN)

In (12), the PT instance is classified as an idiom due to its idiomatic structure (*saber muito de pouco*) and its explicit humorous nature. As with all the others, GT19 supplies the necessary pronoun *en* in FR, but it omits the first MWU (*cada vez mais*), whose co-occurrence with its antonym is unusual. It results in the opposite meaning since the adverbial translated MWU modifies the V *savoir* and does not have any internal argument. DL21-23 produce a perfect solution (*celui qui en sait de plus en plus sur de moins en moins de choses*). DL19 and GT23 propose the same solution for the first MWU but with an adverb instead of the necessary internal argument (NP) in FR (?*celui qui en sait de plus en plus de moins en moins*). GT21 output is nonsense: *Certaines personnes avec une certaine humeur, définies comme un expert qui en sait toujours plus sur un temps moins.*

3.3. Progress, decline, remaining challenges

Progress, decline, and remaining challenges are identified system by system according to error types in Table 2:

	GT19	GT21	GT23	DL19	DL21	DL23
Phraseology	31	42	25	22	19	16
Calque from EN	3	44	2	0	2	2
Calque from PT	8	22	4	6	5	5
Nonsense	4	28	3	4	3	3
Wrong meaning	15	3	4	12	4	4
Omission	9	0	6	4	4	4
Addition	3	5	2	2	0	0
Opposite meaning	2	0	0	0	0	0

Table 2. Progress, decline, remaining challenges.

The phraseology is improving, but it remains a challenge for GT and DL. GT21 can be seen as an unfortunate attempt to improve NMT performance, possibly using EN as a pivot language or poor quality data, so only the evolution between GT19 and GT23 is relevant here. GT23 has fewer errors of

any type than GT19, but fewer acceptable instances (Fig. 1) since error types can combine. Calques from EN are the only slight setback in the progress of DL. As to semantic issues and calques from EN, GT23 and DL23 are very similar, but DL23 outperforms GT23 as far as other error types are concerned, except for calques from PT as in example (13):

(13) Phraseology and calque from PT

- *Cada vez fico mais esclarecido com a instituição com que lido.* (CTP)
- ?*Je suis de plus en plus éclairé sur l'institution avec laquelle je traite.* (GT23)
- **Chaque fois, je deviens plus éclairé sur l'institution avec laquelle je traite.* (DL23)
- *I understand increasingly well the institution I am dealing with.* (EN)

None of the systems proposes an acceptable translation of the idiom *ficar esclarecido com*, which is considered as a phraseological error. Besides, DL23 produces a calque from PT.

(14) Addition

- *Saramago [...] afirma que existe uma analfabetização lenta, que vai minando a área dos alfabetizados, que sabem cada vez menos ler, escrever e «sobretudo pensar».* (CTP)
- *... qui savent de moins en moins lire, écrire et surtout penser.* (DL23)
- ?*... qui savent de moins en moins comment lire, écrire et « surtout penser »*. (GT23)
- *... who increasingly know less about how to read, write and, "above all, think".* (EN)

In this last example, there isn't any specific challenge in PT. DL23 gives a correct output, but GT23 adds *comment*, a calque from EN.

4. Conclusion

The analysis of the small corpus provides some insight into the evolution of DL and GT from 2019 to 2023, it confirms the causal link between atypicality and actual errors, and it identifies some remaining challenges facing machines and humans translating PT MWU expressing PROG, P3S and P2S into FR. Apart from the decline of GT in 2021, 2023 results are encouraging as to literality. Nevertheless, calques from EN are still present in 2023 and calques from Portuguese seem hard to avoid completely, even though the number of word-for-word translations decreases. Phraseology in the broadest sense remains a major challenge in the case of these three MWUs including variables (COMP and QP). These variables are usually short, but challenging, atypical instances were selected on purpose. It is therefore only natural that syntactic, semantic and phraseological errors are more numerous with longer combinations and unusual complex structures that are under-represented in the data since the machine generalizes from the data, without considering unusual MWU structures.

This study only confirms some well-known weaknesses of NMT. As a linguist, I leave it to the engineers to find solutions to the linguistic problems raised here.

References

Bacquelaine, F. (2022a). DeepL et Google Translate face à l'ambiguïté phraséologique. *Journal of Data Mining and Digital* Humanities, 2022, *Towards robotic translation?*. https://doi.org/10.46298/jdmdh.9118.

Bacquelaine, F. (2022b). Traduction d'unités polylexicales du portugais en français par MT@EC et eTranslation. *Revue Traduction et Langues* 21(1), pp. 56-76.

Bacquelaine, F. (2020). Traduction humaine et traduction automatique du quantificateur universel portugais en français et en anglais [unpublished doctoral dissertation]. Faculty of Arts of University of Porto. https://catalogo.up.pt/exlibris/aleph/a23_1/apache_media/FJ5KML8897M4P7HDKXC8RMN37XLAB8.pdf

Koehn, P. (2020). *Neural Machine Translation*. Cambridge: Cambridge University Press.

Leal, A. (2012). Cada vez mais/menos: comparative construction or quantification over eventualities?. In: Schnedecker C., Armbrecht C. (Eds.) *La quantification et ses domaines : actes du colloque de Strasbourg 19-21 octobre 2006*, pp. 355-366. Paris: Honoré Champion.

Santos, D. and Bick, E. (2000). Providing Internet access to Portuguese corpora: the AC/DC project. In: Gavrilidou, M., Carayannis, G., Markantonatou, S. Piperidis, S., Stainhauer, G. (Eds.), *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC 2000) 31 May - June 2, 2000, Athens, Greece*, pp. 205-210. European Language Resources Association (ELRA). Online at: http://www.lrec-conf.org/proceedings/lrec2000/pdf/85.pdf. Access date: February 19, 2023.

Soler Uguet, C., Bane, F., Anna Zaretskaya, A. and Tània Blanch Miró, T. (2022). Comparing Multilingual NMT Models and Pivoting. In: Moniz, H., Macken, L., Rufener, A., Barrault,, L., Costa-jussà, M. R., Declercq, C., Koponen, M., Kemp, E., Pilos, S., Forcada, M. L., Scarton, C., Van den Bogaert, J., Daems, J., Tezcan, A., Vanroy, B., Fonteyne, M. (Eds.) *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pp. 231–239. Online. European Association for Machine Translation. https://aclanthology.org/2022.eamt-1.26. Access date: February 19, 2023.

Tiedemann, J. (2012). Parallel Data, Tools and Interfaces in OPUS. In: Calzolari N., Choukri K., Declerck T., Dogan M. U., Maegaard B., Mariani J., Moreno A., Odijk J., Piperidis S. (Eds.) *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'2012)*. Retrieved from: http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf. Access date: February 19, 2023.

Zhang, B., Williams, P., Titov, I. and Sennrich, R. (2020). Improving Massively Multilingual Neural Machine Translation and Zero-Shot Translation. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (Eds.) *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1628–1639. Online. Association for Computational Linguistics. DOI: 10.18653/v1/2020.acl-main.

An analysis of produced versus predicted French Cued Speech keys

Brigitte Bigi

Laboratoire Parole et Langage, CNRS, Aix-Marseille Univ.
5 avenue Pasteur, 13100 Aix-en-Provence, France.
brigitte.bigi@cnrs.fr

Abstract

Cued Speech is a communication system developed for deaf people to complement speechreading at the phonetic level with hands. This visual communication mode uses handshapes in different placements near the face in combination with the mouth movements of speech to make the phonemes of spoken language look different from each other. This paper presents an analysis on produced cues in 5 topics of CLeLfPC, a large corpus of read speech in French with Cued Speech. A phonemes-to-cues automatic system is proposed in order to predict the cue to be produced while speaking. This system is part of SPPAS - the automatic annotation an analysis of speech, an open source software tool. The predicted keys of the automatic system are compared to the produced keys of cuers. The number of inserted, deleted and substituted keys are analyzed. We observed that most of the differences between predicted and produced keys comes from 3 common position's substitutions by some of the cuers.

Keywords: cued speech, corpus, annotation

1. Introduction

The production of speech naturally involves lip movements; the acoustic information as well as the lipreading are part of the phonological representation of hearing people. For a better comprehension every sound of the language should look different but many sounds look alike on the lips when speaking. The term 'viseme' was introduced to refer to mutually confused phonemes that are deemed to form a single perceptual unit (Fisher, 1968; Massaro and Palmer Jr, 1998). In 1966, R. Orin Cornett invented the Cued Speech (Cornett, 1967), a visual system of communication; it adds information about the pronounced sounds that are not visible on the lips. Cued Speech (CS) is a communication system developed for deaf people to complement speech reading at the phonetic level with hands. It uses hand shapes in different placements near the face in combination with the mouth movements of speech to make the phonemes of spoken language look different from each other. Several studies have been conducted on CS to show how it can help speech perception for deaf or hard of hearing persons. It improves speech perception for hearing-impaired people and it offers a complete representation of the phonological system for hearing-impaired people ; among others, see (Nicholls and Mcgill, 1982; Leybaert and Alegria, 2003; Bayard et al., 2019). Cued Speech is then increasingly popular and has been adapted for more than 65 languages[1]. From both the hand position on the face to represent a vowel 'V' and handshapes to represent a consonant 'C', 'CV' syllables are coded. There are named either *keys* or *cues*. A single CV syllable will be generated or decoded through both the lips position and the key of the hand. Each time a speaker pronounces a 'CV' or '-V' syllable, a cue is produced. Other syllabic structures are produced with several cues - for example, a 'CCV' syllable is coded with the two consecutive keys 'C-' then 'CV'. As a consequence, when sounds look alike on the lips, they are cued differently. Thanks to this code, speech reading is encouraged since the Cued Speech keys match all of the spoken phonemes but phonemes with the same viseme have different keys. Once sounds are made visible and look different, it results in a better understanding of speech.

This paper investigates the automation of the production of keys. A rules-based system is proposed and is performed on time-aligned phonemes of CLeLfPC - Corpus de Lecture en Langue française Parlée Complétée (Bigi et al., 2022), a large open source corpus of French Cued Speech. This automatic annotation was manually checked and the differences between the predicted keys and the produced keys are analyzed.

2. French Cued Speech

The modality of cueing provides a level of visual access to deaf and hard-of-hearing people for spoken languages. Because CS fits the phonological level of a given spoken language, each language is cued differently because its CS chart is created from its phonemic representation and it follows the principles of cueing design defined by its inventor (Cornett, 1994).

The French Cued Speech is named "Langue française Parlée Complétée" - LfPC that literally means "Supplemented Spoken French Language". It makes use of the same 8 handshapes (consonants) and 5 hand positions on or around the face (vowels). Table 1 indicates the naming convention of the handshapes and Table 2 the ones of the hand positions. We used the same naming convention as the one of the British CS (BCS), except we propose to name the cheek bone vowel position (b) which does not exist in BCS. In addition, a 9th handshape is identified with (0) and a 6th hand position is identified with (n). They are respectively representing the neutral shape and neutral position. This is

[1] https://www.academieinternationale.org/list-of-cued-languages visited 2022-09

used along with long silences. Figure 1 illustrates both the positions of vowels and the handshapes for all phonemes.

id.	consonants	id.	consonants
(1)	/p/, /d/, /Z/	**(5)**	/m/, /t/, /f/, no consonant
(2)	/k/, /v/, /z/	**(6)**	/l/, /S/, /J/, /w/
(3)	/s/, /R/	**(7)**	/g/
(4)	/b/, /n/, /H/	**(8)**	/j/, /N/

Table 1: Handshapes identifiers and their corresponding consonants in X-SAMPA

id.	vowels	id.	vowels
(s)	/a/, /o/, /9/, /@/, no vowel	**(m)**	/i/, /O∼/, /a∼/
(c)	/E/, /u/, /O/	**(t)**	/y/, /e/, /9∼/
(b)	/e∼/, /2/		

Table 2: Hand position identifiers and their corresponding vowels in X-SAMPA

3. An automatic prediction system for cues

Despite the significant number of studies demonstrating the benefits of Cued Speech, studies on the automatic CS prediction are rather rare. The Massachusetts Institute of Technology has sought to address this problem in its realization of an Automatic Cue Generator (Bratakos, 1995; Sexton, 1997; Bratakos et al., 1998; Duchnowski et al., 1998). In a room, a speaker is filmed speaking without coding and an Automatic Speech Recognizer (ASR) uses the acoustic speech signal to determine which phoneme is being produced. Once the recognition is completed, in another room, the image of the filmed speaker with the synthesis keys according to the rules of the Cued Speech is displayed on a screen to the deaf individual. Several versions of this system were evaluated and it resulted in at least a small benefit to the cue receiver relative to speech-reading alone. However, the way they get the keys is neither fully described nor evaluated separately. Two French projects were also implementing a Text-to-Cued speech synthesizer between 2002 and 2006 but none of them neither described nor distributed the key generator.

In the scope of creating a Text-to-Cued system, the first required *new* step copes with time-aligned phonemes as input and produces an output with the cue names and their corresponding segmentation. Therefore, the problem we are dealing with is close to the syllabification of phoneme sequences we have previously investigated (Bigi et al., 2010). The phoneme sequences need to be automatically converted into key sequences and time-aligned from the corresponding phoneme time-alignments.

At a first stage, we have to create time groups from the time-aligned phonemes. 'Time Group' (TG) refers to an event sequence with a well-defined boundary condition (Gibbon, 2013). In the present context, a TG is an inter-break group where a break is a pause or any sound except a phoneme (laugh, noise, breath, etc).

The structure of CS assumes that a cue represents each CV combination as a handshape (C) and a specified position (V). Each phoneme of TG are then turned into its class: either labelled with C or V.

Given the sequence of class labels of a TG, the algorithm specifies a sequence of handshape-position pairs according to the rules of CS. Special rules are implemented for atypical class combinations such as VC, C, CC and CVC, instead of the regular 'CV' that makes a key. We developed a grammar corresponding to these rules and implemented this grammar in software a deterministic finite automata (DFA). For clarity, we show in Figure 2 the DFA of a single cue. The DFA accepts or rejects an input string of symbols, based on a deterministic algorithm. All states in consideration exist in a finite list and the abstract machine can only take on one of those states at a time.

When the sequence of class labels of a TG is segmented, we turn back the sequence of classes into phonemes. Each phoneme label is then mapped to its key code according either table 1 for a consonant or table 2 for a vowel. It results in a new time-aligned annotation at the CS key level. Figure 3 illustrates an example of such input and output. This automatic process is implemented in a Python package of SPPAS (Bigi, 2015) and distributed under the terms of the GNU GPL v3 license.

4. Dataset: cues annotation

CLeLfPC - Corpus de Lecture en LfPC, is a large open source multi-speaker dataset of Cued Speech (Bigi et al., 2022). It is under the terms of the CC-BY-NC-4.0, the Creative Commons Attribution-Non-Commercial 4.0 International License, and can be used for any research or teaching purpose about CS. The corpus is made of 4 hours of audio/video recordings: it is the largest available corpus of CS data. Among others, this corpus brings the following tangible benefits:

- an HD video quality of the whole speaker;
- 23 different participants, some are CS certified and some are not;
- 10 different topics, each one read by 2 or 3 participants;
- 4 different sessions in each topic: 32 isolated syllables, 32 isolated words or phrases, 7 up to 10 isolated sentences, a text.

Annotations are under construction but some are already available under the terms of the same license. Five different topics read by participants of level 5 (highly experimented) or 6 (CS certified) were annotated. The 4 sessions of all the 5 topics were time-aligned at the phonetic level, following a semi-automatic procedure. Using SPPAS (Bigi, 2015; Bigi and Priego-Valverde, 2019), Inter-Pausal Units - e.g. sounding segments separated by silences, were identified. The orthographic transcription was then performed manually with Praat (Boersma and Weenink, 2018) by the first author of this paper, and the boundaries of the IPUs were manually verified at the same time. The text transcription was automatically normalized and converted to phonemes. The automatic graphemes-to-phonemes conversion results were manually verified then automatically time-aligned with the recording. The resulting time-aligned

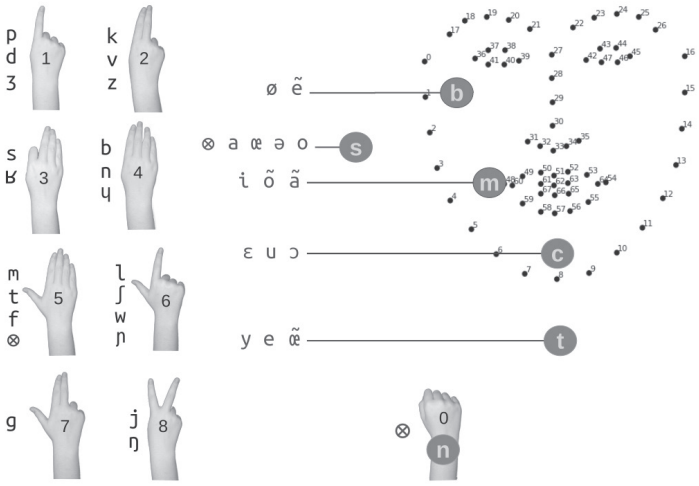


Figure 1: French Cued Speech coding scheme with phonemes in IPA, and a special character to represent "no speech".

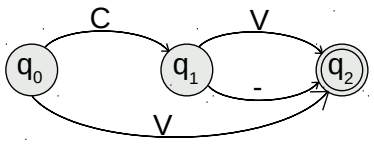


Figure 2: Grammar of a CS key.

phonemes were manually verified with Praat by the first author.

The automatic prediction system for cues was then used in order to get the time-aligned predicted CS keys annotation like illustrated in the first 3 tiers of Figure 3. The videos were viewed in slow motion in order to identify differences between the keys that were predicted by the system and the keys that were coded. It resulted in a new annotation with the time-aligned produced CS keys, represented in the 4th tier of Figure 3. Table 3 indicates the distribution of the 4143 produced keys according to the key structure and session. In addition, 476 neutral handshape and hand position were observed. Table 4 indicates the mean duration

	N	C	V	CV
syllables	165	0	0	159
words	168	187	66	621
sentences	89	309	145	1013
text	54	361	145	1137
total	476	**857**	**356**	**2930**
percent		*20.69%*	*8.59%*	*70.72%*

Table 3: Produced cues in 5 topics of CLeLfPC.

and standard deviation of the produced keys. The 'i1' and 'i2' flags refer to the following reading instructions given to CLeLfPC cuers:

i1 the syllables and the words/phases have to be read clearly, like to teach CS to someone else;

i2 the sentences and the text should be read as naturally as possible, like to tell or read someone a story.

		CV		C	V
		μ	stdev	μ	μ
i1	syllables	0.354	0.105		
i1	words	0.317	0.104	0.182	0.194
i2	sentences	0.268	0.085	0.153	0.170
i2	text	0.250	0.085	0.137	0.154

Table 4: Duration in 5 topics of CLeLfPC.

Perhaps somewhat unsurprisingly, the average duration highlights differences between 'i1' and 'i2'. Duration of 'i2' are about 25% lower than those of 'i1'. It has to be noticed that these are the duration of the phonemes clusterized into cues like illustrated in Figure 3, not the duration of the cues themselves.

5. Predicted versus produced keys

The aims of a comparison between the predicted keys and the produced ones by cuers are twofold. On the one hand, this analysis could reveal implicit rules, i.e. rules of common use that constitute exceptions to the rules of the general definition in order to implement a prediction system closer to the real coding habits. On the other hand, it allows to describe the CS coding as it is practiced, quantifying errors and qualifying them.

We firstly compared the annotations quantitatively. The differences are stated below according 3 categories:

insertion The cuer added 8 keys compared to the predicted ones;

deletion The cuer did not code 47 keys compared to the predicted ones;

substitution The cuer and the prediction system coded 183 keys differently.

The number of inserted and deleted keys is very small relatively to the number of substitutions, and almost anecdotal relatively to the corpus size.

Table 5 shows the details of such differences for each one of the 5 speakers. We can observe that for two of them (AM,

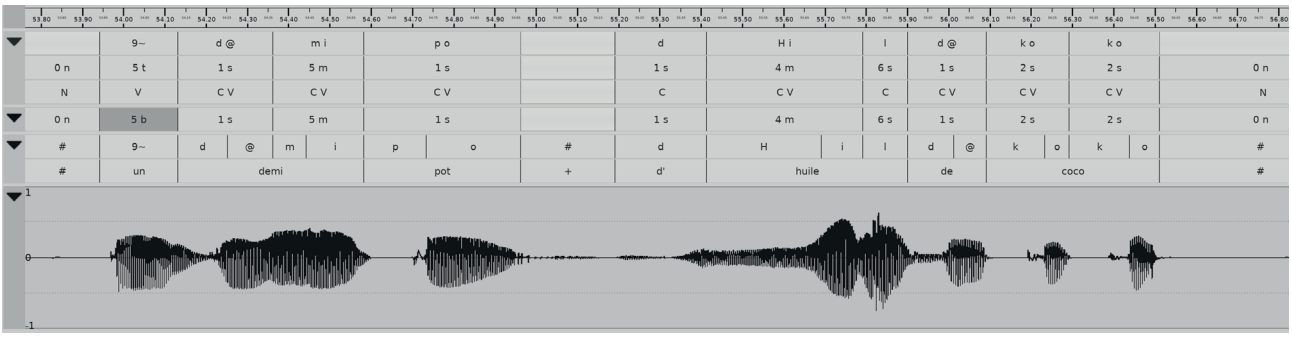


Figure 3: Annotations and waveform of two TG extracted from CLeLfPC. From bottom to top: tokens, phonemes, manually checked CS code, automatic CS keys structure, automatic CS code, automatic CS keys.

ML) there's only a few number of differences, which means that the predicted system and these speakers are consistent in their key production. A detailed analyses of the difference will give some clues to understand in which specific situations the other three speakers are coding differently.

speaker:	CH	VT	AM	ML	LM
insertion	1	4	2	0	1
deletion	16	2	7	4	18
substitution	35	74	12	6	56

Table 5: Produced keys that don't match the predicted ones, depending on the cuer

5.1. Insertion

Among the 8 inserted keys, three are errors of the cuer but five are related the liaison phenomenon. For example, the tokens "pour un" (*for a*) is pronounced /puR9∼/ then the automatic system predicts a sequence of two keys: /pu/ and /R9∼/. However, the cuer is coding a sequence of three keys corresponding to: /pu/ then /R/ then /9∼/. In this case, both coding solutions are acceptable, but this situation is very rare, so it does not need to be taken into account into the prediction system.

5.2. Deletion

Among the 47 keys the cuer did not code compared to the predicted ones, 41 are 'C' and 6 are 'CV'. So, isolated vowels are always coded which is not surprising given that they are the nucleus of syllables. Only 3 of the un-coded sounds are related to the instruction 'i1', so the high majority were from sentences and text. The removed 'C' keys are during 0.102 seconds in average which represents 67% of the average duration of the coded ones. However, 32% of the coded 'C' are during less than 0.102 seconds. As a consequence, we can observe that the un-coded isolated consonants are frequently short but it does not make it a rule for a prediction system because the majority of the short isolated 'C' are coded. We can formulate the hypothesis that, sometimes, the cuer has not had enough time to move the hand at the side position with the expected handshape. As shown in Table 5, among the 5 cuers, two are significantly un-coding the consonants: 18 deletion for LM and 16 for CH. The most frequently un-coded consonants are /t/ (9 times), /R/ (8 times), /p/ (5 times) and /l/ (4 times).

5.3. Substitution

Key substitutions are representing 4.4% of the produced keys, so their analysis is important, particularly because it has never been done in previous studies on CS. As shown in Table 5, three cuers (CH, VT, LM) are producing 90% of the substitutions. Like before, we observe an effect of the instruction. None of the substitutions are occurring during the syllable sessions and only 24 are occurring during the word ones. The high majority of substitutions is from sentences (65) and text (94).
Among the 183 substitutions, 16 are 'C' (8.7 %), 22 (12 %) are 'V' and 145 (79.2 %) are 'CV'. Proportionally to their frequency, it seems that substitutions mostly concern the position (the vowel) than the handshape (the consonant). This tendency is confirmed by the following detailed analysis of the predicted 'CV' keys compared to the produced ones. Among the 145 'CV' cued keys that don't match with the predicted ones:

- 1 substitutes both the shape and the position;
- 6 substitute the shape only;
- 138 substitute the position only.

In the end, we observed 160 vowel substitutions among the 183 referenced ones, that is 87.4 % of the substitutions, 3.86 % of the produced cues of the corpus. A position substitution therefore represents the major difference between predicted and produced keys.
A large number of the vowel substitution (88, that is 48 %) concerns the phoneme /@/ which is coded at position (b) instead of (s). The (b) position is the one of the vowel /2/ but /2/ is never coded at (s) position like /@/. When phonetically realized, schwa (/@/) is a mid-central vowel with some rounding. Many authors consider it to be phonetically identical to /2/ (Anderson, 1982). In the internal position, the acoustic analysis carried out in the reading of a list of words demonstrated the quasi-acoustic identity (Racine et al., 2016). The differences with /2/ are that schwa duration is reduced or that it can be omitted. Such reduction of schwa in French highly depends on the accent: schwa is one of the phenomena that makes it possible to differentiate the northern and southern varieties of French. We observed

that two cuers are significantly coding /@/ at (b) position: VT 45 times and LM 40 times; however VT coded it at (s) position 55 times and LM 45 times like expected by the key rules production. We then sought to understand why they use both solutions (s) and (b), and we found the answer by looking at the words:

- LM: *"de"* is 14 times at (b) against 2 times at (s);
- VT: *"de"* is 11 times at (b) against 5 times at (s);
- VT: *"le"* is 10 times at (b) and never at (s);
- LM: *"le"* is 6 times at (b) against once at (s);
- VT: *"ne"* is 4 times at (b) and never at (s);
- LM: *"que"* is 4 times at (b) and never at (s).

Another significant substitution concerns the vowel /e/ which is coded 32 times at position (c) instead of (t). The (c) position is the one of the vowel /E/. Here again, two speakers are mostly coding this way: 18 times VT and 9 times CH. However, we did not observed any particular trend that could explain this difference in coding. We only found that it affects some words more that others but not systematically. These words are: *c'est* (6 times), *les* (5 times), *ses* (4 times) and *des* (4 times).
The last significant substitution concerns the vowel /9∼/ which is coded at position (b) 17 times instead of (t). The (b) position is the one of the vowel /e∼/. This difference is mainly observed in the word *un* of CH speaker (12 times) who is coding this word only 2 times in (t).

6. Discussion and Conclusion

This paper presented an automatic system to predict CS keys from phonemes. An automatic annotation of cues was performed on 5 topics of CLeLfPC, a large open source corpus of French Cued Speech. This annotation was manually verified to obtain the keys produced by the cuers. An analysis of the differences between the predicted keys and the produced ones allowed to validate the automatic system: this analysis did not reveal implicit rules. Moreover, there is few information on how CS is produced by human coders, so this paper has contributed in this area. This study highlighted some cuer habits. The most significant difference comes from position substitution of some specific phonemes in some specific words by some of the cuers.
Next work will focus on the analysis of duration and timing of the sequences of cues in the time-groups and on the temporal and spatial organization of the code in its speech co-production. It will require to manually re-check the time-alignment of phonemes by an expert phonetician and to time-align the keys with the video in order to annotate the moments they are produced.

References

Anderson, S-R, 1982. The analysis of french shwa: or, how to get something for nothing. *Language*:534–573.

Bayard, C, L Machart, A Strauß, S Gerber, V Aubanel, and J-L Schwartz, 2019. Cued speech enhances speech-in-noise perception. *The Journal of Deaf Studies and Deaf Education*, 24(3):223–233.

Bigi, B, 2015. SPPAS - Multi-lingual Approaches to the Automatic Annotation of Speech. *The Phonetician*, 111–112:54–69, `http://sppas.org/`.

Bigi, B, C Meunier, I Nesterenko, and R Bertrand, 2010. Automatic detection of syllable boundaries in spontaneous speech. In *Language Resource and Evaluation Conference*. La Valetta, Malta.

Bigi, B and B Priego-Valverde, 2019. Search for interpausal units: application to cheese! corpus. In *9th Language & Technology Conference*. Poznań, Poland.

Bigi, B, M Zimmermann, and C André, 2022. CLeLfPC: a Large Open Multi-Speaker Corpus of French Cued Speech. In *The 13th Language Resources and Evaluation Conference*. Marseille, France.

Boersma, P and D Weenink, 2018. Praat: doing phonetics by computer [computer program], version 6.0.37, retrieved 14 march 2018 from http://www.praat.org/.

Bratakos, M-S, 1995. *The effect of imperfect cues on the reception of cued speech*. Ph.D. thesis, Massachusetts Institute of Technology.

Bratakos, M-S, P Duchnowski, and L-D Braida, 1998. Toward the automatic generation of cued speech. *Cued Speech Journal*, 6:1–37.

Cornett, R-O, 1967. Cued speech. *American annals of the deaf*:3–13.

Cornett, R-O, 1994. Adapting cued speech to additional languages. *Cued Speech Journal*, 5:19–29.

Duchnowski, P., L.-D. Braida, M.-S. Bratakos, D.-S. Lum, M.-G. Sexton, and J.-C. Krause, 1998. A Speechreading aid based on phonetic ASR. In *5th International Conference on Spoken Language Processing*. Sydney, Australia.

Fisher, C-G, 1968. Confusions among visually perceived consonants. *Journal of speech and hearing research*, 11(4):796–804.

Gibbon, D, 2013. TGA: a web tool for Time Group Analysis. In *Proceedings of the Tools and Resources for the Analysis of Speech Prosody (TRASP) Workshop*. Aix-en-Provence, France.

Leybaert, J and J Alegria, 2003. The role of cued speech in language development. *Oxford handbook of deaf studies, language, and education*, 1:261.

Massaro, D-W and S-E Palmer Jr, 1998. *Perceiving talking faces: From speech perception to a behavioral principle*. Mit Press.

Nicholls, G-H and D-L Mcgill, 1982. Cued speech and the reception of spoken language. *Journal of Speech, Language, and Hearing Research*, 25(2):262–269.

Racine, I, J Durand, and H N Andreassen, 2016. PFC, codages et représentations: la question du schwa. *Corpus*, 15.

Sexton, M-G, 1997. *A video display system for an automatic cue generator*. Ph.D. thesis, Massachusetts Institute of Technology.

Combining plain language and machine translation for science communication

Lynne Bowker[1,2]

[1]School of Translation and Interpretation, University of Ottawa, Canada, and
[2]NAWA Visiting Researcher, Scholarly Communication Research Group, Adam Mickiewicz University, Poland
lbowker@uottawa.ca

Abstract

The scholarly community is working to make research more open. This includes making findings more accessible to those outside the scientific community (i.e., science communication), as well as encouraging multilingual scholarly communication. Such efforts are often taken independently, but this paper brings them together to investigate whether combining plain language and machine translation (MT) could both amplify the reach of science communication and increase linguistic diversity. We conduct a pilot study to see whether plain language summaries written by researchers are more MT friendly than their corresponding scientific abstracts. We compare back-translations of both the summaries and the abstracts against their original versions and find that the back-translations of the plain language summaries are more readable and contain fewer meaning errors. This suggests that it is worth further investigating the combination of plain language and MT for science communication. Next steps could be to prepare a set of integrated guidelines combining tips for plain language summary writing and writing for MT, to conduct user testing, and to investigate other AI tools.

Keywords: machine translation, multilingual scholarly communication, open science, plain language, science communication

1. Introduction

UNESCO's (2021) Recommendation on Open Science outline core values and guiding principles, including 2 key points of interest for our research. First, UNESCO (2021, p. 4) encourages researchers "to open the processes of scientific knowledge creation, evaluation and communication to societal actors beyond the traditional scientific community" (e.g. policy makers, funders, practitioners, citizens), many of whom are not familiar with specialized research terms. Second, UNESCO (2021, p. 9) notes that open science must embrace linguistic diversity by "encouraging multilingualism in the practice of science, in scientific publications and in academic communications". UNESCO is not the only group to endorse these values. The Helsinki Initiative (2019) on Multilingualism in Scholarly Communication also notes that researchers should be recognized for sharing research results beyond academia, and emphasizes that access to research results must be provided in multiple languages.

In general, science communication and multilingual scholarly communication are treated as important but independent elements of open science. However, this paper brings them together to present a pilot study that investigates whether combining plain language summaries and machine translation (MT) could support open science.

2. Linguistic challenges and potential solutions in open science

Science communication and multilingual scholarly publishing both involve a form of translation. For science communication, this involves intralingual knowledge translation, where complex ideas used by scientists are re-expressed in a way that can be understood by non-experts (Hanauska, 2019; Rushmer et al., 2019), such as by producing a plain language summary. For multilingual scholarly publishing, if all researchers publish in their own language, then interlingual translation is needed to enable speakers of other languages to discover and read this work. Professionally translating every research article into every language is not feasible, so we need options such as MT for support; however, MT comes with its own challenges.

2.1. Plain language summaries

Plain language writing aims to help readers to understand a text as quickly, easily and completely as possible the first time they read it (Cutts, 2009). Oft-cited best practices for plain language writing include using shorter sentences, avoiding jargon, defining terms, explaining abbreviations, and reducing ambiguity (Maurer et al., 2021). While it is not a new concept, plain language has gained global momentum in recent years, especially with governments, who want to ensure that communications with the general public are easier to understand. For example, Former US President Barack Obama signed the Plain Writing Act of 2010 requiring federal agencies to use clear language; the European Commission (2011) began a "Clear writing for Europe" drive to encourage shorter, simpler, jargon-free texts; and experts at the University of Wrocław established the *Pracownia Prostej Polszczyzny* (Plain Polish Lab) to develop Plain Polish and a plain language checker for use in the public and private sectors (Piekot et al., 2019). Other nations (e.g. UK, Canada) are taking similar actions.

The use of plain language summaries is becoming more common in scientific journals too, especially for medical research (Maurer et al., 2021). Plain language summaries do not replace traditional scientific abstracts but appear alongside them, or on a journal's public facing site. Cited reasons for creating plain language summaries include:

- Respecting conditions of public funding agencies;
- Enabling participants in community-based research projects to have access to the findings;
- Making information available to policy makers;
- Supporting students who are new to a field.

But there is another reason to prepare plain language summaries: they may be easier for scientists who are

Contents

reading in a second language to understand. Moreover, if we accept that plain language is easier for people to read and understand, then it should also be easier to translate.

2.2. Machine translation

MT systems try to translate texts automatically from one language to another (e.g. English to Polish). Different approaches have been tried, but the most current is a data-driven approach known as neural MT (NMT) (Koehn, 2020). Developers feed the NMT system with a huge training corpus of parallel texts that have been translated previously by professional translators. Using machine learning, the system's artificial neural network learns to translate new texts by consulting the training corpus.

Data-driven approaches are also data sensitive, so the training corpus plays a key role in determining the quality of an NMT system's output. The training corpus should be carefully selected, which means finding texts not only in the needed language pair but also of the same text type, on the same subject, etc., as the new texts to be translated. Finding the right balance and representativeness for the training corpus is tricky when developing "try anything" systems (e.g. free online NMT systems). NMT tools that have been adapted for a specific domain are more likely to produce better translations in that domain (Chu and Wang, 2018), but it is hard to find large parallel corpora on specialized scientific subjects since most scholars publish in English. In contrast, science communication texts are more likely to be produced in other languages too.

Sensitivity to training data can be an issue, but NMT output quality has still greatly increased as compared to earlier approaches. However, the output may still need to be verified or post-edited to reach the desired level of quality. While post-editing is common, another way to improve the output quality is to improve the input.

Historically, MT systems have been more successful when translating controlled language (CL), which has a highly restricted vocabulary and syntax (Miyata, 2021). CL seeks to minimize ambiguity to improve translation accuracy, but resulting translations may not score highly for readability. Marzouk and Hansen-Schirra (2019) show that while using CL rules has a positive impact on rule-based and statistical MT, this is not true for NMT, where the output quality drops after CL rules are applied. This drop may occur because the text was more idiomatic before the CL rules were applied. The atypical style of CL is unlikely to be well represented in the corpora used to train free online MT systems. The present paper explores plain language, rather than CL. Plain language is more idiomatic than CL, and since it has become more widely used in the past decade, plain language is more likely to be in training corpora for free online MT systems.

3. Methods

This study focuses on whether a combination of plain language and MT can benefit researchers broadly speaking. Therefore, it employs tools and techniques that are easily available to and familiar to most researchers.

3.1. Selection of the language pair

English dominates scholarly publishing, and achieving multilingualism will need a multipronged solution. Writing more plain language summaries—even in English—may increase linguistic diversity if they are more MT-friendly than scientific abstracts. MT-friendly English summaries could be translated into a reader's preferred language, thus diversifying linguistic access. This pilot study explores this potential using English as the source language.

Polish was chosen as the pilot study target language for several reasons. Firstly, it is well known that MT performs better for widely used (i.e., high-resource) languages and for related language pairs. Yet many scholars speak low(er)-resource languages and/or languages not closely related to English. To test whether combining MT and plain language is a viable strategy for a broad range of scholars, I sought a low(er)-resource language from a different family which also features in a free online MT tool. Finally, Polish scholars co-founded the Helsinki Initiative (2019) and are working to build a multilingual publishing ecosystem (e.g. Kulczycki et al., 2020).

3.2. Selection of the corpus

For this study, I needed English-language scientific journals that include plain language summaries alongside scientific abstracts. Canadian Science Publishing (https://cdnsciencepub.com/) publishes 23 journals in various scientific fields and also maintains a public website (https://medium.com/@cdnsciencepub) hosting plain language summaries of the journal articles.

The publisher's high-level guidelines for all its journals include information about plain language summaries (https://cdnsciencepub.com/authors-and-reviewers/writing-a-plain-language-summary), noting that such summaries are optional, but encouraged as a means of increasing accessibility and readership of the article. The site provides basic guidelines for writing a plain language summary and links to toolkits. It also explains how authors can submit and share summaries (e.g. via social media).

Canadian Science Publishing launched its plain language summary site in 2016 and it now has over two hundred summaries, but these come from only 3 of the 23 journals: *Arctic Science (AS)*, *Canadian Journal of Plant Science (CJPS)* and *Facets (F)* (a multidisciplinary science journal). More digging shows that only these 3 journals also put information about plain language summaries in their journal-specific guidelines. For the initial test, I randomly selected 3 summaries from each of the 3 journals (total = 9) and then located the corresponding abstracts.

3.3. Selection of the machine translation system

Researchers beyond computational linguistics use free online MT tools rather than in-house or subscription-based tools. For ecological validity, I chose a free online tool in spite of known limitations (e.g. no domain adaptation). For this pilot study, I chose DeepL Translator, which is well known in Poland (the founder and CEO of DeepL is Polish researcher Jarosław Kutyłowski), and which has outperformed Google Translate and Microsoft Translator for English-Polish in prior evaluations (e.g. Kur, 2019).

3.4. Selection of the evaluation method

Translation quality is notoriously difficult to measure, and multiple methods have been devised to evaluate MT output, including methods relying on human evaluations, those using automated metrics (e.g. BLEU, METEOR, TER) and hybrid methods (Rossi and Carré, 2022). Yet

applying such methods is labor-intensive and may require computational linguistics knowledge. This pilot study asks whether combining MT and plain language summaries can help typical researchers, and so it was vital to select tools and methods that are familiar and available to them. I chose MS-Word and back-translation, which uses an MT tool to translate a text from source to target language (EN>PL), and then retranslates the resulting target text back into the original source language (PL>EN).

As a means of measuring translation quality, back-translation has some limitations and has been criticized by various scholars, including Somers (2005), who notes that it is not possible to tell from the back-translation whether an error was introduced during the first or the second phase, and that any errors occurring in the first translation will be carried forward in the second. I fully agree that back translation does not offer a rigorous and detailed means of evaluating translation quality. However, this pilot study's aim is more modest: to establish in a more impressionistic way whether plain language summaries seem to be more translation friendly than abstracts. This task is more about rough quality estimation than about precise quality evaluation, and so back-translation may be sufficient for this purpose. While Somers (2005) disagrees that back-translation can be used as a rough quality estimator, he acknowledges that his results relied on comparing back-translations to BLEU scores, which themselves have limitations (e.g. operating at sentence level, using one human reference translation as a gold standard). Somers' study also used statistical MT rather than NMT. In NMT, back-translation has been shown to be a viable method for improving low-resource language models (e.g. Sennrich et al., 2016; Hoang et al. 2018).

Researchers outside computational linguistics are not likely to test the translation quality of their plain language summaries using rigorous computational linguistics metrics, but they do use back-translation when using MT as a writing aid. Sun et al. (forthcoming) report that Chinese scholars regularly use back-translation to judge target text quality when working with free online MT tools to create English-language abstracts and in an experiment using back-translation between Japanese and English, Shigenobu (2007) found that the quality of translated and back-translated sentences correlates, "therefore, users may be able to estimate the quality of outward translation by back translation." So while back-translation is not a rigorous measure of translation quality, researchers creating plain language summaries may well use it as a convenient method of roughly estimating a target text's usefulness. In addition, plain language summaries (whether originals or translations) are not intended to be used as the sole decision-making tool for further action. Rather, plain language summaries are more likely to serve as tools for enabling readers to determine the relevance of a study to their needs. If it seems relevant, then readers can go on to learn more, and if it seems irrelevant, then readers can set it aside. Nurminen (2019) reports on the usefulness of MT as a filter to identify texts that merit more scrutiny.

3.5. Translating and comparing texts

Using the free online version of DeepL Translator, I translated all of the English-language texts (9 plain language summaries and 9 abstracts) into Polish. I then back-translated all the Polish texts into English.

I copied the texts into four MS-Word documents: original and back-translated abstracts, and original and back-translated summaries. I used Word's readability checker to find average sentence length, percentage of passives and Flesch-Kincaid Grade Level (FKGL). To spot differences between original English texts and English back-translations, I used Word's "Compare" feature to compare documents and highlight differences. I then sorted the differences into two broad categories with reference to MQM (2021) framework accuracy metrics:

- Differences that change the meaning;
- Differences that do not affect the intended meaning.

4. Results

Key results are summarized in Tables 1, 2 and 3.

	Abstracts		Summaries	
	Original	Back-tr	Original	Back-tr
Total words	1839	1864	2946	2984
Av length	204.3	207.1	327.3	331.5

Table 1. Total words and average length in words.

	Abstracts		Summaries	
	Original	Back-tr	Original	Back-tr
Passive	28%	32%	20%	25%
Av sent length	23.2	23.2	21.2	22.4
FKGL	16.2	16	13.6	13.8

Table 2. Readability scores calculated by MS-Word.

	Back-translations of abstracts		Back-translations of summaries	
	# of meaning errors	# of surface changes	# of meaning errors	# of surface changes
Total	34	205	17	320
Per 100w	1.82	10.99	0.57	10.72

Table 3. Errors and changes in abstracts and summaries.

5. Discussion

As a pilot study, this investigation has several limitations. It involves just one language pair, a small test corpus, and a single MT system. Future iterations could include other languages, more texts and other tools. Moreover, as a quality estimation method, back-translation has some weaknesses, so more testing could be done using other metrics. However, this pilot study is useful as a proof of concept in that it shows that plain language summaries appear to be more translation friendly than conventional scientific abstracts; therefore, further investigation into the combined use of plain language and MT seems warranted.

As Table 1 shows, plain language summaries in our sample are an average of 37.5% longer than abstracts. While a longer length might seem to offer more chances to make translation errors, it also allows writers to clarify content. Abstracts must often meet strict word limits, forcing authors to write in a dense way that is difficult to parse (e.g. noun stacking, omitting relative pronouns). Longer summaries allow authors to be clear and explicit, and the reduced ambiguity can improve translation quality.

Table 2 indicates that the plain language summaries in our sample receive higher scores for readability than do

the abstracts as measured by a smaller number of passive constructions, a shorter average sentence length, and a lower FKGL score (where the score corresponds to the number of years of formal education required to understand the text). In the case of both the abstracts and the plain language summaries, the original texts score more highly for readability than do their corresponding back translations, suggesting that the double translation (EN>PL>EN) inserts some complexity into the constructions; however, in both cases, the plain language summaries (both original and back-translations) remain more readable than the abstracts. Nevertheless, while the plain language summaries are clearly and consistently more readable, we might have expected the difference between the abstracts and the summaries to have been greater. The fact that the summaries are only moderately more readable suggests room for improvement, and it is possible that increased readability could be obtained by providing more plain language guidance to authors.

Table 3 shows that although there are a relatively high number of changes produced as part of the process of translation and back-translation for both abstracts and summaries, the number of changes that result in a meaning error are consistently higher in the case of the scientific abstracts and lower in the case of the plain language summaries. This is true in terms of raw number of errors – in our sample, the number of meaning errors in the abstracts is twice as high as the number in the summaries – and it is even more striking when the error rate per 100 words is calculated. On average, there are 1.82 meaning errors per 100 words in the abstracts and only 0.57 in the summaries, meaning that there are 68.68% fewer meaning errors in the summaries as compared to the abstracts.

Moreover, in some cases, the meaning error contained in the plain language summary is the same as that appearing in the corresponding abstract because the author did not adapt that segment of text. For example, one abstract contains the phrase "safe land travel activities" which has been wrongly translated as "safe travel and land-based activities". Since the author retained the same phrase in the plain language summary, the same error occurred there also. In another, several Latin species names were mistranslated in both the abstract and summary. In the abstracts, other differences between the original texts and their back translations that pointed to meaning errors included *mandates > fines; judiciously > reasonably; fluke > fin; synthesis > synthetic; linking > combining*. Meanwhile, some meaning errors in the plain language summaries can be identified as originating with terms that are rather specialized and could perhaps have been simplified: *georeferenced > placed; viticulture > vine size*. As noted above, better guidelines for writing plain language summaries could likely improve translation quality with regard to accuracy as well as readability.

Both the abstracts and plain language summaries revealed a larger number of superficial differences between the original versions and the back translations; however, this should not be identified as a problem per se since these differences represented surface-level changes that did not affect the underlying message of the text. It is well known that two human translators will rarely produce exactly the same translation of a given source text, and so it is not surprising to see superficial differences arising following the use of an MT system. Examples of differences that did not impact the meaning of the text include *happen > occur; paper > article; asymptomatic > without symptoms; co-develop > jointly develop*.

6. Conclusion

While the ultimate goal is to achieve a fully multilingual scholarly communication ecosystem, this will require a multipronged and multistage approach. As an early step, creating plain language summaries in any language, including English, can help. Greater availability of summaries will advance open science by making it easier for non-experts to access research results, as well as by making it more feasible for people to access these summaries in their own language via MT.

Next steps include studying guidelines for preparing plain language summaries (e.g. from Canadian Science Publishing and elsewhere) and comparing them to the plain language writing guidelines prepared for other sectors, as well as recently published guidelines that specifically target for writing for MT (e.g. Translation Centre for the Bodies of the European Union, 2021). By comparing and testing these 3 types of guidelines, we can develop a more tailored set of guidelines to enable scholars to maximize the combined benefits of plain language and MT. In other words, by integrating guidelines for preparing plain language summaries and guidelines for improving MT friendliness, we may be able to amplify the value of these summaries for speakers of other languages.

Looking further ahead, it will be useful to test any initial guidelines developed on the basis of English to determine their portability to other languages and to identify gaps or adaptations needed to make them useful for other languages. Experience with controlled language shows that while some guidelines can transcend language boundaries, others may be language specific (e.g. Hartley et al., 2012). Customized guidelines for other languages will allow researchers to prepare MT-friendly plain language summaries in those languages, and these summaries could then be machine translated into English and other languages. A multilingual set of guidelines that integrates suggestions for writing in plain language that is also MT friendly will be a useful step towards achieving a more multilingual scholarly communication ecosystem.

Both an enhanced set of guidelines and the summaries produced by following these guidelines must eventually be user tested. User testing of the guidelines will determine how easily they can be applied by authors. User testing of summaries by end users is needed because a high readability score does not necessarily mean that the texts are understandable or useful to readers.

Finally, it will be interesting to investigate other tools, including the latest generation of AI technologies (e.g. ChatGPT and similar tools using machine learning), to determine how well these tools can support multilingual plain language summary production. For example, we can see multiple paths for arriving at multilingual summaries:

- L1 original text > L1 summary > L2 translation;
- L1 original text > L2 translation > L2 summary;
- L1 original text > L2 summary.

More research is needed to determine the best path for producing high-quality summaries in many languages. Likewise, more research is needed to determine whether these can best be achieved by using a pipeline of tools (e.g. output of an automatic summarization tool becomes input to an MT tool, or vice versa), or asking ChatGPT to take a

direct approach (e.g. summarizing an L1 text directly in L2). Jiao et al. (2023) report that ChatGPT translations are currently inferior to translations done by Google Translate and DeepL Translator for low resource languages and domains, but things are evolving quickly. Meanwhile, Zhang et al. (2023) show that the formulation of prompts affects translation quality in large language models.

Acknowledgements

Research funded by NAWA (Polish National Agency for Academic Exchange). Thanks to members of the Scholarly Communication Research Group, Faculty of English, and Faculty of Mathematics and Computer Science at Adam Mickiewicz University in Poznań for fruitful discussions.

References

Chu, C. and Wang, R. (2018). A Survey of Domain Adaptation for Neural Machine Translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1304–1319, Santa Fe, New Mexico, USA. Association for Computational Linguistics. https://aclanthology.org/C18-1111

Cutts, M. (2009). *Oxford Guide to Plain English, 3rd edition*. Oxford: Oxford University Press.

European Commission (2011). How to Write Clearly. Brussels.https://ec.europa.eu/info/departments/translation/clear-writing-for-europe_en

Hanauska, M. (2019). Historical Aspects of External Science Communication. In: Leßmöllmann, A. Dascal, M. and Gloning, T. (Eds.) *Science Communication*, pp. 585-600. Berlin: De Gruyter Mouton.

Hartley, A., Tatsumi, M., Isahara, H., Kaguera, K., Miyata, R. 2012. Readability and Translatability Judgements for Controlled Japanese. *Proceedings of the 16th European Association for Machine Translation Conference*, 237-244. https://aclanthology.org/2012.eamt-1.57/

Helsinki Initiative. (2019). Helsinki Initiative on Multilingualism in Scholarly Communication. Helsinki: Federation of Finnish Learned Societies, Committee for Public Information, Finnish Association for Scholarly Publishing, Universities Norway, European Network for Research Evaluation in Social Sciences & Humanities. https://doi.org/10.6084/m9.figshare.7887059

Hoang, V.C.D., Koehn, P., Haffari, G. and Cohn, T. 2018. Iterative Back-Translation for Neural Machine Translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, 18–24, Melbourne, Australia. Association for Computational Linguistics. https://aclanthology.org/W18-2703

Jiao, W., Wang, W., Huang, J., Wang, X., and Tu, Z. 2023. Is Chat GPT a Good Translator? A Preliminary Study. Preprint on arXive. https://arxiv.org/abs/2301.08745

Koehn, P. (2020). *Neural Machine Translation*. Cambridge: Cambridge University Press.

Kulczycki, E., Guns, R., Pölönen, J., Engels, T. C. E., Rozkosz, E., Zuccala, A. A., Bruun, K., Eskola, O., Istenič Starčič, A., Petr, M. Sivertsen, G. (2020). Multilingual Publishing in the Social Sciences and Humanities: A Seven-Country European Study. *J. of the Assoc. for Info Science & Technology* 71:1371–1385.

Kur, M. (2019). Method of Measuring the Effort Related to Post-Editing Machine Translated Outputs Produced in the English>Polish Language Pair by Google, Microsoft and DeepL MT Engines: A Pilot Study. *Beyond Philology: International Journal of Linguistics, Literary Studies and English Language Teaching* 16(4): 69-99.

Marzouk, S. and Hansen-Schirra, S. (2019). Evaluation of the Impact of Controlled Language on Neural Machine Translation Compared to Other Architectures. *Machine Translation* 33: 179-203.

Maurer, M., Siegel, J., Firminger, K., Lowers, J., Dutta, T. and Chang, J. (2021). Lessons Learned from Developing Plain Language Summaries of Research Studies. *Health Literacy Research and Practice* 5(2): 155-161.

Miyata, R. (2021). *Controlled Document Authoring in a Machine Translation Age*. London: Routledge.

Multidimensional Quality Metrics (MQM). (2021). https://themqm.org/

Nurminen, M. (2019). Decision-making, Risk, and Gist Machine Translation in the Work of Patent Professionals. In *Proceedings of the 8th Workshop on Patent and Scientific Literature Translation*, pp. 32–42, Dublin, Ireland. European Association for Machine Translation. https://aclanthology.org/W19-7204/

Piekot, T., Zarzeczny, G., and Moroń, E. (2019). Standard 'plain language' w polskiej sferze publicznej. In: M. Zaśko-Zielińska and K. Kredens (Eds.). *Lingwistyka kryminalistyczna. Teoria i praktyka*, 197–214. Wrocław: Quaestio.

Rossi, C. and Carré, A. (2022). How to Choose a Suitable Neural Machine Translation Solution: Evaluation of MT Quality. In Kenny D. (Ed.) *Machine Translation for Everyone: Empowering Users in the Age of Artificial Intelligence*, 51-79. Berlin: Language Science Press.

Rushmer, R., Ward, V., Nguyen, T., and Kuchenmüller, T. (2019). Knowledge Translation: Key Concepts, Terms and Activities. In: M. Verschuuren and H. van Oers (Eds.) *Population Health Monitoring*, 127–150. Cham: Springer.

Sennrich, R., Haddow, B. and Birch, A. 2016. Improving Neural Machine Translation Models with Monolingual Data. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Vol. 1)*, 86–96, Berlin, Germany. Association for Computational Linguistics. https://aclanthology.org/P16-1009

Shigenobu, T. (2007). Evaluation and Usability of Back Translation for Intercultural Communication. In Aykin, N. (Ed.). *Usability and Internationalization*, Part II, 259–265. Berlin: Springer.

Somers, H. (2005). Round-Trip Translation: What Is It Good For? In *Proceedings of the Australasian Language Technology Workshop 2005*, pp. 127–133, Sydney, Australia. https://aclanthology.org/U05-1019/

Sun, Y-C., Yang, F-Y. and Liu, H-J. (forthcoming). Exploring the Process and Strategies of Chinese-English Abstract Writing Using Machine Translation Tools. *Journal of Scholarly Publishing*.

Translation Centre for the Bodies of the European Union. (2021). Writing for Machine Translation. Luxembourg. https://cdt.europa.eu/en/news/writing-machine-translation

UNESCO. (2021). Recommendation on Open Science. https://unesdoc.unesco.org/ark:/48223/pf0000378841

United States Government. 2010. H.R.946 – Plain Writing Act of 2010. https://www.congress.gov/bill/111th-congress/house-bill/946/text

Zhang, B., Haddow, B. and Birch, A. Prompting Large Language Model for Machine Translation. Preprint on arXive. https://arxiv.org/abs/2301.07069

Emotion Signals for Sexist and Offensive Language Detection: A Multi-task Learning Approach

Alexandra Ciobotaru,[1] Diana Constantina Höfels,[2] Ștefan Daniel Dumitrescu[3]

[1]University of Bucharest and DRUID AI, alexandra.ciobotaru@unibuc.ro
[2]University of Tübingen, diana-constantina.hoefels@uni-tuebingen.de
[3]Adobe, sdumitre@adobe.com

Abstract

Identification and classification of sexist and offensive content in social media posts present a great deal of complexity and relevance. Detecting and identifying this type of language is more difficult due to the presence of multiple forms of sexist and offensive language. We employ a multi-task learning framework to link emotion detection to sexist and offensive language classification, allowing the two tasks to complement each other. The results of our study demonstrate that the use of emotion signals aids the performance of sexist and offensive language detection - we achieve an F1-score of 87.98% by fine-tuning the Romanian BERT, which becomes the state of the art for sexism and offensive detection in Romanian short texts.

Keywords— MTL, sexist and offensive language detection, emotion detection, BERT, Romanian corpora, low-resourced languages

1. Introduction

Social media platforms have profoundly altered the manner in which we communicate, and these shifts have given rise to egregious practices online, such as the use of offensive or sexist language. A sexist act is a discrimination against a person on the basis of their gender. A wide range of sexist language forms, both overt and covert, affect mostly women and girls across many areas of life, including the workplace (Verniers and Vala, 2018), politics, society, household responsibilities, and even Natural Language Processing (NLP) models (Sun et al., 2019). Despite the lack of a universally accepted definition of offensive language, it is commonly associated with cursing, profanity, blasphemy, epithets, obscenity, and insults (Jay, 1992). Thus, automatically detecting discriminatory language can assist in analyzing it so that preventative measures can be taken. In terms of gender representation in language, most feminist language activists support the change of language as a means of achieving better gender equality (Pauwels, 2003). The use of these systems can be useful in the development, design, and dissemination of policies related to equality, as well as in contributing to social change in a positive direction.

Identifying subtle forms of sexist and offensive language can be quite challenging. At present, the majority of research focuses on each of these tasks individually. Despite their compelling results, these approaches are limited to modeling only the linguistic aspects of discriminatory and offensive language, without taking into account an important aspect, such as emotions. Emotions are highly prevalent in language and thought. Using information obtained from people's emotional states when expressing themselves could support and improve the development of natural language applications. For more complex semantic tasks, such as detecting sexist and offensive language, a unified system may be necessary. The majority of systems, however, do not possess certain features that may help facilitate the development of such a system, as discussed above. The concept of Multi-Task Learning (MTL) is based on human learning activities in which individuals apply their knowledge from auxiliary tasks to assist them in the learning of a new task. It is an approach of inductive transfer in which the domain information contained in the training signals of related tasks is used as an inductive bias to facilitate generalization (Caruana, 1997). At its infancy, MTL is motivated primarily by the goal of alleviating the problem of data sparsity, and by aggregating the labeled data in all tasks, MTL achieves more accurate learning for each task, being therefore useful in reusing existing knowledge and reducing the cost of manual labeling. Lastly, deep MTL models perform better than single-task models (Zhang and Yang, 2022). Our study hypothesizes that adding emotion information to the mix can help in detecting sexism and offensive language in Romanian tweets in a more efficient and effective manner.

Furthermore, most of the sexist and offensive language detection systems are developed for well-resourced languages. Therefore, a key objective of this paper is the development of language technologies in the midst of a scarcity of digital language resources and tools for Romanian language. Romanian is a Romance language spoken by approximately 24 to 26 million people as a native language, while about 4 million speak it as a secondary language.[1]

To the best of our knowledge, this is the first study to employ emotion analysis in the detection of sexist and offensive language in a less-resourced language, such as Romanian.

2. Related Works

Over the past few years, there have been numerous academic events and shared tasks related to the identification of sexist and offensive language. For low-resource languages, however, the detection of offensive language has received relatively little attention in NLP.

The EXIST (Sexism Identification in Social Networks) competition at IberLEF (Rodríguez-Sánchez et al., 2022) was the first collaborative effort aimed at detecting sexism in a broad sense, from outright misogyny to more subtle expressions of sexism. Within the same shared task, del Arco et al. 2021 test the performance of a multi-task learning approach that incorporates sentiment analysis and offensive language detection to identify sexism. In another recent study, del Arco et al. (2022) investigated more linguistic phenomena than sentiment analysis in their research for sexism detection. Using multi-task methods they incorporate emotions, sarcasm, insults, constructiveness, and targets into the learning process.

[1]https://www.britannica.com/topic/Romanian-language

Sharifirad et al. (2019) examine the users' mood when writing sexist tweets. They use the SemEval-2018 task1: Affect in tweets dataset (Mohammad et al., 2018), and examine the types and intensities of emotions associated with categories of sexual harassment. According to their findings, indirect harassment, also known as benevolent harassment, has a mild intensity. On the other hand, hostile sexism is associated with very high levels of disgust, anger, sadness, and even joy. The tweets also demonstrate that users enjoy sending sexist messages to women.

In modelling the linguistic properties of abusive language, Rajamanickam et al. (2020) consider the emotional state of the users and how this may affect their language. Using a multi-task learning framework, they present a joint model of emotion detection and abusive language detection. According to their results, incorporating affective features increases abuse detection performance across datasets significantly.

Using three auxiliary tasks that were automatically created through unsupervised learning from a set of unlabeled and weakly labelled accounts, Abburi et al. (2020) explored neural multitask learning and investigated 23-class fine-grained classifications of accounts of sexism.

3. Corpora

For elaborating the multi-task architecture we used two datasets, CoRoSeOf (Hoefels et al., 2022), and REDv2 (Ciobotaru et al., 2022).

CoRoSeOf is a large corpus of Romanian social media manually annotated for sexist and offensive language. There are covert and overt forms of sexist language included in the corpus, which have been classified into direct, descriptive, and reported statements. It consists of 39245 tweets which have been annotated with the following labels: *sexist direct*, *sexist descriptive*, sexist reporting, *non sexist offensive* and *non sexist*. It is important to note that approximately 80% of this corpus is skewed towards the non sexist class.

To compliment the sexist and offensive language classification by using a multi-task approach, we construct the auxiliary task using the RED dataset. REDv2 is a Romanian emotion detection dataset containing 5449 tweets annotated in a multi-label fashion, with the following emotions: *anger*, *fear*, *joy*, *sadness*, *surprise*, *trust* and *neutral*.

4. System overview

As a preliminary step towards predicting the emotions in each CoRoSeOf text, we developed an emotion detection model by fine-tuning the cased version of the Romanian BERT (Dumitrescu et al., 2020) from Huggingface[2] on the task of classifying emotion labels of REDv2 tweets.[3]

We have created a model that loaded the weights from the bert-base-cased with a linear layer on top, using Transformer's class AutoModelForSequenceClassification, with its *from_pretrained* method, and training was conducted using HuggingFace's Trainer API. The model was trained for five epochs with a batch size of eight and a learning rate of 2e-5. The loss function used was BCEWithLogitsLoss, which combines the sigmoid layer with the Binary Cross Entropy loss in a single class. As a result of this model, we were able to obtain an F1-score of 0.71 on REDv2.

Figure 1 illustrates the distribution of emotion labels in the five CoRoSeOf classes using UpSet plots (Lex et al., 2014). These plots are used to visualize intersections between more than three sets, in a matrix. The horizontal bars in the left part of the matrix represent the total number of texts which contain the specified emotion in each row. Using the vertical bar above the matrix, each unconnected dot on the matrix shows the count of texts containing one unique emotion, while the connected dots indicate the number of texts sharing more than one emotion.

In applying an upset plot to CoRoSeOF texts after emotion prediction, the following insights can be derived: *non sexist* tweets are mainly neutral while in *non sexist offensive* tweets, anger is the predominant emotion expressed, followed by neutral, sadness, and a mixture of sadness and anger. The majority of *sexist direct* tweets express joy, anger, and neutral feelings, while a few express trust, and the combination of trust and joy. *Sexist descriptive* tweets are mainly neutral, followed by a relatively high number that express anger, and a few express trust, sadness, joy, as well as combinations of trust and neutral, sadness and anger, and trust and joy. There is a predominance of anger in *sexist reporting* texts, with neutral representing the second most prevalent emotion.

The first column in each upset plot represents the amount of texts in the specified category which have not received an emotion label by the emotion detection model, because the probability of detection for each of the seven emotion labels was lower than 50%.

Figure 2 outlines the three types of model architectures that were considered. In order to develop a baseline, we start with the simplest architectural model; then, we add extra data to the model in the form of precomputed emotion probabilities; and we compare the results of that model with those of a multi-task model trained on the CoRoSeOf and REDv2 datasets jointly.

Finally, training was standardized, i.e., all models use a 0.1 dropout, the same learning rate and learning schedule, and the same early stopping criterion and patience (including the multi-task model where we early stopping considers only the validation set of CoRoSeOf.

4.1. Data Preprocessing

In order to ensure accuracy and minimal bias in the data, we preprocessed it using the following steps: first, we removed all usernames from the CoRoSeOf dataset. The nature of Twitter responses prompted us to take this action; since they are branched off from the original tweet in a tree-like fashion (Ryosuke Nishi (2016)), it was paramount to avoid our models to be biased towards certain users. Secondly, we deleted from CoRoSeOf 49 texts that had a majority vote ground truth of 'Cannot decide' and 1457 texts with 'Non agreement'. Thirdly, we replaced names with "person" using `roner` python library (Dumitrescu and Avram (2019)), emails with "email" using regex, and also we eliminated telephone numbers, as these do not bring valuable information in the machine learning process and it was important to align the CoRoSeOf preprocessing with the REDv2 preprocessing. Lastly, we split the data in an 80/10/10 fashion, making sure each CoRoSeOF label has an equal distribution for training, testing and validation.

4.2. Baseline Model

The baseline model is a BERT transformer from which the pooled output is forwarded to a dense layer representing the output classes. In the training process, there is a standard dropout of 0.1 before the last layer, in which the cross-entropy loss is computed. This is the most basic model that can be used with transformers; although simple, it provides a strong baseline.

[2] https://huggingface.co/dumitrescustefan/bert-base-romanian-cased-v1

[3] https://huggingface.co/datasets/Alegzandra/REDv2

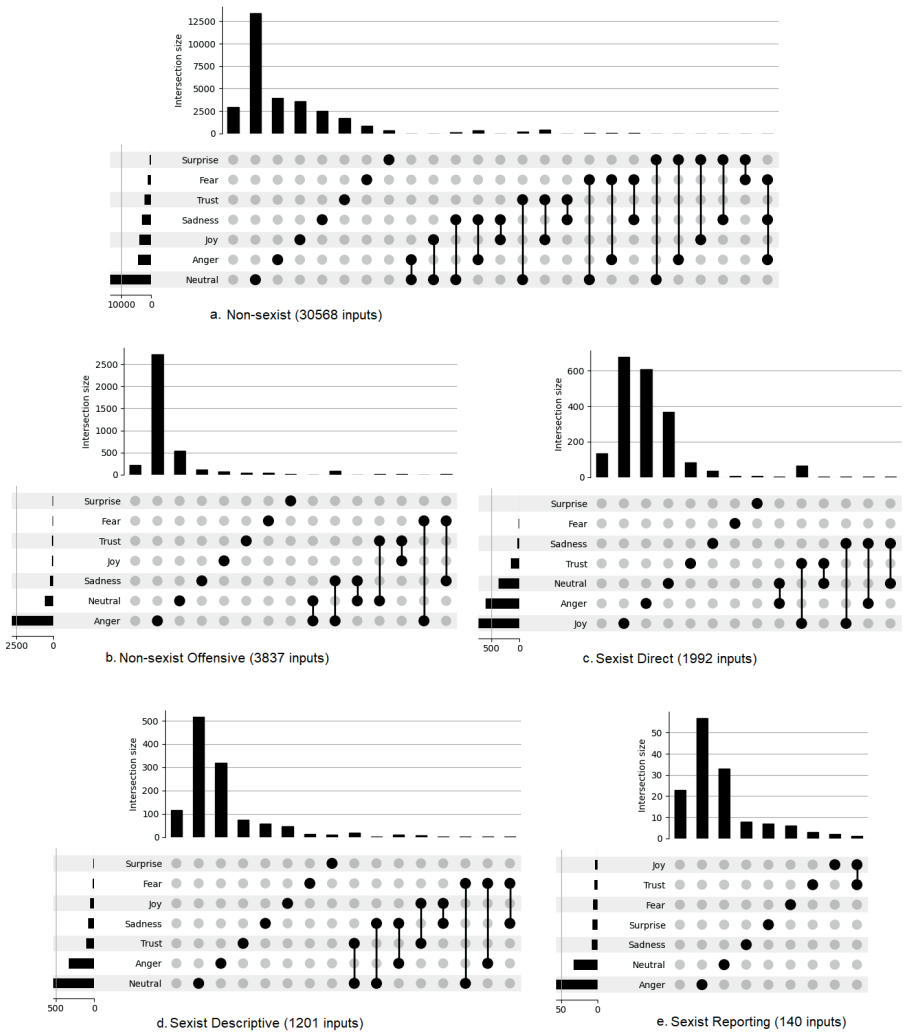


Figure 1: Multi-Label Emotion Distribution in CoRoSeOf Classes: a. non sexist, b. non sexist Offensive, c. Sexist Direct, d. Sexist Descriptive, e. Sexist Reporting.

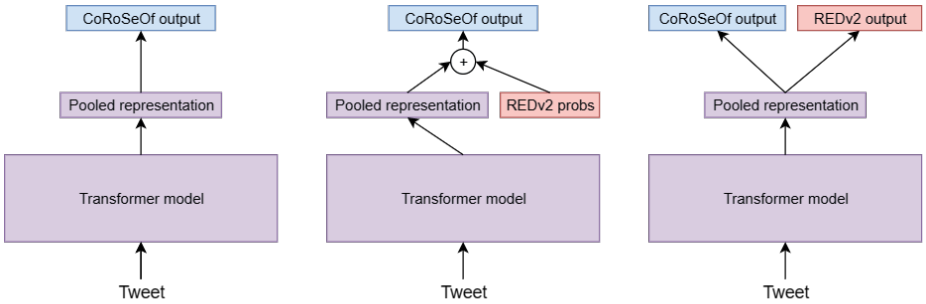


Figure 2: The three architectures tested: baseline (A), feature engineered model where we add the pre-computed REDv2 emotion probabilities for each tweet (B), and the multi-task model with distinct heads for each dataset (C)

4.3. Feature-engineered Model

The feature-engineered model benefits from the results of the existing emotion detection model trained on REDv2, by making use of the emotion prediction for each tweet, in addition to the tweet's text itself.

Architecturally, the model is identical to the baseline, but after pooling all the outputs from the transformer into a single-dimensional 768 vector (as with bert-base), we concatenate seven more values to it, representing the floating point probability predictions of the seven emotions of REDv2. To summarize, we add these seven pre-computed values to the input of the class-output layer.

4.4. Multi-task Model

Our multi-task model architecture follows the standard paradigm where we have a single model that encodes tweets, but different output heads for different tasks. Thus, we have two distinct tasks/heads: (i) CoRoSeOf head: a 5-valued output trained with cross–entropy loss; (ii) REDv2 head: a 7-valued

Model	Transformer Model	F1	Acc
Baseline	romanian-bert-cased	0.8784	0.88
Feature Eng.	romanian-bert-cased	0.8778	0.8835
Multi-task	romanian-bert-cased	**0.8798**	**0.8844**
Baseline	xlm-roberta-large	0.7983	0.8348
Feature Eng.	xlm-roberta-large	0.8117	0.8489
Multi-task	xlm-roberta-large	**0.8399**	**0.8614**

Table 1: Training results on the presented model architectures.

Model	Transformer Model	F1	Acc
Multi-task 0.5	romanian-bert-cased	0.8789	0.8834
Multi-task 0.5	xlm-roberta-large	0.81	0.8465

Table 2: Training results on the multi-task model architecture, with 50% ratio between datasets.

output trained with binary cross–entropy loss. The input for both heads is, as in the case of the previous architectures, the pooled layer from the transformer model.

As a result of the different tweet counts contained in REDv2 and CoRoSeOf datasets, we had to batch them separately during the training process. Thus, at each step, we randomly pick one of the two datasets and create a batch with tweets from one of them, forwarding them up to the corresponding head. The datasets differ substantially in terms of size (CoRoSeOf contains 37.738 texts, whereas REDv2 contains 5.449), therefore we determine a probability threshold of 13.54% of the model choosing a batch from either dataset. In this manner, the model is able to see that the datasets are equally represented according to their size, and does not overfit on the smaller dataset. To verify this, we attempted to examine what the results would be if we forced the ratio to be 50/50. Our findings are illustrated in Table 2.

5. Experiments and Results

Our experiments were performed using the architectures described in Section 4, for all tested models, with a batch size of 16 (including for larger models). We tested more Romanian bert-base models, but for brevity only report the best performing one; we also test the xlm-roberta-large as a strong multilingual contender. Results are reported in Table 1, averaged across 5 runs.

To our expectations, the multi-task model achieves the highest F1-score and accuracy on both romanian-bert-cased and xlm-roberta-large. Between these two transformer models, romanian-bert outperforms the multilingual xlm-roberta by 4% F1-score and 2% accuracy, even if the roberta model is almost 3x larger (355M vs 124M), showing the power of monolingual models.

For variability, we have also trained the multi-task model using a ratio of 50% between both datasets. This means that at training time, the same amount of batches are taken from both CoRoSeOf and REDv2 datasets. It can be seen in Table 2 that using this ratio, both F1-score and accuracy, on both transformer models, are lower than the results in Table 1 when using the default ratio, which considers the disproportion between datasets when assigning training batches.

To compute a confidence interval of the results, we have trained the best performing model, the emotion and sexist and offensive language multi-task model which uses romanian-bert-cased, for a number of 20 times with random seeds. The averaged results are: F1-score 0.8791 and accuracy of 0.8827. The standard deviation for the F1-score is 0.0043, and for the accuracy is 0.0056. It can be observed that the standard deviation on both measures is small enough to consider these results reliable.

Class	Correct	Error	Support	Acc.
Non sexist	2916	141	3057	0.9539
Sexist direct	140	59	199	0.7035
Non sexist off.	210	173	383	0.5483
Sexist descriptive	64	56	120	0.5333
Sexist reporting	0	14	14	0

Table 3: Error analysis for each CoRoSeOf class.

5.1. Error Analysis

A detailed understanding of how failures occur or are distributed within the proposed models is essential. Thus, in Figure 3, we show the confusion matrix for one of our five experiments when creating the averaged multi-task model. This model has an overall accuracy of 0.8826. As can be seen in Figure 1, most of the test predictions belong to the neutral class, which is expected, since this is the class in CoRoSeOf that contains the most training texts.

Also, the *sexist reporting* class was never predicted in this experiment, which corroborates with the fact that this class contains only 140 texts and is also the smallest class in the CoRoSeOf dataset.

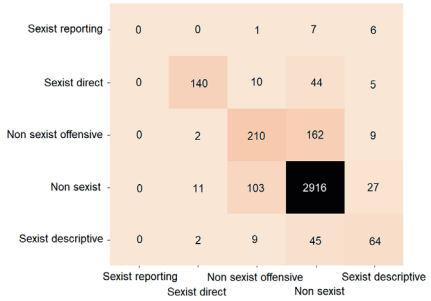


Figure 3: Confusion matrix for the romanian-bert-cased multi-task model

A detailed analysis of the errors is given in Table 3. The highest accuracy of 95.39% is achieved by the *non sexist* class, followed by *sexist direct* with 70.35%. The accuracy values of the *sexist descriptive* and *non sexist offensive* classes are rather similar, with 53.33% and 54.83%, respectively. In the case of *sexist reporting* class, the accuracy of the class is 0.

Analysing both Figure 3 and Table 3 the following phenomena stand out. The classification of *sexist reporting* tweets has never been accurate. The model considered seven of the texts to be *non sexist*, one to be *non sexist offensive*, and six to be *sexist descriptive*. In the example "era un moș beat mort în autobuz și se ținea în continuu după mine" ('there was a dead drunk old man on the bus and he kept following me'), the tweet is labeled as *sexist reporting*, however, the model classified it as *non sexist*. In addition, the emotion detection model indicates that this text reflects fear, which is an accurate prediction.

Sexist direct tweets were mainly classified correctly, with only 10 texts classified as *non sexist offensive*, 44 texts *non sexist*, and 5 texts *sexist descriptive*. The tweet, "bună, ce faci frumoaso" ('hello, how are you beautiful') was predicted as *non sexist* instead of *sexist direct*. In a surprising finding, the emotion detection model identified this text as joyful, showing that predicting the covert forms of sexism is challenging.

Two *non sexist offensive* tweets were miss-classified as *sexist direct*, 162 as *non sexist* and 9 as *sexist descriptive*. The text, "Judecătorii s-au șmecherit mult. Motivările cred că sunt lăsate in seama femeilor de serviciu..." ('The judges have become very sly. I think the motivations are left up to the maids...') was ini-

tially labelled as *non sexist offensive*, however, the model classified this text as *sexist descriptive*, which might be a true label in the opinion of some annotators.

The majority of *non sexist* tweets were classified correctly, although some exceptions were observed, including 11 tweets misclassified as *sexist direct*, 103 tweets misclassified as *non sexist offensive*, and 27 texts misclassified as *sexist descriptive*. The tweet, "Foarte frumoasă și sexy" ('Very beautiful and sexy') was classified as *sexist direct* by the model instead of the gold standard, *non sexist*. It is, however, apparent that the model generalizes correctly in this very subjective case. Furthermore, according to the emotion detection model, the emotion carried by this text is joy.

Finally, we noticed that the *sexist descriptive* tweets were mainly missclassified as *non sexist* (45 out of a total of 120), 2 *sexist direct*, and 9 *non sexist offensive*. The text "Trăiesc bărbat cu nevastă, darămite câine cu pisică" ('Man and wife live together, let alone dog and cat') is *sexist descriptive* but was misclassified as *non sexist*. The above is a classic example of how machine learning models are having difficulties to understand sarcasm, combined with the fact that our multi-task model has a preference for the *non sexist* class.

6. Conclusions and Future Work

We examine how emotions can be used to aid in categorizing sexist and offensive language; our experiments show that our proposed multi-task approach, addressed for the first time in the Romanian language for improving the detection of sexism and offensive language using emotions, is capable of producing convincing results. The dataset, code and results are freely available on GitHub.[4]

Further research will be carried out to test the effectiveness of the model in identifying sexist and offensive language using sarcasm and irony detection within texts as additional tasks. As a result of the imbalanced nature of CoRoSeOf, sampling data would be another approach to experiment with.

References

Abburi, Harika, Pulkit Parikh, Ni Chhaya, and Vasudeva Varma, 2020. Semi-supervised multi-task learning for multi-label fine-grained sexism classification.

Caruana, Rich, 1997. Multitask learning. *Machine Learning*, 28.

Ciobotaru, Alexandra, Mihai V. Constantinescu, Liviu P. Dinu, and Stefan Daniel Dumitrescu, 2022. RED v2: Enhancing RED Dataset for Multi-Label Emotion Detection. Marseille, France: European Language Resources Association (ELRA).

del Arco, Flor Miriam Plaza, M. Dolores Molina-González, Luis Alfonso Ureña López, and María Teresa Martín-Valdivia, 2021. A multi-task learning approach to hate speech detection leveraging sentiment analysis. *IEEE Access*, 9:112478–112489.

del Arco, Flor Miriam Plaza, María Dolores Molina-González, Luis Alfonso Ureña López, and María Teresa Martín Valdivia, 2022. Exploring the use of different linguistic phenomena for sexism identification in social networks. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2022) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2022), A Coruña, Spain, September 20, 2022*, volume 3202 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[4] https://github.com/DianaHoefels/LTC23-MultiTaskLearning

Dumitrescu, Stefan, Andrei-Marius Avram, and Sampo Pyysalo, 2020. The birth of Romanian BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics.

Dumitrescu, Stefan Daniel and Andrei-Marius Avram, 2019. Introducing ronec–the romanian named entity corpus. *arXiv preprint arXiv:1909.01247*.

Hoefels, Diana Constantina, Çağrı Çöltekin, and Irina Diana Mădroane, 2022. CoRoSeOf - an annotated corpus of Romanian sexist and offensive tweets. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association.

Jay, T., 1992. *Cursing in America: A Psycholinguistic Study of Dirty Language in the Courts, in the Movies, in the Schoolyards, and on the Streets*. Cursing in America: A Psycholinguistic Study of Dirty Language in the Courts, in the Movies, in the Schoolyards and on the Streets. J. Benjamins Publishing Company.

Lex, Alexander, Nils Gehlenborg, Hendrik Strobelt, Romain Vuillemot, and Hanspeter Pfister, 2014. Upset: Visualization of intersecting sets. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1983–1992.

Mohammad, Saif, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko, 2018. SemEval-2018 task 1: Affect in tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation*. New Orleans, Louisiana: Association for Computational Linguistics.

Pauwels, Anne, 2003. *Linguistic Sexism and Feminist Linguistic Activism*, chapter 24. John Wiley Sons, Ltd, pages 550–570.

Rajamanickam, Santhosh, Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova, 2020. Joint modelling of emotion and abusive language detection.

Rodríguez-Sánchez, Francisco, Jorge Carrillo de Albornoz, Laura Plaza, Adrián Mendieta-Aragón, Guillermo Marco-Remón, Maryna Makeienko, María Plaza, Julio Gonzalo, Damiano Spina, and Paolo Rosso, 2022. Overview of exist 2022: sexism identification in social networks. *Procesamiento del Lenguaje Natural*, 69(0):229–240.

Ryosuke Nishi, Taro Takaguchi Keigo Oka Takanori Maehara Masashi Toyoda Ken-ichi Kawarabayashi Naoki Masuda, 2016. Reply trees in twitter: data analysis and branching process models. *Social Network Analysis and Mining*:1869–5469.

Sharifirad, Sima, Borna Jafarpour, and Stan Matwin, 2019. How is your mood when writing sexist tweets? detecting the emotion type and intensity of emotion using natural language processing techniques.

Sun, Tony, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang, 2019. Mitigating gender bias in natural language processing: Literature review.

Verniers, Catherine and Jorge Vala, 2018. Justifying gender discrimination in the workplace: The mediating role of motherhood myths. *PLOS ONE*, 13:e0190657.

Zhang, Yu and Qiang Yang, 2022. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609.

Linguistic Information Extraction from Text-based Web to Discover Criminal Activity

Grażyna Demenko, Paweł Skórzewski, Tomasz Kuczmarski, Mikołaj Pieniowski

Adam Mickiewicz University, Poznań
{lin,pawel.skorzewski,tkucz}@amu.edu.pl, mikpie1@st.amu.edu.pl

Abstract

This paper describes an approach for detecting organized crime activities based on Web-harvested linguistic evidence. We describe a fully functional prototype software system that pursues the following objectives: a) defining the subject of crime with the possibility of recognizing different grammatical functions, ambiguities, situational context, and detecting suspicious content, b) creating linguistic identifiers of objects of interest of border guards, and c) clustering such objects. To evaluate the performance and effectiveness of the proposed system, we obtained actual unstructured documents about investigations from users and confirmed the efficiency and effectiveness of the data analytics process in terms of context-aware analytics systems and knowledge management.

Keywords: criminal texts, information extraction, NER, NLP

1. Introduction

The problem of combating organized crime related to cross-border criminal activities (e.g., illegal smuggling of various goods, slavery, prostitution, trafficking in human organs) needs an urgent solution. The crime statistics show how effectively criminals exchange information via the Internet through its "gray area" and on public blogs and websites (Krauz, 2017; Mider, 2019). Even the slightest traces left by these groups enable using new information technologies to work out the "web of connections", particularly relations and key nodes (Andrews et al., 2018; Rowe et al., 2007). Border Guard analysts search the Internet manually to find such information, which is time-consuming. Automating the process of browsing and analyzing Internet resources would enable both acceleration of the process and significant broadening of the scope of searches. The analysis of informal, noisy, and short texts in terms of crime-related content requires recognizing not only standard named entity categories like the names of persons, organizations, and locations, but also domain expression categories, like product names (e.g., drug names, cigarette and alcohol brands), descriptions, or actions, which are equally relevant to crime intelligence analysis. Our research offers a method of processing unstructured big data: actual Polish crime texts collected from the Web, and a practical solution for crime information extraction for the needs of the Border Guard.

2. Towards Criminal Event Ontology

Online criminal events involve various actors, places, times, causes, and relations and use a variety of natural language structures. The extracted events are valuable for information retrieval tasks as a knowledge base. In this paper, we design, implement, and evaluate a mechanism for defining and mining relevant information, especially on Polish web pages. Literature knows multiple examples of applications of criminal events ontologies, e.g., mining data to catch professional criminals (Brown, 1998), or discovering crime patterns using records composed of many attributes (Nath, 2006).

2.1. Domain Entity Categorization

The semantic analysis result for a single text is a list of named entities and their types found in it. We distinguish nine categories of named entities relevant to the task: identifier, object–person, object–item, action, organization, location, time, measure, and description. **Identifier** is a named entity that enables direct or indirect identification of the author of the text, or facilitates the localization of the event, e.g., name, surname, nickname, phone number, e-mail address, URL, website name. **Object–person** is a concept relating to a person, e.g. profession, function, nationality. **Object–item** is a potential crime target (excluding people). This category includes anything that may be the subject of trafficking or other crimes, except people: physical or virtual goods, works, and products, e.g. commodities, works of art, food, drugs. **Action** is a category related to events. Actions can be related to crime directly (smuggling, trafficking) or indirectly (traveling, accessing information, using websites, shipping goods). Verbs in this category denote dynamic situations, i.e., situations that involve a change in the state of the performer of this activity, the object to which the activity relates, or the relationship between the participants of the action. Actions can be signaled directly with verbs (e.g., *walk*, *move*, *condemn*, *work*) or indirectly with other parts of speech (e.g., *sale and purchase*, *distribution*). **Organization** is a category for organization names, e.g. *Border Guard*, *Polish Post*. **Locations** are geographic places, addresses, institution names, e.g.: *Warsaw*, *Russia*, *ul. Słowackiego 8*, *escort agency*. **Time** denotes different types of temporal expressions related to date, time or duration, e.g. *today*, *4 a.m.* **Measure** category includes physical measures, colloquial terms of size, and currency terms, e.g. *123 €*, *0.5 mg*. **Descrip-**

This work was financed by the Polish National Centre for Research and Development (dec. no. DOB-BIO9/19/01/2018)

tions are phrases that describe, explain, or comment expressions referring to potential criminal activity, e.g., *blue*, *from abroad*, *without excise duty*.

For the semantic analysis of Web-harvested texts, we used the lexicon for the Polish language used in the LVCSR system (Demenko, 2015) and a lexicon containing words and phrases related to crime. The corpus consists of 17,764,634 units (tokens) and ensures high coverage in many lexical domains: people, organizations, geographic and geopolitical names, products, events.

2.2. Domain Peculiarities

The domain crime vocabulary is full of domain jargon and numerous linguistic ambiguities. The following problems are crucial for relevant domain information extraction (Adnan and Akbar, 2019; Ku et al., 2008). Unstructured data have no schema; they have multiple formats and come from different sources. Web texts are very noisy, so sometimes morphological or syntactical analysis is not possible. Grammatical errors occur on all levels of linguistic analysis. The texts also contain syntactic and semantic ambiguities.

3. The Context Toolset

The Context Toolset is a part of the AISearcher system, aimed to support searching the Internet for criminal texts that may be of interest to employees of the Polish State Border Guard (Nowakowski and Jassem, 2021). The toolset processes the queries with four levels of linguistic analysis: semantic, syntactic, lexical, and structural/specific. Additionally, the toolset creates the Linguistic Identifier of the Object Card – a unique identifier based on the linguistic properties of texts. This identifier is also used to cluster the objects.

3.1. Semantic Analysis

The semantic analysis for the Context module is performed by the named entity recognition tool (Skórzewski et al., 2022). This tool is based on hand-crafted rules, lexicon lookup, and regular expressions. After testing different solutions, we decided not to use neural networks or statistics-based algorithms for two main reasons. First, it turned out that the annotated dataset we have is not large enough to train a well-performing neural network. Second, although the statistically enhanced classifier returned better results (as measured by F_1-score), the target users (i.e., the Border Guard officers) preferred the rule-based approach because the results were more predictable and explainable.

3.2. Syntactic Analysis

Syntactic analysis is meant to extract the underlying structure of the documents' sentences. In addition, it provides tools for preprocessing text documents before they can undergo other levels of linguistic analysis. First, the text documents are split into tokens, using a regular-expression-based algorithm from the Python NLTK library (Loper and Bird, 2002; Bird, 2006). Next, the resulting tokens go through a morphosyntactic tagging procedure which outputs a set of labels containing rich information about their grammatical categories, such as the general part of speech, the grammatical number, the person or gender of a noun, or the aspect of a verb. In Polish, one string of characters can often be categorized into several different grammatical categories depending on the context, i.e., several homonyms often exist for that same spelling. An additional tool for morphosyntactic disambiguation of tokens has also been added to account for such situations. The morphosyntactic parser and the disambiguator are based on third-party tools; Morfeusz (Kieraś and Woliński, 2017) and Concraft (Waszczuk et al., 2018), respectively. Morfeusz is additionally employed in the current system as a lemmatizer/stemmer, which brings the tokens into their basic grammatical forms as they are input into other levels of analysis.

Given the morphosyntactic categories of disambiguated tokens, Context also performs dependency parsing with the publicly available Malt Parser (Nivre et al., 2006) and a model pre-trained on the current version of the Polish Dependency Bank (Wróblewska, 2014). The graph output by the parser contains the tokens as the nodes, with the morphosyntactic categories as their additional features and different dependency relations as the edges. Dependency graph is transformed into a vector representation (Hamilton et al., 2017) with Graph2Vec algorithm (Narayanan et al., 2017; Rozemberczki et al., 2020). It is based on the Weisfeiler-Lehman hashing method (Shervashidze et al., 2011), which iteratively aggregates and hashes the neighborhood of each of the graph's nodes which can then be input into the standard Doc2Vec (Le and Mikolov, 2014) model. Graph2Vec encodes graphs with node features but does not support edge features, and hence this information is currently lost in the vectorization process. We have accepted this as a necessary trade-off between the representativeness of the vector and the efforts required to implement a new algorithm. The algorithm has been trained on a database of 7188 text documents collected and hand-labeled partially by the Polish Border Guards and partially by specially trained linguists. The vector representation output by this current method provides important features for any successive classification algorithms. It is one of the currently most effective knowledge representation methods that capture deep relations and information in the document (e.g., the subject, object, and predicate of a sentence).

At the last stage of analysis, we also add the counts of all possible punctuation marks found in the document as additional features. Despite their relative simplicity, these might provide valuable information about the nature of the analyzed documents, their authors, and their writing style.

3.3. Lexical Analysis

For lexical analysis, the documents are first transformed into a string of lemmatized tokens with the tools implemented as part of the syntactic toolchain. Next, the stop words (the most frequent non-characteristic words in a language) are filtered out. Then, the list of keywords and the number of their occurrences is extracted based on the specially prepared dictionaries of domain-specific vocabulary

(such as names of illegal drugs). The lexical analysis also includes the standard Text Frequency – Inverse Document Frequency (TF–IDF) analysis (Sammut and Webb, 2010), which calculates how often each word appears in any given text compared to its general frequency. This helps identify words and their combinations (bigrams) which are especially rare in other texts. Finally, the lexical analysis also includes the identification of the most central sentence in the document. The sentence is identified based on the LexRank (Erkan and Radev, 2004) algorithm – a modified version of the PageRank algorithm developed by Google (Alphabet) for ranking web pages based on the counts of their relations.

3.4. Specific/structural Analysis

The specific (or structural) level of analysis calculates several basic stylometric indexes and text statistics. These include a range of statistics, starting with simple measures such as the total and the average number of characters, syllables, words, and sentences, to some more specific metrics such as the number of words, the Coleman-Liau index, the Linsear Write metric, the LIX and RIX readability formulas, or n-syllable word count.

3.5. Linguistic Identifier of the Object Card

The system gathers information about the so-called *objects*. In this context, an object can be anything of interest to the Border Guard: a person, a subject of the crime, etc. One of the tasks of the system is to gather texts and enable linking them to different objects. Objects in the system are represented as *object cards*. A Linguistic Identifier of the Object Card (LIOC) is a numeric feature vector generated automatically based on the results of the linguistic analysis of texts linked to the particular object.

LIOC is implemented as a numeric vector consisting of five parts: four of them are numeric vector representations of the four layers of linguistic analysis described above. The fifth segment is a vector obtained using machine-learning methods of document representation (paragraph vector, or Doc2Vec). The size of the vector and the sizes of the vector's segments are constant for every input object. Different segments have different sizes, resulting from the specificity of individual layers of linguistic analysis.

LIOC vector is generated for the whole object, i.e., for the set of texts linked to the object. If this set of texts contains only one text, the procedure of generating LIOC is the following.

To obtain a LIOC vector for a single text, the results from different layers of analysis are converted to a flat vector of numbers. The results of semantic analysis, i.e., named entity recognition, are converted to a vector using the Doc2Vec algorithm. This vector is constructed from the set of all tokens recognized by a NER tool as a valid named entity, regardless of its category. The results of syntactic analysis consist of a dependency graph and punctuation statistics. The dependency graph is converted to a vector using the Graph2Vec algorithm, and the punctuation statistics already have a form of a numerical vector of a constant length. The result of lexical analysis consists of three parts. First, there is a vector with counts of words from the lexicon of crime-related keywords. Second, there are the results of TF–IDF analysis (also a vector). Finally, there is the index of the central sentence and its score. The specific/structural analysis results are numerical values of stylometric indexes and text statistics. The fifth part of the LIOC vector is a Doc2Vec representation of the whole input text. LIOC vector for a single text is a concatenation of the five segments described above.

If there are more texts assigned to the object, LIOC is calculated as the arithmetic average of vectors calculated for each text separately, in a way described above.

3.6. Object Clustering

One of the main tasks of the Context toolset is to perform clustering of groups of text documents based on the LIOC. The goal of the clustering is to group together objects with similar linguistic characteristics. We use the k-means clustering algorithm and LIOC vectors as representations of objects. The number of clusters k is given by the user. The distance metric is a standard Euclidean distance between LIOC treated as points in space.

4. Evaluation and Discussion

4.1. Semantic Analysis Evaluation

An evaluation corpus was prepared to evaluate the named entity recognition tool used for semantic analysis. The texts were collected from various sources: Polish classifieds websites and marketplaces accessible through widely available search engines, and Polish and international marketplaces on the Darknet. The Clearnet sources allowed unrestricted access to their contents, enabling them to be localized conveniently through the aforementioned search engines. Once localized, relevant linguistic material was collected automatically using dedicated web scrapers adjusted to each website individually. However, this was not the case with the Darknet sources, given their covert nature, which rendered the automatic approach to data extraction inefficient as opposed to the manual approach, which we eventually employed. The Darknet websites containing relevant linguistic data were firmly secured with advanced challenge-response tests. Moreover, the fact that TOR (The Onion Router) websites are not being listed in traditional, widely available search engines posed an additional challenge in the localization of the sources. The texts localized on Darknet were therefore appended manually to the dataset.

Eventually, we obtained 3337 full texts originating from 5 Darknet and 3 Clearnet websites. The collected texts were annotated with named entities belonging to the nine categories listed in Subsection 2.1.. Nine linguists were involved in the process of annotation. Raw texts were evenly distributed among individual annotators, who were provided with comprehensive guidelines and were given a dedicated workspace. The annotators remained in constant contact with the coordinator, who tracked and reported the work progress throughout the entire undertaking. The whole annotated dataset consists of 6240 anno-

	lex.-based algorithm			manual annotation		
Category	prec.	F_1	rec.	prec.	F_1	rec.
Identifier	41%	53%	75%	39%	28%	22%
Object	76%	57%	45%	73%	70%	67%
Action	54%	29%	20%	48%	46%	44%
Location	11%	10%	9%	43%	12%	7%
Time	18%	18%	18%	46%	39%	34%
Measure	76%	59%	48%	92%	85%	79%
Descript.	20%	22%	25%	10%	14%	23%

Table 1: Multi-label precision, F_1-score, and recall for lexicon-based algorithm and manual annotation.

category	# documents	category	# doc.
neutral	3251	documents	300
drugs	2327	amber	121
tobacco	1177	human trafficking	12

Table 2: Structure and counts of the data set employed for the evaluation of text document clustering.

tated text fragments. We will refer to this set as A_{all}. Additionally, its subset of 554 text fragments was annotated by two independent annotators. We will refer to this subset as A_6.

For the evaluation, we used precision, recall, and F_1-score. Our approach obtained the F_1-score of 0.487 on the A_6 subset (compared to 0.621 for manual annotation) and 0.219 on the A_{all} set. For comparison, we evaluated Nerf – a general-purpose named entity recognition tool – in the same way. Nerf scored only 0.005 on the A_6 set and 0.017 on the A_{all} set. Precision, recall and F_1-scores for manual annotation were obtained by calculating these metrics on the A_6 subset for one of the annotators while treating the other one as a ground truth. Comparing the results achieved by our method with manual annotation results by category (Table 4.1.) shows that our algorithm scored comparably to humans for categories like identifier or description. On the other hand, the time and location categories proved more challenging. The results achieved by Nerf show that general-purpose named entity recognition tools are not suitable for specialized entity recognition tasks. Scores obtained from manual annotation evaluation are far from 100%, which means a substantial disagreement between annotators. Some categories like description or location are not precisely defined, which influences the ambiguity of the task. Another valuable observation is that the analytic measures of NER performance like the F_1-score do not always coincide with the users' perception. Although some of the statistical and neural models we tested achieved better F_1-scores than the lexicon-based algorithms, the system's users preferred the latter approach because they perceived it as more "coherent" and "predictable" than the former.

4.2. Object Clustering Evaluation

The effectiveness of the clustering method was evaluated along with the relative effectiveness of the features extracted using different levels of linguistic analysis. The evaluation was performed on a database of 7188 documents collected and labeled partially by the Polish Border Guards and partially by specially trained linguists. Each document in the database was manually tagged with one of the criminal categories or a non-criminal category (neutral). The structure of the whole database and the document count of individual categories are presented in Table 4.2.. Five samples of 20 documents were randomly selected from the evaluation dataset, and an ablation study was performed using different configurations of linguistic features used for clustering. Individual levels of analysis and the resulting feature sets were systematically included and excluded from the LIOC feature vector, and clustering was performed on this basis. For each of the different clustering results, two standard metrics were calculated, i.e., the Adjusted Rand Score (ARS) and the Adjusted Mutual Information Score (AMIS) (Emmons et al., 2016). These metrics compute different similarity measures between two clusterings by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusterings. These metrics are both symmetric and insensitive to permutation. The closer the metric is to 1.0, the better the clustering, and the closer it is to 0.0, the worse the results. In case of large discrepancies, the metrics can even report negative values. The evaluation showed that the best results were reported for clustering based solely on distributive semantic features, i.e., on the Doc2Vec algorithm (ARS $\approx$ AMIS ≈ 0.3). Similar results can be observed for clustering based on a feature vector enriched with the semantic features (i.e., named entities), lexical features, syntactic features, and all their combinations. The worst results were observed in the case of a vector composed exclusively of semantic features (ARS $= 0.11$, AMIS $= 0.07$). Interestingly, comparatively good results were reported for a combination of syntactic and lexical analysis, excluding Doc2Vec features (ARS $= 0.24$, AMIS $= 0.23$).

This demonstrates that most levels of analysis might provide a valuable contribution to the resulting text document vector representation. Because, at the core, the current problem of text document clustering is similar to the supervised classification of text documents, it could have been anticipated that the features based on the Doc2Vec algorithm would show good results as methods based on the vector representation of distributive semantic features are currently the state-of-the-art in this problem domain. On the other hand, the relatively low score for a feature vector composed exclusively of lexical features is somewhat surprising. Although specific analysis turned out to be the least effective analysis method in the context of clustering, it might still be helpful for some particular use cases of the system in the future. It is also possible that the clustering method based on the centroid algorithm (k-means) used in this study is not the best choice given the nature of the feature vector and that other methods might demonstrate significantly different results.

5. Conclusions

Extracting information from short, informal texts full of specialized jargon and grammatical errors of various kinds found on public and restricted multilingual websites to detect criminal activity and grouping these texts to identify potential criminal groups is a particularly complex task. The evaluation of our system showed that the use of advanced, comprehensive linguistic analysis carried out simultaneously on several levels: semantic, structural/specific, lexical, and syntactic, and the use of rule-based and neural classification methods, even on a modest amount of text resources, brings good results. We expect that as the collection of criminal texts expands and our domain lexicon is supplemented with new terms related to crime that come into use, the classification based on neural networks will be dominant. However, at the present research stage, a complete exclusion of rules seems to be inefficient.

References

Adnan, Kiran and Rehan Akbar, 2019. Limitations of information extraction methods and techniques for heterogeneous unstructured big data. *International Journal of Engineering Business Management*, 11:1847979019890771.

Andrews, Simon, Ben Brewster, and Tony Day, 2018. Organised crime and social media: a system for detecting, corroborating and visualising weak signals of organised crime online. *Security Informatics*, 7(1):3.

Bird, Steven, 2006. NLTK: The Natural Language Toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*. Sydney, Australia: Association for Computational Linguistics.

Brown, Donald E., 1998. The Regional Crime Analysis Program (ReCAP): a framework for mining data to catch criminals. In *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218)*, volume 3.

Demenko, Grażyna, 2015. *Korpusowe badania jezyka mówionego*. Lingwistyka Komputerowa. Warszawa: Akademicka Oficyna Wydawnicza EXIT.

Emmons, Scott, Stephen Kobourov, Mike Gallant, and Katy Börner, 2016. Analysis of network clustering algorithms and cluster quality metrics at scale. *PLOS ONE*, 11(7):1–18.

Erkan, Günes and Dragomir R Radev, 2004. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Hamilton, William L, Rex Ying, and Jure Leskovec, 2017. Representation learning on graphs: Methods and applications. *IEEE Data Engineering Bulletin*, 40(3):52–74.

Kieraś, Witold and Marcin Woliński, 2017. Morfeusz 2 – analizator i generator fleksyjny dla jezyka polskiego. *Jezyk Polski*, XCVII(1):75–83.

Krauz, Antoni, 2017. Mroczna strona internetu – TOR niebezpieczna forma cybertechnologii. *Dydaktyka Informatyki*, 12:63–74.

Ku, Chih Hao, Alicia Iriberri, and Gondy Leroy, 2008. Natural language processing and e-Government: Crime information extraction from heterogeneous data sources. In *Proceedings of the 2008 International Conference on Digital Government Research*, dg.o '08. Digital Government Society of North America.

Le, Quoc and Tomas Mikolov, 2014. Distributed representations of sentences and documents. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*. Bejing, China: PMLR.

Loper, Edward and Steven Bird, 2002. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics.

Mider, Daniel, 2019. Czarny i czerwony rynek w sieci The Onion Router – analiza funkcjonowania darkmarketów. *Przeglad Bezpieczeństwa Wewnetrznego*, 11(21):154–190.

Narayanan, Annamalai, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal, 2017. graph2vec: Learning distributed representations of graphs. In *Proceedings of the 13th International Workshop on Mining and Learning with Graphs (MLG)*.

Nath, Shyam Varan, 2006. Crime pattern detection using data mining. In *2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops*.

Nivre, Joakim, Johan Hall, and Jens Nilsson, 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *LREC*, volume 6.

Nowakowski, Artur and Krzysztof Jassem, 2021. Neural translator designed to protect the eastern border of the European Union. In *Proceedings of Machine Translation Summit XVIII: Users and Providers Track*. Virtual: Association for Machine Translation in the Americas.

Rowe, Ryan, German Creamer, Shlomo Hershkop, and Salvatore J Stolfo, 2007. Automated social hierarchy detection through email network analysis. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, WebKDD/SNA-KDD '07. New York, NY, USA: Association for Computing Machinery.

Rozemberczki, Benedek, Oliver Kiss, and Rik Sarkar, 2020. Karate Club: an API oriented open-source Python framework for unsupervised learning on graphs. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*.

Sammut, Claude and Geoffrey I. Webb (eds.), 2010. *TF–IDF*. Boston, MA: Springer US, pages 986–987.

Shervashidze, Nino, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt, 2011. Weisfeiler-Lehman graph kernels. *Journal of Machine Learning Research*, 12(9).

Skórzewski, Paweł, Mikołaj Pieniowski, and Grazyna Demenko, 2022. Named entity recognition to detect criminal texts on the web. In *Proceedings of the Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association.

Waszczuk, Jakub, Witold Kieraś, and Marcin Woliński, 2018. Morphosyntactic disambiguation and segmentation for historical Polish with graph-based conditional random fields. In Petr Sojka, Aleš Horák, Ivan Kopeček, and Karel Pala (eds.), *Text, Speech, and Dialogue*. Cham: Springer International Publishing.

Wróblewska, Alina, 2014. *Polish Dependency Parser Trained on an Automatically Induced Dependency Bank*. Ph.D. dissertation, Institute of Computer Science, Polish Academy of Sciences, Warsaw.

A Study in the Generation of Multilingually Parallel Middle Sentences

Matthew Eget, Xuchen Yang, Yves Lepage

IPS, Waseda University
2-7 Hibikino, Kitakyushu, 808-0135, Japan
{matthew.eget@toki., yang_xuchen@asagi., yves.lepage@}waseda.jp

Abstract

Multilingual parallel sentences are a precious resource for machine translation. However, such verified data is hard to find, evaluate, and generate. Computing middle sentences is a new data generation technique in which the middle of two sentences is created. The middle sentence is a new sentence at minimal and equal distance to the start and end sentences. By using this technique, in theory, multilingual parallel sentences can be generated. By utilizing a tri-lingual parallel sentence corpus and trained decoders, we compute middle sentences monolingually and check for their correspondence in meaning across languages. Our results show that parallel data can be generated in this way under the right circumstances. We show that the similarity between the start and end sentences has a significant impact on the quality of multilingual parallelism of the generated middle sentences.

1. Introduction

Multilingual parallel sentences, i.e., tuples of sentences in several languages that have exactly the same meaning, are a rare yet indispensable resource in machine translation. Parallel data is often limited to some language pairs, like French and English. It is difficult to unearth such data for less common language pairs, like Japanese and Chinese, and it is difficult to find such data for more than two languages, e.g., English—French—Japanese simultaneously. In addition, while there exist various competing methods for testing parallelism, there is no simple and accurate way of doing so, resulting in evaluation inconsistencies. Generating such data is a promising direction, but controlling the generation is a difficult, unsolved problem.

We propose to compute middle sentences to generate multilingual parallel data, i.e., to use the mean of the vector representations of a given start sentence and a given end sentence as guidelines to newly generate a middle sentence. By construction, it is expected that the new middle sentence should meet the basic properties associated with the middle of two values: being relatively close to both the start and the end sentences while being more similar to each of them than they are to each other. Examples of middle sentences are shown in Table 1. Here, our focus is less on what makes a middle sentence, but rather on the technique used to generate multilingual parallel data.

2. Previous Work and Proposed Method

2.1. Background

Previous work on generating middle sentences focused on monolingual text generation. Middle sentence generation is based on the concept of text morphing (Huang et al., 2018), and was proposed as a new way to generate text morphing data (Wang et al., 2021). Further middle sentence generation work includes generating text morphing data by combining with the sentence-level analogy (Pan et al., 2022) and data augmentation for style transfer (Osawa and Lepage, 2021). In both works, results suggest that, since generation is based on word or sentence embeddings, generated sentences retain stylistic properties relative to their start and end sentences. These findings are the most applicable to our work, as parallel data should also retain relationships between the start and end sentences similarly across different languages. The originality of our work resides in the investigation of the computation of middle sentences for the creation of multilingual parallel data.

There have also been various works in multilingual embedding spaces. LASER (Artetxe and Schwenk, 2019) and mUSE (Chidambaram et al., 2019) map multilingual sentences and words into the same embeddeding space, respectively, to solve multilingual problems using only a single model. Unlike the above two methods, DistilmBERT (Reimers and Gurevych, 2020) uses a monolingual embedding space to generate sentence vectors, and maps the vectors of the translated sentences in the target language to the same embedding space as the source language. For our research, we use DistilmBERT both as an embedding space for some experiments' decoders as well as an effective means for comparing the similarity between multilingual generated sentences.

2.2. Justification and Method

The justification for our method in a scenario of data augmentation lies in the size of the expected newly created data.

In the best case, it is quadratic in the size of the initial data. Let us suppose that we start with a relatively small initial corpus of one thousand parallel sentences in, say, three languages English—French—Japanese. We might expect to create $(1,000 \times 1,000) \ / \ 2 = 500,000$ new parallel triples of sentences (the division by 2 comes from the symmetrical role of the start and end sentences).

Now, in the worst case where the start and the end sentences, s and e, are the same, the vector representations are the same ($\vec{s} = \vec{e}$) and thus, the middle sentence, m, should be as follows:

$$\vec{m} = \frac{1}{2} \times (\vec{s} + \vec{e}) = \vec{s} = \vec{e}$$

The third row of Table 1 displays the actual output obtained for this trivial case using our method. In this case, the size of created data is linear in the size of the initial

Start sentence	Middle sentence	End sentence
it 'll snow today .	it is not clear for today .	there are no comments yet .
she likes apples .	i eat apples .	i eat everyday .
i 'm busy now .	i 'm busy now .	i 'm busy now .

Table 1: Examples of middle sentences computed on start and end sentences from the Tatoeba[2] corpus. The last row is an example of the trivial case, or when the start and end sentence are the same.

Experiment Name	Use of Buckets	Embedding Space	Number of Sentence Pairs	Dataset
fastText without Buckets	No	fastText	100,000	Randomly Selected
fastText with Buckets	Yes	fastText	25,000	BAMD
DistilmBERT with Buckets	Yes	DistilmBERT	25,000	BAMD

Table 2: Experiment settings — model structures vary by dataset and embedding space used. The "BAMD" dataset comprised of five buckets of start-end sentence pairs based on cosine similarity.

data, but indeed the data is not new, so the size of newly created data is actually a constant of zero. In between the best and the worst cases, we expect to create data with a size between linear and quadratic.

If we consider the worst case's ability to generate, by construction, corresponding data across multiple languages when the inputs are the same, generating multilingual, parallel data with different start and end sentences should also be possible. In other words, given multilingual, semantically parallel sentences, we should be able to generate new data for each language that is also parallel with the other languages. While the trivial case shows the possibility of our proposal, it also is naïvely simple. When reaching across language boundaries, language word embedding spaces and monolingual systems exhibit discrepancies, resulting in similar, but non-parallel data. As a result, we investigate various methods to improve parallelism between middle sentences.

The computation for the middle sentence vector for this research is as follows:

$$\vec{m} = \frac{1}{2} \times (\vec{s} + \vec{e})$$

Where $\vec{m}$ is computed from a start and an end vector, $\vec{s}$ and $\vec{e}$ respectively, which are obtained from a given start and end sentence. The final text representation of the middle sentence is computed from the input of the middle sentence vector on its respective decoder.

3. Experiments

3.1. Dataset

Our data consists of English–Japanese–French semantically parallel sentences from the Tatoeba corpus. The English sentences were selected by using a maximal length of 20 words. We use a 68k / 8k / 8k training / test / validation split. We tokenized the sentences using SpaCy[3] pre-trained small models for each language (Honnibal and Montani, 2017). We use all possible combinations for our test set when generating middle sentences, i.e., $8,000 \times 8,000/2 = 32,000,000$ combinations, leading to the same amount of middle sentences in all languages.

[3] https://spacy.io

3.2. Experiments

Our basic method uses an auto-encoder (AE) model to generate middle sentences (Wang and Lepage, 2020). The encoder's embedding space and the decoder model itself vary from experiment to experiment (as a result of different embedding spaces), but all use Tatoeba data. The middle sentences are computed monolingually, so there is a separate decoder for each language. Each decoder is separately trained on the training datasets using a given word embedding space, where the training/test/validation corpus is expected to be multilingual parallel data.

We present three experiments in this paper, all using AE models.

Our first experiment, fastText without Buckets, uses the fastText (Bojanowski et al., 2017) embedding space and generates 100,000 middle sentences, each language on its respective embedding space.

Our second experiment, fastText with Buckets, is the same as the first, but on our specialized BAMD dataset.

Our final experiment, DistilmBERT (Reimers and Gurevych, 2020) with Buckets, uses the DistilmBERT embedding space on our BAMD dataset. An overview of our experiments and their data is given in Table 2.

3.2.1. Parallelism Evaluation

For all sentence comparisons, we use the DistilmBERT model to compute cosine similarity. Part of the challenge associated with our task is to confirm what is "parallel" and "not parallel". For example, if we compare fastText's average of word embeddings and DistilmBERT's sentence embedding space for a given sentence, the cosine similarity will be different. Throughout our experiments, we found that DistilmBERT was the most consistent for high scores, and as such, is the best fit for our task. DistilmBERT is also capable of multilingual sentence embedding comparison, which avoids the use of translation needed for comparison across fastText monolingual embeddings. Cosine similarity scores range from 0.0 to 1.0. A value of 1.0 means an exact match, thus the same meaning monolingually and multilingually is generated. The trivial case mentioned in Section 2.2. corresponds to a cosine similarity of 1.0. We consider that cosine similarity scores above

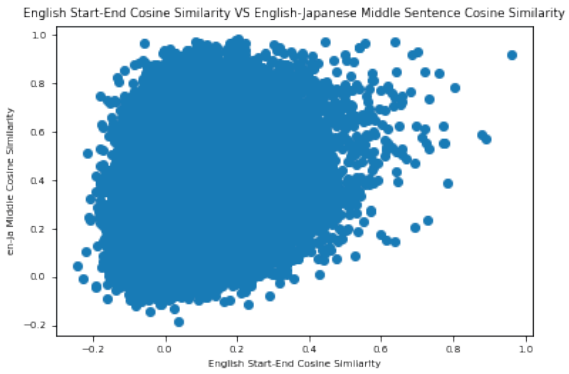


Figure 1: English start–end (SE) vs. English-Japanese middle sentence cosine similarity from fastText without Buckets

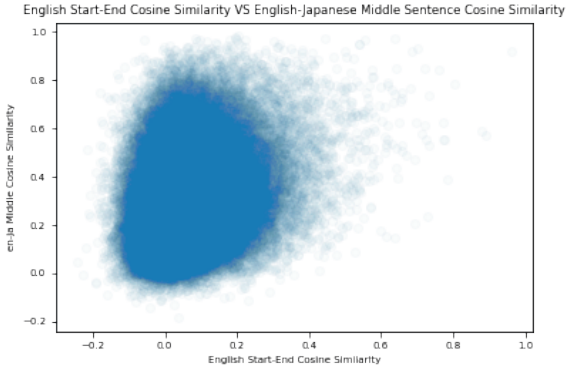


Figure 2: Same as Fig. 1 but with high-point transparency (better visualization of clusters)

0.9 reflect parallel candidates.

3.2.2. fastText without Buckets Experiment

In the first, preliminary experiment, we randomly select 100,000 sentence pairs from our test set start–end combinations. However, randomly selecting sentence pairs produced menial results, as few multilingual parallel sentences were generated as seen in Table 3. However, when considering Fig. 1 and Fig. 2, there is a trend in the relationship between English start–end cosine similarity and multilingual middle sentence similarity — there appears to be a relatively positive correlation where higher multilingual middle cosine similarities are found in conjunction with higher English start–end cosine similarity.

3.2.3. fastText with Buckets / DistilmBERT with Buckets Experiments

Based on the previous experiment, we further investigate a potential relationship between monolingual start–end cosine similarity and multilingual middle sentence cosine similarity. We split the 30,000,000 sentence pairs into "buckets" based on their English start–end cosine similarity, (0.0 to 0.2, 0.2 to 0.4... 0.8 to 1.0). This allows us to see if monolingual start–end cosine similarity is an impacting factor on the cosine similarity of the gener-

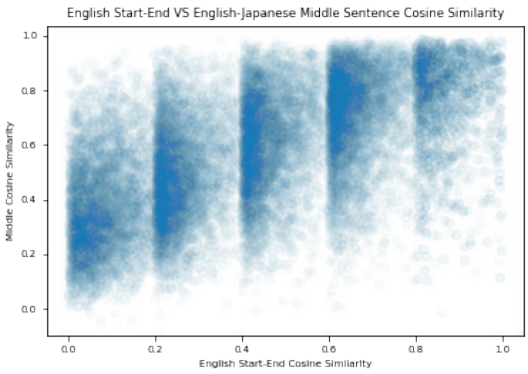


Figure 3: English start–end (SE) cosine similarity vs. English–Japanese middle sentence cosine similarity from DistilmBERT with Buckets. English–French figure not shown as results are comparable.

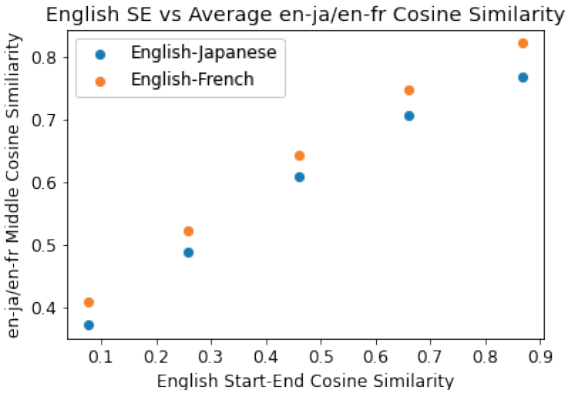


Figure 4: English start–end (SE) cosine similarity vs. average middle sentence cosine similarity per bucket for English—Japanese (en-ja) and English—French (en-fr) from DistilmBERT with Buckets

ated multilingual middle sentences. For this new dataset that we call BAMD, we randomly sampled 5,000 sentence pairs to fit into each bucket according to their start–end cosine similarity. However, we were not able to gather 5,000 sentences to fill the 0.8-1.0 bucket, so this bucket remains smaller with around 2,000 sentence pairs. In total, we have five buckets of increasing cosine similarity ranges, from 0.0-0.2 up to 0.8-1.0, with the last bucket being smaller than the rest.

Experiments using the BAMD dataset are introduced in Table 3. We checked that, in our data, there is no significant language difference in parallel start–end pair cosine similarity (same sentence pair in a different language). Consequently, we will just be comparing with English start–end similarity in this paper.

4. Results and Analysis

4.1. Results

Our results from DistilmBERT with Buckets for English–Japanese are shown in Fig. 3, with a Pearson's correlation of 0.66. For different visualization, from DistilmBERT with Buckets experiment, the average multilingual middle

Embedding space	Use of buckets	Bilingual parallel (ratios: $\times 10^{-3}$)						Trilingual parallel (ratios: $\times 10^{-3}$)	
		English–Japanese		English–French		Japanese-French			
fastText	No	88	(0.9)	206	(2)	79	(0.8)	4	(0.04)
	Yes	609	(30)	819	(40)	590	(30)	182	(10)
DistilmBERT		1,218	(50)	2,110	(90)	1,376	(60)	475	(20)

Table 3: Amount of sentences produced with a 0.90 cosine similarity or higher. Absolute numbers followed by ratio of parallel middle sentences over the total generated middle sentences (to be multiplied by 10^{-3}) in parentheses.

English	Japanese	French
he's tired tired .	彼 は 疲れ 疲れ た 。	il est fatigué fatigué .
you've changed .	あなた は 変わった ね 。	tu as changé .
i hope it would rain tomorrow .	明日 は 雨 が 降る そう だ と 思う 。	il serait qu' il va pleuvoir demain .
i have to obey the rules .	私 は 規則 を 守ら なけれ ば なら ない 。	je dois obéir les règles .
i have given up smoking .	私 は 煙草 を やめ た 。	je suis arrêté de fumer .

Table 4: Example results of trilingual parallel middle sentences from DistilmBERT with Buckets with a cosine similarity above 0.90. Notice that the repetition of a word on the first row is consistently found across all languages, ensuring high parallelism, despite being questionable grammatically.

sentence cosine similarity per bucket is shown in Fig. 4. As visible in this graph, there is a clear correlation between the buckets and the average scores, with an almost ideal Pearson's correlation of 0.99.

For data evaluation, our primary interest is the amount of multilingual middle sentences produced which have a cosine similarity of 0.90 or higher with the other language's middle sentences. A high cosine similarity of 0.90 does not mean that the data is definitely parallel, but rather that the data is highly similar. The quality of results does not vary significantly among our primary experiments, but the numbers of produced sentences do vary. Table 3 shows the number of sentences generated with cosine similarities over 0.90 per experiment model.

We also have two additional categories for our generated sentences, trilingual parallel, and bilingual parallel, where former sentences are parallel in all three languages and latter sentences are only in two languages (or one language pair). Trilingual parallel results are shown in Table 4 with bilingual parallel shown in Table 5. As before, the example results are all from the DistilmBERT with Buckets experiment where multilingual middle cosine similarities are 0.90 or higher.

4.2. Analysis and Future Work

As shown in Table 3, we were able to generate parallel data, but the majority of generated multilingual middle sentences although similar are not necessarily parallel.

Another interesting trend observed in all our data is that there is a strong correlation between the start and end cosine similarity and the generated multilingual middle sentence cosine similarity. As seen in Figures 3 and 4, when the start–end cosine similarity is low, there is a higher chance for the generated sentence to be of low cosine similarity to its other language counterparts. In contrast, when the start–end similarity is higher, the chance of generating data closer in meaning is also higher. In this case, the generated middle sentences can be considered translations of one another. This ultimately suggests that the start–end sentence vector similarity plays a significant role in the generation of a middle sentence.

While perfectly parallel middle sentences can be generated from any start–end similarity value, the likeliness of parallelism of the middle sentences increases as start–end similarity increases. We have found closely parallel data, if not perfectly parallel, at cosine scores lower than 0.90. This may indicate that 0.90 cosine similarity is indeed too strict, even for testing parallel data, which leaves possibilities for the creation of much more parallel data.

5. Conclusion

We inspected how to successfully generate multilingual, semantically parallel data by computing middle sentences. Our results show that moderately similar sentences are best for generating middle sentences with the same meaning across several languages, and that, in general, the closer the start and end sentence, the higher the chance that the middle sentences generated be parallel in meaning. We will further explore the reasons why middle sentences are not generated at lower similarities.

We will also apply our generated data to downstream tasks like machine translation and conduct extrinsic evaluation of this generated data. A promising application is machine translation for low-resource language pairs like French-Japanese.

6. Acknowledgement

The work reported in this paper has been supported in part by a grant for research (Kakenhi C) from the Japanese Society for the Promotion of Science (JSPS), n° 21K12038 "Theoretically founded algorithms for the automatic production of analogy tests in NLP."

References

Artetxe, Mikel and Holger Schwenk, 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

English–French	
it 's very cold this morning .	il fait très froid aujourd'hui .
i 'm glad to see you .	je suis content de vous voir .
tom sat on the sofa .	tom s' assit sur le canapé .
i 'd like to eat a few of cake .	j' aimerais manger un peu de gâteau .
English–Japanese	
what are you afraid of ?	あなた は 何 が 怖い の ？
i drank a coffee .	私 は コーヒー を 飲ん で だ 。
tom caught a big fish .	トム は 大きな 魚 を やっ た 。
do you have anything else on ?	何 か 他 の もの を 持っ て い ます か 。
French–Japanese	
tu es médecin ?	あなた は お 医者 さん です か 。
c' est une beauté séduisante .	それ は 、 美し な の でしょう 。
c' est un joueur de baseball .	彼 は 野球 選手 です 。
tom a laissé la ville .	トム は 町 から 離れ て いっ た 。

Table 5: Example results of bilingual parallel middle sentences from DistilmBERT+Buckets with a cosine similarity above 0.90.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov, 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Chidambaram, Muthu, Yinfei Yang, Daniel Cer, Steve Yuan, Yunhsuan Sung, Brian Strope, and Ray Kurzweil, 2019. Learning cross-lingual sentence representations via a multi-task dual-encoder model. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*.

Honnibal, Matthew and Ines Montani, 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.

Huang, Shaohan, Yu Wu, Furu Wei, and Ming Zhou, 2018. Text morphing. *arXiv preprint arXiv:1810.00341*.

Osawa, Koki and Yves Lepage, 2021. Data augmentation of parallel data for style transfer by generation of middle sentences (in japanese). In *Proceedings of the 28th Annual Conference of the Association for Natural Language Processing*. Hamamatsu, Japan: Association for Natural Language Processing.

Pan, Zhicheng, Xinbo Zhao, and Yves Lepage, 2022. Sentence analogies for text morphing. In *Proceedings of the workshop 'Analogies: from Theory to Applications (ATA@ICCBR 2022)', held in conjunction with the 30th International Conference on Case-Based Reasoning (ICCBR)*. Nancy, France.

Reimers, Nils and Iryna Gurevych, 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics.

Wang, Liyan and Yves Lepage, 2020. Vector-to-sequence models for sentence analogies. In *Proceedings of the 2020 International Conference on Advanced Computer Science and Information Systems*.

Wang, Pengjie, Liyan Wang, and Yves Lepage, 2021. Generating the middle sentence of two sentences using pre-trained models: a first step for text morphing. In *Proceedings of the 27th Annual Conference of the Association for Natural Language Processing*. Kokura, Japan: Association for Natural Language Processing.

Automatic Classification of Spontaneous vs. Prepared Questions in Speech Transcriptions

Iris Eshkol-Taravella[1,2], Angèle Barbedette[3], Xingyu Liu[2], Valentin-Gabriel Soumah[2]

[1]MoDyCo UMR7114
ieshkolt@parisnanterre.fr
[2]Université Paris Nanterre
xingyu.liu@univ-grenoble-alpes.fr
soumahvg@gmail.com
[3]ERTIM, INALCO
angele.barbedette@gmail.com

Abstract

This work aims at developing a linguistic model to automatically classify questions from speech transcripts of ESLO2 and ACSYNT corpora into two categories "spontaneous" and "prepared". We first provide a list of criteria to define questions. Experiments based on supervised machine learning methods are conducted using a multiclass classification including "spontaneous", "prepared" and "non-question" categories and a binary classification including "spontaneous" and "prepared" categories only. The best results for traditional machine learning methods are obtained with a logistic regression combined with significant linguistic criteria only (F-score of 0.75). Finally, these results are compared with those obtained using deep learning techniques.

Keywords: spontaneus speech, prepared speech, speech transcriptions, questions, automatic classification, supervised machine learning

1. Introduction

Spontaneous speech is often associated with a very limited use of spoken language. However, there are different forms of spoken language that are not only restricted to spontaneous conversations, such as interviews for example (Biber and Finegan, 2008). Spoken language is in fact a continuum of various situations such as read speech, professional speech (used by journalists or teachers) or speech recorded in the most natural conditions (during a meal, in a familiar environment, with close friends and family for example) (Delgado-Martins and Freitas, 1991). Spontaneous speech is built during speech production while prepared speech is based on the elaboration of speech before its production : prepared speech implies a spatio-temporal separation between the discourse elaboration and its production (Dutrey et al., 2014; Jousse et al., 2008; Guerin, 2015). Various indicators have been defined to characterize spontaneous speech. In (Jousse et al., 2008; Shriberg et al., 2009), the authors consider that the degree of spontaneity of speech can be partly established on the basis of prosodic criteria such as vowel duration or word final lengthening but also on the presence of disfluencies, i.e., interruptions in speech that do not bring any additional information to the interaction.

In this work, we focused on interrogative utterances from oral transcripts to study and identify spontaneous speech, while not taking prosodic information into account. We think the results obtained can be valuable to be able to distinguish a spontaneous question from a prepared one in the context of political speech analysis for example. The results could also be used in automatic text generation systems to improve some aspects of human-computer dialogue or chatbots by making the speech more natural, as spontaneity is a characteristic of human language (Qader et al., 2017).

2. Datasets and preprocessing

The data was collected via the Ortolang platform (www.ortolang.fr) and come from various dialogue and discussion situations of two French oral corpora ESLO2 (Baude and Dugua, 2011; Eshkol-Taravella et al., 2011) and ACSYNT (CLLE-ERSS, 2013). The data are composed of :

- The guided interviews and the guided interviews of public figures modules (and their associated guidelines), the cinema module (recordings and transcriptions of people giving their opinions at the exit of movie theaters) and the meals module (recordings and transcriptions of people sharing a meal and discussing various subjects) from ESLO2 ;
- The guided interviews (and their associated guidelines) from ACSYNT.

Questions were automatically extracted using the transcribed question mark, as well as the five speech turns preceding and following each question. This five turns threshold is considered sufficient and not too large to provide potentially relevant information to facilitate the manual annotation of the questions. Also, it will allow us to determine context-based linguistic features that will help the automatic classification. The final cleaned corpus contains 1298 samples from ESLO2 and 588 from ACSYNT.

3. Manual Labelling

The Dynamic Interpretation and Dialogue Theory (DIDT) enables us to define a question as a task-oriented dialogue act that consists of an information retrieval (Bunt, 1999). Thus, non-questions are defined as any other act of dialogue having an interrogative syntactic form but which does not consist in information seeking. Consistently with the DIDT, the Dialog Act Markup in Several Layers or DAMSL (Jurafsky, 1997) explains that a question belongs to the field of information request. According to (Ginzburg and Sag, 2000), an interrogative utterance is made of propositions that constitute a question. A question is determined on the basis of pragmatic, syntactic and semantic criteria. For example, a declarative question is syntactically not a question, but its meaning is consistent with an information request. On the contrary, a rhetorical question is syntactically a question, but no request for information is made. From the commitment perspective (Beyssade and Marandin, 2009), a question is a speech act that both engages the speaker as "desiring to get information" but also invites the interlocutor to agree to accept this act as a desire to get information.

The corpus was split among seven annotators and guidelines were established in advance of the task to ascertain the criteria to rely on and the labels to use to annotate the questions. The first step of each question analysis is to determine whether the phrase to annotate is a question, a non-question or a non annotable question.

- Non questions :
  - Tag questions : ways to make sure the interlocutor has processed and accepted what has been said (Bunt, 1999) ;
  - Repeat requests : interrogative clarification request (Ginzburg, 2012; Purver et al., 2003; Boritchev, 2021) ;
  - Understanding verifications : similar to repeat requests and tag questions (Bunt, 1999). The speaker makes sure that he has understood the previous speech turn by repeating a part or a whole. (Ginzburg, 2012; Purver et al., 2003; Boritchev, 2021) consider these utterances as clarification requests ;
  - Rhetorical questions : despite having question-like syntax, does not need to be answered (Boritchev, 2021) ;
  - Injunctive questions : action requests ;
  - Social obligation questions : not motivated by the seeking of new information, but are the result of social obligations. They are acts of dialogue control, part of communication management (Bunt, 1999; Jurafsky, 1997) ;
- Non annotable questions (not usable because of a pre-existing transcription issue) : cut questions, overly broad questions, reported speech or questions that cannot be understood ;
- Questions (requests for information that adds value to the ongoing dialogue) :
  - Wh-questions : contain an interrogative word and expect neither "yes" nor "no" as an answer (Bunt, 1999; Jurafsky, 1997; Ginzburg and Sag, 2000; Boritchev, 2021) :
  - YN-questions : expect "yes" or "no" as an answer ;
  - Alternative questions : possible answers are included in the question (Bunt, 1999; Aarts et al., 2018; Boritchev, 2021).

If the utterance appears to be a real question, it can then be classified as a spontaneous or prepared question. The criteria listed in table 1 are helpful for the manual labelling task but not definitive: they are combined with the overall understanding of the question and its context of production. The interview guidelines from ESLO2 and ACSYNT are also considered during the annotation.

The inter-annotator agreement was calculated with the Cohen's Kappa on the manual annotation of 200 questions produced by two expert annotators. The value of Kappa obtained is 0.75 which can be considered as a substantial agreement according to (Landis and Koch, 1977).

4. Automatic Classification

Experiments based on supervised machine learning methods were performed according to both a multiclass classification including the "spontaneous", "prepared" and "non-question" categories and a binary classification including the "spontaneous" and "prepared" categories only. They were done on an annotated dataset composed of 731 spontaneous questions, 478 prepared questions and 284 non-questions from the ESLO2 and ACSYNT corpora that were divided into 0.75 and 0.25 for the training and test datasets respectively. The questions were represented with vectors obtained with the CBOW model based on the FrWaC corpus (200 dimensions) (Fauconnier, 2015). Questions normalization was obtained using TF-IDF weights to take into account the importance, rarity, and discriminative function of the corpus words. The Skip-Gram model was also tested for the experiments but yielded overall inferior results. Our selection of relevant criteria for the classification task is based on the linguistic criteria mentioned in (Blanche-Benveniste and Bilger, 1999), as well as on observations made from an extract of the corpus and intuitions :

1. Length of the question (number of tokens)
2. Presence of disfluencies (ratio between the number of disfluencies and all words) ;
3. Presence of an upcoming question announcement (binary feature) : prepared questions can sometimes be announced before being actually asked ;
4. Presence of a subject inversion (binary feature) : subject inversion is more likely to appear in prepared questions ;
5. Presence of the repetition of a named entity (binary feature) : repeating a named entity is typical of prepared questions while using an anaphora is not ;
6. Vectorial distance with Word2Vec vectors between the question and the preceding context : the purpose of this measure is to highlight the topic change that can appear in prepared questions ;

Spontaneous	Anaphoras	spk1 : je suis contrôleur divisionnaire aux PTT	spk1 : I am a divisional inspector at the Post Office
		spk4 : et en quoi *ça* consiste ?	spk4 : and what is *it* about?
	Clarification requests or request for additional information (Purver, 2004)	spk1 : mais // ce qui serait intéressant c'est que justement // on puisse euh // les enfants puissent apprendre de très bonne heure // euh très jeunes // une ou deux langues étrangères une au moins	spk1 : but // what would be interesting is if // we could hum // children could learn very early // hum at an early age // one or two foreign languages one at least
		spk4 : *et pourquoi ?*	spk4 : *and why?*
	Topic preceding the comment[1]	spk1 : mais alors là vous êtes en France pour combien de temps maintenant ?	spk1 : but then you are in France for how long now?
	Final interrogative word	spk3 : oui non mais cette équipe elle est constituée *comment* ?	spk3 : yes no but this team is constituted *how*?
	Disfluencies	spk4 : que feriez-vous de *de* ce temps libre ?	spk4 : what would you do with *with* this free time?
Prepared	Named entity repetition	spk4 : depuis combien de temps habitez-vous Orléans ?	spk4 : since when do you live in Orléans?
		spk1 : oh ça fait neuf ans depuis dix neuf cent soixante	spk1 : oh it's been nine years since nineteen sixety
		spk4 : vous vous plaisez à *Orléans* ?	spk4 : do you like *Orléans*?
	Topic change	spk1 : oui oui // et vous comptez rester ?	spk1 : yes yes // and do you plan to stay?
		spk2 : oui enfin // tant que je serai célibataire	spk2 : yes I mean // as long as I'll be single
		spk1 : ah oui	spk1 : ah yes
		spk2 : après on en sait rien de ça	spk2 : but we don't know about that
		spk1 : *alors est-ce qu'on pourrait parler un peu de votre travail ?*	spk1 : *so could we talk a bit about your job?*
	Comment preceding the topic[2]	spk4 : qu'est-ce qui vous plaît dans votre travail ?	spk4 : what do you like about your job?
	Subject inversion	spk4 : *faites-vous* un brouillon ?	spk4 : *are you* making a draft?
	Initial interrogative word	spk4 : et *qu*'est-ce que vous pensez du latin à l'école ?	spk4 : and *what* do you think about Latin in school?
	Upcoming question announcement	spk4 : spk4 : alors maintenant je vais vous poser des questions euh // peut-être un peu // un peu plus pers- personnelle mais	spk4 : so now I'm going to ask you some questions hum // that may be // a little more pers- personal but

Table 1: Indicators that help defining spontaneous and prepared questions, associated with examples and their translations

7. Position of the interrogative word (three binary features depending on whether the interrogative word is in initial (a), intermediate (b) or final (c) position) : the interrogative word is more likely to appear in the end of a spontaneous question ;
8. Presence of the interrogative form "est-ce que" (binary feature) : this form is more likely to appear in spontaneous questions ;
9. Presence of the subject inversion marker "t-il" (binary feature) : this marker is more likely to appear in prepared questions.

The criteria were extracted from the following Python libraries : SpaCy (Honnibal and Montani, 2017) for named entity recognition and POS tagging, NLTK (Bird et al., 2009) for tokenization and lemmatization, Gensim (Rehurek and Sojka, 2011) and Sk-Learn (Pedregosa et al., 2011).

The results of our experiments with the different algorithms are summarized in table 2. They were obtained without balancing the data in order to keep the natural distribution in each category. Also, the proportion of the minority class was considered sufficient to run the algorithms as it represents 19% of the data used for the multiclass classification and 39,5% of the data used for the binary classification.

[1]The topic and comment concepts are related to the given vs. new information distinction (Gundel, 1988). In this example, the new information about the topic "mais alors là vous êtes en France", i.e. the comment, is brought by the question "pour combien de temps maintenant" which is following the topic.

[2]This example shows that the topic "dans votre travail" is following the new information, i.e. the comment, that is conveyed by the question "qu'est-ce qui vous plaît".

The best results are obtained with a score of 0.74 for the binary classification (spontaneous and prepared questions only) and with a score of 0.66 for the multiclass classification (spontaneous, prepared and non-questions) when running the logistic regression algorithm with linguistic features only.

	W2V and TF-IDF	Linguistic features	Combined features
RF (multi)	0.53	0.58	0.6
KNN (multi)	0.51	0.55	0.49
SVM (multi)	0.59	0.59	0.6
LR (multi)	0.55	**0.66**	0.6
RF (bin)	0.6	0.68	0.59
KNN (bin)	0.6	0.69	0.61
SVM (bin)	0.61	0.45	0.62
LR (bin)	0.63	**0.74**	0.68

Table 2: Summary of the weighted average of F-scores obtained with the combination of each algorithm and selected features (Word2Vec and TF-IDF representation, linguistic features or both) using a multiclass or a binary classification with unbalanced categories

In order to analyse the relevance of linguistic features for the classification task, a logistic regression model was built using the Python statsmodels library (Seabold and Perktold, 2010) to get p-values for each of the eleven chosen linguistic features and for each of the two categories "spontaneous" and "prepared" questions. The features that appear to be significant and relevant for the classification (number of tokens, ratio between disfluencies and total amount of words, subject inversion, initial position of the interrogative word, presence of "est-ce que" and "-t-il"), i.e. that have a p-value lower than the 0.05 standard threshold, were selected to rerun the logistic regression algorithm for the binary classification. As shown in table 3, scores were slightly increased.

	Precision	Recall	F-score
Spontaneous	0.76	0.83	0.8
Prepared	0.74	0.64	0.69
Micro			0.75
Macro	0.75	0.74	0.74
Weighted average	0.75	0.75	0.75

Table 3: Results for the binary classification using logistic regression with relevant linguistic features only ($p<0.05$)

The results in table 2 show higher performances when using a binary classification. The multiclass classification gets the best precision and recall for spontaneous questions. Non-questions may share the same syntactic features as spontaneous questions, without actually being "real" questions, which could explain why they were more misclassified. Also, spontaneous questions are the majority class and non-questions are the minority class. To test the hypothesis of an effect on the results, the corpus was balanced by reducing the data and the algorithms rerun. The results of these new experiences are similar to the first ones, with scores for spontaneous questions that are generally lower, a recall that is higher for non-questions but with a precision lower than 0.6.

About the selection of relevant linguistic features only, the improvement is minor but the scores remained stable. The relevant features highlighted by the statistical analysis are relevant and sufficient enough to obtain the satisfactory score of 75% of good predictions and show a certain consistency with the manual analysis of the corpus which showed that disfluencies are representative of spontaneous questions and that an interrogative word at the beginning of a question is typical of prepared questions.

One of the possible developments for this work is to test deep learning methods with the same data and relevant linguistic features. Another possible outcome is the automatic prediction of questions vs. non-questions. Preliminary results were obtained first by reusing the defined criteria to classify spontaneous and prepared questions, and then by using a pre-trained language model, without additional criteria, to predict questions and non-questions :

- Forward Propagation Neural Network : the expert features were used to optimize the neural network for our data with the Keras python library (Chollet et al., 2015). A heuristic method was used to adjust the algorithm until the best possible results were obtained. The final architecture of the model is simple (three dense layers separated by dropout layers) and has been trained for 200 epochs. The corpus was balanced by reducing the data to avoid a huge gap between scores for spontaneous questions and for prepared questions. The training set contains 747 questions among which 142 are used for the validation set. The test set is composed of 239 examples. The weighted F-score obtained is 0.7 (table 4).
- CamemBERT (Martin et al., 2019) : the Simple Transformers library (Rajapakse, 2019) was used (default parameters, 5 epochs) with a balanced dataset composed of 956 questions. A weighted average F-score of 0.73 was obtained for the spontaneous/prepared question classification (table 4), a score close to the results obtained with the logistic regression combined with the significant features, with slightly improved results for prepared questions, and 0.84 for the question/non-question classification (table 5), a score that could be improved by determining relevant traits to distinguish these two categories.

5. Conclusion

The purpose of our study was to verify the existence of features allowing to distinguish spontaneous and prepared questions from speech transcriptions, without relying on the audio information. The best results that were obtained for traditional machine learning methods are with the logistic regression algorithm combined with relevant linguistic features only (F-score of 0.75).

According to all the results, our features are not relevant enough for non-questions detection and there seem to be common characteristics between non-questions and sponta-

	Forward Propagation NN			CamemBERT		
	Precision	Recall	F-score	Precision	Recall	F-score
Spontaneous	0.71	0.74	0.73	0.77	0.67	0.72
Prepared	0.68	0.65	0.67	0.71	0.8	0.75
Weighted average	0.7	0.7	0.7	0.74	0.73	0.73

Table 4: Results for the binary classification using a forward propagation neural network with relevant linguistic features only ($p<0.05$), using CamemBERT pre-trained language model

	Precision	Recall	F-score
Question	0.85	0.84	0.84
Non-question	0.84	0.85	0.85
Weighted average	0.84	0.84	0.84

Table 5: Results for the binary classification "question" vs. "non-question" using CamemBERT pre-trained language model

neous questions that complicate the classification task. The question vs. non-question classification with CamemBERT resulted in 84% of good predictions.

It would be interesting to refine the linguistic features to be more precise and to be able to discriminate spontaneous questions, prepared questions and non-questions, but also to extend this work to other types of interrogative utterances like injunctive of social obligation questions for example.

References

Aarts, B, S Chalker, E Weiner, and OU Press, 2018. The oxford dictionary of english grammar .(2014). *URL https://en. oxforddictionaries. com/definition/heterogeneous. Accessed*.

Baude, Olivier and Céline Dugua, 2011. (re) faire le corpus d'orléans quarante ans après: quoi de neuf, linguiste? *Corpus*, (10):99–118.

Beyssade, Claire and Jean-Marie Marandin, 2009. Commitment: une attitude dialogique. *Langue française*, (2):89–107.

Biber, Douglas and Edward Finegan, 2008. Longman: grammar of spoken and written english. In *Longman: grammar of spoken and written english*. pages 1204–1204.

Bird, Steven, Ewan Klein, and Edward Loper, 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Blanche-Benveniste, Claire and Mireille Bilger, 1999. Français parlé-oral spontané. quelques réflexions. *Revue française de linguistique appliquée*, 4(2):21–30.

Boritchev, Maria, 2021. *Modélisation dynamique des dialogues*. Ph.D. thesis, Université de Lorraine.

Bunt, Harry, 1999. Dynamic interpretation and dialogue theory. *The structure of multimodal dialogue*, 2:139–166.

Chollet, François et al., 2015. keras.

CLLE-ERSS, 2013. Acsynt. ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr.

Delgado-Martins, Maria Raquel and Maria Joao Freitas, 1991. Temporal structures of speech:" reading news on tv". In *Phonetics and Phonology of Speaking Styles*.

Dutrey, Camille, Sophie Rosset, Martine Adda-Decker, Chloé Clavel, and Ioana Vasilescu, 2014. Disfluences dans la parole spontanée conversationnelle: détection automatique utilisant des indices lexicaux et acoustiques. *XXXe Journées d'Étude sur la Parole (JEP'14)*:366–373.

Eshkol-Taravella, Iris, Olivier Baude, Denis Maurel, Linda Hriba, Céline Dugua, and Isabelle Tellier, 2011. Un grand corpus oral «disponible»: le corpus d'orléans 1 1968-2012.

Fauconnier, Jean-Philippe, 2015. French word embeddings.

Ginzburg, Jonathan, 2012. *The interactive stance*. Oxford University Press.

Ginzburg, Jonathan and Ivan Sag, 2000. *Interrogative investigations*. Stanford: CSLI publications.

Guerin, Emmanuelle, 2015. *Observer, décrire,... enseigner, le français" langue vivante"*. Ph.D. thesis, Univ. Poitiers.

Gundel, Jeanette K, 1988. Universals of topic-comment structure. *Studies in syntactic typology*, 17(1):209–239.

Honnibal, Matthew and Ines Montani, 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1):411–420.

Jousse, Vincent, Yannick Esteve, Frédéric Béchet, Thierry Bazillon, and Georges Linares, 2008. Caractérisation et détection de parole spontanée dans de larges collections de documents audio. *JEP*, 2008:9–13.

Jurafsky, Dan, 1997. Switchboard swbd-damsl shallow-discourse-function annotation coders manual. *Institute of Cognitive Science Technical Report*.

Landis, J Richard and Gary G Koch, 1977. The measurement of observer agreement for categorical data. *biometrics*:159–174.

Martin, Louis, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de La Clergerie, Djamé Seddah, and Benoît Sagot, 2019. Camembert: a tasty french language model. *arXiv preprint arXiv:1911.03894*.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al., 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Purver, Matthew, Jonathan Ginzburg, and Patrick Healey, 2003. On the means for clarification in dialogue. In *Current and new directions in discourse and dialogue*. Springer, pages 235–255.

Purver, Matthew Richard John, 2004. *The theory and use of clarification requests in dialogue*. Ph.D. thesis, Citeseer.

Qader, Raheel, Gwénolé Lecorvé, Damien Lolive, and Pascale Sébillot, 2017. Ajout automatique de disfluences pour la synthèse de la parole spontanée: formalisation et preuve de concept. In *Traitement automatique du langage naturel (TALN)*.

Rajapakse, T. C., 2019. Simple transformers. `https://github.com/ThilinaRajapakse/simpletransformers`.

Rehurek, Radim and Petr Sojka, 2011. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2).

Seabold, Skipper and Josef Perktold, 2010. Statsmodels: Econometric and statistical modeling with python. In *Proceedings of the 9th Python in Science Conference*, volume 57. Austin, TX.

Shriberg, Elizabeth, Benoit Favre, James Fung, D Hakkani-Tur, and Sébastien Cuendet, 2009. Prosodic similarities of dialog act boundaries across speaking styles. *Linguistic Patterns in Spontaneous Speech*, (A25):213–239.

XAI in Computational Linguistics: Understanding Political Leanings in the Slovenian Parliament

Bojan Evkoski[*†], **Senja Pollak**[*]

[*]Jozef Stefan Institute
[†] Jozef Stefan International Postgraduate School
bojan.evkoski@ijs.si
senja.pollak@ijs.si

Abstract

The work covers the development and explainability of machine learning models for predicting political leanings through parliamentary transcriptions. We concentrate on the Slovenian parliament and the heated debate on the European migrant crisis, with transcriptions from 2014 to 2020. We develop both classical machine learning and transformer language models to predict the left- or right-leaning of parliamentarians based on their given speeches on the topic of migrants. With both types of models showing great predictive success, we continue with explaining their decisions. Using explainability techniques, we identify keywords and phrases that have the strongest influence in predicting political leanings on the topic, with left-leaning parliamentarians using concepts such as *people* and *unity* and speak about *refugees*, and right-leaning parliamentarians using concepts such as *nationality* and focus more on *illegal migrants*. This research is an example that understanding the reasoning behind predictions can not just be beneficial for AI engineers to improve their models, but it can also be helpful as a tool in the qualitative analysis steps in interdisciplinary research.

1. Introduction

Artificial intelligence, in particular machine learning, is extensively used for solving many real-world problems both in research (Jordan and Mitchell, 2015) and industry (Shinde and Shah, 2018). In recent years, with AI being more present and the introduction of stricter guidelines (Interpol, 2019) and regulations (Sartor and Lagioia, 2020), it became apparent that the sheer power of machine learning models for predicting tasks does not justify their widespread potentially inconsiderate usage. Today, understanding and explaining the previously considered black-box models is equally important as developing and training them. This process enables engineers to detect biases in the model, come up with ideas for improvement, and most importantly examine the security of the system. Explainable AI (or simply XAI) is gradually becoming a must both in industry and research.

Another aspect of AI interpretability is its effect of bridging the gap between heavy quantitative black-box research and qualitative research more common among humanities and social sciences scholars. As machine learning engineers get feedback from a model on why it makes a particular prediction, humanities and social sciences scholars would take these signals to further examinations that can lead to more qualitative explanations of why certain features play a significant role in a particular problem.

This work covers a case of interdisciplinary research between computational linguistics (or natural language processing) and political science where we explain AI models to gain insight from a political linguistics perspective. We focus on parliamentary debates, a salient research topic in both humanities and social science disciplines, such as sociology, political science, sociolinguistics, and history (Skubic and Fišer, 2022). Our goal is to bring more insight into the speeches of parliamentarians (MPs) of different political leanings. First, by training machine learning models to predict if a parliamentarian is left or right-leaning based on their speech. And then, by using explainability techniques on the models, extract data and derive knowledge on what actually differentiates left and right political speeches in the parliament. To make things more politically relevant and methodologically clear, we are focusing on the concerning topic of migrants and the European migration crisis (Barlai et al., 2017) where the left and right have evident divergent stances: left-leaning parties showing consistent support to immigrants from Asia and Africa throughout Europe, and right-leaning parties showing moderate to strong opposition against immigration to Europe and their country in particular (van der Brug and Harteveld, 2021). We apply our analysis to the Slovenian Parliament from 2014 to 2020. For data, we use the open-access ParlaMint dataset (Erjavec et al., 2022) which provides complete parliamentary transcriptions for 17 countries, including Slovenia.

The paper is organized as follows. In Section 2., we briefly cover the related work on using explainable AI in social sciences, as well as recent applications of computational linguistics for parliamentary debates. In Section 3. we provide a description of the ParlaMint SI dataset we are working with, as well as the preprocessing steps required before training the models. In Section 4. we dive into the training of the classical machine learning and the modern language models. Finally, we discuss the results in Section 5. and conclude in Section 6..

2. Related Work

By the end of the 2010s and the beginning of 2020s, explainable AI has become a topic that involved both computer and social scientists in discussions on how to interpret and use the knowledge drawn from model explanations (Miller, 2019). Social scientists urge the importance of qualitative investigation experts joining in XAI

projects (Johs et al., 2022). Yet, social qualitative investigation done by non-experts is still very common. Combining qualitative and quantitative approaches (mixed research) has a variety of hardships (Brannen, 2017), and researchers from both sides of the spectrum rarely have insight into the benefits of combining knowledge with the opposite side. With this work focusing on a specific social theme, we show an example of how XAI research can be the initial step for a more thorough qualitative approach (Molina-Azorin, 2016).

Parliamentary discourse, although extensively researched in a qualitative setting (Ilie, 2015), is still underexplored in a quantitative manner. Naderi and Hirst (2015) use computational methods to analyze framing structures in speeches of the Canadian Parliament. Greene and Cross (2015) use dynamic topic modeling to explore the evolution of the political agenda in the European Parliament. Eskişar and Çöltekin (2022) analyze the emotion structure of speeches in the Turkish ParlaMint dataset.

Due to the European Migration crisis in 2015, researchers have been studying the political stances of politicians (Wallaschek, 2020), news media (Krotofil and Motak, 2018), and the effect of social media on the integration of migrants (Alencar, 2018). Computational techniques were also applied. For example, (Greussing and Boomgaarden, 2017) use techniques such as the bag-of-words and principal component analysis to understand media framing on the topic, while (Heidenreich et al., 2019) apply topic modeling to observe the dynamics of the migration narrative. In a more recent work by (Skubic et al., 2022), the authors create mention networks for multiple parliaments on the topic of migration and analyze the role of gender on speech influence.

3. Data

Our choice of parliamentary transcriptions is the richly annotated ParlaMint subset of the Slovenian Parliament. It covers all sessions of the National Assembly from August 2014 to July 2020, with more than 20M words. In order to prepare the data for training a political leaning model based on the transcriptions, we first applied a few preprocessing steps according to the metadata. First, we removed all guest and Chair speeches, leaving only regular MPs. Next, used the party name data for each parliamentarian to extract the main label (right or left) using Wikipedia's "political position" English metadata on the party. Far-right, right, and center-right are labeled as right; far-left, left and center-left are labeled as left. Finally, we applied speech selection according to the topic of our interest — migration. For this purpose, we prepared a set of keywords directly connected to the migration topic[1]. The speech selection is very straightforward. If any of our keywords appear in the transcription (observed as lemmas), we select the speech. Table 1 shows a general overview of the dataset magnitude, while Figure 1 shows a pie chart of speech share selections from the set of keywords. The final dataset turned out to be quite balanced between the two classes, with 1519 of the speeches classified as "left", and 1455 classified as right.

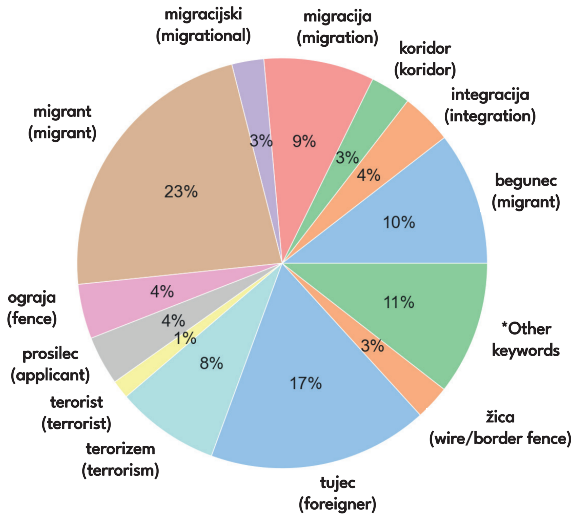


Figure 1: **Pie chart on the migration keywords.**

4. Models and Explainability

We followed two approaches to **training machine learning models** for text classification. The first is a classical bag of words feature extraction combined with a Linear SVM classifier (Liu et al., 2010). The second approach is using the state-of-the-art Transformer language models (Vaswani et al., 2017). We applied several variations of both approaches, evaluated using 5-fold cross-validation, and compared the prediction accuracy. Finally, we selected the best model for each approach and trained it on the full dataset in order to apply explainability analysis.

For the classical approach, we trained a Linear SVM on three different bag-of-words types: unigram; unigram and bigram; unigram, bigram, and trigram. As the common practice, we used only lemmas which were already available in SI ParlaMint. We used a well-refined stopwords list for the Slovenian language to remove noise from the bag of words[2], but we also used a minimum frequency of 5 appearances in the whole dataset, and a maximum frequency of a word appearing in 35% of the speeches which proved effective in removing the procedural parts of speeches and improving the results. We optimized the model on each of the training folds in a nested 5-fold cross-validation and we evaluated the optimized model on the main testing folds, practically optimizing and evaluating five models of the same kind, reporting the average and its 95% confidence interval. Finally, using the hyperparameters of the best-performing model, we trained one final model which we used for the explainability experiments.

Opposed to the common understanding of machine learning models, a Linear SVM classifier is not a black box that cannot be explained in its calculations. It creates a hyperplane that uses support vectors to maximize the distance between the two predicted classes (in our case "left"

[1]The list of migration keywords was prepared in collaboration with social scientists Andreja Vezovnik and Veronika Bajt. It contains 95 lemmas with all their corresponding word forms prepared by Anka Supej.

[2]https://github.com/stopwords-iso/stopwords-sl

Dataset	#Parties	#Speakers (Left/Right)	#Speeches (Left/Right)	AWS	MSS
SI ParlaMint	12	353 (114/48)	75122 (70.6%/20.4%)	1244.1	30.0
SI ParlaMint (MPs)	12	166 (114/48)	31151 (55.9%/40.9%)	1210.5	125.5
SI Parlamint (MPs on migration)	12	153 (105/48)	2974 (51.1%/48.9%)	854.7	12.0

Table 1: **General statistics of the Slovenian ParlaMint dataset and two of its subsets.** AWS - Average words per speech. MSS - Median speeches per speaker.

and "right"). The weights that represent the Linear SVM equation hyperplane are the vector coordinates that are orthogonal to the hyperplane. So, their direction indicates the predicted class. The absolute size of the coefficients in relation to each other can then be used to determine feature importance for the data separation task. For our task, the minimum (most negative) weight coefficients correspond to words in the bag-of-words representation that led the model to classify a speech as a "leftist", while the maximum (most positive) weight coefficients correspond to words that led it to classify a speech as "right". Using this explainability logic, we derived an understanding of what makes the difference between a leftist and a rightist ideology's thought and word choice.

For language models, we used two types: the multilingual BERT and the SloBERTa. The multilingual BERT (Devlin et al., 2018) was trained in 104 languages and is one of the first successful language models which serves as an appropriate baseline for fine-tuning the model to any kind of textual task. SloBERTa (Ulčar and Robnik-Šikonja, 2021) uses the BERT architecture and it is monolingual, meaning it is trained purely on Slovenian data, making it more suitable for tasks where the model is applied specifically to Slovenian text, as is ours. Both pretrained models were fine-tuned on the raw SI ParlaMint transcriptions using 30 epochs. The Learning rate of the BERT model was 5e-6, for the SloBERTa was 6e-6. As in the SVM case, we optimized the number of epochs and the learning rate using 5-fold nested cross-validation. For preprocessing, we used the standard respective tokenizers of BERT and SloBERTa. As both models are limited to 512 tokens, we applied an automatic truncation, using the first 512 words in each speech for training and predicting. Both models are limited to a maximum of 512 tokens, and speeches above that limit were truncated.

For **explaining the deep learning models,** we used a technique introduced Lundberg and Lee (2017) called by SHAP. It is a powerful technique that uses classic equations from cooperative game theory to compute explanations of model predictions. Shapley values are feature importance values derived when a model is trained on all feature subset combinations, mathematically calculating the importance of each. Calculating Shapley values in this manner is computationally expensive, so what SHAP manages to do is derive these values by sampling approximations. Reading the SHAP results is very similar to reading the explainability analysis of the Linear SVM: more negative values refer to words leading the model to classify a speech as "leftist" and more positive values refer to the "rightist" words. There are two ways of presenting the token importance using Shapley values: its total Shapley in the entire dataset or its maximum/minimum recorded value. The issue with the first technique is that it prioritizes common words such as stopwords and should be generally avoided. We use the second technique, as it leads to more sensible results. A third variation of normalizing the Shapley sum according to the number of occurrences is also an option, yet we leave out this experimentation and plan it as our further work.

Model	Accuracy
Random Baseline	0.511
Tf-Idf + Lin. SVM (1-grams)	0.866 ± 0.01
Tf-Idf + Lin. SVM (1, 2-grams)	0.903 ± 0.01
Tf-Idf + Lin. SVM (1, 2, 3-grams)	$\mathbf{0.913 \pm 0.02}$
Multilingual BERT	0.819 ± 0.01
SloBERTa	$\mathbf{0.877 \pm 0.03}$

Table 2: **Classification scores on predicting political leanings based on speech transcriptions.**

5. Results and Discussion

Table 2 shows the results of classifying speeches on the topic of migration as "leftist" or "rightist", depending on the political affiliation of the speaker in the Slovenian Parliament. The classical Tf-idf approach using unigrams, bigrams and trigrams showed the best results, with around 91% accuracy. From the language models, the SloBERTa outperformed BERT yet with only 87% accuracy. One reason behind the worse performance of the language models could be its inability to operate on very long sequences, as more than 60% of the speeches in our dataset contain more than 512 tokens (maximum sequence for BERT).

The models showed that predicting parliamentary political leaning from speech transcriptions is possible. Since statistical models can differentiate "leftist" and "rightist" speeches, we investigated if their explanation through feature importance makes sense from a political linguistics perspective. Figures 2 and 3 show the results of the feature importance analysis. Here, we can notice some obvious similarities and differences between the classical and the language model approach. The first difference is that the SVM feature importance can catch not only tokens (words) but also bigrams and trigrams, which can be useful for a better linguistic context. Since it uses lowercase lemmas, it mitigates duplicates in different word forms, yet it can potentially lead to ambiguities. The most important difference is that the SVM feature importances are the general values for the whole dataset, while the Shapley values are detected maximums for a particular comment, which does not mean that these words have the same negative (leftist) or positive (rightist) effect for all samples.

Both model interpretations show that right-wing parliamentarians' motif is to mention the country name (Slovenia) and its forms. The SVM interpretation also reveals that the rightists emphasize their party names. Regarding migration, their motif is to focus on the illegal aspect of the migration, with the SVM models catching the

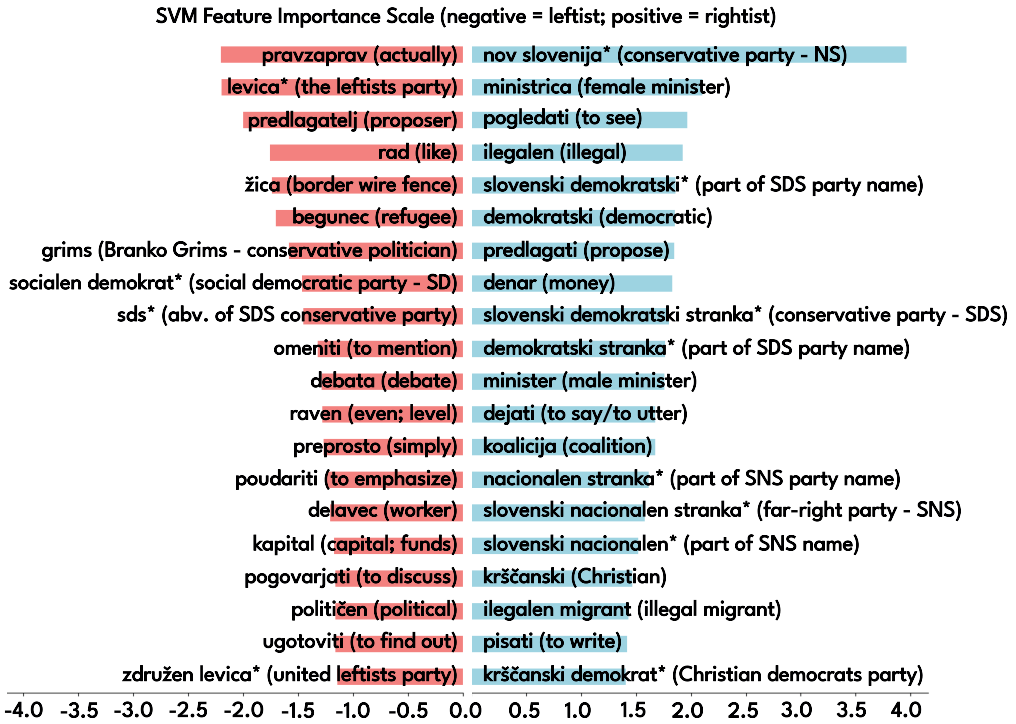


Figure 2: **SVM feature importance.** The left (red) side contains phrases that have the highest significance for the model predicting a speech as "leftist". The same explanation is for the right (blue) side for the "rightist" speeches. Phrases marked with $\star$ refer to party names.

SloBERTa Shapley Values (negative = leftist; positive = rightist)

-25.40	-------	(presiding) predsedujoči	Novi (New)	-------	+30.00
-25.25	-------	(we work) delamo	Nove (New plu.)	-------	+29.03
-24.27	-------	(Thank you) Hvala	Slovenske (Slovenian)	-------	+27.99
-24.07	-------	(United) Združeni	Slovenska (Slovenian)	-------	+24.93
-21.75	-------	(jokes) vici	Slovenski (Slovenian)	-------	+24.92
-20.89	-------	(part-of-word token) kevi	premalo (too little)	-------	+22.95
-18.71	-------	(peak) vrhunec	Slovenije (Slovenia)	-------	+21.72
-17.89	-------	(chosen f.) izbrala	časi (the times)	-------	+20.09
-17.63	-------	(touched) dotaknil	gospe (ladies)	-------	+18.58
-17.53	-------	(discuss) razpravi	kolega (colleague)	-------	+17.48
-17.41	-------	(bet) stavili	Kr (part-of-word token)	-------	+16.65
-17.15	-------	(you go out) izhodiš	Spoštovani (Dear)	-------	+16.62
-16.85	-------	(task) naloga	naj (most)	-------	+16.05
-16.74	-------	(consistent) skladna	Sloveniji (Slovenija)	-------	+14.93
-16.38	-------	(a word) besedo	poglejte (look)	-------	+14.65
-15.55	-------	(or) ali	v (in)	-------	+14.54
-15.22	-------	(not) ni	žal (unfortunately)	-------	+14.13
-14.97	-------	(jokes) vice	vaše (yours)	-------	+13.99

Figure 3: **Maximum and minimum Shapley values for the SloBERTa model.** Left/right (red/blue) side shows the words that had the highest signal in predicting a "leftist/rightist" speech.

phrase "illegal migrant" as something that almost exclusively right-wing politicians use. The leftists tend to use the word "begunec" (angl. refugee) instead of "migrant", as the meaning behind the former refers to someone who is fleeing. They tend to use words such as "united" and "debate" and they often times refer to their party opponents by the party abbreviations instead of the full name. One interesting case is that the SVM manages to recognize the leftists emphasizing the word "grims". Grims is the surname of the right-wing politician Branko Grims who is strongly against immigrants in Slovenia and Europe and was the most active politician in the parliament on the topic.

Although much more can be drawn from the feature importance, this is where our analysis stops. The model

interpretability through the list of words that had the most meaningful impact on the classification opens up a great possibility for further qualitative work. Researchers studying political linguistics could observe the usages of these lists and find patterns in the connotation of the words used. They could confirm or debunk ideological concepts used by the different sides of the political spectrum. Or, they could analyze if these words and phrases actually contain the primary message of a sentence or if their role is more on the stylistic side of political speeches.

6. Conclusion

Explaining the decision-making of machine learning models (known as XAI) can be a great tool in interdisciplinary research. When the goal is not just to classify, but to understand patterns that appear across classification groups, XAI can complement qualitative research.

In this work, we used XAI to bridge the gap between computational and political linguistics. We developed classical machine learning as well as deep learning language models that can classify parliamentary speeches as "leftist" and "rightist" for the topic of migration. We applied our approach to the Slovenian parliament using the ParlaMint dataset with data from 2014 to 2020. With both approaches showing great predictive success, we applied methods (such as calculating the Shapley values) that can explain the decisions of our models and show which words and phrases differentiate "leftist" from "rightist" speeches and vice versa. While left-leaning parliamentarians use concepts such as "unity" and "debate", the right-leaning parliamentarians put more emphasis on the national symbols (mentioning Slovenia) and their party names.

We leave an opening for further work in multiple directions. One is the improvement of the interpretability methods for models that work with text. Different feature importance techniques could be applied and a comparative study could help us understand the strengths and weaknesses of the methods. Another direction is exploring the ways the model explanations can be used in interdisciplinary research. How to use the words that give the most value to models' predictions in a qualitative continuation of the work. Or, how to understand and communicate model flaws with experts from the social sciences in order to improve them.

Acknowledgements The work was supported by the Slovenian Research Agency research projects and programmes P6-0436: Digital Humanities: resources, tools and methods (2022–2027), P6-0280: Economic, Social and Environmental History of Slovenia (2022–2027), P2-0103: Knowledge Technologies (2022–2027), J6-2581: Computer-assisted multilingual news discourse analysis with contextual embeddings (2020-2023) and J5-3102: Hate speech in contemporary conceptualizations of nationalism, racism, gender and migration (2021-2024), as well as the DARIAH-SI research infrastructure (2022-2027). We also acknowledge the financial support from the RobaCOFI project, which has indirectly received funding from the European Union's Horizon 2020 research and innovation action programme via the AI4Media Open Call #1 issued and executed under the AI4Media project (Grant Agreement no. 951911). We also thank Andreja Vezovnik and Veronika Bajt for their help in preparing the migration keywords.

Code The source code is publicly available at: `github.com/boevkoski/xaicl_parliaments`

References

Alencar, Amanda, 2018. Refugee integration and social media: A local and experiential perspective. *Information, Communication & Society*, 21(11):1588–1603.

Barlai, Melani, Birte Fähnrich, Christina Griessler, and Markus Rhomberg, 2017. *The migrant crisis: European perspectives and national discourses*, volume 13. LIT Verlag Münster.

Brannen, Julia, 2017. Combining qualitative and quantitative approaches: an overview. *Mixing methods: Qualitative and quantitative research*:3–37.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Erjavec, Tomaž, Maciej Ogrodniczuk, Petya Osenova, Nikola Ljubešić, Kiril Simov, Andrej Pančur, Michał Rudolf, Matyáš Kopp, Starkaur Barkarson, Steinór Steingrímsson, et al., 2022. The parlamint corpora of parliamentary proceedings. *Language resources and evaluation*:1–34.

Eskişar, Gül M Kurtoğlu and Çağrı Çöltekin, 2022. Emotions running high? a synopsis of the state of turkish politics through the parlamint corpus. In *Proceedings of the Workshop ParlaCLARIN III within the 13th Language Resources and Evaluation Conference*.

Greene, Derek and James P Cross, 2015. Unveiling the political agenda of the european parliament plenary: A topical analysis. In *Proceedings of the ACM web science conference*.

Greussing, Esther and Hajo G Boomgaarden, 2017. Shifting the refugee narrative? an automated frame analysis of europe's 2015 refugee crisis. *Journal of ethnic and migration studies*, 43(11):1749–1774.

Heidenreich, Tobias, Fabienne Lind, Jakob-Moritz Eberl, and Hajo G Boomgaarden, 2019. Media framing dynamics of the 'european refugee crisis': A comparative topic modelling approach. *Journal of Refugee Studies*, 32(Special_Issue_1):i172–i182.

Ilie, Cornelia, 2015. Parliamentary discourse. *The International Encyclopedia of language and social interaction*:1–15.

Interpol, Unicri, 2019. Artificial intelligence and robotics for law enforcement. *Interpol/Unicri, Lyon/Turin*.

Johs, Adam J, Denise E Agosto, and Rosina O Weber, 2022. Explainable artificial intelligence and social science: Further insights for qualitative investigation. *Applied AI Letters*, 3(1):e64.

Jordan, Michael I and Tom M Mitchell, 2015. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260.

Krotofil, Joanna and Dominika Motak, 2018. Between traditionalism, fundamentalism, and populism: a criti-

cal discourse analysis of the media coverage of the migration crisis in poland. In *Religion in the European refugee crisis*. Springer, pages 61–85.

Liu, Zhijie, Xueqiang Lv, Kun Liu, and Shuicai Shi, 2010. Study on svm compared with the other text classification methods. In *2010 Second international workshop on education technology and computer science*, volume 1. IEEE.

Lundberg, Scott M and Su-In Lee, 2017. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., pages 4765–4774.

Miller, Tim, 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38.

Molina-Azorin, José F, 2016. Mixed methods research: An opportunity to improve our studies and our research skills.

Naderi, Nona and Graeme Hirst, 2015. Argumentation mining in parliamentary discourse. In *Principles and practice of multi-agent systems*. Springer, pages 16–25.

Sartor, Giovanni and Francesca Lagioia, 2020. The impact of the general data protection regulation (gdpr) on artificial intelligence. *European Parliamentary Research Service*.

Shinde, Pramila P and Seema Shah, 2018. A review of machine learning and deep learning applications. In *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*. IEEE.

Skubic, Jure, Jan Angermeier, Alexandra Bruncrona, Bojan Evkoski, and Larissa Leiminger, 2022. Networks of power: Gender analysis in selected european parliaments. In *2nd Workshop on Computational Linguistics for Political Text Analysis (CPSS)*.

Skubic, Jure and Darja Fišer, 2022. Parliamentary discourse research in sociology: Literature review. In *Proceedings of the Workshop ParlaCLARIN III within the 13th Language Resources and Evaluation Conference*.

Ulčar, Matej and Marko Robnik-Šikonja, 2021. Sloberta: Slovene monolingual large pretrained masked language model.

van der Brug, Wouter and Eelco Harteveld, 2021. The conditional effects of the refugee crisis on immigration attitudes and nationalism. *European Union Politics*, 22(2):227–247.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wallaschek, Stefan, 2020. Contested solidarity in the euro crisis and europe's migration crisis: A discourse network analysis. *Journal of European Public Policy*, 27(7):1034–1053.

Empirical Analysis of Oral and Nasal Vowels of Konkani

Swapnil Fadte[1], Edna Vaz[2], Atul Kr. Ojha[3], Ramdas Karmali[1], Jyoti D. Pawar[1]

[1] Discipline of Computer Science & Technology, Goa Business School, Goa University
swapnil.fadte@unigoa.ac.in, rnk@unigoa.ac.in, jdp@unigoa.ac.in
[2]Govt. College of Arts Science and Commerce Quepem, Goa. And Dept. of Linguistics, University of Mumbai
edna.vaz22@gmail.com
[3]Insight SFI Research Centre for Data Analytics, Data Science Institute, University of Galway
atulkumar.ojha@insight-centre.org

Abstract

Konkani is a highly nasalised language which makes it unique among Indo-Aryan languages. This work investigates the acoustic-phonetic properties of Konkani oral and nasal vowels. For this study, speech samples from six speakers (3 male and 3 female) were collected. A total of 74 unique sentences were used as a part of the recording script, 37 each for oral and nasal vowels, respectively. The final data set consisted of 1135 vowel phonemes. A comparative F1-F2 plot of Konkani oral and nasal vowels is presented with an experimental result and formant analysis. The average F1, F2 and F3 values are also reported for the first time through experimentation for all nasal and oral vowels. This study can be helpful for the linguistic research on vowels and speech synthesis systems specific to the Konkani language.

Keywords: Konkani, oral vowel, nasal vowel, formant analysis

1. Introduction

Konkani belongs to the Indo-Aryan branch of the Indo-European family of languages. It is a member of the southern group of Indo-Aryan languages, and is most closely related to Marathi within this group (Miranda, 2018)). Konkani is mainly spoken in Goa and in some parts of the neighbouring states of Maharashtra, Karnataka, and Kerala, where Konkani speakers migrated after the Portuguese arrival in Goa. The 1991 Census of India records the number of Konkani speakers to be 1,760,607 out of which 602,626 (34.2 %) were from Goa, 312,618 (17.8 %) were from Maharashtra, 706,397 (40.1 %) from Karnataka, and 64,008 (3.6 %) from Kerala (Miranda, 2018). Konkani is written in different scripts in the regions where it is spoken.

1.1. Phonological Features of Konkani

As regards the phonology of Konkani, different scholars have mentioned different numbers of vowels and consonants in the language. Also, there is no consensus on the exact specification of the vowels as regards their place in the vocal tract. Nasalisation is phonemic in Konkani (as shown by i and ii below).

i hɛ tat͡ʃɛ b^hurgɛ
these.mas.pl. he.gen.mas.pl. children.mas.pl.
'These are his (male) children.'

ii hɛ̃ tat͡ʃɛ̃ b^hurgɛ̃
this.neut.sg. he.gen.neut.sg. child.neut.sg.
'This is his child.'

1.2. Related Work on Konkani

(Sardesai, 1986), (Sardesai, 1993), in her dialect-specific work refers to nine Konkani Vowel phonemes: Front [i, e, ɛ]; Central [ɨ ə, a]; and Back [u, o, ɔ]. The author mentions that all these vowel phonemes can be nasalised. Her work is summarised in Table 1.

(Almeida, 1989) makes a reference to eight oral vowels for Konkani: Front [i, e, ɛ] and Back [u, o, ɔ, θ, a]. His classification of oral vowels is provided in Table 1. The author mentions that all vowels present in the language can be nasalised. Examples for both oral and nasal vowels are presented by him in his work. The author seems to consider vowel length to be phonemic in Konkani, which is not the case, at least for the Konkani varieties spoken in Goa.(Miranda, 2018) mentions nine Vowel phonemes for Konkani, along with their corresponding nasal counterparts. These are: [i, e, ɛ, ə, ʌ, a, u, o, ɔ].
(Fadte et al., 2022) provide a vowel chart for Konkani based on their acoustic analysis of vowels. They also provide the properties of vowel pairs which have different phonetic realisations but the same written representation in the script. Their vowel classification work is presented in Table 2, which includes equivalent vowels in different scripts, namely Devanagari, Roman and Kannada. Their work also acknowledges that all oral vowels could be nasalised.

Author and Year	Vowels	Classification
(Sardesai, 1986)	i, e, ɛ, u, o, ɔ, a, ə, ɨ	Dialect-specific
(Almeida, 1989)	i, e, ɛ, u, o, ɔ, a, θ	General
(Miranda, 2018)	i, e, ɛ, u, o, ɔ, a, ə ʌ	General
(Fadte et al., 2022)	i, e, ɛ, u, o, ɔ, a, ə, ɨ	General

Table 1: Comparison of Konkani Vowel classifications

1.3. Related Work on Other Languages

(Shosted et al., 2012) have presented work on Hindi nasal vowels, where F1-F2 values are used for calculating the

Approximate IPA Notation	Equivalent grapheme			Examples	English meaning	Vowel type
	Roman	Devanagari	Kannada			
i	i	इ and ई	ಇ and ಈ	[v i: s]	twenty	Front
				[k ə v i:]	poet	
e	e	ए	ಎ	[p e: r]	guava tree	
				[k^h e: ⌊]	game	
				[m e: dʒ]	count (imperative.2 p.sg.)	
ɛ		ऍ	ಏ	[p ɛ: r]	guava fruit	
				[k^h ɛ: ⌊]	play (imperative.2 p.sg.)	
				[m ɛ: dʒ]	table	
ɨ	a	अ	ಅ	[k ɨ :r]	do (imperative.2 p.sg.)	Central
				[b ɨ s]	sit (imperative.2 p.sg.)	
				[p ɨ ɖ]	a traditional measure	
ə				[k ə r]	tax	
				[b ə s]	bus	
				[p ə ɖ]	fall (imperative.2 p.sg.)	
a		आ	ಆ	[a ɖ s ə r]	tender coconut	
				[r a dʒ a]	king	
u	u	उ and ऊ	ಉ and ಊ	[u:s]	sugarcane	Back
				[p u: l]	bridge	
o	o	ओ	ಒ	[ts o: r]	thief	
				[d o: n]	two	
ɔ		ऑ	ಓ	[ts ɔ:r]	thieves	
				[b ɔ: l]	ball	

(Fadte et al., 2022)

Table 2: Oral vowels of Konkani

positions of the tongue in case of the nasal vowels. They have showed that the position of the tongue is generally lowered for Back vowels, fronted for Low vowels, and raised for Front vowels.

(Feng and Castelli, 1996) have presented work on the nasalisation of 11 French vowels. They show that the first two resonance frequencies are at about 300 and 1000 Hz.

(Carignan, 2014) is an acoustic study of three French oral-nasal vowel pairs /a//ã/, /ɛ/-/ɛ̃/, and /o/-/õ/. His study shows that the oral articulation of French nasal vowels is not arbitrary.

1.4. Objectives of the Present Study

As mentioned earlier, there are differences among Konkani scholars on the exact number of Vowel phonemes in the language. In the absence of more accurate descriptions of the Vowel system of the language and nasalisation of vowels, this study makes an attempt to target an important aspect of Konkani vowel phonemes namely, their nasalisation using acoustic analysis. For this study, we have taken into consideration the nine vowel phonemes mentioned in the Standard variety of the language which are the same as the ones mentioned in (Sardesai, 1986) and later cited in (Fadte et al., 2022).

This work is arranged into four sections. Section 1. - the Introduction section above highlights the linguistic features of the language and states the objective of the study. Section 2. discusses the methodology followed in the experiment. The results of the experiment are presented in section 3.. Section 4. concludes the paper with the scope for further studies.

1.5. Hypothesis

Given that nasalisation is phonemic in the language, each vowel phoneme of the language will have a nasal counterpart. In other words, the status of nasalisation as being phonemic in the language will become more explicit through this study.

2. Methodology

This section presents the details of the experimental work carried out and the methodology that was used for the experiment. (Fadte et al., 2022)'s methodology was followed for carrying out this experiment.

2.1. Recording Script

The recording script of this work was based on the Vowel phonemes mentioned in the classification provided by (Fadte et al., 2022). The Vowel phonemes in the script were

arranged according to their classification which was established using the minimal and near-minimal pairs. A Phoneme is the smallest distinctive/contrastive unit in the sound system of a language. It is that unit of sound (a phone) that can distinguish one word from another in a particular language. The inventory of phonemes of a language is created using the minimal pairs (or near-minimal pairs in the absence of minimal pairs) of the language. Minimal pairs are pairs of words or phrases in a particular language that differ in only one phonological element and have distinct meanings. Near minimal pairs are pairs which have one or more additional differences elsewhere in the word besides the crucial position. Thus minimal pairs are an important tool that helps in establishing phonemes of a particular language. Pronunciation of phones is shown using square brackets whereas phonemes established using the minimal pairs are written in between slashes. To give an example, the Konkani words [na:k] 'nose' and [na:g] 'king cobra' differ only in the sounds [k] and [g]. Thus, the phones [k] and [g] in these words produce a difference in meaning. Using the [na:k] and [na:g] (minimal) pair we can now establish that the consonants [k] and [g] are phonemes of the language. These phonemes will therefore be written as /k/ and /g/. The recording script was created with Konkani sentences consisting of minimal pairs that aimed at establishing the vowel phonemes. A few examples of minimal pairs targeting vowel phonemes used in the recording script are provided in Table 3, and the entire script can be accessed from **here**.[1] The recording script consisted of 74 unique sentences, 37 for oral and 37 for nasal vowels, respectively. At least two different sets of minimal pairs were used for each vowel phoneme.

Oral Vowel				Nasal Vowel			
Konkani Examples	Vowel	IPA	Gloss	Konkani Examples	vowel	IPA	Gloss
शी	i	[ʃi:]	ugh' excl.	शीं	ĩ	[ʃĩ:]	cold' n/f.sg.
केस	e	[ke:s]	suit' f.sg.	केंस	ẽ	[kẽ:s]	a hair' m.sg.
वेत	ɛ	[vɛt]	cane' n.sg.	वेंत	ɛ̃	[vɛt̃]	span' n.sg.
वय	ə	[və:j]	'age'.neut.sg.	वंय	ə̃	[və̃j]	'fence'.f.sg.
बाय	a	[ba:j]	'girl'.neut.sg.	बांय	ã	[bã:j]	'well'.f.sg.
खूट	u	[kʰu:t]	shortage' f.sg.	खूंट	ũ	[kʰũ:t]	stake' m.sg.
घोगो	ɔ	[gʰɔgɔ]	fall' m.sg.	घोंगो	ɔ̃	[gʰɔg̃ɔ]	horn' m.sg.

Table 3: Example of oral and nasal vowels in Konkani

2.2. Speakers' Detail

Three male and three female native speakers of Konkani were selected for this experiment. Speakers belonged to different geographical locations and spoke diverse regional dialects (details of these can be accessed from **here**[1]).This ensured that phone variability across regions was captured. All the speakers selected for the recording were literate.

2.3. Data Elicitation and Recording

The reading material consisting of sentences having minimal pairs was provided to the speakers as a printed copy. The speakers were given some time to familiarise themselves with the meaning of the sentences. Then, they were instructed to read the sentences in the most natural way they could. Each sentence in the recording script was pronounced thrice by the speakers. This was done in order to capture any dialectal variation in the pronunciation of the target phonemes. It also helped to detect speaker-specific errors in phone production. The recording was performed in a closed room with less ambient noise. The audio was recorded using a Zoom-H6 recorder at a sampling rate of 48 kHz and was stored in non-lousy WAV format.

2.4. Annotation

Phoneme-level annotation of audio data was then carried out. Only the vowel phonemes which were to be used for analysis were annotated. A total of 1135 vowel phones were annotated in the data set. The frequency distribution of phones is presented in Figure 1.

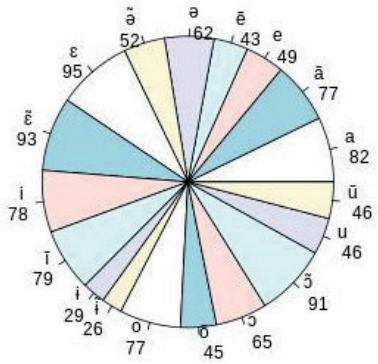


Figure 1: Frequency distribution of phones in dataset

Annotation was performed using the Praat software (Paul and Weenink, 1992). The start and end of a phone boundary was marked as perceived by the ear of the annotator and with the help of a spectrogram in the Praat tool. A sample of an audio signal, spectrogram and phoneme level annotation done in the Praat tool is shown in Figure 2.

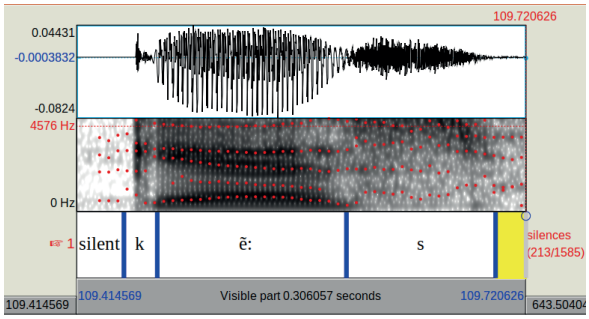


Figure 2: Annotations and spectrogram of phones

2.5. Formant Extraction

A Praat script was used to perform formant extraction on annotated data. This script extracts the formant details from the mid-temporal interval of the phoneme, which is stored

[1] https://github.com/shashwatup9k/dhvani-konkani

in a text file. For formant extraction, values for speakers' frequency were set to standard values, i.e. 5 kHz for male and 5.5 kHz for female speakers. The data was stored in a text file and later converted to a CSV file for plotting results and analysis.

2.6. Data Verification

After formant details were extracted, they were plotted using a box plot for verifying visually any outliers that may have occurred due to wrong annotation. This step helped in identifying certain incorrect annotations, which were then corrected. As discussed in section 2.5., formant extraction was performed again, and boxplots were replotted. Box plots of the F1 and F2 formant for male speakers are shown in Figure 3. After the above corrections, a few outliers can still be seen.

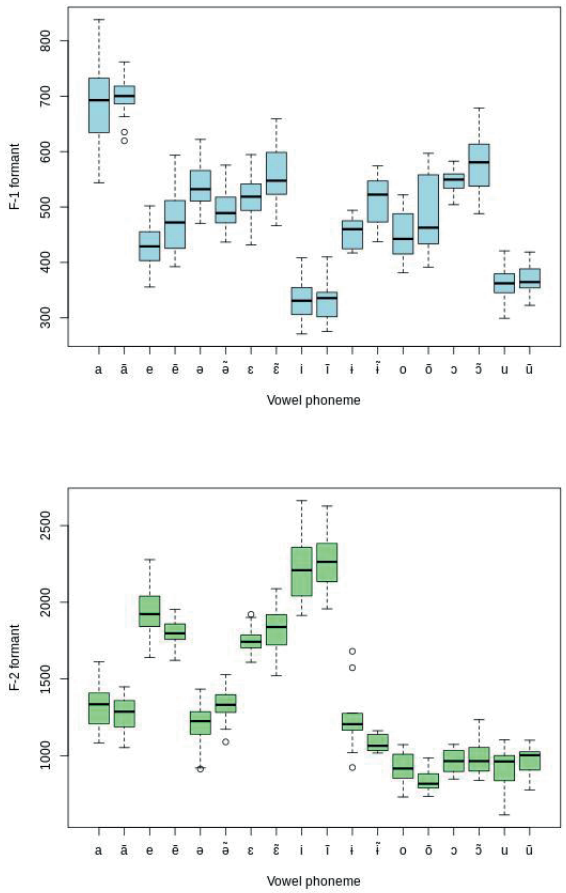


Figure 3: Box plot of F1 and F2 formants for oral and nasal vowels for male speakers

Final data verification was done with the help of a linguist who simultaneously listened to the phones and viewed the spectrogram to verify the label given to them. Errors in the phoneme production were noted down and are discussed in detail in section 2.7. below.

2.7. Substitution Analysis

The verified data showed some deviations from the expected production of the phonemes. Table 4 presents the phoneme substitution that occurred during the elicitation process. All these deviations were not used for the formant analysis. The close-mid central vowel [θ] with a frequency of 12 (see Table 4) occurring in place of the schwa ([ə]), is its allophone which occurs in the environment wherein it is followed by the open-mid back (rounded) vowel [ɔ] as in the words [gəvɔ] 'bison' n.mas.sg, [bərɔ] 'good' adj.mas.sg.

Phoneme	Substitution	Frequency	% of total substitution
i	None	0	0.0
ĩ	i	28	12.4
e	a	1	0.4
	ɛ	3	1.3
ẽ	e	6	2.7
	ɛ	12	5.3
	ɛ̃	4	1.8
ɛ	e	6	2.7
ɛ̃	ɛ	20	8.9
ɨ	ə	6	2.7
	ə̃	3	1.3
ɨ̃	ə	7	3.1
	ə̃	10	4.4
	θ̃	1	0.4
ə	θ	12	5.3
	ɨ	4	1.8
ə̃	ɨ	8	3.6
	ə	6	2.7
	ã	1	0.4
	o	1	0.4
a	ã	8	3.6
ã	a	12	5.3
u	ũ	3	1.3
ũ	u	19	8.4
o	ɔ	12	0.9
o	o	8	3.1
	ɔ̃	2	0.9
ɔ	ɔ̃	1	0.4
ɔ̃	ɔ	32	14.2

Table 4: Substitution Analysis

3. Experimental Results and Analysis

A formant analysis of Vowel phonemes was performed as part of this study. R script was written with the use of the *phonR* package(McCloy, 2016) to plot the experimental results.

F1-F2 plots for oral and nasal vowels of male speakers are presented in Figure 4. A well-defined grouping of vowels in formant space is observed. From Figure 4, it is clearly seen that the Front oral vowels /i/, /e/ and /ɛ/ occupy non-intersecting space in the formant chart. In the same figure, we can see that the three extreme vowels (/i/, /a/, and /u/) occupy three corners in formant space. Other vowels also do

not have many intersections in formant space. F1-F2 plots

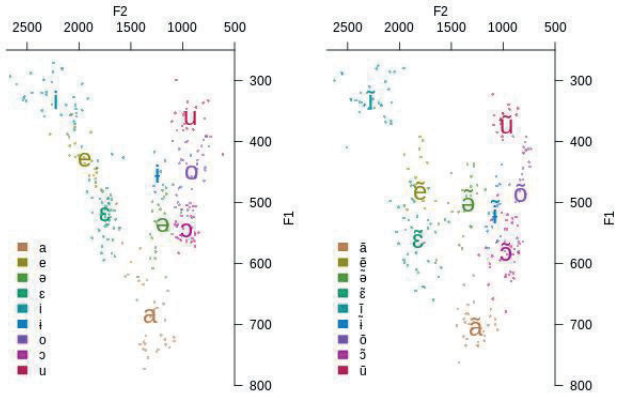


Figure 4: Formant chart for oral and nasal vowels for male speakers

for oral and nasal vowels of female speakers are presented in Figure 5, which have similar features as seen in the male formant chart. A comparative chart for additional details can be accessed **here**[1]. Apart from the formant charts, we have

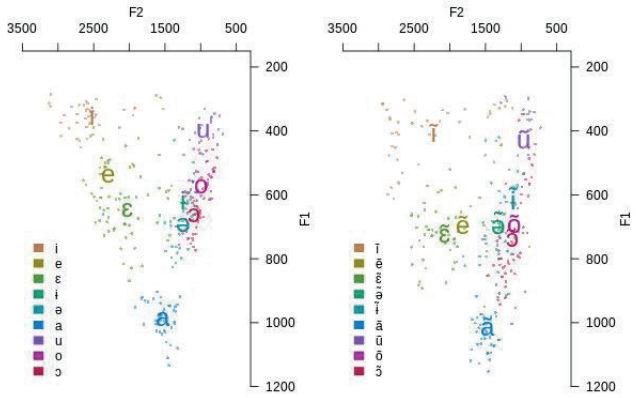


Figure 5: Formant chart for oral and nasal vowels for female speakers

also listed the average values of F1, F2, and F3 formants for male and female speakers. These are presented in Table 5. The average values for the F1 and F2 for oral vowels are similar to those reported by (Fadte et al., 2022). Since no previous work was reported for the nasal values, formant values provided in Table 5 may be considered as the first reporting of such work.

4. Conclusion and Future Work

This work provides a comparative study of the Konkani Vowel phonemes (i.e. oral and nasal vowels). The results have shown that all oral vowel sounds in Konkani can be nasalised. It is observed that the different vowels in the formant chart are in their expected position as per (Fadte et al., 2022) vowel classification. It is also seen that nasalisation changes the F1-F2 values for the vowel phonemes.

	female			male		
Phoneme	F1	F2	F3	F1	F2	F3
i	353	2518	3055	331	2229	2730
e	535	2240	2773	428	1943	2470
ɛ	641	2518	3055	331	1749	2450
ɨ	636	1339	2978	453	1245	2548
ə	690	1224	3081	537	1193	2508
a	982	2518	3055	685	1319	2446
u	417	972	2904	362	930	2464
o	574	990	3072	450	914	2585
ɔ	670	1089	2964	544	966	2452
ĩ	393	2204	3049	328	2272	2747
ẽ	673	1861	2699	477	1800	2512
ɛ̃	708	2057	2724	556	1818	2434
ɨ̃	630	1150	2890	514	1082	2515
ə̃	688	1326	2914	496	1341	2526
ã	974	1461	2737	699	1271	2413
ũ	402	999	2749	368	969	2442
õ	691	1103	2941	480	838	2644
ɔ̃	726	1138	2985	576	977	2497

Table 5: Average F1, F2, and F3 values for Vowel phonemes.

The average F1, F2, and F3 values for nasal vowels are reported for the first time through experimentation. This work can be helpful for the linguistic study of vowels and speech synthesis systems specific to Konkani language. Although oral and nasal studies have been presented, other phones and combinations of phones, like consonants, diphthongs have not been explored in this work or rather there have not been acoustic studies done related to the properties of such phones in Konkani language. We wish to explore these in our future work.

Acknowledgements

Atul Kr. Ojha would like to acknowledge the support of the Science Foundation Ireland (SFI) as part of Grant Number SFI/12/RC/2289_P2, Insight SFI Centre for Data Analytics.

References

Almeida, Matthew SJ, 1989. *A Description of Konkani*. Miramar, Panaji: Thomas Stephens Konknni Kendr.

Carignan, Christopher, 2014. An acoustic and articulatory examination of the oral in nasal: The oral articulations of french nasal vowels are not arbitrary. *Journal of phonetics*, 46:23–33.

Fadte, Swapnil, Edna Vaz Fernandes, Ramdas Karmali, and Jyoti D. Pawar, 2022. Acoustic Analysis of Vowels in Konkani. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 21(5):1–13.

Feng, Gang and Eric Castelli, 1996. Some acoustic features of nasal and nasalized vowels: A target for vowel nasalization. *The Journal of the Acoustical Society of America*, 99(6):3694–3706.

McCloy, D. R., 2016. Normalizing and plotting vowels with phonR 1.0.7. University of Washington, USA.

Miranda, Rocky V., 2018. The Languages of Goa,. In G. N. Devy, Madhavi Sardesai, and Damodar Mauzo (eds.), *People's Linguistic Survey of India, The Languages of Goa,*,

chapter The Konkani Language. Delhi: Orient Black-Swan, volume 8, part 2 edition, page 20.

Paul, Boersma and David Weenink, 1992. Praat: doing phonetics by computer [computer program].

Sardesai, Madhavi, 1986. Some aspects of konkani grammar. *Department of Linguistics, Deccan College, Pune*.

Sardesai, Madhavi, 1993. *Bhasabhas Article on Linguistics*. Goa: Goa Konkani Academy, 1st edition.

Shosted, Ryan, Christopher Carignan, and Panying Rong, 2012. Managing the distinctiveness of phonemic nasal vowels: Articulatory evidence from hindi. *The Journal of the Acoustical Society of America*, 131(1):455–465.

Investigating parallelograms: Assessing several word embedding spaces against various analogy test sets in several languages using approximation

Rashel Fam Yves Lepage

Graduate School of Information, Production, and Systems, Waseda University
2-7 Hibikino, Wakamatsu-ku, Kitakyushu-shi, 808-0135 Fukuoka-ken, Japan
fam.rashel@fuji.waseda.jp, yves.lepage@waseda.jp

Abstract

The famous example *man is to woman as king is to queen* was claimed to demonstrate the effectiveness of word embedding spaces. The idea of using analogy to assess the quality of word embedding spaces implies the existence of parallelograms between the four terms of an analogy. We investigate the presence of analogy parallelograms in various word embedding spaces for various languages by relying on analogies contained in several analogy test sets. We report a negative result: no parallelogram is found. We also discuss another possibility to approach the word as a small n-sphere instead of being a point inside the embedding space. Thus an analogy is formed as parallelogram between four n-spheres.

Keywords: word embeddings, analogy test sets, analogy, parallelogram

1. Introduction

Previous works, like (Mikolov et al., 2013b) and (Levy and Goldberg, 2014), claimed that there are linguistic regularities in word embedding spaces, and even sentence embeddings (Zhu and de Melo, 2020). These regularities emerge as parallelograms on hyperplanes in the embedding space. Figure 1 presents an illustration of the claim where the four terms in an analogy make a parallelogram in the embedding space.

This claim was challenged by (Murena et al., 2018). When the space is curved as in differential manifolds, the equality i will not hold for the analogy $A : B :: C : D$. They proposed a parallelogramoid procedure using geodesic shooting and parallel transport to explain the analogical relation between words along curvatures in Riemannian manifolds. Figure 2 shows an illustration of the proposal.

In this paper, we perform an investigation of analogies that possibly exist in word embedding spaces. To investigate the existence of parallelograms in embedding spaces, we perform experiments in discovering the analogies contained in various analogy test sets. We explore embedding spaces and try to extract all analogies from these analogy test sets.

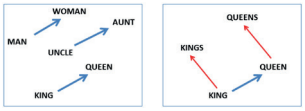


Figure 1: Parallelograms in word embedding space. Figure copied from (Mikolov et al., 2013b)

2. Extraction of analogies from word embedding space

In the following section, we describe how the parallelogram defines an analogy and its implication on how we can extract analogies from a word embedding space.

2.1. Analogy as parallelogram

Figure 1 shows that linguistic regularity between four words, which is an analogy, makes a parallelogram. This parallelogram implies that there is an equality between ratios on the left hand and on the right hand of the analogy. For example, for the analogy *man* : *woman* :: *king* : *queen,* we should have the equality between the ratios of *man* : *woman* and *king* : *queen* (top-right facing arrows in Fig. 1). In addition, to make a parallelogram, equality in the other direction is necessary: *man* : *king* = *woman* : *queen.* This is also true for analogy between other objects than words, like numbers. There is an equality between the ratios $2 : 4$ and $3 : 6$ for the analogy $2 : 4 :: 3 : 6$ (because of the properties of subtraction, the equality $2 : 3$ and $4 : 6$ is implied).

Unfortunately, previous works, like (Mikolov et al., 2013b; Pennington et al., 2014), did not use this definition to solve the analogical equation: coin the word D given the tree words, A, B and C. After calculating the vector $\vec{D}$ from $\vec{A}$, $\vec{B}$ and $\vec{C}$, they will take vector $\vec{D}'$ which is the closest vector to $\vec{D}$. This is a relaxation of the claim that an anal-

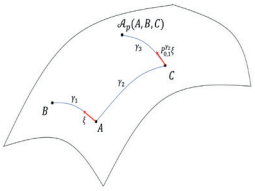


Figure 2: Parallelogramoid procedure on a Riemannian manifold. Figure copied from (Murena et al., 2018)

Lang.	Pre-trained embedding model		
	fastText	word2vec	SENNA
bel	*звер-жанчына*	-	-
zho	万凰		
fra	*reine*†	*reine*†	*princesse*
deu	*königin*†	*Sibylle*	
ind	*kerajaan*	*rajanya*	-
jav	*Kirata*	*raja-raja*	-
sun	*Warmadewa*	-	-
tha	*ราชินี* †	*ราชินี* †	

Table 1: Solution of the analogical equation *man* : *woman* :: *king* : *x* in various languages using various pre-trained embedding models. A dagger mark (†) shows a correct answer according to human judgement (5 times out of 14). A hyphen ('-') means that there was no available pre-trained model for that particular language at the time the experiments were conducted.

ogy is a parallelogram. Prior to this, our intuition is that it will be very hard to find true parallelograms inside a word embedding space.

Table 1 shows the result of a preliminary experiment on solving the analogical equation *man* : *woman* :: *king* : *x* using three different pre-trained embedding models: fastText, word2vec and SENNA. This experiment was conducted in various languages: Belarussian (bel), Chinese (zho), French (fra), Indonesian (ind), Javanese (jav), Sundanese (sun) and Thai (tha). The answers were checked by native speakers of the language. Results show that most of the answers are incorrect, except for French (fastText and word2vec), German (fastText) and Thai (fastText and word2vec).

2.2. Notions on analogy and extraction of analogical clusters

An analogy between four words, A, B, C and D, is noted as $A : B :: C : D$. The condition for an analogy to hold is an equality between the ratios, as shown in Formula (1).

$$A : B :: C : D \overset{\Delta}{\iff} \begin{cases} A : B &= C : D \\ A : C &= B : D \end{cases} \quad (1)$$

The ratio between two words, A and B, is defined as the difference of the vector representations of the words: $A : B \overset{\Delta}{=} \vec{A} - \vec{B}$. We thus replace Formula 1 by Formula 2. With difference between vectors, similarly as with numbers, the two equalities in the right part of Formula (2) are equivalent.

$$A : B :: C : D \overset{\Delta}{\iff} \begin{cases} \vec{A} - \vec{B} &= \vec{C} - \vec{D} \\ \vec{A} - \vec{C} &= \vec{B} - \vec{D} \end{cases} \quad (2)$$

Based on that, an analogical cluster is defined as a group of word pairs with the same ratio. This is basically the same as categories found in analogy test sets like capital-common-countries, currency, etc. (see Section 3.).

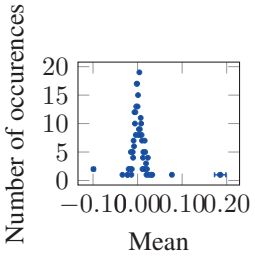


Figure 3: Distribution of fastText's vector means for each dimension with their standard deviation

$$\begin{matrix} A_1 : B_1 \\ A_2 : B_2 \\ \vdots \\ A_n : B_n \end{matrix} \overset{\Delta}{\iff} \begin{matrix} \forall (i,j) \in \{1,\ldots,n\}^2, \\ A_i : B_i :: A_j : B_j \end{matrix} \quad (3)$$

3. Data

There are two main resources used in this work: pre-trained word embedding models and analogy test sets. We investigate whether the analogies contained in analogy test sets emerge as true parallelograms in pre-trained word embedding spaces.

3.1. Pre-trained word embedding models

We use fastText (Mikolov et al., 2018) pre-trained models. They provide models in various languages which allow us to compare across different languages. The models were trained on Common Crawl and Wikipedia using CBOW with position-weights. The models we used were trained in 300 dimensions, with character n-grams of length 5, a window of size of 5 and 10 negative samples.

Let us now turn to the distribution of values inside the vector. We sample 1,000 vectors from the embedding model. For each dimension of the vector, we calculate the mean and standard deviation. Figure 3 plots the means of values for the 300 dimensions of the fastText pre-trained model for English. We observe that the graph roughly follows a Gaussian distribution centered around zero. The error bars in the figure show the standard deviation of the mean. These error bars are not visible in the figure because most of the standard deviations are around 0.05, which is very small.

3.2. Analogy test sets

We survey several analogy test sets that are publicly available. Table 2 shows the language availability of different analogy test sets.

- **Google** analogy test set[1] (Mikolov et al., 2013a) is probably the first analogy test set widely used since the emergence of word embedding models. It contains general knowledge questions, like country-capital, and morphological questions, like singular-plural form of nouns. The analogy test set was originally only available in English.

[1] `http://download.tensorflow.org/data/questions-words.txt`

Test set	Language						
	en	fr	hi	pl	ru	ar	ja
Google	✓						
fastText		✓	✓	✓			
BATS	✓						✓
MGAD			✓		✓	✓	

Table 2: Survey on the availability of analogy test sets

- **fastText** analogy test set[2] (Grave et al., 2018) is provided alongside the pre-trained models. The test set follows the format of its predecessor, Google analogy test set, and is available in French, Hindi and Polish.
- Bigger Analogy Test Set, usually called **BATS**[3] (Gladkova et al., 2016), is a bigger and more balanced analogy test set in comparison to Google and fastText analogy test set. The analogy test set is also available for Japanese in a version called jBATS[4] (Karpinska et al., 2018).
- Multilingual Generation of Analogy Datasets (**MGAD**)[5] (Abdou et al., 2018) is an analogy test set extracted from Universal Dependency treebanks. Thus, the analogical questions are restricted only to morphological phenomena. It is available in Hindi, Russian and Arabic.

Table 3 gives examples of the analogies contained in one of the analogy test sets used in this work, which is the Google analogy test set. The analogies are grouped into categories. Thus, all word ratios of analogies belong to the same category shall represented by the same ratio.

4. Experimental protocol

The purpose of our experiment is to investigate the existence of parallelograms inside the embedding spaces. We rely on analogy test sets as our ground truth. We investigate whether analogies contained in the analogy test sets actually make parallelograms. As the analogy test sets are already organised into categories, we check whether ratios in analogies that belong to the same categories are actually the same, i.e, whether one category makes one analogical cluster.

We carry out experiments in extracting analogical clusters from sets of words contained in each category of an analogy test set. Words are represented as vectors given by a pre-trained word embedding model. The extracted analogical clusters are expected to be similar with the categories contained in the analogy test set. To extract the analogical clusters, we use two different approaches.

The first approach relies on the strict definition of analogies where the equality of ratios has to hold in order to have an analogy. The algorithm to extract analogies from a given set of words is already presented elsewhere, such as (Lepage, 2014; Fam and Lepage, 2017). However, to ensure the equality of ratios, these techniques apply only to natural numbers (integer values). We convert the real values found on the vector dimensions into integer values by approximation, up to a certain precision after the decimal point. Formula (4) illustrates the approximation on a vector, with a precision of 3.

$$\begin{pmatrix} 0.1435 \\ 0.3496 \\ \vdots \\ 0.1180 \end{pmatrix} \implies \begin{pmatrix} 143 \\ 349 \\ \vdots \\ 118 \end{pmatrix} \tag{4}$$

The second approach involves a common clustering algorithm. We perform DBSCAN clustering algorithm to cluster ratios. The reason behind it is the scalability and the geometry used (distances between points) which is aligned with the constraint that we use here with analogy. In this work, we use the implementation provided by the scikit-learn [6] library.

5. Results and analysis

Table 4 shows the number of analogical clusters extracted from different analogy test sets with various precision values using the strict definition of equality of ratios. The table is simply full of zeros. No parallelogram between words in the analogical test sets, as represented by vectors in any of the pre-trained embedding spaces considered, was found. This observation, which constitutes a negative result, gives support to the construction proposed in (Murena et al., 2018).

Table 5 shows the number of analogical clusters extracted from different analogy test sets with various precision values using DBSCAN clustering algorithm.

6. Discussion

In the following section, we discuss on the insights from the result while also considering some previous works done so far in the community.

6.1. A word as an area in the space

The analogy test sets are mainly used to assess the quality of a word embedding space. The test sets demand the embedding space to follow certain linguistic regularities, which are claimed to be semantical. However, in practice, some heuristics and tricks are introduced while performing the analogy task. For example, deleting the words included in the problem itself (the term A, B and C) from the candidates of the solution. Word D is enforced to be different from the words A, B and C even when the true vector D that is calculated by the algorithm is closer to any of these words.

We propose to think of a word not as a point, but rather as a small n-sphere in the embedding space. By adopting

[2] https://fasttext.cc/docs/en/crawl-vectors.html
[3] https://vecto.space/projects/BATS/
[4] https://vecto.space/projects/jBATS/
[5] https://github.com/rutrastone/MGAD
[6] https://scikit-learn.org/stable/modules/clustering.html#dbscan

Category	#	Example
capital-common-countries		*Athens* : *Greece* :: *Baghdad* : *Iraq*
capital-world		*Abuja* : *Nigeria* :: *Accra* : *Ghana*
currency		*Algeria* : *dinar* :: *Angola* : *kwanza*
city-in-state		*Chicago* : *Illinois* :: *Houston* : *Texas*
family		*boy* : *girl* :: *brother* : *sister*
gram1-adjective-to-adverb		*amazing* : *amazingly* :: *apparent* : *apparently*
gram2-opposite		*acceptable* : *unacceptable* :: *aware* : *unaware*
gram3-comparative		*bad* : *worse* :: *big* : *bigger*
gram4-superlative		*bad* : *worst* :: *big* : *biggest*
gram5-present-participle		*code* : *coding* :: *dance* : *dancing*
gram6-nationality-adjective		*Albania* : *Albanian* :: *Argentina* : *Argentinean*
gram7-past-tense		*dancing* : *danced* :: *decreasing* : *decreased*
gram8-plural		*banana* : *bananas* :: *bird* : *birds*
gram9-plural-verbs		*decrease* : *decreases* :: *describe* : *describes*

Table 3: Excerpt of analogies contained in the Google analogy test set

Test set	P	Language						
		en	fr	hi	pl	ru	ar	ja
Google	1	0	-	-	-	-	-	-
	2	0	-	-	-	-	-	-
	3	0	-	-	-	-	-	-
	4	0	-	-	-	-	-	-
fastText	1	-	0	0	0	-	-	-
	2	-	0	0	0	-	-	-
	3	-	0	0	0	-	-	-
	4	-	0	0	0	-	-	-
BATS	1	0	-	-	-	-	-	0
	2	0	-	-	-	-	-	0
	3	0	-	-	-	-	-	0
	4	0	-	-	-	-	-	0
MGAD	1	-	-	0	-	0	0	-
	2	-	-	0	-	0	0	-
	3	-	-	0	-	0	0	-
	4	-	-	0	-	0	0	-

Table 4: Number of analogical clusters extracted using the strict definition of equality if ratios from approximated vectors with various precision (P). The character hyphen ('-') means that there is no test set available in the corresponding language.

Test set	P	Language						
		en	fr	hi	pl	ru	ar	ja
Google	1	0	-	-	-	-	-	-
	2	0	-	-	-	-	-	-
	3	0	-	-	-	-	-	-
	4	0	-	-	-	-	-	-
fastText	1	-	0	0	0	-	-	-
	2	-	0	0	0	-	-	-
	3	-	0	0	0	-	-	-
	4	-	0	0	0	-	-	-
BATS	1	0	-	-	-	-	-	0
	2	0	-	-	-	-	-	0
	3	0	-	-	-	-	-	0
	4	0	-	-	-	-	-	0
MGAD	1	-	-	0	-	0	0	-
	2	-	-	0	-	0	0	-
	3	-	-	0	-	0	0	-
	4	-	-	0	-	0	0	-

Table 5: Same as Table 4 but using the DBSCAN clustering algorithm.

this approach, we may find that this small n-sphere, standing for a given word, includes several words. The visibility and representation of the meaning of a word in the embedding space is extended by the proximity of the words in the neighbourhood. Thus, the analogy is now formed by the four small n-spheres instead of just four points in the embedding space. Here, we can imagine that the words *king*, *duke*, *prince*, *count*, etc. may have their extended n-sphere intersect or even that one in included in another one. This makes the heuristics and tricks that we mentioned earlier sound more natural.

6.2. Hypernymy and hyponymy

Capitalising on the approach of a word as a small n-sphere, we may propose another explanation on how an embedding space can point at candidate solutions which are hypernyms or hyponyms of the true answer. For example, we may get a king's name instead of the word *king* itself. This varies depending on the corpus of which the embedding space is trained on. The discussion comes to whether there is any feature for the degree of generality of a word in embedding spaces; and whether distributional semantics can capture hyponymy and hyponymy. (Yu et al., 2015; Sanchez and Riedel, 2017) provide experiments on several datasets. They observe whether hypernymy structures exist and whether the relation is preserved inside the embedding space.

6.3. Task of analogy

Let us now reflect back on the task of analogy. It is important to ask again on what better analogies can be designed. One possible approach is to extract all possible analogies from a word embedding space. We need to have a critical view or be able to analyse these extracted analogies to draw conclusions about their validity or acceptability. Of course, we have to be more precise about the task at hand. If the goal is to assess the quality of the embedding space, then it is strictly demanded that previously mentioned tricks are not fair.

7. Conclusion

We investigated the existence of true parallelograms inside word embedding spaces relying on analogies contained in analogy test sets. The analogies were defined as equality of ratios between the four terms. This constrains the analogies to be true parallelograms. Experimental results showed that no analogy can be extracted from any of the word embedding spaces we had at our disposal. We thus confirm that no true parallelogram corresponding to an analogy exists inside embedding spaces.

This negative result supports the construction proposed in (Murena et al., 2018) where parallelograms for analogies are claimed not to exist in differential manifolds. Instead, they propose that analogies should follow the *Ricci* curvature rather than making parallelograms.

In this paper, we discussed another way to approach the representation of a word in the embedding space: a word is not a point but rather a small-n-sphere.

Acknowledgement

The work reported here was supported by a JSPS Grant, Number 21K12038 (Kakenhi C), entitled "Theoretically founded algorithms for the automatic production of analogy tests in NLP".

References

Abdou, Mostafa, Artur Kulmizev, and Vinit Ravishankar, 2018. MGAD: Multilingual generation of analogy datasets. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: ELRA.

Fam, Rashel and Yves Lepage, 2017. A study of the saturation of analogical grids agnostically extracted from texts. In *Proceedings of the Computational Analogy Workshop at the 25th International Conference on Case-Based Reasoning (ICCBR-CA–2017)*. Trondheim, Norway.

Gladkova, Anna, Aleksandr Drozd, and Satoshi Matsuoka, 2016. Analogy-based detection of morphological and semantic relations with word embeddings: What works and what doesn't. In *Proceedings of the NAACL-HLT SRW*. San Diego, California, June 12-17, 2016: ACL.

Grave, Edouard, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov, 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Karpinska, Marzena, Bofang Li, Anna Rogers, and Aleksandr Drozd, 2018. Subcharacter Information in Japanese Embeddings: When Is It Worth It? In *Proceedings of the Workshop on the Relevance of Linguistic Structure in Neural Architectures for NLP*. Melbourne, Australia: ACL.

Lepage, Yves, 2014. Analogies between binary images: Application to Chinese characters. In Henri Prade and Gilles Richard (eds.), *Computational Approaches to Analogical Reasoning: Current Trends*. Berlin, Heidelberg: Springer, pages 25–57.

Levy, Omer and Yoav Goldberg, 2014. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*. Ann Arbor, Michigan: ACL.

Mikolov, Tomás, Kai Chen, Greg Corrado, and Jeffrey Dean, 2013a. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun (eds.), *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Mikolov, Tomas, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin, 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Mikolov, Tomas, Wen-Tau Yih, and Geoffrey Zweig, 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT–2013)*. Atlanta, Georgia: ACL.

Murena, Pierre-Alexandre, Antoine Cornuéjols, and Jean-Louis Dessalles, 2018. Opening the parallelogram: Considerations on non-euclidean analogies. In *Proceedings of the 26th International Conference (ICCBR–2018)*. Stockholm, Sweden.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning, 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Sanchez, Ivan and Sebastian Riedel, 2017. How well can we predict hypernyms from word embeddings? a dataset-centric analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: ACL.

Yu, Zheng, Haixun Wang, Xuemin Lin, and Min Wang, 2015. Learning term embeddings for hypernymy identification. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15. AAAI Press.

Zhu, Xunjie and Gerard de Melo, 2020. Sentence analogies: Linguistic regularities in sentence embeddings. In *Proc. of the 28th COLING*. Barcelona, Spain (Online).

Text Classification for Subjective Phenomena on Disaggregated Data and Rater Behaviour

Ewelina Gajewska[1], Barbara Konat[1]

[1]Faculty of Psychology and Cognitive Sciences,
Adam Mickiewicz University, Poznań, Poland
ewegaj@st.amu.edu.pl, bkonat@amu.edu.pl

Abstract

Phenomena such as emotional experience and offensive language perception are highly subjective in nature. Yet, the dominant approach in building automatic emotion and hate speech detection systems is based on the opinion of the majority. Recently, however, a personalised or human-centred approach has been proposed by the computational social scientists. In the current paper, we propose a novel method for modelling individual perspective in emotion detection and abusive language recognition, following existing works in this area (Miłkowski et al., 2021). We show that the personalised approach that implements our *Personalisation Metric* (PM) outperforms traditional majority-based methods in regard to subjective phenomena such as emotion and abusive language detection. Proposed method could be successfully used in the development of more accurate classification models suitable for the opinions of individuals as well as in recommendation systems.

Keywords: EDO 2023, emotion recognition, human-centred NLP, offensive language, recommendation systems

1. Introduction

Emotion detection in textual data has become a topic of interest in recent years, both for academic scholars and the industry. Emotion analysis is commonly employed in mining customer opinions and investigating public attitudes towards candidates in political campaigns. It is also used in psychotherapy sessions for assessing emotional states of patients (Tanana et al., 2021).

On the other hand, the growth of social media foregrounded the problem of hate speech and offensive language. However, currently available tools are insufficient to deal with the moderation problem. Thus, offensive language detection remains challenging due to the subjective nature of the task and the quality of data annotation, which is usually based on the opinion of the majority (Binns et al., 2017; Sap et al., 2021). For example, Waseem (2016) found systematic differences in the annotations given by the experts (anti-racism activists) and the crowd on hate speech. Therefore, the aggregated or the average vote does not fully represent the viewpoint of any side here and cannot be regarded as high quality data. Moreover, Sap et al. (2021) investigated demographic and political factors that influence the ratings on toxic language. The results indicate that more conservative annotators are more likely to rate African American English dialect as toxic, for example.

In recent years, researchers proposed to change the perspective from data-centric to human-centred in the NLP field, accounting for social factors in tasks such as offensive language detection, sentiment analysis and sarcasm recognition (Kocoń et al., 2021). However, techniques proposed so far are limited to specific types of data (numeric label scores in the case of (Miłkowski et al., 2021)) or require additional information about individuals such as demographic features, personality types or previous activity on social media platforms that could be difficult to obtain (Hovy, 2015; Lukin, Anand, Walker and Whittaker, 2017; Kocoń et al., 2021). The proposed Personalisation Metric (described in detail in Section 3) does not rely on any private information about users and could be calculated based only on labels given by individuals. Moreover, it is designed for categorical labels, which are predominant in machine learning tasks.

Researchers in the field also called for releasing annotator-level data instead of only so-called "ground truth" labels obtained through the majority voting technique (Prabhakaran, Davani and Diaz, 2021). Some noticed that those aggregated labels do not take into account perspectives of minority groups and that they do not even reflect the general opinion of the "average Jane" (Rizos and Schuller, 2020). Akhtar, Basile and Patti (2021) state that the NLP field needs novel methods to model conflicting perspectives in the automated systems for abusive language detection. Instead of majority-voted labels, they propose to differentiate different groups of individuals based on socio-demographic factors and develop separate gold standards and algorithms for each of them. Similarly, Basile (2020) emphasises that current machine learning techniques in NLP need to be adapted to subjective and pragmatic phenomena in order to create fair and inclusive models.

2. Related Work

In the current paper we adopt a human-centred approach introduced in (Kocoń et al., 2021). In particular, we focus on the microscopic level – the perspective of individuals on highly subjective tasks such as emotion and abusive language perception. Kocoń et al. (2021) differentiate also the mesoscopic and macroscopic levels, that is the perspective of selected groups of individuals and the general view, respectively. In their own study models

Contents

developed on the group-based labels performed better than the general view approach. For the microscopic level they include ids of annotators that also boost the performance of offensive content classifiers.

Previous works show also the improvement of classification performance on sentiment analysis and topic identification when demographic factors are taken into account (Hovy, 2015).

Several works reported high disagreements between raters on hate speech annotation and therefore called for modelling annotator perspectives in hate speech detection systems (Binns et al., 2017; Akhtar, Basile and Patti, 2020). Moreover, Larimore et al. (2021) demonstrate that racial identity as well as textual features of tweets influence annotator perceptions of racism. Similar problems could be observed in sentiment annotation. For example, Prabhakaran, Davani and Diaz (2021) found that around one-third of annotators achieve very low agreement scores with the majority voted labels.

Davani, Díaz and Prabhakaran (2022) experiment with 3 different techniques to implement individual perspectives into machine learning models on detection of online abuse and emotions. First, they present a multi-task approach that treats the prediction of labels for each annotator as separate subtasks. Second, they conceptualise the task as a multi-label classification where each label corresponds to individual annotators' labels. Third, the authors train an ensemble of models, one dedicated for each annotator where a final label, however, is majority voted. Alternatively, they propose to estimate uncertainty of a model's prediction to reflect the disagreements in the annotation.

Another strand of work that shows the importance of a personalised approach is the study of annotator bias in data. Results of (Wich, Bauer and Groh, 2020) indicate that political bias negatively impacts the automated detection of hate speech and in result could lead to racial discrimination.

3. Methodology

3.1. Personalisation metric

Proposed in the current paper *Personalisation Metric* (PM) extends *Personal Emotion Bias* (PEB) measure introduced by Miłkowski et al. (2021). PEB is based on Z-scores between the average and the individual user rating and is suitable for real-valued ratings (numeric variables) of emotional intensity. We introduce the PM metric that is suitable for categorical labels instead. PM comprises two well-known measures of agreement, i.e. the Cohen's Kappa statistic (Cohen, 1960) and percent agreement (accuracy). Therefore, the PM metric measures the similarity of opinions between two entities given a set of categories. Here, we consider the majority as one entity and an individual rater as the other entity.

We use accuracy and Kappa are complementary statistics here because of the so-called prevalence problem (generated by an imbalanced distribution of categories) that might yield low Kappa coefficients in cases where accuracy indicates almost perfect agreement between two entities (Eugenio, Glass, 2004). As a result, each rater is assigned with two values – Kappa and accuracy score which we jointly call the PM metric. In order to calculate these values, a sample of data annotated by an individual is compared against the same sample of data annotated with majority-voted labels. Thus, the PM metric indicates how similar is the opinion of an individual rater compared to the majority of people.

In both statistics, the values close to 1 signal perfect agreement, whereas scores around 0 indicate no agreement in the case of accuracy, and random agreement in the case of Cohen's Kappa. Negative values of Kappa reflect less than random agreement between two entities. Thus, we treat disagreements as an additional source of information instead of noise, which is common in aggregation-based approaches.

3.2. Model

Although Transformer-based architectures are currently considered state-of-the-art, Convolution Neural Networks (CNN) as well as Recurrent Neural Networks (RNN) still achieve good performance and are commonly used in text classification tasks (Tam, Said, Tanriöver, 2021). In particular, models based on the combination of CNN and RNN networks achieve superior performance compared with other architectures (Wang, Jiang and Luo, 2016). Therefore, our model makes use of CNN and bidirectional Long Short Term Memory (BiLSTM) networks. As a text representation method we employ GloVe 100-dimensional embeddings (Pennington, Socher, Manning, 2014). ReLu is the activation function in all layers except the last classification layer where we use sigmoid or softmax function in the case of emotion and abuse detection, respectively. Summarised description of the model used in the study can be found in Table 1.

Layer	Parameters
Embedding	GloVe 100-dim
CNN	Filters: 128; size: 5
MaxPooling	Size: 3
BiLSTM	Units: 64
GlobalMaxPooling	–
Dense	Units: 264
Dropout	Rate: 0.3
Concatenate	–
Dense	Units: 128

Table 1. Architecture of the proposed CNN-BiLSTM model.

3.2.1. Emotion detection

We make use of the *GoEmotions* dataset (Demszky et al., 2020) collected from Reddit and annotated with 27 categories of emotion. Here, in addition to majority voted labels, the authors release all individual annotations assigned by different raters. It therefore allows us to study individual perspectives in regard to emotion perception. For the purpose of the study, we included annotations only on 6 basic emotions in Ekman's taxonomy – fear, anger, sadness, surprise, joy, disgust, as well as a neutral category. We also decided to discard raters that annotated less than 334 data points (25th percentile) in order to

achieve stable scores between train and test sets for the PM metric. In addition, in order to have more data available for the training we merged annotations on selected emotions into the chosen 6 categories based on the correlation analysis conducted by the authors of GoEmotions. In result, the available corpus comprises over 128k data points (over 50k unique comments) that were annotated by 61 raters in total.

The GoEmotions dataset allows for multi-category multi-label classification of emotions, as annotators were allowed to indicate more than one emotion in a given Reddit comment. The data is however highly imbalanced as the neutral label is present in 42% of texts, and other categories are observed in 4% to 22% of the cases.

We design two conditions with respect to emotion detection at the individual level. First, we calculate the PM metric between each annotator and the majority voted label, separately for the train and test sets. Separate calculation of PM values for users in a test set allows to imitate a new set of users instead of copying PM scores from the train set. Thus, training is conducted on a different set of users and PM scores than evaluation. PM scores comprise here additional features next to text embeddings (PM condition). Second, we use ids of annotators transformed into one-hot encoded vectors as a set of additional features following related works (Kocoń et al., 2021) (vector-id condition).

In addition, we compare the performance achieved by those models with the traditional approach to text classification, i.e. based on the opinion of the majority (majority condition). Here, we make use of the results reported by the authors of the GoEmotions (Demszky et al., 2020) – BERT model fine-tuned for the classification of Ekman's 6 basic emotions. Therefore, we could compare not only the performance of the proposed approach (a model with PM metric features) with another method designed for individualised emotion recognition, but also the usefulness of a personalised approach with the traditional one based on the majority aggregated labels.

3.2.2. Abusive language detection

With respect to abusive language detection, we use the subset of 4k examples of the *ConvAbuse* corpus released by the authors (Curry, Abercrombie and Rieser, 2021). It encompasses short human-machine dialogues. Each data sample comprises 4 dialogical turns – 2 generated by a chatbot (conversational AI system) and 2 created by a user. For the purpose of our study, we concatenate all 4 turns into one text for each example that is later fed to a deep learning model. We made use of the annotations on abusive language on a 5-point scale: non-abusive, ambiguous, negative and mildly offensive, negative and insulting/abusive attitude, and strongly negative with overt incitement to hatred, violence or discrimination. The authors also make available individual annotations assigned by human subjects which allows us to study the subjectivity of abusive language perception and develop classification models that account for the individual view on abusive content perception. In total, there are over 12k instances of text annotated by 8 raters. Similarly, as in the emotion detection task, the data is highly imbalanced – over 78% of cases are assigned with the ambiguous category, and the other classes comprise from 2% to 7% of the data.

In regard to abusive language detection, we compare two approaches to text classification – the personalised one, which makes use of the PM metric (PM condition) and the popular majority-based approach (majority condition). In addition, we conduct cross-examination, i.e. training a model on majority aggregated labels and testing on individual annotations (cross condition). Similarly as in the case of emotion detection, we compute the PM metric for each annotator, separately for a train set and a test set. The PM metric comprises a pair of additional features fed to a model, next to word embeddings. In regard to the majority and cross conditions, text features (GloVe embeddings) comprise an input to a model. We release our source code regarding the abusive language detection study in Google Colab[1].

4. Results

We report the average results from 5 random splits of data. Each time the training set comprises 80% of data, and the remaining 20% is used for evaluation purposes. Performance is evaluated in terms of macro-averaged F1 scores as it weights equally all categories considered in the classification.

4.1. Emotion detection

In regard to emotion detection, in Table 2 we report F1 scores for all 3 models. The proposed CNN-BiLSTM-PM model achieves superior results compared with the other two approaches. It outperforms the BERT model by 4 percentage points (6%), and the vector-id model by 14 percentage points (26%). Furthermore, the proposed CNN-BiLSTM model is designed for multi-label prediction instead of single emotion detection as in the case of the BERT model.

Model	F1-macro (%)
CNN-BiLSTM-PM	**67.68 (0.55)**
CNN-BiLSTM-vector-id	53.72 (0.26)
BERT-GoEmotions-majority	64.00 (n/a)

Table 2. Results of the emotion recognition task (standard deviations reported in parentheses).

4.2. Abusive language detection

In regard to the cross condition, 20% of unique texts were sampled for a test set and annotations from all raters for those texts were retrieved from the full dataset. Therefore, in the evaluation phase the model is fed with text samples not seen before and has to predict labels for all raters that annotated a given text.

Results reported in Table 3 indicate that the proposed personalised model with PM features outperforms the traditional approach by almost 7 percentage points in terms of F1 scores. However, both models achieve superior results with respect to the model in the third (cross) condition.

[1]https://colab.research.google.com/drive/1x2FDbRPx9d_B9YlhP6nr1P8TJAzZp0SW?usp=sharing

Model	F1-macro (%)
CNN-BiLSTM-PM	**47.60 (2.76)**
CNN-BiLSTM-majority	40.90 (3.84)
CNN-BiLSTM-cross	38.64 (1.87)

Table 3. Performance results for abusive language detection (standard deviations reported in parentheses).

5. Discussion

We provide further evidence that the traditional "gold standard" approach to the study of highly subjective phenomena such as emotion and abusive language detection is no longer suitable in computational linguistics. We show the value of implementing a personalised approach to supervised machine learning models, in particular in highly subjective tasks. We introduce the Personalisation Metric, which significantly improves the quality of prediction of deep learning models on the one hand, and is easy to implement on the other hand. Although other features related to the users or annotators proved useful in the previous studies, for example demographic factors (Hovy, 2015), they are often difficult to obtain due to privacy issues or missing data on social media platforms, among others.

In the current study, we propose a method to model individual factors that influence the perception of highly subjective phenomena such as emotions and abusive language. The introduced Personalisation Metric is suitable for categorical labels and classification tasks (i.e., categorical labels are taken to calculate the metric score), and therefore extends previous solutions proposed for numerical labels (i.e., numerical labels are taken to calculate the metric) and regression models (see Miłkowski et al., 2021).

A personalised approach to text classification constitutes an alternative to a popular majority-based method where a machine learning model is both trained and evaluated on majority-aggregated labels obtained from a set of annotators. However, when applied to predict individual users' preferences in reality, those models perform rather poorly (Gordon et al., 2021). Thus, a new approach that takes into account not only the opinion of the majority but also individual users is needed. Others provide solutions with the use of demographic information (Hovy, 2015), we propose the one based on annotation behaviour of an individual. We demonstrate that the proposed method outperforms standard majority-based classifiers applied to both majority-aggregated and individual labels (BERT-GoEmotions-majority and CNN-BiLSTM-majority, and CNN-BiLSTM-cross models, respectively) as well as models that incorporate information about id number of annotators (CNN-BiLSTM-vector-id model). Results obtained in the current study corroborate previous findings in this area (Kocoń et al., 2021; Miłkowski et al., 2021).

Scalability of the proposed method to new users could be addressed in two ways. First, new users could be provided with a sample of data to annotate in order to compute the PM metric and define their deviation from the majority opinion. Second solution makes use of a collaborative filtering technique, broadly used in recommendation systems. Here, the PM metric could be computed as the average value from several users similar in some aspects to a new user. This similarity could regard user behaviour on a platform or demographic information such as gender or age. Future studies could examine the suitability of those two methods for the proposed personalised approach to text classification.

Although the traditional approach to text classification works well for the majority of people, the alternative acknowledges those individuals that do not have enough representation in the majority perspective to work satisfactorily for them. It concerns in particular subjective phenomena such as emotion and abusive language perception. Further research in natural language processing could examine the impact of incorporation of user-based information on classification performance and advance the field not only in terms of new state-of-the-art performance but also practical suitability for the end users (Gordon et al., 2021).

Acknowledgments

The work reported in this paper was partially supported by the Polish National Science Centre under grant 2020/39/D/HS1/00488.

References

Akhtar, S., Basile, V. and Patti, V. (2020, October). Modeling annotator perspective and polarized opinions to improve hate speech detection. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing (Vol. 8, pp. 151-154)*.

Akhtar, S., Basile, V. and Patti, V. (2021). Whose opinions matter? perspective-aware models to identify opinions of hate speech victims in abusive language detection. *arXiv preprint arXiv*:2106.15896.

Basile, V. (2020). It's the end of the gold standard as we know it. on the impact of pre-aggregation on the evaluation of highly subjective tasks. In *2020 AIxIA Discussion Papers Workshop* (Vol. 2776, pp. 31-40). CEUR-WS.

Binns, R., Veale, M., Kleek, M.V. and Shadbolt, N. (2017, September). Like trainer, like bot? Inheritance of bias in algorithmic content moderation. In *International conference on social informatics (pp. 405-415)*. Springer, Cham.

Cohen, J. (1960). Kappa: Coefficient of concordance. *Educ Psych Measurement*, 20(37), 37-46.

Curry, A.C., Abercrombie, G. and Rieser, V. (2021). ConvAbuse: Data, analysis, and benchmarks for nuanced abuse detection in conversational AI. *arXiv preprint arXiv*:2109.09483.

Davani, A.M., Díaz, M. and Prabhakaran, V. (2022). Dealing with disagreements: Looking beyond the majority vote in subjective annotations. In *Transactions of the Association for Computational Linguistics*, 10, 92-110.

Demszky, D., Movshovitz-Attias, D., Ko, J., Cowen, A., Nemade, G. and Ravi, S. (2020). GoEmotions: A dataset of fine-grained emotions. *arXiv preprint arXiv*:2005.00547.

Eugenio, B. D., & Glass, M. (2004). The kappa statistic: A second look. *Computational linguistics*, *30*(1), 95-101.

Gordon, M. L., Zhou, K., Patel, K., Hashimoto, T., & Bernstein, M. S. (2021, May). The disagreement deconvolution: Bringing machine learning performance metrics in line with reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (pp. 1-14)*.

Hovy, D. (2015, July). Demographic factors improve classification performance. In *Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing (pp. 752-762)*.

Kocoń, J., Figas, A., Gruza, M., Puchalska, D., Kajdanowicz, T. and Kazienko, P. (2021). Offensive, aggressive, and hate speech analysis: From data-centric to human-centered approach. In *Information Processing & Management*, 58(5), 102643.

Larimore, S., Kennedy, I., Haskett, B. and Arseniev-Koehler, A. (2021). Reconsidering annotator disagreement about racist language: noise or signal?. *SocialNLP 2021*, 81.

Lukin, S.M., Anand, P., Walker, M. and Whittaker, S. (2017). Argument strength is in the eye of the beholder: Audience effects in persuasion. *arXiv preprint arXiv*:1708.09085.

Miłkowski, P., Gruza, M., Kanclerz, K., Kazienko, P., Grimling, D., and Kocoń, J. (2021, August). Personal bias in prediction of emotions elicited by textual opinions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop (pp. 248-259)*.

Pennington, J., Socher, R. and Manning, C.D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543)*.

Prabhakaran, V., Davani, A.M. and Diaz, M. (2021). On Releasing Annotator-Level Labels and Information in Datasets. In *Proceedings of The Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop* (pp. 133-138).

Rizos, G. and Schuller, B.W. (2020, June). Average jane, where art thou?–recent avenues in efficient machine learning under subjectivity uncertainty. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (pp. 42-55)*. Springer, Cham.

Sap, M., Swayamdipta, S., Vianna, L., Zhou, X., Choi, Y. and Smith, N.A. (2021). Annotators with attitudes: How annotator beliefs and identities bias toxic language detection. *arXiv preprint arXiv*:2111.07997.

Tam, S., Said, R.B. and Tanriöver, Ö.Ö. (2021). A ConvBiLSTM deep learning model-based approach for Twitter sentiment classification. In *IEEE Access*, 9, 41283-41293.

Tanana, M.J., Soma, C.S., Kuo, P.B., Bertagnolli, N.M., Dembe, A., Pace, B.T., Srikumar, V., Atkins, D.C. and Imel, Z.E. (2021). How do you feel? Using natural language processing to automatically rate emotion in psychotherapy. *Behavior Research Methods*, 53(5), 2069-2082

Wang, X., Jiang, W. and Luo, Z. (2016, December). Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In *Proceedings of COLING 2016* (pp. 2428-2437).

Waseem, Z. (2016, November). Are you a racist or am I seeing things? Annotator influence on hate speech detection on Twitter. In *Proceedings of the first workshop on NLP and computational social science* (pp. 138-142).

Wich, M., Bauer, J. and Groh, G. (2020, November). Impact of politically biased data on hate speech classification. In *Proceedings of the Fourth Workshop on Online Abuse and Harms* (pp. 54-64).

A new approach to generate teacher-like questions guided by text spans extraction

Thomas Gerald[1], **Sofiane Ettayeb**[1], **Louis Tamames**[1], **Ha-Quang**[3] **Le, Patrick Paroubek**[2], **Anne Vilnat**[2]

[1] Université Paris Saclay, CNRS, SATT Paris Saclay, LISN
[2] Université Paris Saclay, CNRS, LISN
firsname.lastname@lisn.upsaclay.fr
[3] Professorbob.ai
ha-quang.le@professorbob.ai

Abstract

Generating teacher-like questions and answers remains an open issue while being useful for students, teachers and teaching aid application providers. Given a textual course material, we are interested in generating non-factual questions which require an elaborate answer (implying some sort of analysis or reasoning). Despite the availability of annotated corpora of questions and answers, two main obstacles prevent the development of such generator using deep learning. Firstly, the amount of qualitative data is not sufficient to train generative approaches. Secondly, for a stand-alone application, we do not have an explicit support to guide the generation towards complex questions. In this article, we propose and compare several new retargetable language algorithms for answer text span support extraction and complex question generation, on secondary education course material use-case in French. We study the contribution of deep neural syntactic parsing and transformer based semantic representation, relying on the question type (according to our specific question typology) and the support text span in the context. We highlight the important role of nominal noun phrases and dependency relations, as well as the gain brought by recent transformer language models.

Keywords: corpus, question-answering, question generation

1. Introduction

In education, few mature approaches for teaching assistance using deep-learning methods are deployed. However, recent advances in Natural Language Processing (NLP) allows us to envision applications for the extraction, processing or generation of information for pedagogical purposes. We aim to develop a teaching assistant for generating non-factual questions (leading to elaborated answers) guided by text courses materials, implying some analysis or reasoning going beyond the simple restitution of factual data. The use-case chosen for our experiences concerns secondary courses of history in French language, in the context of a project funded by a technology transfer accelerator[1] in collaboration with a company[2] specialized in applications for education. The project aims to produce a question answering system with high pedagogical value inside an application able to guide students by partially answering course questions, redirecting them to relevant articles/courses, or proposing Multiple Choice Question (MCQ) to consolidate their knowledge.

To fulfill this objective we collected French question-answer pairs on education materials from both school books and Wikipedia. We currently have about 500 manually annotated question-answer pairs. Given a text course material, annotators qualified in the field of the course, were instructed to produce a set of questions of various types, indicating for each the text span of the course material from which the answer can be inferred.

The amount of annotated data we have now does not allow us to consider training or fine-tuning deep-learning generative approaches. Additionally, for a stand-alone application, we do not have access to the answer spans to guide the generation of the question, i.e. the passage of the text where the question generation system must focus on. In this paper we propose to train different transformer generation model for question generation using the corpus as evaluation material. Specifically, we study various support spans (named support or question support) extraction algorithms and compare them on their ability to produce teacher-like questions.

In the following, we first discuss works related to the question generation task; secondly, we describe the French educational corpus collected; in a third section we introduce the question support extraction algorithms and discuss our choices; then we present the experimental settings and protocol; we subsequently reports results of the experiments and discuss the abilities of the different approaches to generate teacher-like questions; finally, we conclude with a discussion of proposed and future approaches.

2. Related Works

Summaries, questions and answers generation have been and remain central topics in the NLP community. These different tasks have benefited from machine learning and deep learning advances. The "transformer" neural archi-

[1] *SATT Paris-Saclay, convention de maturation AVE-TAL*
[2] *ProfessorBob.ai, https://professorbob.ai/en/*

tecture (Vaswani et al., 2017) has provided significant improvements for generative approaches. These architectures have been revised in many ways by addressing multi-tasks (Raffel et al., 2020; Radford et al., 2019) or by scaling and increasing the size of the models and datasets used (Brown et al., 2020). Primarily developed for the English language these pre-trained models are now available in French with CamemBERT and FlauBERT (Martin et al., 2020; Le et al., 2020) language models (LM) or the BARThez generation model (Eddine et al., 2021). Most of the effective approaches now consider the multi-lingual settings for pre-training LM (Liu et al., 2020).

To adapt these models to a specific task, a common approach consists in fine-tuning language models on task oriented corpora. The corpus SQuAD (Rajpurkar et al., 2016) strongly participates in improving question-answering task, providing a large dataset of questions and extractive answers. More recently, Google published the corpus Natural Question (Kwiatkowski et al., 2019): a corpus with natural language questions, with long and short paragraphs for answers (extracted from the English Wikipedia). In conversational QA the corpus CANARD and QUAC (Elgohary et al., 2019; Choi et al., 2018) are available. For retrieval-based question-answering where documents are answers, the MSMarco passage dataset (Nguyen et al., 2016) is today the reference for training or fine-tuning models. If most QA corpora are available in English, French community also produced corpora such as FQuAD (Martin et al., 2020), Piaf (Keraron et al., 2020) or CALOR-QUEST (Bechet et al., 2019) for extractive QA. More recently, the CALOR-DIAL (Béchet et al., 2022) corpus addresses dialogue question answering for the French language. However, these corpora mainly rely on factual QA, where the answer is a short text such as a named entity, an event, a date, a quantity, or a location. Recently, a new corpus Autogestion (Antoine et al., 2022) has been created to address non-factual questions, the associated study demonstrates the inability of standard models to address most complex questions. All those corpora can be used for question generation (QG), answer generation, or answer extraction tasks.

Many QA works rely on those datasets particularly in machine reading comprehension (Liu et al., 2018; Yamada et al., 2020; Zhang et al., 2021). Moreover *QA* can be addressed within different frameworks such as the retrieval (Khattab and Zaharia, 2020; Karpukhin et al., 2020) or conversational (Anantha et al., 2021) one. Recent works have focused on explainable answers by Chain of Thought prompting (Wei et al., 2022) leveraging huge LM, similarly (Huang et al., 2022) proposed improvement of the approaches with no additional data needed. For QG, different kinds of approaches have been explored, such as the template-based approach where a pre-set of templates is filled with document information (Wolfe, 1976), the sequence-to-sequence approaches (Zi et al., 2019) or considering both (Fabbri et al., 2020). In a sequence to sequence model, additional information is usually given to guide the generation, the "question support". Extracting salient text spans is thus a key sub-task for text generation, it can be leveraged without any task prior, relying on part-of-speech extraction (Toutanvoa and Manning, 2000), dependency parsing (Surdeanu and Manning, 2010) or keyword extraction with KeyBERT (Grootendorst, 2020) using Bert embedding.

Although generation of either question or answer is getting closer from human writing, the lack of metrics still remains an issue. Generally in language generation the n-gram based approach such as BLEU (Papineni et al., 2002) or ROUGE (Lin, 2004) metrics are commonly involved. Some approaches using language model have been proposed, allowing to take into account meaning similarity instead of words similarity, such as the BERTScore (Zhang et al., 2020) based on embedding. Even if specific question generation metrics exist, as the Q-metric (Nema and Khapra, 2018), at the best of our knowledge none are as reliable as human judgment.

3. A French corpus for education

To gather a qualitative French corpus for education, we collect textbooks from middle and high school about History, Geography, Life science, and Civic Education from the "Livrescolaire"[3]. In addition, we retrieve Wikipedia articles related to this corpus. We filter them using Wikipedia API with queries based on the titles from the educational textbooks, then we bring together the subsections selected. We present to annotators paragraphs of a document and we asked them to create the following annotations:

- **A question:** written by the annotator.
- **The question type:** factual, descriptive, course or synthesis.
- **The question support(s):** extracted spans targeting the subject of the question.
- **Answer element(s):** the different passages allowing to answer the question.
- **The hand written answer:** from the annotator, using the answer elements.

Notice that for each document we asked annotators to create many annotations. We will focus here on the first three annotations. Translated examples are provided in table 1. We already launched two first annotation campaigns where we obtained 412 questions. In the following, QAE will refer to questions answers for education, with QAE-A the dataset obtained with teachers and QAE-B the dataset from a private annotation organism. In the future, the goal is to collect around 10.000 question/answer pairs in a final annotation campaign, which will provide a corpus to train deep neural network on complex question/answer generation.

4. Extracting the support to generate a question

Although a question support is available for generating questions and answers in the collected corpus, in real case this information is unavailable. In the following we define the automatic extraction of the question support.

[3]https://www.lelivrescolaire.fr/

Type	Question	Support
Factual	In which year did Christopher Columbus reach America ?	Christopher Columbus reached America (1492)
Descriptive	What is a rotary press ?	A rotary press is a typographic press mounted on a cylinder, allowing continuous printing.
Course	How did the Europeans legitimize their domination?	Europeans rethink the hierarchy of people within a Christian and European-centered scheme which then serves to legitimize their domination
Synthesis	Why did some French people support the state of emergency after the 2015 Paris attacks ?	• protects them against the terrorist threat and the risk of a new attack, which is feared by all. • This exceptional regime continues to appear as "a necessity".

Table 1: Examples for the four question types

4.1. Generate the question

In our experiments, our generative models take as input a context and a question support in order to generate a question. The context chosen is the paragraph from which a support is extracted. We make use of a special token $< hl >$ which is set around the question support. The format given is the following:

$$[pre\_context] < hl > [support] < hl > [post\_context]$$

For training, our cost function relies on minimizing the cross-entropy loss.

4.2. Extracting the question generation support

When building an automatic tool, the support for generating question is rarely given. In this study, we focus on the selection of such support by extracting specific information for each sentence of the original support extracted. For instance, given the sentence:
"After the failure of the Hungarian revolt during the Springtime of Nations, the Empire of Austria and its dynasty, the Habsburgs, reclaimed their full power"
***Original sentence**: Après l'échec de la révolution populaire hongroise lors du Printemps des peuples, l'Empire d'Autriche et sa dynastie, les Habsbourg, sont restaurés dans toute leur puissance.*
which part should be given to the model to generate the following question:
"What are the consequences of the Hungarian revolt failure during the Springtime of Nations?"
***Original sentence**: Quelle est la conséquence de l'échec de la révolution populaire hongroise?*
In the corpora available in the literature, a named entity, usually the answer, is given as the additional input in the form of a text span to guide the question generation system. However, a named entity is rarely sufficient to produce complex questions as it leads generation to only focus on a factual element. We thus try to automatically extract relevant text spans which would better guide the generation, using entities, syntactical units (such as objects of the predicate) or group of words standing together as a semantic unit (keyphrases). We studied approaches based on the following elements:

- **Source** (**SRC**): The question support selected by human annotators in our corpus, for instance: "the Empire of Austria and its dynasty, the Habsburgs, reclaimed their full power". In few cases, the selected support is not contiguous (annotators selected support in different paragraph), in this case one question by contiguous support will be generated leading to have many questions by annotation.
- **Named entities (ENT)** : A selection of named entities from any sentence overlapping with the source, for instance: "Springtime of Nations", "Empire of Austria", "Hasburgs".
- **Noun phrases (NP)** : A selection of noun phrases from any sentence overlapping with the source, for instance: "their full power", "its dynasty", "the Hungarian revolt". We did not take into account noun phrases overlapping with entities.
- **Object (OBJ)** : the object, i.e. the subtree annotated as OBJ by a dependency parser, from sentences overlapping with the source. For instance: "the Empire of Austria and its dynasty, the Habsburgs, reclaimed their full power" (here, it is the same as the source).
- **Keyphrase (KP)** : A selection of extracted "key passages" with a KeyBERT model (based on CamemBERT) from any sentence overlapping with the source. The KeyBERT model averages the embedding of the sentence and computes the cosine similarity of the contextual embeddings of text portions with this average. For each sentence we sample the top two key-phrases from 2 to 15 tokens using a diversity parameters of 0.6 (using Maximal Marginal Relevance). For contextual embeddings we used the *camenBERT-base* model[4] For the current example we obtained the following supports: "After the failure of the Hungarian revolt during the Springtime of Nations," and "reclaimed their full power"

We use the spacy [5] library with the *fr_core_news_lg* model to extract the support from the sentences. Notice that we use SRC support as default value.

5. Experimental settings

Model. In our experiments we fine-tuned three different models: **BARThez**[4] a French model designed for generative tasks having both encoder and decoder pretrained; **MBARThez**[4] model trained with objective similar to BARThez model using a multilingual setting (using the MBART architecture); **MBART**[4] model, a multilingual model trained on translation tasks in many languages. For multi-lingual approaches, we use a special token to specify the language of the input or output text. The code can be find on github[6]

Corpus. To train and validate our model we use the French datasets *Piaf* and *FQuAD*, for multilingual model the *SQuaD* datasets is additionally considered. We split the datasets to obtained the set described in table 2, and

[4]https://huggingface.co
[5]https://spacy.io/
[6]https://github.com/tgeral68/EFRQA

Corpus	Train	Validation	Evaluation
SQuAD	87599	–	–
FQuAD	20731	1641	1547
Piaf	7375	1082	767
QAE-A	–	–	252
QAE-B	–	–	182

Table 2: Number of question and answer pairs in the different corpora. SQuAD has only been used for training.

Dataset	MBART	BARThez	MBARThez
FQuAD	41.8 / 90.9	42.4 / 91.0	**45.2 / 91.5**
Piaf	38.5 / 90.5	38.3 / 90.3	**39.7 / 90.6**
QAE-A	27.5 / 89.2	28.0 / **89.3**	**28.6 / 89.3**
QAE-B	35.4 / 90.4	37.3 / 90.5	**38.4 / 90.7**

Table 3: Results for the different models and dataset. We report the RougeL / BERTScore metrics.

only use French datasets for validation. On the evaluation side, we use both our own educational corpus, QAE-A and QAE-B, and the test set of *Piaf* and *FQuAD*. For multilingual approaches, we duplicate each training corpus having the question translated (for FQuAD and Piaf the question is translated into English, for SQuAD into French) using the pre-trained MBART model.

Training. All models are fine-tuned using the same hyper-parameters: during the first 1000 iterations we linearly increase the learning rate to reach 1^{-4} (starting at 1^{-7}). The batch size is fixed to 128 samples using gradient accumulation. The context is truncated if exceeding 512 tokens. The optimizer is AdamW (Kingma and Ba, 2015); an epoch is specified to use 2000 batches and the training samples are randomly sampled. We stop the training if the RougeL value does not increase during 5 epochs (on validation set), the model with the best validation is saved and used in later experiments.

Evaluation. For the evaluation, we choose two metrics: RougeL, and BERTScore. **RougeL** [4], originally designed for machine translation, measures the number of n-grams shared between the prediction and the ground truth. We do not consider approaches based on geometrical mean like BLEU since the number of questions for each context may vary depending on the extraction approach. **BERTScore**[4] evaluates the similarity between two spans based on contextual embeddings extracted from a RoBERTa model[4] (xlm-roberta-large).

6. Results and analysis

In table 3 we report the performances of the different models for the two metrics. We observe that the MBARThez model outperforms the other models, including its monolingual counterpart. This result confirms the trend that increasing the number of parameters and exploiting the knowledge of datasets in different languages improve performance. The two models take advantage of being trained on generative tasks. Results for MBART demonstrate that translation models are not the best suited for question generation, the model being potentially biased by this initial task.

Although we did not report the side experiments where both the MBART and MBARThez models were fine-tuned on French corpora only, the validation results were lower, as could be expected. This emphasizes the improvement brought by training on datasets from different languages and demonstrate that we can augment the training sets with foreign corpora. We also note that BERTScore is less informative as values vary only within a small range and, in most cases, it follows the tendencies of the RougeL score. In the following, we will only consider RougeL and the MBARThez model.

6.1. Performances related to question support

In the following experiment, we evaluate performances based on the extracted question support. Table 4 reports the results obtained when considering the different question support. In addition to mean RougeL score, we process the mean for the best and worst rated questions, as for each source, several supports can be extracted and one question is generated per support, the mean number of supports extracted is also given. Extracting the object (**OBJ**) give the overall best (Mean in the table) generated question according to the four datasets while also maximizing the worst (Min) question generated. This extracting approach is thus reliable to generate valid questions and minimize the risk of generating poor or incorrect questions. The noun phrases (**NP**) maximize the best (Max in the table) generated questions. However, it is difficult to draw conclusions as a high number of supports were extracted for each source, and hence more questions were generated than for the other supports. Thus, extracting certain noun phrases allows us to get the questions most similar to those of the annotators. However, these results do not allow us to judge the quality of the questions, i.e. many other relevant questions can be produced for a same passage.

6.2. Human evaluation

If the previous metrics offer insights into how the models perform and can reproduce questions from the test set, we still lack a qualitative evaluation. We asked human annotators to evaluate the question quality. Each question was evaluated on a scale from 1 to 4 on three criteria: the syntax correctness, the meaning of the question, the answerability according to the context. The annotators also classified whether a question is factual or not.

Table 5 show the judgment of 5 annotators, with 30 questions each from QRE datasets. The best performances for syntax and question relevance are obtained with **OBJ** extraction, this reinforces the results of section 6.1.. However, for answerability, the **ENT** extraction achieves better results, it may be a consequence of the format of such questions (factual) and the format of the answer (unique entity). This intuition is supported by the factuality ratio (FACT), where only few (10%) questions are classified as factual. On the contrary, less factual questions are obtained through

Dataset	Sup	Mean	Max	Min	N
fquad	ENT	32.5	38.8	26.9	2.3
	NP	28.8	**45.7**	16.2	7.1
	KP	30.7	37.7	23.7	2.0
	OBJ	**33.6**	36.9	**30.6**	1.7
piaf	ENT	29.9	35.3	25.1	2.4
	NP	26.7	**40.9**	16.0	6.4
	KP	28.0	34.2	21.8	2.0
	OBJ	**31.5**	34.5	**28.7**	1.6
QAE-A	ENT	**25.9**	30.4	21.8	2.5
	NP	24.1	**38.1**	13.3	8.5
	KP	25.7	34.4	17.4	3.4
	OBJ	**25.9**	29.6	**22.8**	2.0
QAE-B	ENT	28.2	33.2	24.2	2.5
	NP	28.8	**43.6**	16.9	8.7
	KP	31.3	41.1	23.0	3.9
	OBJ	**32.9**	37.7	**28.9**	2.3

Table 4: MBARThez results for the different extraction (see section 4.2.). RougeL is reported for the average (**Mean**), maximum (**Max**) and minimum (**Min**) performance of each question grouped by source, with *N* indicating its mean number.

	COR	REL	ANS	FAC
SRC	3.63	3.17	3.17	60%/30%
ENT	3.53	2.73	**3.47**	83%/10%
NP	3.40	2.47	2.87	83%/ 7%
KP	3.60	2.60	2.83	83%/16%
OBJ	**3.67**	**2.87**	3.23	70%/**20%**
Total	3.57	2.77	3.11	75%/16%

Table 5: Human evaluation for the QRE-A & QRE-B datasets (on a scale from 1 to 4) with **COR** the syntax correctness, **REL** the question relevance, **ANS** the answerability and, **FAC** if the question is factual or not.

the **OBJ** extraction approach, which enhance its relevance to generate more complex questions.

6.3. Performances according to question types

We show in table 6 the *RougeL* performances for the generated question from **SRC** support and **OBJ** support (as best performances are reached considering this last extraction approach). As a reminder, the course (COUR) and synthesis (SYNT) questions are usually more sophisticated, and thus harder to generate. The synthesis questions are particularly challenging since they often rely on different passages of the text. On the contrary, vocabulary questions are much easier to generate: firstly, because the associated text mostly relies on an explicit definition in the context, which limits the possibles questions, e.g. "Discrimination: treating someone differently because of their origin, skin color, religion, gender or sexual orientation, political or trade union orientation."; secondly, the questions are often simple and have few templates available, e.g. "What is discrimination?". Although we obtain the best performances

Dataset	QType	Mean	Max	Min	N
SRC (manually annotated support)					
QAE-A	FACT	26.2	"	"	1.0
	VOCA	**57.6**	"	"	1.0
	COUR	26.0	"	"	1.0
	SYNT	24.2	"	"	1.0
QAE-B	FACT	53.0	"	"	1.0
	VOCA	**53.7**	"	"	1.0
	COUR	35.0	35.5	34.4	1.1
	SYNT	19.3	23.9	15.3	2.4
OBJ (support based on sentence object extraction)					
QAE-A	FACT	22.1	23.5	20.6	1.5
	VOCA	**52.8**	**56.0**	**49.6**	1.4
	COUR	24.6	28.8	21.3	1.9
	SYNT	21.7	25.7	18.2	2.3
QAE-B	FACT	36.0	38.8	33.2	1.3
	VOCA	**48.1**	**51.9**	**44.6**	1.6
	COUR	33.6	36.8	30.3	1.6
	SYNT	18.3	27.0	12.3	4.2

Table 6: Results based on the different question type for support based on object and source (*RougeL* score).

within the manually annotated support (Mean column) on *RougeL* score, the results obtained considering the object of the sentence are not far behind, thus it remains a good candidate for generating this kind of questions. Interestingly, we observe better scores for **OBJ** based generation when looking at the average of maximum performances (Max column) for both courses and synthesis questions. We thus, empirically demonstrate that we can generate questions closer to the manually created ones and less factual using extraction of object sentences.

7. Conclusion

We address the question generation task to generate teacher-like questions. To this end, we developed different algorithms and evaluated them on a new French corpus focused on educational question generation. Furthermore, we studied the impact of many support extraction methods in order to develop a reliable automatic question generation system for education. Then we spotlight the performances according the type and complexity of the question. We empirically demonstrate that it remains possible to improve the quality of questions using corpora in multiple languages (section 6.). Then, we compared different extraction methods to get the generation support. Even if human-extracted passages led to the best performances, we show that extracting sentence objects is a good candidate to automatically generate questions. Our human evaluation emphasized the relevance of object extraction, showing its ability to generate non factual question. Finally, we reported performances based on question types, showing that the difficulty of the questions fits our intuition, where reasoning questions are difficult to reproduce according to the current model training protocol and configuration. It is still

difficult to automatically state the quality of the questions; we evaluate each generated question according to its likelihood with the annotated question which does not express how good it is. In future work, we plan to explore evaluation metrics to better judge the quality of the generated questions. Furthermore, for complex questions, the answer may rely on several passages, the approaches designed here only take into account a single span for generating questions. This point will be addressed once the amount of collected data allows us to fine-tune transformers models (annotation of 10.000 questions-answers pairs is planned). Last but not least, according to the long-term objectives, the answer extraction/generation are foreseen, particularly to produce answers within an explanation scheme.

References

Anantha, Raviteja, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi, 2021. Open-domain question answering goes conversational via question rewriting. In *NAACL-HLT*. Association for Computational Linguistics.

Antoine, Elie, Jeremy Auguste, Frédéric Béchet, and Géraldine Damnati, 2022. Génération de questions à partir d'analyse sémantique pour l'adaptation non supervisée de modèles de compréhension de documents. *29e conférence sur le Traitement Automatique des Langues Naturelles (TALN)*.

Bechet, Frederic, Cindy Aloui, Delphine Charlet, Geraldine Damnati, Johannes Heinecke, Alexis Nasr, and Frederic Herledan, 2019. CALOR-QUEST : un corpus d'entraînement et d'évaluation pour la compréhension automatique de textes. In *Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN) PFIA 2019*.

Béchet, Frédéric, Ludivine Robert, Lina Rojas-Barahona, and Géraldine Damnati, 2022. Calor-Dial : a corpus for Conversational Question Answering on French encyclopedic documents. In *CIRCLE (Joint Conference of the Information Retrieval Communities in Europe)*. Samatan, France.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei, 2020. Language models are few-shot learners. *CoRR*.

Choi, Eunsol, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer, 2018. Quac: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Eddine, Moussa Kamal, Antoine J.-P. Tixier, and Michalis Vazirgiannis, 2021. Barthez: a skilled pretrained french sequence-to-sequence model. In *EMNLP (1)*.

Elgohary, Ahmed, Denis Peskov, and Jordan L. Boyd-Graber, 2019. Can you unpack that? learning to rewrite questions-in-context. In *EMNLP-IJCNLP*. Association for Computational Linguistics.

Fabbri, Alexander R., Patrick Ng, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang, 2020. Template-based question generation from retrieved sentences for improved unsupervised question answering. Association for Computational Linguistics.

Grootendorst, Maarten, 2020. Keybert: Minimal keyword extraction with bert.

Huang, Jiaxin, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han, 2022. Large language models can self-improve.

Karpukhin, Vladimir, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih, 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*. Association for Computational Linguistics.

Keraron, Rachel, Guillaume Lancrenon, Mathilde Bras, Frédéric Allary, Gilles Moyse, Thomas Scialom, Edmundo-Pavel Soriano-Morales, and Jacopo Staiano, 2020. Project piaf: Building a native french question-answering dataset. In *Proceedings of The 12th Language Resources and Evaluation Conference*.

Khattab, Omar and Matei Zaharia, 2020. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *SIGIR*. ACM.

Kingma, Diederik P. and Jimmy Ba, 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Kwiatkowski, Tom, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov, 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*.

Le, Hang, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab, 2020. Flaubert: Unsupervised language model pre-training for french. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020*.

Lin, Chin-Yew, 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics.

Liu, Xiaodong, Yelong Shen, Kevin Duh, and Jianfeng Gao, 2018. Stochastic answer networks for machine reading comprehension. In *ACL (1)*. Association for Computational Linguistics.

Liu, Yinhan, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke

Zettlemoyer, 2020. Multilingual denoising pre-training for neural machine translation.

Martin, d'Hoffschmidt, Vidal Maxime, Belblidia Wacim, and Brendlé Tom, 2020. FQuAD: French Question Answering Dataset. *arXiv e-prints*.

Martin, Louis, Benjamin Müller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot, 2020. Camembert: a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*.

Nema, Preksha and Mitesh M. Khapra, 2018. Towards a better metric for evaluating question generation systems. In *EMNLP*. Association for Computational Linguistics.

Nguyen, Tri, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng, 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@NIPS*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu, 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al., 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu, 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*.

Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang, 2016. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*. The Association for Computational Linguistics.

Surdeanu, Mihai and Christopher D. Manning, 2010. Ensemble models for dependency parsing: Cheap and good? In *HLT-NAACL*. The Association for Computational Linguistics.

Toutanvoa, Kristina and Christopher D. Manning, 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *EMNLP*. Association for Computational Linguistics.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*.

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou, 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

Wolfe, John H., 1976. Automatic question generation from text - an aid to independent study.

Yamada, Ikuya, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto, 2020. LUKE: deep contextualized entity representations with entity-aware self-attention. In *EMNLP (1)*. Association for Computational Linguistics.

Zhang, Tianyi, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi, 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Zhang, Zhuosheng, Junjie Yang, and Hai Zhao, 2021. Retrospective reader for machine reading comprehension. In *AAAI*. AAAI Press.

Zi, Kangli, Xingwu Sun, Yanan Cao, Shi Wang, Xiaoming Feng, Zhaobo Ma, and Cungen Cao, 2019. Answer-focused and position-aware neural network for transfer learning in question generation. In *KSEM (2)*, Lecture Notes in Computer Science.

A Voice-Based Neural Network System for Accessing Embedded Home Automation Devices

Monika Grajzer[1], Mikołaj Pabiszczak[1], Agnieszka Betkowska Cavalcante[1], Michał Raszewski[2]

[1]Gido Labs sp. z o.o.
m.grajzer@gidolabs.eu, m.pabiszczak@gidolabs.eu, a.b.cavalcante@gidolabs.eu
[2]LARS Andrzej Szymański
michal.raszewski@lars.pl

Abstract

In this article we present a custom system for controlling access to embedded, offline home automation devices. The core of the system is based on the Keyword Spotting technology. We propose and evaluate 2 ResNet neural networks suitable to realize such a system for recognizing a custom keyword spoken in Polish. In addition, we present the accompanying modules which enable False Positive Rate reduction to address the challenges imposed by practical applications. The proposed solution was implemented on an embedded device and tested in realistic field trials in order to prove its computational efficiency and ability to constitute an Access Control system for the commercial home automation device.

Keywords: keyword spotting, home automation, Polish language

1. Introduction

Recently we can observe an increasing popularity of home automation systems controlled by voice (Baby et al., 2017). The core element of such systems is an Automatic Speech Recognition (ASR) along with a dedicated Dialogue System allowing for giving voice-based control prompts in a natural or semi-natural language. These technologies, however, may use a significant amount of computational and battery resources. Many of the home automation systems, such as Amazon Alexa or Google Home, operate in a set-up, where those resource-consuming techniques are implemented in the cloud. Hence, they require the audio recordings to be sent to the server which leads to the risk of revealing them by the third parties (Cuthbertson, 2019). As a result, there is a need for solutions targeting strictly offline environments which enable to fully support end-users privacy demands. These are typically implemented on small, custom-made embedded devices, where battery efficiency and resource demands become of paramount importance.

In this paper we target offline home automation solutions. Due to their limitations, its necessary to equip them with a front-side mechanism of Access Control (AC) in order not to perform ASR continuously, but rather rely on the method that activates this technology only when needed (Kaklauskas, 2015).

The key components of such an access system are Voice Activity Detection (VAD), which detects when speech occurs in an audio signal and switches the device to the listening mode, as well as Keyword Spotting (KWS) module which aims at detecting, if the particular keyword phrase has been spoken (such as, e.g., "Hey, Siri"). Other methods of Access Control for home automation devices may include also Speaker Recognition (SR) technology, which is based on voice biometrics and allows only authorized users to access the system (Kaklauskas, 2015, Pabiszczak et al., 2020).

Developing such an AC solution is challenging, not only due to the scarcity of computational power and battery resources, but also due to a strict demand for a very low False Positives Rate (FPR) – so that the number of false alarms is limited. Especially in the case of the KWS module, even though the most recent research results are very promising, the reported FPR levels are still not sufficient for commercial application when applied without any supporting intelligence (Sainath and Parada, 2015, Tang and Lin, 2018, Lengerich and Hannun, 2016, Pabiszczak et al., 2020).

To address the above requirements, we present a voice-based AC system based on neural network (NN) technology. The solution allows to efficiently make decisions on granting access to the home automation system and its ASR engine. The core Decision Support System constituting this technology was presented in (Pabiszczak et al., 2020), where the wider context was described, including the Speaker Recognition module. Since a core part of the proposed Access Control system is the KWS module and the related inference engine, in this paper, we primarily present our research on these technologies, focusing on the design of a neural network suitable for recognising a custom keyword from a non-English language. Moreover, we also target the computationally lightweight procedures that were proposed by us to address the problem of high FPR. The presented results depict the performance of the KWS system achieved for different configuration and training procedures as well as the decrease in FPR achieved with the additional AC mechanisms. The experiments were conducted in the real-life scenarios on the custom-made embedded device built for the needs of home automation.

The rest of the paper is organised as follows: in Section 2, we present a short overview of related work, followed by the description of the proposed Access Control system in Section 3. In Section 4, we discuss in detail the process of

Contents

creating a KWS system for a custom keyword and evaluate the performance of the proposed AC mechanism. Finally, we present closing conclusions in Section 5.

2. Related Work

In the design of user interfaces for the voice-controlled home automation systems, access control methods are one of the key components required by a final system envisioned for practical implementation (Sidhartha et al., 2014, Lee et al., 2011, Team, 2017). As the most straightforward solution, they are composed of VAD/KWS module (Tang and Lin, 2018), however, the KWS system trained on examples including silence and background noise, can also play the role of a VAD solution. This makes the KWS module a core part of the Access Control technologies. Prior solutions for KWS systems where often relying on the HMM-based ASR – one example of such a set-up being the pocketsphinx software, which can be set-up in a keyword spotting mode instead of a full continuous speech recognition mode. However, the most recent methods focus on applying neural networks – with particular interest in the Convolutional Neural Networks (CNNs), which require less computational power than Recurrent Neural Networks (RNNs) typically used in the case of speech recognition (Zhang et al., 2016, Amodei et al., 2016, Sainath and Parada, 2015).

In this context, particularly promising are Residual Neural Networks architectures (ResNet) with skip connections between blocks of selected layers (Tang and Lin, 2018, Xie et al.,) – they are characterised by a lower complexity and faster training phase and were proved to obtain very good performance even for relatively small- sized networks – reaching the accuracy of 95% (Tang and Lin, 2018). Substantially, this accuracy was obtained for a model trained on a large database of examples. It constitutes a significant challenge to create a ResNet model that would recognize custom keyword with a limited number of keyword recordings.

In addition, while performance results reported in the literature are very good, they are far from being industrially applicable: assuming that FPR is 2% (Tang and Lin, 2018) and the system makes a prediction every second, there will be 72 false alarms in one hour — a number unacceptable for practical deployments (Pabiszczak et al., 2020). Some methods to address this problem incorporate "push to talk" (Lee et al., 2011), active audio loudness estimation (Sidhartha et al., 2014) or application of advanced features – e.g., where KWS is followed by additional reasoning using HMMs and re-checking in the cloud (Team, 2017). The latter kind of solutions are often very complex and likely too resource-consuming for small embedded devices (Pabiszczak et al., 2020).

3. System Architecture

We are considering an AC system for embedded home automation devices which is working locally on the device without any access to the Internet and cloud-based resources.

In the envisioned design, the home automation device is activated by pronouncing the specified keyword – after the AC module has decided to grant an access, a signal is generated (buzz) and the user may start making prompts to the core ASR-based dialogue system of this device (Cavalcante et al., 2022).

In a basic approach, the AC module would include only the VAD and KWS blocks. In our design, we introduce also a speaker recognition block (Cavalcante et al., 2022) and additional modules that allow to decrease the FPR of the entire Access Control solution, without a significant loss in its TPR. Those additional modules include (Pabiszczak et al., 2020):

- "Loudness Checker" – requires the incoming audio to have a certain minimum level of loudness
- "Timer" – limits the time duration of an audio buffer that is being processed; if the triggering word is not observed shortly after that, it is assumed that the audio resulted from some background noise or other conversational sounds, music, radio, etc. and does not need to be processed further
- "Score Smoothing" – performs the smoothing of the results obtained from KWS; it is the key intelligence engine that is making a final decision on whether the keyword was properly registered

Below we present a more detailed overview of the proposed elements.

3.1. Keyword Spotting module

The KWS system is based on the neural network with ResNet architecture "res8" presented in (Tang and Lin, 2018), , which is characterised by a small number of network parameters (approx. 110K). The output of the last residual block in this architecture is fed into an average pooling layer, flattened, and fed into dense layer of size corresponding to the number of labels that can be recognised (Pabiszczak et al., 2020). Input audio signal of the envisioned length of 1s is preprocessed, before being fed to the ResNet, by computing 40 Mel-Frequency Cepstral Coefficients (MFCC) acoustic features. The network described in (Tang and Lin, 2018) was originally trained on the Google Speech Commands Dataset (GSCD) for recognizing 10 keywords from this dataset and, in addition, the labels of "silence" and "unknown". Network output is a vector of numerical scores corresponding to each label. The system was trained for the English keywords. Our solution is targeting Polish language speakers and the selected keyword is a product's trade name. Hence, the above-mentioned solution cannot be used directly for our goal. In this situation, the challenge was to produce a well-performing classifier having available only limited database of related audio samples containing custom keyword. To address this issue, we have used a transfer learning procedure (Yosinski et al., 2014) in which a reference "res8" network (Tang and Lin, 2018) was used as a "feature extractor" and only its last layer (i.e., the classifier) was re-trained with a smaller, problem-specific database.

3.2. Training database

In order to train the classifier layer of our solution with 2 outputs (keyword vs. non-keyword), we have gathered a dataset of 698 positive examples of the selected keyword from 36 people. The negative examples were extracted from a larger database of phonetically representative Polish speech (containing the recordings of 86 people), which has been collected by us by means of the mPASS platform (Cavalcante et al., 2016). The recordings were adjusted to the length of 1s. In the training phase, the examples were augmented with background noises of variable loudness, originating from the GSCD dataset – with the probability of 10%. These background noises were also used to create negative examples of silence. In addition, for the augmentation of negative samples, the time shift of 100ms (randomly to the left or right) was applied. This procedure was not performed for the positive samples to ensure that the entire audible part of recording is kept intact.

3.3. Alternative design

The originally proposed "res-8" architecture was used with the MFCC audio features at the preprocessing stage. However, filterbank (FB) features are known to have smaller computational complexity and are widely applied for audio-related tasks. Therefore, they were intersting candidates from the perspective of practical application on an embedded device. As such, we've created a second version of the KWS system using FB features with the same ResNet architecture as depicted above. The process covered training a new set-up on the GSCD for the recognition of 12 labels and performing transfer learning with the smaller keyword-specific dataset.

In order to enable FPR reduction, there are 3 new modules introduces in the AC system. The Loudness Checking module constantly reads the audio input from the microphone and checks if its mean amplitude exceeds the specified threshold. Only in such a case the signal is fed for further processing. This threshold was set experimentally, as it depends on: 1) the particular microphone used, 2) the format of the audio encoding, and 3) a microphone driver (Pabiszczak et al., 2020).

For the system working in real-time, a sliding window of 1s is used on the audio input, which moves by 0.2s. Since the duration of the desired keyword is typically not longer then 1s, the Timer module limits the audio input being processed by the KWS model to the first 1.2s of the signal (the lacking part of the 1s sliding window in the beginning of listening is filled with zeros). With the overlapping between consecutive frames of 80%, this gives the input to the KWS module of maximum 7 frames, which constitutes an attention window of size 7 (Pabiszczak et al., 2020).

The third module, Score Smoothing, is placed after the NN-based keyword spotter and contains the final evaluation engine. It implements an additional logic introduced to extract the most meaningful knowledge from the KWS module output. Based on our observations, the FPR is oo high if an access is granted after observing only a single frame containing the keyword. Therefore, the Score Smoothing module aims at smoothing the scores of the KWS classifier. Based on the observations from the initial field trials, our approach for this block is based on calculating the mean of the last n predictions – in the beginning of listening, when less number of predictions is available, the lacking results are assumed to have score 0. This way, a single, strong trigger could still be considered as a positive activation (Pabiszczak et al., 2020). Based on the evaluation described in more detail in (Pabiszczak et al., 2020), this value has been set to 3 for best performance. The Score Smoothing module makes a decision based on the observed mean value – once it exceeds the threshold set for KWS, the input audio signal is passed for further processing by the SR module.

4. Evaluation

In this chapter we present evaluation results related to the consecutive components of the Access Control System as well as to the overal performance of the final solution in the real-world set-up. The goal was to come up with a system, capable of running on an embedded device, that would offer good accuracy along with a possibly minimal FPR measures.

4.1. KWS model

The evaluation has been performed in a laboratory environment on the dataset depicted in Section 3.1.1 with the aim to come up with model parameters and set-up which would best address our objectives depicted above. 10-fold cross validation (CV) procedure was applied during the transfer learning. For each CV division in each training epoch the resulting model was converted to the tensorflow lite version which allows for better performance on the small-scale devices. The impact of the conversion to the lite version on the performance should be negligible. In addition, we've been also experimenting with weight quantization, which yields smaller model sizes and further reduction in computational complexity. This is, however, achieved on the cost of decreased performance. During the experiments we also wanted to evaluate this imapct, in particular on the FPR. In each validation division, the Equal Error Rate (EER) point has been estimated for each epoch (on the validation set) and for further evaluation an epoch was chosen with the smallest EER. This resulted in 2 models (lite and quantized) per each CV division. In the next step the performance of those models (in terms of EER) was evaluated on the test set. The above procedure has been performed for both investigated KWS systems: using MFCC or FB features. The averaged results over 10 divisions are depicted in Table 1.

It can be observed that both MFCC-based models perform significantly better then FB models. Moreover, the relatively high standard deviation values for FB models suggests potential instability of this model type, which was confirmed by further observations. In addition, for the MFCC-based model, the performance (in terms of EER) drops by approximately 3p.p. for the quantized version – the difference being significant in the context of our objectives.

	MFCC (l)	*MFCC (q)*	*FB (l)*	*FB (q)*
Average	5.74	8.69	10.50	10.42
Std. dev.	2.00	2.37	7.68	8.27

Table 1: Averaged EER values over the results of EER estimation on the test set in 10-fold cross validation for models with MFCC and FB features generated in 2 versions: *lite* (l) and *quantized* (q).

ERR	*MFCC (l)*	*MFCC (q)*	*FB (l)*	*FB (q)*
set 1	2.77	9.18	1.54	1.69
set 2	3.45	5.36	5.36	4.31
set 3	**5.29**	**10.02**	**3.48**	**3.44**

Table 2: EER values evaluated on the validation set (1), test set (2) and combined validation and test set together with additional recordings (3), obtained for the final NN models with MFCC and FB features generated in 2 versions: *lite* (l) and *quantized* (q).

Following the standard approach, the final "production" models for each of the 4 cases should be selected as the ones with the highest EER on the test set, among all the models generated for transfer learning in cross-validation experiments. However, in case of models using FB features this would give us a case where the EER on the test set was 3.3%, while the EER on the validation set exceeded 25% (for lite version, with similar results in case of the quantised one). These values suggest a very instable model. Hence, we've followed a heuristic approach and have selected the model which performs well in both cases. The 4 selected "production" models are depicted in Table 2.

Typically, the final step would require to re-train the selected models with a low learning rate on the remaining recordings, but due to a small database size, we've been observing instability during this process. Hence, we've left selected models intact, but have chosen the KWS system operation point (i.e. the decision threshold for NN output corresponding to the EER value) from the EER, which was computed on the combined test set, validation set and additional recordings of 5 people which ware not present in the initial dataset. The final EER evaluation results are presented in the last column in Table 2. They depict slightly better performance for the models with FB features, however the lite model with MFCC features is also having good results. On the contrary, its quantized version should be rejected due to a significant performance drop. However, for a full overview and in order to make potentially best design decision from the perspective of practical evaluation, we should also take into account results presented in Table 1. They reveal potentially high instability of the FB models, while the performance of MFCC-based models is not far from the best results obtained for FB models and seem to be stable across different observations. Hence, for the final system design, we've selected the MFCC-based lite NN model. It's performance is comparable to the one presented in (Tang and Lin, 2018), which is a good result baring in mind a very small dataset of examples used to create our system.

4.2. Field trials on an embedded device

Although the results presented in Section 4.1. are good from the scientific perspective, they are still not enough to allow for a practical implementation of the envisioned Access Control system. Therefore, we have proposed additional modules in Section 3.2. and have evaluated the overall system performance in the field trials on the physical embedded device. We performed two tests in diversified conditions.

The entire AC system was implemented on a RaspberryPi 3B platform (CPU: 1,2 GHz quad-core ARM-8 Cortex-A53 (64-bit); 1 GB RAM). The device was equipped with a custom-made microphone matrix with 5 independent microphones.

In the firs test, we have been evaluating the system performance as number of false positive activations caused by the background voices, which may occur in the household. During these trials, 18min and 14s-long audio of varying loudness was analysed. Radio conversations were chosen as the audio source, containing the voices of various people. The activation keyword was not present in the audio recordings. The AC system was processing this audio and the number of false alarms was counted. FPR was calculated as the ratio of the number of these false alarms to the number of all analysed audio frames (5471 in total) (Pabiszczak et al., 2020).

For a reference system containing only the KWS module described in the preceding sections, FPR of 2.23% was obtained (122 unwanted activations per 5471 analysed frames). Adding a Loudness Checking module allowed to further reduce FPR 2.5 times, while using this module together with a Timer resulted in the FPR of 0.64%. After all additional modules ware combined (together with Score Smoother) into one processing pipeline, the number of false alarms was reduced to 0 (Pabiszczak et al., 2020).

The aim of the second test was to identify the influence of the selected designs on the TPR and false alarms in a challenging task, where speech samples included words that are phonetically similar to the keyword or are household-related. Among the 3 new modules, the Score Smoother is the one that can potentially have negative impact on the TPR measures (Pabiszczak et al., 2020), since Loudness Checking and Timer modules mainly reduce the computational overhead and by limiting the amount of data processed by the KWS module, limit also the number of false alarms.

In this trial the audio signals were recorded live from 13 users (both female and male). For each person, the test consisted of uttering the keyword 30 times and uttering 10 other words 3 times each. Approximately 20% of them were phonetically similar to the keyword, which makes this trial particularly challenging. For each utterance, the binary result assigned by the Access Control system at the Score Smoother output was recorded (Cavalcante et al., 2022).

The results are presented in Table 3. Incorporating additional modules resulted in a decrease of FPR by approx.

Design	TPR [%]	FPR [%]	Acc. [%]
Only KWS	90.77	5.90	92.44
Full AC system	**86.41**	**4.87**	**90.77**

Table 3: TPR, FPR and accuracy in 2 system set-ups.

1p.p. The overall performance exceeding 90% in terms of accuarcy in the challanging live test with a significant number of "difficult" samples gives promising results in the context of future application in the home automation device. Moreover, in the final design, the AC system was also implemented by us in the same way with the same parameters on the custom-made device with STM32MP157 microprocessor (ARM-7, 2 cores at 650 MHz, 1GB RAM). In such a configuration, it was still performing in real-time allowing for practical system usage. The detailed evaluation on that machine has been presented in (Cavalcante et al., 2022).

5. Conclusion

We have proposed an Access Control system, which grants access to the voice-controlled home automation de- vices with the aim to decrease the FPR and allow for prac- tical system realisation as a part of the home automation device which is continuously processing the collected au- dio signals. For this purpose, 2 KWS system architec- tures have been evaluated and 4 different ResNet mod- els ware trained in a transfer learning procedure to allow for a recognition of a custom keyword. In addition, we have introduced 3 additional modules for the purpose of increasing system performance in real-life scenarios. The performed evaluation enabled to assess these candidate so- lutions and select the variant, which would be the most suitable for addressing challenges related to practical de- velopment. The selected KWS design allowed to achieve EER at the level of 5.29set-up resulted in highly decreased number of false sys- tem activations under realistic conditions, while retaining acceptable TPR and keeping overall accuracy above 90As a result, with the proposed computationally lightweight modifications, we have come up with an Access Control system that is commercially applicable.

Acknowledgement

The research was supported by the National Centre for Research and Development in Poland under the grant no. POIR.01.01.01-00-0044/17

References

Amodei, Dario, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, Jie Chen, Jingdong Chen, Zhijie Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Ke Ding, Niandong Du, Erich Elsen, Jesse Engel, Weiwei Fang, Linxi Fan, Christopher Fougner, Liang Gao, Caixia Gong, Awni Hannun, Tony Han, Lappi Vaino, Bing Jiang, Cai Ju, Billy Jun, Patrick LeGresley, Libby Lin, Junjie Liu, Yang Liu, Weigao Li, Xiangang Li, Dongpeng Ma, Sharan Narang, Andrew Ng, Sherjil Ozair, Yiping Peng, Ryan Prenger, Sheng Qian, Zongfeng Quan, Jonathan Raiman, Vinay Rao, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Kavya Srinet, Anuroop Sriram, Haiyuan Tang, Liliang Tang, Chong Wang, Jidong Wang, Kaifu Wang, Yi Wang, Zhijian Wang, Zhiqian Wang, Shuang Wu, Likai Wei, Bo Xiao, Wen Xie, Yan Xie, Dani Yogatama, Bin Yuan, Jun Zhan, and Zhenyao and Zhu, 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *Proc of the the 33rd International Conference on Machine Learning (ICML)*, volume 48. JMLR.org.

Baby, C. J., F.A. Khan, and J. N. Swathi, 2017. Home automation using web application and speech recognition. In *Proc. of International conference on Microelectronic Devices, Circuits and Systems (ICMDCS)*.

Cavalcante, A. B., M. Grajzer, and M. Raszewski, 2022. Decision support system for controlling home automation appliances with resource constraints. IARIA. ISSN: 2308-4375.

Cavalcante, Agnieszka Betkowska, , and Monika Grajzer, 2016. Proof-of-concept evaluation of the mobile and personal speech assistant for the recognition of disordered speech. *International Journal on Advances in Intelligent Systems*, 9(589).

Cuthbertson, A., 2019. Google defends listening to private conversations on google home: But what intimate moments are recorded? `https://www.independent.co.uk/tech/google-home-recordings-listen-privacy-amazon-alexa-hack-a9002096.html`.

Kaklauskas, A., 2015. *Intelligent Decision Support Systems*. Springer International Publishing, page 31–85. ISBN: 978-3-319-13659-2.

Lee, K. A., A. Larcher, B. Thai, Ma B., and Li H., 2011. Joint application of speech and speaker recognition for automation and security in smart home. In *Proc. of 12th Annual Conference of the Inter- national Speech Communication Association (INTERSPEECH)*. Florence, Italy.

Lengerich, Chris and Awni Hannun, 2016. An end-to-end architecture for keyword spotting and voice activity detection. ArXiv:1611.09405.

Pabiszczak, M., M. Grajzer, and L. Sawicki, 2020. Access control method for the offline home automation system. In *Proc of the Twelfth International Conference on Information, Process, and Knowledge Management (eKNOW)*.. IARIA.

Sainath, T. N. and C. Parada, 2015. Convolutional neural networks for small-footprint keyword spotting.

Sidhartha, Ch., S. Siddharth, S. S. Narayanan, and J. H. Prasath, 2014. Voice activated home automation system. In *Proc. of National Conference on Man Machine Interaction*.

Tang, R. and J. Lin, 2018. Deep residual learning for small-footprint keyword spotting. IEEE.

Team, Siri, 2017. Hey siri: An on-device dnn-powered voice trigger for apple's personal assistant. `https:`

//machinelearning.apple.com/research/hey-siri.

Xie, Weidi, Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Utterance-level aggregation for speaker recognition in the wild.

Yosinski, Jason, Jeff Clune, Yoshua Bengio, and Hod Lipson, 2014. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Zhang, Ying, Mohammad Pezeshki, Philémon Brakel, Saizheng Zhang, César Laurent, Yoshua Bengio, and Aaron Courville, 2016. Towards end-to-end speech recognition with deep convolutional neural networks. In *Proc. of INTERSPEECH*.

Development of Japanese WSC273 Winograd Schema Challenge Dataset and Comparison between Japanese and English BERT Baselines

[1]Ryo Hashimoto, [2]Masashi Takeshita, [3]Rafal Rzepka, [3]Kenji Araki

[1]School of Engineering, Hokkaido University
[2]Graduate School of Information Science and Technology, Hokkaido University
[3]Faculty of Information Science and Technology, Hokkaido University
{Ryo79676, takeshita.masashi.68}@gmail.com
{rzepka, araki}@ist.hokudai.ac.jp

Abstract

Winograd Schema Challenge (Levesque et al., 2012) has become a popular benchmark for natural language understanding and commonsense reasoning research for English laguage, and there are many related studies, but few studies have dealt with a similar benchmark in Japanese. In this study, we use the Winograd Schema Challenge dataset (WSC273) translated into Japanese (WSC273-ja) and propose a method to improve the commonsense reasoning ability of Japanese language models. After conducting the same experiment in English and comparing the results, we discuss the differences in commonsense reasoning ability between Japanese and English language models. Specifically, we first provide a baseline by evaluating a pre-trained language model using WSC273-ja in a zero-shot test. Next, we fine-tune the language models using WSCR-ja, a Japanese dataset of pronoun disambiguation problems similar to WSC273-ja. This WSCR-ja is a simpler but larger dataset than WSC273-ja. The model is then tested to see how well it can correctly answer the original WSC273-ja, which consists of more complex questions. The results are used to analyze the differences and similarities between questions with low correct response rates in Japanese and English, as well as to identify future issues to be addressed.

Keywords: Winograd Schema Challenge, Dataset creation, Japanese language

1. Introduction

In recent years, the emergence of pre-trained models with self-supervised learning on large-scale unlabelled datasets has dramatically improved model performance and achieved remarkable results in a variety of machine learning tasks. In the field of natural language processing, many pre-trained language models, including GPT (Radford et al., 2018) and BERT (Devlin et al., 2018), have emerged and continued to update the state-of-the-art in various tasks over the past few years. For example, in WSC273, a commonsense reasoning task, the accuracy was around 60% before the pre-trained language models were introduced, but in the current state-of-the-art (Sakaguchi et al., 2021), the accuracy is over 90%, which is almost the same level as that of humans. However, research on this topic is most often limited to English, because the lack of a complete Japanese WSC273 dataset, no study using WSC in this language has been conducted. In this paper, we evaluate the commonsense reasoning ability of the Japanese language BERT model by constructing a baseline using WSC273 translated into Japanese (hereafter referred to as WSC273-ja) and verifying whether the model can achieve higher accuracy on WSC273-ja by fine-tuning with a similar dataset in Japanese. As a result, it was confirmed that fine-tuning with similar data is also effective in the Japanese model. The structure of this experiment is described below. First, we evaluate the commonsense reasoning ability of BERT, which was pre-trained on a Japanese corpus, using WSC273-ja in a zero-shot test. Next, the same model is fine-tuned using the Japanese translation of WSCR (Shibata et al., 2015) (hereafter referred to as WSCR-ja), and we test whether the scores improved when the original WSC273-ja is used for evaluation. The results show that Japanese scores are lower than English scores in the same experiment, both at baseline and after fine-tuning. At baseline, the accuracy in Japanese is 0.567, while in English it is 0.619. In the Japanese language model, the accuracy between baseline and after fine-tuning is improved from 0.567 to 0.578. These results confirm that fine-tuning with similar dataset is effective in improving commonsense reasoning ability of the Japanese BERT.

2. Background

2.1. Winograd Schema Challenge

The Winograd Schema Challenge is a pronoun disambiguation problem proposed by Levesque et al. (2012) as a more practical alternative to the Turing Test. The Turing Test is a method using dialogue system, but its high adaptability and flexibility made it possible to deceive the evaluator through various deceptions and tricks. To eliminate these undesirable effects, it is more effective to impose a task that is easy for humans but difficult for machines. The specific task of the Winograd Schema Challenge (WSC) is a reference resolution, in which two sentences, each with a few words different, are paired as shown in **Ex 1** below. Each of these sentences contains an anaphora and two candidate antecedents, and the system is asked to correctly identify the antecedent to which the anaphora corresponds. These questions are designed in such a way that human can easily identify the antecedent, but are difficult for systems using only selectional restrictions or statistical methods to answer correctly.

Ex 1. *John couldn't see the stage with Billy in front of him because he is so [short/tall].*
Who is so [short/tall]?
Answers: John/Billy.

Specifically, the WS must satisfy the following requirements. (Levesque et al., 2012)

1. Containing two antecedent candidates in a sentence
2. A pronoun or possessive adjective is used as the referent for one of the antecedents, but it is grammatically appropriate to use it for the other antecedent
3. The question is to determine the referent of the antecedent
4. It contains a word called, "special word", which, when replaced by another word called "alternate word", changes the antecedent to which the antecedent refers, although the sentence still makes sense.

In the case of **Ex 1**, the special word is "short" and the alternate word is "tall", and in each case, the antecedent pointed to by the anaphora "he" changes between "John" and "Billy". The fourth requirement makes the two sentence pairs in WSC statistically very similar to each other in terms of context, but the correct antecedents are different from each other. This prevents the system from immediately exploiting statistical features even if it has access to a large corpus. The key point of WSC is that it is extremely unlikely that there are statistical or other features of special or alternate words that can be reversed from one answer to another. In a WSC constructed in this way, background knowledge that does not appear in the context is needed to understand what is happening and to select an answer. Levesque et al. state that bringing this background knowledge is what thinking is all about.

The original Winograd Schema Challenge consisted of 137 pairs of sentences and a few sentences added later, and consisted of 284 questions, one for each pair of sentences, as shown in **Ex 1**. 273 of these questions(after excluding 11 of exceptional form) are widely used as the WSC273 evaluation set, which is also used in this study.

2.2. BERT

BERT uses a Masked Language Model (MLM) pre-training objective inspired by the cloze task (Taylor, 1953), which masks some tokens of the input by replacing them with [MASK] tokens and predicts the masked words based on context alone. In addition to MLM, BERT also uses Next Sentence Prediction (NSP), which jointly pre-trains text pairs, for pre-training purposes. The two unsupervised learning methods are used to pre-train each other. After pre-training with each of these two unsupervised learning methods, the model initialized with the parameters pre-trained here is fine-tuned with labeled data for downstream tasks. These fine-tuned models are initialized with the same pre-trained parameters, but after fine-tuning they become distinct models with different parameters for each task. In addition to the pre-training task, model size also has a significant impact on performance. Using the same hyper-parameters and training procedure but varying the number of layers, hidden units, and attention heads, larger models achieve higher accuracy.

3. Related Work

In recent years, as in case of other NLP tasks, methods using pre-trained, fine-tuned language models have dramatically improved the accuracy of WSC. However, WSC273 has only 273 examples. It is too small to fine-tune pre-trained model. WSCR (Rahman and Ng, 2012) was originally proposed to improve tasks that cannot be answered correctly by simple statistical methods such as WSC and consists of 941 sentence pairs. Each sentence is divided into a first half and a second half by a conjunction. The first half contains multiple candidate antecedents, and the second half contains an anaphora that refers to one of the candidate antecedents. Two sentences with the same first half, different second halves, and different antecedents to which the antecedent refers are paired. The dataset is about seven times larger than WSC273, but it is not strictly the same as WSC in that it does not necessarily require background knowledge not represented in the input sentences, and the conditions are slightly relaxed. In addition, Rahman and Ng's study (2012) did not evaluate WSC273 itself. Kocijan et al. (2019) were the first to use WSCR for fine-tuning BERT. They showed that fine-tuning BERT on the this dataset and a dataset generated from Wikipedia can robustly improve performance. Kocijan et al. updated the then atate-of-the-art by fine-tuning BERT with WSCR and achieved an accuracy of 72.5%. The WSCR dataset used in this study consists of a training set of 1,316 sentences and a test set of 564 sentences, totaling 1,880 sentences, excluding duplications with the WSC273 dataset. The model that has achieved the highest accuracy in this task as of 2023 is the one proposed by Sakaguchi et al. (2021). This is a pre-trained language model based on BERT called RoBERTa (Liu et al., 2019), fine-tuned on a dataset called WinoGrande, which recorded 90.1% accuracy when evaluated using WSC273. So far, we have shown that fine-tuning of pre-trained language models with similar data is effective for WSC, and that state-of-the-art is reaching the human level. However, these are only studies on the original WSC273, and all the experiments were conducted in English. In our study, we conduct fine-tuning using WSCR-ja and the Japanese version of BERT, which was pre-trained by the Tohoku University[1], and evaluate WSC273-ja to verify whether the above architecture is also effective for Japanese, and compare the differences between languages.

4. Our Approach

4.1. Construction of Japanese WSC dataset

To begin the experiment, we first constructed a Japanese version of the WSC dataset. Originally, WSC273 was translated into Japanese by Language Media Laboratory at Hokkaido University(WSC273-ja), but due to a few grammatical errors and shortcomings, we made some corrections and additions following the original WSC273. We

[1]https://www.nlp.ecei.tohoku.ac.jp/news-release/3284/

name the dataset WSC273-ja to match its English equivalent. In addition, to support input in BERT, we replaced the anaphora with [MASK] and reformatted it as shown in **Ex 2**.

Ex 2. *Sentence:*
じゃがいもの入った袋が，小麦粉の入った袋の上にあるので，最初に *[MASK]* を動かさなければならない.
The sack of potatoes had been placed above the bag of flour, so [MASK] had to be moved first.
Candidates：じゃがいもの入った袋, 小麦粉の入った袋
The sack of potatoes, The bag of flour
Answer：じゃがいもの入った袋 *The sack of potatoes*

4.2. Baseline construction by zero-shot

Next, a baseline is constructed by evaluating zero shots without fine-tuning. The model is BERT-large, which is published by Tohoku University. It follows the architecture of the English version of BERT-large, with 24 layers, 1024 hidden size, and 16 attention heads. The number of parameters is also 340M, as in the original BERT-large, but when comparing them, it should be noted that the size of the corpus used for pre-training is smaller in the Japanese version than in the English version. We experiment with the Japanese version of the corpus without changing the corpus size to match that of the English version. Furthermore, we tackle the WSC273-ja task with a model class called MASKedLM (MLM) in Japanese BERT as described above. This model class is used in the pre-training phase to predict the probability of [MASK] words. In this case, the process of replacing the anaphora with [MASK] was performed in advance. The candidate with the higher probability is taken as the predicted answer. We compared them with the correct answer and calculated its accuracy, which is used as the baseline. During tokenization, there may be multiple tokens for a candidate answer. In this case, [MASK] tokens are added to equal the number of tokens. Then, the harmonic mean of the probability of each of these multiple tokens being in [MASK] is taken as the probability of the candidate answer and compared with the probability of the other candidate answer. Similar experiments were conducted using the original WSC273 and BERT-large, as well as constructing an English version of the baseline.

4.3. Fine-tuning BERT with WSCR dataset

Finally, Japanese BERT is fine-tuned using the WSCR-ja dataset, and then evaluated using WSC273-ja. WSCR-ja follows Kocijan et al. (2019) and uses a training set of 1,316 sentences and a test set of 564 sentences, for a total of 1,880 sentences, like the original WSCR. The model used is the same as the one used for baseline construction in the previous section. However, in addition to MLM, a model class called MultipleChoice (MC) is used. This is a model class that predicts the probability of each candidate sentence following the previous sentence given a certain preamble and multiple candidate sentences. In this experiment, the problem sentences are separated before and after the [MASK] token, with the part before the appearance of [MASK] as the pre-sentence and the latter part including [MASK] as the candidate sentence. Furthermore, two sentences are used as candidate sentences, with the [MASK] token in the candidate sentence replaced by each candidate answer. The probability of each of these two sentences following the pre-sentence is predicted, and fine-tuning is performed with the one with the larger probability as the predicted answer. The hyperparameters for fine-tuning are explored among a learning rate of 1e-5, batch size of {8, 16, 32, 64}, and number of epochs of {5, 10, 15, 30}. Like WSCR, MLM is also fine-tuned following Kocijan et al. (2019) with a learning rate of 5e-6, batch size of 64, and number of epochs of 30, and the loss function is varied as indicated in **Eq 1** below. Note that c_1 and c_2 are candidate correct and incorrect answers, s is a training sentence, and $\mathbb{P}(c|s)$ represents their predicted probability. In this case, the hyperparameters α and β are 20 and 0.2, respectively.

Eq 1. $L = -\log \mathbb{P}(c_1|s) +$

$$\alpha \cdot \max(0, \log \mathbb{P}(c_2|s) - \log \mathbb{P}(c_1|s) + \beta)$$

As mentioned in the original BERT paper (Devlin et al., 2018), BERT-large tends to be unstable when fine-tuning on small datasets. Therefore, multiple experiments are conducted using random seeds, and the model with the best results is selected. In this study, each hyperparameter is fine-tuned three times, and the model with the highest accuracy on the WSCR-ja test set is selected as the proposed model among all models. As described in the previous section, we also conduct fine-tuning of the English version of the models using the original WSCR and BERT-large. These models are evaluated using WSC273-ja and WSC273, respectively. In addition, we construct WSCR-ja-small dataset, which excludes grammatically and culturally unnatural examples from WSCR-ja. The resulting WSCR-ja-small consists of 1,196 training sets and 530 test sets, for a total of 1,726 examples. Then, WSCR-small is constructed from the original WSCR, excluding the same examples that were excluded from WSCR-ja-small. By comparing the two, we analyze the impact of the above inappropriate examples on fine-tuning.

5. Evaluation

5.1. Baseline

The baseline accuracy was 0.567 in Japanese and 0.619 in English as shown in Table 1. This accuracy in English is identical to the one reported in the related study by Kocijan et al. (2019) described in Section 3.

	baseline	WSCR	WSC273
BERT-ja (MC)	0.567	0.719	0.575
BERT-ja (MLM)	0.567	0.728	0.578
BERT (MC)	0.619	0.820	0.688
BERT (MLM)	0.619	0.785	0.652

Table 1: Accuracy for each learning objective (WSCR-ja and WSCR)

	WSCR-small	WSC273
BERT-ja (MC)	0.700	0.578
BERT-ja (MLM)	0.718	0.578
BERT (MC)	0.811	0.673
BERT (MLM)	0.784	0.659

Table 2: Accuracy for each learning objective (WSCR-ja-small and WSCR-small)

BERT-ja	5	10	15	30
8	0.595	0.693	0.698	0.716
16	0.585	0.684	0.710	**0.719**
32	0.586	0.695	0.663	0.705
64	0.547	0.595	0.581	0.648
BERT	5	10	15	30
8	0.707	0.796	0.799	0.519
16	0.702	0.769	0.804	**0.820**
32	0.542	0.744	0.794	0.808
64	0.553	0.647	0.684	0.803

Table 3: Accuracy of the WSCR test set for each hyperparameter of MC model, where the first column represents the batch size and the column headers the number of epochs. Each value is selected as the highest among the experiments with three different seeds (learning rates are all 1e-5)

5.2. Adaptation by fine-tuning

The results of fine-tuning with WSCR or WSCR-small are shown in Tables 1 and 2, respectively. The values in the tables represent the accuracy when evaluating the test set and WSC273 dataset for each model. The accuracies of the test set at each hyperparameter when fine-tuning MC model with the WSCR training set are shown in Table 3. The top of Table 3 shows the results for Japanese BERT and the bottom for the original English BERT. The first column indicates the batch size, and column headers shows the number of epochs. These are the highest accuracy results from the three different seeding experiments. The model with the highest accuracy (0.719 in Japanese and 0.820 in English) is the one with batch size 16 and number of epochs 30 in both Japanese and English, and was selected as the proposed model for MC. As shown in Table 1, the result for WSC273 using the proposed model is 0.575 for Japanese, while for English it is 0.688. The MLM used the hyperparameters described in the previous section, and three experiments were conducted with different seeds, and the one with the highest accuracy on the test set was used as the proposed model. Fine-tuning with WSCR-small was performed for both MC and MLM using the same hyperparameters as the above proposed model. The results are shown in Table 2.

6. Discussion

The accuracy of the model at baseline and after fine-tuning is 0.567 and 0.578 for the Japanese model, respectively, while it is 0.619 and 0.688 for the English model. The results show that the accuracy of the Japanese model is lower than that of the English model both at baseline and after fine-tuning. There is a large gap between the accuracies on the WSCR test set and on the WSC273 for both the Japanese (0.728 vs. 0.578) and English (0.820 vs. 0.688) models. In addition, although both are improved by fine-tuning, the accuracy of Japanese is not as good as the accuracy of English. In this section, we discuss the results from two viewpoints: the influence of the model and the influence of the dataset. Based on these discussion, we propose some perspectives for future research.

6.1. The influence of the model

As mentioned in Section 4, Japanese BERT was created following BERT, so the architecture and number of parameters are the same as the original BERT. However, the size of the corpus used for pre-training Japanese BERT is only about 1/4 of the original. We believe that this difference in corpus size is one of the reasons why Japanese BERT was lower than BERT in the final score. This is also mentioned in Shibata et al.'s study (2019), which shows that the larger the corpus size, the better the performance of the model. One reason for the difference in results between Japanese and English seems to be the pre-trained models themselves. It is not clear at this point whether this is due to the corpus size alone, or whether the pre-training of the language model in a language other than English has some other effects. The first step is to build and compare pre-trained models in English and Japanese, under several conditions including matched corpus size. Another possible reason for the difference in results between Japanese and English is linguistic differences. This is discussed in detail in the next subsection.

Next, we compare the model after fine-tuning with the existing studies: as already mentioned in Section 5, the baseline results obtained in this study for English BERT is the same as the value shown in the related study by Kocijan et al. (2019) We believe that this is a reasonable baseline for Japanese BERT, which was evaluated under the same conditions (except for the corpus size in the pre-training phase). However, the score after fine-tuning is only 0.688 for the MC model in this study, while Kocijan et al. recorded 0.714 using only WSCR. The same MLM model as Kocijan et al. only reaches a score of about 0.652. In this experiment, we used Hugging Face Transformer library, but it is not clear if this is the reason why we could not reproduce the results of Kocijan et al. It is necessary to continue to investigate the causes through, for example, the search for hyperparameters. On the other hand, in the Japanese model, a slight increase in accuracy was observed for both the test set and WSC273-ja dataset when MLM was used compared to MC. This suggests that MLM may be more compatible with the Japanese WSC than MC.

Finally, as for the change in performance depending on the hyperparameters, the accuracy improves with the number of epochs in both the MC models, and the batch size of 16 appears to be most optimal.

6.2. The influence of the dataset

In the previous section, we looked at the difference between the Japanese and English results mainly in terms of the in-

fluence of the model, but in this section, we will discuss it in terms of the influence of the dataset. After analyzing both sets, we have not observed any particular differences due to differences between languages, however it should be noted that in Japanese subject is often omitted, which sometimes led to creating redundant sentences for masking. As shown above, there is a clear difference in performance improvement between the Japanese and English models after fine-tuning. At baseline, the accuracy of Japanese is 0.567 and that of English is 0.619, with a difference of 0.052. After fine-tuning, the accuracy of Japanese is 0.578 and that of English is 0.688, with a difference of 0.110. The difference is even larger than when comparing the two baselines. This may be due to the WSCR-ja training set. As noted in the original paper (Shibata et al., 2015), the English WSCR contains inappropriate examples. In addition to these examples, WSCR-ja is known to contain several examples that are inappropriate due to the translation of English into Japanese. These examples include words that are common in English texts but do not usually appear in Japanese texts due to cultural differences. Therefore, we created WSCR-small by removing those inappropriate examples (120 in the training data and 34 in the test data) and observed the change in adaptation to WSC, as shown in Table 2. As can be seen from the table, despite the reduction in the size of the training data set, the accuracy of WSC273 remains the same or even improves for the Japanese BERT. Combined with the decrease in accuracy in the test set, it appears that the removal of inappropriate examples from the WSCR data set has made the Japanese BERT more adaptable to the WSC273. This can be explained by the fact that the accuracy of the test set and WSC273 decreased in the English MC model. However, in the English MLM model, there was a slight improvement (from 0.652 to 0.659) in the accuracy of WSC273, suggesting that some of the removed cases were inappropriate before translation. In addition to these problems, there is also issue of WSC itself being vulnerable, as described by Sakaguchi et al. (2021) To begin with, WSC was designed so that it could not be solved by simple statistical methods, but as technology has advanced, it is no longer necessarily a problem that cannot be solved by statistical methods alone. However, as Levesque et al. noted in their original paper (2012), statistical methods have not been completely ruled out.

7. Conclusion

In this study, we constructed a baseline by evaluating the Japanese WSC dataset with zero-shot using BERT, which was pre-trained on a corpus in Japanese.

We also conducted fine-tuning of Japanese BERT with WSCR-ja, a large similar dataset of WSC translated into Japanese, and confirmed an improvement in accuracy from 0.567 to 0.578. This indicates that fine-tuning with Japanese WSCR is effective in improving the common sense reasoning ability of Japanese BERT, and that the Japanese model is not as well adapted to WSC as the English model. Furthermore, we demonstrated that the Japanese model is more adaptable to WSC by removing inappropriate examples from the Japanese WSCR. Future work includes further cross-language comparisons using different models, testing whether background knowledge can be used more clearly by incorporating graphs, etc., and improving the Japanese dataset.

References

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Kocijan, Vid, Ana-Maria Cretu, Oana-Maria Camburu, Yordan Yordanov, and Thomas Lukasiewicz, 2019. A surprisingly robust trick for the winograd schema challenge. In Anna Korhonen and David Traum (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019*. Association for Computational Linguistics.

Levesque, Hector, Ernest Davis, and Leora Morgenstern, 2012. The Winograd Schema Challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Radford, Alec, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al., 2018. Improving language understanding by generative pre-training. *Technical Report. OpenAI*.

Rahman, Altaf and Vincent Ng, 2012. Resolving complex cases of definite pronouns: the Winograd Schema Challenge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Sakaguchi, Keisuke, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi, 2021. WinoGrande: An adversarial Winograd Schema Challenge at scale. *Communications of the ACM*, 64(9):99–106.

Shibata, Tomohide, Daisuke Kawahara, and Sadao Kurohashi, 2019. Improved accuracy of Japanese parsing with BERT in Japanese. *Proceedings of the Twenty-fifth Annual Meeting of the Association for Natural Language Processing*:205–208.

Shibata, Tomohide, Shotaro Kohama, and Sadao Kurohashi, 2015. Construction and analysis of the Japanese Winograd Schema Challenge in Japanese. *Proceedings of the Twenty-first Annual Meeting of the Association for Natural Language Processing*:493–496.

Taylor, Wilson L, 1953. "Cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

Using amplitude envelope modulation spectra to capture differences between rhetorical and information-seeking questions

Friederike Hohl, Bettina Braun

Department of Linguistics, University of Konstanz, Germany
{friederike.hohl, bettina.braun}@uni-konstanz.de

Abstract

Rhetorical questions (RQs) and information-seeking questions (ISQs) differ in their pragmatic function (the first making a point, the second requesting information). They may have identical surface forms. For human-computer interaction, but also for interaction between humans, it is important to decode the intended function. Laboratory experiments have established that RQs are longer, more often realized with breathy voice quality than ISQs and differ in intonational realization. However, annotation is labor-intensive. Here we test whether the prosodic differences between RQs and ISQs are evident in the amplitude envelope modulation spectra. These capture the slow-changing energy distribution over utterances and do not demand manual annotation. Since amplitude envelope modulation spectra are sensitive to rhythmic differences between languages, they may be well-suited to capture the duration differences. We compare RQs and ISQs in three closely-related languages (English, German, Icelandic) to investigate whether RQs have different amplitude envelope modulation spectra than ISQs and whether these differences are language-specific. The results show differences between RQs and ISQs but, depending on language, in different frequency bands. We show that the differences cannot be explained by durational differences between RQs and ISQs alone, but that the amplitude envelopes capture the signal more holistically.

Keywords: question, prosody, amplitude envelopes, cross-linguistic, general additive mixed models

1. Introduction

The analysis of amplitude envelopes has become a widely used method in the speech sciences, language acquisition and neurolinguistics (Frota et al., 2022; Gross et al., 2013; Leong & Goswami, 2015; Poeppel, 2014; Poeppel & Assaneo, 2020). Amplitude envelopes track the amplitude distribution over an utterance and hence represent the part of the signal that is relevant to convey rhythm (Arvaniti, 2009). Furthermore, the method is easy to apply without demanding manual annotation (cf. Gibbon, 2021 for discussion of advantages of modulation-theoretic methods). Despite the increasingly wide-spread usage across disciplines, there is little research on *which aspects* of the speech signal influence the amplitude envelopes in *what way*. Cross-linguistic research has shown that a stress-timed language (German) led to lower power around 2Hz and to higher power between 7 and 10Hz than more syllable-timed Brazilian Portuguese (Frota et al., 2022), cf. Tilsen & Arvaniti (2013). Others have explored the use of amplitude envelopes for differences in speech style (Gibbon, 2021). In this paper, we test whether amplitude envelope modulation spectra can distinguish also between rhetorical vs. information-seeking questions.

Rhetorical questions do not seek information from the addressee but serve to make a make a point and commit the interlocutor to the presupposition expressed by the RQ. For instance, the questions in (1), uttered as rhetorical questions, attempt to commit the interlocutor to the statement that nobody likes phonetics (Biezma & Rawlins, 2017)

(1)	Who likes phonetics?	wh-question
	Does anyone like phonetics?	polar question

Since the questions in (1) can also be uttered to seek information (e.g. to find a suitable student assistant), it is sometimes only the prosodic realization that can help disambiguate between the two meanings. This disambiguation is not only important for human communication, but also for human-computer interaction and sentiment analysis.

Previous production data have shown consistent differences between RQs and ISQs across intonation languages such as English (Dehé & Braun, 2019), German (Braun et al., 2019, 2020), and Icelandic (Dehé et al., 2018), cf. Dehé et al., (2022) for an overview: In all of these intonation languages, RQs have longer constituent durations. Less consistent is the greater use of non-modal voice quality (breathy, glottalized) in RQs compared to ISQs. Differences in the intonational realization are language-specific. English speakers more often produced the nuclear (last) accent on the subject pronoun 'anyone' in RQs (but not in ISQs), followed by a high plateau, while the nuclear accent was typically produced on the final noun in German and Icelandic. In Icelandic, the boundary tone was always falling in both RQ and ISQs. Icelandic speakers more often produced an early-rise in RQs (i.e. the rise started early in the final noun), but the difference was not strong. In German, speakers more frequently produced a prominent rising accent (L*+H) in RQs (compared to a low accent, L*, in ISQs). Using classification and regression trees, German questions could be classified as RQ or ISQ with an accuracy of 87.5% with these parameters (Braun et al., 2018).

However, manual annotation of prosody is cost-intensive. In this paper, we therefore test whether RQs and ISQs also differ in terms of amplitude envelopes. Amplitude envelopes capture the wideband energy distribution and therefore capture suprasegmental differences such as differences in duration or voice quality (resulting in lower energy in high frequency areas). Modulation frequencies can be extracted from the speech signal in a number of ways (Poeppel & Assaneo, 2020). Most procedures first filter the sound into a number of frequency bands (spaced either logarithmically or such that they are equidistant on the cochlea), typically in the range between 100 and 8,000Hz

(or 10,000Hz). These signals are then filtered to remove the high-frequency components, leaving frequencies in the range of 0 to approximately 10Hz. These narrowband envelopes are then summed and the modulation frequencies are derived by Fourier analysis. The result is a spectrum, i.e. power values across frequency. We then compare the patterns holistically, rather than extracting single parameters (Tilsen & Arvaniti, 2013).

2. Data

2.1. Methods

The data were collected in separate production experiments. For all three languages (English, German, Icelandic), participants saw a context description, which was constructed to trigger a rhetorical or information-seeking intention (illocution). They then produced a visually presented question so that it fit the respective context.

2.1.1. Participants

For English, 21 participants (mean age 22.5 years, 14 female, 7 male), for German 12 participants (mean age 21 years, SD = 2.3 years, 10 female, 2 male,) and for Icelandic, 32 participants (aged 20–65, 20 female and 12 male) took part in the data collection for a small fee. All participants gave informed consent.

2.1.2. Materials

We constructed 11 *wh*-interrogatives that fitted both a rhetorical and an information-seeking reading (e.g., *Who likes celery?*). To this end, we used predications that – out of context – may be true for some people and false for others (e.g., 'liking celery'). From these *wh*-interrogatives, we derived polar questions by replacing the *wh*-word by the indefinite pronominal subject *anyone* and adapted the syntactic structure to verb-first (V1).

For each polar question, we constructed two contexts, one triggering an information-seeking interpretation of the interrogative and one triggering a rhetorical one. An example of polar question contexts is given in Table 1. To control for information structure and specifically to avoid effects of information structure on nuclear accent position and type, each context introduced the predication expressed in the sentence radical (e.g., *liking celery* in Table 1), rendering the referents of the constituents in the verb phrase discourse-given (see Braun et al., 2019 for more details).

ISQ	**RQ**
You cooked a dish with celery. You would like to know whether your guests like this vegetable and will eat it or not. You say to your guests:	In the canteen they have casserole with celery on the menu. However, you know that nobody likes this disgusting vegetable. You say to your friends:
'Does anyone like celery?'	

Table 1. Example contexts for information-seeking (ISQ, left) and rhetorical questions (RQ, right)).

The rhetorical contexts contained a sentence stating that it is generally known (or that the speaker knows) that nobody agrees with a certain proposition (e.g., *you know that nobody likes celery*). The information-seeking contexts differed from the rhetorical contexts in that they stated that the speaker was looking for some piece of information.

Additionally, 24 fillers with different syntactic structures were added to reduce awareness of the experimental manipulation. The materials were first designed for German, and then translated into English and Icelandic, with minor adaptions to account for cultural and phonological differences.

2.1.3. Procedure

Recording. Each participant produced both the rhetorical and the information-seeking version of each target interrogative in randomized order. Each experiment started with four familiarization trials, followed by a short break in which participants were allowed to ask questions if anything was unclear. The experiment was controlled using the experimental software *Presentation* (Neurobehavioral-Systems, 2000). Each trial started with the visual display of the context, which the participant had to read silently, followed – upon button press – by the target interrogative on the next screen. The target sentence had to be produced aloud and was recorded onto disk (44,100Hz, 16Bit).

Extraction of amplitude envelope modulation spectra. All productions (N = 1000) were cut at utterance start and end. Average durations across conditions are shown in Table 2 and show lengthening of RQs as compared to ISQs.

Language	RQ	ISQ	Proportional lengthening of RQs
English	1.456	1.311	11.1%
German	1.551	1.330	16.7%
Icelandic	1.419	1.140	24.5%

Table 2. Average durations across languages (rows) and illocution types (columns) in seconds, including the proportional durational increase from ISQ to RQ.

Amplitude envelopes for all questions were extracted, following the descriptions in the literature (Chandrasekaran et al., 2009; Frota et al., 2022; Gross et al., 2013). First, we calculated the narrowband amplitude envelopes (He & Dellwo, 2016, 2017). The speech signal was first down-sampled to 22,050Hz and then filtered into nine frequency bands in the range from 100–10,000Hz, which are equidistant on the cochlear map (Gross et al., 2013). The cutoff frequencies were 100.5Hz, 250.7Hz, 458.6Hz, 748.8Hz, 1159.0Hz, 1449.0Hz, 2619.8Hz, 3954.2Hz, 6121.8Hz and 10000.8Hz. To remove high-frequency components, the signals were low-pass filtered (Hann filter between 0 and 10Hz with 1 Hz smoothing). The resulting narrowband envelopes were then added to compute the wideband amplitude envelope. These were spectrally analyzed in 100 0.1Hz steps. This approach is conceptually similar to approaches that do not compute narrowband envelopes (Gibbon, 2021; Tilsen & Johnson, 2008). The wideband envelope was spectrally analyzed in 100 0.1Hz steps (fast Fourier transform). All signal processing was done in Praat (Boersma & Weenink, 2018).

Statistical modeling. To model the effect of *language* and *illocution type* across frequency bands, we used generalized additive mixed models, GAMMs (Wieling et al., 2012; Wood, 2006, 2015; Wood & Saefken, 2016; Zahner et al.,

2019). They are well-suited to pinpoint in which frequency bands differences occur; taking into account non-linear relationships and auto-correlation. The response variable was log-normalized power. We modelled non-linear dependencies of *language* and *illocution type* first as separate smooth terms (e.g., s(fband_Hz, by = language, bs='tp', k = 20)). These smooth functions include a pre-specified number of base functions of different shapes, e.g., linear and parabolic functions of different complexity. The two factors *language* and *illouction type* were further added as parametric effects. Smooths for *speakers* (random intercept and over frequency bands) were also included (s(speaker, fband_Hz, by='re'). For model fitting, we employed the R package *mgcv* (Wood, 2015). The model was corrected for auto-correlation in the data using a correlation parameter, determined by the acf_resid() function. We use the function gam.check() to check whether the number of smooth functions (k) and the smoother (thin plate regression, 'tp') were adequate and adjusted if necessary.

2.2. Predictions

All the languages lengthened RQs compared to ISQs, most strongly in Icelandic (24.5%), see Table 2. This lengthening is expected to affect the amplitude envelopes in all three languages and is predicted to result in higher power in lower-frequency bands in RQs compared to ISQs. The differences are expected to be strongest in Icelandic and weakest in English, based on the extent of lengthening.

2.3. Results

For reasons of space, we do not show the spectra of the two illocution types separately, but directly present the *differences* in power spectra for RQs vs. ISQs (Figs. 1-3).

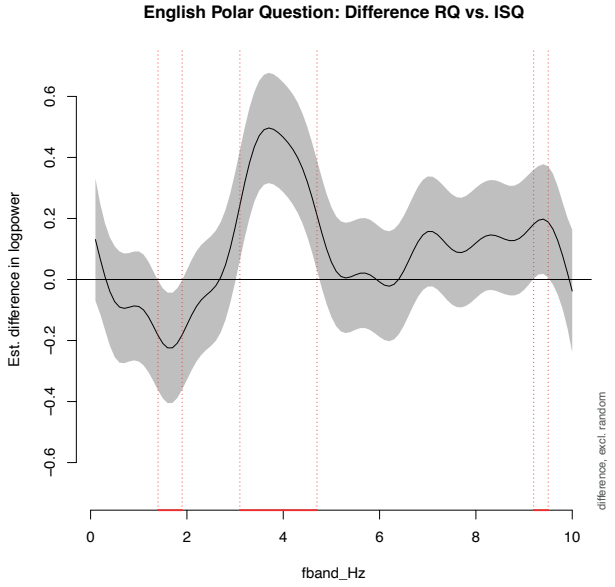


Fig. 1: Effect of illocution type (RQ minus ISQ) in English. Positive values indicate higher power for RQs than RQs. If the gray band of the confidence interval does not include 0, the difference is considered statistically significant at $\alpha = 0.05$.

The languages differ in how strongly illocution type affects the amplitude envelope modulation spectra. On the one hand, there were strong effects of illocution type on the English data (Fig. 1). English polar RQs had a lower power in the frequency range 1.4 – 1.9Hz and, prominently, higher power in the frequency range 3.1 – 4.7Hz.

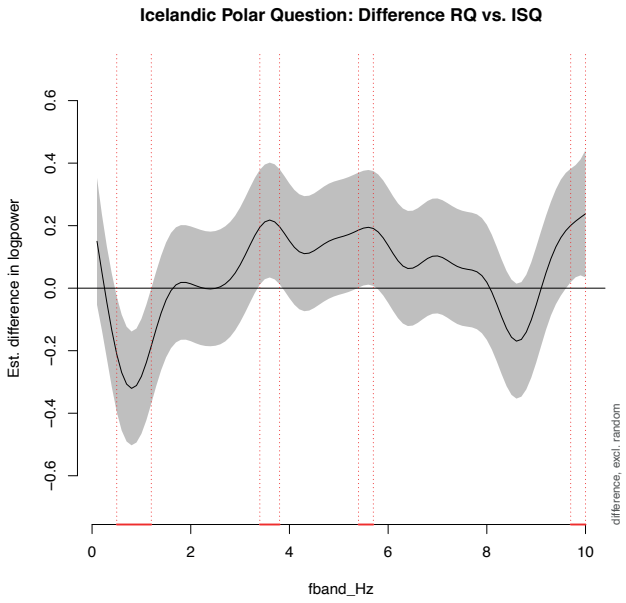


Fig. 2: Effect of illocution type (RQ minus ISQ) in Icelandic.

Icelandic (Fig. 2) shows differences as well, but in a smaller frequency range (0.5 – 1.2Hz and 3.4 – 3.8Hz) and with smaller differences in power. Furthermore, the differences occur in a slightly lower frequency band. German (Fig. 3) is again different: It exhibits a biphasic pattern very late, in the area between 7.3Hz and 9.1Hz, first lower power for RQs, then higher power for RQs. However, compared to English, the differences in power are small.

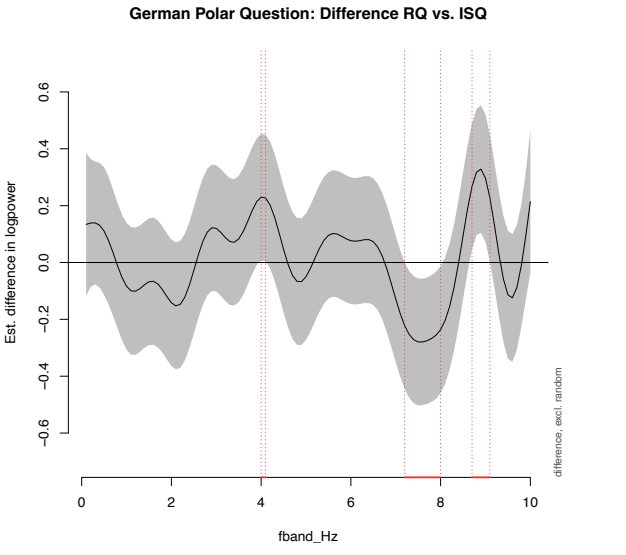


Fig. 3: Effect of illocution type (RQ minus ISQ) in German.

To investigate whether the differences across languages are significant, we fit a model with a smooth for the interaction *language* and *illocution type* and compared it with a model with smooths for the individual terms *language* and *illocution type*, using the package *itsadug* (van Rij et al., 2015). Model comparison showed that the model with the interaction-smooth provided a significantly better fit than the model without ($X2(14.00)=62.450$, $p<2e-16$). To corroborate the interaction between *language* and *illocution type* indicated in the model, we constructed additional models containing binary difference smooth terms that capture *the difference of the difference* over frequency band between two languages (English vs. German, English vs. Icelandic, German vs. Icelandic), closely following the procedure described in van Rij et al., (2019, pp. 11–13) and Wieling (2018, p. 109ff).

The results showed a number of frequency bands with *significant differences across language pairs*. These differences are as follows:

- English differed from German in the frequency band from 3 – 5Hz and 6.5 – 8.5Hz.
- English differed from Icelandic in the frequency band from 3 – 5Hz and 7.8 – 9.5Hz.
- German differed from Icelandic in the frequency band from 0.2 – 1.2Hz, 4.8 – 5.2Hz and 7 – 9.5Hz.

2.4. Discussion

All languages showed an effect of illocution type on the amplitude envelopes. Since information structure was controlled across illocution types (i.e. the same for ISQs and RQs), the differences cannot be related to that factor. There were significant differences between the three intonation languages on the frequency bands in which RQs differed from ISQs. The power differences were strongest in English, with a pronounced peak in energy around 4Hz. The amplitude envelope differences across languages do not mirror the durational lengthening (Table 1). Therefore, it is unlikely that the amplitude envelopes only track the durational differences between RQs and ISQs. Interestingly, the English and Icelandic power differences show a similar pattern in the lower frequency range, but the Icelandic differences are much smaller. The German data show a pronounced difference in the higher frequency band (from 7.5 – 10 Hz).

If duration is a bad predictor for these power differences between RQs and ISQs, we need to take a closer look at other prosodic cues that may explain the cross-linguistic differences. In terms of **voice quality**, German is the only language with differences in voice quality across illocution types (in German 36% of the first words in RQs were breathy, compared to 10% in ISQs, cf. Braun et al., 2019). This cue may explain the biphasic pattern in the high frequency bands, which is absent in English and Icelandic. These latter two languages do not show voice quality differences for polar questions (Dehé & Braun, 2019; Dehé & Wochner, 2022). **Intonationally**, Icelandic and English are similar in terms of accent placement: in both languages, the subject ('anyone' in (1)) has a higher probability of receiving an accent in RQs compared to ISQ (28.8% vs. 0% in English, 12.3% vs. 0.6% in Icelandic). This may explain the power differences below 2Hz and around 4Hz. On the contrary, the fact that both German and English end RQs with high plateau boundary tones (and ISQs with high rising boundary tones) does not seem to be reflected in the amplitude modulation. For automatic classification of questions as ISQ or RQ, parallel consideration of f0 may prove useful (Gibbon, 2021; Ludusan et al., 2011).

Taken together, amplitude envelope modulation spectra differ across illocution types and are most likely influenced by differences in voice quality and accent placement, and less by intonational contour.

3. General Discussion

We showed that amplitude envelope modulation spectra distinguish rhetorical and information-seeking questions in three closely related Germanic languages. We predicted that differences would be largest in Icelandic because this language showed the largest duration differences between RQs and ISQs. However, the amplitude envelope modulation spectra were not largest for Icelandic, but for English. Therefore, the amplitude envelope modulation spectra differences were not (or at least not only) caused by durational differences between RQs and ISQs. Relating amplitude envelope modulation spectra differences to prosodic differences across conditions suggests that voice quality differences and differences in accent placement may play a significant role. In particular, voice quality differences on the first word of the question (more often breathy in RQs in German) seem to have an effect on higher frequency bands, most likely because breathy voice reduces the spectral power of the words. Furthermore, English and Icelandic often placed an accent on the subject pronoun 'anyone', which affects the macro-prosodic rhythm of the utterance (Jun, 2012).

In future work, we plan to include typologically different languages, e.g., such as tone languages (Zahner-Ritter et al., 2022 for Chinese) or accentual phrase languages to get a better overview on the factors that influence amplitude envelope modulation spectra. Furthermore, we plan to use the parameters from the general additive mixed models for automatic classification of utterances as RQs vs. ISQs.

4. Conclusion

This paper adds to our understanding of the factors that influence amplitude envelope modulation spectra by testing three intonation languages. Previous research has shown differences between rhythmically different languages (stressed-timed German vs. more syllable-timed Brazilian Portuguese, cf. Frota et al., 2022). We show that even within one and the same language, amplitude envelope modulation spectra can differ quite extensively (in particular in English polar RQs vs. ISQs). The results showed that lengthening is a poor predictor of differences in amplitude envelope modulation spectra across languages. Differences in voice quality and accent placement also seem to play a role. Clearly, more analyses of carefully controlled materials from typologically different languages are necessary to understand better, which information is encoded in which way in amplitude envelope modulation spectra.

5. Acknowledgements

We thank Volker Dellwo for providing the praat-scripts for extracting the narrowband amplitude envelopes. Furthermore, we thank Antje Strauß and Sonia Frota for discussion on amplitude envelope modulation spectra. Data collection was funded by the German research foundation (BR 3428/4-1, 4-2).

6. References

Arvaniti, A. (2009). Rhythm, timing and the timing of rhythm. *Phonetica*, *66*, 46–63.

Biezma, M., & Rawlins, K. (2017). Rhetorical questions: Severing asking from questioning. In D. Burgdorf, J. Collard, S. Maspong, & B. Stefánsdóttir (Eds.), *Proceedings of SALT 27* (pp. 302–322).

Boersma, P., & Weenink, D. (2018). *Praat: Doing phonetics by computer*.

Bögel, T., & Braun, B. (2022). Rhetorical questions in Persian. *Phonetik Und Phonologie Im Deutschsprachigen Raum*, Phonetik und Phonologie im deutschsprachigen Raum. https://doi.org/10.11576/PUNDP2022-1042

Braun, B., Daniela Wochner, Zahner, K., & Nicole Dehé. (2018). *Classification of interrogatives as information-seeking or rhetorical questions*. 17th Speech Science and Technology Conference. Sydney, Australia.

Braun, B., Dehé, N., Neitsch, J., Wochner, D., & Zahner, K. (2019). The prosody of rhetorical and information-seeking questions in German. *Language and Speech*, *62*(4), 779–807. https://doi.org/10.1177/0023830918816351

Braun, B., Einfeldt, M., Esposito, G., & Dehé, N. (2020). *The prosodic realization of rhetorical and information-questions in German spontaneous speech*. Speech Prosody.Tokyo, Japan.

Chandrasekaran, C., Trubanova, A., Stillittano, S., Caplier, A., & Ghazanfar, A. A. (2009). The Natural Statistics of Audiovisual Speech. *PLOS Computational Biology*, *5*(7), e1000436. https://doi.org/10.1371/journal.pcbi.1000436

Dehé, N., & Braun, B. (2019). The prosody of rhetorical questions in English. *English Language and Linguistics*, *24*(4), 607–635. https://doi.org/10.1017/s1360674319000157

Dehé, N., Braun, B., Einfeldt, M., Wochner, D., & Zahner-Ritter, K. (2022). The prosody of rhetorical questions: A cross-linguistic view. *Linguistische Berichte*, *269*, 3–42. https://doi.org/10.46771/978-3-96769-175-7

Dehé, N., Braun, B., & Wochner, D. (2018). *The prosody of rhetorical vs. Information-seeking questions in Icelandic*. 403–407.

Dehé, N., & Wochner, D. (2022). Voice quality and speaking rate in Icelandic rhetorical questions. *Nordic Journal of Linguistics*, 1–10. https://doi.org/10.1017/S0332586522000014

Frota, S., Vigário, M., Cruz, M., Hohl, F., & Braun, B. (2022). *Amplitude envelope modulations across languages reflect prosody*. https://doi.org/10.21437/SpeechProsody.2022-140

Gibbon, D. (2021). The rhythms of rhythm. *Journal of the International Phonetic Association*, 1–33. https://doi.org/10.1017/S0025100321000086

Gross, J., Hoogenboom, N., Thut, G., Schys, P., Panzeri, S., Belin, P., & Garrod, S. (2013). Speech Rhythms and Multiplexed Oscillatory Sensory Coding in the Human Brain. *PLOS Biology*. https://doi.org/10.1371/journal.pbio.1001752

He, L., & Dellwo, V. (2016). *A Praat-based algorithm to extract the amplitude envelope and temporal fine structure using the Hilbert transform*. 530–534.

He, L., & Dellwo, V. (2017). *Amplitude envelope kinematics of speech signal: Parameter extraction and applications. Elektronische Sprachsignalverarbeitung 2017. Dresden: TUDpress, 1-8*.

Jun, S.-A. (2012). *Prosodic Typology Revisited: Adding Macro-Rhythm*. Speech Prosody, Shanghai, China.

Leong, V., & Goswami, U. (2015). Acoustic-emergent phonology in the amplitude envelope of child-directed speech. *PLoS One*, *10*(12), e0144411.

Ludusan, B., Origlia, A., & Cutugno, F. (2011). *On the Use of the Rhythmogram for Automatic Syllabic Prominence Detection*. Interspeech 2011. Florence, Italy.

Poeppel, D. (2014). The neuroanatomic and neurophysiological infrastructure for speech and language. *Curr Opin Neurobiol*, *28*, 142–149.

Poeppel, D., & Assaneo, M. F. (2020). Speech rhythms and their neural foundations. *Nature Review Neuroscience*, *21*, 322–334.

Tilsen, S., & Arvaniti, A. (2013). Speech rhythm analysis with decomposition of the amplitude envelope: Characterizing rhythmic patterns within and across languages. *The Journal of the Acoustical Society of America*, *134*(1), 628–639.

van Rij, J., Hendriks, P., van Rijn, H., Baayen, R. H., & Wood, S. N. (2019). Analyzing the Time Course of Pupillometric Data. *Trends in Hearing*, *23*, 233121651983248.

van Rij, J., Wieling, M., Baayen, R. H., & van Rijn, D. (2015). *itsadug: Interpreting Time Series and Autocorrelated Data Using GAMMs*. https://cran.r-project.org/package=itsadug

Wieling, M. (2018). Analyzing dynamic phonetic data using generalized additive mixed modeling: A tutorial focusing on articulatory differences between L1 and L2 speakers of English. *Journal of Phonetics*, *70*, 86–116.

Wieling, M., Margaretha, E., & Nerbonne, J. (2012). Inducing a measure of phonetic similarity from pronunciation variation. *Journal of Phonetics*, *40*(2), 307–314. https://doi.org/10.1016/j.wocn.2011.12.004

Wood, S. N. (2006). *Generalized additive models: An introduction with R*. Chapman & Hall/CRC Press.

Wood, S. N. (2015). *mgcv: Mixed GAM computation vehicle with GCV/AIC/REML smoothness estimation*.

Wood, S. N., & Saefken, B. (2016). Smoothing parameter and model selection for general smooth models. *Journal of the American Statistical Association*, *111*, 1548–1575.

Zahner, K., Kutscheid, S., & Braun, B. (2019). Alignment of f0 peak in different pitch accent types affects perception of metrical stress. *Journal of Phonetics*, *74*, 75–95.

Zahner-Ritter, K., Chen, Y., Dehé, N., & Braun, B. (2022). The prosodic marking of rhetorical questions in Standard Chinese. *Journal of Phonetics*, *95*, 101190.

Semantic Information Investigation for Transformer-based Rescoring of N-best Speech Recognition

Irina Illina, Dominique Fohr

Université de Lorraine, CNRS, Inria, Loria, F-54000 Nancy, France
{illina, dominique.fohr}.loria.fr

Abstract

This article proposes to improve an automatic speech recognition system by rescoring N-best recognition lists with models that could enhance the semantic consistency of the hypotheses. We believe that in noisy parts of speech, the semantic model can help remove acoustic ambiguities. The estimate of a pairwise score for each pair of hypotheses is performed by *BERT* representations. The acoustic likelihood and LM scores are used as features in order to incorporate acoustic, language, and textual information together. In this research work, two new ideas are investigated: to use a fine-grained semantic representation at the word token level and to rely on the previously recognized sentences. On the TED-LIUM 3 dataset, in clean and noisy conditions, the best performance is obtained by leveraging context beyond the current utterance, which significantly outperforms the rescoring using the state-of-the-art GPT-2 model and the work of Fohr and Illina (2021).

Keywords: automatic speech recognition, semantic context, Transformer-based language models.

1. Introduction

Nowadays, automatic speech recognition systems (ASR) are widely used in everyday life. However, in the presence of noise, the degradation in performance can be detrimental to real applications (Deng *et al.*, 2014). In noisy conditions, the speech signal is less reliable and other knowledge is required to guide the recognition process. One possibility is to take into account the long-term context through a *semantic model*.

Semantic information is increasingly explored in recent works. Zhao *et al.* (2021) explore the denoising autoencoder for pretraining sequence-to-sequence semantic correction method and use transfer learning. Level *et al.* (2020) introduce the notions of a context part and possibility zones. Kumar *et al.* (2017) extract the semantic relations from the *DBpedia* (Auer *et al*., 2007) and uses them as features for rescoring.

An efficient solution to incorporate long-range semantic information can be through the *rescoring of the ASR N-best hypotheses list*. Ogawa *et al.* (2018, 2019) introduce N-best rescoring through a Long Short-Term Memory (LSTM) based encoder network. Liu *et al.* (2021) present a domain-aware rescoring framework for achieving domain adaptation during second-pass rescoring. A large range of textual information from different NLP models and a procedure to automatically estimate their weights are used by Song *et al.* (2021). A domain-aware rescoring framework to achieve domain adaptation during second-pass rescoring is proposed by Liu *et al.* (2021). In Xu *et al*. (2022), for second-pass rescoring the authors propose to train a *Bidirectional Encoder Representations from Transformers* (BERT) (Devlin *et al*., 2019; Wang and Cho, 2019) on a discriminative objective such as minimum word error rate.

Some studies have attempted to include semantic information in ASR using a *context larger than the current sentence* to be recognized. Irie et *al.* (2019) train language models based on LSTM and transformers using long training sequences obtained by concatenation of sentences and study their robustness. Parthasarathy *et al.* (2019) focus on the ability of LSTM and transformer language models to learn context across sentence boundaries. Futami *et al.* (2020) exploit both left and right contexts of an utterance by applying *BERT* as an external language model through knowledge distillation. All these works show that it is relevant to rely on a broad context beyond sentence boundaries.

In previous works, Fohr and Illina, (2021) and Illina and Fohr, (2021) incorporated *sentence-level semantic information* (SI) into ASR. For this, the rescoring of the list of N-best hypotheses is carried out using distant contextual dependencies, which are important, especially for noisy conditions. In noisy parts of speech, the semantic model can help remove acoustic ambiguities. An efficient DNN architecture, based on *BERT*, and using semantic, acoustic and language model scores has been proposed. This model deals with pairs of N-best hypotheses to provide a pseudo-probability of the former being semantically more likely than the other. For example, in the following hypotheses for one sentence to recognize, taken from the TED-LIUM 3 corpus: "*hyp1: in antarctica we observe now a negative eyes balance"; "hyp2: in antarctica we observe now a negative ice balance",* the second hypothesis is more coherent semantically.

Compared to the work of Fohr and Illina (2021), the aim of the current paper is to extend this model. Two N-best rescoring approaches are proposed: the first one uses fine-grained information at the word token level; the second one relies on the previously recognized sentences. The combination of these two ideas is also studied. Compared with Ogawa *et al.* (2019), we use the *BERT* model that benefits from the pre-training on large corpora and not just on speech training corpus. Moreover, we exploit previous sentence information. In comparison to (Shin *et al.* (2019), where the *BERT* model computes word-level pseudo probabilities, we use the sentence prediction capability of the *BERT* model and the Generative Pre-Training Transformer (GPT-2) model (Radford *et al.,* 2019). Regarding Irie *et al.* (2019), where previous sentence information is used to improve the language model for the lattice rescoring, we integrate the information from the previous sentence into the *BERT*-based pairwise model for N-best re-ranking.

Our proposed approach using the previous sentence significantly outperforms the state-of-the-art GPT-2 rescoring and the rescoring model of Fohr and Illina (2021). This research work was carried out as part of an industrial project.

Contents

2. Proposed methodology

2.1 DNN based rescoring model

Methods, proposed in this article, are based on the methodology presented in (Fohr and Illina, 2021) where it is proposed to take into account the SI by rescoring the best hypotheses list of the ASR system. In this section, we give a brief overview of this methodology.

In this approach, to improve the ASR system, semantic model is introduced and combined with the acoustic probability $P_{ac}(h_i)$, the language model probability $P_{lm}(h_i)$, and the semantic score $P_{sem}(h_i)$, using specific weights α, β and γ:

$$\widehat{W} = argmax_{h_i \epsilon H} \; P_{ac}(h_i)^{\alpha} * P_{lm}(h_i)^{\beta} * P_{sem}(h_i)^{\gamma} \quad (1)$$

where h_i is a hypothesis from the N-best list H. The goal is to estimate the semantic score $P_{sem}(h_i)$ using a DNN-based model.

The rescoring is based on a comparison of ASR hypotheses, two per two to obtain a tractable size of the rescoring DNN input vectors. DNN rescoring model (denoted $BERT_{alsem}$) computes SI, associated with each pair of hypotheses.

For each hypothesis pair (h_i, h_j), during the training the expected DNN *output v* is: (a) *1,* if the WER of h_i is lower than the WER of h_j; (b) otherwise, 0.

The computation of $P_{sem}(h_i)$ is done as follows. For each hypothesis h_i of a given sentence, the cumulated score $score_{sem}(h_i)$ is evaluated. For this, for each pair of hypotheses (h_i, h_j) of the N-best list of this sentence: (a) the output value v_{ij} (between 0 and 1) is obtained by DNN model, which relies on the *BERT* model. A value v_{ij} close to *1* means that h_i is better than h_j. This value is used to compute the scores for these hypotheses; (b) the scores of both hypotheses are updated:

$$score_{sem}(h_i) \mathrel{+}= v_{ij}; \qquad score_{sem}(h_j) \mathrel{+}= 1\text{-}v_{ij}$$

We normalise the cumulated score $score_{sem}(h_i)$ by dividing by *N-1* and use it as *pseudo probability* $P_{sem}(h_i)$. The obtained value is combined with the acoustic and language model likelihoods (see eq. (1)). Finally, the hypothesis with the best score is chosen as the recognized sentence.

2.2 $BERT_{alsem}$ rescoring model

In this section, we recall the architecture of the $BERT_{alsem}$ model from (Fohr and Illina, 2021), used as the starting point for the current work. *alsem* denotes "Acoustic, Linguistic and SEMantic" information, because we use *acoustic* and *textual information*. The advantage of this model is that the relative importance of acoustic, language model, and SI is learned together to provide a powerful model.

In Figure 1 (without the dotted block), the text of the pair of hypotheses (h_i and h_j) is given to the *BERT* model. The outputs of *BERT* are given to a bi-LSTM layer, max pooling, average pooling, and then to a fully connected (FC) layer with a ReLU (*Rectified Linear Unit*) activation function (Nair and Hinton, 2010). The output of this FC layer, the acoustic probabilities, and language model probabilities are concatenated. The final FC layer (followed by a sigmoid activation function) computes output v_{ij}.

2.3 Fine grained rescoring model $BERT_{alsem\text{-}fg}$

This section presents the first rescoring method proposed in this paper. The objective is to provide $BERT_{alsem}$ model with fine-grained information (at the word token level and not just at the sentence level as in $BERT_{alsem}$). We would like to integrate the probability of each word token of a given hypothesis. This value represents the probability of a token given *all previous tokens* of the hypothesis. For a pair of hypotheses, two vectors of token probabilities are generated, one for each hypothesis (see Figure 1, dotted part). Each vector is assigned as input to a neural network layer. Since such a vector is a temporal sequence, bi-LSTM or CNN are the best suited to process this type of information and to obtain a fixed length vector. The outputs of these two layers (one for each hypothesis) are concatenated with the acoustic and language model scores, and SI of the hypothesis pair is calculated by *BERT*. This concatenation is passed through an FC layer followed by a sigmoid activation function. Finally, the output v_{ij} of this network is obtained. We call this model *fine-grained* $BERT_{alsem\text{-}fg}$.

To estimate the probability of each word token of a given hypothesis, GPT-2 is used. The first advantage of using GPT-2 is its attention mechanisms allowing the model to selectively focus on the most relevant word tokens. The second potential advantage is to provide complementary information compared to the *BERT* model included in $BERT_{alsem}$.

2.4 Rescoring using previous sentences $P\text{-}BERT_{alsem}$

This part focuses on the second proposed method taking into account the ASR output of the previously recognized sentences for improving the recognition of the current sentence. We would like to combine the SI of one or more previous sentences with the SI of the current sentence. Indeed, the SI contained in the previous sentences and in the current sentence of a discourse are related (Irie *et al*., 2019). This relationship can link some words from the previous sentences with words from the current sentence. Our objective is to take into account these semantic relations to select the best hypothesis.

The proposed rescoring model using the previously recognized sentences is denoted $P\text{-}BERT_{alsem}$. Compared to the $BERT_{alsem}$, we added the words from the previously recognized sentences to *each hypothesis of a hypothesis pair*. This information is given to the *BERT* model. Concerning the acoustic and language model information part of $BERT_{alsem}$, we modify the language model probabilities by replacing them with the conditional probabilities $P_{lm}(h_i \mid prev\_sent)$ and $P_{lm}(h_j \mid prev\_sent)$. The acoustic probabilities are unchanged.

2.5 Combined rescoring model $P\text{-}BERT_{alsem\text{-}fg}$

The two proposed approaches perhaps contain complementary information and can be combined into a single model, denoted $P\text{-}BERT_{alsem\text{-}fg}$. In this model, for a given pair of hypotheses, the model input is composed of $P_{ac}(h_i)$, $P_{ac}(h_j)$, $P_{lm}(h_i|prev\_sent)$, $P_{lm}(h_j|prev\_sent)$, text of each hypothesis preceded by the text of the previously recognized sentences. The rest of the methodology is unchanged.

3. Experimental conditions

3.1 Corpus description

We use the publicly available TED-LIUM 3 corpus (Fernandez *et al.,* 2018), containing recordings from TED conferences. Each conference of this corpus focuses on a particular subject; thereby the data are well suited to our study. The train, development, and test partitions are provided within the TED-LIUM 3 corpus: (a) train: 2,351 talks, 4.8M words, 452h; (b) development: 8 talks, 17,783 words, 1h36; (c) test: 11 talks, 27,500 words, 2h37. We use the development set to choose the best parameter configuration, and the test set to evaluate the proposed methods with the best configuration.

In this paper, the study of the ASR in noisy conditions was performed because this work is a part of an industrial project (noisy ASR, more precisely in fighter aircrafts). We add noise to the train, development and test sets to get closer to the actual conditions of an aircraft. For the train part, we add different noises from NOISEX-92 corpus (Varga and Steeneken, 1993)

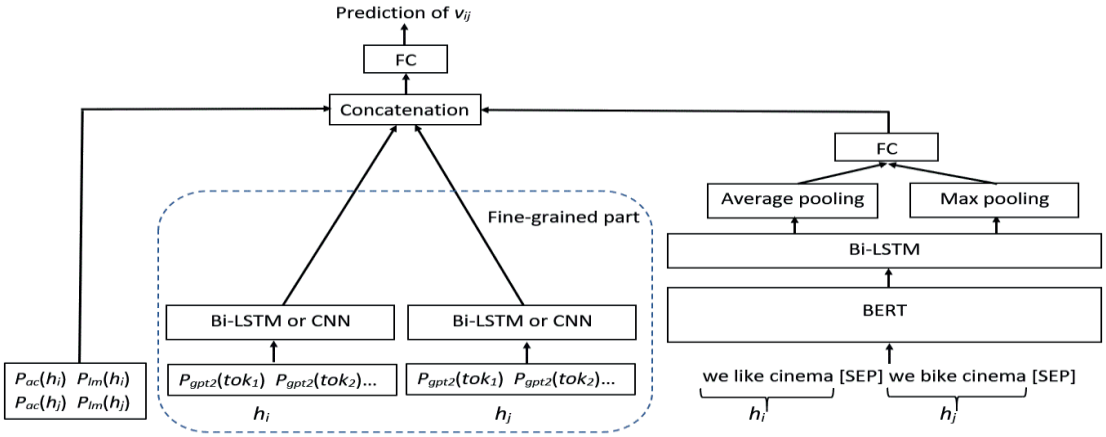


Fig. 1: Architecture of the proposed $BERT_{alsem\text{-}fg}$ rescoring model

(excluding F16 noise, used for development and test data) at SNR from 0 to 20 dB. We keep the size of the training set unchanged (no training data augmentation). Furthermore, we evaluate the proposed approaches in clean conditions.

3.2 Speech recognition system

Our recognition system is based on the *Kaldi speech recognition toolbox* (Povey *et al.*, 2011). Time Delay Neural Network (TDNN) (Waibel *et al*., 1989) triphone acoustic models are trained on the training part of TED-LIUM 3 using sMBR training (*State-level Minimum Bayes Risk*) (Kingsbury, 2009). The lexicon and language models were provided in the TED-LIUM 3 distribution. The lexicon contains 150k words. The LM used for the lattice generation has 2 million 4-grams and was estimated from a textual corpus of 250 million words. We perform N-best list generation with a more powerful LM: the RNNLM model (LSTM) (Sundermeyer *et al.,* 2012). Since this DNN model only compares two hypotheses and cannot *output* the word probabilities, it is not possible to calculate the perplexity of this model. We compute the word error rate (WER) to measure the performance.

It is worth noting that in (Fohr and Illina, 2021) the acoustic model was trained only on clean speech. In the current work, we carry out training on noisy data. The obtained model is more accurate for noisy ASR and the WERs are lower than in (Fohr and Illina, 2021).

3.3 Rescoring models

We have chosen to use an N-best list of 20 hypotheses in all experiments (Illina and Fohr, 2021). During the training of the proposed models, we do not use the hypothesis pairs which obtain the same WER. When evaluating (development and testing), we consider all hypothesis pairs, because the WERs are not available for these hypotheses.

For each model, the combination weight values α, β, and γ achieving the best rescoring performance on the development set are selected as the optimal value for the test data. For all the experiments, optimal values of the combination weights are: α=1, β is between 8 and 10, and γ is between 80 and 100. This large difference between the values is explained by the fact that we use likelihood or pseudo probabilities that are not normalized.

For our semantic models, we downloaded Google's pre-trained *BERT* model (110M parameters, 12 layers, and the size of the hidden layers is 768) (Turc *et al.*, 2019). We use the Adam optimizer (Kingma and Ba, 2015) and binary cross-entropy loss function.

We iterate the training of $BERT_{alsem}$ as follows: during the first epoch, the layer weights of the *BERT* model are frozen, and during the following epochs all *BERT* weights are updated. The dropout is 30 %. Two methods could be employed to use *BERT* with application-specific data: *masked LM* and *next-sentence prediction*. We use next-sentence prediction because we put two hypotheses as input to the *BERT* model (see figure 1, right part).

We downloaded pre-trained GPT-2 LM from the Hugging Face site. The model has 117M parameters and was trained by OpenAI on 40GB of Internet text. In our experiments, this model is used for several purposes: (a) as a language model $P_{lm}(h_i)$ during N-best rescoring (see eq. (1)); (b) inside $BERT_{alsem}$ to represent the language model score $P_{lm}(h_i)$ of each hypothesis (see figure 1); (c) inside $BERT_{alsem\_fg}$ to compute the score of each word token $P_{gpt2}(tok)$ of each hypothesis; (d) inside $P\text{-}BERT_{alsem}$ to compute $P_{lm}(h|prev\_sent)$. In all configurations, GPT-2 is fine-tuned on the transcriptions (references) of the train part of TED-LIUM 3.

In our preliminary experiments, during N-best rescoring, a Masked Language Model (MLM) (Salazar *et al.,* 2020) performed worse than GPT-2 and therefore the results will be not presented here.

4. Experimental results

We report the WER for the development and test sets of TED-LIUM 3 in clean speech and under noisy conditions (noise added at 10 and 5 dB). We recall that the acoustic model is trained on noisy speech. In Table 1, different notations are introduced: (a) ***Random*** represents the random selection of the recognition result from the N-best hypotheses (without using a rescoring model); (b) ***Baseline*** corresponds to the standard speech recognition system (without using a rescoring model); (c) ***Oracle*** gives the *maximum performance* that can be obtained by selecting in the N-best hypotheses: the hypothesis which minimizes the WER for each sentence is chosen; (d) ***GPT-2 resc***. corresponds to a state-of-the-art rescoring based on the fine-tuned GPT-2 model. We perform this rescoring to fairly compare the proposed transformer-based models to a state-of-the-art transformer-based model introducing long-range context dependencies. We rescore N-best hypotheses using eq. (1), where P_{lm} *(h)* is computed by the GPT-2. The semantic model is not used (γ=0); (e) ***$BERT_{alsem}$ with GPT-2 resc*** (Fohr and Illina, 2021) is performed to compare the

	Methods/systems	SNR 5 dB		SNR 10 dB		no added noise	
		Dev	Test	Dev	Test	Dev	Test
1	Random system	15.1	19.4	10.8	13.9	9.2	11.0
2	Baseline system	13.6	17.1	8.6	10.9	6.9	7.4
3	Baseline system with GPT-2 resc.	11.6	14.6	7.3	8.9	5.8	6.0
4	$BERT_{alsem}$ with GPT-2 resc. (Fohr and Illina, 2021)	11.4	14.5	7.1	8.9	5.6	5.9
5	$BERT_{alsem\text{-}fg}$ (CNN) with GPT-2 resc.	11.5	14.5	7.1*	8.8*	5.6*	5.9
6	$BERT_{alsem\text{-}fg}$ (bi-LSTM) with GPT-2 resc.	11.4*	14.5	7.1*	8.8*	5.6*	5.9
7	$P\text{-}BERT_{alsem}$ with GPT-2 resc, 1sent	11.2*~	14.3*	6.9*~	8.5*~	**5.3***~	**5.7***~
8	$P\text{-}BERT_{alsem}$ with GPT-2 resc,1sent30w, stop wrds remov.	**11.1** *~	**14.2***~	6.9*~	**8.4***~	**5.3***~	**5.7***~
9	$P\text{-}BERT_{alsem}$ with GPT-2 resc, 2sen30w, stop wrds remov.	11.2 *~	**14.2***~	**6.8***~	8.5*~	**5.3***~	**5.7***~
10	Oracle	9.5	11.3	5.4	6.1	4.0	3.6

Table 1. ASR WER (%) on the TED-LIUM 3 development and test sets, SNR of 10 and 5 dB, 20-best hypotheses. "*" denotes significantly different result compared to GPT-2 resc. configuration (line 3). "~" denotes significantly different result compared to $BERT_{alsem}$ with GPT-2 resc. configuration (line 4)

transformer-based models, proposed in this paper, with $BERT_{alsem}$ proposed by Fohr and Illina (2021). It corresponds to the rescoring of the N-best hypotheses using eq. (1) with $P_{sem}(h)$ given by $BERT_{alsem}$ and $P_{lm}(h)$ given by the GPT-2. The other lines of Table 1 give the performance of the proposed approaches. The best results are presented in bold.

For the rescoring models proposed in this article, we studied three configurations: (a) ***$BERT_{alsem\text{-}fg}$ with GPT-2*** represents rescoring using $BERT_{alsem\text{-}fg}$. Configurations with CNN and bi-LSTM models are shown; (b) ***$P\text{-}BERT_{alsem}$ with GPT-2*** gives the results for the approach taking into account the previous sentence; (c) ***$P\text{-}BERT_{alsem\text{-}fg}$ with GPT-2***: the combined model gives no additional improvement compared to ***$P\text{-}BERT_{alsem}$ with GPT-2*** and the results are not presented in this paper.

For $P\text{-}BERT_{alsem}$ rescoring model, to avoid the overflow of the number of the *BERT* input tokens, we use at most *M* last words from the previous sentence (*M=30*). Nevertheless, to compute $P_{lm}(h|prev\_sent)$ with the GPT-2, the whole previous sentences are used.

To analyse the results, we make the comparisons with: (a) the state-of-the-art rescoring model with GPT-2 (line 3); (b) the best configuration of $BERT_{alsem}$ rescoring model (line 4, (Fohr and Illina, 2021)).

The *significance of the results* is indicated by "*" in Table 1 compared to line 3, and by "~" compared to line 4. The confidence interval at the 5% significance level is calculated using the matched pairs test (Gillick and Cox, 1989), considering the effects of two different treatments (algorithms) on equivalent subjects (speech segments) aligned by a dynamic programming algorithm.

$BERT_{alsem}$ rescoring model. By studying the results of $BERT_{alsem}$ (line 4), we see that the conclusions given by Fohr and Illina (2021) are still valid when the noisy acoustic model is used: the $BERT_{alsem}$ provides consistent WER reduction compared to the baseline model with GPT-2 rescoring (line 3).

Fine-grained rescoring model: ***$BERT_{alsem\text{-}fg}$*** The $BERT_{alsem\text{-}fg}$ shows an improvement over the baseline system with GPT-2 rescoring (line 6 versus line 3, the significance is indicated by "*" in Table 1). The CNN architecture (line 5) shows similar results to the bi-LSTM one (line 6).

The proposed $BERT_{alsem\text{-}fg}$ displays a similar performance as $BERT_{alsem}$ (lines 5, 6 versus line 4). This means that probably adding fine-grained information (GPT-2 probabilities at the word token level) does not bring complementary information compared to the $BERT_{alsem}$ model. It is difficult to predict whether pre-trained GPT-2 and *BERT* models contain complementary information because these two models are learned on different corpora but are based on the same principle (Transformers).

Rescoring model using previous sentences: ***$P\text{-}BERT_{alsem}$*** The lines 8 and 9 display the results for our $P\text{-}BERT_{alsem}$ model. We use one or two previous sentences. Two configurations were studied: using M words of previous sentences or using only non-stop M words of the previous sentences (stop words contain little semantic information and were removed). The results for the second case are slightly better, therefore we present only the results for the second case.

Statistically significant improvements are observed for all noise levels and clean speech for the $P\text{-}BERT_{alsem}$ compared to the *GPT-2 resc* configuration (lines 8 and 9 versus line 3, the significance is indicated by "*"). Comparing $P\text{-}BERT_{alsem}$ with $BERT_{alsem}$ model (Fohr and Illina, 2021) (without the previous sentence information, lines 8 and 9 versus line 4), we see that the integration of the previous sentence information helps in the selection of the best hypothesis. This improvement is significant in almost all configurations (the significance is indicated by "~" in Table 1).

Analysing the results of $P\text{-}BERT_{alsem}$, we observe that the model corrects syntactic and semantic errors, compared to $BERT_{alsem}$ (lines 8 and 9 versus line 4). Here is one example of a semantic error corrected by $P\text{-}BERT_{alsem}$ model for 5dB noisy condition, test set:

ref: *I got to **lhasa** that i understood the face behind the statistics you hear about six thousand sacred monuments…*

hyp1: *I got to **loss** that i understood the face behind these statistics you hear about six thousand sacred monuments…*

hyp2: *I got to **lhasa** that i understood the face behind these statistics you hear about six thousand sacred monuments…*

The second hypothesis is selected as the sentence recognized by $P\text{-}BERT_{alsem}$ because the previous sentence contains the word "Tibet".

Using one or two previous sentences (lines 8 and 9) gives similar results. We performed the experiments using three previous sentences and obtained no improvement. The results are not given here.

In conclusion, the best system $P\text{-}BERT_{alsem}$ gives between 1% and 3% relative WER reduction compared to $BERT_{alsem}$ rescoring model (Fohr and Illina, 2021) (lines 8, 9 versus line 4). These improvements are *statistically significant* according to the matched pairs test (Gillick and Cox, 1989) (see "~" in Table 1).

5. Conclusion

The aim of this article is to improve ASR in clean and noisy environments. In the framework of the pairwise rescoring of ASR N-best hypotheses, we would like to enrich the rescoring

model $BERT_{alsem}$. We have introduced two rescoring approaches, based on semantic representations. The first one is designed to integrate fine-grained information at the word token level. The second one exploits the context beyond the current utterance by considering the previously recognized sentences. The proposed models are based on DNN, *BERT,* and GPT-2 models. Experimental evaluation, carried out on TED-LIUM 3 corpus with clean and noisy speech, showed that the approach using one previous sentence gives a statistically significant improvement, outperforming the state-of-the-art rescoring using the advanced GPT-2 model and previous work of Fohr and Illina (2021) in almost all configurations.

References

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z. (2007). DBpedia: A Nucleus for a Web of Open Data. *The Semantic Web Lecture Notes in Computer Science*, Volume 4825/2007, pp. 722-735.

Devlin, J., Chang, M.-W. and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT*.

Fernandez, H., Nguyen, H., Ghannay, S., Tomashenko, N. and Esteve, Y. (2018). TED-LIUM 3: Twice as Much Data and Corpus Repartition for Experiments on Speaker Adaptation. *Proceedings of the SPECOM,* pp. 18–22.

Fohr, D. and Illina, I. (2021). BERT-based Semantic Model for Rescoring N-best Speech Recognition List. *Proceedings of Interspeech*.

Futami, H. Inaguma, H., Ueno, S., Mimura, M., Sakai, S. and Kawahara, T. (2020). Distilling the Knowledge of BERT for Sequence-to-Sequence ASR. *Proceedings of Interspeech*.

Gillick, L. and Cox, S. (1989). Some Statistical Issues in the Comparison of Speech Recognition Algorithms. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, vol. 1, pp. 532–535.

Illina, I. and Fohr, D. (2021). DNN-based semantic rescoring models for speech recognition. *Proceedings of the Internat. Conference on Text, Speech and Dialogue*, TSD.

Irie, K., Zeyer, A., Schluter, R. and Ney, H. (2019). Training Language Models for Long-span Cross-Sentence Evaluation. *Proceedings of IEEE Automatic Speech Recognition & Understanding*, ASRU.

Kingma, P. D. and Ba, J. (2015). A Method for Stochastic Optimization. *International Conference on Learning Representations*.

Kingsbury, B. (2009). Lattice-based optimization of sequence classification criteria for neural-network acoustic modelling. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pp. 3761–3764.

Kumar, A., Morales, C., Vidal, M.-E., Schmidt, C. and Auer, S. (2017). Use of Knowledge Graph in Rescoring the N-best List in Automatic Speech Recognition. *arXiv:1705.08018v1*.

Level, S., Illina, I. and Fohr, D. (2020) Introduction of Semantic Model to Help Speech Recognition, *Proceedings of the International Conference on Text, Speech and Dialogue*, TSD.

Li, J., Deng, L., Gong, Y. and Haeb-Umbach, R. (2014). An Overview of Noise-robust Automatic Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777.

Liu, L., Gu, Y., Gourav, A., Gandhe, A., Kalmane, S., Filimonov, D., Rastrow, A. and Bulyko, I. (2021). Domain-Aware Neural Language Models for Speech Recognition. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. *ICML*.

Ogawa, A., Delcroix, M., Karita, S., and Nakatani, T. (2018). Rescoring N-best Speech Recognition List Based on One-on-One Hypothesis Comparison Using Encoder-Classifier Model. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP.

Ogawa, A., Delcroix, M., Karita, S., and Nakatani, T. (2019). Improved Deep Duel Model for Rescoring N-best Speech Recognition List Using Backward LSTMLM and Ensemble Encoders. *Proceedings of Interspeech*.

Parthasarathy, S., Gale, W., Chen, X., Polovets, G. and Chang, S. (2019). Long-span Language Modeling for Speech Recognition. *CoRR abs/1911.04571*.

Radford, A., Wu, J., Child, R., Luan, Amodei, D. and Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. *Technical Report OpenAI*.

Salazar, J., Liang, D., Nguyen, T. and Kirchhoff, K. (2020). Masked Language Model Scoring. *Proceedings of ACL*.

Shin, J., Lee, Y. and Yung, K. (2019). Effective Sentence Scoring Method Using BERT for Speech Recognition. *Proceedings of ACML*.

Song, Y., Jiang, D., Zhao, X., Xu, Q., Wong, R., Fan, L. and Yang, Q. (2021). L2RS: a Learning-to-rescore Mechanism for Automatic Speech Recognition. *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 1157–1166.

Sundermeyer, M., Schluter, R. and Ney, H. (2012). LSTM Neural Networks for Language Modeling. *Proceedings of Interspeech*.

Turc, I., Chang, M.-W., Lee, K. and Toutanova, K. (2019). Well-Read Students Learn Better: On the Importance of Pre-training Compact Models. *arXiv:1908.08962v2*.

Varga, A. and Steeneken, H. (1993). Assessment for automatic speech recognition II. NOISEX-92: A Database and an Experiment to Study the Effect of Additive Noise on Speech Recognition Systems. *Journal Speech Communication*, Volume 12, Issue 3, pp. 247-251.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K. and Lang, K. (1989). Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 3, pp. 328–339.

Wang, A. and Cho, K. (2019). BERT has a Mouth, and it Must Speak: BERT as a Markov Random Field Language Model. *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation,* pages 30–36.

Xu, L., Gu, Y., Kolehmainen, J., Khan, H., Gandhe, A. Rastrow, A. Stolcke, A., Bulyko, I. (2022). RescoreBERT: Discriminative Speech Recognition Rescoring with BERT. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP.

Zhao, Y., Yang, X., Wang, J., Gao, Y., Yan, C. and Zhou, Y. (2021). BART based Semantic Correction for Mandarin Automatic Speech Recognition System. *Proceedings of Interspeech*.

I'm Smarter than the Average BERT! – Testing Language Models Against Humans in a Word Guessing Game

Balázs Indig[1,2], **Dániel Lévai**[3]

[1]Eötvös Loránd University, Department of Digital Humanities
[2]National Laboratory for Digital Heritage
[3]The Upright Project
indig.balazs@btk.elte.hu, levai753@gmail.com

Abstract

Even though language modelling in recent years is a constantly evolving topic in natural language processing and new generation models are used to solve all traditional downstream tasks accurately, their operation is very complex and we know as little part of it as of human thinking. We compare several language models (n-gram and vector-based ones) to gain insight into how the mental language model relates to them. The task is a word guessing game similar to the cloze-test, in which the word in question has to be guessed based on one or more different contexts (e.g. concordance). This task is the same task in which neural language models are trained. We gathered a small amount of data from human participants, which led us to several surprising results and conclusions. In this paper, we use Hungarian, a moderately agglutinating language, but with our code experiments can be conducted for any language equally well.

Keywords: cloze-test, language modelling, word context, BERT, Word2Vec, KenLM

1. Introduction

The last decade in NLP is dominated by the new generation language models. This revolution has come to the point where GPT-3 (Elkins and Chun, 2020) can generate text that is indistinguishable from human-created ones.

In this paper we utilise a form of *Cloze-test* (Taylor, 1953) as this task can be often found in language comprehension tasks and exams to evaluate language learners' capabilities and neural language models are trained on this task as well. We slightly modified the task: missing words can have left and right context with specifiable length – we call this an example – and by adding other examples to help the guessing we are actually creating a concordance for the missing words (Indig and Lévai, 2022).

To test it with human subjects we implemented a game which can be played alone or in two player mode against a machine opponent (language model). Using this game we could create multiple experimental setups in a versatile manner. We think this gamification approach as a platform could be great starting point for further psycholinguistic experiments on human language comprehension as well as automatically comparing the comprehension of artificial language models from a less studied perspective.

While writing and speaking happen by advancing forward in time monotonically, in reading and hearing we can observe delay and non-monotonous eye movements (Laubrock and Kliegl, 2015). This asymmetry has preoccupied many researchers over the past decade.

Incremental parsers which try to utilise the left context only deal with loss of accuracy, which can be mostly restored by using a look-ahead window (Prószéky et al., 2014) or abandoning strong monotonicity (Köhn and Menzel, 2014). These results fit into the contemporary trend which can be summarised by (Indig et al., 2016) and culminated in the modern language models which use 4-4 tokens from the left and right context (9-gram in total) as a safe default for convenience (Collobert et al., 2011).

2. The Different Language Models

N-gram language models like *KenLM* (Heafield, 2011) use many tricks to handle the data sparsity problem due to the primitive nature of n-grams[1]. It is perfect to showcase the practical maximum which an n-gram model can achieve in an environment with constrained contexts. The new generation embedding-based language models change n-grams to high-dimensional vector representation of words and their contexts. This abstraction solves the data sparsity via implicit SVD (Levy and Goldberg, 2014) and the word ordering problem at once with the *continuous bag-of-words model (CBOW)* with very fast computation times (Mikolov et al., 2013b). To test these non-contextual neural models (Pennington et al., 2014) we used *Gensim* (Rehurek and Sojka, 2011).

The de facto standard of such models nowadays is *BERT* (Devlin et al., 2019), a multi-layer, bidirectional transformer-based encoder model (Vaswani et al., 2017). As an encoder it creates word embedding: it assigns a vector for every word–context pair and tries to guess the masked word similar to language models, but accepts arbitrary length contexts as input. As being sensitive to complex, long-range relationships (Goldberg, 2019), this model is ideal for guessing the missing word.

An important common property of these models is that the user has to define the used context size – or the order of the used n-grams – for the trained model without any solid evidence to aid the decision (Collobert et al., 2011).

3. The Used Corpora

For test corpus, our main candidate was *Hungarian Webcorpus 2.0* (Nemeskey, 2021b) which the first Hungarian BERT model called *huBERT* (Nemeskey, 2021b; Nemeskey, 2021a) was created from, but as we tested

[1]For example *Word2Vec* (Mikolov et al., 2013b) solves this problem with the high-dimensional vectorspaces.

	Orig. no. of Sentences	Filtered no. of Sentences	%	Orig. no. of Words	Filtered no. of Words	%
Webcorpus 1.0	42,482,107	13,915,132	32.75	589,080,971	272,544,786	46.26
Webcorpus 2.0	589,398,448	199,627,778	33.86	9,217,857,283	4,036,428,613	43.78

Table 1: The size of the selected corpora before and after filtering. About one third of the sentences remains with about the half of the original size for both corpora which is quite surprising considering the lax filtering rules.

huBERT the underlying corpus could be used for training purposes only. We needed another corpus for testing which has minimal or no text in common with Hungarian Webcorpus 2.0. Nowadays most of the publicly available corpora come from the currently available web pages. This fact ruled out several other corpora. However, the *Hungarian Webcorpus 1.0* (Halácsy et al., 2004) had the highest possibility of differing text content, due to its age.

We cleaned the selected corpora (see Table 1.) to filter all possible garbage that might spoil the user experience or would give an advantage to the machine player (see details the repository). From the untokenised form of sentences we generated contexts for words which contained 4 to 40 lowercase letters only, and kept those which could be analysed with the *emMorph* morphological analyser (Novák et al., 2016) to eliminate non-words. We kept KWICs with at least 30 unique context and random sampled them to balance all KIWC to 30 occurrences, before selecting 8 000 random KWIC with 240 000 unique context for the *context banks* for each corpus for comparison.

4. The Setup of the Experiment

All models were given a fixed word list to choose from. This list was created from the 3 million most frequent words of the Hungarian Webcorpus 2.0. The test examples were sampled from the Hungarian Webcorpus 1.0 to avoid training on the test set.

4.1. KenLM

We trained a 5-gram model and to reduce the memory requirements, we pruned the low frequency n-grams to the following lower frequency boundary: 2-grams to 4, 3-grams to 9, 4-grams to 16 and 5-grams to 25.

The model filters the candidate words with matching size from the aforementioned list. For each remaining word, the program inserts the word to all displayed examples and computes the log probability for the resulting word sequences and sums up the results to get the joint log probabilities. The guess will be the candidate word with the highest joint log probability.

4.2. BERT (huBERT)

No training is needed for huBERT, as it was trained on the Hungarian Webcorpus 2.0. BERT uses the *WordPiece* (Schuster and Nakajima, 2012) tokenizer algorithm which segments words into subwords. We must help the guesser and tell how many subwords it should search for.

To optimise the running time, we tokenized each candidate word to know how many tokens they contain. We built a trie where the edges are the words and the vertices are defined by inclusion relation of the edges they connect. On guessing a multi-subword word, we choose all the vertices for the specified depth, we compute the probabilities for the subwords which adds up each word and multiply them for each vertex. BERT does not handle joint-probability of examples.

4.3. Context-free Word Embedding

We experimented with multiple models: CBOW, skip-gram (Mikolov et al., 2013b) and *FastText* (Bojanowski et al., 2017). The Gensim library's function to predict missing words gave very wild guesses so we ceased further experiments with this model. This is not surprising because these kind of models are used mainly for word analogy (Mikolov et al., 2013a) and in downstream tasks. This gave us the idea to keep the FastText model as a helper for the human player as it can yield usable information on the analogical similarity between the guessed and missing words.

5. Results

We will go through how the models perform against each other with regards to different context sizes and sides, and present our findings on the human vs. computer cases.

In case of the computer models, we are interested in two questions: *what is the minimal context size based on which the models can guess the missing word*; the second one is *how many contexts are needed for a given word and context size in order to guess the missing word*.

5.1. Two-sided Contexts

We measured the *smallest size of the context needed* to correctly predict the target word (see Figure 1). First and foremost, KenLM performed a lot worse than the BERT, and the number of cases where the KenLM could not guess correctly while the BERT could was very low (4.4%). In addition to this, BERT guessed with any context size much more words (65.0% vs. 32.1%).

KenLM hardly improves when given a third or a fourth context word (7- and 9-gram), meaning that the words in this distance provide a very low amount of information for the (5-gram) model. In contrast to this, BERT steadily improves as the context size increases – it can use the information provided by long distance relationships, which is a well-known property of BERT-like architectures.

In our second measurement, we compared *how many contexts the models need* to guess the missing word with the restriction that they can only guess each word once, effectively limiting the vocabulary. At every step, we appended another context for the previous contexts and the models guessed based on every context (see Figure 2.). KenLM cannot use the information provided by multiple

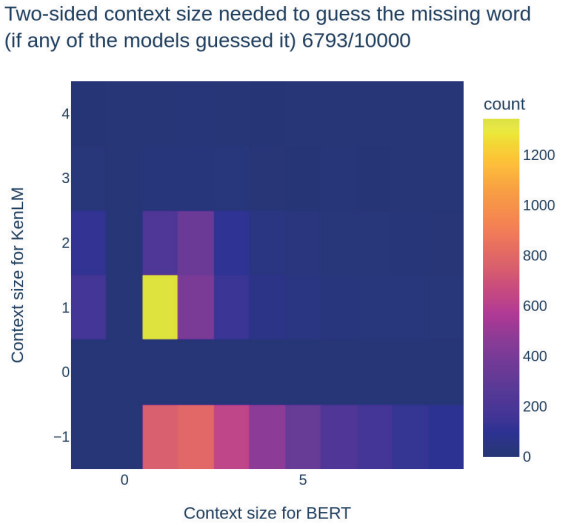


Figure 1: The context size needed for each model. -1 means that the model did not guess the target word with any context size, the 6793/10000 in the title means that from the 10000 word–context input, only 6793 have been guessed by at least one model. The peaks are reached at 1 long contexts (both sides + word = 3 gram) for both models, but BERT has slow decay (fading rows), KenLM's maximum is at 2 (columns top at 2, i.e. 5-grams).

contexts – only in 6.9% of the cases guessed the missing word, compared to the 33.6% of cases in case of BERT.

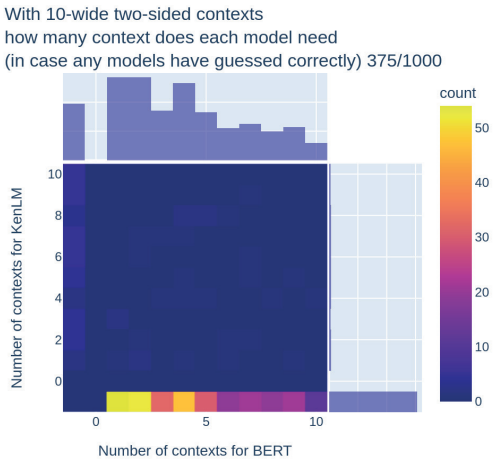


Figure 2: The number of contexts needed for each model. -1 means that the model did not guess the target word with any number of contexts, the 375/1000 in the title means that we are only showing 375 concordances – the rest has not been guessed by any of the models thus are omitted from the figure. BERT reached the peak at 1-2 contexts (last row), while KenLM performed poorly (columns).

The models return the 10 most likely words. In case of BERT, the guesses show a considerable variation as the number of contexts grows. The guesses of KenLM often "freeze" (do not change): the guess after the first context can only be correct if the target word was already among the top 10 guesses, since the the most likely word in the list of guesses was discarded after each guess.

The number of correct guesses was much lower in the case of already 10-long contexts (see Figure 2.) compared to the increasing context lengths (see Figure 1). This could be explained by the method we used to select the missing words – we opted for high-frequency words, which in turn have common neighbouring words, thus it is easier for the models to guess the missing word from their neighbours only, while in case of the longer contexts, BERT looked contexts at their entirety and guessed less common words.

Based on this, we examined whether the target word was among the top 10 guesses. In case of KenLM, if the target was not among the top 10 guesses for the first context, it only appeared in the later context with a percentage of 1.3% – in case of BERT, the target word appeared in 17.5% of the cases. It is clear that BERT benefits greatly from the increasing number of contexts, as for KenLM, the probability distribution of guesses is so flat that the guesses hardly change after the first one or two contexts.

5.2. One-sided Contexts

Similarly to the two-sided contexts, we measured the *minimal context size needed* and the *number of the contexts needed* for each model. The main observations are similar for both models: the performance of both models are mostly independent of the side. The BERT model (see Figure 3.) outperformed the KenLM by a large margin. For all of the correct guesses, about half of them could only be guessed from a left-sided context, the other half from a right-sided context, and the overlap between two groups, e.g. the number of words that could be guessed both from left and right contexts independently was lower than expected. This overlap for BERT was 23% (if the target word could be guessed from either side), meaning that 23% could be guessed from both left-sided and right-sided context individually. KenLM was more sensitive to the side: only 9% of the correct words could be guessed from both left-sided and right-sided contexts separately.

We experimented with concordances with shorter one-sided contexts to compare the effect of multiple concordances, this can be seen on Figure 4. We would expect a steady decrease in newly guessed words – but this is not the case for BERT. There is an inflection point around the third-fourth context, until that point, the number of newly guessed ones increase, and after that point, they decrease – showing that for the BERT model, the third and fourth contexts give the most additional information on the target word. This shows that the BERT can not only detect and use longer relations, but it can gather information from multiple contexts as well.

5.3. Human Evaluation

After seeing these surprising results, we have chosen 500 contexts with 8-wide context on both sides from the context bank and inflated it to 3000 by taking the 4-wide one-sided contexts, the 8-wide one-sided contexts, and

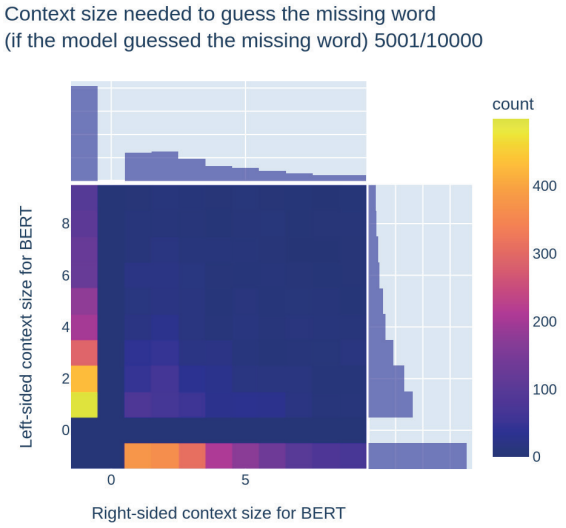


Figure 3: The context size needed for BERT on a given side (right side = rows, left side = columns). -1 means that the model did not guess correctly with any context size. There is hardly any overlap between the sides (both axes with positive value and good guesses), meaning that the target words are only guessable from one side.

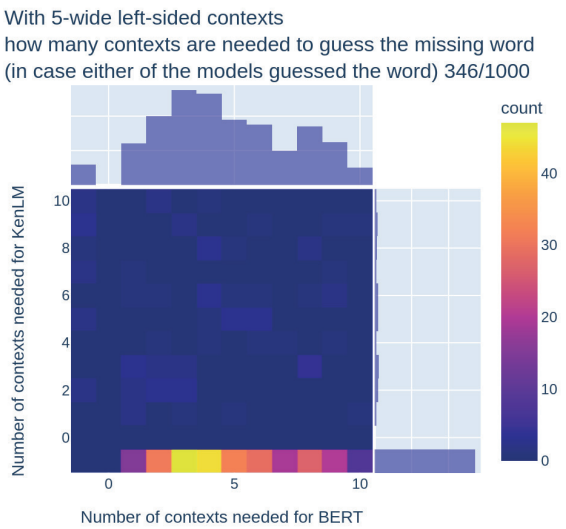


Figure 4: How many contexts are needed for each model. -1 means that the model did not guess correctly with any number of contexts. It can be clearly seen that the words KenLM guessed are roughly the subset of those that BERT guessed (the first column's values are very low).

the 4+4 and 8+8-wide two-sided contexts. The human evaluators received randomly from this context bank with uniform distribution. In the end we have received 1258 guesses on 550 contexts from 8 evaluators out of the 3000 contexts we have chosen, thus we were able to evaluate it against the language models - with the caveat that we are unable to make quantitative analysis on the different sides and widths. Figure 5. shows the number of guesses it took to guess the correct word for each player. BERT guessed the most contexts correctly (33.4%), the second one is the human evaluation (23.2%), the last one is KenLM with 22.2%. We can also see that humans did not make 10 guesses in general – in most cases, they gave up after 2-3 guesses, so this comparison is not entirely fair.

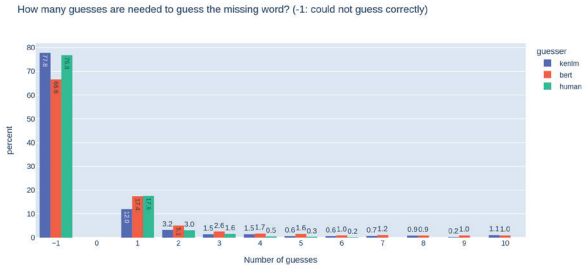


Figure 5: The number of correct guesses needed. -1 means that the model/human did not guess the word correctly.

Figure 6. shows the distribution of the cosine similarities to the target word until the correct guess – meaning that the correct guess was excluded (since it has a similarity of 1). It is well known that the cosine similarity in word embeddings strongly correlate with the semantic similarity (Chandrasekaran and Mago, 2021). This plot's aim is to quantify the *semantic correctness* of the guesses based on the cosine similarity between the guess and the target word. We can see that the distribution of human guesses shown in green is skewed to the right, indicating that the guesses made by humans had higher cosine similarity to the target word, therefore the guesses are more semantically similar to the target word.

We can also see that humans guessed better when normalised over the number of guesses rather than over the number of contexts (as in Figure 5.) – meaning that the human evaluators guessed less, but more correctly. It is important that the marginal box plots are notched, indicating statistical significance between the median cosine similarities: the humans' guesses are more related to the target word by a statistical significant margin.

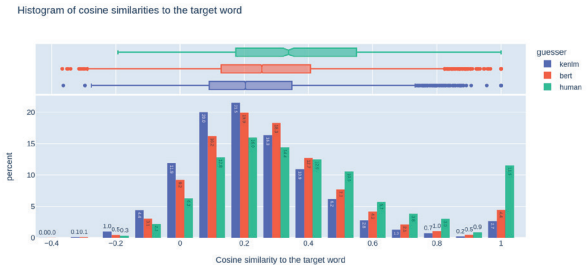


Figure 6: The distribution of cosine similarities. There is a notched boxplot marking the quartiles and the significance of the medians.

6. Conclusion and Discussion

We created a platform and conducted multiple experiments with open source code base[2]. We compared different artificial language models to the human language comprehension procedure under different constraints.

Context-free word embedding are suitable only for quantifying analogical similarity, but not for guessing. The guesses of the examined n-gram model do not improve over 3-gram context, while the BERT-based guesser improved slightly with slow decay. Contextual word embedding models were hardly able to improve their initial guesses by using the other examples, while the n-grams were not able to improve them at all. In the question of left and right contexts, head-to-head results were obtained, there was little overlap between the sides: if one side of the context could guess the word, the other hardly could.

Humans tend to give up more easily, but generally are guessing better: both in terms of accuracy and semantic relatedness. It is not surprising based on our experiences with the evaluators – when in this test, people tend to think in terms of semantic relatedness, i.e. *what is the meaning of the missing word*, and only after that start searching for words of given length, compared to the language models, which first filter based on the length, and then choose the best word based on the probability distribution.

As future work, it would be interesting to see whether we can input the semantic relatedness as a BERT training or fine-tuning objective, because the current target of the training is based on whether the given target word is guessed, and semantic similarity is not considered.

7. References

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov, 2017. Enriching word vectors with subword information. *Transactions of the ACL*, 5:135–146.

Chandrasekaran, Dhivya and Vijay Mago, 2021. Evolution of semantic similarity—a survey. *ACM Comput. Surv.*, 54(2).

Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa, 2011. Natural language processing (almost) from scratch. *Journal of ML research*, 12(ARTICLE):2493–2537.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of North American Chapter of the ACL: Human Language Technologies, Volume 1*. USA.

Elkins, Katherine and Jon Chun, 2020. Can GPT-3 Pass a Writer's Turing Test? *J. of Cult. Analytics*, 1(1):17212.

Goldberg, Yoav, 2019. Assessing BERT's Syntactic Abilities. *ArXiv*, abs/1901.05287.

Halácsy, Péter, András Kornai, László Németh, András Rung, István Szakadát, and Viktor Trón, 2004. Creating open language resources for Hungarian. In *Proceedings of the 4th LREC*. Lisbon, Portugal: ELRA.

Heafield, Kenneth, 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the 6th Workshop on Statistical Machine Translation*. ACL.

Indig, Balázs, Noémi Vadász, and Ágnes Kalivoda, 2016. Decreasing entropy: How wide to open the window? In Carlos Martín-Vide et al. (ed.), *Theory and Practice of Natural Computing*. Cham: Springer.

Indig, Balázs and Dániel Lévai, 2022. Okosabb vagy, mint egy xxxxxxxx? –. egy nyelvi játéktól a nyelvmodellek összehasonlításáig. In *XVIII. MSZNY konferencia*.

Köhn, Arne and Wolfgang Menzel, 2014. Incremental predictive parsing with TurboParser. In *Proceedings of the 52nd Annual Meeting of the ACL*. ACL.

Laubrock, Jochen and Reinhold Kliegl, 2015. The eye-voice span during reading aloud. *Frontiers in Psychology*, 6.

Levy, Omer and Yoav Goldberg, 2014. Neural word embedding as implicit matrix factorization. In Z. Ghahramani et al. (ed.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Mikolov, Tomas, Kai Chen, G.s Corrado, and Jeffrey Dean, 2013a. Efficient estimation of word representations in vector space. *Proc. of Workshop at ICLR*, 2013.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean, 2013b. Distributed representations of words and phrases and their compositionality. In *Proc. of the 26th Int. Conf. on Neural Information Processing Systems*, NIPS'13. USA: Curran Associates Inc.

Nemeskey, Dávid Márk, 2021a. Introducing huBERT. In *XVII. Magyar Számítógépes Nyelvészeti Konferencia*.

Nemeskey, Dávid Márk, 2021b. *Natural language processing methods for language modeling*. Ph.D. thesis, Doctoral School of informatics, Eötvös Loránd University, Faculty of Informatics.

Novák, Attila, Borbála Siklósi, and Csaba Oravecz, 2016. A new integrated open-source morphological analyzer for Hungarian. In *Proc. of the 10th LREC*. ELRA.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning, 2014. GloVe: Global vectors for word representation. In *Empirical Methods in NLP*.

Prószéky, Gábor, Balázs Indig, Márton Miháltz, and Bálint Sass, 2014. Egy pszicholingvisztikai indíttatású számítógépes nyelvfeldolgozási modell felé. In *X. Magyar Számítógépes Nyelvészeti Konferencia*.

Rehurek, Radim and Petr Sojka, 2011. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University*, 3(2).

Schuster, Mike and Kaisuke Nakajima, 2012. Japanese and Korean voice search. In *2012 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*.

Taylor, Wilson L., 1953. "cloze procedure": A new tool for measuring readability. *Journalism Quarterly*, 30(4):415–433.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, 2017. Attention is All you Need. In I. Guyon et al. (ed.), *Adv. in Neural Inf. Processing Systems*, volume 30. Curran Associates, Inc.

[2] `https://github.com/ELTE-DH/word-guessing-game` and `https://github.com/ELTE-DH/BERTfluff`

Bags and Mosaics: Semi-automatic Identification of Auxiliary Verbal Constructions for Agglutinative Languages

Balázs Indig[1,2], Tímea Borbála Bajzát[3]

[1]Eötvös Loránd University, Department of Digital Humanities
[2]National Laboratory for Digital Heritage
[3]Eötvös Loránd University, Doctoral School of Linguistics
indig.balazs@btk.elte.hu, timimimi@student.elte.hu

Abstract

This paper discusses a computational linguistic analysis of Hungarian modal and infinitive constructs. The aim is to identify the constructional realisations of the mentioned grammatical structure. Our research is motivated by functional construction grammar and holistic cognitive grammar. The rationale for the present study is that (i) a comprehensive description of the Hungarian auxiliary + infinitive has not yet been made using corpus-driven methods, and (ii) it is assumed that the method used in our research can be adapted to other agglutinative languages. The detection of pattern similarities extracted by a semi-automatic process is not only relevant to linguistics, but can also be used in the applications of NLP, because the similar formal structures found may indicate functionally related expressions. In order to exploit the different abstraction patterns of the components of the constructs, we used a method based on mosaic n-grams. This semi-automatic procedure shortens the time for finding candidates and reduces the bias of manual analysis. We compared the results with a similar approach based on a bag of words model to identify patterns where the order of the words can be altered to change the emphasis.

Keywords: n-grams, bag of words, auxiliaries, corpus-based, constructions, Sketch Engine

1. Introduction

In everyday conversations, as in systematic learning of foreign languages, different linguistic patterns are noticeably represented. Various types of descriptive grammar attempt to formalise these patterns from different assumptions.

Our research is motivated by *functional construction grammar* and *cognitive grammar* (Diessel, 2015; Goldberg, 1995). In this linguistic approach, it is widely assumed that production interacts dynamically with the user's linguistic knowledge (Bybee, 2006; Langacker, 1987), and thus usage influences the representation of mentally stored schema (Bybee, 2010, 9). Constructs are form-meaning pairs on the lexicon-syntax continuum, with limited prediction of meaning from segmented processing of their components (Goldberg, 1995, 8).

A comprehensive semantic description of the Hungarian auxiliary + infinitive has not yet been made from a constructional approach using corpus-driven methods. Descriptions of Hungarian auxiliaries, focusing on semantics and word order, are not without precedent, but with a few exceptions (Timári, 2019) they do not exploit the corpus as a linguistic resource (Kálmán et al., 1989; Tolcsvai, 2009). We assume (i) that the category of auxiliaries is not closed (in accordance with (Imrényi, 2013; Timári, 2019; Tolcsvai, 2009), but rather can be handled as a construction type and (ii) the slight differences between the patterns of the auxiliary types are likely to be visible in the results of usage-based constructional investigations. The verbal phrases selected for the study consist of quasi-synonymous expressions and elements related to different semantic spaces as well. The aim of the selection was to include expressions that occur with the infinitive and that activate some mental/emotional domain. These are the conceptual domains that can potentially create with relation of possibility by crossing from the premodal domain to the modal (Van Linden, 2010).

2. Data sources

In order to set our hypothesis on the first corpus and test it on the second one, we used two independent corpora: the same steps were performed on both corpora and the results were compared. The possible false results due to the specificities of each corpus were eliminated by this method.

We chose the *Hungarian Gigaword Corpus 2.0.5* (Oravecz et al., 2014) as the first corpus, as it is the de facto standard corpus for most linguistic experiments. It consists of 1.04 billion words. Its texts originate from six stylistic layers, and are divided into five regional language varieties. The other corpus we used is *WebCorpus 2.0* (Nemeskey, 2021), which contains 9 billion words and are derived from *Common Crawl*[1].

The two corpora are expected to be disjoint in content as far as language usage allows. Both are available through the *NoSketch Engine* corpus query framework (Kilgarriff et al., 2014), but with different POS-tagging. We took the largest possible sample size, taking into account the capacity of the corpus and the limits of the system. For selecting the example sentences, we used the available annotations, but the selected complete sentences were re-analysed to obtain a uniform and state-of-the-art annotation. The *e-magyar framework (emtsv)* (Indig et al., 2019) containing the morphological codes of the *emMorph* morphological analyser (Novák et al., 2016) was chosen because it is

[1]https://commoncrawl.org/the-data/

accurate and matches language-specific morphological features well, while only one tag needs to be managed, compared to the UD's 'main POS tag' and 'features' partitioning scheme (Nivre et al., 2017). The characteristics of the samples is shown in Tables 1. and 2., however, we can only describe a few examples in details as an illustration.

Type of Schemes	Original	Filtered	%
Akar ['want to']	610 836	419 324	68.65
Bír ['can']	22 191	15 387	69.34
Imád ['love']	5 081	2 323	45.72
Kíván ['wish to']	192 678	139 498	72.40
Mer ['dare to']	63 729	39 177	61.47
Óhajt ['wish to']	5 500	4 232	76.95
Szeret(ne) ['(would) like (to)']	484 448	278 834	57.56
Tud ['can']	675 000	466 863	69.16
Utál ['hate']	1 448	947	65.40

Table 1: The sample size of MNSz2: 2 302 963 clauses, reduced to 1 508 468 (avg. 65.50%) after filtering.

Type of Schemes	Original	Filtered	%
Akar ['want to']	650 000	518 123	79.71
Bír ['can']	179 846	112 112	62.34
Imád ['love']	79 430	32 997	41.54
Kíván ['wish to']	650 000	391 413	60.22
Mer ['dare to']	473 966	278 887	58.84
Óhajt ['wish to']	21 225	15 161	71.43
Szeret(ne) ['(would) like (to)']	650 000	448 324	68.97
Tud ['can']	650 000	540 175	83.10
Utál ['hate']	16 652	9 464	56.83

Table 2: Sample size of WebCorpus 2.0: 4 579 915 clauses, reduced to 3 143 574 (avg. 68.64%) after filtering.

3. Method

As currently there is no real way to look inside the state-of-the-art deep learning-based language models to find out if they are utilising patterns similar to intuitive linguistic constructions or they were actually seen literal fragments of text in the training data which are utilised well as they work like a black box, and most of them are trained on corpora not available publicly for further examination. Therefore the only real way for linguists to gather evidence for the existence of mental constructions is finding examples in a large representative corpora.

One great tool for querying corpora is *(No)Sketch Engine* (Kilgarriff et al., 2014). It defines the *Corpus Query Language (CQL)* which can be seen as an extension of the common regular expression syntax to formalise complex queries in search of lexical patterns. The simplest case is when one defines a list of tokens, – which can also be considered an n-gram, – but each word can be specified on a different level with the help of the predefined features, which traditionally include the word form, lemma and POS tags. The matching examples can be examined manually or grouped by the frequency of their elements, etc. to draw linguistic conclusions. This method can be very precise depending on the cleanness of the corpus, however, can lead the linguist in a biased direction that strengthens their own intuition based on the content of the initial query. It is very hard to cross-check without a doubt if there is a more appropriate or more general query to formalise the sought linguistic phenomenon. Still, this method is very popular and accepted among linguists because of its simplicity. The idea of *mosaic n-grams* (Indig, 2017) comes from this recognition.

3.1. Mosaic n-grams and bag of mosaics

One can easily take any n-gram from the specified (sub)corpus and generate all possible patterns from it by substituting words with their features. For example, if we choose a trigram which has words with lemmas and POS tags, we can create 27 distinct mosaic n-grams (no. of features per word raised to the power of the no. of words). The resulting mosaic n-grams could be counted for their frequencies and the most frequent ones should contain the important patterns found in the corpus.

As constructions can span up to five or more tokens exhaustively generating and counting all possible forms this is both a very difficult task and in most cases unnecessary. We can make a lot of a priori assumptions that reduces the number of candidates significantly while keeping the generality of the model. For example, examining the vocabulary of the corpus and omitting features that do not have a predefined minimal frequency. Similarly, those features that have too broad or too narrow meaning can also omitted: ideally, every word has a lemma and a POS-tag, but e.g. the lemma of function words are most likely equal with their form as they have only one form at all. On the other hand POS-tags for function words are probably too broad in the context of constructions as they might have different distributions in the corpora. These elements have high overall frequency and a significant reduction (about 33% of feature value types) is achieved by removing them[2].

The candidate patterns (mosaic n-grams) which were defined this way can be statistically examined without the bias of the researchers' intuition, as it reveals all relevant constructions which is contained in the corpus and ranks them automatically. To reduce the number of candidates further automatically, one can check each pattern against their source examples, as all mosaic n-grams are generated from a specific set of example clauses and those sets may yield entries on different specificity level, which do not cover other examples. The algorithm eliminates unnecessary abstractions and rare elements in the following way:

1. All entries below a specific frequency are discarded
2. For the most frequent mosaic n-gram we check all other remaining mosaic n-gram entries if they are generated from the same subset of examples

[2]These assumptions are widely used, in machine learning to reduce the size of vocabulary to fit in the memory of the GPU.

3. If a matching entry has equal frequency, we keep the most concrete one only (unnecessary abstraction)
4. Matching mosaic n-grams with lower frequency are classified as inferior, because they are less abstract aliases (i.e. matches less sentence) than the main one
5. Steps (2), (3) and (4) can be iterated while there are independent mosaic n-grams available

This structure allows a better insight compared to the flat list of mosaic n-grams. In order to manually check if the selected mosaic n-gram covers relevant examples or is just an artefact, one can easily generate a CQL expression from it for use in Sketch Engine. While this method is not perfect to yield only good constructions, it can safely reduce the search space to a size where manual postprocessing becomes feasible, without the fear that one's investigation will not be sufficiently objective or thorough. Constructions with variable word orders can be easily represented as one entry by substituting n-grams to *bag of words (BoW)* in the matching stage of step 2. Using mosaic n-grams also allow us to find typical word orders of a bag of words construction if required.

Our method can be adapted to languages for which a POS-tagging tool is available (e.g. UD). The algorithms we offer can be implemented language-independently on morphologically and lexically annotated texts. Due to the rich morphology and the relatively flexible word order-tendencies, it is assumed that the methods will be more noticeably exploited in the study of agglutinative languages.

3.2. Our process

As we search for constructions in the at most few-word context of infinitival structures within a single clause, we had to define clauses first to reduce the example sentences to such clauses or n-grams, whichever is narrower. Edited texts clauses were separated by punctuation, but if a clause was too long we took a three token long window outwards from the modal and infinite verbs. This helped us to overcome the missing punctuation in unedited texts. We took only one instance of each clause regardless of its frequency in the corpus. This eliminated the bias coming from counting raw frequencies.

The next step was the targeted reduction of the number of features to fit our experiments. We performed the following modifications on the example clauses:

- Deleting false examples with specific POS-tags
- Keeping only finite verbs' lemmas
- Deleting specific POS-tags and keeping only their tokens
- Deleting specific tokens and keeping only their POS-tags
- Using lowercase first word (normalise sentence start)
- Substituting specific POS-tags to simplify them

This procedure reduced the number of possible mosaic n-grams to such an extent that they can be generated without major infrastructure[3]. We generated the mosaic n-grams from the clauses with the procedure described in Section 3.1. and performed manual evaluation.

4. Evaluation

Due to space constraints, we can not provide full insight to the results of the manual evaluation, but we would like to highlight some of the fruitful aspects of the role that the mosaic n-gram-based method and the bag of words model can play in linguistic pattern identification. The length of the clauses ranges from 2 to 9 and their frequency usually peaks at 5. We have chosen 4 long clauses because they exhibit the most diverse information in a compact form.

Tables 3. and 4. show the top ten most frequent overall patterns (first column) extracted from WebCorpus 2.0, mosaic 4-gram and bag of words respectively. The other columns show the rank of the pattern for the individual verbal structures in their own frequency list. Patterns occurring less than 25 times were filtered out ('$<$ 25').

The word order is primary for the mosaic n-grams, but BoWs only reflect the types of elements contained. The word order in construction patterns can, but does not always have a prominent role in semantic functions elaboration in Hungarian, since the rich morphology allows a freer word order. To capture the similarity of the lexical elements and the differences in their typical word orders as well, it is appropriate to use the two models simultaneously.

Among the general patterns (first columns), the occurrence of NEGATIVE environments is prominent for most of the types studied (except *imád* ('love') and *utál* ('hate')). This tendency can be observed in the first, seventh and ninth rows of Table 3. and in the fourth, eighth and ninth rows of Table 4. The difference is not due to the word order, but specifically to the auxiliary verbal environment itself. The semantic profile of the exceptions is very similar, because all of them express the subject's (or the speaker's) emotional attitudes with greater intensity. The more grammaticalized *szeret(ne)* ('(would) like (to)') tends to converge towards the general schema, but not as strongly as the verbs with ABILITY default meaning. While we can still detect the attitude marking function in it, the patterns of use shift more towards the expression of intentions. The BoW model reveals that negation can occur in the context of verbs denoting stronger emotional intensity, but with a lower frequency of occurrence. This trend also holds for mosaic n-grams and BoWs of lengths 3 and 5.

In the seventh row of Table 3., can be seen a verb-initial pattern that is common for attitudinal constructions, but differs from non- or less emotional types (like *tud* ('can') or *akar* ('want'). In this case, *szeret(ne)* ('(would) like (to)') tends to be more similar to the attitudinal group, but in negating contexts it is closer to the other cluster. The word-order in these constructions has impact on the observed patterns, which is also detectable in the non-attitudinal types (seventh row of Table 4).

The use of mental space-building (Fauconnier, 1994) *ha* ('if') is more common in the context of auxiliary verbs that express INTENTION. This also confirms that the results of the mosaic n-gram method can be interpreted more mean-

[3]It also shows if the corpus contains enough examples for a specific structure to draw statistical conclusions or not.

Top 10 4-gram patterns	**Akar** 'want to'	**Bír** 'can'	**Imád** 'love'	**Kíván** 'wish to'	**Mer** 'dare to'	**Óhajt** 'wish to'	**Szeret(ne)** '(would) like (to)'	**Tud** 'can'	**Utál** 'hate'
[/Cnj] ***nem*** *'not'* ***[/V] [/V][Inf]***	2nd	1st	< 25	3rd	1st	1st	5th	1st	< 25
[/Cnj] [/Prev] ***[/V] [/V][Inf]***	1st	11th	< 25	8th	2nd	< 25	8th	2nd	< 25
[/Cnj] [/N][Acc] ***[/V] [/V][Inf]***	3rd	< 25	< 25	2nd	32nd	4th	1st	20th	< 25
[/Adj][Nom] ***[/N][Acc]*** ***[/V] [/V][Inf]***	5th	< 25	< 25	1st	286th	2nd	3rd	5th	< 25
[/Cnj] [/N][Nom] ***[/V] [/V][Inf]***	7th	< 25	19th	38th	75th	< 25	4th	40th	2nd
[/V] [/V][Inf] ***[/Det|Art.Def]*** ***[/N][Acc]***	191st	< 25	2nd	37th	15th	< 25	2nd	75th	4th
hogy 'that' **nem** 'not' **[/V] [/V][Inf]**	17th	3rd	< 25	24th	4th	< 25	98th	4th	< 25
hogy 'that' **[/Prev]** **[/V] [/V][Inf]**	11th	29th	< 25	< 25	9th	< 25	119th	3rd	< 25
nem *'not'* ***[/V]*** ***[/N][Acc]*** ***[/V][Inf]***	9th	< 25	< 25	15th	10th	3rd	23rd	9th	< 25
ha *'if'* ***[/N][Acc]*** ***[/V] [/V][Inf]***	34th	< 25	< 25	79th	< 25	11th	6th	365th	< 25

Table 3: The top 10 4-gram patterns in total and their frequency position by auxiliary verb type (WebCorpus 2.0).

ingfully compared to the BoW model. Specifically, we can observe in Table 4 that INTENTION-type auxiliaries are frequently used in conditional environments, which is not apparent from the mosaic n-gram results alone. This pattern can also be found for the most frequent mosaic 3-grams and BoWs, but at length 5 the conditional frame is no longer prominent in the 10 most frequent patterns.

The four length mosaic n-gram provides insights into the behavioural tendencies of auxiliary verbal phrase embedding in compound sentences. In Table 3. one can see seven rows in which the conjunction is elaborated at different levels of abstraction. The subordinating conjunctive structure, *hogy* ('that') was more prominent in the NEGATIVE contexts for *tud* ('can'), *mer* ('dare'), and *bír* ('can'). This tendency slightly changes in declarative subordinate clauses. The prominence of *mer* ('dare') is reduced, while the frequency of *akar* ('want') instances is increased, however the occurrence of other INTENTION-type verbs remain low. The prominence of the *hogy* ('that') subordination in the case of mosaic 4-grams may be due to the fact that the word order of this subordination type is relatively fixed, the *hogy* always appears in the first position of the clause. This assumption may be confirmed by the results of the BoW method, can be seen in Table 4. the dominance of *hogy*-type subordination is disappearing, although the frequency of conjunctions is similarly significant in the patterns.

5. Conclusion

With the demonstrated methods (mosaic n-grams and bag of words) massive amount of examples can be examined semi-automatically to discover linguistic patterns corresponding to construction grammar which can save a lot of time compared to manual testing. Moreover, the simultaneous use of different levels of abstraction offers the possibility to reveal patterns that would potentially remain hidden in a homogeneous data representation. It is important that the quality of the corpus chosen can affect the results of these methods. It might be worth to consider integrating collostructional measurements in the future to perform a more sophisticated linguistic analysis.

The methodological details dominate the description to overcome linguistic limitations, and the evaluation on Hungarian infinitive constructions showcases a real-life scenario. Many observations are made using the two methods side by side. For further results and the code under copyleft license see Github repository: `https://github.com/bajzattimi/Research-of-infinitive-structures-related-to-the-modal-semantic-domain`.

Top 10 BoW for 4-length	**Akar** 'want to'	**Bír** 'can'	**Imád** 'love'	**Kíván** 'wish to'	**Mer** 'dare to'	**Óhajt** 'wish to'	**Szeret(ne)** '(would) like (to)'	**Tud** 'can'	**Utál** 'hate'
***[/Cnj] nem** 'not' [/V] [/V][Inf]*	1st	1st	< 25	3rd	1st	1st	6th	1st	< 25
[/Cnj] [/N][Acc] [/V] [/V][Inf]	3rd	60th	6th	2nd	5th	3rd	1st	4th	1st
[/Cnj] [/Prev] [/V] [/V][Inf]	2nd	9th	68th	7th	2nd	17th	13th	2nd	< 25
***nem** 'not' [/V] [/N][Acc] [/V][Inf]*	5th	7th	< 25	4th	6th	2nd	14th	3rd	34th
[/Adj][Nom] [/N][Acc] [/V] [/V][Inf]	10th	68th	35th	1st	29th	4th	3rd	5th	< 25
[/Cnj] [/N][Nom] [/V] [/V][Inf]	6th	81th	5th	15th	13th	6th	4th	10th	2nd
[/V] [/V][Inf] [/Det|Art.Def] [/N][Acc]	11th	< 25	4th	5th	16th	< 25	2nd	13th	4th
nem 'not' **[/N][Nom] [/V] [/V][Inf]**	7th	6th	< 25	16th	7th	< 25	20th	8th	33rd
de 'but' **nem** 'not' **[/V] [/V][Inf]**	13th	5th	< 25	32nd	3rd	10th	35th	6th	< 25
***ha** 'if' [/N][Acc] [/V] [/V][Inf]*	9th	< 25	< 25	10th	30th	11th	5th	42th	< 25

Table 4: The top 10 bags of words in total and their frequency position by auxiliary verb type (WebCorpus 2.0).

References

Bybee, J., 2006. From usage to grammar: The mind's response to repetition. *Language*, 82(4):711–733.

Bybee, J., 2010. *Language, usage and cognition*. Cambridge: Cambridge University Press.

Diessel, H., 2015. Usage-based construction grammar. In Dabrowska, E. and Divjak, D. (eds.), *Handbook of Cognitive Linguistics*. Mouton de Gruyter, pages 295–321.

Fauconnier, G., 1994. *Auxiliaries: Cognitive Forces and Grammaticalization*. Cambridge University Press.

Goldberg, A. E., 1995. *Constructions: A construction grammar approach to argument*. Chicago: University of Chicago Press.

Imrényi, A., 2013. A beférkőző segédigés szerkezetek függőségi nyelvtani elemzéséhez. *Magyar Nyelv*, 109(3):291–308.

Indig, B., 2017. Mosaic n-grams: Avoiding combinatorial explosion in corpus pattern mining for agglutinative languages. In Vetulani, Z., Paroubek, P., and Kubis, M. (eds.), *Human Language Technologies as a Challenge for Computer Science and Linguistics*. Poznań: Adam Mickiewicz University.

Indig, B., Sass, B., Simon, E., Mittelholcz, I., Vadász, N., and Makrai, M., 2019. One format to rule them all – the `emtsv` pipeline for Hungarian. In *Proceedings of the 13th Linguistic Annotation Workshop*. Florence: Association for Computational Linguistics.

Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlỳ, P.,, and Suchomel, V., 2014. The Sketch Engine: ten years on. *Lexicography*, 1(1):7–36.

Kálmán, C. Gy., Kálmán, L., Nádasdy, Á., and Prószéky, G., 1989. A magyar segédigék rendszere. In Telegdi, Zs. and Kiefer, F. (eds.), *Általános Nyelvészeti Tanulmányok XVII. Tanulmányok a magyar mondattan köréből*. Budapest: Akadémiai Kiadó, pages 49–103.

Langacker, R., 1987. *Foundations of cognitive grammar: Theoretical prerequisites*, volume 1. Stanford: Stanford university press.

Nemeskey, M. R., 2021. *Natural language processing methods for language modeling*. Ph.D. thesis, Doctoral School of informatics, Eötvös Loránd University, Faculty of Faculty of Informatics.

Nivre, J., Zeman, D., Ginter, F., and Tyers, F., 2017. Universal Dependencies. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*. Valencia: Association for Computational Linguistics.

Novák, A., Siklósi, B., and Oravecz, Cs., 2016. A new integrated open-source morphological analyzer for Hungarian. In Calzolari, N. et al. (ed.), *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Paris: European Language Resources Association (ELRA).

Oravecz, Cs., Váradi, T., and Sass, B., 2014. The Hungarian Gigaword corpus. In Calzolari, N. et al. (ed.), *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik:

European Language Resources Association.
Timári, M., 2019. A magyar segédigék korpuszalapú vizsgálata. *Ösvények*, 6(1):62–100.
Tolcsvai, Nagy G., 2009. A magyar segédige + igenév szerkezet szemantikája. *Magyar Nyelvőr*, 133(4):373–393.
Van Linden, A., 2010. From premodal to modal meaning: Adjectival pathways in english. *Cognitive Linguistics*, 21:537–571.

Stylometry: A Need for Standards

Patrick Juola[1]

[1]Evaluating Variations in Language Laboratory, Duquesne University, Pittsburgh PA USA
juola@duq.edu

Abstract

A defamatory email is received from an anonymous mailer—who wrote it? When confronted, the presumptive author denies having anything to do with it. By studying the style in which it was written, it may be possible to determine (with high accuracy) whether it really was written by that person. But what if we got it wrong? A wrong decision, especially a legal decision, could be a horribly unjust experience. This paper discusses the role of formal consensus standards in forensic science and how they can be applied to improve the practice of stylometric analysis within and also outside the legal system.

Keywords: stylometry, forensics, text analysis, standards

1. Introduction

Stylometry is an important developing field in NLP research. By looking at the writing style of a document, one can identify the author with reasonable accuracy. Given how common documents and documentary evidence are in the legal system, it is unsurprising that courts are increasingly accepting of evidence obtained through stylometry—Ainsworth and Juola (2019) list more than a dozen example cases that hinge on questions of authorship.

However, the consequences of a judicial decision can be significant, and a wrong decision can be a disaster. This paper discusses issues of reliability in the context of forensic science generally, with particular attention paid to issues in NLP and stylometry, and discusses how consensus standards created to address other issues of reliability in forensic science can be used to improve reliability in our fields.

2. Background

2.1. Stylometry

Stylometry, also called stylistics or simply authorship attribution, is the analysis of the language and writing style of a document to determine the author (or, failing that, attributes of the author such as nationality, age, gender, etc.) In some sense, it is the reverse of common NLP tasks such as sentiment analysis: instead of identifying common meanings irrespective of individual expressive quirks, it focuses on stylistic quirks independent of context or meaning. Like sentiment analysis, it is often structured as a machine learning classification task where the analysis identifies patterns in a training set, to be searched for in as set of questioned documents, and the results of the search yield a classification of these documents.

Stylometric theory hinges, ultimately, upon a theory of language choices. "[F]orensic authorship attribution begins with the linguistics-based premise that language users have individual preferences and habits that determine their use of language." (Ainsworth and Juola, 2019) As Coulthard (2013) puts it, "all speaker/writers of a given language have their own personal form of that language, technically labeled an idiolect. A speaker/writer's idiolect will manifest itself in distinctive and cumulatively unique rule-governed choices for encoding meaning linguistically in the written and spoken communications they produce." For example, Mosteller and Wallace (1963) were able to show clear and persistent differences in the frequency of common English words between the writers of *The Federalist Papers*, even among words that were part of their shared vocabulary.

In the past thirty years, there has been a great increase in the amount of research in this area and a corresponding improvement in the potential applicability and accuracy of stylistic analysis. In this paper, we focus specifically on stylometry as a form of legal evidence. For example, the words used in a ransom note could reveal the identity of the kidnapper (Shuy, 2001), or the writing style of a piece of email could show fraudulent and forged evidence (McMenamin, 2010). Stylometry has even helped to solve murders (Chaski, 2005; Grant, 2013). But at the same time, we must ask ourselves whether the courts should be relying upon stylometric evidence. Or, perhaps more accurately, we should be asking ourselves—because courts will continue to rely on stylometric evidence regardless of our wishes—how best to ensure that stylometric evidence can be relied upon.

2.2. Forensic Science and Reliability

The (US) National Academies and National Research Council (2009) recognized in an influential report that "significant improvements are needed in forensic science" (generally), that too many unreliable reports are admitted, and that "faulty forensic science analyses may have contributed to wrongful convictions of innocent people." Forensic odontology/bitemark analysis, in particular, was singled out as having "no evidence of an existing scientific basis." The President's Council of Advisors on Science and Technology (Lander, 2016) agreed that there were issues with the validity and reliability of many forms of forensic science— "It has become apparent, over the past decade, that faulty forensic [analysis] has led to numerous miscarriages of justice." (For example, a study of hair comparisons in criminal cased "revealed that 11 percent of hair samples found to match microscopically actually came from different individuals" when DNA was compared.) PCAST drew a key distinction between "foundational validity," showing that "a method can, *in*

principle, be reliable," and "validity as applied," showing that the method has been reliably applied in *a specific case*.

Numerous studies have shown that stylometry is foundationally valid, with a well-developed theory as discussed above and numerous formalizations and software packages that will reliably perform with usefully low error rates. However, not all analyses submitted to courts are reliable (Rudman, 1997; see also the following section). Human error, bias, and ignorance may result in problematic analyses, faulty reports, and wrong conclusions. Among PCAST's recommendations are the development and establishment of standards of practice, best practices, and protocols to minimize the chances of such errors.

2.3. The Yukos Report Controversy

The Yukos case[1] (Coulthard, 2022; Grant, 2022) is recent illustrative example. As Grant (2022) puts it, "In 2005, HVY, a consortium of former owners of 60 per cent of the Yukos oil company, took the Russian Federation to the Permanent Court of Arbitration in The Hague. In 2014, a tribunal – comprising three arbitrators along with tribunal assistant Martin Valasek – found that the Russian Federation had illegally seized assets from Yukos, and awarded HVY US$50 billion in compensation." The Russian Federation contested this finding on, among other grounds, the basis that the awards document had been illicitly written by an unauthorized person. As of this writing, the appeals continue—with more than $50 billion at stake.

Many researchers (Coulthard, 2022; Grant, 2022; Juola and Napolitano Jawerbaum, 2022) have analyzed the expert reports supporting this claim of illicit authorship and found them wanting. Among other issues, they have found eight questionable assumptions or methodological flaws in the reports, some of which can only be characterized as "rookie mistakes." For example, the training data have not been shown to be representative of the disputed award document; the documents in question are not pure samples of the relevant authors' writings, being (like most legal documents) full of quotations from the law or other cases, and the samples may not be large enough to provide sufficient statistical power. Indeed, the two reports in question, while agreeing on the main fact of authorship, disagreed with each other on enough other points to be largely contradictory.

Given these flaws, it is the authors' contention that the report should not have been admitted into evidence, and that, even if admitted, should have been largely ignored as unreliable. In other words, even though stylometry may be foundationally valid, it is not, in this case, valid as applied. However, admission decisions are made by judges, not by domain experts. How could the judges be educated effectively in such matters?

For scholarly publications (conference presentations and journal articles), the peer review process provides validation, if partial and not-completely-reliable, of the quality of the information. In the legal system, there are usually no "peers" to review and the quality judgments are made by the judge.

3. Standards and Practices

3.1. Standards in the Legal System

In most legal systems, the admissibility of scientific evidence is covered by rules of evidence. Although these rules are administered by judges, they usually follow a reasonable layman's understanding of the scientific process and scientific validity. In the United States, in particular, scientific evidence is controlled by the *Daubert*[2] rule, which suggests several factors for the court to consider as part of an admissibility decision: "(1) whether the scientific theory behind the evidence can be (and has been) tested; (2) whether the theory behind the evidence has been subjected to peer review; (3) the known or potential rate of error for the methods used to generate the evidence; (4) the existence and maintenance of standards controlling the technique's operation, and (5) whether the technique has achieved general acceptance in the relevant expert community." (Ainsworth and Juola, 2019). Of course, this specific standard does not apply world-wide, but most jurisdictions have similar reliability requirements for admissibility. For example, the UK Crown Protection Service's guidance for expert evidence (Crown Protection Service, 2022) states that the expert's evidence is admissible only if it is ``reliable,'' and specifically cites case law\footnote{Lundy v R [2013] UKPC 28} to enumerate four factors very similar to the *Daubert* factors, including the standards question.

As discussed above, there is a well-established theory of how stylometry works (factors 1 and 2) (Coulthard, 2013) a substantial body of empirical work measuring error rates (factor 3) (Juola, 2006;Juola, 2021), and a relevant expert community (factor 5) including the audience of the present paper.

The fourth factor should be of interest to this community. Rules for expert testimony exist precisely because it is not reasonable to expect legal practitioners to know the details of all other scholarly disciplines. As a simple example, simple classification tasks are often handled by *t*-tests (Student, 1908), using estimates of group means and variances. The *t*-test is understandable, easy to use, robust, and powerful. However, it assumes that the underlying data is normally distributed. With small samples of data (as is typical of forensic problems) and non-normal data (for example, as in potentially co-authored data), the analysis may prove unreliable or incorrect. This is well-understood by any statistician, but not necessarily by everyone who can use the T.TEST() function in a spreadsheet, or by everyone who may read the report that contains such an unreliable statistical practice.

[1] Formally: HVY v. The Russian Federation [2020] – *Hulley Enterprises Limited (Cyprus), Veteran Petroleum Limited (Cyprus), Yukos Universal Limited (Isle of Man) v. The Russian Federation [2020]* The Hague Court of Appeal Case No. ECLI:NL: GHDHA:2020:234, Judgment dated 18 February 2020.

[2] *Daubert vs. Merrell Dow Pharmaceuticals, Inc*. 509 U.S. 579, 593-97 (1993).

Despite numerous critical analyses of individual practices (see, for example, Rudman, 1997), the stylometric community has not embraced the idea of codifying best practices in a citable and teachable form. There is no "handbook of stylometry" or "lab manual for authorship attribution," nor are there any easy ways for lawyers and judges to learn how to distinguish good science from bad. In many other types of forensic science, this role is played by community standards, formal document written by expert working groups, reviewed and commented upon by interested parties, and finally published as formal statements that represent the "consensus" of what is and is not acceptable.

3.2. Standards-Making

Both the NRC and PCAST recommended that forensic science communities should establish voluntary standards and guidelines for the improvement of reliability and validity. As the NRC wrote, "standards and best practices create a professional environment that allows organizations and professions to create quality systems, policies, and procedures and maintain autonomy from vested interest groups. Standards ensure desirable characteristics of services and techniques such as quality, reliability, efficiency, and consistency among practitioners." One example of this process is the 2015 creation by the American Academy of Forensic Sciences of an "Academy Standards Board," with the remit of developing standards and similar documents that represent the consensus opinions of practitioners and stakeholders in a variety of areas. Current "consensus bodies" include Anthropology, Bloodstain Patterns, Crime Scene Investigation, DNA, and many others. While Stylometry does not have a CB of its own, there is a Forensic Document Examination Consensus Body. Normally, FDEs are asked to opine upon physical documents—e.g. "questioned document examination involves the comparison and analysis of documents and printing and writing instruments in order to identify or eliminate persons as the source of the handwriting" (NRC, 2009). However, many of the questions asked of document examiners—did this person write this document?—are also asked of stylometrists, and thus the document examination standards may be useful as models.

While the process of creating a standard can be complex, there are some key factors that practitioners should understand. First, standards created by the AAFS ASB and similar organizations are voluntary—there is no law or professional rule requiring that they be followed. However, if judges and lawyers are aware of a published standard, they can certainly compare the report against the standard, and ask a testifying expert some pointed questions if there appear to be discrepancies. This provides an enforcement and gatekeeping function where it is needed (in court, taking important decisions with real-world consequences) while still allowing freedom of research and scholarship.

Secondly, the process is designed to establish a broad consensus among all stakeholders. The membership of the consensus body ensures that "All appropriate interests that are directly and materially affected by the standards activity of the AAFS Standards Board have the opportunity for fair and equitable participation" (Academy Standards Board, 2021). Members are chosen to reflect a variety of interest categories including (among others) researchers, legal experts, organizations, and users. Proposed standards go through a rigorous, repeated, period of public review and commentary, during which anyone in the world can comment and/or request changes. These comments must be addressed before publication.

Only after all public commentary has been resolved will a draft standard be sent to a standards-publishing agency such as the American National Standards Institute (ANSI) or the International Standards Organization (ISO). This helps ensure that proposed standards represent, as broadly as possible, the scientific consensus and not the narrow interests of a few (possibly biased) individuals.

Finally, standards themselves come with a built-in expiration date and must be reviewed and updated on a regular basis. This should ensure that standards reflect the current state-of-the-art as scholarship and practice improves.

Of course, other countries than the United States exist, but the procedures for establishing standards are broadly similar in that that they are typically a lengthy process involving multiple rounds of consultation by domain experts followed by public commentary and revision.

3.3. Standards Content

What lessons can we draw from the existing FDE standards? For example, one recently published standard[3] (related to the analysis of handwriting, for example, in the case of a possibly-forged signature) requires (in section 6.2.5) that "[t]he examiner shall analyze the submitted item(s) to determine sufficiency relative to the scope." In simpler language, the examiner should make sure that that s/he has enough data (both training and test) available. Insufficiency of data was one of the issues raised about the Yukos reports by Coulthard (2022) and is often a practical issue in many ML projects generally.

We digress briefly to explain some aspects of standards language. The word "shall," in a standard, means that it is mandatory for this step to be done—an analysis that omits this step does not comply with the standard, and a smart lawyer will notice this and ask about it. Furthermore, applying the principle that "if you didn't write it down, you didn't do it," a compliant report *must* include a statement to the effect that "I checked the data and found it sufficient because …" Thus, this standard would not only have provided a potential critical analysis of the Yukos reports, but also a relatively simple way to find this potential flaw.

Similarly, another section (6.4.3) states that "If there are unresolved inconsistencies within [the data] (for example, suggestive of multiple writers), contact the submitter for authentication. If any inconsistencies are not resolved to the examiner's satisfaction, discontinue these procedures for the affected group(s), and report accordingly." Again,

[3] https://www.aafs.org/asb-standard/standard-examination-handwritten-items. Accessed 2 November 2022.

this is a "shall" clause, meaning that a standards-compliant report would have dealt explicitly with this issue.

Furthermore, this standard provides the equivalent of a checklist. A canny attorney can read the report with the standards document in hand and note any inconsistencies with the standards; in the event that a report is offered that does comply, the attorney can move to block the report from being admitted on the grounds that it is inconsistent with the standards and therefore unreliable.

Of course, the standards for stylometric analysis will vary in detail from the standards for the examination of physical document; the FDE standard also demands that that the examiner have magnifying equipment available to look at pertinent fine detail, something that is probably not necessary for examination of digital documents. But this example should make it clear that a simple list of errors to avoid would create a powerful tool for the gatekeeping to exclude unreliable evidence, and by extension, to prevent miscarriages of justice. For example, it is clear that the Yukos report is littered with external quotations (Coulthard, 2022). If a similar standard for stylometric document examination existed, a standards-compliant report would have dealt explicitly with this issue and enabled the lawyers to argue over whether or not the external quotations had been appropriately dealt with. A noncompliant report would have been a red flag for potential exclusion as above. In either case, the Yukos litigation would have gone differently, and presumably in a more informed way.

It is clear that there is need (or at least, a use case) for standards in stylometric analysis.

4. Discussion and Conclusions

As stylometry becomes more and more common, and as it also becomes more and more widely accepted, one can expect an increase in the number of legal cases that use stylometric evidence. While one hopes that scholarship and professionalism mean that the average quality, accuracy, and reliability go up, sheer volume argues that more bad analysis will be offered. This paper argues that by developing and publishing consensus standards through organizations such as the AAFS ASB (or its equivalents, including in other countries) can be one key factor to limit the damage done by bad analyses. By listing elements that an examiner *shall* and *shall not* do (or even "should" and "should not" do), these standards provide a way for non-experts to evaluate whether or not a proposed study is fit to be relied upon.

This gatekeeping role can apply even outside of the legal system. For example, plagiarism in education is often dealt with harshly, and a false accusation of plagiarism can results in serious damage to a person's career and lifestyle. To the extent that plagiarism detection systems are not compliant with the best practices for legal evidence, they are also of questionable use in making academic decisions. In general, any decision we would like to be accurate (which one hopes is most of them) should have positive answers to most if not all of the following questions:

- Is there science behind the decision?
- Have other people validated the science?
- How accurate is the system in practice?
- Are there standards to make sure you are doing it right?
- Is this a normal use of this particular science?

Otherwise, if there's no science behind a decision to fail a student or deny a mortgage, it's a bad (or at least unfair) decision.

It is clear that there is need (or at least, a use case) for standards in stylometric analysis. We are all familiar with bad analytic practices, but the broader public may not recognize them as such. Standards, however, will not write themselves; the research community as a whole will need to participate in writing them. Indeed, society as a whole needs to participate in writing them. But the participation of technical experts is key to creating an important tool with important public consequences.

References

Ainsworth, J., & Juola, P. (2019). Who wrote this: Modern forensic authorship analysis as a model for valid forensic science. *Wash. UL Rev.*, *96*, 1159.

Academy Standards Board. (2021). "Academy Standards Board Procedures for the Development of American National Standards." Available at https://www.aafs.org/sites/default/files/media/documents/ASB%20Procedures_ANSI%202021.pdf

Crown Prosecution Service. (2022). "Expert Evidence." Available at https://www.cps.gov.uk/legal-guidance/expert-evidence

Coulthard, M. (2013). On admissible linguistic evidence. *Journal of Law and Policy*, XXI(2):441--466, 2013.

Coulthard, M. (2022). "Rigour and Transparency in Forensic Linguistics Casework" Keynote at *IAFLL 2022*, Porto, Portugal.

Grant, T. (2013). TXT 4N6: method, consistency, and distinctiveness in the analysis of SMS text messages. *JL & Pol'y*, *21*, 467.

Grant, T. (2022). *The Idea of Progress in Forensic Authorship Analysis.* Cambridge University Press.

Juola, P. (2008). Authorship attribution. *Foundations and Trends® in Information Retrieval*, 1(3), 233-334.

Juola, P. (2021). Verifying authorship for forensic purposes: A computational protocol and its validation. *Forensic Science International*, *325*, 110824.

Juola, P. and Napolitano Jawerbaum, A. (2022). "Coauthored Documents Are More than Mixtures of the Styles of the Component Authors." *IAFLL 2022*, Porto, Portugal.

Lander, E, S., and PCAST Working Group. (2016), *"Forensic science in criminal courts: ensuring scientific validity of feature-comparison methods*."

McMenamin, G. R. (2010). "Declaration of Gerald R. McMenamin in Support of Motion for Expedited Discovery." *Ceglia v. Zuckerberg*, No 1:2020-cv-00569 (W.D.N.Y July 9, 2010).

Mosteller, F. & Wallace, D. L (1963). "Inference in an Authorship Problem: A Comparative Study of Discrimination Methods Applied to the Authorship of the Disputed Federalist Papers," Journal of the American Statistical Association 58, pp. 275

National Research Council. (2009). *Strengthening forensic science in the United States: a path forward*. National Academies Press.

Rudman, J.. 1997. "The State of Authorship Attribution Studies: Some Problems and Solutions." *Computers and the Humanities* 31, no. 4 (1997): 351–65 Available at http://www.jstor.org/stable/30200436.

Shuy, R. W. (2001). "DARE's Role in Linguistic Profiling." DARE Newsleter, 4:3 (Summer), pp. 1-5. Available at https://dare.wisc.edu/wp-content/uploads/sites/1051/2008/03/DARENEWS-43.pdf

Student. (1908). "The Probable Error of a Mean." *Biometrika*, vol. 6, no. 1, 1908, pp. 1–25. *JSTOR*, https://doi.org/10.2307/2331554.

ASSETUKR: a Dataset for Ukrainian Text Simplification

Olha Kanishcheva

University of Jena
kanichshevaolga@gmail.com

Abstract

This paper describes the current state of text simplification for Ukrainian and presents the analysis of the Ukrainian word complexity in the text simplification task. All experiments were carried out on the ASSETUKR dataset, which was obtained by translating the ASSET dataset into Ukrainian and then manually checking the translation. Using ASSETUKR, such features as the number of syllables, word length, and frequency were analyzed. Experiments have shown these properties can be used to define a complex/simple term and simplify the text in Ukrainian.

Keywords: text simplification, lexical simplification, Ukrainian language, complex term

1. Introduction

The task of text simplification is not new, but quite challenging. The main goal of text simplification is to make complex text accessible to a wide audience by increasing its readability. Automatic text simplification (TS) is the process of reducing the linguistic complexity of a text, so to improve its understandability and readability, while still maintaining its original information content and meaning (Saggion, 2017). This task is similar to text summarization, in which key content is selected to remain in the summary and other content is elided. In text simplification, ideally, all relevant content is also preserved (Al-Thanyyan et al., 2021; cf Laban et al., 2021:1).

Text simplification commonly focuses on two tasks, which are lexical simplification and syntactic simplification (Al-Thanyyan and Azmi, 2021). Lexical simplification attempts to identify and replace complex words with simpler synonyms. Syntactic simplification tries to simplify the grammatical complexity by identifying complicated syntactic structures such as coordination, subordination, relative clauses, and passive relative clauses, which may be difficult to read or understand by certain readers. Text simplification for the Ukrainian language is not sufficiently developed, this is due to the limited number of linguistic resources and the corresponding models. More detailed research on this issue will be described in the next section.

This paper has the following structure. Section 2 shows a review of works of lexical and syntactic simplification, existing lexical resources, and corpora that can be used to simplify the Ukrainian text. Section 3 describes word complexity analysis for lexical text simplification in Ukrainian. This section also presents the new dataset for the Ukrainian text simplification – ASSETUKR. It can be used to train the model and evaluate the quality of different models on text simplification. Finally, in Section 4 the conclusions and plans for the further development of models and lexical resources for Ukrainian text simplification were described.

2. Related Work

Lexical simplification (LS) is the technique that aims to reduce text complexity by identifying and substituting complex words with simpler, more understandable, synonyms without simplifying the syntax of the text. Typically, this is a four-step process (Al-Thanyyan et al., 2021; Alarcon et al., 2019): (1) complex word identification to identify the complex terms in a document, (2) substitutions generation to produce a list of substitutions for each one, (3) substitutions selection to refine those substitutions to keep the most appropriate synonyms for the given context, and (4) substitutions ranking to rank the remaining substitution according to their simplicity. Research on LS can be divided into two approaches, rule-based and data-driven. The rule-based approach is the oldest in TS but is still used for languages where large parallel corpora do not exist in order to allow for a data-driven approach.

Syntactic simplification (SS) is the task of simplifying the complex syntactic structures in a text while preserving its information content and original meaning.

The emergence of models using the architecture of transformers has made a breakthrough in the task of simplifying the text. So, the paper (Štajner et al., 2022) shows the use of various models based on transformers to simplify the texts. And in the work (Sheang et al. 2021), the authors paid special attention to the use of the T5 model and did experiments with various control tokens, such as the account of the frequency of words and their length, etc. This work also considers the fact that long words are very often complex.

Regarding the simplification of the Ukrainian language and studies of the complexity of terms, in the works (Cherednichenko et al., 2018, 2021) the authors presented their studies of the complexity of terms in the medical domain.

Thus, the main problem for the development of text simplification models of the Ukrainian language is the lack of a special dataset that can be used both for validation and for tuning the simplification model.

3. Word Complexity Analysis

This section provides a description of Ukrainian word complexity estimation in the text simplification problem.

3.1. Data Description

In this work, the experiments used the ASSET dataset. It is a dataset for evaluating Sentence Simplification systems in English with multiple rewriting transformations (Alva-Manchego et al., 2020). The authors of the dataset extended TurkCorpus (Xu et al., 2016) by using the same original sentences, but crowdsourced manual simplifications that encompass a richer set of rewriting transformations. The corpus is composed of 2000 validation and 359 test original sentences that were each simplified 10 times by different annotators. Hereinafter, we will name the original sentences "original", and those that were created by the annotators "simplified".

The original dataset is presented in English, but we need a Ukrainian dataset for our experiments. Accordingly, we translated the ASSET[1] dataset into Ukrainian and manually checked all sentences. For the translation used the Helsinki-NLP/opus-mt-en-uk model[2]. The quality of translation is rather high. However, there were still inaccuracies in the translation, so the entire dataset was checked manually by two native Ukrainian speakers.

During text translation, questions arose about how to translate proper names from English into Ukrainian. It was decided to translate those terms for which there is an established translation in Ukrainian. Proper names such as company names like *Sunflowers, NRC, CMLN, World Working Entertainment* were left unchanged in English. Such proper names as *Едді Ґерреро, Ранчо-Палос-Вердє, Азорський півострів, Вікіпедія* were translated.

This dataset is downloaded on GitHub[3]. We named it ASSETUKR. This dataset not only allows for the analysis of complex and simple Ukrainian words, but its main focus is to evaluate various methods for text simplification in Ukrainian. Examples of some sentences from the ASSETUKR are presented in Table 1.

Datasets	Sentences
ASSET	The New York City Housing Authority Police Department existed from 1952 to 1995.
ASSETUKR	Поліцейський відділ правоохоронних органів Нью Йорка існував з 1952 по 1995 рік.
ASSET	Saint Martin is an island located in the Caribbean, 300 km east of Puerto Rico.
ASSETUKR	Святий Мартін – це острів у Карибському морі, 300 кілометрів на схід від Пуерто-Рико.

Table 1. Examples of sentence translation from English to Ukrainian.

Fig. 1 and Fig. 2 show information about of ASSETUKR dataset: total number of sentences, number of tokens, and average number of tokens per sentence for validation and test parts of the dataset. The average token length in "Original texts" is 5.74 and 5.63 in "Simplified texts". These values were received on the lists of unique tokens.

For all data, the coefficient of the text syntactic complexity was calculated by the formula (1).

$$K = 1 - \frac{P}{W}, \quad (1)$$

where K is the coefficient of syntactic complexity, P is the number of sentences, W is the number of words in the entire text. The larger the fraction (within [0;1]), the more verbose the sentences of such a text are in general, and therefore, the higher the possibility of a variety of syntactic relations between words in a separate sentence.

	Value
Original texts (valid)	13,12
Simplified texts (valid)	9,04
Original texts (test)	13,93
Simplified texts (test)	9,06

Table 2. The coefficient of syntactic complexity for all subsets of ASSETUKR.

Table 2 shows that the syntactic complexity of the simplified Ukrainian texts is much less than that of the original texts. Thus, the translation process of the ASSET dataset into Ukrainian did not affect the text complexity.

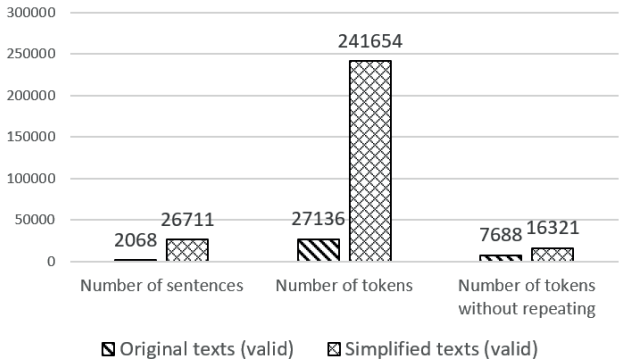


Fig. 1: The statistical information of the ASSETUKR dataset (validation dataset)

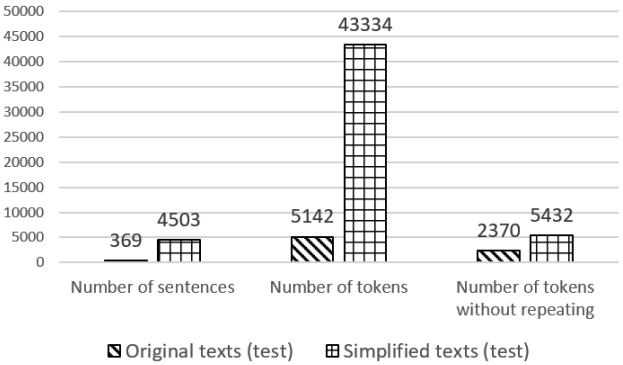


Fig. 2: The statistical information of the ASSETUKR dataset (test dataset)

3.2. Features of Complex Word

The works (Al-Thanyyan et al., 2021; Sheang, 2021) present various features that allow one to identify a word as complex and then search for its synonyms and simplify

[1] https://github.com/facebookresearch/asset
[2] https://huggingface.co/Helsinki-NLP/opus-mt-en-uk
[3] https://github.com/olgakanishcheva/Ukrainian-Text-Simplification/tree/main/ASSETUKR

it. Almost all authors identify the following features: the number of syllables in a word and the frequency of use (usually in some corpus). However, these are mainly works related to English, Spanish, etc. For the general text simplification of the Ukrainian language, such research has not been conducted, and just the presence of the ASSETUKR corpus allows us to analyze complex and simplified data in the Ukrainian language.

Many works on lexical simplification point to the fact that complex terms usually have a low frequency of use. The frequency of use is usually taken from the corpus of the language for which the lexical simplification is performed. For example, in the work (Sheang, 2021) the authors wrote "word length can be an additional factor as long words tend to be hard to read. Moreover, corpus studies of original and simplified texts show that simple texts contain shorter and more frequent words".

For the Ukrainian language, the most popular is the GRAK[4] corpus (General Regionally Annotated Corpus of Ukrainian) (Shvedova, 2020). It's the largest and most comprehensive corpus of Ukrainian as of today. GRAC is a large representative collection of texts in Ukrainian accompanied by a program that enables customization of subcorpora, searching words, grammatical forms, and their combinations as well as post-processing of the query results. The corpus encompasses the timespan between 1816 and 2022 and includes more than 130,000 texts by about 30,000 authors. All frequency values of Ukrainian words for experiments were obtained from this corpus.

3.3. Experiments

All the obtained experiments were carried out on the ASSETUKR corpus, which contains original sentences in Ukrainian and their corresponding simplified sentences. For each original sentence, there are 10 simplified variants of the sentence.

The original data and the simplified data were processed separately. In the beginning, splitting into sentences, removing punctuation, tokenization, and lemmatization (data preprocessing) were realized for each string.

At the next stage, these tokens were analyzed for the number of syllables, since this feature is one of the most significant in determining the word complexity. The results of the experiments are presented in Table 3.

Table 4 shows the same data but as a percentage. From the obtained results, it can be seen that in simplified texts, compared to the original ones, the number of words with two syllables decreases, but the number of words with 4 or more syllables slightly increases. Perhaps this is due to the peculiarities of the Ukrainian language, and once again proves that the number of syllables cannot be the only feature that determines the term complexity.

Certainly, the number of syllables affecting the lexical complexity is also true for the Ukrainian language as for the other languages. For example, words like *'пропонувати', 'будяковий', 'відокремитися', 'ненавмисно', 'перетворюючи', 'посилатися', 'модернізувати', 'заворушення', ' найвпливовіший', 'порівнянний', 'заблокувати'* (eng. translation: *'offer', 'thistle', 'separate', 'inadvertently', 'transforming', 'reference', 'modernize', 'riot', 'the most influential, 'comparable', 'block'*) have more than three syllables and are hard to understand.

	Number of tokens		
	2 syllables	3 syllables	4 and more syllables
Original texts (valid)	3010	2233	2445
Simplified texts (valid)	6033	4552	5736
Original texts (test)	958	687	725
Simplified texts (test)	2040	1567	1823

Table 3. Information about the token's syllables.

	Percent		
	2 syllables	3 syllables	4 and more syllables
Original texts (valid)	39.15%	29.05%	31.80%
Simplified texts (valid)	36.96%	27.89%	35.14%
Original texts (test)	40.42%	28.99%	30.59%
Simplified texts (test)	37.57%	28.86%	33.57%

Table 4. Information about the token's syllables in percent.

There are such words as *'вінок', 'суддя', 'вимір', 'хід', 'рис', 'дрова', 'вхід', 'ключ', 'знак'* (en. translation: *'wreath', 'judge', 'measure', 'course', 'rice', 'firewood', 'entrance', 'key', 'sign'*) they are simple. But there are exceptions, for example, words have many syllables, but are quite common in use, or words are short, but rarely used. For example, complex words 2 syllables *'спіраль', 'флора', 'кратер', 'канцлер', 'зонд' etc. (en. translation 'spiral', 'flora', 'crater', 'chancellor', 'probe').*

For example, simple words with 3 or 4 syllables *'половина', 'рукавичка', 'історія', 'негативно', 'держава', 'співачка' etc. (en. translation 'half', 'glove', 'history', 'negative', 'state', 'singer').*

Therefore, the next step was to check what frequency corresponds to words with a large number and with a small number of syllables. Fig. 3 and Fig. 4 show information about syllable numbers and frequency in the example of one complex file and corresponding simple file.

The list of lemmatized Ukrainian words with their frequency was obtained on the GRAK corpus. The value of 5000 was taken as the threshold frequency value, however, experiments were carried out with other values. However, the proportional distribution between the number of syllables and frequency has almost always been similar.

Figures 3 and 4 show that in the simplified texts, there are more words with two syllables, and the number of words with three or more syllables has decreased.

In (Qiang, Jipeng et al., 2020), Zipf's law is used to select the best simple candidate to replace a compound word in a text. We also analyzed the distribution of words according to Zipf's law in the ASSETUKR dataset and identified that not always a simple word has a coefficient

[4] http://uacorpus.org/Kyiv/ua

higher than a complex one. This assumption works for a large number of words. For example, the word “екстраординарний”/”extraordinary” has a higher frequency in the GRAC corpus than the word “острів”/“island”. And there are many such examples.

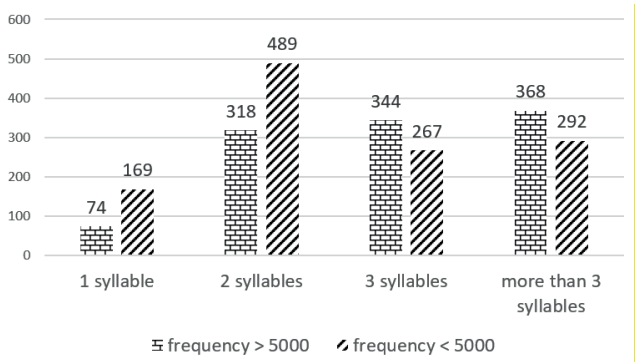


Fig. 3: The frequency and syllable numbers of complex texts

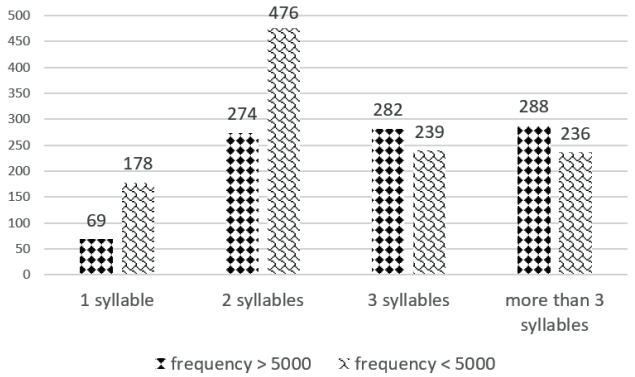


Fig. 4: The frequency and syllable numbers of simple texts

4. Conclusions

This paper describes the ASSETUKR corpus that can be used for the Ukrainian text simplification task. It was obtained by translating the ASSET dataset into Ukrainian, followed by manual verification of the translation. The resulting dataset can be used both for validating models to simplify the Ukrainian text and for fine-tuning models.

As part of this work, an analysis of complex and simplified texts in the ASSETUKR dataset was also carried out. The analysis was aimed at understanding whether it is possible to use such features as the number of syllables, word length, and frequency to identify a complex term and, in general, whether these features affect the complexity of the text or not. The conducted experiments showed that this is practically indeed the case, these features are important for the task of simplifying texts in the Ukrainian language.

Acknowledgments

I thank Maria Shvedova and the reviews of the “Language & Technology Conference” for their expertise and for her help in writing this paper. This research was funded by the Volkswagen Foundation.

I would like to thank Reviewers for taking the time and effort necessary to review the manuscript. I sincerely appreciate all valuable comments and suggestions, which helped us to improve the quality of the manuscript.

References

Alva-Manchego, F., Martin, L., Bordes, A., Scarton, C., Benoît Sagot, and Specia, L. (2020). *ASSET: A Dataset for Tuning and Evaluation of Sentence Simplification Models with Multiple Rewriting Transformations*. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4668–4679.

Cherednichenko, O., Kanishcheva, O. and Babkova, N. (2018). *Complex term identification for Ukrainian medical texts*. Proceedings of the 1st International Workshop on Informatics & Data-Driven Medicine (IDDM 2018), Vol. 2255, 2018, pp. 146–154.

Cherednichenko, O., and Kanishcheva. O. (2021). *Readability evaluation for Ukrainian medicine corpus (UKRMED).* Proceedings of the 5th International Conference on Computational Linguistics and Intelligent Systems. Volume I: Main Conference. Lviv, Ukraine, April 22-23, 2021, pp. 402-412.

Laban, P., Schnabel, T., Bennett, P. N., Hearst, M. A. 2021. *Keep it Simple: Unsupervised Simplification of Multi-Paragraph Text*. ACL-IJCNLP July 2021. https://doi.org/10.48550/arxiv.2107.03444

Alarcon, R., Moreno, L., Segura-Bedmar, I., Martinez, P. (2019). *Lexical simplification approach using easy-to-read resources*. Sociedad Española para el Procesamiento del Lenguaje Natural. DOI:10.26342/2019-63-10.

Saggion, H. (2017). *Automatic Text Simplification*. San Rafael, CA: Morgan &Claypool Publishers.

Sheang, K., C., Saggion, H. (2021). *Controllable Sentence Simplification with a Unified Text-to-Text Transfer Transformer*. In Proceedings of the 14th International Conference on Natural Language Generation, pages 341–352, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Shvedova, M. (2020). *The General Regionally Annotated Corpus of Ukrainian (GRAC, uacorpus.org): Architecture and Functionality.* Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2020). Volume I: Main Conference. Lviv, Ukraine, April 23-24, 2020. P .489-506.

Štajner, S., Sheang, K., C., Saggion, H. (2022). *Sentence Simplification Capabilities of Transfer-Based Models*. Proceedings of the AAAI Conference on Artificial Intelligence 36 (11): 12172-80. https://doi.org/10.1609/aaai.v36i11.21477

Suha S. Al-Thanyyan, Aqil M. Azmi. (2021). *Automated Text Simplification: A Survey*. ACM Comput. Surv. 54, 2 Article 43 (March 2021), 36 pages. https://doi.org/10.1145/3442695.

Qiang, Jipeng, Yun Li, Yi Zhu, Yunhao Yuan and Xindong Wu. (2020). *LSBert: A Simple Framework for Lexical Simplification*. *ArXiv* abs/2006.14939 (2020): n. page.

Unsupervised Syntactic Analysis of the Georgian Language Clause

Oleg Kapanadze[1], Nunu Kapanadze[2], Gideon Kotzé[3], Natia Putkaradze[4]

[1] Iv. Javakhishvili Tbilisi State University, Georgia
okapanadze@uni-potsdam.de
[2]Iv. Javakhishvili Tbilisi State University, Georgia
nunu,kapanadze@tsu.ge
[3]Masaryk University, Brno, Czechia
gideon.kotze@mail.muni.cz
[4]Iv. Javakhishvili Tbilisi State University, Georgia
natia.putkaradze@tsu.ge

Abstract

Until recently, most basic research in Natural Language Technology (NLT) has been performed on "major" languages such as (predominantly) English but also German, Japanese, Chinese, French, and Spanish. At the same time, Low-Density Languages (LDL) are competing to take advantage of modern digital technologies implemented in high-quality computing systems. As a result, the long-term viability of languages not specifically supported by NLT is at risk, which can lead to digital extinction.

We discuss the development of a crucial NLT tool: a Feature-Based Context-Free Grammar (FCFG) or Featured Grammar parser for the Less-Resourced Georgian language. Generative lexicalized parsing models, which are the mainstay for probabilistic parsing, do not perform as well when applied to languages with free word order or rich morphology. Based on the syntactic valency property of the verb and language-specific features such as productive morphology, we designed a prototype FCFG parser for automatic syntactic chunking/shallow parsing of the Georgian clause, which we present here.

Keywords: Less-Resourced Languages, Georgian Language Processing, Feature-Based Grammar, Syntactic Parser

"The Georgian verb is a sentence in miniature".
Arnold Chikobava

1. Introduction

Human languages, supported by key software products offering powerful computational facilities for text processing, provide almost unlimited access to information in those languages. To the extent that language technology becomes involved, giving explicit access to the particular languages in which content is transmitted, it becomes an important key to a language's future. The long-term viability of languages not specifically supported by Language Technologies (LT) is therefore put at risk and they can seriously face digital extinction.

There are a multitude of academic grammars and dictionaries of the Georgian language designed for human users. However, computational resources and applications currently available for Georgian text processing are still limited.

This paper presents an undertaking for developing computational applications involving Georgian in order to fill a gap with computationally well-equipped languages and to lower the current scarcity of technological resources for Georgian text processing.

Here, we will focus on a crucial tool for the Georgian language - a shallow syntactic parser drawing on the Feature-Based Context-Free Grammar (FCFG). Generalized lexical models do not seem to be as easily adaptable to languages with rich morphology and such a different language-specific property as free word order.

Georgian is an agglutinative language that uses both suffixing and prefixing for wordform production. Since morphological analysis is one of the basic concerns for agglutinative languages, a morphoparser is considered an indispensable computational tool for Georgian text analysis.

The initial step for building FCFG for the unsupervised syntactic analysis of Georgian clauses is a morphological parser that is capable to tokenize, POS tag, and lemmatize the Georgian text.

2. A finite-state morphological parser/POS tagger for Georgian

The first version of the morphological transducer for Georgian developed a decade ago drew on XEROX finite-state libraries (Kapanadze, 2010). FST techniques have been very popular and successful in computational morphology and other lower-level applications in natural language engineering. The basic claim of finite-state approach is that a morphological analyzer for a natural language can be implemented as a data structure called a Finite-State Transducer (Beesley and Karttunen, 2003). The finite-state approach is built around two practical concepts: constructing lexicographical descriptions of the language using a tool called *Lexc* and expressing morphophonological variations as regular expression rules. *Lexc* supports a simple right-linear morphosyntactic grammar formalism. Consequently, for every morphological description, we have collections of lexicons (lists of morphemes) that start with a root lexicon. Each morpheme in a lexicon has continuation lexicons, which in turn determine the set of morphemes that can succeed the morpheme (Drobac *et al*., 2014).

Derived from the Georgian conventional grammar, this means that each collection of lemmas for nouns, pronouns and adjectives is considered as a single Root lexicon (N_St**)**. Each entry of the corresponding Root lexicon can be succeeded by a plural morpheme (PL_MK*)* that in turn must be succeeded by a lexicon of seven case markers (C_MK**)** possibly followed by postfixes (PSTF). Some of those morphemes can be followed by a so-called emphatic vocal (Eph_V) and emphatic particles (Eph_ PT) at the end of a wordform.

N_St + *PL_MK* **+ C_MK +** *Eph_V + PSTF + Eph_V + Eph_ PT*

In the above scheme, mandatory lexicons are in **bold** and the optional ranks in the noun patterns are indicated by the fonts in *cursive*.

For the Georgian verb, a similar scheme can be depicted as a chain of interconnected lexicons marked with alphabet letters:

A + B + C + **ROOT** *+ E + F + D + H + G*

A is a lexicon of optional preverbs containing also a null allomorph manifested in the present tense (an uncompleted aspect) group. They can be succeeded by *B* lexicon of verb subject-object marker morphemes, which in turn are succeeded by a *C* lexicon of passive-active mood morphemes. The last two lexicon entries are in complementary distribution with each other. Just the fourth lexicon in the row **ROOT** is a lexicon of verb stems. Every verb stem can be succeeded by five lexicons (*E+F+D+H+G***)** of suffixes. All in all, a single Georgian verb may be built from left to right as a chain of morphemes from nine lexicons including a verb stem.

To prevent combinatorial blow-ups in lexical transducers, special symbols - an extension of the Xerox finite-state implementation - "flag diacritics" are used. They provide feature-settings and feature-unification operations that keep transducers small and simplify grammars. Flag Diacritics are also used for "long distance" constraints on the co-occurrence of morphemes within words. They are normal multi-character symbols that appear in strings. However, they do not match against the symbols of the input string, and they do not appear in the output string. A network saturated with Flag Diacritics typically contain many illegal paths that would normally result in overrecognition. Flag Diacritics allow to block illegal paths at runtime by keeping the transducer small (Beesley and Karttunen, 2003).

During the morphological transduction, the tokens of the plain input text are annotated with necessary morphological features, since morphological analysis is one of the basic challenges for agglutinating languages. It provides useful clues for resolving syntactic ambiguity, and the parsing model should have a way of utilizing these hints. A lexicon-based parse engine has been oriented to highlight aspects of the predicate-argument structure of the Georgian clause. It captures the specifics of the Georgian verb manifesting rich structural clues (the syntactic valency and the predicate-argument structure) as "a sentence in miniature" (Chikobava, 1928).

An output of the morphoparser after reimplementation of the initially developed lexical transducer for Georgian, as a deterministic part-of-speech tagger, is capable to produce a morphologically annotated Georgian corpus achieving almost 100% accuracy after manual disambiguation.

A tokenized, lemmatized and tagged output for a small Georgian clause:

"ქალებმა უთხარით კაცებს სიმართლე" (1)

(Lit. "let [you] women tell [the] men [the] truth")

looks as follows:

```
ქალებმა
  <lemma='ქალ' morph="Pl.Erg" pos="NN"/>
  <lemma='ქალ' morph="Pl.Erg" pos="NN"/>
უთხარით
TV[VAL=EDN,SR=1,voice=ACT,mood=IND,prs=Pl3]
კაცებს
  <lemma='კაც' morph="Pl.Dat" pos="NN"/>
სიმართლე
  <lemma='სიმართლე' morph="Sg*.Nom"
pos="NI"/>
  <lemma='სიმართლე' morph="Sg*.Voc"
pos="NI"/>
.
  <lemma=" " pos="$."/>
```

Tokens are annotated with POS tags such as Normal Noun (NN), Transitive Verb (TV), Infinitival Noun (NI) and end of clause ($.) without lemma.

They are saturated with morphological features of number Pl(ural) and Sg*(singular tantum) for Infinitival Noun; case characteristics - Ergative (Erg), Dative (Dat), Nominative (Nom) for Normal Nouns. Punctuation marks (pos="$,"/ pos="$.") and tokens without inflections are tagged with lemma =″--″.

Transitive Verb is annotated with features for Syntactic Valency (VAL) characteristics such as E(rgative)D(ative)N(ominative), Tense form (SR=1), voice=ACT(ive)**,** mood=IND(icative) and person prs=Pl3 (3rd person plural).

The next step is a manual disambiguation of the tagged text, since the token '**ქალებმა**' can be lemmatized as **lemma='ქალ'** for two source lexical entities – '**ქალი**' (a woman) and '**ქალა**' (a skull). Another token with morphological homonymy is '**სიმართლე**' that is annotated with Nom(inative) and Voc(ative) case features. In this specific case it is assigned the Nominative case feature.

3. A syntactic chunker for unsupervised parsing of the Georgian clause

Shallow syntactic parsing (also known as Chunking) aims at identifying syntactic constituents like a noun or a verb phrase within a clause. Theoretically, it can consist of a VP (Verb Phrase) and ***n*** mandatory NPs. The number of NPs in the clause (resp. graph/tree) is determined by the syntactic valency (VAL) parameter of the Georgian Finite Verb Form. VAL is inferred in the process of morphological analysis of the Georgian text, unlike the Discriminative Parsing procedures used for German (Versley and Rehbein, 2009) in which syntactic information is determined from a special Weighted

Constraint Dependency Grammar Parser lexicon. Using information from the WCDG parser for German (Foth and Menzel, 2006) verbs are marked according to the arguments that they can take.

However, a "static" source in the form of a lexicon would not work for Georgian verbs, since their syntactic valency frame may change according to the series as suggested in Table 1. Therefore, for each finite verb, it is determined "on the fly" in the process of morphological analysis.

Derived from valency data/syntactic frames, one can perform automatic shallow parsing to produce chunks of text as NPs. They can be constructed as the maximal projection (using the longest match rule) of a head word_ j (normally noun or equivalent) of the corresponding NP_j.

Pursuant to syntactic valency and case marker distributions of NPs across the 3 series adopted in the Georgian academic grammar, a table is suggested that specifies syntactic frames for 14 clusters of Transitive (TV) and Intransitive (ITV) verb sets:

Cluster	I serie	II serie	III serie
1 / ITV	VAL=ND	VAL=ED	VAL=D (D+postf)
2 / ITV	VAL=N	VAL=E	VAL=D
3 / TV	VAL=ND	VAL=ED	VAL=DN
4 / ITV	VAL=ND	VAL=ED	VAL=D (D+postf)
5 / TV	VAL=NDD	VAL=EDD	VAL=DN (D+postf)
6 / ITV	VAL=N	VAL=N	VAL=N
7 / ITV	VAL=ND	VAL=ND	VAL=ND
8 / ITV	VAL=N	VAL=N	VAL=N
9 / ITV	VAL=N	VAL=N	VAL=N
10 / ITV	VAL=ND	VAL=ND	VAL=DN
11 / ITV	VAL=N	VAL=N	VAL=N
12 / ITV	VAL=DN	VAL=DN	VAL=DN
13 / ITV	VAL=D	VAL=D	VAL=D
14 / ITV	VAL=0	VAL=0	VAL=0

Table 1. Syntactic frame distribution for Georgian verb sets.

In the table the valency parameter (VAL) points to verb arguments (resp. syntactic valency). Their number can vary depending to the verb transitivity and series. As it can be observed, VAL for cluster 1/ITV and 4/ITV in the I serie are bi-valent, though, in the III serie it is reduced to a monovalent verb as the second argument is inflected with a postfix (D+postf) and, consequently, it may not be identified as a syntactic place holder or syntactic argument in the verbal frame.

The same phenomenon can be traced also for cluster 5/TV with transitive verbs, which from a trivalent option in the I and II series is represented as a bi-valent invariant in the III serie with (D+postf).

The capital letters in VAL parameter stand for the case markers of the head words in consequent NP phrases, which number (*n*) in a clause can vary as [0,1,2,3]:

If *n=1*, the consequent NP phrase as a maximal projection of its head word is assigned a syntactic function of *Subject*.

If *n =2* with transitive verb, clausal constituents NP1 and NP2 (derived from consequent head words' case markers) will be labelled as maximal projections of a *Subject* and a *Direct_Object*.

For *n=2* with intransitive verb, NP1 and NP2 will be distributed as maximal projections of a *Subject* and a *Direct_Object*.

With *n =3*, by default, NP1, NP2 and NP3, depending to their consequent head word's case formants, are longest matches of a *Subject*, a *Direct_Object* and an *Indirect_Object* phrasal constituents.

Clauses with zero ('*0*') syntactic valency denoting the natural phenomena are without any phrasal constituent (Cluster 14 in the table).

We already mentioned that in some VAL parameter samples the parentheses contain the case marker (D+postf) of a head word for NPs that may appear for specific verb clusters (1,4,5) in the III serie presented in the Syntactic frame distribution in Table 1.

The verb Transitivity in the suggested shallow parsing scheme is a redundant feature. It is included in Table 1 just to refer to the grammatical characteristics used in traditional syntactic analysis systems for most natural languages.

Once all the mandatory and optional constituents as NP's maximal projections are identified in a clause/sentence, applying production rules, a syntactic parser can produce a shallow syntactic parse tree of a clause/sentence under analysis.

Drawing on the sketched principles and compiled Grammar with the Lexical production rules for Georgian text, the NLTK libraries come into play for the automatic construction of morphologically and syntactically annotated shallow parse trees. Production rules can be created completely manually or semi-automatically, where a parser assigns some syntactic structure to a text that is then checked by linguists and, if necessary, corrected.

The number of NPs in a clause (resp. graph/tree) is determined by the syntactic valency of a head verb of the clause, identified by the morphoparser and percolated as a feature in VP. The annotated text is usually in a form of syntactic bracket labelling so that the pre-processed sentence will be grouped into a hierarchical form of phrase structure.

An instance of NLTK output of clause (1) as a syntactic tree is introduced in Fig. 1:

< Tree(S[], [Tree(NP[SF='SUB'], [Tree(NN[SF='HEAD', case='Erg'], ['ქალებმა'])]), Tree(VP[SF='HEAD', VAL='ED'], [Tree(TV[SF='HEAD', SR=1, VAL='EDN', mood='IND', prs='Pl3', voice='ACT'], ['უთხარით'])]), Tree(NP[SF='IOB'], [Tree(NN[SF='HEAD', case='Dat'], ['კაცებს'])]), Tree(NP[SF='DOB'], [Tree(NN[SF='HEAD', case='Nom'], ['სიმართლე'])])])>

Fig. 1: An instance of NLTK output of clause (1)

A more readable output with indented nodes is presented in Fig. 2:

```
(S[]
 (NP[SF='SUB'](NN[SF='HEAD',case='Erg']ქალებმა))
 (VP[SF='HEAD', VAL='EDN']
  (TV[SF='HEAD',SR=1,VAL='EDN',mood='IND',
      prs='Pl3', voice='ACT'] უთხარით))
 (NP[SF='IOB'] (NN[SF='HEAD', case='Dat'] კაცებს))
 (NP[SF='DOB'] (NN[SF='HEAD', case='Nom']
```

სიმართლე)))

Fig. 2: An output of the NLTK parser with indented nodes

The same morphologically and syntactically annotated clause with feature structures as a directed acyclic graph (DAG) equivalent to Attribute-Value Matrix (ATM) is depicted in Fig. 3:

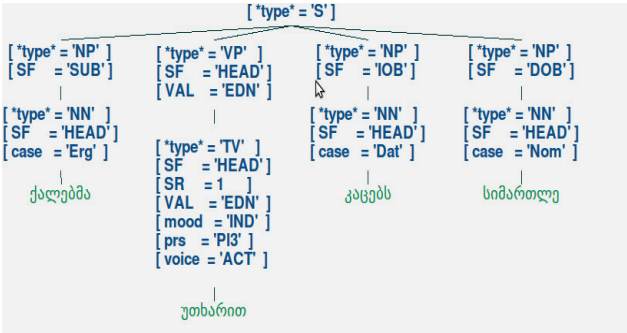


Fig. 3: morphologically and syntactically annotated clause with feature structures as a directed acyclic graph

In the parse process, the NLTK FBG parser generates an output as a graph of adjacent nodes (resp. adjacent vertices). Every graph node is linked with the rest of the graph nodes by edges. The graph mathematically is described by means of an adjacent matrix. Its elements *A(ij)* =1, if *i* and *j* are linked by an edge and *A(ij)* = 0, if the nodes are not connected with each other.

The graph reflects two sorts of rules instantiated in the Feature-Based Grammar: the phrasal-level grammar rules and lexical rules. The last one builds the nodes saturated with a bunch number of different features and morpho-syntactic class labels (resp. POS tags). The 'type' feature on a clausal level denotes the phrasal labels such as S, VP and NP. The same nodes are supplied also with SF (Syntactic Function) indicators as SUB (Subject), DO (Direct Object), and HEAD (Head of clause/phrase).

The VP node contains a grammatical feature TENSE with a value SR=1 (the First Sery of the Tense Set) and VAL='END' that are percolated from the terminal node of 'type'='TV' (Transitive Verb). A value 'END' of VAL-feature (syntactic valency) points to

- a case marker (case='Erg[ative]') of the head word in the left NP with syntactic function (SF='SUB');
- a case marker (case= 'Nom[inative]') of the head word in the last right NP as SF='DOB' (Direct_Object).
- a case marker (case='Dat[ive]') of the head word in the right NP as SF='IOB' (Indirect_Object).

The general scheme of the graph for phrasal categories and their syntactic relations in the clause is constructed in the X-Bar projection tradition (Kornai, 1990). The clause predicate-argument structure (syntactic functions/labels) is based on the grammatical concept of *syntactic valency*, an analogue of "head feature principle" in (Pollard and Sag, 1994).

Frequently, the Georgian clauses are not complete sentences, despite the "predicted" syntactic valency by the finite verb form. Some mandatory phrasal constituents might be completely dropped, although their meaning in he clause is compensated for by the finite verb grammatical form. In Georgian linguistic literature, these kinds of clauses are referred to as "Incomplete Sentences" (SU).

Theoretically, the Georgian clause, as in other languages, may also comprise optional NPs which cannot be "counted" by the verb syntactic valency. However, the Weighted Constraint Dependency Grammar Parser for German (Foth and Menzel, 2006) makes use of information about entries for genitive or clausal complements alongside which also the accusative and dative complements are encoded in the WCDG lexicon.

But this is not the case in the proposed FBG parser for Georgian. In the syntactic analysis process, optional arguments can be determined by a case marker or an adjacent postposition formant of the subsequent NP head word that will be assigned the label of O(*rdinal*)_OB(*jects*). As an example with the same verb from clause (1) but with an alternative syntactic valency distribution, we suggest a clause:

'ქალებმა უთხარით კაცებს ომის შესახებ'. (2)

(Lit. "let [you] women tell [the] men about [the] war")

The output of tokenizing, lemmatizing and POS tagging procedures for (2) looks as follows:

ქალებმა
```
<lemma='ქალ' morph="Pl.Erg" pos="NN"/>
<lemma='ქალ' morph="Pl.Erg" pos="NN"/>
```
უთხარით
```
TV[VAL=EDN,SR=1,voice=ACT,mood=IND,prs=Pl3]
```
კაცებს
```
<lemma='კაც' morph="Pl.Dat" pos="NN"/>
```
ომის
```
<lemma='ომ' morph="Sg.Gen" pos="NN"/>
```
შესახებ
```
<lemma='შესახებ' pos="PPS_GEN"/>
.
<lemma="--" pos="$."/>
```

This time, the syntactic parser output will construct a syntactic tree for an incompleted clause (SU) with bi-valent syntactic frame, though morphological analysis has identified the clause head verb ('**უთხარით**') with a feature **VAL=EDN** for Subject (E), Indirect Object (D) and Direct Object(N).

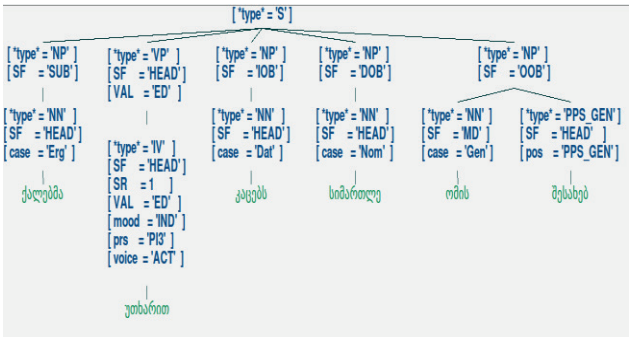


Fig.4: A sample of a clause with an optional NP assigned the Syntactic Function of "Ordinal Object" (SF='OOB')

At the same time the graph in Fig. 4 presents an additional NP which by default could not be anticipated as a syntactic place holder in the clause under analysis. Its syntactic function is an Ordinal_Object (SF='OOB') with two sister edges and consequent two terminal nodes '**ომის შესახებ**' *(about [the] war')*. The second terminal node ('**შესახებ**') as the head word of the NP with SF='OOB' is assigned a Syntactic Function (SF= 'HEAD'), whereas the preceding terminal node as NN '**ომის**' is marked with a Syntactic Function of Modifier (SF= 'MD') annotated morphologically as Normal Noun (NN) in Genitive (case='Gen').

4. Conclusion and Future Plans

In the presented paper, we have featured an option for developing a syntactic chunker/shallow parser for the Georgian language clause.

As the initial step to the syntactic analysis, we reimplemented a Finite-State Morphological Transducer with output in TIGER XML format (Brants and Hansen, 2000) for Georgian text morphological analysis, lemmatization and POS tagging.

As a necessary step in the syntactic valency-driven FBG parser implementation, we have studied the Georgian verb stock from (Melikishvili, 2001). Fourteen verb clusters with different valency distributions, bound with syntactic frames, are identified to date. For each cluster, we intend to compile and train a prototype FBG grammar.

As a syntactic parsing testbed, we have utilized the broadly recognized open-source library NLTK (Natural Language Toolkit) developed using the Python programming language (Bird *et al*., 2009). To build an interface between the TIGER XML scheme and an input format for NLTK, we had to disambiguate manually and reformat the output of the Georgian morphoparser.

The NLTK output consists of powerful database-oriented representations for graph structures in which each leaf (= token) and each node (= linguistic constituent) has a unique identifier. It visualizes a hybrid approach to the syntactic annotation issue as tree-like graphs and integrates annotation according to the constituency representations (NP, VP, etc.) and functional relations – Subject (SUB), Head of Phrase (HD), Indirect Object (IOB), Direct Object (DOB) and MD (Modifier).

For training of the Georgian Feature-Based Grammar using the NLTK parser, we have built manually the set of grammar and lexical production rules. To increase the coverage of syntagmatic patterns used by the future NLTK Georgian parser, we intend to extract the phrase-structure grammar rules from a monolingual Georgian TreeBank (Kapanadze O. and Kapanadze, N., 2017, Kapanadze, *et al*., 2022).

Currently the development of a converter that would automatically reformat the results of morphological analysis in TIGER XML into the format acceptable for NLTK input is in its last phase. It will render the developers an opportunity of linking the proposed two technological tools into a pipeline in which the output of the morphoparser will feed the input of the syntactic chunker engine with necessary data for the automatic building of morphologically and syntactically annotated phrase structure trees of the Georgian clause/sentence.

Acknowledgement

This research was supported by **the Shota Rustaveli National Science Foundation of Georgia.**

Reference

Kapanadze, O. (2010). Describing Georgian Morphology with a Finite-State System. In A. Yli-Jura et al. (Eds.): *Finite-State Methods and Natural Language Processing 2009, Lecture Notes in Artificial Intelligence,* Volume 6062, pp.114-122, Springer-Verlag, Berlin Heidelberg.

Beesley, K. R. and Karttunen L. (2003). Finite State Morphology. CSLI Publications.

Drobac, S., Lindén, K., Pirinen, T.A. and Silfverberg, M. (2014). *Heuristic Hyper-minimization of Finite State Lexicons. LREC*, 2014. La Valetta, Malta

ჩიქობავა, არნ. (1928; 1968). მარტივი წინადადების პრობლემა ქართულში (Chikobava, Arn. (1928; 1968), The problem of simple sentences in Georgian. Tbilisi University Publishing House. Tbilisi, Georgia, 292 p. 1928; 1968).

Kornai, A. (1990). The X-bar theory of phrase structure, Language, 66(1), pp. 24–50, 1990.

Pollard C. and Sag I.A. (1994). Head-Driven Phrase-Structure Grammar. Chicago. University of Chicago Press. 1994.

Brants, S. and Hansen, S. (2000). Developments in the TIGER Annotation Scheme and their Realization in the Corpus. In *Proceedings of the Third Conference on Language Resources and Evaluation (LREC 2002)*, Las Palmas, pp. 1643–1649.

მელიქიშვილი, დ. (2001). „ქართული ზმნის უღლების სისტემა". თბილისი, ლოგოსპრესი. (Melikishvili, D. (2001). The Georgian Verb Conjugation System. Tbilisi. Logospress).

Bird, S., Loper, E. and Klein, E. (2009). Natural Language Processing with Python. O'Reilly Media Inc.

Versley, Y. and Rehbein, I. (2009). Scalable Discriminative Parsing for German. Proceedings of the 11th International Workshop on Parsing Technologies (IWPT-2009), 2009, Paris, France.

Foth, K. and Menzel, W. (2006). Hybrid parsing: Using probabilistic models as predictors for a symbolic parser. In ACL 2006.

Kapanadze, O. and Kapanadze, N. (2017). Entwicklung von parallelen Baumbanken – Bewältigun von sprachlichen Strukturdivergenzen. *Germanistische Studien. Eine Zeitschrift Vereins Deutsche Sprache. Georgien. B. N12. Theorie und Praxis der Text- und Diskursanalyse.* (Site 68-86). Tbilissi-Dortmund. Verlag Universali. 2017.

Kapanadze, O., Kotzé, G. and Hanneforth, Th. (2022). Building Resources for Georgian Treebanking-based NLP. In: Özgün, A., Zinova, Y. (eds). *Language, Logic, and Computation. TbiLLC 2019.* Lecture Notes in Computer Science, vol 13206. Springer, Cham.

Optimizations of Some Well-Known NLP Algorithms

Irakli Kardava

Adam Mickiewicz University in Poznań
irakar@amu.edu.pl

Abstract

This paper discusses the optimizations of well-known NLP algorithms for the Georgian language (however, it can be useful for a computer processing of other languages in the future). As the Georgian language is unique, it has such features which are characteristic only for it. Consequently, the direct application of approaches that work successfully for well-developed languages (English, German, French, Russian, etc.) may not be sufficiently flexible and effective for Georgian. Or, in some cases, it can give us the same result, but the amount of work to be done is greater and the process is more complicated (that still leaves a problem). Therefore, we present an approach that uses the peculiarity of the Georgian language and its grammar as a tool to modify already well-known algorithms, such as: Minimum Editing Distance (MED), Text Classification, Language Modeling (LM). Generally speaking, we can say that our approach will work successfully where there is a need to have a words' corpus.

Keywords: Language Modeling, MED, NLP, Optimization, Text Classification.

1. Introduction

During the previous researches, when we were creating systems of morphological synthesis and analysis of Georgian words, it became known to us that about 5000 grammatically correct word-forms can be produced from the unchanged part of one Georgian word. In addition, the given word can be considered simultaneously as a verb, as a noun and as an adjective. Of course, not all words have such characteristic, but there are quite a lot of words with similar properties in the Georgian language. Therefore, it was necessary to develop such a system, which could produce all grammatically correct word-forms from the unchanged part of the word (lemma), according to the rules given by us in advance. By realizing this idea, it was possible to create a words' base / corpus not by typing all the words from the keyboard by a person, but by our system (in general, not by human, but by a computer). That is, we can currently produce all the correct word-forms of an interesting word with the above system without any additional effort. In addition, it should be noted that if one copy of any type works correctly, then any other representatives of the same type will be produced correctly. This possibility, in turn, saves the time required to create a large corpus and the amount of work to be carried out. Besides it should be emphasized that it is impossible to get a wrong form of a word into the database (because all types of word production modules are tested on at least one copy). Now let's ask the question, if we can produce dynamically all the needed words using the appropriate system, what is the need to create and use a database of words? It is obvious that performing some NLP manipulations based on a given corpus, where for example we have ten million words, will require more resources and time than just producing the necessary words and then using them. If it is not known (there will be many such cases), which type the given word belongs to, we use the morphological analysis system of Georgian words created by us to clarify this issue. It breaks down the word into morphemes, can describe each element and determine which type the word stem belongs to, and then passes it to the production module to give all the word-forms. We also use this approach in one of our current investigations, which deals with the identification and determination of ancient manuscript authors. In this direction, there are many cases when the author of the found manuscript is unknown and it is necessary to conduct a multifaceted analysis to determine it. If it is impossible to determine the author, it is at least possible to determine the approximate geographic origin of the manuscript. To achieve this, we start with the morphological synthesis module of our system. Thanks to it, we can determine the overall structure of the word order in the sentences of the given text. This allows us to identify the unique writing style of an individual author. Then, we compare it with other data we have and obtain approximate answers (this direction is not yet totally perfect).

So, this paper at this stage contains the synthesis of the capabilities discussed above and the Minimum Editing Distance, Text Classification, Language Modeling algorithms (Jurafsky and Martin, 2019), in order to reduce the dependence on the words' database, work time, and the amount of work required to achieve the desired result.

2. Existing NLP algorithms and their modification

2.1 Minimum Edit Distance

To find the most correct candidate $w_{correct}$ with the wrongly given word w_{error}, first of all, it is necessary to calculate the minimal editing distance of the word. This approach requires checking all the members of the word block (a unique list of words). That is, if V=1000000, then the major cycle's iteration is i=V=1000000, too. Finding the minimal editing distance could be done by the following classical algorithm (1) with deletion, insertion and substitution (Jurafsky and Martin, 2019):

$$D[i,j] = \min of \rightarrow \rightarrow \begin{cases} D[i-1,j] + del - cost(source[i]) \\ D[i,j-1] + ins - cost(target[j]) \\ D[i-1,j-1] + sub - cost(source[i], target[j]) \end{cases} \quad (1)$$

Contents

In this case the value of every single activity is: insert=1, delete=1 and substitution=1; also, the number of minimum editing steps = minimum sum of the weights of the editing activities. This is because the weight of each of the three activities equals to one another. There is an approach where the substitution = 2. In this case accordingly, the third branch of the algorithm looks differently. See Formula (2) (Jurafsky and Martin, 2019):

$$D[i,j] = min \begin{cases} D[i-1,j] + del - cost(source[i]) \\ D[i,j-1] + ins - cost(target[j]) \\ D[i-1,j-1] + \begin{cases} 2; if\ source[i] \neq target[j] \\ 0; if\ source[i] = target[j] \end{cases} \end{cases} \quad (2)$$

The principle of its operation is shown in the dynamic programming table and how it is possible to transform one word by the best replacement candidate. See Figure 1.

Now back to the number of iterations needed to reach the desired goal. As we said, for example i = V = 1000000. However, this value in the case of the Georgian language can be indefinitely high. Based on the results of our earlier studies, we can claim that Georgian possesses words by whose unchanging part (so-called lemma) and the correct concatenation of the matching morphological representatives around several thousand grammatically correct forms of the given word may be generated (Kardava et al., 2019; Kardava et al., 2022); e. i, V in this case will be extremely high, maybe tens of millions. See the picture where using a root 'წერ' (Eng.: to write) all the word-forms have automatically been generated by our program (Antidze et al., 2013). See Figure 2.

It presents one side of the issue. On the other hand, for instance, in a decentralized system information is scattered around on different locations, such a high iteration rate for each word will lift the problem to a much higher degree (Kardava and Esiava 2022).

	#	e	x	e	c	u	t	i	o	n
#	0	← 1	← 2	← 3	← 4	← 5	← 6	← 7	← 8	← 9
i	↑ **1**	↖←↑ 2	↖←↑ 3	↖←↑ 4	↖←↑ 5	↖←↑ 6	↖←↑ 7	↖ 6	← 7	← 8
n	↑ 2	↖←↑ **3**	↖←↑ 4	↖←↑ 5	↖←↑ 6	↖←↑ 7	↖←↑ 8	↑ 7	↖←↑ 8	↖ 7
t	↑ 3	↖←↑ 4	↖←↑ **5**	↖←↑ 6	↖←↑ 7	↖←↑ 8	↖ 7	←↑ 8	↖←↑ 9	↑ 8
e	↑ 4	↖ 3	← 4	↖← **5**	← **6**	← 7	←↑ 8	↖←↑ 9	↖←↑ 10	↑ 9
n	↑ 5	↑ 4	↖←↑ 5	↖←↑ 6	↖←↑ 7	↖←↑ **8**	↖←↑ 9	↖←↑ 10	↖←↑ 11	↖↑ 10
t	↑ 6	↑ 5	↖←↑ 6	↖←↑ 7	↖←↑ 8	↖←↑ 9	↖ **8**	← 9	← 10	←↑ 11
i	↑ 7	↑ 6	↖←↑ 7	↖←↑ 8	↖←↑ 9	↖←↑ 10	↑ 9	↖ **8**	← 9	← 10
o	↑ 8	↑ 7	↖←↑ 8	↖←↑ 9	↖←↑ 10	↖←↑ 11	↑ 10	↑ 9	↖ **8**	← 9
n	↑ 9	↑ 8	↖←↑ 9	↖←↑ 10	↖←↑ 11	↖←↑ 12	↑ 11	↑ 10	↑ 9	↖ **8**

Figure 1. Diagram design after Gusfield (1997)
(Jurafsky and Martin, 2019).

Now, let's go back to the needed iteration number to achieve the aim. As we mentioned in the example i=V=1000000; This value can be limitlessly higher for the Georgian language, though.

Our approach enables to generate all the correct forms automatically by giving the unchanging part of the word (Melikishvili, 2001; Melikishvili, 2008).

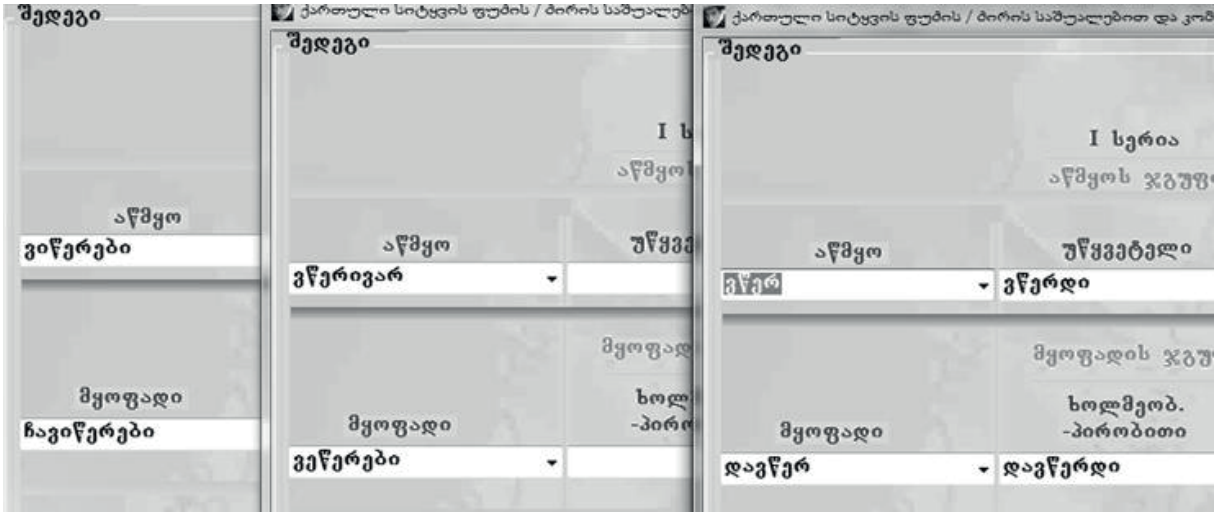


Figure 2. Generated words.

That means, if we say that in the classical case, for example i = V = 1000000, as a result of this optimization, we use only those words that the program itself gave us, i.e. i=V=5000 (Antidze et al., 2013). The difference is big and obvious. It, also, should be underlined that in the first case V is constant or in worse case a growing value. After the modification, vice versa, V is no longer constant and has a low value in many cases - based on the nature of the word it may not be 5000 but considerably less (of course not every word has about 5000 grammatically correct word forms). In fact, the number of V will change from static to dynamic and will be different at different times depending on the characteristics of the given word.

2.2 Multinomial Naïve Bayes Model

Now, let us generalize this method and apply it to other cases, for example, Multinomial Naive Bayes Model (3).

$$P(w_i|c_j)=\frac{\text{count}\left(w_i,c_j\right)}{\sum_{w\in V}\text{count}\left(w,c_j\right)} \quad (3)$$

In this case, the probability that the given word or document w_i belongs to the given class c_j depends on the number of unique words contained in the corpus, i.e., On V. This unit plays a vital role in the quantity of iteration as in the value of probability. In our situation, decreasing V increases the calculation speed and (it also requires less computer resources), in addition, the probability - a new value (unit) specific to the context of this algorithm. It is necessary to mention that the classical V - is the unit size and, in our approach - the consequence or collection subset (4).

$$V_{New}=\{v_1, v_2, v_3 \ldots v_n\} \quad (4)$$

In the specific case we have (5):

$$V_{New}= V_{Particular}$$

And therefore end up with (6):

$$P(w_i|c_j)=\frac{\text{count}(w_i,c_j)}{\sum_{w\in v_{particular}}\text{count}(w,c_j)} \quad (6)$$

Now let's consider the general case when the given word is entirely unknown to the model. Without a doubt, according to the formula mentioned above, its quantity will equal zero, and consequently, the classification process in sum will be zero, too. It is obvious that such a case should be avoided.

To avoid this, there is of course a simple solution to ensure that such issues do not interfere with the probability calculation process.

2.3 Laplace (add-1) smoothing for Naïve Bayes

Hence, let's use the Laplace (add-1) smoothing for Naïve Bayes which is represented in the following formula (Algorithm for calculating probability) (7):

$$P(w_j|c)=\frac{\text{count}(w_i, c)+1}{\sum_{w\in V}\text{count}(w,c)+|V|} \quad (7)$$

The (6) depicts that even if in the numerator, count (w_i, c) is equal to zero, it will always have a constant one added to it, which completely excludes the existence of zero in the sum. It should be mentioned that the input of a constant value is normalized-compensated by the number of words in the corpus. Using our method increases the probability and spares the calculating resource expanses, as shown in the formula below:

$$P(w_j|c)=\frac{\text{count}(w_i, c)+1}{\sum_{w\in v_{particular}}\text{count}(w,c)+|v_{particular}|} \quad (8)$$

On the other hand, our approach looks at cases whose implementation simplifies the decision-making process for the system. For instance, it implies improving the notion of a confusion matrix by giving additional mechanisms. However, this issue is not discussed in detail in this paper.

2.3 Language Modeling - Add-one (Laplace) smoothing

In the direction of creating a language model, we additionally used the morphological analysis module. We tried to generalize the existing approaches and reduce the dependence on specific words. In particular, at first, let's try to calculate the probability that the word 'გიორგის' (Eng.: Giorgi, a name) is followed by 'გადაუწერია' (Eng.: copied) with the algorithm (9) (Maximum Likelihood Estimation).

$$P_{MLE}(w_i|W_{i-1})=\frac{c(w_{i-1},w_i)}{c(w_{i-1})} \quad (9)$$

For this, of course, it is necessary to have the initial text (or in other words a train set – a raw information) and then perform the actions corresponding to the above algorithm. To avoid division by zero, add-one estimation should be used:

$$P_{Add-1}(w_i|W_{i-1})=\frac{c(w_{i-1},w_i)+1}{c(w_{i-1})+V} \quad (10)$$

Our approach consists in reducing V and also having a morphological description of words instead of specific words in the word base (in general, we can say that all the algorithms used for natural language computer processing depend more or less on the number of unique words, or in our particular case – $V_{particular}$. Hence, we can assume that it is possible to optimize the algorithms that are not discussed in this paper).

$$P_{Add-1}(w_i|W_{i-1})=\frac{c(w_{i-1},w_i)+1}{c(w_{i-1})+V_{particular}} \quad (11)$$

About 145 different types of verbs and 12 types of nouns are included in our morphological synthesis and analysis system. It is possible on them:

a) Morphological synthesis, and
b) conducting morphological analysis.

These two modules allow us to generate the desired word / word-form (if such exists with the data selected by the user) by giving the morphological categories and unchanged part of some x word. Or vice versa, it is possible to decompose the given word into morphemes and determine the category of its elements. See images: Figure 3, Figure 4 Figure 5.

Figure 4. Module for morphological synthesis of verbs.

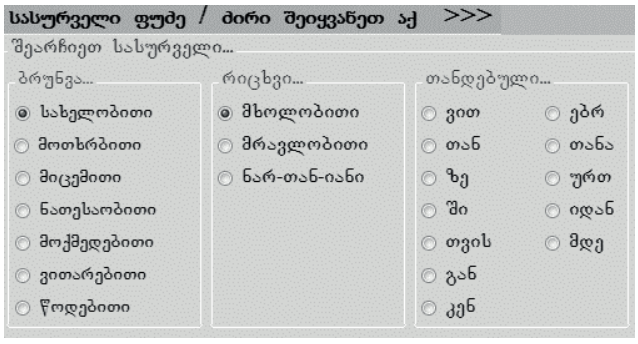


Figure 5. Module for morphological synthesis of nouns.

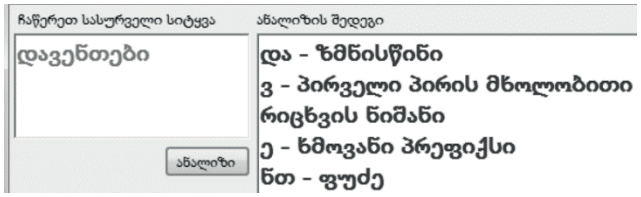


Figure 5. Module for morphological analysis of verbs

Figure 5 – on the left side is given the verb 'დავეწთები' and on the right side is the result of its morphological analysis (Aho and Ullman, 1972; Kardava et al., 2017):

და - pre verb;
ვ - singular number sign;
ე - vowel prefix;
ნ თ - root;
...

In short, as a result of the analysis of the given text, we can generally have:
noun_t1 -> verb_t45. In more detail: (noun_t1)_root_t1+morph_1-> (verb_t45)_morph1+morph2+root_t45+morph3+morph4

If a word data base and a language model are built using this approach, their size will be significantly reduced. And those words that have exactly the same characteristics, but are not stored in them, will still be processed. That is, instead of 'გიორგის' (Eng.: Giorgi, a name) we can use any other name that is exactly the same type from the 12 mentioned above.

We can give a more general example from the case of CMU Sphinx. As you know, along with other required data, we need to have audio files to train the acoustic model. If the training process is successful, the speech recognition system can only recognize the words that are in the language model (that is, words whose audio versions were present during training). If it becomes necessary to recognize other words, then they should be additionally included in the language model (their audio versions are no longer needed after successful training) (Kardava, 2016; Kardava et al. 2016). According to our approach, instead of words, morphological definitions of words should be stored in the model, which does not require the existence of a specific word(s). That is, let's make a comparison with the elementary issue of programming. In a strongly typed programming language, if we have a variable of type 'int', it can take any value within the range of type 'int' at various times. So, we do not need to write all the values specifically, instead we have only one variable.

Similarly, implementing our approach reduces the number of words in the database and makes the system more flexible to unknown words. However, it should be noted here that this part is still in testing mode, but it is working successfully so far.

3. Final Remarks

This research was inspired and supported by Shota Rustaveli National Science Foundation within the project: "Natural science (chemical-technological and mineralogical-gemological) knowledge in Georgian and Eastern manuscripts preserved in the antiquities of Georgia "(grant agreement FR-21-620). This project aims to determine the importance of Georgian, Arabic and Persian natural science (chemistry-technology, mineralogy-gemology) manuscripts preserved in the antiquities of Georgia in the world scientific heritage - by including interdisciplinary research, systematization and international circulation of the received data, interactive catalog (electronic database), geo-informational map and through the creation and distribution of relevant public ations. In the given paper, we discussed how to reduce

according to our approach the size of the word database and how to optimize some existing NLP algorithms. We believe that the mentioned technique will work successfully in the direction of computer processing of the Georgian language, even after its completion test. In addition, we do not rule out the possibility of new circumstances that may make the process even easier for us. However, since we are still in the experimental stage, it is difficult to say anything with certainty in advance. But it should be underlined the fact that so far its positive potential is obvious. As for the generalization of the same approaches to other similar languages, this is a matter of further collaboration and research.

4. Acknowledgements

I would like to thank Professor Zygmunt Vetulani for his valuable and important advice.

References

Jurafsky, D. and Martin, J.H. *Speech and Language Processing - An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Draft of October 16.

Kardava, I., Gulua, N., Antidze, J., Toklikishvili, B., Kvaratskhelia, T. (2022). *Computer Application of Georgian Words*. In: Vetulani, Z., Paroubek, P., Kubis, M. (eds) Human Language Technology. Challenges for Computer Science and Linguistics. LTC 2019. Lecture Notes in Computer Science(), vol 13212. Springer, Cham. https://doi.org/10.1007/978-3-031-05328-3_7

Acho, A. and Ulman, J., (1972). The Theory of Parsing, Translation and Compiling, vol. 1, USA: Englewood Cliffs.

Kardava, I., Esiava, E. (2022). *NATURAL LANGUAGE PROCESSING IN THE STATE DECENTRALIZED SYSTEMS OF GEORGIA.* GESJ - Computer Sciences and Telecommunications 2(62). Pp. 19-24.

Kardava, I., Gulua, N., Antidze, J., Toklikishvili, B. (2019). *Morphological Synthesis and Analysis of Georgian Words.* Human Language Technologies as a Challenge for Computer Science and Linguistics - 2019. Pp. 232-235.

Kardava, I. (2016). *Georgian Speech Recognizer in Famous Searching Systems and Management of the Software Package by Voice Commands in Georgian Language*. Conference Proceedings – Third International Conference on Advances in Computing, - Electronics and Communication. 10.15224/978-1-63248-064-4-02. 6-9.

Antidze, J., & Gulua, N., Kardava, I. (2013). *The Software for Composition of Some Natural Languages' Words.* Lecture Notes on Software Engineering. 10.7763/LNSE.2013.V1.64. pp. 295-297.

Kardava, I., Antidze, J., & Gulua, N. (2016). *Solving the Problem of the Accents for Speech Recognition Systems.* International Journal of Signal Processing Systems. 4. 235-238. 10.18178/ijsps.4.3. p.p 235-238.

MelikiShvili, D. (2001). The System of Georgian Verbs Conjugation, Tbilisi, Georgia: Logos Press.

Kardava, I., Tadyszak, K., Gulua, N., Jurga, S., (2017). *The software for automatic creation of the formal grammars used by speech recognition, computer vision, editable text conversion systems, and some new functions.* Proceedings Volume 10225, Eighth International Conference on Graphic and Image Processing (ICGIP 2016); 102251Q https://doi.org/10.1117/12.2267687.

Melikishvili, D., (2008). *On Georgian Verb-forms Classification and Qualification Principles,* Problems of Linguistics, 1, pp. 30-35.

DARIAH.PL MultiCo Multimodal Corpus

Maciej Karpiński, Ewa Jarmołowicz-Nowikow, Katarzyna Klessa, Michał Piosik, Janusz Taborek

{maciej.karpinski|ewa.jarmolowicz-nowikow|katarzyna.klessa|michal.piosik|janusz.taborek}@amu.edu.pl

Faculty of Modern Languages and Literatures, Adam Mickiewicz University, Poznań, POLAND

Abstract

Contemporary studies on interpersonal communication confirm that in order to understand and model this multifaceted process, not only speech but also other components, including gestures, facial expressions, or body postures, must be taken into account. In our contribution, we present a corpus designed to support and facilitate such a research approach. The 15-hour corpus will comprise three sections, representing monologues, dialogues and multilogues. While monologue and multilogue sections will be based on materials available in public media or archives, the dialogue section will contain task-oriented dialogues recorded specifically for the present resource. The speakers will be young to middle-aged Polish adults. Speech will be transcribed orthographically and phonemically, segmented into words, syllables and phones. Body movement annotation will vary among the subcorpora. Along with the dialogue recordings, synchronised depth-sensor data will be available. In the monologue and multilogue subcorpora, manual annotation of selected categories of gestures will be provided. The corpus is aimed to fill a significant gap in the body of Polish resources, and to encourage studies of multimodal communication from a number of perspectives. Potential applications of the corpus include education, media and industry. The corpus will be freely available for research purposes via the CLARIN-PL infrastructure upon project completion.

Keywords: multimodal corpus, multimodal communication, Polish, persuasive speech, sports language

1. Motivation and scientific background

Interpersonal face-to-face communication typically involves more than one sensory modality [e.g., Higham and Hebets, 2013; Bonacchi and Karpiński, 2014; Fröhlich et al., 2019]. While their roles and contributions may differ, each of them brings in information which may prove essential for the process of interaction and for achieving communication goals. In order to understand, explain, and model its mechanisms, visual and auditory modalities are taken most often into account. Even though this perspective seems to be commonly accepted, the theoretical complexity of such a holistic, interdisciplinary approach, and technological challenges related to data acquisition, processing, and analysis are often intimidating and may severely impede research progress. Data acquisition and annotation for multimodal communication analysis is expensive and time-consuming as it requires extensive use of technical infrastructure, dealing with numerous issues related to sensitive data, working with large groups of recording participants, and tedious, often manual, identification and description of various categories of phenomena. A limited number of multimodal corpora has been reported for the Polish language (e.g., Jarmołowicz et al. 2007; Lis 2012; Czyżewski et al. 2017), but most of them are highly specialised (e.g., visual support for speech recognition), while noone features combined multilevel gesture and speech segmentation and annotation, being based on quasi-spontaneous speech. In order to fill this gap and address the requirements of multimodal communication research, a new corpus of Polish is being developed. It will consist of three sections, containing media (movie and sound files), time-aligned annotations (including transcription of speech and description of selected body movements), and metadata on speakers and recording settings.

2. Corpus design

Three categories of communicative situations will be represented in the corpus, (1) monologue, (2) dialogue, and (3) multilogue, to provide researchers and commercial users with information on a wide spectrum of communicative behaviour. In monologues, speakers may be focused on the audience as a group of people present at the location or only imagined by the speaker. In dialogues, speakers turn to each other and tend to address each other. In multilogues, the entire group or its part may be addressed by each participant. In dialogues and multilogues, the process of turn-taking may be tracked, while in monologues, one may observe the arrangement of speeches less influenced by the flow of interaction. We assume that these differences entail also changes in communicative behaviour that can be detected in prosody and gesticulation.

In order to find samples which would represent each of the categories and would meet other requirements regarding the coherence of the content and technical quality, publicly available media resources were explored.

a. Monologues: Parliamentary speeches and motivational speeches were selected as examples of persuasive speeches addressed to the public present in the room and listening to the speaker. In the former case, the material was acquired from online archives of the Polish Parliament, while in the latter, from the TEDx YouTube platform (see: acknowledgements).

b. Dialogues: Dialogue recordings meeting our initial assumptions related to the topic (same for all the talks),

structure (stable, relatively fixed structure), coherent profile of the participants (young, educated speakers), or duration, are difficult to find in public media resources except for interviews but these are often characterised by a high degree of asymmetry and framing which is unnatural for a dialogue (speakers sitting side-by-side), or significantly limits visibility (speakers standing face-to-face, photographed from one side). Because of the special importance of dialogue recordings as representing the most prototypical communication settings, it was decided to record task-oriented dialogue interaction in a laboratory using multiple cameras and a motion capture system.

c. Multilogues: Obtaining multilogue recordings of acceptable quality and meeting interactivity and spontaneity criteria would be technically challenging and expensive (e.g., Carletta 2007; Campbell 2009). Therefore, publicly available media materials were selected for this section of the corpus. Sports programs engaging a number of participants moderated by a journalist turned out to meet the criteria best as they often involve emotionally engaging discussion and expressive participants while still being moderated by a journalist and following a certain scenario.

2.1 Monologues: Persuasive speeches

The monologue section will comprise two subcorpora of persuasive speech, each of a total duration of 2.5 hours. The parliamentary subcorpus will contain 33 speeches from the parliamentary debate that took place on March 2nd and March 27th 2020, and concerned the preparation of the Polish state to prevent the spread of the SARS-CoV-2 pandemic in Poland and Europe. The duration of the speeches ranged from ca. one minute to 14 minutes. In a situation where the speaker spoke twice on a given day, only her/his longer speech was taken into account. Speeches lasting less than a minute in which the speaker formulated questions addressed to the government, were also excluded.

The parliamentary speeches are strongly persuasive, meant to convince the audience to the views presented by the speaker and to criticise and ridicule the opposition. Taking stances by politicians is often accompanied by strong emotions. During speeches, speakers do not change their places, remain behind the rostrum, so only the upper part of the body is visible to the public.

The subcorpus of motivational monologues will consist of eleven speeches delivered during the TEDx conferences organised in Poland. The duration of speeches ranges from 10 to 20 minutes. They touch upon events or situations that strongly, usually positively, influenced the lives of the speakers. The speakers support the members of the audience by sharing their life experiences and knowledge. They often move around the stage but remain visible all the time, with certain limitations. The main criterion for the selection of speeches is their motivational character, the final appeal addressed to the audience, and the technical quality of the sound.

2.2. Dialogues: Task-oriented scenario

Dialogue is the primary form of language communication. In order to obtain data on dialogue interaction, a task was designed in which the participants were asked to arrange a room for a student. They were provided with images of all the pieces of furniture they could use. They were not given any hints on the gender of the student and were not instructed to adjust the room more to (hypothetical) boys' or girls' preferences. However, some pieces of furniture or equipment might have looked as stereotypically more "male" or "female". The instruction included a suggestion that the room to be arranged is very similar to the laboratory where the recording session was held regarding the placement of the door and windows. This way, the participants could have referred to the real space around them. Dialogue sessions were recorded in an acoustically treated room in two conditions: (1) mutual visibility of the participants and (2) the lack of mutual visibility of the speakers. The recordings for each condition were carried out with a different group of speakers. Participants were offered gift vouchers for taking part in the recording.

2.3. Multilogues: Sports television broadcast

There is a shortage of corpora containing different genres of texts that represent sports language and sports communications and there is no corpus of the Polish sports language. The only available resources of Polish sports language are football reports "Minute by minute" (MBM), as part of the Multilingual Research Resource on the Language of Football Reports collected at fussballlinguist.de (Meier, 2017).

The sports collection in the MultiCo corpus will contain 15 television broadcasts entitled "4-4-2", from the Polish Television TVP, broadcast in the period from February 1st to May 10th 2021. Each program lasts about 50 minutes, including a discussion with a moderator and three football experts in particular areas (ex-players, coaches, journalists), as well as commercial ads. In some parts of the program discussions are accompanied by excerpts from football games or other sports events. Those parts, as well as commercials, were excluded from transcription and annotation. Twelve programs were transcribed orthographically and six of them were additionally annotated for non-verbal behaviour, cf. Section 4. The result is also 2.5 hours of transcribed and annotated multilogues.

The MultiCo sport multilogues on the topics of football exemplify conversational uses of special terms of the football language (as a language for specific purposes), as well as numerous proper nouns (countries, football teams, players, coaches, referees, etc.).

3. Data acquisition and processing

3.1. Data acquisition

As mentioned in Section 2, the data included in the MultiCo corpus differ in their characteristics and were either acquired from diverse internet sources or recorded in a laboratory according to a custom-designed recording scenario. The subcorpora containing the recordings of parliamentary speeches, TVP Sport discussions and TEDx talks were downloaded directly from the respective institutions' official websites or YouTube channel, i.e.,

https://www.sejm.gov.pl (as .flv files), https://sport.tvp.pl/77422/magazyn-pilkarski-442 (.mp4) and https://www.youtube.com/@TEDxTalks (.mp4), after consulting the conditions of usage with the data holders. As a result of the consultations, the licences for using and sharing particular subcorpora were defined (cf. also Section 5 below).

The task-oriented dialogues were recorded in laboratory

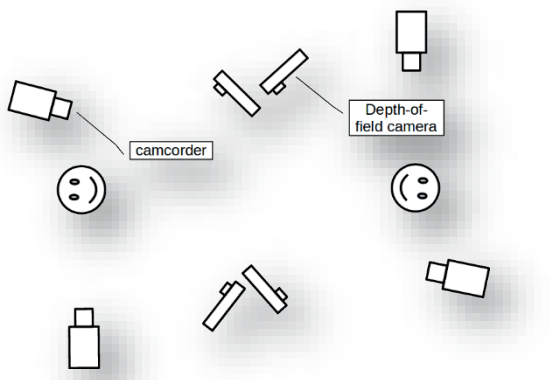


Fig. 1: Video recording setup for dialogue corpus (four camcorders and four depth-of-field cameras)

conditions. Each participant was recorded using (1) two Sony PXW-90 camcorders (at 50 fps, one frontal and one placed on the right-hand side of the speaker; the frontal one at the distance of ca. 4 meters while the side one in the distance of ca. 2 meters from the speaker; (2) two depth-sensor Kinect 2 cameras arranged at a ca. 80-degree angle, each ca. 2.7 meters from the speaker; (3) a miniature head-on condenser microphone DPA4080, with the capsule located in the corner of the lips, and the large membrane Neumann TLM103 microphone placed in front of the speaker, slightly above the head, in the distance of 30-40 cm from the lips. Depth sensor cameras were linked to an MS Windows workstation running iPi Soft (www.ipisoft.com) markerless motion capture system in order to capture moving images with depth information. Microphones were connected to a multi-channel audio interface Steinberg UR816C, and the sound was recorded using a separate workstation at 44.1kHz/24bit on four independent channels with Steinberg Cubase DAW software. The video recording setup for the mutual visibility condition is shown in Figure 1. The same equipment in a similar configuration was used for the second condition where a blend was placed between the interlocutors and they did not see each other during conversation.

3.2. File formats and (meta)data management

For each media item (e.g., a single TV sports broadcast or a single parliamentary speech) acquired for the MultiCo corpus (henceforth, a session), a bundle of data files was created. Each bundle includes multimedia files (audio and video), speech and gesture multilayer annotation files, and metadata.

For the monologues and multilogues, all video files were – if not downloaded in this format – converted to .MP4 (with the resolution of 1920x1080 or 1280x720 pixels). Two audio files (.WAV format; 16 kHz/16bit and 44.1 kHz/16bit) were extracted for each of the sessions.

For the dialogue subcorpus, the following files were stored in each session: (a) four high-resolution video files; (b) a minimum of two video files recorded with depth sensor cameras; (c) four monophonic audio files coming from two head-on and two large-membrane microphones. Amplitude in the audio files was normalised. They were trimmed and synchronised with corresponding video files. Motion capture files were converted into skeleton animations in the BVH Animation format.

Multimedia files and accompanying data were imported to Corpus Mini database system (Karpiński and Klessa, 2018, 2021) which supported the management of analysis and annotation. Prior to the import procedure, each file bundle was inspected for technical issues, the file names were adjusted according to the requirements of the system, and the crucial metadata were attributed to each session (e.g., basic information about the speaker, including coded name and gender information as well as subcorpus type). At the next stage, the annotation files were generated in Corpus Mini based on previously designed templates for EAF (ELAN) format and ANTX (Annotation Pro). Data management was arranged using Corpus Mini system based on client-server architecture. In this way, easy authorised access, data sharing, efficient collaborative work on the material, and security of the data were simultaneously ensured.

4. Annotation specifications and procedures

Multilayer annotations of speech and gestures were created for each session using freely available software tools (Annotation Pro, Praat, and ELAN). Thanks to the interoperability of the file formats, it is possible to easily convert file formats (e.g., using the Annotation Pro import/export options) and integrate all the layers within one desired file format, e.g., in order to perform annotation mining for cross-modal analyses of interaction (as, for example, in Czoska et al., 2015).

4.1. Speech transcription and segmentation

All recording sessions were transcribed orthographically and segmented into phrases by a team of trained annotators using Annotation Pro (Klessa et al., 2013). The phrase boundaries were inserted based on grammatical and prosodic rules following the paradigms used, e.g., by Karpiński et al., 2005. In the case of parliamentary speeches, the procedure was enhanced to a certain extent by using the orthographic transcripts provided at the Polish parliament website. Since the original stenographic reports of the parliament transcripts are usually stylistically edited, they do not precisely match what was said and thus cannot be used directly as transcriptions. In order to provide synchronised transcripts, human annotators inspected both the recordings and original transcripts, and introduced necessary adjustments.

Phonetic transcription and segmentation of the recordings were based on the above orthographic input and were: (1) generated automatically using the ANNPRO transcription and segmentation module (Klessa et al., 2022) developed as

a desktop variant of CLARIN-PL online speech alignment tools (e.g., Koržinek et al., 2017); (2) manually verified by a team of expert phoneticians to eliminate major transcription and boundary position errors (Machač and Skarnitzl, 2009).

Apart from the transcription of speech, each multilayer annotation file includes additional layers for paralinguistic information (e.g., speaker noises and hesitation markers) as well as extralinguistic information (stationary and intermittent noises; cf. also Fischer et al., 2000; Karpiński and Klessa, 2021, pp. 77–78).

4.2. Body movement annotation

The annotation system is focused on hand gestures, head and trunk movements. Hand movements were annotated as gesture phrases as defined by Kendon (1972, 2005) and Kita (1990), separately, on two independent layers, for the right hand and left hand. Only phrase boundaries were marked. Head and trunk movements were annotated on separate layers. The description of head movements was based on the categories described by Allwood and Cerrato (2003) and Kousidis et al. (2013) and included the following tags: 1. Nod (forward movement of the head going up and down, single), 2. Nod multiple, 3. Jerk (backward movement of the head which is usually single), 4. Turn (rotation left or right), 5. Tilt (sideways nod), 6. Turn/Nod (rotation right or left and head going up and down), 7. Reading (head down, the participant looks at notes). The inventory of tags used for the description of the trunk movements comprised of 1. Move forward (forward movement of the whole trunk), 2. Move backwards (backward movement of the whole trunk), 3. Side bends (side movement to the right or to the left), 4. Side (rotation movement to the right or left). The system was applied in the description of parliament speeches (subcorpus of persuasive speeches) and task-oriented dialogues corpus. In the TEDx section of the subcorpus of persuasive speeches, efficient and precise annotation of head and trunk movements proved impossible because the speakers tended to continuously move on the stage. The annotation of nonverbal behaviour in the sports tv broadcast multilogues is based on similar principles as in the case of parliamentary speeches. Separate tiers are defined for each participant. Head and trunk movements are annotated only in the sections of the recordings where hand movements occur.

5. Availability and applications

The corpus described in the present contribution will be available for research applications after the project is finished (December 31st, 2023). The shared resources will include multilayer annotations of speech and gesture as well as audio and video recordings. The way of sharing the multimedia material will differ for particular sub-corpora, depending on the copyright agreements and licensing details. The multimedia recordings for the Sport, Parliament and Dialogues sub-corpora will be shared directly from project-established infrastructure (in collaboration with CLARIN-PL, e.g., Pol et al., 2018), while for the TEDx sub-corpus, the recordings will be linked to their original online locations within TEDx services while annotations will be stored and shared using the same project infrastructure as the rest of the resource.

MultiCo corpus provides an insight into the present shape of selected interpersonal communication styles and of the Polish language usage. It can be used for virtually any studies on spoken language and speech itself, but its primary purpose is to enable studies of the interactions between various layers of communicative events, both in terms of linguistic and paralinguistic features, within and across different modalities. The corpus will be used in basic phonetic research, including prosodic analyses, as well as in multimodal communication studies, including speech-gesture coordination analyses, as well as for multimodal meaning composition exploration, or multimodal emphatic speech analysis. Dialogue recordings will provide research material for multimodal interaction and alignment analysis, including the process of turn-taking. Other applications of the corpus may include interpersonal communication training or rhetoric courses as a searchable library of examples. Motion capture data along with annotated speech recordings may be used in the design of virtual agents and in full body visual speech synthesis systems. Already at the present stage, preliminary analyses of the data in some corpora have been conducted in two MA theses. Larysz (2022) analysed verbal and non-verbal means of expressing emotions that can be found in sports discussions, while Wicijowska (2022) explored the structural metaphors according to Lakoff and Johnson's Conceptual Metaphor Theory (CMT). More studies dedicated to the multimodal aspects of monologues and conversations are planned in further research projects.

Acknowledgements

The development of the Multico laboratory and the corpus has been supported by the project: *Digital Research Infrastructure for the Humanities and Arts DARIAH-PL*, funded from the Intelligent Development Operational Programme, Polish National Centre for Research and Development, ID: POIR.04.02.00-00-D006/20.

We are pleased to acknowledge the support from TVP Sport represented by the Director Marek Szkolnikowski as well as TEDx Team for agreeing to use their materials.

We are grateful to CLARIN-PL for supporting the process of transcription and to Anna Gralińska-Brawata, Izabela Grabarczyk, Brygida Sawicka-Stępińska, and Maria Szymańska for their engagement in the process of orthographic transcription.

We are deeply indebted to all those who took part in the recording of the dialogue section of the corpus.

References

Allwood, J. and Cerrato, L. (2003, September). A study of gestural feedback expressions. In: *First nordic symposium on multimodal communication,* pp. 7-22. Copenhagen.

Boersma, P. and Weenink, D. (1992–2022). Praat: doing phonetics by computer [Computer program]. Version 6.2. Retrieved on 10 November 2022 from: https://www.praat.org

Bonacchi, S. and Karpiński, M. (2014). Remarks about the use of the term "multimodality". In: *Journal of Multimodal Communication Studies*, vol. 1, pp. 1-7.

Campbell, N. (2009). An audio-visual approach to measuring discourse synchrony in multimodal conversation data. In: *Linguistic Theory and Raw Sound*, 40, 199.

Carletta, J. (2007). Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus. In: *Language Resources and Evaluation*, 41(2), pp. 181-190.

Czoska, A., Klessa, K., Karpiński, M. and Nowikow-Jarmołowicz, E. (2015). Prosody and gesture in dialogue: Cross-modal interactions. In: *Proceedings of 4th Gesture and Speech in Interaction (GESPIN) Conference*, Nantes, France, pp. 83-88.

Czyżewski, A., Kostek, B., Bratoszewski, P., Kotus, J. and Szykulski, M. (2017). An audio-visual corpus for multimodal automatic speech recognition. *Journal of Intelligent Information Systems,* 49, pp. 167-192.

Fischer, V., Diehl, F., Kiessling, A. and Marasek, K. (2000). Specification of Databases – Specification of annotation. *SPEECON Deliverable D214.*

Fröhlich, M., Sievers, C., Townsend, S. W., Gruber, T. and van Schaik, C. P. (2019). Multimodal communication and language origins: integrating gestures and vocalizations. *Biological Reviews*, 94(5), pp. 1809-1829.

Higham, J. P. and Hebets, E. A. (2013). An introduction to multimodal communication. In: *Behavioral Ecology and Sociobiology*, 67(9), pp. 1381-1388.

Jarmołowicz, E., Karpiński, M., Malisz, Z. and Szczyszek, M. (2007). Gesture, prosody and lexicon in task-oriented dialogues: multimedia corpus recording and labelling, In A. Esposito, M. Faundez-Zanuy, E. Keller, M. Marinaro (Eds.) *Verbal and Nonverbal Communication Behaviours*, LNAI 4775, 99-110, Springer, 2007

Karpiński, M., Kleśta, J., Baranowska, E. and Francuzik, K. (2005). Interphrase pause realization rules for high quality Polish speech synthesis. In: *Speech Analysis, Synthesis and Recognition (SASR) in Technology, Linguistics and Medicine*, Szczyrk 2003. pp. 85-89. AGH Kraków.

Karpiński, M. and Klessa, K. (2018). Methods, tools and techniques for multimodal analysis of accommodation in intercultural communication. In: *Computational Methods in Science and Technology*, 24(1), pp. 29-41.

Karpiński, M. and Klessa, K. (2021). *Linguist in the field: a practical guide to speech data collection, processing, and management.* Poznań: Wydawnictwo Rys. ISBN 978-83-66666-89-4.

Klessa, K., Karpiński, M. and Wagner, A. (2013). Annotation Pro – a new software tool for annotation of linguistic and paralinguistic features. In: Hirst D., Bigi B. (Eds.) *Proceedings of the Tools and Resources for the Analysis of Speech Prosody (TRASP) Workshop*, Aix en Provence, pp. 51-54.

Klessa, K., Koržinek, D., Sawicka-Stępińska, B. and Kasperek, H. (2022). ANNPRO: A Desktop Module for Automatic Segmentation and Transcription. In: Vetulani, Z., Paroubek, P., Kubis, M. (Eds.) *Human Language Technology. Challenges for Computer Science and Linguistics. LTC 2019. Lecture Notes in Computer Science, vol 13212.* Springer, Cham. https://doi.org/10.1007/978-3-031-05328-3_5

Kendon, A. (1972). Some relationships between body motion and speech. In: Pope B., Siegman A. W. (Eds.) *Studies in dyadic communication.* New York: Pergamon Press, pp. 177-210.

Kendon, A. (2005). *Gesture. Visible Action as Utterance.* Cambridge: Cambridge University Press.

Kita, S. (1990). *The temporal relationship between gesture and speech: A study of Japanese-English bilinguals.* [Unpublished Master Thesis, University of Chicago].

Koržinek, D., Marasek, K., Brocki L. and Wołk, K. (2017). Polish read speech corpus for speech tools and services. In: Selected papers from the CLARIN Ann. Conf. 2016, Aix-en-Provence, 26–28.10.2016, *CLARIN Common Language Resources and Technology Infrastructure.* pp. 54–62. No. 136. Linköping: Linköping University Electronic Press.

Kousidis, S., Malisz, Z., Wagner, P. and Schlangen, D. (2013). Exploring annotation of head gesture forms in spontaneous human interaction. In: *Proceedings of the Tilburg Gesture Meeting* (TiGeR 2013).

Larysz, J. (2022). *Sprache, Gestik und Emotionen in den Sportdiskussionen. Eine multimodale Analyse des Programms 4-4-2.* [Unpublished Master Thesis, Adam Mickiewicz University, Poznań].

Lis, M. (2012, May). *Polish Multimodal Corpus-a collection of referential gestures*. In: LREC, pp. 1108-1113.

Machač, P. and Skarnitzl, R. (2009). *Principles of phonetic segmentation*. Prague: Epocha.

Maier, S. (2017). Korpora zur Fußballlinguistik – eine mehrsprachige Forschungsressource zur Sprache der Fußballberichterstattung. In: *Zeitschrift für germanistische Linguistik,* 42(2), pp. 345-349.

Pol M., Walkowiak T. and Piasecki M. (2018). Towards CLARIN-PL LTC Digital Research Platform for: Depositing, Processing, Analyzing and Visualizing Language Data. In: Kabashkin I., Yatskiv I., Prentkovskis O. (Eds.) Reliability and Statistics in Transportation and Communication. *RelStat 2017. Lecture Notes in Networks and Systems, vol 36.* Springer, Cham. https://doi.org/10.1007/978-3-319-74454-4_47

Wicijowska, J. (2022): *Metaphern im Sportdiskurs. Am Beispiel der polnischen Diskussionen zu Fußballspielen im Programm 4-4-2*. [Unpublished Master Thesis, Adam Mickiewicz University, Poznań].

Wittenburg, P., Brugman, H., Russel, A., Klassmann, A. and Sloetjes, H. (2006). ELAN: a professional framework for multimodality research. In: *Proceedings of the 5th Language Resources and Evaluation Conference*, Genoa, Italy, pp. 1556-1559.

Text classification dataset and analysis for Uzbek language

Elmurod Kuriyozov[1,2], Ulugbek Salaev[1], Sanatbek Matlatipov[3], Gayrat Matlatipov[1]

[1]Urgench State University, 14, Kh.Alimdjan str, Urgench city, 220100, Uzbekistan
{elmurod1202, ulugbek.salaev,gayrat}@urdu.uz
[2]Universidade da Coruña, CITIC, Grupo LYS, Depto. de Computación y Tecnologías de la Información, Facultade de Informática, Campus de Elviña, A Coruña 15071, Spain
e.kuriyozov@udc.es
National University of Uzbekistan named after Mirzo Ulugbek, 4 Universitet St, Tashkent, 100174, Uzbekistan
s.matlatipov@nuu.uz

Abstract

Text classification is an important task in Natural Language Processing (NLP), where the goal is to categorize text data into predefined classes. In this study, we analyze the dataset creation steps and evaluation techniques of multi-label news categorisation task as part of text classification. We first present a newly obtained dataset for Uzbek text classification, which was collected from 10 different news and press websites and covers 15 categories of news, press and law texts. We also present a comprehensive evaluation of different models, ranging from traditional bag-of-words models to deep learning architectures, on this newly created dataset. Our experiments show that the Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) based models outperform the rule-based models. The best performance is achieved by the BERTbek model, which is a transformer-based BERT model trained on the Uzbek corpus. Our findings provide a good baseline for further research in Uzbek text classification.

Keywords: Text classification, news categorization, Uzbek language, dataset

1. Introduction

Text classification is a critical task in the field of natural language processing (NLP), where the goal is to categorize a text document into predefined classes. This task is essential in many real-world applications such as sentiment analysis, spam detection, and topic modelling. With the massive amount of unstructured data generated daily, text classification provides a means to make sense of this data and derive meaningful insights.

In recent years, deep learning models have been widely used in text classification (Minaee et al., 2021), yielding excellent results. However, most research works in text classification have focused on high-resource languages such as English (Cruz & Cheng, 2020). There is a significant gap in text classification research for low-resource languages, Uzbek being no exception.

The primary objective of this work is to contribute to the NLP research community by addressing the text classification challenge for the Uzbek language, in the example of a multi-label news categorization task. We present a new Uzbek text classification dataset and evaluate the performance of various models on this dataset. The models range from traditional rule-based approaches, such as word and character n-grams-based support vector machine (SVM), to more advanced deep learning models, such as recurrent neural networks (RNN) and convolutional neural networks (CNN). We also analyse the dataset further by using transformer-based models, such as mBERT - a multilingual BERT model trained on more than 100 languages (Devlin et al., 2019), and BERTbek - a monolingual BERT language model trained on an Uzbek news corpus. Our experiment results indicate that neural-network-based models outperform the rule-based ones, and the BERTbek model achieves the best result with over 85% of the F1-score.

Uzbek language. The Uzbek language is spoken by over 30 million people and is primarily used in Uzbekistan and surrounding Central Asian countries. It is a Turkic language that has been heavily influenced by both Russian, Arabic and Persian for geographic and historical reasons. As a low-resource language, there is limited research and resources available for Natural Language Processing (NLP) tasks in Uzbek, making the creation and utilization of NLP resources a crucial step towards promoting the digitalization of the Uzbek language. Despite this, Uzbek has a rich literary history and continues to be an important part of the cultural heritage of the Uzbek people.

Its official alphabet is Latin, and its grammar is close to other languages in the Turkic family, which differs vastly from the more commonly studied languages in NLP such as English and Chinese. This presents a challenge for NLP tasks in Uzbek, as models trained on those languages may not be effective in handling the nuances of Uzbek text. The development of NLP resources and models specifically for Uzbek can help advance research in the field and promote the use of technology in Uzbek-speaking communities[1].

The rest of the paper is organized as follows: We provide an overview of text classification and highlight some recent NLP works on Uzbek in Section 2. It is followed by the Methodology in section 3, where we describe the data collection and dataset creation process. In the Experiments section (Section 4), we describe the models used for evaluation. Moving on, Section 5 covers the results of the experiments and is followed by Section 6, where we discuss the effects and their implications. Finally, in the Conclusion

[1] More about the Uzbek language:
https://en.wikipedia.org/wiki/Uzbek_language

and Future Work section (Section 7), we provide a conclusion of the work and outline future directions.

2. Related work

Text classification has been a fundamental problem in the field of Natural Language Processing (NLP) and has numerous applications in various domains such as sentiment analysis (Medhat et al., 2014), spam detection (Jindal & Liu, 2007), and categorization of news articles (Haruechaiyasak et al., 2008). With the advancement of machine learning techniques, the performance of text classification has improved dramatically in recent years. In the early days, traditional machine learning methods such as Support Vector Machines (SVM) (Joachims & others, 1999) and Naive Bayes (McCallum et al., 1998) were used for text classification. However, the growing size of text data and the increased complexity of the tasks led to the development of deep learning methods.

One of the major breakthroughs in text classification was the use of Convolutional Neural Networks (CNNs) for sentiment analysis by Kim (Kim & Lee, 2014). This work showed that the use of convolutional layers with different kernel sizes could effectively capture local and global information from texts. Recurrent Neural Networks (RNNs) have also been widely used for text classification tasks due to their ability to model sequential data. LSTMs, GRUs, and Bi-LSTMs have been popular variants of RNNs for text classification (Liu et al., 2016; Minaee et al., 2021). The use of attention mechanisms has further improved the performance of text classification tasks. The Transformer architecture introduced by Vaswani et al. (Vaswani et al., 2017) revolutionized the NLP field with its self-attention mechanism, and the BERT model (Devlin et al., 2018) based on the Transformer architecture has become a benchmark in various NLP tasks including text classification.

NLP works on the Uzbek language.

Despite the fact that Uzbek is considered a low-resource language, there have been some efforts to develop NLP resources and models for it. Some notable works include the creation of sentiment analysis datasets (Kuriyozov et al., 2022; Matlatipov et al., 2022), semantic evaluation datasets (Salaev et al., 2022b), and stopwords datasets (Madatov et al., 2022). NLP tools such as part-of-speech taggers (Sharipov et al., 2023), stemmers, and lemmatizers (Sharipov & Yuldashov, 2022) have also been developed to support NLP research and applications on Uzbek texts. However, further efforts are needed to improve the performance of NLP models on Uzbek texts.

Rabbimov and Kobilov (Rabbimov & Kobilov, 2020) focus on a similar task of multi-class text classification for texts written in Uzbek. The authors try to create a functional scheme of text classification and develop models using six different machine learning algorithms, including Support Vector Machines (SVM), Decision Tree Classifier (DTC), Random Forest (RF), Logistic Regression (LR) and Multinomial Naïve Bayes (MNB). The authors used the TF-IDF algorithm and word-level and character-level n-gram models as feature extraction methods and defined hyperparameters for text classification using 5-fold cross-validation. Through experiments conducted on a dataset developed from articles on ten categories from the Uzbek "Daryo" online news edition, the authors achieved a high accuracy of 86.88%. The only drawbacks of this paper are that the dataset is only limited to a single news source, hence working on a relatively small amount of data, the categories are also limited to ten classes, and the analysis is limited to machine learning techniques. We aim to fill these gaps in our current work by collecting more data, creating more text classes, as well as analysing the new dataset with deep learning models.

3. Methodology

In this section, we describe the steps of data collection in detail, as well as the efforts taken to clear the collected data, make some adjustments, and create the text classification dataset.

3.1. Data collection

Since text classification requires a labelled dataset for training and evaluating the models. For our research, we collected text data from 10 different Uzbek news websites, as well as press portals, including news articles and press releases. The websites were chosen to represent a diverse range of categories, such as politics, sports, entertainment, technology, etc. The data was collected using web scraping techniques, such as Scrapy framework for Python[2] and Beautiful Soup[3] preserving the source link, source category name, its title, and the main body. Each article was labelled with its corresponding category information. The collected dataset consisted of approximately 513K articles with more than 120M words in total, providing a large and diverse corpus for text classification. All the names of sources, a number of articles obtained from each source, as well as some information regarding the volume of the text are presented in Table 1.

3.2. Dataset creation

The dataset creation process involved several steps to ensure the quality and sustainability of the data for text classification. First, repetitive news and law decrees were removed to eliminate redundancy in the data. References to images, emojis, and URLs were also removed to ensure the data only contained text relevant to the classification task.

Additionally, some of the crawled texts in the dataset were written in the Cyrillic script. To address this, the texts were transliterated into the Latin script using the UzTransliterator tool (Salaev et al., 2022a).

Initially, there were more than 40 distinct categories when all the news texts were collected, but many of them were either synonymous or very close to one another, belonging to the same field. To ensure a better representation and a balanced distribution of the data, categories with identical or very close labels and some categories with a very small number of news articles were merged together. This helped to avoid the model getting confused over categories of very similar fields, as well as being biased towards certain categories with a larger number of samples.

[2] https://scrapy.org/

[3] https://pypi.org/project/beautifulsoup4/

Category/Label	Source(s)*	# of Articles	%	# of Words	Avg. # of Words	Avg. # of Char-s
Local (Mahalliy)	1, 3, 5	149312	29.1	34.7M	232	1995
World (Dunyo)	1, 2, 3, 5	136732	26.7	21.1M	155	1282
Sport (Sport)	1, 2, 3, 4, 5	59784	11.7	11.3M	189	1512
Society (Jamiyat)	1, 2, 4, 5	55018	10.7	13.9M	253	2114
Law (Qonunchilik)	6, 7	33089	6.5	27.0M	815	7466
Tech (Texnologiya)	1, 2, 3, 5	17541	3.4	3.1M	179	1467
Culture (Madaniyat)	2, 3	12798	2.5	2.9M	226	1838
Politics (Siyosat)	1, 2, 4, 8	12247	2.4	3.4M	279	2468
Economics (Iqtisodiyot)	1, 2, 4, 5	12165	2.4	3.1M	257	2166
Auto (Avto)	3	6044	1.2	0.9M	153	1273
Health (Salomatlik)	2, 3, 4	5086	1.0	1.3M	257	2107
Crime (Jinoyat)	2	4200	0.8	0.8M	181	1488
Photo (Foto)	1, 3	4037	0.8	0.6M	150	1225
Womens (Ayollar)	3	2657	0.5	0.7M	270	2156
Culinary (Pazandachilik)	3, 9	2040	0.4	0.1M	62	498

** Notes: 1 - bugun.uz, 2 - darakchi.uz, 3 - daryo.uz, 4 - gazeta.uz, 5 - kun.uz, 6 - lex.uz, 7 - norma.uz, 8 - president.uz, 9 - zira.uz*

Table 1. Detailed information of the categories, names of their sources, percentage over the overall dataset, as well as the total and average number of words & characters per category.

All the above steps were taken to clean and pre-process the data and make it suitable for the text classification task. The final dataset consisted of a total of 512,750 news articles across 15 distinct categories, representing the Uzbek language as much as possible.

4. Experiments

For experiments on the newly created dataset, we randomly split the dataset with a 5:3:2 ratio for training, validation, and testing, respectively. During the splitting, we made sure that all the parts would have evenly distributed article categories.

In this study, we have carried out several experiments to evaluate the performance of different models on the Uzbek text classification task. The following models have been used for experiments:

- $\boldsymbol{LR_{Word\text{-}ngrams}}$**:** Logistic regression with word-level n-grams (unigram and bi-gram bag-of-words models, with TF-IDF scores);
- $\boldsymbol{LR_{Character\text{-}ngrams}}$**:** Logistic regression with character-level n-grams (bag-of-words model with up to 4-character n-grams);
- $\boldsymbol{LR_{Word+Char\text{-}ngrams}}$**:** Logistic regression with word and character-level n-grams (concatenated word and character TF-IDF matrices);
- ***RNN:*** Recurrent neural network without pretrained word embeddings (bidirectional GRU with 100 hidden states, the output of the hidden layer is the concatenation of the average and max pooling of the hidden states);
- $\boldsymbol{RNN_{Word\text{-}embeddings}}$**:** Recurrent neural networks with pretrained word embeddings (previous bidirectional GRU model with the SOTA 300-dimensional FastText word embeddings for Uzbek obtained from (Kuriyozov et al., 2020));
- ***CNN:*** Convolutional neural networks (multi-channel CNN with three parallel channels, kernel sizes of 2, 3 and 5; the output of the hidden layer is the concatenation of the max pooling of the three channels);
- $\boldsymbol{RNN + CNN}$**:** RNN + CNN model (convolutional layer added on top of the GRU layer);
- ***mBERT***: Multilingual BERT model, trained using more than a hundred languages, (including Uzbek) (Devlin et al., 2019);
- ***BERTbek:*** Monolingual BERT model trained on Uzbek news corpus[4].

We trained each model with the training dataset, fine-tuned using the evaluation dataset, and tested the model performance using the test dataset.

The rule-based models have been used as baselines to measure the performance of the neural network models. The *RNN* and *CNN* models were used to explore the ability of the recurrent and convolutional neural networks to capture the sequence information and the semantic representation of the Uzbek text data. Finally, the *BERT* model was used to evaluate the performance of the state-of-the-art language representation model in the Uzbek text classification task.

5. Results

In this section, we present the results of our experiments with the different models used for text classification on the Uzbek language dataset. We evaluated the performance of our models using several metrics including accuracy, F1-score, and precision. For each category in the dataset, the F1-scores of all experiment models and their mean scores are reported in Table 2.

Based on the model performance results, it can be concluded that the logistic regression models work best

[4] The BERTbek-news-big-cased model was used from https://huggingface.co/elmurod1202/BERTbek

Models	F1	Local	World	Sport	Society	Law	Tech	Culture	Politics	Economics	Auto	Health	Crime	Photo	Women	Culinary
$LR_{Word\text{-}ngram}$	73.6	89.8	86.5	79.2	62.3	76.1	63.4	66.3	77.1	74.5	80.7	69.2	72.2	68.5	61.2	77.1
$LR_{Char\text{-}ngram}$	72.5	88.5	89.7	76.8	60.1	77.0	60.3	64.4	75.9	73.7	81.4	71.2	68.3	65.7	60.5	74.1
$LR_{Word+Char\text{-}ngram}$	75.6	91.1	90.1	81.7	66.0	73.5	65.0	68.4	81.4	77.5	83.1	71.9	74.9	67.7	63.1	79.4
RNN	79.0	91.5	92.4	86.1	64.9	82.7	66.0	71.6	84.1	79.7	88.7	79.2	77.2	70.5	67.8	82.5
$RNN_{Word\text{-}emb.}$	80.4	93.6	**93.0**	88.1	66.8	81.6	66.9	73.4	82.9	82.5	89.1	82.5	80.5	73.7	66.9	83.9
CNN	80.8	92.6	90.5	92.5	68.9	86.3	64.3	69.4	86.2	82.6	90.8	80.7	82.1	70.9	64.1	90.6
$RNN + CNN$	83.3	94.0	92.3	94.1	72.4	84.6	**68.4**	74.0	86.7	86.1	92.1	83.7	**85.7**	75.0	69.5	91.0
$mBERT$	83.4	92.1	91.2	**93.5**	74.7	89.5	67.6	76.8	89.4	86.6	91.4	86.5	83.5	71.8	67.3	89.5
$BERTbek$	**85.2**	**94.1**	**93.0**	93.2	**74.9**	**91.5**	67.1	**78.7**	**90.0**	**88.2**	**93.4**	**88.2**	85.6	**75.8**	**71.7**	**93.3**

Table 2. Text classification evaluation results for all models. F1 scores per model and category and their mean values are reported, best scores overall and for each category are highlighted.

when both the word level and character level n-grams are considered (by concatenating their TF-IDF matrices).

Neural network models, such as RNN and CNN, perform better than rule-based models, and their performance is of 85.2%, compared to its multilingual counterpart (with 83.4% F1-score).

The results of our experiments demonstrate the effectiveness of deep learning models for text classification in the Uzbek language and provide a strong foundation for further research in this area.

6. Discussion

Analysing the performance results of the models over the newly obtained dataset, one can say that the text distribution of the news data over categories plays an important role, as the categories with significantly more data (such as *Local*, *World*, *Law*, etc.) achieve higher performance results, overall evaluation models, compared to other categories. The counter-wise situation is also true since some categories with very small amounts of data (such as *Women*, *Photo*, *Culture*, etc.) perform less overall.

Some categories with distinct keywords that are only used in their own field, such as *Sport* (most common keywords: sports names, and names of teams and players), *Auto* (most common keywords: car brands), as well as *Culinary* (most common keywords: names of ingredients, cooking terms), that can be easily predicted also reflect in the overall models‘ performance, showing high scores for those categories. Although the category *Tech* can be easily predicted like the previously-mentioned categories, it achieves the lowest performance scores in our case, due to the fact that the news data in that category look like other categories like *Auto* and *Photo*, making it hard for the models to predict the labels right.

Lastly, it can also be observed that the monolingual $BERTbek$ model outperforms the multilingual $mBERT$ model in many cases, due to the fact that the multilingual model includes a very small portion of texts in Uzbek. Only in the cases of predicting the labels for the *Tech* and *Sport* categories, $mBERT$ outperforms the $BERTbek$, which is caused by the fact that most of the key terms used in those texts are either named entities or international terms.

enhanced by adding specific knowledge of the language, such as pretrained word-embedding vectors. Among the transformer-based models, the monolingual $BERTbek$ model achieved the highest performance with an F1-score

7. Conclusion and Future Work

In this paper, we aimed to tackle the task of text classification for the low-resource Uzbek language. Our contribution to the field includes a new dataset consisting of more than 512K labelled news texts with more than 120M words, spanned over 15 categories collected from 10 different news and press websites. The dataset was pre-processed to remove unwanted text, such as duplicates, references to images, emojis, and URLs, and transliterated from Cyrillic to Latin. In our experiments, we compared the performance of various models including rule-based models, deep learning models, as well as multilingual and monolingual transformer-based language models.

Our evaluation results showed that the BERT-based models outperform other models, while the monolingual BERT-based model achieved the highest score.

In conclusion, we have shown that deep learning models can effectively handle text classification tasks for the Uzbek language. In future work, we plan to improve the performance of the models by fine-tuning them on a larger dataset, and also to extend the study to other NLP tasks such as sentiment analysis, named entity recognition, and machine translation. Furthermore, we aim to develop open-source tools to make Uzbek NLP resources easily accessible to researchers and practitioners in the field.

Data availability

The newly created Uzbek text classification dataset and the Python codes used for the evaluation of the models are publicly available at the project repository[5] as well as an open-access data platform[6].

This dataset will serve as a valuable resource for further NLP research on Uzbek language, and we hope it will stimulate further work in this area. By making the data and codes openly accessible, we aim to foster reproducibility and collaboration in the field.

[5] https://github.com/elmurod1202/TextClassification

[6] https://doi.org/10.5281/zenodo.7677430

Acknowledgements

This research work was fully funded by the REP-25112021/113 - "UzUDT: Universal Dependencies Treebank and parser for natural language processing on the Uzbek language" subproject funded by The World Bank project "Modernizing Uzbekistan national innovation system" under the Ministry of Innovative Development of Uzbekistan.

Declarations

The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analysis, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

Cruz, J. C. B., & Cheng, C. (2020). Establishing baselines for text classification in low-resource languages. *ArXiv Preprint ArXiv:2005.02068*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv Preprint ArXiv:1810.04805*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Haruechaiyasak, C., Jitkrittum, W., Sangkeettrakarn, C., & Damrongrat, C. (2008). Implementing news article category browsing based on text categorization technique. *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, *3*, 143–146.

Jindal, N., & Liu, B. (2007). Review spam detection. *Proceedings of the 16th International Conference on World Wide Web*, 1189–1190.

Joachims, T., & others. (1999). Transductive inference for text classification using support vector machines. *Icml*, *99*, 200–209.

Kim, J., & Lee, M. (2014). Robust lane detection based on convolutional neural network and random sample consensus. *Neural Information Processing: 21st International Conference, ICONIP 2014, Kuching, Malaysia, November 3-6, 2014. Proceedings, Part I 21*, 454–461.

Kuriyozov, E., Doval, Y., & Gomez-Rodriguez, C. (2020). Cross-Lingual Word Embeddings for Turkic Languages. *Proceedings of The 12th Language Resources and Evaluation Conference*, 4054–4062.

Kuriyozov, E., Matlatipov, S., Alonso, M. A., & Gómez-Rodr\'\iguez, C. (2022). Construction and evaluation of sentiment datasets for low-resource languages: The case of Uzbek. *Human Language Technology. Challenges for Computer Science and Linguistics: 9th Language and Technology Conference, LTC 2019, Poznan, Poland, May 17–19, 2019, Revised Selected Papers*, 232–243.

Liu, P., Qiu, X., & Huang, X. (2016). Recurrent neural network for text classification with multi-task learning. *ArXiv Preprint ArXiv:1605.05101*.

Madatov, K., Bekchanov, S., & Vičič, J. (2022). *Automatic detection of stop words for texts in the Uzbek language*.

Matlatipov, S., Rahimboeva, H., Rajabov, J., & Kuriyozov, E. (2022). Uzbek Sentiment Analysis Based on Local Restaurant Reviews. *CEUR Workshop Proceedings*, *3315*, 126–136. www.scopus.com

McCallum, A., Nigam, K., & others. (1998). A comparison of event models for naive bayes text classification. *AAAI-98 Workshop on Learning for Text Categorization*, *752*(1), 41–48.

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, *5*(4), 1093–1113.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning–based text classification: a comprehensive review. *ACM Computing Surveys (CSUR)*, *54*(3), 1–40.

Rabbimov, I. M., & Kobilov, S. S. (2020). Multi-class text classification of uzbek news articles using machine learning. *Journal of Physics: Conference Series*, *1546*(1), 12097.

Salaev, U., Kuriyozov, E., & Gómez-Rodríguez, C. (2022a). A Machine Transliteration Tool Between Uzbek Alphabets. *CEUR Workshop Proceedings*, *3315*.

Salaev, U., Kuriyozov, E., & Gómez-Rodríguez, C. (2022b). SimRelUz: Similarity and Relatedness scores as a Semantic Evaluation Dataset for Uzbek Language. *1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages, SIGUL 2022 - Held in Conjunction with the International Conference on Language Resources and Evaluation, LREC 2022 - Proceedings*.

Sharipov, M., Kuriyozov, E., Yuldashev, O., & Sobirov, O. (2023). UzbekTagger: The rule-based POS tagger for Uzbek language. *ArXiv Preprint ArXiv:2301.12711*.

Sharipov, M., & Yuldashov, O. (2022). UzbekStemmer: Development of a Rule-Based Stemming Algorithm for Uzbek Language. *ArXiv Preprint ArXiv:2210.16011*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Simulating Domain Changes in Task-oriented Dialogues

Tiziano Labruna[1], Bernardo Magnini[2]

[1]Fondazione Bruno Kessler, Trento, Italy and Free University of Bozen-Bolzano, Italy
tlabruna@fbk.eu
[2]Fondazione Bruno Kessler, Trento, Italy
magnini@fbk.eu

Abstract

While task-oriented dialogue systems are widespread for several application scenarios (e.g., booking a restaurant, executing commands, taking an appointment), a major issue concerns their capacity to adapt to different conversational situations. Particularly, when the domain knowledge of the system changes (e.g., new restaurants open), current data-driven dialogue models are not robust enough to capture those changes, and performance tends to decrease significantly. In this paper we investigate domain changes (a common situation) through a dialogue simulation environment, allowing us to predict the performance of a dialogue model when the conversational domain changes. A key aspect of the simulator is that test data used for performance evaluation are automatically produced through a dialogue adaptation process. We provide a number of experiments based on the MultiWOZ shared dataset, showing how the dialogue simulation environment can be practically used in concrete situations.

Keywords: Conversational Agents, Domain Adaptation, Evaluation

1. Introduction

Task-oriented dialogue systems (McTear, 2020; Young et al., 2013), (Henderson et al., 2014) allow users to achieve specific tasks (e.g., booking a restaurant, buying a train ticket, ordering some food) through dialogues in natural language. While in recent years there has been a large diffusion of such conversational systems, a major bottleneck for their development is that conversational domains are very dynamic and are subject to continuous changes, which soon make initial dialogue models inadequate to manage new situations. As an example, a chatbot for giving information about Covid-19 needs to be frequently updated, as new regulations are introduced and others are changed. A similar issue happens in the case of booking restaurants in a region, where new restaurants open and others introduce new food. In such situations initial dialogue models (e.g., intent and slot-filling) soon become obsolete and the system performance rapidly decreases.

The current practice in case of domain changes consists of manually updating the training dialogues, typically adding human-annotated sentences with new intents and entities that reflect the changes. However, this practice is extremely expensive and requires specialized competencies. In addition, there are no tools for simulating the impact that a certain domain change might have on the performance of the dialogue system and its components. Being able to approximate the impact of, for instance, adding or removing a certain slot in the system knowledge base would allow a more precise estimation of the re-training costs, with a significant saving of time and money. Although dialogue simulators have been proposed (e.g., Simdial (Zhao and Eskenazi, 2018)), to the best of our knowledge, none of them are designed to simulate domain changes.

In this paper, we propose an innovative methodology to simulate the performance of a task-oriented dialogue system when domain changes occur. The core question the simulator allows to answer is the following: if the domain knowledge of the system changes (e.g., a certain amount of slot-values or instances are added or removed), how a dialogue model trained before such changes would perform in the new situation? Providing an answer to such kind of questions is crucial for developers because it permits them to estimate in advance the behaviour of the system and, in turn, the cost of updating the training data and the dialogue models. We provide the simulator with the capacity to manage fine-grained changes, for instance modifying a single slot value (e.g., change FOOD=MEDITERRANEAN with FOOD=POKE), and predicting performance exactly for those changes.

We consider a typical dialogue architecture based on three components: natural language understanding (NLU), performing intent detection and slot filling of the user utterance, dialogue manager (DM), which, based on the content of the knowledge base (KB), indicates the action for the system response, and a natural language generation (NLG) component, whose task is to produce a sentence in natural language.

The major research challenge that needs to be considered when designing a dialogue simulator able to account for domain changes concerns performance evaluation. More specifically, in order to evaluate a certain change (e.g., a new type of food for a restaurant is introduced, which was not present before), we need a gold standard (i.e., test dialogues) reflecting the changes we intend to simulate. In this paper, we suggest that recent dialogue adaptation techniques (Labruna and Magnini, 2021; Labruna and Magnini, 2022) can be applied for the automatic creation of test dialogues to be used in a dialogue simulator. We also suggest that the generative power of recent pre-trained language models may offer encouraging opportunities in the direction of automatic dialogue adaptation.

2. Background on Task-oriented Dialogues

Figure 1 depicts a general architecture of a data-driven conversational agent, showing three main components: Natural Language Understanding (NLU), Dialogue Manager (DM), and Natural Language Generation (NLG). The user sends the message to the agent, the NLU

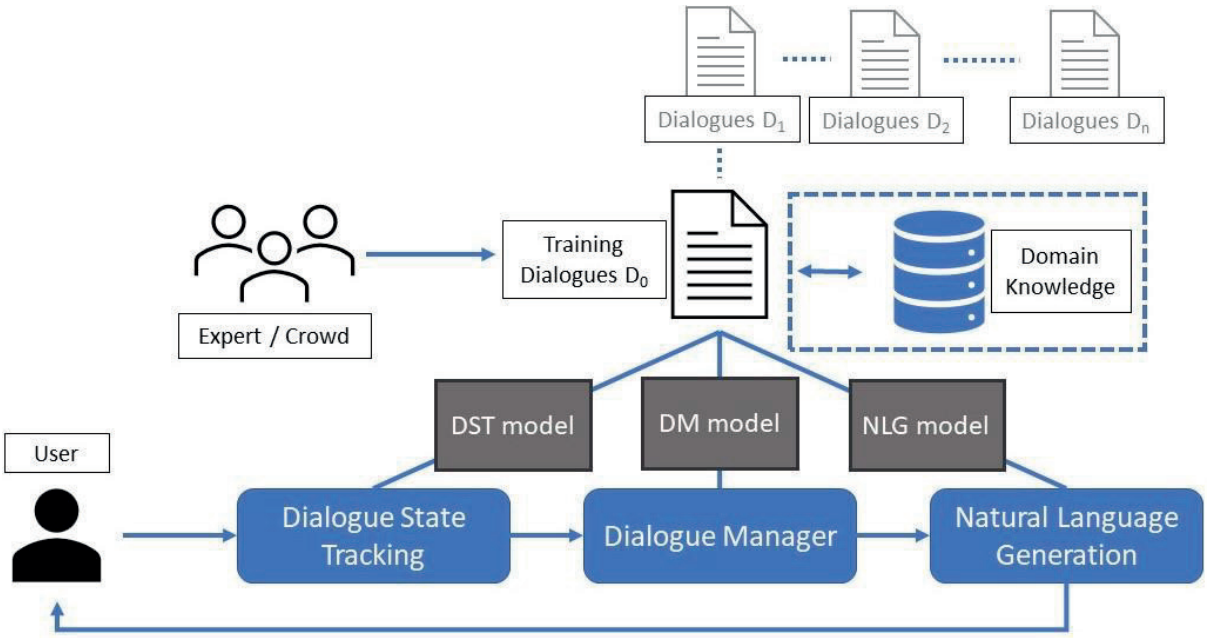


Fig. 1: representation of a typical data-driven conversational system flow. The user sends a message, which is parsed by a dialogue state tracking component; the output is passed to a dialogue manager component, which decides the best next action of the system; finally, a natural language generation component generates the utterance to be returned to the user. Each component is based on a model trained on annotated dialogues, which are typically manually created and then annotated through links to a knowledge base. Training dialogues will vary when domain changes occur (D_1 to D_n).

component is responsible for extracting relevant information from the message and passing it to the DM component, which, based on that information, decides which action to take; finally, the NLG component takes the action as input and returns a natural language message to be sent back to the user.

Domain Knowledge Base. According to most of the recent literature (Budzianowski et al., 2018; Bordes et al., 2017; Mrkˇsi´c et al., 2017), we consider a task-oriented dialogue between a system and a user as composed of a sequence of turns {*t_1, t_2,...t_n*}. The goal of the dialogue system is to retrieve a set of entities (possibly empty) in a domain knowledge base (*KB*) that satisfy the user's needs. A domain ontology *O* provides a schema for the *KB* and typically represents entities (e.g., RESTAURANT, HOTEL, MOVIE) according to a pre-defined set of slots *S* (e.g., FOOD, AREA, PRICE, for the RESTAURANT domain), and values that a certain slot can assume (e.g., EXPENSIVE, MODERATE and CHEAP, for the slot PRICE).

On the basis of the entities defined in the domain ontology, the *KB* is then populated with instances of such entities. As in most of the literature, we distinguish *informable slots*, which the user can use to constraint the search (e.g., AREA), and *requestable slots* (e.g., PHONENUMBER), whose values are typically asked only when a certain entity has been retrieved through the dialogue.

At each turn in the dialogue, both the user and the system may refer to facts in the *KB*, the user with the goal of retrieving entities matching his/her needs, and the system to propose entities that can help the user to achieve the dialogue goals.

Natural Language Understanding. The goal of the NLU component (Louvan and Magnini, 2020) is to extract relevant information from the user message. Such information typically consists of an *intent* (the communicative goal of the user's utterance) and a certain number of *entities* that are contained in the message. The prediction of the former is known as Intent Recognition, while the prediction of the latter is called Entity Extraction or Slot Filling. The prediction of intents and entities is usually evaluated in terms of F1-score of the system when confronted with human annotation on dialogues.

Dialogue Manager. The DM component takes an intent and a certain number of entities (possibly empty) as input and returns the best next action to take, which typically consists of an intent and slot-value pairs. While taking this decision, the dialogue manager also considers a number of dialogue state variables, such as the conversation history up to a certain point in the past. The selection of the best action is usually evaluated in terms of accuracy.

Natural Language Generation. As the last component of process, NLG is responsible for converting the output of the DM into a natural language sentence. The NLG component needs to take a structured representation of information in input and produce a natural language utterance that will be returned to the user. The correct generation of the utterance is typically evaluated using string comparison metrics (a common one is BLEU (Papineni et al., 2002)).

3. A Dialogue Simulator

We propose a methodology to investigate the impact of domain changes in dialogue systems based on a *Domain Changes Simulator* (DCS), an architecture that simulates different types and different amounts of domain changes, chooses a model for every dialogue component, and produces a report on the performances of the models given a certain configuration of the simulator. We implemented the DCS simulator on top of RASA (Bocklisch et al., 2017), an open-source platform developed to facilitate the production of task-oriented conversational agents. RASA adopts the architecture shown in Figure 1 and allows to use custom developed dialogue models for each of the three components. As a default, Rasa does not use a domain knowledge base, which needs to be added for every specific need.

The use of the DCS simulator includes the following three steps, which are detailed in the rest of the section: (i) define an initial domain *KB*, notated with *KB_0*, and then a modified *KB*, notated with *KB_1*; (ii) run the dialogue components over training dialogues *D_0* (consistent with the initial *KB_0*); (iii) estimate component performance on *D_1* (consistent with the modified *KB_1*).

3.1 Define KB Changes

Both the initial *KB_0* and the modified *KB_1* are supposed to be consistent with the domain ontology *O* (see section 2). While *KB_0* can be uploaded (typically a JSON file), *KB_1* needs to be defined through changes to be applied to *KB_0*. The DCS simulator provides an intuitive way to define the domain changes through a graphical interface, shown in Figure 2. In the current version of the simulator we allow two kinds of changes:

Slot-value changes. Slot-values are used to describe properties of instances (e.g., the MARIO'S restaurant offers ITALIAN food). Through the DCS graphical interface, it is possible either to add new slot-values in *KB_1* (e.g., CARIBBEAN food starts to be served by some restaurants), or to remove existing slot-values from *KB_1* (e.g., no more restaurants that serve INDIAN food). In both cases, the developer of the conversational system can specify the amount of the change. For instance, assuming that five new slot-values are added in *KB_1*, if the developer selects 20%, it means that 20% of the slot-value occurrences (randomly selected) in *KB_0* are replaced with occurrences of the new slot-values, equally split for the five new values.

Instance changes. Instances are individual entities in the conversational domain (e.g., a specific restaurant). Through the DCS graphical interface it is possible either to add new instances in *KB_1* (e.g., a new restaurant opens), or to remove existing instances from *KB_1* (e.g., all restaurants that serve INDIAN food close down). As instances are described by means of slot-value pairs, adding or removing instances affects the distribution of the slot-values. Finally, as with slot-values, the amount of added or removed instances can be defined by the developer.

3.2 Run dialogue models

In addition to domain changes, the DCS simulator manages different models for the dialogue components described in Section 2. In this paper we focus on NLU and DM components.

Fig. 2: Graphical interface of the dialogue simulator.

The NLU component requires an annotated collection of user utterances in natural language, in order to be able to recognize and extract intents and entities from a message. In this paper, we use the training data of the MultiWOZ 2.3 dataset (Budzianowski et al., 2018), although there are several available datasets annotated for the slot-filling task (see (Louvan and Magnini, 2020) for a detailed list). The NLU simulator is trained on existing annotated dialogues (*D_0* dialogues) and the performance of the model is tested on dialogues reflecting the changes that occurred in *KB_1*.

The DM component predicts the next action of the system during the dialogue. We implemented a simple DM that receives the output of NLU, composes a query to the current *KB*, and on the base of the retrieved entities, takes a decision for the next system's action.

3.3 Estimate Component Performance

The main purpose of the DCS simulator is to predict how the performances of the dialogue components evolve when certain domain changes occur. A crucial issue here is to develop test data for each component and for each configuration of domain change we are interested to evaluate. Test data vary according to the dialogue component: we need dialogue annotated with intents and slot-value pairs for NLU, and ground truth actions to be performed by the system at each dialogue turn for the DM. While in principle such test data should be collected through human intervention (e.g., Wizard of Oz), this is practically unfeasible given the high number of potential configurations we want to simulate.

To overcome this issue, the DCS simulator adopts a *dialogue adaptation* strategy for the automatic creation of the test data. The idea behind dialogue adaptation is that domain knowledge described in the *KB* is somehow reflected in training and test dialogues, and that, when a domain change occurs, it is possible to adapt the initial dialogues so that the domain change is adequately reflected.

More formally, we define the problem of Dialogue Adaptation as follows: starting from a dialogue *D_0* collected for a certain knowledge base *KB_0*, the goal is to modify *D_0* such that it reflects a knowledge base *KB_1*, where *KB_0* and *KB_1* share the same domain ontology *O* (i.e., they share domain entities and slots).

Domain Change	KB size	NLU			DM
		Precision	Recall	F1-score	Accuracy
No change	KB_0	0.70	0.94	0.80	0.96
Add 50% slot-values	KB_1 same size as KB_0	**0.58**	**0.73**	**0.64**	0.15
Remove 50% slot-values	KB_1 same size as KB_0	0.59	0.75	0.65	**0.08**
Add 50% instances	KB_1 50% bigger than KB_0	0.60	0.76	0.67	0.70
Remove 50% instances	KB_1 50% smaller than KB_0	0.60	0.77	0.66	0.23

Table 1: Results show the impact of different domain changes on both NLU and DM components. Experiments evaluate the performance of the same models trained on MultiWOZ, over different domain change scenarios. The values refer to informable slots only. Values in bold indicate the change that caused the biggest impact for each category.

We have adopted a dialogue adaptation strategy that slightly revises the method proposed in (Labruna and Magnini, 2022). We first fine-tune a pre-trained language model (we use BERT (Devlin et al., 2019)) on *KB_1*. This is done by extracting textual patterns from the *KB* and further training the model on this domain-specific data.

Once BERT has been fine-tuned, we use the resulting model (i.e., *BERT_{KB_1}*) to predict slot-values to be substituted in *D_0* test dialogues, thus producing *D_1*. For instance, the following user utterance from *D_0* dialogues:

> "I'm looking for an **Indian** restaurant in the **north** part of town."

will first be masked as follows:

> "I'm looking for an **[MASK]** restaurant in the **[MASK]** part of town."

and, finally, we will ask *BERT_{KB_1}* to predict the substitutions to the masks, which will produce something like:

> "I'm looking for an **English** restaurant in the **north-wes** part of town."

which will contribute to populating the *D_1* dataset. Note that the substitutions are produced sequentially from left to right, therefore the first prediction will condition the subsequent ones.

4. Experiments and Results

This section illustrates a practical use of the DCS simulator. The goal of the experiments is to compare the performance of both NLU and DM when trained on initial *KB_0* and *D_0*, and then tested on modified *KB_1* and adapted dialogues *D_1*. Intuitively, we expect some degradation in the performance, particularly when the domain changes are relatively consistent (e.g., changing 50% of the slot-values).

4.1 Experimental Setting

Dataset. For all the experiments we use the MultiWOZ 2.3 dataset, which has been adapted in order to match the data format required by the RASA platform. MultiWOZ has been collected through the Wizard of OZ technique and it contains a total of more than ten thousand dialogues, each with an average of around 13 turns, spanning over 7 domains. The context of the dialogues relates to a user asking for information about activities to do in Cambridge and the system provides responses following the setting of a task-oriented dialogue system.

Domain changes. We run experiments on five domain changes scenarios:

- No change setting. This is the initial situation with *KB_0*. We use the MultiWOZ dataset and, for the sake of simplicity, we considered only the Restaurant concept, which is the most representative of all seven concepts, since it is based on a *KB* with more than 100 instances, each one with 10 slots, including both requestable and informable ones.
- Add 50% of new slot-values. The slot-values to be added have been manually identified. For instance, we have added CARIBBEAN as a new value for the slot FOOD. Notice that in this setting *KB_1* has the same number of instances as *KB_0*.
- Remove 50% of existing slot-values in *KB_0*. The slot values to be removed are randomly selected, till they reach 50% of the slot-values present in *KB_0*. Also in this setting, *KB_1* has the same number of instances as *KB_0*.
- Add 50% of new instances in *KB_0*. The new instances have been manually identified. Here *KB_1* has 50% more instances than *KB_0*.
- Remove 50% of existing instances in *KB_0*. The instances to be removed are randomly selected, till they reach 50% of the instances present in *KB_0*. Here *KB_1* has 50% fewer instances than *KB_0*.

Dialogue models. As for NLU, we used DIET (Bunk

et al., 2020), a model integrated within the pipeline of RASA. DIET is based on a multi-task transformer architecture for performing both Intent Classification and Entity Recognition. It leverages the knowledge of well-known pre-trained models like BERT, GloVe, and ConveRT, with the advantage of being more modular and faster to train, rather than using the pre-trained models directly.

For the experiments reported in this paper, we consider a rule-based DM based on *KB_0*, which takes as input the intents and slot-values predicted by the NLU component, and, based on that, makes a query on the *KB* and returns the slot-value pairs that will be used by the NLG component to generate the final response to the user.

Evaluation. Performance is estimated on two components, NLU and DM. For NLU we compare the performance on slot-filling (F1 score calculated by RASA) of the model trained on *D_0* and tested on adapted *D_1* dialogues (see section 3.3 for dialogue adaptation). The intuition is that the NLU model is likely to be affected mainly by slot-value changes, rather than by the number of instances.

As for DM, we compare the performance on the next action prediction of the model aligned with *KB_0* with the model aligned with *KB_1*. We consider a next action prediction as correct if it is consistent with *KB_1*. We consider three cases of consistency (i.e., correct next action prediction): (i) when both the DM and the *KB_1* agree that there are no entities satisfying the user request (this is the case of failure); (ii) when both the DM and the *KB_1* agree that there is only one instance satisfying the user request, and the instance is the same; (iii) when both the DM and the *KB_1* agree that there is more than one instance satisfying the user request. The intuition behind the DM evaluation is that DM is likely to be affected by instance domain changes, rather than slot-value changes.

4.2 Results and Discussion

Table 1 shows the results of the experiments on both NLU and DM.

As for NLU in the initial situation (first line), the model is trained on the original MultiWOZ training-set and tested over the corresponding MultiWOZ test-set, obtaining a F1-score of 0.8. With the four different domain changes the NLU performance decreased significantly, spanning from a loss of 13 points for the addition of 50% new instances to a loss of 16 points for the addition of 50% new slot-values. We also note that the performances of the experiments with domain changes are similar to each other. This is due to the fact that adding or removing slot-values and instances does not affect significantly the NLU performance. This is somehow surprising, as it would have been expected that detecting new slot-values (add 50% of new slot-values) could be a more difficult task than managing a different distribution of the same values (remove 50% of new slot-values). On the other side, it was expected that the NLU model is not much sensitive to the size of the *KB*, as it is revealed by the low difference between the experiments on slot-values (*KB_0* same size as *KB_1*) and on instances (*KB_0* has different size with respect of *KB_1*).

As for DM, we used an average of the accuracy as the evaluation metric, which, for each turn in the test-set, is equal to 1 if the prediction of the next action given by the DM model is compatible with *KB_1* (e.g., if the system believes that there are ten restaurants that satisfy the user's request and, in fact, the same number of instances is present in *KB_1*), 0 otherwise. In this case, the accuracy obtained for the no-change setting is 0.96, and we observe a critical degradation in the model's performance, ranging from 26 points of loss in the case of the addition of the instances to a loss of even 88 points for the reduction of the slot-values.

This is probably due to the fact that in the instance addition setting the initial instances remain unchanged in *KB_1*, as the addition of new instances that leaves untouched the ones that were already there. On the contrary, for the slot-value reduction setting, we are taking out half of the slot-values and increasing the occurrences for the ones that remain, thus causing the production of wrong answers for almost all users' requests.

To sum it up, the experiments that we presented here show a strong degradation in the model's performance for both NLU and DM components when different types of changes are introduced, thus highlighting the necessity of retraining the models on updated datasets. Different quality and quantity of changes bring different impacts and therefore the DCS simulator that we are proposing can be extremely helpful for investigating the degradation trend and understanding when a certain model is outdated for the changed domain.

5. Conclusions

We have presented a dialogue simulator focused on estimating the impact of domain changes on two crucial components of a dialogue system, NLU and DM. We made experiments on the usage of the simulator on the MultiWOZ 2.3 dataset, whose Knowledge base has been modified in two respects: slot-values addition and deletion, and instances addition and deletion. Results provide quantitative evidence that both the NLU and the DM models are significantly affected by those domain changes.

The experience reported in the paper reinforces the view that a dialogue simulator allowing to manipulate a number of parameters for estimating the impact of domain changes, can be an important tool for making it easier and less expensive to develop and maintain a conversational system.

A major research challenge for a domain change simulator is the capacity to automatically generate test dialogues that approximate the domain changes. This capacity is crucial for evaluation purposes, and can be achieved through dialogue adaptation techniques, i.e., through incremental substitutions in the initial training dialogues, exploiting the generative power (e.g., masked tokens, prompting) of pre-trained language models.

6. References

Bocklisch, Tom, Joey Faulkner, Nick Pawlowski, and Alan Nichol, 2017. Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181*.

Bordes, Antoine, Y-Lan Boureau, and Jason Weston, 2017. Learning end-to-end goal-oriented dialog. In*ICLR. OpenReview.net.*

Budzianowski, Paweł, Tsung-Hsien Wen, Bo-Hsiang Tseng, Ĩnigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gaˇsi´c, 2018. MultiWOZ - a large-scale multi-domain wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.* Brussels, Belgium: Association for Computational Linguistics.

Bunk, Tanja, Daksh Varshneya, Vladimir Vlasov, and Alan Nichol, 2020. Diet: Lightweight language understanding for dialogue systems. *arXiv preprint arXiv:2004.09936.*

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers).* Minneapolis, Minnesota: Association for Computational Linguistics.

Henderson, Matthew, Blaise Thomson, and Jason D.Williams, 2014. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL).* Philadelphia, PA, U.S.A.: Association for Computational Linguistics.

Labruna, Tiziano and Bernardo Magnini, 2021. Addressing slot-value changes in task-oriented dialogue systems through dialogue domain adaptation. In *Proceedings of RANLP 2021.*

Labruna, Tiziano and Bernardo Magnini, 2022. Finetuning Bert for generative dialogue domain adaptation.In *Text, Speech, and Dialogue 2022.*

Louvan, Samuel and Bernardo Magnini, 2020. Recentneural methods on slot filling and intent classification for task-oriented dialogue systems: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics. Barcelona, Spain (Online): International Committee on Computational Linguistics.*

McTear, Michael, 2020. Conversational ai: Dialogue systems, conversational agents, and chatbots. *Synthesis Lectures on Human Language Technologies, 13(3): pp 1–251.*

Mrkˇsi´c, Nikola, Diarmuid ´O S´eaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young, 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Ass Papers).* Association for Computational Linguistics.

Papineni, Kishore, Salim Roukos, Todd Ward, and WeiJing Zhu, 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*.

Young, S., M. Gaˇsi´c, B. Thomson, and J. D. Williams, 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE, 101(5):1160–1179.*

Zhao, Tiancheng and Maxine Eskenazi, 2018. Zero-shot dialog generation with cross-domain latent actions. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue.* Melbourne, Australia: Association for Computational Linguistics.

Language-Independent Sentiment Labelling with Distant Supervision: A Case Study for English, Sepedi and Setswana

Koena Ronny Mabokela*, Tim Schlippe$, Mpho Raborife*, Turgay Celik‖

*University of Johannesburg, South Africa
$IU International University of Applied Sciences, Germany
‖University of the Witwatersrand, South Africa
krmabokela@gmail.com

Abstract

Sentiment analysis is a helpful task to automatically analyse opinions and emotions on various topics in areas such as *AI for Social Good*, *AI in Education* or marketing. While many of the sentiment analysis systems are developed for English, many African languages are classified as low-resource languages due to the lack of digital language resources like text labelled with corresponding sentiment classes. One reason for that is that manually labelling text data is time-consuming and expensive. Consequently, automatic and rapid processes are needed to reduce the manual effort as much as possible making the labelling process as efficient as possible. In this paper, we present and analyze an automatic language-independent sentiment labelling method that leverages information from sentiment-bearing emojis and words. Our experiments are conducted with tweets in the languages English, Sepedi and Setswana from *SAfriSenti*, a multilingual sentiment corpus for South African languages. We show that our sentiment labelling approach is able to label the English tweets with an accuracy of 66%, the Sepedi tweets with 69%, and the Setswana tweets with 63%, so that on average only 34% of the automatically generated labels remain to be corrected.

1. Introduction

Sentiment analysis helps analyze and extract information about polarity from textual feedback and opinions. Sentiment analysis draws attention in business environments (Rokade and Kumari, 2019) and other areas, like medicine (Zucco et al., 2018), education (Mabokela et al., 2022; Rakhmanov and Schlippe, 2022) and AI for Social Good (Mabokela and Schlippe, 2022b).

Sentiment analysis for under-resourced language still is a skewed research area. Although, there are some considerable efforts in emerging African countries to develop resources for under-resourced languages, some languages such as indigenous South African languages still suffer from a lack of datasets. One reason for that is that manually labelling text data is time-consuming and expensive. Consequently, automatic and rapid processes are needed to reduce the manual effort as much as possible making the labelling process as efficient as possible. In this paper, we present and analyze an automatic language-independent sentiment labelling algorithm that leverages information from sentiment-bearing emojis[1] and words. We will evaluate our algorithm on a subset of our *SAfriSenti* corpus (Mabokela and Schlippe, 2022a; Mabokela and Schlippe, 2022b) with English, Sepedi and Setswana tweets. Sepedi is mainly spoken in the northern parts of South Africa by 4.7 million people and Setswana by 4.5 million people (Statista, 2022).

In the next section, we will describe related work. In section 3 we will present our language-independent algorithm for sentiment labelling. The experimental setup will be characterised in Section 4. In Section 5 we will summarise the results of our experiments. We will conclude our work in Section 6 and indicate possible future work.

[1]Emojis are pictorial representations of emotions, ideas, or objects in electronic communication to add emotional context.

2. Related Work

Previous studies investigated sentiment data collection strategies for under-resourced languages on Twitter (Pak and Paroubek, 2010; Vosoughi et al., 2016). The methods focus on labelling only two sentiment classes —positive and negative. Meanwhile other research work has explored strategies to label three sentiment classes in Twitter—positive, neutral, and negative —using human annotators (Vilares et al., 2016; Pak and Paroubek, 2010; Pang et al., 2002; Nakov et al., 2019). Despite the attempt to automate the data labelling process (Kranjc et al., 2015), the hand-crafted annotation is to date the most preferred method of data labelling in many natural language processing tasks (Chakravarthi et al., 2020). However, manual annotation presents challenges and it is deemed an expensive process. Notably, (Jamatia et al., 2020; Gupta et al., 2021) employed manually annotated tweets, while other studies focus on automated data labelling solutions (Kranjc et al., 2015). (Vosoughi et al., 2016) investigated various pipelines to collect data on Twitter using distant supervised learning. In this approach, they use positive and negative emojis as indicators to annotate tweets.

(Go et al., 2009) explored distant supervision methods to label millions of tweets using positive and negative search terms (i.e. term queries) in the Twitter API and emojis to pre-classify the tweets. (Vilares et al., 2016) investigated *SentiStrength* scores to label an English-Spanish code-switching Twitter corpus. *SentiStrength* is an online sentiment analysis system available for a few languages (Thelwall et al., 2011).

Compared to (Vilares et al., 2016; Cliche, 2017; Jamatia et al., 2020), we also investigate a distant supervised annotation method. However, we automatically build up lists with sentiment-bearing words after we have made use of emojis as indicators for the sentiment classes. This way we can label all tweets—first those tweets that contain

sentiment-bearing emojis, and then the rest of the tweets based on the words in the tweets with the sentiment-bearing emojis.

3. Sentiment Labelling with Distant Supervision

As illustrated in Figure 1, 2 and 3, we propose the following algorithm for sentiment labelling that leverages information from sentiment-bearing emojis and words:

- *Step* 1_{emojis}: Classify tweets with sentiment-bearing emojis into the classes *negative*, *neutral* and *positive* (Figure 1).
- *Step* 2_{lists}: Create lists with sentiment-bearing words (Figure 2):
  1. Collect all words from *negative*, *neutral* and *positive*.
  2. Then remove words that occur in one or both other lists.
- *Step* 3_{words}: Classify remaining tweets without sentiment-bearing emojis into the classes *negative*, *neutral* and *positive* based on the highest word coverage with the lists of sentiment-bearing words (Figure 3).

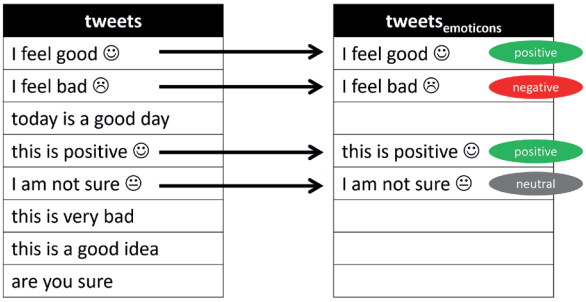


Figure 1: Classify tweets with sentiment-bearing emojis into the 3 classes ($step1_{emojis}$).

4. Experimental Setup

In this section, we will describe the dataset for our experiments and how we used emojis as indicators for sentiments.

4.1. SAfriSenti

To evaluate our algorithm for sentiment labelling with comparable numbers of tweets in three languages, we applied it to monolingual 7,000 English tweets, 7,000 Sepedi tweets and 7,000 Setswana tweets from the *SAfriSenti* corpus. *SAfriSenti* is to date the largest sentiment dataset available for South African languages with 64.3% of monolingual tweets in English, Sepedi and Setswana and 36.6% of code-switched tweets between these languages (Mabokela and Schlippe, 2022a). The monolingual tweets' distributions of the classes *negative*, *neutral*, *positive* are demonstrated in Table 1, 2 and 3.

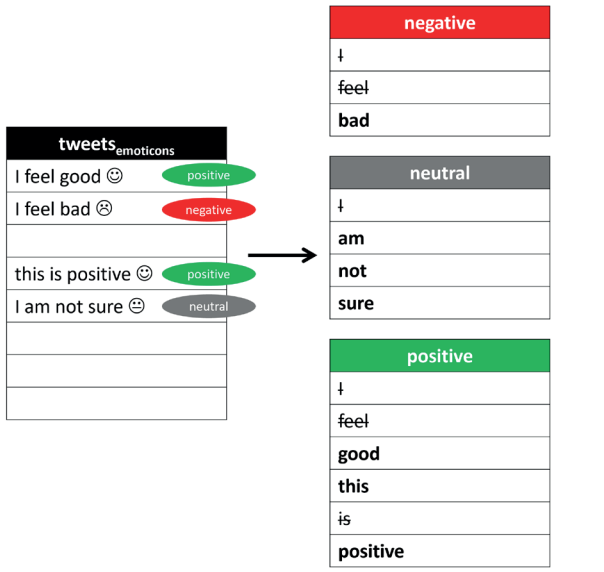


Figure 2: Create lists with sentiment-bearing words ($step2_{lists}$).

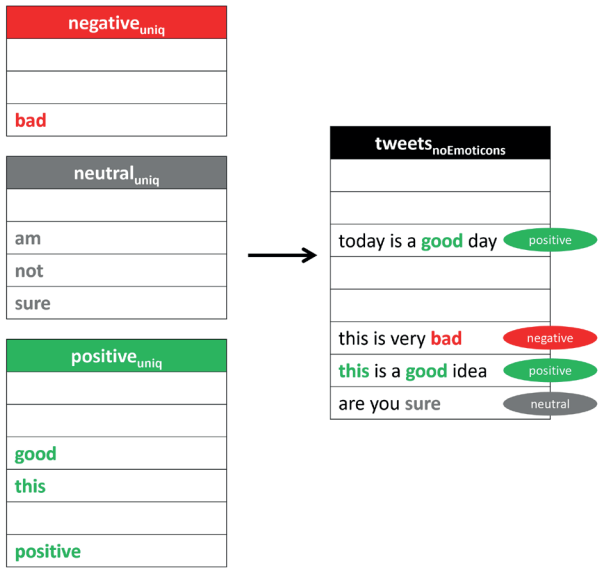


Figure 3: Sentiment-bearing words as indicators for remaining tweets' sentiment classes ($step3_{words}$).

Class	Number	%
positive	2,052	29.3
negative	3,448	49.3
neutral	1,500	21.4
Total	7,000	

Table 1: Distribution of English tweets.

Class	Number	%
positive	3,500	50.0
negative	2,270	32.4
neutral	1,230	17,6
Total	7,000	

Table 2: Distribution of Sepedi tweets.

Class	Number	%
positive	3,230	46.1
negative	2,180	31.1
neutral	1,590	22.8
Total	7,000	

Table 3: Distribution of Setswana tweets.

4.2. Emojis

For our experiments, we defined 12 emojis as *negative* indicators, 10 emojis as *neutral* indicators, and 12 emojis as *positive* indicators as listed in Table 4, for which were sure that they would represent the corresponding sentiments well. Of course, our emoji list can be extended based on further information such as the Emoji Sentiment Ranking[2] (Kralj Novak et al., 2015).

sentiment	#emojis
negative	12
neutral	10
positive	12

Table 4: Emojis for *negative*, *neutral* and *positive*.

A subset of such emojis is illustrated in Figure 4. To be platform-independent, our algorithm finds and compares the emojis in Unicode. If a tweet contains multiple emojis representing different sentiments, the tweet is labelled with the sentiment class which has the most emojis in the tweet. The tweet is not labelled at this step if there is no majority.

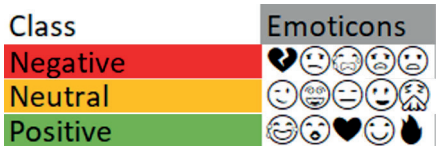


Figure 4: Examples of emojis

5. Experiments and Results

We evaluated our algorithm for sentiment labelling on the English, Sepedi and Setswana tweets from *SAfriSenti*. Table 5 demonstrates the absolute numbers and percentages of tweets with sentiment-bearing emojis in $step\ 1_{emojis}$ and the absolute numbers and percentages of the remaining tweets which were labelled using our sentiment-bearing words in $step\ 3_{words}$.

Table 6 and Table 7 show the accuracies and F-scores of automatically classifying tweets with sentiment-bearing emojis in $step\ 1_{emojis}$, the accuracies and F-scores of automatically classifying the remaining tweets using our sentiment-bearing words in $step\ 3_{words}$ as well as the accuracies and F-scores of all 7k labeled tweets together ($[step\ 1\text{-to-}step\ 3]$). For computing the F-scores, we took the *Macro* F_1 *Score Calculation*, i.e. the average of each class's F_1 score.

The results of the two tables demonstrate that the quality of the automatic labeling in $step\ 1_{emojis}$ is better than

[2] https://kt.ijs.si/data/Emoji_sentiment_ranking

language	$step\ 1_{emojis}$	$step\ 3_{words}$
English	4,210 (60.1%)	2,790 (39.9%)
Sepedi	5,871 (83.9%)	1,129 (16.1%)
Setswana	3,249 (46.4%)	3,751 (53.6%)

Table 5: #tweets and %tweets for $step1$ and $step3$.

language	$step\ 1_{emojis}$	$step\ 3_{words}$	$[step\ 1\text{-to-}step\ 3]$
English	68.7%	63.5%	66.2%
Sepedi	69.5%	64.6%	68.7%
Setswana	66.1%	59.7%	62.7%

Table 6: Accuracies.

language	$step\ 1_{emojis}$	$step\ 3_{words}$	$[step\ 1\text{-to-}step\ 3]$
English	66.8%	62.2%	64.6%
Sepedi	68.2%	63.4%	67.9%
Setswana	65.2%	58.7%	61.5%

Table 7: F-scores.

that in $step\ 3_{words}$. This shows that in $step\ 2_{lists}$ the automatically created word lists are worse indicators than the emojis. The goal of our algorithm is to label a large part correctly so that the annotators no longer have to label all labels from scratch and thus minimize the workload. From the accuracies and the F-scores we see that we were able to reduce the manual effort, but there is still room for improvement: With an accuracy of 66%, the annotators would still have to change 34% of the labelled tweets for English. Our algorithm performs best for Sepedi with 68% accuracy, which would require 32% changes. With Setswana, the accuracy is 63%, which is why 37% of the labels would have to be changed.

Both the quality of $step\ 1_{emojis}$ but also the quantity of tweets with sentiment-bearing emojis in the corpus, based on which the word lists are generated in $step\ 2_{lists}$, have an impact on the quality of the labeling in $step\ 3_{words}$: The more tweets with sentiment-bearing emojis, the higher the chance that qualified sentiment-bearing words remain in the lists after $step\ 2_{lists}$. We believe that Sepedi especially performs best since out of the 7k Sepedi tweets, 84% contain sentiment-bearing emojis and thus more text can be used for the generation of the word lists than for English (60%) and Setswana (47%). Still, with an accuracy of 70%, the challenge remains to figure out how to use emojis as a better indicator.

Adding more features for pre-labeling such as a translation of the tweets or word lists translated from other languages could help. But then the algorithm would no longer be completely language-independent and would have to deal with information coming from outside the corpus.

6. Conclusion and Future Work

In this paper we have presented a language-independent algorithm for sentiment labelling. Our algorithm uses sentiment-bearing emojis as initial features to build lists with sentiment-bearing words. Since our approach is only based on frequencies, no training of a machine learning system is required. This ways we completely avoid any manual or higher computational effort.

Our analyses on the under-resourced languages Sepedi and Setswana plus English demonstrated that accuracies between 63% and 69% are possible using our distant supervision approach. This significantly reduces the manual effort to label tweets with sentiment classes since the human annotators need to change between 31%–37% of the tweets that have been pre-labelled with our algorithm instead of adding all labels from scratch. After our proof of concept with three languages, it can be assumed that our approach works for other languages as well, since people often use emojis in their posts—no matter what language their posts are.

Consequently, it is interesting to apply our approach to more languages and to experiment with cross-lingual features. Future work may also include investigating whether it is helpful to label a tweet as *neutral* if it contains a comparable number of *positive* and *negative* emojis. Our current algorithm does not assign a label to the tweet in this case. Furthermore, we plan to combine our approach with active learning, i.e. an iterative process where annotators manually classify tweets which are then used to re-train machine learning systems for classification. Additionally, our goal is to create a multilingual natural language processing model and investigate the synergy effects across languages in sentiment analysis. (Makgatho et al., 2021)'s word embeddings could be a good basis for such a model.

7. References

Chakravarthi, Bharathi Raja, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John P. McCrae, 2020. Corpus creation for sentiment analysis in code-mixed tamil-english text. In *Workshop on Spoken Language Technologies for Under-resourced Languages*.

Cliche, Mathieu, 2017. BB_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs. In *11th International Workshop on Semantic Evaluation (SemEval-2017)*.

Go, Alec, Richa Bhayani, and Lei Huang, 2009. Twitter Sentiment Classification Using Distant Supervision. *Processing*, 150.

Gupta, Akshat, Sargam Menghani, Sai Krishna Rallabandi, and Alan W. Black, 2021. Unconscious Self-Training for Sentiment Analysis of Code-Linked Data. *ArXiv*, abs / 2103.14797.

Jamatia, Anupam, Steve Durairaj Swamy, Bjorn Gamback, Amitava Das, and Swapan Debbarma, 2020. Deep Learning Based Sentiment Analysis in a Code-Mixed English-Hindi and English-Bengali Social Media Corpus. *International Journal on Artificial Intelligence Tools*, 29.

Kralj Novak, Petra, Jasmina Smailović, Borut Sluban, and Igor Mozetič, 2015. Sentiment of Emojis. *PLoS ONE*, 10(12).

Kranjc, Janez, Jasmina Smailovic, Vid Podpecan, Miha Grcar, Martin Znidari, and Nada Lavrac, 2015. Active Learning for Sentiment Analysis on Data Streams: Methodology and Workflow Implementation in the ClowdFlows Platform. *Inf. Process. Manag.*, 51:187–203.

Mabokela, Koena Ronny, Turgay Celik, and Mpho Raborife, 2022. Multilingual sentiment analysis for under-resourced languages: A systematic review of the landscape. *IEEE Access*:1–22.

Mabokela, Koena Ronny and Tim Schlippe, 2022a. A Sentiment Corpus for South African Under-Resourced Languages in a Multilingual Context. In *The 1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages (SIGUL 2022)*.

Mabokela, Koena Ronny and Tim Schlippe, 2022b. AI for Social Good: Sentiment Analysis to Detect Social Challenges in South Africa. In *South African Conference for Artificial Intelligence Research (SACAIR 2022)*.

Makgatho, Mack, Vukosi Marivate, Tshephisho Sefara, and Valencia Wagner, 2021. Training Cross-Lingual embeddings for Setswana and Sepedi. *Journal of the Digital Humanities Association of Southern Africa (DHASA)*, 3(03).

Nakov, Preslav, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov, 2019. SemEval-2016 task 4: Sentiment analysis in Twitter. *arXiv preprint arXiv:1912.01973*.

Pak, Alexander and Patrick Paroubek, 2010. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In *LREC*.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan, 2002. Thumbs up? Sentiment classification using machine learning techniques. In *ACL-02 Conference on Empirical Methods in Natural Language Processing—Volume 10*.

Rakhmanov, Ochilbek and Tim Schlippe, 2022. Sentiment Analysis for Hausa: Classifying Students' Comments. In *The 1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages (SIGUL 2022)*. Marseille, France.

Rokade, Prakash P. and Aruna Kumari, 2019. Business Intelligence Analytics using Sentiment Analysis—A Survey. *International Journal of Electrical and Computer Engineering (IJECE)*.

Statista, 2022. African Countries with the Largest Population as of 2020.

Thelwall, Mike A, Kevan Buckley, and Georgios Paltoglou, 2011. Sentiment in Twitter Events. *J. Assoc. Inf. Sci. Technol.*, 62:406–418.

Vilares, David, Miguel A Alonso, and Carlos Gómez-Rodríguez, 2016. EN-ES-CS: An English-Spanish Code-Switching Twitter Corpus for Multilingual Sentiment Analysis. In *Tenth International Conference on Language Resources and Evaluation (LREC'16)*.

Vosoughi, Soroush, Helen Zhou, and Deb Roy, 2016. Enhanced Twitter Sentiment Classification using Contextual Information. *CoRR*, abs/1605.05195.

Zucco, Chiara, Huizhi Liang, Giuseppe Di Fatta, and Mario Cannataro, 2018. Explainable Sentiment Analysis with Applications in Medicine. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*.

Uzbek text's correspondence with the educational potential of pupils: a case study of the School corpus

Khabibulla Madatov[1], Sanatbek Matlatipov[2], Mersaid Aripov[2]

[1]Urgench State University, 14, Kh.Alimdjan str, Urgench city, 220100, Uzbekistan
habi1972@mail.ru
[2]National University of Uzbekistan named after Mirzo Ulugbek, 4 Universitet St, Tashkent, 100174, Uzbekistan
{s.matlatipov, mirsaid.aripov}@nuu.uz

Abstract

One of the major challenges of an educational system is choosing appropriate content considering pupils' age and intellectual potential. In this article the experiment of primary school grades (from 1st to 4th grades) is considered for automatically determining the correspondence of an educational materials recommended for pupils by using the School corpus where it includes the dataset of 25 school textbooks confirmed by the Ministry of preschool and school education of the Republic of Uzbekistan. In this case, TF-IDF scores of the texts are determined, they are converted into a vector representation, and the given educational materials are compared with the corresponding class of the School corpus using the cosine similarity algorithm. Based on the results of the calculation, it is determined whether the given educational material is appropriate or not appropriate for the pupils' educational potential.

Keywords: School corpus, semantic similarity, cosine similarity, term frequency, inverse document frequency

1. Introduction

The main goal of the education system is to educate young people who are physically healthy, mentally and intellectually developed, think independently, and have a strict point of view on life. In this regard, fundamental reform of the quality of general primary schools' textbooks and didactic materials that is appropriate for the intellectual potential of pupils is one of the urgent issues. The main purpose of this article is to automatically match the Uzbek textbook to the educational potential of students. In order to solve this problem we use text similarity for Uzbek texts. Text similarity is one of advanced methods of text analysis in the field of NLP. Based on the sources studied up to now, it is worth noting that the similarity of texts is used in a number of fields, such as information retrieval, categorization, machine translation, and automatic essay evaluation. Therefore, this article talks about the TF-IDF(SPARCK JONES, 1972) approach, which provides pupils with appropriate educational resources using the similarity of texts. That is to say, we determine whether the educational material corresponds to the potential of a pupil (predetermined classes) or not by choosing the most similar score compared to other classes which as a result helps to improve the quality of education.

Uzbek language. Uzbek is a low-resource Turkic language spoken by over 30 million people primarily in Uzbekistan and other neighboring countries in Central Asia. It is the official language of Uzbekistan and is widely used in education, media, and official communications. The Latin script being the official alphabet, but the old Cyrillic script is equally used in documents, websites and social media, requiring additional step of transliteration when dealing with(Salaev et al., 2022a). The Uzbek language, like many other closely related Turkic languages, is characterized by its use of vowel harmony and agglutinative grammar, which involves stringing together morphemes to create complex words[1]. These linguistic features present unique challenges for natural language processing tasks. Despite this, there have been recent efforts to develop NLP resources for Uzbek, including corpora, lexical databases, and machine learning models(Kuriyozov et al., 2022). These resources hold great potential for advancing the field of Uzbek NLP and improving access to information and communication technology for Uzbek-speaking populations.

All the resources, including the School corpus dataset, are publicly available[2]. The remainder of this paper is organised as follows: after this Introduction, Section 2 describes related works. It is followed by a description of the methodology in Section 3 and continues with Section 4. which focuses on Experiments \& Discussions. The final Section 5 concludes the paper and highlights the future work.

2. Related works

So far, there have been numerous approaches developed for the task of measuring the semantic similarity of words and texts.

One of the biggest challenges when dealing with large numbers of documents is finding the information you're looking for that fits your problem. This problem can be easily solved by methods of determining the similarity of texts. The article(Matlatipov, 2020) presents the algorithm of cosine similarity of Uzbek texts, based on TF-IDF to determine similarity. Semantic relationships between words are one of the key concepts in assessing natural language processing. In this paper(Salaev et al., 2022b), the authors present the SimRelUz-set dataset for evaluating the semantic model of the Uzbek language. This article(Elov et al., 2022) examines the process of sorting documents in the Uzbek language corpus by keywords using the TF-IDF method. The paper(Pradhan et al., 2015) describes different

[1] More on the Uzbek language: https://en.wikipedia.org/wiki/Uzbek_language

[2] https://zenodo.org/record/5659638 - the dataset of School corpus

types of similarity like lexical similarity, semantic similarity. The article also effectively classifies the measurement of text similarity between sentences, words, paragraphs, and documents. Based on this classification, we can get the best relevant document that matches the user's request. Paper(Islam & Inkpen, 2008) presents a text semantic similarity measurement method, a corpus-based measure of word semantic similarity, and a normalized and modified version of the longest common subsequence (LCS) string matching algorithm.

Existing methods for computing text similarity mainly focus on large documents or individual words. In this paper(Keleş & Özel, 2017), research has been carried out on methods such as similarity calculation between Turkish text documents, plagiarism detection and author detection, text classification and clustering. (San'atbek, 2018) paper established automation linguistic processes of dictionary-thesauruses for Uzbek language. As an pre-processing tool (Matlatipov Sanatbek and Tukeyev, 2020) paper offered lexicon-free stemming tool for Uzbek language whereas(Sharipov & Sobirov, 2022) offered rule-based algorithm.

Although there has been a rapid growth in the research production of NLP resources and tools for the low-resource Uzbek language, there is still a huge gap left to catch up with the current need for the trending technologies, such as artificial intelligence (AI).

In document analysis, an important task is to automatically find keywords which best describe the subject of the document. One of the most widely used techniques for keyword detection is a technique based on the term frequency-inverse document frequency (TF-IDF) heuristic. This technique has some explanations, but these explanations are somewhat too complex to be fully convincing. In this paper(Havrlant & Kreinovich, 2017) authors provide a simple probabilistic explanation for the TF-IDF heuristic. In(De Boom et al., 2016) the authors defined a novel method for the vector representations of short texts. The method uses word embeddings and learns how to weigh each embedding based on its IDF value. The proposed method works with texts of a predefined length but can be extended to any length. The authors showed that their method outperforms other baseline methods that aggregate word embeddings for modelling short texts.

In this article, using the "School Corpus", we provide information on the issue of determining the suitability of recommended educational materials for schoolchildren to the intellectual potential of students based on the lexical similarity of texts. The paper considers a problem-solving method based on TF-IDF. The TF-IDFs of the texts are determined, they are converted into a vector, and the given educational material is compared with the corresponding class of the "School Corpus" using the cosine similarity(Han et al., 2012) algorithm of the text similarity. According to the calculation results, it is determined whether the given educational material corresponds to the student's scientific potential or not.

3. Methodology

In this section, we describe the methodology based on TF-IDF and cosine similarity of corpus-based texts. Primary Uzbek school grades consist of {1, 2, 3, 4}-classes. So, we select and analyze texts which are suitable only from 1st to 4th grade pupils that are included in the School corpus even though there are more classes which we are not considering.

3.1 Data collection & pre-processing

The development of spoken language starts at home in the local environment, but the school plays a key role in the development of human thinking(Madatov et al., 2022a, 2022b, 2022c). Therefore, it is a natural way to start studying the automatic analysis of texts from school textbooks. Because of this point of view, we decided to collect from the best open source available websites related to school educational materials. We found two best available websites (www.ziyonet.uz, www.kitob.uz). Among them, we decided to choose kitob.uz because of the availability of the same book in multiple languages which can be a great potential as a parallel corpus which accelerates our future works. Overall, 34 books have been downloaded and converted from pdf to txt format, manually. As a result, the School corpus according to Uzbek primary school consists of the following (table 1.):

From Table 1 we can see that Class 1 corpus has 24107 tokens, 7978 unique words, Class 2 has 56650 tokens, 14858 unique words, Class 3 has 90255 tokens, 21124 unique words and 4 tokens. The class corpus was found to contain 109024 tokens and 24736 unique words. The total number of unique words in primary school classes was 42,797.

Classes	1st class	2nd class	3rd class	4th class
Number of total tokens	24,107	56,650	90,225	24,736
Number of unique tokens	7,978	14,858	21,124	24,736

Table 1. School corpus which is constructed using primary school textbooks.

3.1.1 Algorithm

The main problem is to determine the appropriate class for the target resource. We present algorithm 1. for solving this problem based on the TF-IDF method as follows.

Algorithm of finding which classes the given text corresponds to:

1. Tokenization of the given text.
2. A separate vocabulary is created for each class (based on textbooks) and the given text. These words are called unique words(bag-of-words)
3. If all unique words of the given text belong to the set of the class's unique words then go to 8.
4. TF-IDFs are calculated for each class and the given text. Vectors are created whose coordinates are equal to the TF-IDF values of the unique words. The order of the vector coordinates corresponds to the order of unique words. If the given unique word does not occur in the text in question, this coordinate of the corresponding vector will be zero.
5. Let these vectors be *v1, v2, v3, v4* according to classes 1,2,3,4 and be *v* according to the given text. For each class, we consider vector pairs (v, v_i). i = 1, 2, 3, 4. In this case, the size of the vector v is changed according to the size of the vector v_i.
6. Cosine similarity of v with each v_1, v_2, v_3, v_4 are calculated in (1).

$$\cos(v, v_i) = \frac{(v, v_i)}{|v| \cdot |v_i|} \qquad (1)$$

here (v, v_i) is a scalar product of vectors v and v_i. $i = 1, 2, 3, 4$.

7. Max value of $\cos(v, v_i)$ is chosen.
8. It is concluded that this text corresponds to class i.

4. Experiments & Discussions

Cosine similarities values are shown for each class from 1st to 4th grades in the comparison symmetric matrix, diagonally (Table 2). Let the vector of i-th class be v_i, i=1,2,3,4 respectively. One notable part is that the percentage of each class is increasing horizontally with row elements above the main diagonal, which in fact, means that pupils' lexicon increases from class to class. That is to say, the reason 4th class pupils have knowledge of previous classes is natural. Let the vector of i-th class is vi respectively, i=1,2,3,4. From the similarity of the vectors follow similarity of the texts respectively.

Classes	1st class	2nd class	3rd class	4th class
1st class	1 7978	0.34 4252	0.34 4755	0.34 4792
2nd class	0.39 4252	1 14858	0.42 7852	.044 8124
3rd class	0.36 4755	0.44 7852	1 21124	0.45 10349
4th class	0.34 4795	0.42 8128	0.45 10353	1 24736

Table 2. The similarity score of the classes, which includes the number of unique words, respectively.

At the next stage, texts were taken from various sources in order to evaluate the algorithm including, the Journal texts (from pupils journal “Gulxan”) and internet materials. Table 3 shows sources of the texts.

№	File name	The source
1	class-1.txt	Total textbooks for 1st grade
2	class-2.txt	Total textbooks for 2ndgrade
3	class-3.txt	Total textbooks for 3rd grade
4	class-4.txt	Total textbooks for 4th grade
5	mujiza.txt	Internet resource
6	ayiq.txt	Gulxan jurnal
7	vatan.txt	Hozir
8	sariq-dev.txt	Sariq devni minib
9	toshkent.txt	Topic of 2nd class
10	hikoya.txt	3rd grade
11	kichik-vatan.txt	4th grade text

Table 3. School corpus which is constructed using primary school textbooks.

In the Table 4 texts similarity are considered as a result of vector similarity, respectively. From the Table 4., it can concluded:

1. Texts *mujiza.txt, ayiq.txt, vatan.txt, kichik-vatan.txt* more similar to 4-th class. It means that these texts-correspondence with the educational potential of pupils of 4-th class. So, these texts are recommended to teach in 4-th class.
2. The text -toshkent.txt is similar to 2-nd class. So, this text is recommended to teach in 2-nd class.
3. The text-hikoya.txt is similar to a 3-rd class. So, this is recommended to teach in 3-rd class.

Classes	class-1	class-2	class-3	class-4
mujiza.txt	0.07	0.08	0.08	0.11
ayiq.txt	0.05	0.05	0.06	0.07
vatan.txt	0.1	0.14	0.14	0.15
sariq-dev.txt	0.2	0.21	0.2	0.3
toshkent.txt	0.07	1	0.05	0.06
hikoya.txt	0.1	0.1	1	0.1
kichik-vatan.txt	0.1	0.1	0.08	1

Table 4. School corpus which is constructed using primary school textbooks.

The detailed explanation of the proposed algorithm with all the steps is given in Annex 1.

5. Conclusion and future work

In conclusion, this research aimed to tackle one of the major challenges in the educational system, which is selecting appropriate educational content for primary school pupils based on their age and intellectual potential. By utilizing the School corpus and the cosine similarity algorithm, this study aimed to determine the suitability of educational materials for pupils in grades 1st to 4th. The results of the experiment showed that the method of converting TF-IDF scores into vector representations and comparing the educational materials with the corresponding class in the School corpus was effective in identifying whether a given educational material was appropriate or not for a particular grade level.

These findings hold important implications for education professionals, policymakers, and researchers in the field. By demonstrating the potential of NLP techniques to support the selection of appropriate educational content, this study lays the foundation for future research on the application of NLP in the educational domain. Overall, this study has the potential to contribute to the improvement of the educational system in Uzbekistan and beyond, by providing a data-driven approach to selecting educational content that is aligned with pupils’ age and intellectual potential.

Data availability

All the Python codes used for the evaluation of the proposed models for the School Corpus are publicly available at the project repository. The application code of the proposed methodology will serve as a valuable resource for further NLP research on Uzbek language.

Acknowledgements

This research work was fully funded by the REP-25112021/113 - “UzUDT: Universal Dependencies Treebank and parser for natural language processing on the Uzbek language” subproject funded by The World Bank project “Modernizing Uzbekistan national innovation system” under the Ministry of Innovative Development of Uzbekistan.

Annex 1

Algorithm of finding which classes the given text corresponds to.

```
1:INPUT: (class-1.txt, class-2.txt, class-3.txt, class-4.txt, text.txt{given text})
2:        Token(class-1.txt, class-2.txt, class-3.txt, class-4.txt, text.txt)
3:        // reset tokens
4:              class-1.txt :=Token(class-1.txt)
5:              class-2.txt :=Token(class-2.txt)
6:              class-3.txt :=Token(class-3.txt)
7:              class-4.txt :=Token(class-1.txt)
8:  Procedure similarity (class.txt, text.txt);
9:        begin
10:                m:=dictionary of (class.txt, text.txt) {Creation unique words of (class.txt,
              text.txt}
11:                //Creating of vectors of the class.txt and the text.txt
12:                s:=0:p:=0:t:=0
13:                for j:=1 to length(m) do
14:                  begin if m(j) in text.txt then v(j):= TF-IDF(m(j))  else v(j):=0
15:                        if m(j) in class.txt then v1(j):= TF-IDF(m(j)) else v1(j):=0
16:                        s:=s+v(j)*v1(j): p:=p+v(j)* v(j); t:=t+ v1(j)* v1(j)
17:                        cosine:=s/(abs(sqrt(p))* abs(sqrt(t)))
18:                  end
19:           end
20:   if All tokens of text.txt in class-1.txt then
21:              begin    print('given text similar  to', class-1.txt)
22:                       break: go to 47
23:              end
24:  else if  All tokens of text.txt in class-2.txt then
25:              begin print('given text similar  to', class-2.txt)
26:                       break: go to 47
27:              end
28:  else if  All tokens of text.txt in class-3.txt then
29:              begin print('given text similar  to', class-3.txt)
30:                       break: go to 47
31:               end
32:  else if  All tokens of text.txt in class-4.txt then
33:              begin print('given text similar  to', class-4.txt)
34:                       break: go to 47
35:               end
36: else
37:              begin
38:                    similarity(class-1.txt, text): max=cosine A:=' class-1.txt'
39:                    similarity(class-2.txt, text)
40:                    if  cosine>max then  begin max=cosine: A:= ' class-2.txt' end
41:                    similarity(class-3.txt, text)
42:                    if  cosine>max then  begin max=cosine: A:= ' class-3.txt' end
43:                    similarity(class-4.txt, text)
44:                    if  cosine>max then  begin max=cosine: A:= ' class-4.txt' end
45:              end
46: print ('given text similar to'-A)
47: end
```

References

De Boom, C., Van Canneyt, S., Demeester, T., & Dhoedt, B. (2016). Representation learning for very short texts using weighted word embedding aggregation. *Pattern Recognition Letters*, *80*, 150–156. https://doi.org/https://doi.org/10.1016/j.patrec.2016.06.012

Elov, B., Khusainova, Z., & Khudayberganov, N. (2022). *A STATISTICAL INDEX CALCULATED USING THE TF-IDF FOR TEXTS IN THE UZBEK LANGUAGE CORPUS*. https://doi.org/10.5281/zenodo.7440059

Han, J., Kamber, M., & Pei, J. (2012). 2 - Getting to Know Your Data. In J. Han, M. Kamber, & J. Pei (Eds.), *Data Mining (Third Edition)* (Third Edition, pp. 39–82). Morgan Kaufmann. https://doi.org/https://doi.org/10.1016/B978-0-12-381479-1.00002-2

Havrlant, L., & Kreinovich, V. (2017). A simple probabilistic explanation of term frequency-inverse document frequency (tf-idf) heuristic (and variations motivated by this explanation). *Int. J. Gen. Syst.*, *46*(1), 27–36.

Islam, A., & Inkpen, D. (2008). Semantic Text Similarity Using Corpus-Based Word Similarity and String Similarity. *ACM Trans. Knowl. Discov. Data*, *2*(2). https://doi.org/10.1145/1376815.1376819

Keleş, M. K., & Özel, S. A. (2017). Similarity detection between Turkish text documents with distance metrics. *2017 International Conference on Computer Science and Engineering (UBMK)*, 316–321. https://doi.org/10.1109/UBMK.2017.8093399

Kuriyozov, E., Matlatipov, S., Alonso, M. A., & Gómez-Rodr\'\iguez, C. (2022). Construction and evaluation of sentiment datasets for low-resource languages: The case of Uzbek. *Human Language Technology. Challenges for Computer Science and Linguistics: 9th Language and Technology Conference, LTC 2019, Poznan, Poland, May 17–19, 2019, Revised Selected Papers*, 232–243.

Madatov, K., Bekchanov, S., & Vičič, J. (2022a). *Accuracy of the Uzbek stop words detection: a case study on "School corpus."* arXiv. https://doi.org/10.48550/ARXIV.2209.07053

Madatov, K., Bekchanov, S., & Vičič, J. (2022b). Dataset of stopwords extracted from Uzbek texts. *Data in Brief*, *43*, 108351. https://doi.org/https://doi.org/10.1016/j.dib.2022.108351

Madatov, K., Bekchanov, S., & Vičič, J. (2022c). Automatic detection of stop words for texts in the Uzbek language. In *Preprints*.

Matlatipov Sanatbek and Tukeyev, U. and A. M. (2020). Towards the Uzbek Language Endings as a Language Resource. In K. and S. E. Hernes Marcin and Wojtkiewicz (Ed.), *Advances in Computational Collective Intelligence* (pp. 729–740). Springer International Publishing.

Matlatipov, S. G. (2020). Cosine Similarity and its Implementation to Uzbek Language Data. *Central Asian Problems of Modern Science and Education*, *2020*(4), 95–104.

Pradhan, N., Gyanchandani, M., & Wadhvani, R. (2015). A Review on Text Similarity Technique used in IR and its Application. *International Journal of Computer Applications*, *120*, 29–34. https://doi.org/10.5120/21257-4109

Salaev, U., Kuriyozov, E., & Gómez-Rodr\'\iguez, C. (2022a). A machine transliteration tool between Uzbek alphabets. *ArXiv Preprint ArXiv:2205.09578*.

Salaev, U., Kuriyozov, E., & Gómez-Rodr\'\iguez, C. (2022b). SimRelUz: Similarity and Relatedness Scores as a Semantic Evaluation Dataset for Uzbek Language. *Proceedings of the 1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages*, 199–206. https://aclanthology.org/2022.sigul-1.26

San'atbek, M. (2018). Modeling WordNet type thesaurus for Uzbek language semantic dictionary. *International Journal of Systems Engineering*, *2*(1), 26.

Sharipov, M., & Sobirov, O. (2022). *Development of a rule-based lemmatization algorithm through Finite State Machine for Uzbek language*. arXiv. https://doi.org/10.48550/ARXIV.2210.16006

SPARCK JONES, K. (1972). A STATISTICAL INTERPRETATION OF TERM SPECIFICITY AND ITS APPLICATION IN RETRIEVAL. *Journal of Documentation*, *28*(1), 11–21. https://doi.org/10.1108/eb026526

Uzbek text summarization based on TF-IDF

Khabibulla Madatov[1], Shukurla Bekchanov[2], Jernej Vičič[3]

[1]Urgench state university, 14, Kh. Alimdjan str, Urgench city, 220100, Uzbekistan
habi1972@mail.ru
[2]Urgench state university, 14, Kh. Alimdjan str, Urgench city, 220100, Uzbekistan
shukurla15@gmail.com
[3]Research Centre of the Slovenian Academy of Sciences and Arts, The Fran Ramovš Institute, Novi trg 2, 1000 Ljubljana, Slovenija. University of Primorska, FAMNIT, Glagoljaska 8, 6000 Koper, Slovenia
jernej.vicic@upr.si

Abstract

The volume of information is increasing at an incredible rate with the rapid development of the Internet and electronic information services. Due to time constraints, we don't have the opportunity to read all this information. Even the task of analyzing textual data related to one field requires a lot of work. The text summarization task helps to solve these problems. This article presents an experiment on summarization task for Uzbek language, the methodology was based on text abstracting based on TF-IDF algorithm. Using this density function, semantically important parts of the text are extracted. We summarize the given text by applying the n-gram method to important parts of the whole text. The authors used a specially handcrafted corpus called "School corpus" to evaluate the performance of the proposed method. The results show that the proposed approach is effective in extracting summaries from Uzbek language text and can potentially be used in various applications such as information retrieval and natural language processing. Overall, this research contributes to the growing body of work on text summarization in under-resourced languages.

Keywords: Text summarization, Uzbek language, TF-IDF, School corpus.

1. Introduction

The task of text summarization is to reduce a given text to a shorter version while retaining the most important information. The Internet, web pages, news, articles, status updates, blogs, as well as the works of scientists who make new discoveries every day, etc., are the source for the immensity of text files. The sheer amount of information generated and distributed in everyday life presents a lot of problems, we have to find, use and process the necessary textual information. In this case, we are faced with the problem of text summarization for text data analysis. Text summarization is one of the most important applications of natural language processing (NLP).

Automatic document summarization has the following preferences:

1) reduces the time of studying the data;
2) speeds up the process of document analysis;
3) facilitates the process of selecting documents when investigating documents;
4) provides more summary options than the traditional text summary.

The minimum requirement for automatic summarization is that the number of text words obtained as a result of automatic summarization does not exceed approximately 30% (Torres-Moreno, 2014, page 33) of the number of original text words.

Automatic text summarization is a complex process, consisting of several phases. The first phase is an initial processing of the text, which often includes the following steps:

- separating the text into parts, sentences, paragraphs, etc.;
- parsing segments into words or tokenization;
- standardization of the words (lemmatization, stemming, etc.);
- removal of the stop words.

Initial processing is a difficult task that largely depends on the language in which the text is written. For example, sentence boundaries are marked by punctuation. Their usage varies considerably from language to language. Also, not all words are separated by spaces in all languages. There are great differences between a text written in an Oriental language such as Chinese, Japanese, or Korean, and a text written in European Latin characters.

This problem also occurs in other steps, such as lemmatization or stemming, the normalization of words using the grammatical part of speech tagging. Even the removal of stop words[1] (X. A. Madatov et al., 2021) depends on the natural language and again it is not a trivial task.

This article is dedicated to Uzbek text summarization based on an algorithm using TF-IDF(Aizawa, 2003). Although the issue of summarization for rich-resourced languages has been solved by various methods, the issue of text summarization for low-resourced languages such as Uzbek still remains an open issue. That is the main motivation for the presented research on text summarization based on TF-IDF. In this case, a dictionary is created from the words contained in the texts and a probability distribution law is created based on their TF-IDF values. In order to assess the results of the proposed method, a corpus named “School corpus" was used, which was created using freely available school books such as “Reading book", “Mother tongue" and “Literature".

[1] If the removal of those words from the text not only does not change the context meaning but also leaves the minimum number of words possible that can still hold the meaning of the context, then such words can be called stop words for this work.

Uzbek language. The Uzbek language is a Turkic language spoken by more than 30 million people in Uzbekistan and neighboring Central Asian countries. It is the official language of Uzbekistan and is written using the Latin alphabet. Due to its rich history and culture, and the language structure has been heavily influenced by other languages such as Arabic, Persian, and Russian. The Uzbek language has several dialects, with the main dialects being Karakalpak, Khorezm, and Samarkand, among others. Despite its importance, there has been relatively little research on text summarization for the Uzbek language. Our approach aims to fill this gap and provide a useful tool for summarizing Uzbek text.

This paper is structured as follows: We first start with an introduction, in Section 1. Section 2 presents, related works. The main contribution of this work, is presented in Section 3 in the form of methodology. Following, experiments and results are presented in Section 4, which includes information regarding results obtained in this work. Lastly, we conclude this paper with the discussion and conclusion in Section 5.

2. Related works

Automatic text summarization originated in the 1950s through the research of Hans Peter Luhn. Luhn first created a model of the summarization of scientific and technical articles(Luhn, 1958). Inspired by his work, many other researchers produced a valuable amount of research output, such as the developed an automatic abstracting system using sentence selection and rejection techniques, and a Word Control List (Edmundson & Wyllys, 1961), which produced high-quality abstracts that warrant large-scale testing (Rush et al., 1971).

Pollock and Zamora studied Chemical Abstracts Service's research on text summarization using a modified Rush-Salvador-Zamora algorithm. They found that some subjects are better suited for automatic extraction and suggest customizing the algorithm for narrow subject areas for better results, they also discussed the viability of automatic extraction (Pollock & Zamora, 1975).

In recent years, researchers have proposed various approaches to improve the effectiveness of text summarization, from cpombining TF-IDF algorithm with the Latent Dirichlet Allocation (LDA) algorithms to generate more informative summaries, to neural network-based approachs that incorporate the attention mechanism to capture the semantic information of the text and improve the summarization quality (Lehnert & Ringle, 2014).

The following works are to the research of the natural language processing of the Uzbek language in (K. Madatov, 2019; K. A. Madatov et al., 2022).

The relationship of Uzbek words is observed in there have been works on gap-filling tasks extracting Uzbek stop words on the example of School corpus is presented in (K. Madatov et al., 2022b) and (K. Madatov et al., 2022c). The level of accuracy of Uzbek stop words detection is presented in (K. Madatov et al., 2022a), Uzbek stemming and lemmatization is discussed in (Sharipov & Sobirov, 2022), are the basic ones used in this research work.

Above all, the Uzbek language has seen a recent grooving trend in the production of NLP-related research works and resources, among them, a machine transliteration tool (Salaev et al., 2022a), sentiment analysis dataset and analyser models (Kuriyozov et al., 2022), as well as semantic evaluation datasets (Salaev et al., 2022b) are some of the many.

3. Methodology

The Uzbek language belongs to the family of agglutinative languages, most of the methods for inflected languages text summarizing cannot be directly used for the Uzbek language.

The scientific novelty of the article: Creating Uzbek text summarization method based on TF-IDF for the School corpus (K. Madatov et al., 2022c).

In the article (K. Madatov et al., 2022a) considered and proved the problem of finding a part of Uzbek texts, containing stop words for the School corpus. Using this method, it is possible to find the important part of the given text. Below, we present the algorithm of the Uzbek text summarization method. A brief description of steps taken in the implementation of the proposed method is given in Algorithm 1, and the full algorithm can be found in Annex 1.

Algorithm 1: Uzbek text summarization

1. The given text (initial text) is separated to words, later addressed as Text
2. Removal of stop words using the dataset extracted from School corpus
3. A unique dictionary from the resulting Text was created, addressed as Text UW
4. For each a_i, which belongs to Text_UW, w_i=TF-IDF(a_i) is calculated. Then w_i is transferred $p_i(a)$ by using the formula $p_i=w_i/\Sigma w_i$.
   Density function is created for Text_UW. This step includes the following computational process:(K. Madatov et al., 2022a, 4-5 pages)
   - $E=\Sigma i \cdot p_i$- the mathematical expectation of the unique words
   - $D=\Sigma (i-E)^2 \cdot p_i$- dispersion of the unique words
   - $\sigma=\sqrt{D}$- standard deviation of the unique words
   - $E_k=\Sigma p_i \cdot i^k$ -of the unique words
   - $\mu_3=E_3-3 \cdot E_1 \cdot E_2+2 \cdot E_1^3$ k -third central moment of the unique words
   - $A_s=\mu_3/\sigma^3$- The asymmetry of the theoretical distribution
5. Important parts of the Text_UW depend on A_s. Let $k=E-\sigma$ and $m=E+\sigma$. We find words respectively k- and m-positions Text_UW and position of these words in initial

Text. Let they are k_1 and m_1 respectively.

6. a) If $A_s>0$, then the part of the Text from the beginning word up to k_1-th word part is reloaded into the Text, deleting the rest;
   b) If $A_s<0$, then the part of the Text from the m_1-th word up to the last word is reloaded into the Text, deleting the rest.;
   c) If $A_s=0$, then the part of the Text from k_1-th word up to m_1-th word is reloaded into the Text, deleting the rest.
7. The 3 – gram method is applied to the Text
8. The result is printed.

4. Experiments and results

The methodology presents the applying of the Uzbek text summarization algorithm to the given text that, solves the complicated problem of finding the significant part and summarization of the given text based on TF-IDF. In this part of the paper we show an example of applying the Uzbek text summarization algorithm.

The adventure novel "Sariq devni minib (Riding the Yellow Giant)" by Kh. Tokhtabayev was chosen as an example for the experiment. A part of this masterpiece is given in the literature textbook of 8th class.

The application of the Uzbek text summarization algorithm to this text can be explained in a step by step recipe as the following:

1. This masterpiece consists of 49,705 tokens and among them 13,740 are unique words.
2. Remove stop words (K. Madatov et al., 2021). As a result, we get Text.
3. Creates a dictionary of Text. This id called unique words and the set of these words are denoted as Text_UW. In this case len(Text_UW)=13740.
4. Each unique word is denoted as a_i; iϵ[1...13740]. For each a_i, w_i=TF-IDF(a_i) is calculated. The probability of the word a_i can be calculated using the following formula:

$$p_i=w_i/\sum w_i$$

Numerical values of the probability of unique words are presented in Table 1.

Table -1: Numerical values of the probability of unique words

Variable	Item
E	4379,22
D	16019213,55
Σ	4002,4
σ^3	64115315874
E_1	4379,22
E_2	35196780,35
E_3	3,40214E+11
μ_3	45776660238
A_s	0,714

5. K=377.
6. In our case As>0. The corresponding word for K in Text_UW is “shavla”. We find the first occurrence of the word "shavla" in the Text.We cut the part from the first word of the Text to the word "Shavla" and reload to the Text.
7. We apply the 3rd gram to the Text.
8. Print the result. Figure1

The method is presented in Figure 1.

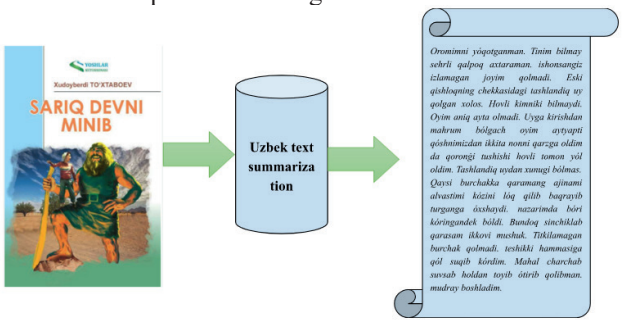


Fig. 1: The result of the application of the Uzbek text summarization algorithm

5. Discussion and Conclusion

3 gram were used to summarize the text taken in the experiment. In general, the problem of applying the n-gram to the problem of the Uzbek text summarization and choosing the best n-gram remains open. Despite the openness of this problem, similar problems can be solved using the algorithm, which is the main result of the article. For this, i-gram is taken instead of 3-gram. Using the results obtained for all i-grams, the expert selects the best n-gram.

This article presents an algorithm based on TF-IDF for solving the text summarization for Uzbek language problem. In the process of applying the algorithm, we removed the stop words using the previously created School corpus. The usefulness of the obtained result for further research can be expressed as follows:

1) solving Uzbek natural language processing problems;
2) choosing the best n-gram in the problem of Uzbek text summarization based on TF-IDF;
3) in solving the problem of Uzbek text summarization, how to productive Uzbek text summarization based on TF-IDF in comparison with the trend methods of the Uzbek text summarization. And many other use-cases like these.

Acknowledgments. The authors gratefully acknowledge the Erasmus+ program, ELBA (Establishment of training and research centers and courses development on intelligent big data analysis in Central Asia) project (project reference number: 610170-EPP-1-2019-1-ES-EPPKA2-CBHE-JP).

The authors of this paper also present a sincere gratitude to the NLP team at Urgench State University for the enormous support in discussions during the model creation and experiments.

References

Aizawa, A. (2003). An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, *39*(1), 45–65.

Edmundson, H. P., & Wyllys, R. E. (1961). Automatic abstracting and indexing—survey and recommendations. *Communications of the ACM*, *4*(5), 226–234.

Knorz, G., Krause, J., & Womser-Hacker, C. (1993). Information Retrieval '93. Von der Modellierung zur Anwendung. *Proceedings Der*, *1*, 67–81.

Kupiec, J., Pedersen, J., & Chen, F. (1995). A trainable document summarizer. *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 68–73.

Kuriyozov, E., Matlatipov, S., Alonso, M. A., & Gómez-Rodrguez, C. (2022). Construction and Evaluation of Sentiment Datasets for Low-Resource Languages: The Case of Uzbek. *Human Language Technology. Challenges for Computer Science and Linguistics: 9th Language and Technology Conference, LTC 2019, Poznan, Poland, May 17–19, 2019, Revised Selected Papers*, *13212*, 232–243.

Lehnert, W. G., & Ringle, M. H. (2014). *Strategies for natural language processing*. Psychology Press.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, *2*(2), 159–165.

Madatov, K. (2019). *A prolog format of uzbek WordNet's entries*.

Madatov, K. A., Khujamov, D. J., & Boltayev, B. R. (2022). Creating of the Uzbek WordNet based on Turkish WordNet. *AIP Conference Proceedings*, *2432*(1), 60009.

Madatov, K., Bekchanov, S., & Vičič, J. (2021). *Lists of uzbek stopwords*. Univerza na Primorskem, Inštitut Andrej Marušič.

Madatov, K., Bekchanov, S., & Vičič, J. (2022a). Accuracy of the Uzbek stop words detection: a case study on" School corpus". *ArXiv Preprint ArXiv:2209.07053*.

Madatov, K., Bekchanov, S., & Vičič, J. (2022b). *Automatic detection of stop words for texts in the Uzbek language*.

Madatov, K., Bekchanov, S., & Vičič, J. (2022c). Dataset of stopwords extracted from Uzbek texts. *Data in Brief*, *43*. https://doi.org/10.1016/j.dib.2022.108351

Madatov, X. A., Sharipov, M. S., & Bekchanov, S. K. (2021). O'zbek tili matnlaridagi nomuhim so'zlar. *KOMPYUTER LINGVISTIKASI: MUAMMOLAR, YECHIM, ISTIQBOLLAR Respublika I Ilmiy-Texnikaviy Konferensiya*, *1*, 156–162.

Pollock, J. J., & Zamora, A. (1975). Automatic abstracting research at chemical abstracts service. *Journal of Chemical Information and Computer Sciences*, *15*(4), 226–232.

Rush, J. E., Salvador, R., & Zamora, A. (1971). Automatic abstracting and indexing. II. Production of indicative abstracts by application of contextual inference and syntactic coherence criteria. *Journal of the American Society for Information Science*, *22*(4), 260–274.

Salaev, U., Kuriyozov, E., & Gómez Rodr\'\iguez, C. (2022a). A machine transliteration tool between Uzbek alphabets. *The International Conference on Agglutinative Language Technologies as a Challenge of Natural Language Processing (ALTNLP)*.

Salaev, U., Kuriyozov, E., & Gómez Rodr\'\iguez, C. (2022b). SimRelUz: Similarity and Relatedness scores as a Semantic Evaluation dataset for Uzbek language. *LREC2022 Special Interest Group on Under-Resourced Languages (SIGUL 2022) Workshop*.

Sharipov, M., & Sobirov, O. (2022). Development of a rule-based lemmatization algorithm through Finite State Machine for Uzbek language. *ArXiv Preprint ArXiv:2210.16006*.

Torres-Moreno, J.-M. (2014). *Automatic text summarization*. John Wiley & Sons.

Annex

Annex 1: The full algorithm of a TF-IDF based text summarization for Uzbek texts.

```
1:INPUT: (Text)
2:      Token(Text)
3:              Remove(Collacation_Two_Words(Text))
4:              Remove(Unigram(Text))
5:              Remove(RuleBase(Text))
                ///{pronoun, adverb, conjunction, introductory word, adverbial word, auxiliary word, prepositions}
6:              TF-IDF(Text)
7:              p(Text_UW)= TFIDF(Text_UW)/SUM(TFIDF(Text_UW))
                // Text_unique words- Text_UW
8:              E=SUM(Text_UW)*p(Text_UW)
9:              D=SUM(Text_UW -E)^2*p(Text_UW)
10:             SIGMA=SQR(D)
11:             EI(m)=SUM(p(Text_UW))* Text_UW ^m// m – 1..len(Text_UW)
12:             M1=E
13:             M2= SUM(p(Text_UW))* Text_UW ^2
14:             M3= SUM(p(Text_UW))* Text_UW ^3
15:             MIYU=M3-3*M1*M2+2*M1^3
16:             AS= MIYU/ SIGMA^3
17:     If(AS>0) then Text2=TFIDF[0;E-SIGMA]
18:     If(AS<0) then Text2=TFIDF[E-SIGMA- len(Text_UW)]
19:     If(AS=0) then Text2=TFIDF[E-SIGMA; E+SIGMA]
20:     Gram3(Text2)
21:             Token(Text2)
22:             SummarText:= P(token(i)|token(i+1)| token(i+2))
23:     Print(SummarText)
24:END
```

Easy-to-Read in Germany: A Survey on its Current State and Available Resources

Margot Madina[1], Itziar Gonzalez-Dios[2], Melanie Siegel[1]

[1]Darmstadt University of Applied Sciences (Hochschule Darmstadt)
margot.madina-gonzalez@h-da.de
melanie.siegel@h-da.de
[2]HiTZ Zentroa-Ixa, Euskal Herriko Unibertsitatea (UPV/EHU)
itziar.gonzalezd@ehu.eus

Abstract

Easy-to-Read Language (E2R) is a controlled language variant that makes any written text more accessible through the use of clear, direct and simple language. It is mainly aimed at people with cognitive or intellectual disabilities, among other target users. Plain Language (PL), on the other hand, is a variant of a given language, which aims to promote the use of simple language to communicate information. German counts with Leichte Sprache (LS), its version of E2R, and Einfache Sprache (ES), its version of PL. In recent years, important developments have been conducted in the field of LS. This paper offers an updated overview of the existing Natural Language Processing (NLP) tools and resources for LS. Besides, it also aims to set out the situation with regard to LS and ES in Germany.

Keywords: Easy-to-Read, Leichte Sprache, einfache Sprache, readability, accessibility

1. Introduction

Access to information, knowledge and culture is a right for all citizens. As stated in the Convention on the Rights of Persons with Disabilities of the United Nations (CRPD), access to information and communication is a fundamental right, and states should facilitate information in accessible ways such as easy to read and understand forms (United Nations, 2006). However, written text does not always match our ability to understand what we read; a lot of the textual information we find nowadays is too complicated to understand by people with communication disabilities. This causes their exclusion from society. Easy-to-Read (E2R) is a pivotal part of the inclusion for people with communication disabilities (Hansen-Schirra et al., 2020; Maaß and Hansen-Schirra, 2022).

E2R is a language variant of a standard language, with reduced complexity, and with the aim to improve the readability and comprehensibility of texts (Nitzke et al., 2022). One of its functions is to make content accessible, and to ensure participation for people with communication impairments (Hansen-Schirra and Maaß, 2020). It counts with a set of rules that cover the vocabulary, grammar structures and layout of the text, among others.

E2R is mainly aimed towards people with cognitive or intellectual disabilities; however, other target groups may also benefit from it. E2R's target groups include, but are not limited to, the following: people with intellectual, cognitive or developmental disabilities, people with auditory disabilities, people with low literacy, migrants, or children in need of reading reinforcement (Bredel and Maaß, 2016; Hansen-Schirra and Maaß, 2020; Maaß and Garrido, 2020; Maaß, 2019).

E2R usually receives a different name depending on the standard language it is based on; in the case of German, its E2R variant is known as *Leichte Sprache* (LS). It is not to be confused with what is known as Plain Language (PL) *Einfache Sprache* (ES) in German). PL is a variant of a given language, which aims to promote the use of simple language to communicate information. There are some important differences between them: (1) PL focuses on text while E2R covers text, illustrations and layout, (2) E2R is usually aimed at people with intellectual disabilities, as PL might be too challenging for them, (3) PL was initially focused on legal and governmental texts, due to their intrinsic meaning, while E2R is usually applied to all sorts of texts.

The rest of the paper is structured as follows: Section 2 will discuss the current state of LS in Germany. Section 3 will overview the existing resources and tools for LS, briefly describing how they are relevant in the LS field. Finally, Section 4 will present the conclusions.

2. LS in Germany

The first rules for LS were developed by Inclusion Europe back in 1998 (Pottmann, 2019). However, this first set of rules, even thought they were written in German, were generic rules for E2R that could be applicable to any standard language. In 2002, the German government was obliged to provide accessible information to everyone due to the establishment of the *Gesetz zur Gleichstellung von Menschen mit Behinderungen* (the German equality law for disabled people) and the Barrierefreie-Informationstechnik-Verordung (the accessible technology enactment). The *Netzwerk Leichte Sprache* (the plain language association) was founded in 2006, and they developed the rules for LS in 2013 (Pottmann, 2019)[1].

There is currently an ongoing debate related to LS rules, terminology, and its possible stigmatization effect. At present, there are three main currents regarding LS in Germany.

Andreas Baumert defends the use of ES over LS, claiming LS to be "falsches Deutsch" (bad German) (Baumert, 2018, 3) that cannot be used by many of its target groups. He strongly criticizes LS and, among the shortcomings he found, he says that it is all a business and that it cannot be generalized for all target groups. In his

[1] For a more detailed overview of how the legal state of LS has changed over time, refer to Maaß, 2020

view, ES has a better chance of taking shape in the foreseeable future; however, there is no generally accepted version of a simple language that is used by authorities, industry and associations alike (Baumert, 2016, 94).

Bettina Bock proposes that LS should not be understood as strictly bound to rules; even though they might be useful, they should not be seen as strict norms but rather as suggestions. She claims that LS should be used as an umbrella term that covers all approaches that this term encompasses. The focus should be on the users of LS and seek what is best for them, not so much on the rules, as there are some texts in LS that, even though they do not adhere strictly to the rules, are considered good by the community (Bock, 2018, 11).

Christiane Maaß advocates for the use of LS and the implementation of its rules. She proposed a set of rules for LS (Maaß, 2015) and is currently working on making them more precise and their regularization. She also claims that the term "Easy-to-Read" is not an adequate term for this language variant, as it is used in other forms of realization such as screen readers, sightseeing or museums, or interpreting in inclusive meetings and conferences. She proposes the term "Easy Language" (EL) instead, as it is open to broader conceptualizations (Maaß, 2020, 56). Besides, she also highlights that E2R and LS texts may have a stigmatizing effect on their target groups. Therefore, she proposes a new model, Leichte Sprache Plus (Easy Language Plus), which stands between PL and E2R (in the case of German, between LS and ES). This model "profits from the comprehension and perception principles of EL, but also from the non-stigmatising and more acceptable features of PL" (Hansen-Schirra and Maaß, 2020, 32) However, it is not an established approach yet, but it is only on its first steps.

Nowadays, LS has a very active practice in Germany, while ES is rarely used. A reason for this might be that it is safer for public bodies to go for LS, as ES might still be challenging for some target users, whereas LS reaches a wider audience (Maaß, 2020).

3. Resources and Tools for LS

This section will introduce the available resources and tools for LS. We concentrate on the most recent and relevant ones related to LS.

3.1. LS Corpora

A corpus (plural, corpora) is a linguistic resource consisting of a large, structured set of texts. It can be parallel (combines a simplified version of a text with its original version) or comparable (a collection of simplified documents and standard-language documents that share the same topic).

Klaper et al., 2013: Klaper et al. developed the first parallel, sentence-aligned corpus with German and LS texts. They crawled the data from five publicly available webpages. The corpus consists of around 70,000 to kens, and spans various topics. The quality of the sentence alignment obtained an F-score of 0.085.

LeiSa (Lange, 2018): this corpus was created in order to do an explorative corpus-based analysis as part of the research project *Leichte Sprache im Arbeitsleben* (LeiSA, Easy-to-Read in Work Contexts, University of Leipzig). It is a collection of texts, but it is not a parallel nor aligned corpus. The aim was to obtain a systematic corpus-based description of the distinctive linguistic structures of LS by contrasting it to similar approaches of text simplification (i.e. *einfache Sprache* and *Leicht Lesen*). The corpus contains 639,826 tokens of LS, 779,278 tokens of *Einfache Sprache*, and 350,872 tokens of *Leicht Lesen*.

Battisti and Ebling, 2019: this corpus was compiled to be used in automatic readability assessment and automatic text simplification (ATS) in German. It was compiled from web sources and consisted of both monolingual (LS) and parallel data (German and LS). It also contained information on text structure, typography and images; according to their authors, these features can indicate whether a text is simple or complex. The monolingual data consists of 1,916,045 tokens and the parallel data consists of 347,941 tokens of German and 246,405 tokens of LS. However, there was no sentence-alignment in this corpus.

TextComplexityDE (Naderi et al., 2019a): this dataset consists of 1000 sentences taken from 23 Wikipedia articles in 3 different article-genres. 250 of those sentences have also been manually simplified by native speakers. Besides, it also contains subjective assessment (complexity, understandability and lexical difficulty) of the simplified sentences, provided by a group of language learners of A and B levels. This dataset is aimed to be used for developing text-complexity predictor models and ATS.
APA (Säuberli et al., 2020): the Austria Presse Agentur (APA) corpus is the first parallel corpus for data driven ATS for German. It consists of 3,616 sentence pairs. The authors manually simplified original sentences into their A2 and B1 equivalents and aligned them.

LeiKo (Jablotschkin and Zinsmeister, 2020): it is a comparable corpus of LS news texts, systematically compiled and linguistically annotated for linguistic and computational linguistic research. It contains approximately 50,000 tokens, and is divided into four sub-corpora according to the websites from which they were extracted.

KED (Jach, 2020): Korpus Einfaches Deutsch (KED) is a collection of texts from genres of educational and public discourse in LS and *Einfache Sprache*, scraped from different online websites. It has a total of 3,698,372 words, and it is divided into different sub-corpora depending on the provider.

20m (Rios et al., 2021): it is a corpus collected from the Swiss news portal *20 Minuten*, which includes 18,305 articles paired with shortened summaries. There is no sentence-alignment in this corpus, and the dataset does not distinguish different simplification levels; they do not stick to any simplification standard.

Capito: *capito*[2] is the largest provider of human simplification services for German; they translate information into easy-to-understand language, offer trainings, and develop digital solutions around the topic of comprehensibility. It has a dataset that covers a wide range of topics and levels A1, A2 and B1 (Rios et al., 2021).

Geasy (Hansen-Schirra et al., 2021): the German Easy Language corpus (Geasy) is a parallel corpus, aligned at sentence level, which contains professional translations from standard German into LS. It currently contains 1,087,643 words of source text and 292,552 words of LS translations.

[2] https://www.capito.eu (last accessed: 2022-11-04)

Toborek et al., 2022: this is a new monolingual sentence-aligned corpus for German, LS and ES, spanning different topics. This corpus consists of publicly available articles of 7 different webpages that publish news articles in German and their corresponding LS version. They also included articles from a website in ES in an aim to achieve a larger vocabulary size. They refer to all simplified versions of German as *Simplified German*. The corpus has a total of 250,093 tokens of *Simplified German* and 404,771 tokens of German, contains 708 aligned documents and a total of 5,942 aligned sentences. The quality of the sentence alignments has a F1-score of 0.28.

SNIML (Hauser et al., 2022): simple news in many languages (SNIML) is a multilingual corpus of news in simplified language. It includes articles in Finnish, French, Italian, Swedish, English and German, published between 2003 and 2022, and originates from different news providers in different countries. It is a dataset of raw text. Besides, the level of simplification varies depending on the provider, that is, the texts have been created according to different simplification guidelines and for different target audiences. However, the authors claim that the corpus is useful for automatic readability assessment and for unsupervised, self-supervised or cross-lingual learning. They plan to release a new version of SNIML every month, and their future work may consist of aligning the articles to related articles in standard language. By the time this article is being written, it contains 4,936,181 tokens in total, 123,021 of which are of German.

Klexicon (Aumiller and Gertz, 2022): this is a document-aligned corpus by using the German children encyclopedia "Klexikon". It contains 2,898 articles from "Klexicon", with an average of 436.87 tokens each, and 2,898 documents from Wikipedia, with an average of 5,442.83 tokens each. The authors aligned the documents by choosing corresponding articles from Wikipedia; however, it is unlikely that specific sentences are matched.

3.2. Other Resources

3.2.1. Dictionaries

E2R texts may also include explanations of certain terms. E2R dictionaries can be useful for this end, as they provide translations from a standard language term into its E2R equivalent, or an explanation for complex words that cannot be adapted into an easier variant. German counts with *Hurraki*[3], a dictionary with explanations of German words in LS.

3.2.2. LS Language Checkers

They are employed to check texts for grammar and style mistakes. For E2R, they can help checking whether any E2R rule has been broken. Siegel and Lieske (Lieske and Siegel, 2014; Siegel and Lieske, 2015) implemented some EL rules in Acrolinx[4] and LanguageTool[5].

3.3. Tools for LS Adaptation

E2R adaptation refers to the processes that standard texts undergo in order to be transformed into E2R texts. These processes may include syntactic simplification (reducing the grammatical complexity of a text), lexical simplification (replacing complex words with easier variants) or summarization (conveying the most important information), among others.

EasyTalk (Steinmetz and Harbusch, 2020): EasyTalk is a system for assisted typing in LS. It uses a paraphrase generator based on a lexicalized, unification-based Performance Grammar.

SUMM[6]: this is the first AI-powered tool that automatically turns any text into EL. It is still only available in its Beta version, but its founders claim that it increases adaptation productivity by 85%. Once registered, users can directly go to the translation interface and adapt any text. Currently, the glossary is based on *Hurraki*, although users can also create their own glossary based on their texts.

3.4. LS Readability Assessment

Readability assessment is used to classify texts according to their degree of complexity; it determines how difficult or easy a text is or which level/grade it has (Bengoetxea and Gonzalez-Dios, 2021; Vajjala, 2022). It can help authors prepare simplified material, inform readers about the difficulty of a piece of text, or facilitate choosing of learning material for second language learners, among others (Aluisio et al., 2010).

Battisti et al., 2019: they present an unsupervised machine learning approach to analyse texts in simplified German in an aim to investigate evidence of multiple complexity levels. They also exploited structural and typographic characteristics of simplified texts. Their findings prove that there is not just one complexity level in German simplified texts.

Ebling et al., 2022: this is the first sentence-based NMT approach towards automatic simplification of German and the first multi-level simplification approach for German. Besides, this paper offers an overview of four parallel corpora of standard/simplified German, compiled and curated by their group. They report a gold standard of sentence alignments from these four sources.

Naderi et al., 2019b: in this study, they developed an automated readability assessment estimator based on supervised learning algorithms over German text corpora. They employed the TextComplexityDE corpus. They extracted 73 linguistic features and employed feature engineering approaches to select the most informative ones. They implemented 4 regression estimators to assess the readability of the sentences, among which Random Forest obtained the best result, with a 0,847 RMSE.

Mohtaj et al., 2022b: they present a new model for text complexity assessment for German text based on transfer learning. They used the TextComplexityDE dataset to train the models. Their findings show that fine-tuning the BERT model can outperform the other approaches.

Weiss and Meurers, 2022: this study presents a sentence-wise readability assessment model for German L2 readers. They built a machine learning model with

[3] https://hurraki.de/wiki/Hauptseite (last accessed: 2022-11-03)

[4] Acrolinx is a software package to support authors of technical documentation

[5] LanguageTool is an open source text checking software developed since 2003

[6] https://summ-ai.com/en/ (last accessed: 2022-11- 03)

linguistic insights and compare its performance based on predictive regression and sentence pair ranking. They found that it yielded top performances across tasks.

Blaneck et al., 2022: in this study, they combined the fine-tuned GBERT and GPT-2-Wechsel models with linguistic features. They evaluated their models in the GermEval 2022 Shared Task on Text Complexity Assessment with the TextComplexityDE dataset. The combined models performed better than non-combined GBERT or GPT-2-Wechsel models. On out-of-sample data, their best ensemble achieved a RMSE of 0.435.

Mosquera, 2022: this paper describes the winning approach in the first automated German text complexity assessment shared task as part of KONVENS 2022. The only resource provided by the organizers was the TextComplexityDE dataset. They followed two main approaches to train the dataset: feature engineering based on morphological and lexical information, and transfer learning via pre-trained transformers.

Mohtaj et al., 2022a: this paper offers an overview of the GermEval 2022 Shared Task on Text Complexity Assessment of German Text. Due to space constraints, not all studies that took part in it have been included in the paper at hand. However, this study offers an overview of the task and the approaches. Among 24 participants who registered for the shared task, ten teams submitted their results on the test data.

4. Conclusion

This paper presents the current situation of LS in Germany and offers an overview of the most important developments regarding resources and tools for LS adaptation. It can be stated that there is a dynamically developing research situation of LS in Germany. Nonetheless, as it has been highlighted, there are also different currents with regard to LS rules, terminology and its possible stigmatising effect. In spite of LS being the established language variant, there is a lack of consensus regarding the accuracy of its name and the users it is aimed at. This may lead to potentially remodeling LS rules and validating *Leichte Sprache Plus* in the near future (Lindholm and Vanhatalo, 2021, 209). It is possible that the concept of LS and E2R will be reshaped, and different names will be used to refer to them; however, to this day, it would be advisable to stick to the terms "Leichte Sprache" and "Easy-to-Read", since these are the established terms and the use of other terminology might cause confusion. Regarding the available resources and tools, it is worth highlighting that there is a recent interest in developing corpora. Nonetheless, it can be observed that there is no consistency within the databases; they might be parallel, comparable, aligned, and can include texts of different complexity levels. This might be due to different reasons: (1) they have been thought to serve for different purposes, or (2) there is not enough data to create the corpora from. Some webpages offer information both in standard German and E2R, but some others do not. This makes it difficult to create parallel, sentence-aligned corpora. Besides, many so-called E2R resources that can be found in many websites are often not linked to a single corresponding German document, but are high-level summaries of multiple German documents (Klaper et al., 2013). Many readability assessment methods have been developed recently; it would be helpful for LS and ES users to see these methods implemented, so that they can assess the difficulty of a given text. The existing LS adaptation tools may aid in the creation of LS texts; however, they are still not able to automatize the insertion of examples, explanations and illustrations or to create a proper layout for LS. Taking everything that has been said into account, we could say that EL and ATS have been a recurrent field of study in these recent years and seem to stay that way. However, much remains to be done, especially in terms of developing products based on all available tools and resources. These products would make information more accessible to people with communication disabilities.

References

Aluisio, Sandra, Lucia Specia, Caroline Gasperin, and Carolina Scarton, 2010. Readability Assessment for Text Simplification. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative use of NLP for Building Educational Applications*.

Aumiller, Dennis and Michael Gertz, 2022. Klexikon: A German Dataset for Joint Summarization and Simplification. *arXiv preprint arXiv:2201.07198*.

Battisti, Alessia and Sarah Ebling, 2019. A corpus for Automatic Readability Assessment and Text Simplification of German. *arXiv preprint arXiv:1909.09067*.

Battisti, Alessia, Sarah Ebling, and Martin Volk, 2019. An Empirical Analysis of Linguistic, Typographic, and Structural Features in Simplified German Texts.

Baumert, Andreas, 2016. Leichte Sprache-Einfache Sprache.

Baumert, Andreas, 2018. Einfache Sprache und Leichte Sprache.

Bengoetxea, Kepa and Itziar Gonzalez-Dios, 2021. MultiAzterTest: A Multilingual Analyzer on Multiple Levels of Language for Readability Assessment. *arXiv preprint arXiv:2109.04870*.

Blaneck, Patrick Gustav, Tobias Bornheim, Niklas Grieger, and Stephan Bialonski, 2022. Automatic Readability Assessment of German Sentences with Transformer Ensembles. In *Proceedings of the GermEval 2022 Workshop on Text Complexity Assessment of German Text*.

Bock, Bettina M, 2018. ‚Leichte Sprache'-Kein Regelwerk.

Bredel, Ursula and Christiane Maaß, 2016. *Leichte Sprache: Theoretische Grundlagen? Orientierung für die Praxis*. Bibliographisches Institut GmbH.

Ebling, Sarah, Alessia Battisti, Marek Kostrzewa, Dominik Pfütze, Annette Rios, Andreas Säuberli, and Nicolas Spring, 2022. Automatic Text Simplification for German. *Simple and Simplified Languages*.

Hansen-Schirra, Silvia, Walter Bisang, Arne Nagels, Silke Gutermuth, Julia Fuchs, Liv Borghardt, SILVAN DEILEN, Anne-Kathrin Gros, Laura Schiffl, and Johanna Sommer, 2020. Intralingual Translation into Easy Language–or How to Reduce Cognitive Processing Costs. *Easy Language Research: Text and User Perspectives. Berlin: Frank & Timme*:197–225.

Hansen-Schirra, Silvia and Christiane Maaß, 2020. Easy Language, Plain Language, Easy Language Plus: perspectives on comprehensibility and stigmatisation.

Easy Language Research: Text and User Perspectives, 2:17.

Hansen-Schirra, Silvia, Jean Nitzke, and Silke Gutermuth, 2021. An Intralingual Parallel Corpus of Translations into German Easy Language (Geasy Corpus): What Sentence Alignments Can Tell Us About Translation Strategies in Intralingual Translation. In *New Perspectives on Corpus Translation Studies*. Springer, pages 281–298.

Hauser, Renate, Jannis Vamvas, Sarah Ebling, and Martin Volk, 2022. A Multilingual Simplified Language News Corpus. In *2nd Workshop on Tools and Resources for REAding DIfficulties (READI)*.

Jablotschkin, Sarah and Heike Zinsmeister, 2020. LeiKo: A Corpus of Easy-to-Read German.

Jach, Daniel. Korpus Einfaches Deutsch (KED), https://daniel-jach.github.io/simple-german/simple-german.html (last accessed: 2022-09- 08).

Klaper, David, Sarah Ebling, and Martin Volk, 2013. Building a German/Simple German Parallel Corpus for Automatic Text Simplification.

Lange, Daisy, 2018. Comparing 'Leichte Sprache','einfache Sprache'and 'Leicht Lesen': A Corpus-Based Descriptive Approach. In *Eds. SUSANNE J. JEKAT, and GARY MASSEY. Barrier-free communication: methods and products: proceedings of the 1st Swiss conference on barrier-free communication. Winterthur: ZHAW Digital Collection*.

Lieske, Christian and Melanie Siegel, 2014. Verstehen Leicht Gemacht. *technische kommunikation*, 1:44–49.

Lindholm, Camilla and Ulla Vanhatalo, 2021. *Handbook of Easy Languages in Europe*. Frank & Timme.

Maaß, Christiane, 2015. *Leichte Sprache. Das Regelbuch*. Lit-Verlag.

Maaß, Christiane, 2019. Easy Language and Beyond: How to Maximize the Accessibility of Communication. Invited Plenary Speech at the Klaara 2019 Conference on Easy-to-Read Language Research (Helsinki, Finland. 19-20 September 2019).

Maaß, Christiane, 2020. *Easy Language–Plain Language–Easy Language Plus: Balancing comprehensibility and acceptability*. Frank & Timme.

Maaß, Christiane and Sergio Hernández Garrido, 2020. Easy and Plain Language in Audiovisual Translation. *Easy language research: Text and user perspectives*, 2:131.

Maaß, Christiane and Silvia Hansen-Schirra, 2022. Removing Barriers: Accessibility as the Primary Purpose and Main Goal of Translation. *Translation, Mediation and Accessibility for Linguistic Minorities*, 128:33.

Mohtaj, Salar, Babak Naderi, and Sebastian Möller,

2022a. Overview of the GermEval 2022 Shared Task on Text Complexity Assessment of German Text. In *Proceedings of the GermEval 2022 Workshop on Text Complexity Assessment of German Text*.

Mohtaj, Salar, Babak Naderi, Sebastian Möller, Faraz Maschhur, Chuyang Wu, and Max Reinhard, 2022b. A Transfer Learning Based Model for Text Readability Assessment in German. *arXiv preprint arXiv:2207.06265*.

Mosquera, Alejandro, 2022. Tackling Data Drift with Adversarial Validation: An Application for German Text Complexity Estimation. In *Proceedings of the GermEval 2022 Workshop on Text Complexity Assessment of German Text*.

Naderi, Babak, Salar Mohtaj, Kaspar Ensikat, and Sebastian Möller, 2019a. Subjective Assessment of Text Complexity: A Dataset for German Ganguage. *arXiv preprint arXiv:1904.07733*.

Naderi, Babak, Salar Mohtaj, Karan Karan, and Sebastian Möller, 2019b. Automated Text Readability Assessment for German Language:Aa Quality of Experience Approach. In *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE.

Nitzke, Jean, Silvia Hansen-Schirra, Ann-Kathrin Habig, and Silke Gutermuth, 2022. Translating Subtitles into Easy Language: First Considerations and Empirical Investigations. *Translation, Mediation and Accessibility for Linguistic Minorities*, 128:127.

Pottmann, Daniel M., 2019. Leichte Sprache and Einfache Sprache–German Plain Language and teaching DaF German as a Foreign Language. *Studia Linguistica*, 38:81–94.

Rios, Annette, Nicolas Spring, Tannon Kew, Marek Kostrzewa, Andreas Säuberli, Mathias Müller, and Sarah Ebling, 2021. A New Dataset and Efficient Baselines for Document-level Text Simplification in German. In *Proceedings of the Third Workshop on New Frontiers in Summarization*. Online and in Dominican Republic: Association for Computational Linguistics.

Säuberli, Andreas, Sarah Ebling, and Martin Volk, 2020. Benchmarking Data-driven Automatic Text Simplification for German. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI)*. Marseille, France: European Language Resources Association.

Siegel, Melanie and Christian Lieske, 2015. Beitrag der Sprachtechnologie zur Barrierefreiheit: Unterstützung für Leichte Sprache. *Zeitschrift für Translationswissenschaft und Fachkommunikation*, 8(1):40–78.

Steinmetz, Ina and Karin Harbusch, 2020. Enabling Fast and Correct Typing in 'Leichte Sprache' (Easy Language). In *Proceedings of the The Fourth Widening Natural Language Processing Workshop*.

Toborek, Vanessa, Moritz Busch, Malte Boßert, Pascal Welke, and Christian Bauckhage, 2022. A New Aligned Simple German Corpus. *arXiv preprint arXiv:2209.01106*.

United Nations, 2006. Convention on the Rights of Persons with Disabilities of the United Nations (CRPD). https://www.un.org/development/desa/disabilities/convention-on-the-rights-of-persons-with-disabilities/article-9- accessibility.html (last accessed: 2022-09-08).

Vajjala, Sowmya, 2022. Trends, Limitations and Open Challenges in Automatic Readability Assessment Research. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association.

Weiss, Zarah and Detmar Meurers, 2022. Assessing Sentence Readability for German Language Learners with broad Linguistic Modelling or Readability Formulas: When do Linguistic Insights make a Difference? In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*.

Vulgar Remarks Detection in Chittagonian Dialect of Bangla

Tanjim Mahmud[1], Michal Ptaszynski[1], Fumito Masui[1]

[1]Text Information Processing Laboratory, Kitami Institute of Technology, Kitami, Japan
tanjim_cse@yahoo.com, {michal,f-masui}@mail.kitami-it.ac.jp

Abstract

The negative effects of online bullying and harassment are increasing with Internet popularity, especially in social media. One solution is using natural language processing (NLP) and machine learning (ML) methods for the automatic detection of harmful remarks, but these methods are limited in low-resource languages like the Chittagonian dialect of Bangla. This study focuses on detecting vulgar remarks in social media using supervised ML and deep learning algorithms. Logistic Regression achieved promising accuracy (0.91) while simple RNN with Word2vec and fastTex had lower accuracy (0.84-0.90), highlighting the issue that NN algorithms require more data.

Keywords: EDO 2023, Vulgarity detection, Data annotation, Logistic Regression (LR), Recurrent Neural Network (RNN)

1. Introduction

The use of vulgar language, swearing, taboo words, or other offensive language is referred to as vulgarity or obscenity (Cachola et al., 2018; Wang, 2013). Although in society, vulgar language in conversation is frequent (Mehl et al., 2007), it has become very popular on social media sites like Twitter (Wang et al., 2014). Although vulgar language can be employed in a positive context, such as indicating informality of dialogue, communicating one's anger, or identifying with a group (Holgate et al., 2018), in reality, it is most often used in online harassment.

Therefore, in our study, we focused on detecting such vulgar expressions in the Chittagonian dialect of Bangla. Chittagonian is a language from the Indo-Aryan language family[1] with between 13 and 16 million speakers, the great majority of them living in Bangladesh (LewisM, 2009). Chittagonian is widely spoken alongside Bengali, to the extent that many linguists consider it a distinct language (Masica, 1993). It is a variation of Bangla, with distinct features in pronunciation, vocabulary, and grammar. Over the last two decades, Internet use in Bangladesh has grown exponentially. According to BTRC, there are well over 125 million Internet users in Bangladesh as of November 2022[2]. In addition, Chittagong is the second largest city of Bangladesh[3], and due to the Digital Bangladesh initiative[4], the majority of the population now has access to the Internet and is able to use social media. Moreover, with the benefit of Unicode on gadgets, they express their thoughts in their native Chittagonian dialect. Chittagonian people regularly use such social media as Facebook[5], imo[6], various blogs, WhatsApp[7], and others. On social networking sites, people feel free to express themselves in casual ways. However, due to its wide pervasiveness, it is difficult to escape also the negative influence of social media.

Excessive social media use has the potential to become addictive[8]. Young people spend more time on social media than they do with family and friends[9]. Social media use has been linked to cyberbullying and online abuse, which has an impact on self-esteem and can be a violation of one's privacy[10]. Social media has also been used to disseminate hatred and deception online, leading to an increase in violent incidents in society[11].

One of the realizations of such unwanted and harmful incidents is receiving messages filled with vulgar expressions. Moreover, with the increased usage of social media comes the increased probability of being exposed to such vulgar remarks.

To contribute to the mitigation of this problem, in this work, we propose a system capable of automatically recognizing such vulgar remarks. Using Logistic Regression (LR) and Recurrent Neural Networks (RNN) as a classifier, backed by various feature extraction methods, we test the limitations of such methods for vulgar remark detection in the low-resource language scenario.

The key contributions of this paper are as follows:

1. We collect a dataset in the Chittagonian dialect consisting of 2,500 comments or posts. The data was gathered purely from widely accessible accounts on the Facebook platform.
2. We manually and rigorously annotate the dataset

[1]https://en.wikipedia.org/wiki/Chittagonian_language
[2]http://www.btrc.gov.bd/site/page/ 347df7fe-409f-451e-a415-65b109a207f5/-
[3]https://en.wikipedia.org/wiki/Chittagong
[4]https://www.undp.org/bangladesh/blog/ digital-bangladesh-innovative-bangladesh-road-2041
[5]https://www.facebook.com
[6]https://imo.im
[7]https://www.whatsapp.com
[8]https://www.addictioncenter.com/drugs/social-media-addiction/
[9]https://en.prothomalo.com/bangladesh/Youth-spend-80-mins-a-day-in-internet-adda
[10]https://www.un.org/en/chronicle/article/cyberbullying-and-its-implications-human-rights
[11]https://www.accord.org.za/conflict-trends/social-media/

Contents

into vulgar and non-vulgar categories and validate the data annotation process by Cohen's Kappa statistics (Cohen, 1960).
3. Finally, we compare Machine Learning (ML)-based, and Deep Learning (DL)-based approaches for identifying vulgar remarks in the Chittagonian dialect on social media content.

The paper is organized as follows: Section 2. covers related works. Section 3. presents the proposed technique, including data collection and pre-processing. Section 4. discusses evaluation and analysis. Finally, Section 5. offers concluding remarks and future work.

2. Related works

Vulgar language in user-generated content on social media can lead to sexism, racism, hate speech, or other forms of online abuse (Cachola et al., 2018). Vulgarity detection, while typically solved by creating lexicons of vulgar expressions, can also be treated as a classification problem with two classes: vulgar and not. Traditional methods using vulgarity lexicons require constant updates. ML methods can use the surrounding context of vulgarities to classify new vulgarities without a lexicon. Few studies have approached vulgarity detection with methods beyond lexicon-based, so in this review, we also included studies from closely related domains.

Many linguistic and psychological studies have been conducted on the purposes and pragmatic goals of vulgar language(Andersson and Trudgill, 1990; Pinker, 2007; Wang, 2013). On the other hand, for machine learning-related studies, for example, (Eshan and Hasan, 2017) compared various ML algorithms with N-gram features to find out which algorithms perform better. With 2,500 comments they labeled their dataset into two classes. Finally, by SVM with trigram TF-IDF vectorizer features they got the highest accuracy of 0.89. (Akhter et al., 2018) suggested using machine learning methods and using user data to identify cyberbullying in the Bangla language. They applied NB, J48, SVM, and KNN, with the performance of each method evaluated using a 10-fold cross-validation. According to the results, SVM performed better in terms of Bangla text, with the highest accuracy of 0.9727. (Holgate et al., 2018) introduced a dataset of 7,800 tweets from users with known demographics, every incident containing vulgarities was categorized into one of the six categories of vulgar word use. They examined the pragmatic components of vulgarity and their relationships to societal issues using the data they collected, obtaining 0.674 of macro F1 over six classes. (Emon et al., 2019) created a system for detecting abusive Bengali text and applied different deep learning and machine learning-based algorithms. They collected 4,700 comments from Facebook, YouTube, Prothom Alo online and labeled this data into seven different classes. They got the highest accuracy of 0.82 with the Recurrent Neural Network algorithm. (Awal et al., 2018) proposed Naive Bayes system to detect abusive comments and collected 2,665 English comments from YouTube and then translated these English comments into Bengali in two ways, namely, i) Direct translation to Bangla, and ii) Dictionary-based translation to Bangla. Finally, their proposed system produced the highest accuracy of 0.8057.

For detecting abusive Bangla comments a technique developed by (Hussain and Al Mahmud, 2019) applied a root-level algorithm with unigram string features. They collected 300 comments from Facebook pages, news portals, and YouTube. They divided their data set into three sets with 100, 200, and 300 comments and tested their system with the result of 0.689 (accuracy) on average. (Das et al., 2022) studied hate speech detection in low-resourced language Bengali and Romanized Bengali. They collected their data from Twitter which contain 5071 Bengali samples, and 5107 Romanized Bengali samples. For training these datasets they used XML-RoBERTa, MuRIL, m-BERT, and IndicBERT models, from which XML-RoBERTa gives the best accuracy of 0.796. (Sazzed, 2021) manually separated 7,245 YouTube reviews into categories of vulgar and non-vulgar content to establish two benchmark corpora. The DL-based BiLSTM model produced the highest recall scores for detecting vulgarity in both datasets. (Faisal Ahmed et al., 2021) marked user comments from publicly visible Facebook postings made by sportsmen, public servants, and celebrities. Then, the English- and mixed-language comments were separated from the Bengali-language comments. Their research showed that 14,051 of all remarks, or 31.9% of the total, were directed at male victims, and 29,950 of all, or 68.1%, were directed at female victims. In this study, 9,375 comments were directed towards victims who were social influencers, followed by 2,633 comments directed at politicians, 2,061 comments directed at athletes, 2,981 comments directed at singers, and 61.25% of the remarks, or 26,951, directed at actors.

None of the preceding research particularly looked for vulgarities in the Chittagong dialect of Bengali. In the context of the data from the Chittagong dialect on social media, the following is the first effort to precisely identify and evaluate the frequency of vulgarity in social media posts.

3. Proposed methodology

The layout of the proposed system is demonstrated in Figure 1 and the process is explained as follows.

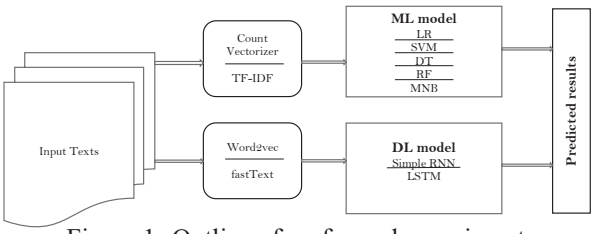


Figure 1: Outline of performed experiments

3.1. Data collection

Since there is no state-of-the-art dataset for the Chittagonian dialect to detect vulgar texts, one of the first major challenges in this study was collecting the data. We collected comments from Facebook. The comments were collected manually from several independent sources including public profiles and pages of famous people. Table 1 shows three examples from the dataset.

3.2. Data annotation

To detect vulgarity, a dataset needed to be annotated with a set of standards (Pradhan et al., 2020; Khan et al., 2021).

English	Translation
অডা তুই শুয়োরের বাচ্চা।	You are piglet
তোরা বেগ্‌গুন খানকির ফুয়া	You are all son of whores
বাংলাদেশত নতুন বেইশ্যাদেহা যার।	New prostitutes are appearing in Bangladesh.

Table 1: Three examples from dataset

We appointed three native Chittagonian dialect speakers, one of whom was a male with a higher education (MSc degree) and two of whom were women (BSc degree). The dataset contained 2,500 samples and each of them was manually annotated according to the process shown in Figure 2. Firstly, three annotators annotated each review which resulted in 7,500 judgments. Any disagreements about the annotation were resolved by majority voting of the annotators. Consequently, the raw dataset consisted of 1,009 vulgar samples, and 1,476 non-vulgar samples with 15 conflicts (discarded from the dataset after discussion with annotators). Some of the most frequently used vulgar words in our dataset were shown in Table 2. After annotating the data, we examined the inter-rater agreement. As a result, using Cohen's Kappa (Cohen, 1960), we obtained an average agreement value of 0.91, indicating a very strong agreement among annotators.

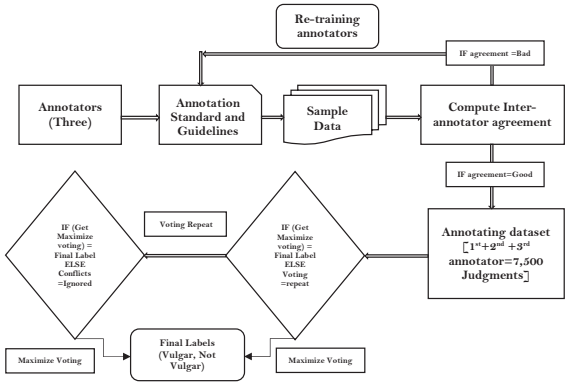


Figure 2: Data annotation process.

Chittagonian Dialect	English	Frequency
মাগির	Slut's	333
চোদা	Fucker	201
কুত্তা	Dog	192
খানকি	Whore	105
সোনা	Female private parts	75

Table 2: Top frequency vulgar words in our dataset.

3.3. Data preprocessing

Our collected dataset of text-based comments was not in fixed length and does not follow any specific structure, which means it could contain noise. The dataset comments could contain excessive data unimportant for analysis. To limit the possible influence of such unwanted redundant features, we processed the data. Table 3 shows each preprocessing step.

Original Text
ছুদানির ফুয়ার গলাধে গলা।1 No.(...)
English Translation (Literal)
The son of a slut's singing voice is very sweet.1 number (. .....)
Removing Functuations
ছুদানির ফুয়ার গলাধে গলা 1 No(...)
Removing Emojis and Emoticons
ছুদানির ফুয়ার গলাধে গলা 1 No
Removing English Characters
ছুদানির ফুয়ার গলাধে গলা 1
Removing English Digits
ছুদানির ফুয়ার গলাধে গলা
Removing Stopwords
ছুদানির ফুয়ার গলাধে গলা
Tokenizations
['ছুদানির',' ফুয়ার',' গলাধে',' গলা']

Table 3: Step-by-step data preprocessing.

3.4. Feature engineering

Since ML algorithms cannot take as input textual data directly, it is necessary to convert lexical features (words) into numerical features in order to be able to extract patterns and perform classification. Since after preprocessing our dataset contains only strings of characters (words), we transformed the textual data into numerical features. To do this, following previous studies (Emon et al., 2019; Sazzed, 2021; Mahmud et al., 2022) we applied four different feature extraction techniques, namely, Count Vectorizer, TF-IDF Vectorizer, Word2vec, and fastText' in order to obtain the features applicable in classification.

3.5. Classification

We experimented with five machine learning and two deep learning algorithms. Specifically, we used Logistics Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and Multinomial Naive Bayes (MNB) for classic ML as well as simple Recurrent Neural Network (simpleRNN) and Long-Short-Term Memory network (LSTM).

3.6. Performance evaluation metrics

Model evaluation is the process of validating the model performance on the test data. In this study, we used the following model performance evaluation metrics, namely, Precision, Recall, F1-score, and Accuracy, which are calculated on the numbers of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN), according to the following formulas.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad \ldots\ldots\ldots\ldots\ldots\ldots(1)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3)$$

$$\text{F1 score} = 2 * \left(\frac{Precision * Recall}{Precision + Recall}\right) \quad \ldots\ldots\ldots\ldots(4)$$

Model	Vulgar			Non Vulgar			Accuracy
	Precision	Recall	F1-score	Precision	Recall	F1-score	
LR	0.80	0.92	0.86	0.91	0.76	0.83	0.91
SVM	0.65	0.72	0.68	0.68	0.60	0.63	0.66
DT	0.62	0.86	0.72	0.77	0.47	0.58	0.67
RF	0.67	0.94	0.79	0.90	0.53	0.67	0.87
MNB	0.81	0.91	0.86	0.90	0.79	0.84	0.84

Table 4: Model performance using count vectorizer

Model	Vulgar			Non-Vulgar			Accuracy
	Precision	Recall	F1-score	Precision	Recall	F1-score	
LR	0.82	0.92	0.87	0.90	0.80	0.85	0.91
SVM	0.81	0.89	0.85	0.88	0.79	0.83	0.84
DT	0.56	0.96	0.71	0.85	0.21	0.34	0.67
RF	0.64	0.97	0.77	0.94	0.45	0.61	0.88
MNB	0.80	0.91	0.85	0.89	0.77	0.83	0.83

Table 5: Model performance using TF-IDF.

4. Experimental results and discussion

4.1 Machine learning models for vulgarity detection

Below are the results of classic ML algorithms applied in vulgarity detection in the Chittagonian language, specifically, LR, SVM, DT, RF, and MNB using Count Vectorizer. Count Vectorizer is a process of converting textual features into numerical values, specifically into numbers representing a basic frequency of words in the text. To train and test the models we divided the dataset in an 80-20 ratio for training and testing sets, respectively, as represented in Table 6. The results achieved by ML models using this feature extraction technique were represented in Table 4.

Training	Testing
80%	20%
1,988	497

Table 6: Training and test data ratio

Table 4 shows the overall performance of machine learning models using the Count Vectorizer feature extraction technique. Here, the LR model outperforms other models with the highest accuracy of 0.91. The class-wise performance also achieved higher precision and recall values for both classes. The second-best model was RF, which achieved 0.87 but in terms of precision, and recall MNB also performed well compared to LR.

On the contrary, TF-IDF stands for the Term Frequency-Inverse Document Frequency of records. It assesses the word's relevance within a corpus or a dataset. The frequency of a term in the corpus represents how many times it appears in the text. The highest score was also achieved by LR, with an accuracy of 0.91, and precision, recall and F1-score also scored high for both classes. MNB achieved 0.83 of accuracy, less than RF but in terms of other metrics, the results were well balanced, as shown in Table 5.

4.2 Deep learning models for vulgarity detection

Deep Learning-based models like RNN, LSTM, or GRU have shown great achievements in various NLP tasks in recent years. In this study, we used RNN and LSTM with Word2vec and fastText used for generating word embeddings. We divided the dataset into three parts, namely, train, validate (to check for overfitting), and test (to evaluate the model) as shown in Table 7.

Using Word2vec word embeddings SimpleRNN outperformed the LSTM model achieving 0.84 accuracy while LSTM achieved 0.63. In terms of precision and recall for both classes, SimpleRNN also provided better results, as represented in Table 8.

Training	Validation	Testing
70%	15%	15%
1,741	372	372

Table 7: Training, validation, and test data ratio.

Also, both applied Deep Learning models showed better performance using fastText word embedding technique. Here also SimpleRNN acquired 0.90 of accuracy and more importantly models performed quite well in detecting both classes as shown in Table 8.

5. Conclusion and future work

In this study, we focused on the detection of vulgar remarks in social media posts using ML and DL classifiers, namely LR, SVM, DT, RF, MNB, simpleRNN, and LSTM with various feature extraction techniques such as Count Vectorizer, TF-IDF Vectorizer, Word2vec, and fastText. We have constructed a dataset of 2,485 comments where vulgar and non-vulgar were evenly distributed. In our study, LR with Count Vectorizer, or TF-IDF Vectorizer, as well as simpleRNN with Word2vec and fastText were an effective approach for detecting vulgar remarks. Based on the performed study, in the future, we plan to pursue a resource-constrained strategy for recognizing vulgarity, mostly focusing on the Chittagonian dialect. As in this study for the classification we used only the simplest baseline methods, we will apply other more robust methods, such as

bidirectional RNNs(Schuster and Paliwal, 1997), and transformers (Aurpa et al., 2022).

Word2vec	Vulgar			Non Vulgar			Accuracy
	Precision	Recall	F1-score	Precision	Recall	F1-score	
SimpleRNN	0.78	0.98	0.86	0.97	0.70	0.81	0.84
LSTM	0.61	0.81	0.70	0.68	0.45	0.54	0.63
fastText	Vulgar			Non Vulgar			Accuracy
	Precision	Recall	F1-score	Precision	Recall	F1-score	
SimpleRNN	0.94	0.87	0.90	0.87	0.94	0.90	0.90
LSTM	0.63	0.89	0.74	0.79	0.45	0.57	0.68

Table 8: Model performance using Word2vec and fastTex

References

Akhter, S. (2018, December). Social media bullying detection using machine learning on Bangla text. In *2018 10th International Conference on Electrical and Computer Engineering (ICECE)* (pp. 385-388). IEEE.

Andersson, L. G., & Trudgill, P. (1990). Bad language. *(No Title)*.

Aurpa, T. T., Sadik, R., & Ahmed, M. S. (2022). Abusive Bangla comments detection on Facebook using transformer-based deep learning models. *Social Network Analysis and Mining*, *12*(1), 24.

Awal, M. A., Rahman, M. S., & Rabbi, J. (2018, October). Detecting abusive comments in discussion threads using Naïve Bayes. In *2018 International Conference on Innovations in Science, Engineering and Technology (ICISET)* (pp. 163-167). IEEE.

Cachola, I., Holgate, E., Preoţiuc-Pietro, D., & Li, J. J. (2018, August). Expressively vulgar: The socio-dynamics of vulgarity and its effects on sentiment analysis in social media. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 2927-2938).

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, *20*(1), 37-46.

Das, M., Banerjee, S., Saha, P., & Mukherjee, A. (2022). Hate Speech and Offensive Language Detection in Bengali. *arXiv preprint arXiv:2210.03479*.

Emon, E. A., Rahman, S., Banarjee, J., Das, A. K., & Mittra, T. (2019, June). A deep learning approach to detect abusive Bengali text. In *2019 7th International Conference on Smart Computing & Communications (ICSCC)* (pp. 1-5). IEEE.

Eshan, S. C., & Hasan, M. S. (2017, December). An application of machine learning to detect abusive Bengali text. In *2017 20th International conference of computer and information technology (ICCIT)* (pp. 1-6). IEEE.

Ahmed, M. F., Mahmud, Z., Biash, Z. T., Ryen, A. A. N., Hossain, A., & Ashraf, F. B. (2021). Bangla text dataset and exploratory analysis for online harassment detection. *arXiv preprint arXiv:2102.02478*.

Holgate, E., Cachola, I., Preoţiuc-Pietro, D., & Li, J. J. (2018). Why swear? analyzing and inferring the intentions of vulgar expressions. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 4405-4414).

Hussain, M. G., & Al Mahmud, T. (2019). A technique for perceiving abusive Bangla comments. *Green University of Bangladesh Journal of Science and Engineering*, 11-18.

Khan, M. M., Shahzad, K., & Malik, M. K. (2021). Hate speech detection in roman Urdu. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, *20*(1), 1-19.

LewisM, P. (2009). Ethnologue: Languages of the world, Sixteenth Edition.

Mahmud, T., Das, S., Ptaszynski, M., Hossain, M. S., Andersson, K., & Barua, K. (2022, October). Reason Based Machine Learning Approach to Detect Bangla Abusive Social Media Comments. In *Intelligent Computing & Optimization: Proceedings of the 5th International Conference on Intelligent Computing and Optimization 2022 (ICO2022)* (pp. 489-498). Cham: Springer International Publishing.

Masica, C. P. (1993). *The indo-aryan languages*. Cambridge University Press.

Mehl, M. R., Vazire, S., Ramírez-Esparza, N., Slatcher, R. B., & Pennebaker, J. W. (2007). Are women really more talkative than men?. *Science*, *317*(5834), 82-82.

Pinker, S. (2007). *The stuff of thought: Language as a window into human nature*. Penguin.

Pradhan, R., Chaturvedi, A., Tripathi, A., & Sharma, D. K. (2020). A review on offensive language detection. *Advances in Data and Information Sciences: Proceedings of ICDIS 2019*, 433-439.

Sazzed, S. (2021). Identifying vulgarity in Bengali social media textual content. *PeerJ Computer Science*, *7*, e665.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673-2681.

Wang, N. (2013). An analysis of the pragmatic functions of "swearing" in interpersonal talk. *In: Griffith Working Papers in Pragmatics and Intercultural Communication*, *6*, 71-79.

Wang, W., Chen, L., Thirunarayan, K., & Sheth, A. P. (2014, February). Cursing in English on Twitter. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing* (pp. 415-425).

Aspect Based Sentiment Analysis by Morphological Features of The Kazakh Language

Madina Mansurova[1], Nurgali Kadyrbek[1], Talshyn Sarsembayeva[1]

[1]Al-Farabi Kazakh National University, Almaty, Kazakhstan
mansurova.madina@gmail.com
nurgaliqadyrbek@gmail.com
sarsembayeva.talshyn@gmail.com

Abstract

The article deals with the approaches to the analysis of the sentiment of texts in relation to a given object as well as its characteristics (aspects). The article analyzes approaches to the automatic determination of the sentiment of text data, investigates deep learning methods taking into account the combination of lexical-semantic units (lemmas) and morphological features without dividing aspect terms into categories to determine the author's attitude to aspects/attributes of the sentiment analysis object. This study was conducted for the Kazakh language which is an agglutinative language, the morphological features of which play a dominant role in determining its semantics: it is possible to determine the aspects of the model limited by the level of morphology without syntactic-semantic analysis of the sentence components. The factors influencing the quality and accuracy of identification and analysis of the text sentiment are determined and classified. A comparative analysis of the investigation results is given trained taking into account morphological features and trained only taking into account lemmas, as well as a description of the experimental results of the study on the influence of morphological features on identification of aspects.

Keywords: aspect approach, sentiment analysis, morphological features, aspect terms, neural networks.

1. Introduction

An automatic analysis of the text sentiment, i.e., determination of the author's opinion on the subject area being discussed in the text, is one of the most actively developing directions in the field of processing a natural language in the recent decades. The actuality of this direction is conditioned by the increase in the influence of social networks in marketing and economics, a fast distribution of online services and online stores. The possibility of monitoring and consideration of the user and consumer opinions on different products and services, the use of methods for analyzing the sentiment of reviews and for analyzing opinions allow to develop and successfully use recommendation systems and decision support systems. The role of automatic analysis of the sentiment of user opinions in social networks in the course of political and social investigations increases, for example, when determining political priorities, predicting election results (Vepsalainen et al 2017 and Vilares et al. 2015), determining the public attitude to various political decisions.

The ways of gaining information contained in the comments depend on the genres of the texts being analyzed. For example, one of the most studied genres in sentiment analysis is user reviews of products or services. Though in such reviews the author expresses a subjective opinion on a definite object, the text may contain useful information on many properties (features) of the object under study, and analysis of the opinion of a great number of users allows to make general conclusions.

Aspect based sentiment analysis is a method of sentiment analysis directed to determination of the attitude of the subject of opinion to component/attribute of the object of opinion separately.

The goal of the work is to determine the attitude of the author to aspects/attributes of the object of sentiment analysis taking into account lexico- semantic and morphological features of the Kazakh language.

The investigation methods are methods of deep learning taking into account combinations of lexical-semantic units (lemmas) and morphological features without dividing aspect terms into categories.

2. Aspect terms

In the course of analysis, aspects may be grouped into categories, for example, for the phone: operating system, interface, functional, design, etc. Besides, in the text, one can come across a generalized assessment of an object: excellent smartphone. This category can be considered as an aspect (for example, whole objects). The words that refer to the content of the aspect are called aspect terms. Types of aspect terms: explicit, implicit, evaluative facts (Gupta 2013), resource problem (Liu and Zhang 2012).

From the point of view of an explicit aspect, any component of the opinion object is named and specified directly in the context; the attribute in which the word moves can be easily found. Explicit aspects are usually represented by nouns or groups of nouns, indefinite forms of the verb. Explicit aspects can be distributed the dependent forms: e.g. speaker phone, headphones, power supply, etc. ("telefonnyn dinamig-i, qulakkab-y, kuattalu-y").

Implicit aspect terms combine emotional connotation and aspect into one term, i.e., reading the term, we can say what it is about and what kind of emotional assessment it carries, for example, the term "light" refers to the weight in the context of smartphone, generally (ideally) the weight of the phone is light ("telefonnyn salmagy jenil"). In our study, priority is given to explicit aspect terms: if in the context there are words corresponding to two terms, the explicit aspect term is flagged.

Evaluative facts usually describe a technical process (often related to defects). For example, breakdown, stagnation, freezing, etc. Such evaluative facts often occur as words that are not included in the vocabulary of emotional assessment with a positive or negative meaning.
The terms related to the problems of resources are often used with "many-few" qualifiers and verbs in the sense of use (necessity, use etc). For example, consumes too much energy uses too much traffic.
Aspect terms in the subject area can be classified according to several criteria.
The most frequent type of aspect terms are aspect terms that clearly indicate the object, its parts or characteristics, e.g., display in smartphone comments, battery, etc. Evident (explicit) aspect terms often consist of nouns or substantiated (reified) nouns and verbs.
In the comments concerning smartphones, one of the problems related to the semantics of the text is the presence of various connotations related to the context,
e.g.: “akkumuliator uzzak zariadtalady” (the battery takes a long time to charge), “akkumuliator uzzak olmeidi” (the battery does not discharge for a long time).
Also, simple words that do not appear in the assessment vocabulary can contain an emotional connotation in a definite context, especially verbs: e.g. “azhyratu” (disconnect), “zhondeu” (repair), etc.
The task of aspect-based sentiment analysis includes the following actions:
- identification of aspect terms,
- classification of terms by aspects,
- automatic identification of aspects related to the highlighted categories.
There are four main ways of automatic extraction of aspect terms from the text (Liu and Zhang 2012):
1. The approach based on the frequency of nouns and groups of nouns (Ku et al.2006).
2. The approach that uses the interrelationship between assessment expressions and aspect terms (Blair-Goldensohn 2008)
3. The approach based on machine learning with a teacher (Yukun 2018).
4. The approach based on statistical thematic models (Titov 2008)

3. Preparing data for training

To carry out the experiments, we have chosen the subject area "smartphone analysis" taking 2000 user reviews of smartphones. As is shown in figure 1, the "sentiment" column is written in compliance with the sentiment of the opinion, the text of the "text"-comment, the "mark" form - aspect terms in the given context. The aspects are chosen on the basic of properties and morphological features considered in the above section (Aspect terms).

sentiment	text	mark
0	Батареясы біраз қолданған кейін істен шығып кетіп жатады және ұзақ уақытқа шыдамсыз	батареясы
0	Камерасы әлсіз мегапикселін жаңарту қажет сурет анық емес	камерасы сурет
0	Не деген сұмдық жады аз неліктен коробкасында 16гбт жазылып тұр бірақ менеджер фай	жады
0	телефон салмағы не деген ауыр темірден жасаған ба? Жеңілдетулеріңізді сұраймын біра	телефон салмағы
0	телефонда түнгі режим жоқ қолтаңбасы әлсіз бекер алдым бұл телефонды	режим қолтаңбасы

Fig. 1: Comments and aspects

In the course of morphological analysis, for each token we determined the following metainformation:
- lemma or a morpheme close to it (stemma):
- part of speech;
- affixes (if present): plural ending, possessive ending, case ending, personal ending, participle and adverbial participle suffixes, indefinite form of the verb, negativity.

The process of data preparation is shown in Figure 2. If we take into account the fact that sentiment analysis can be considered as a separate case of semantic analysis, it becomes clear why Word Embedding (Word2Vec) is used here.

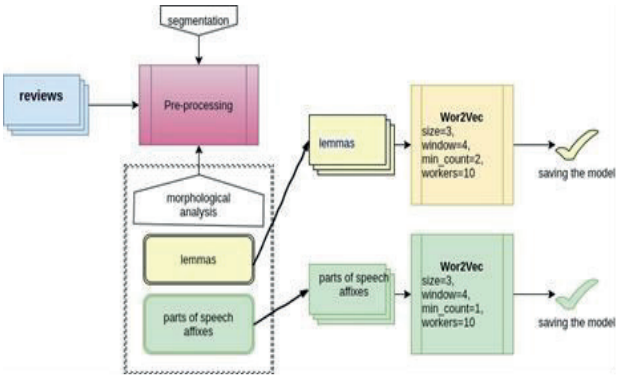


Fig. 2: Data preparation stage

The process of data preparation is shown in Figure 2. If we take into account the fact that sentiment analysis can be considered as a separate case of semantic analysis, it becomes clear why Word Embedding (Word2Vec) is used here.

Affixes play an important role in determination of semantic/syntactic relations between words in case grammar. Description of the full role structure of se n-tences in the Kazakh language, valency and categories of verbs, the semantic structure of arguments is a too complex and labour consuming task. However, we can obtain empirical data on the basis of the case grammar concept. We studied Word2Vec models independently using each category, collecting lemmas (or morphemes close to them) as a separate document and morphological components as a separate document and obtained a point in a three-dimensional space for each lemma, each morphological feature. In the end of the process presented in Figure 2, we have two vector models: for lemmas and morphological features. Now we add to the vector corresponding to the lemma one or more vectors) corresponding to the morphological sign to get the final vector for each token.

Thus, the sum of vectors for each lemma and morphological features in accordance with the units found in the token was used to determine the final vector for training LSTM model and predicting aspects. To illustrate it, let us consider an example (Figure 3) where:

vector1: kuaty (power) – kuat (noun)+y(pp)
vector2: saktal – saktal (verb)
vector3: saktaluy (conservation) – saktal (verb)+u(infinitive)+ y(pp)
vector4: saktalyp– saktal (verb)+ yp (participle).

The reason for the choice of this example is to show the degree owing to which we can compare the vector of an implicit aspect “Saktaluy” (conservation) with the vector of an explicit aspect “kuaty” (power) under the influence of morphological features. In their nature, verbs are not always aspect terms but they can be present in the form of an infinitive and/or with a personal ending indicating the

attribute of the object of emotional criticism. This description can be not only a physical unit but also a functional (the name of an action), e.g. “smartphonnyn zariadtaluy ~ smartphon akkumu-liatorynyn zariadtaluy” (smartphone charging ~ smartphone battery charging).

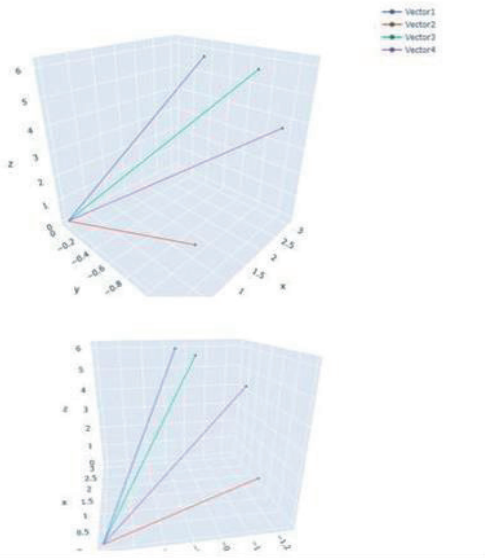


Fig. 3: The relationship between morphological elements and lemma semantics

Let us note once again: vectors in Figure 3 are not abstract images, but vectors based on Word2Vec model studied in the course of the work.

If aspect terms in the input data are negative, they are marked by the value of 1, if positive - by the value of 2, in the other cases - by the value of 0, e.g.: for the text “dinamikterinin dauysy salystyrganda tomen bolyp keledi. dauys katty shukpaidy” (the sound of the speakers is comparatively lower; the sound does not turn on loudly).

We used the model of recurrent neural network LSTM the architecture of which is presented in Figure 4. The length of the sentence (the number of tokens) used for learning and predicting makes up 40 units, each unit is presented by a three- dimensional vector. In the process of training, 40% of input data were used for validation.

tokens	dinamikterinin	dauysy	salystyrganda	tomen	bolyp	keledi	.	dauys	katty	shukpaidy	<EOS>	...	<EOS>
Vectors												...	
target			0	0	0	0	0	1	0	0	0	...	0

Fig. 4: LSTM neural network architecture

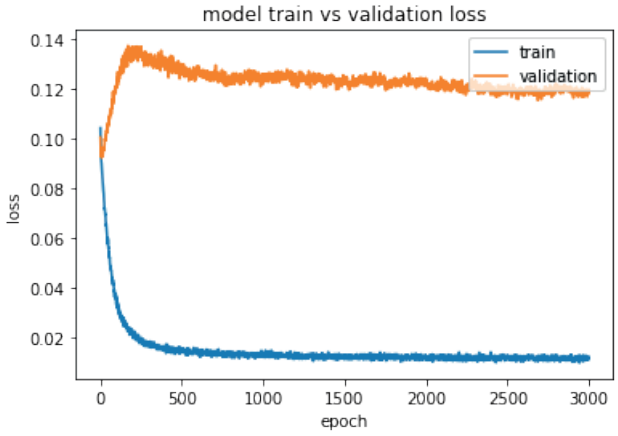
As is seen, Exponential linear unit (ELU) was used as a function of activation and RMSProp was used as an optimizer, the model itself was trained over 3000 epochs.

In the process of training, Dropout layers were used to reduce the neuronal saturation degree. In machine learning, regularization is a way of preventing retooling. Regularization reduces retooling by adding penalty to the loss function. With addition of regularization the model is trained in such a way that it does not study the interdependent set of feature weights. Dropout is an approach to regularization in neural networks, that helps to reduce interdependent learning among neurons.

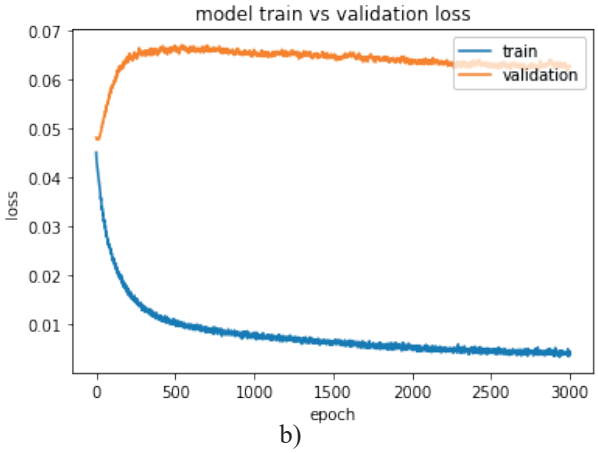
For the vectorization experiment, two different methods were used:

1) only lemmas and their vectors (Fig. 5, a);

2) the final vector obtained by adding the vector of morphological features to the vector of lemmas (Fig. 5, b).



a)



b)

Fig. 5: Loss rate during model training and validation

As is seen in Figure 5, when testing the model, the approach with the use of morphological features showed a more stable result achieved by presenting aspect terms using morphological features.

4. Using of LSTM networks in sentiment analysis of documents in kazakh language

Due to the growing confidence in information in social media resources, interest in the field of sentimental analysis is growing. Because sentiment analysis is one of the main technologies for monitoring the opinions of millions of users of social networks.

The article discusses the use of LSTM networks in the sentimental analysis of texts in the Kazakh language. 1000 mobile phone user reviews were used to train the neural network. The research work was considered in two ways: the comments received were pre-processed and not pre-processed. Here, the accuracy of a model built using the LSTM architecture reached 80%. This figure is 11% higher than a model trained with previously unprocessed data. As a result of the research, the use of pre-processed reviews showed good results. Currently, due to the rapid development of information technologies, opinions on social networks are often used to evaluate the market, determine the popularity and degree of a specific product, service, show business, sport, and even political positions. Such comments can be positive, negative or neutral. Determining which group these opinions belong to is carried out using sentiment analysis, a branch of computer linguistics. Coherence analysis - determining the coherence of opinions using natural language processing (NLP) methods, statistics, machine learning. At the same time, coherence analysis is used to identify spam in comments, analyze the usefulness of comments, and search for comparisons. As a rule, the comments received from social networks do not follow the rules of grammar, various signs, abbreviations, etc. may be. Therefore, in such a case, pre-processing of data allows to achieve good results (Hemalatha et al. 2012, Muhammad and Shahid. 2018). However, in this research, we will compare the results of both cases with pre-processed and non-pre-processed feedback.

In recent years, neural networks have been widely revived as powerful models of machine learning, showing excellent results in areas such as video recognition and natural language processing (Sarsembayeva et al.2021).

Bag of words, along with classifiers using traditional models such as the Bayesian method, have been used rationally to obtain highly accurate predictions in coherence analysis problems (Narayan et al 2013). With the emergence of deep learning technologies and their application in natural language processing, there is an opportunity to improve the accuracy of these methods in two main directions: data preprocessing and the use of trained and untrained neural networks in training clusterers and classifiers.

A total of 1000 opinions of mobile phone users were used to train the neural network during the research. In the collection, each opinion is stored in the structure "class:opinion", where 0 is positive and 1 is negative opinion (Fig.6).

	Class	Data
0	0	камерасы әлсіз мегапикселін жаңарту қажет зам...
1	0	не деген сұмдық жады аз неліктен коробкасында...
2	0	телефон не деген ауыр темірден жасаған ба жең...

Fig.6: Classification of opinions

For example: the lemma for the token "phones" is "phone". Here, as an analyzer tool, the tool developed during our project was used (Wang et al. 2018). In the course of work, we experiment with and without using lemmatization. The main difference between regular neural networks and recurrent networks is the time-dependent aspect of recurrent networks. In recurrent grids, each word input sequence is associated with a specific time step.

Each time step h_t is associated with a new component called a hidden state vector. From its highest level, this vector seeks to encapsulate and accumulate all the information observed in previous time steps. Therefore, x_t is a vector containing all the information related to a specific word, and h_t is a vector that accumulates information from previous time steps. The hidden state is a function of the current word vector as well as the hidden state vector from the previous time step. Sigma indicates that the sum of the two terms is fitted by an activation function (usually sigmoid or tangent).

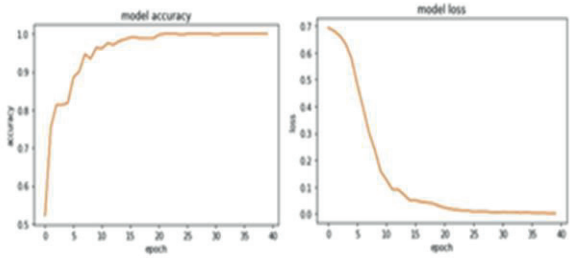
$$h_t = \sigma(W^H h_{t-1} + W^X x_t) \quad (1)$$

The members of W are weight matrices. The input vector is multiplied by the weight matrix W^X, and the hidden state vector at the previous time step is multiplied by the recurrence weight matrix W^H. W^H is a matrix that remains the same for all time steps, while the measurement matrix W^X is different for each input signal. These weight matrices affect either the current or previous hidden state of the hidden state vector.

Each element takes x_t and h_t-*1* (not shown in the figure) as input data and performs some calculations to obtain intermediate states. Each intermediate state arrives at a different network and eventually the information is aggregated to form ht. Here, each element plays its own role: the input element determines how much attention to pay to each input, the forget element determines the information we discard, and the output element determines the final ht based on an intermediate state.

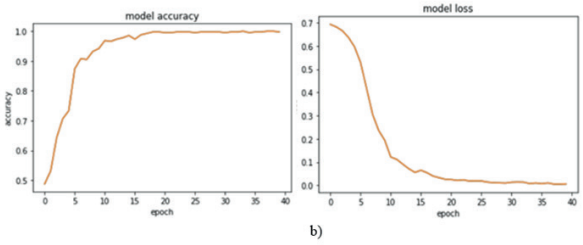
In our model trained by LSTM, the batch size, that is, the number of comments to be trained, is 100, and the number of epochs is 40. The softmax function was used as the activation function. This is because the network uses categorical crossentropy and softmax is the optimal solution for us.

The experiment was carried out using two different methods. Without pretreatment (Fig.7a) and with pretreatment (Fig.7b). The pictures show the progress of neural network training. In the first case, the training accuracy is depicted, and in the second, the training error.



a)

a)



b)

b)

Fig.7: a) Learning with normalization b) Learning without normalization

Without pre-processing				With pre-processing		
	precision	recall	f1-score	precision	recall	f1-score
negative	0.68	0.77	0.72	0.81	0.80	0.80
positive	0.71	0.62	0.66	0.78	0.80	0.79
accuracy	0.69			0.80		

Table 1. Evaluation of results.

5. Conclusion

Aspect based sentiment analysis is one of the new tendencies in the field of sentiment analysis. This direction allows to analyze more accurately and distribute the emotional connotation expressed by the subject at the level of detalization of the object attribute. Taking into account the fact that for the Kazakh language that is an agglutinative language, the morphological features play a dominating role in determination of its semantics, it is possible to determine the aspects of the model limited by the level of morphology without a syntactic-semantic analysis of the sentence components. During vectorization of words and morphological features it is better to use interrelated vectors, not numerical identifiers. In this case it would be helpful to use a vector by depicting semantic relations as in Word2Vec. As a result of investigations, the model trained taking into account morphological features showed a more stable result than the model trained taking into account only lemmas. The first model showed the accuracy of 92%, the latter one had the accuracy of about 87%. We found experimentally that morphological features exert a positive influence on identification of aspects.

Syntax analysis is a fundamental problem in computational linguistics. Since the Kazakh language is among languages with few resources, research in this direction requires a lot of work. In the considered work, a harmony analysis was performed on the processed text with the help of the morphological analyzer created within the framework of the project. Here, the accuracy of the model built using the LSTM architecture has reached 80%. This figure is 11% higher than the model trained on unnormalized data. On the one hand, this can be explained by the lack of data used for training. In turn, through normalization, the compactness of the model and the ability to generalize many data are achieved.

Acknowledgment. This work was funded by Committee of Science of Republic of Kazakhstan AR09261344 "Development of methods for automatic extraction of geospatial objects from heterogeneous sources for information support of geographic information systems" (2021-2023).

References

Vepsalainen T., Li H., Suomi R. (2017) *Facebook likes and public opinion: Predicting the 2015 Finnish parliamentary elections* //Government Information Quarterly.

Vilares D., Thelwall M., Alonso M. A. (2015) *The megaphone of the people? Spanish SentiStrength for realtime analysis of political tweets* //Journal of Information Science. 41. №. 6. P. 799-813.

Gupta N. K. (2013) *Extracting phrases describing problems with products and services from twitter messages* // Computacion y Sistemas. V. 17, №2. P. 197–206

Liu B., Zhang L. (2012) *A survey of opinion mining and sentiment analysis* // Mining Text Data. Springer: US, P. 415–463.

Ku Lun-Wei, Yu-Ting Liang, Hsin-Hsi Chen (2006). *Opinion extraction, summarization and tracking in news and blog corpora* // Proceedings of AAAI-CAAW'06.

Blair-Goldensohn S., Hannan K., McDonald R., Neylon T., Reis G. A., Reynar J. (2008) *Building a sentiment summarizer for local service reviews* // Proceedings of WWW Workshop on NLP in the Information Explosion.

Yukun Ma, Haiyun Peng, Erik Cambria. (2018) *Targeted Aspect-Based Sentiment Analysis via Embedding Commonsense Knowledge into an Attentive LSTM* Conference: AAAI At: New Orleans

Titov I., McDonald R. (2008) *A joint model of text and aspect ratings for sentiment summarization* // Urbana, 51, 61801.

I.Hemalatha, G. P. Saradhi Varma, A.Govardhan (2012) *Preprocessing the Informal Text for efficient Sentiment Analysis* // International Journal of Emerging Trends & Technology in Computer Science (IJETTCS). Volume 1, Issue 2– P. 58–61.

Muhammad Javed, Shahid Kamal (2018) *Normalization of Unstructured and Informal Text in Sentiment Analysis* // International Journal of Advanced Computer Science and Applications // (IJACSA), Vol. 9, No. 10, 2018. – P. 78–85.

Sarsembayeva, T., Mansurova, M., Chikibayeva, D., Karymsakova, D. (2021) *The Problem of Named Entities Unification based on Geographical Ontologies* // 2020 IEEE 8th Workshop on Advances in Information, Electronic and Electrical Engineering, AIEEE 2020 - Proceedings

Narayanan V., Arora I., Bhatia A. (2013) *Fast and accurate sentiment classification using an enhanced Naive Bayes model* // International Conference on Intelligent Data Engineering and Automated Learning. – P. 194–201.

Jenq-Haur Wang, Ting-Wei Liu, Xiong Luo, Long Wang (2018) *An LSTM Approach to Short Text Sentiment Classification with Word Embeddings* // The 2018 Conference on Computational Linguistics and Speech Processing ROCLING, - P. 214-223.

Evaluation of Foreign Accent Prosody in L2 English Using CNNs

Hansjörg Mixdorff[1] and Roberto Togneri[2]

[1]Berliner Hochschule für Technik, Germany
hmixdorff@bht-berlin.de
[2]University of Western Australia, Perth, Australia
roberto.togneri@uwa.edu.au

Abstract

In the context of computer-aided pronunciation training (CAPT) automatic speech recognition (ASR) is an important component. ASR is traditionally based on Hidden-Markov-Models and mainly centered on the segmental aspects of L2 pronunciation. However, also prosodic deviations are indicative of foreign accent. In the current study we examine whether new recognition technologies such as convolutional neural networks which appear to surpass HMMs in other ASR domains, such as large vocabulary recognition, are feasible for rating, inter alia, accented speech in pre-existing corpora of English. To this effect we trained neural networks with a corpus of English accents, some of them native, some Mandarin or Cantonese, and tested their performance in terms of accent separation and accent evaluation. With respect to the former it appears as if chunk-sized units work equally well as syllables, especially when fewer accent types are compared. As regards accent evaluation on unseen data, the network output only confirms tendencies as long as the network categories themselves are not based on proficiency levels, but only on types of accent.

Keywords: Computer Aided Pronunciation Training (CAPT), Foreign Accent, Prosody, Deep Neural Networks

1. Introduction

Computer-Aided Language Learning facilitates individualized language learning when a human teacher is not available, as a computer-based language trainer is always accessible and indefatigable. However, in fact, providing useful and robust feedback on learner errors based on the speech signal is far from being a solved problem (see, for instance, Eskenazi, M., 2009). Still ASR has been successfully employed to detect errors on the phoneme level (see, for instance, Cucchiarini et al., 2014), but the correction of prosodic errors appears more problematic (compare, for example, Busà, 2008). This mirrors the fact that traditionally foreign accent has mostly been associated with segmental features and therefore pronunciation training has concentrated on this area. Nevertheless, prosodic differences are certain to contribute to foreign accent as well, especially with respect to fluency and intelligibility, as argues, for instance, Hahn, 2004).

Considerable research has addressed the usability of ASR technology in L2 pronunciation work (for an overview, see Cucchiarini and Strik, 2017). Early studies primarily addressed pronunciation assessment and showed relatively strong correlations between human pronunciation ratings and machine scores (Cucchiarini et al., 2000, Neumeyer et al., 2000, Witt and Young, 2000). More recent speech technology research has developed and investigated ASR-based measures for pronunciation error detection (Qian et al., 2012, Strik et al., 2009, van Doremalen et al., 2013). Although most CAPT systems use Hidden Markov Models (HMMs) to deal with the temporal variability of speech, more recently, convolutional neural networks (CNNs) that have many hidden layers and are trained using novel methods have been shown to outperform HMMs on a variety of speech recognition benchmarks, sometimes by a large margin (Hinton, Geoffrey, et al., 2012) and also applied successfully to mispronunciation detection (Hu et al., 2015). For problems related to accent detection and recognition, deep neural networks can capture the essential information that is pertinent to accents thereby automatically filtering out or normalizing against the effects of the acoustic environment and speaker characteristics (Jiao et al., 2016).

In the current study we aim to examine the applicability of CNNs to the task of supra-segmental foreign accent scoring by training them with corpora of different foreign accents. We hope to determine whether *accent recognition scores* can be exploited to evaluate the prosodic nativeness of a given speech utterance. In other words, we train the network on two or more accent categories (native/non-native) and evaluate to what extent the probabilities calculated for these accents in the classification reflect the prosodic proficiency or fluency of the talker. Instead of building a sophisticated model of what constitutes the foreign accent, we aim to develop a purely data-driven approach which nonetheless could flag conspicuous deviations in a learner utterance to be integrated in a CAPT system. In order to yield rather the probability of each of the accents in question than a hard decision for just a single accent type we use a *softmax* cross entropy criterion in the output layer.

2. Speech Data, Acoustic Features and CNN structure

In earlier works the first author and his co-workers performed prosodic studies on foreign accented English (Mixdorff and Ingram, 2009, Mixdorff and Munro, 2013, Munro et al., 2016) with native languages being Russian, Mandarin and Cantonese. After considering several databases of English accents, we decided to employ the Speech Accent Archive of George Mason University (Weinberger, 2015). Although it only contains a fairly short phonetically balanced reading passage with duration

Contents

of 20+ seconds, this passage was produced by 2696 speakers with different native languages including Russian, Cantonese and Mandarin, but also native speakers of US American and Canadian English. For a limited number of speakers segmentations are available which concern the phone and phrase levels. On examination we realized that the phone segmentations were not always reliable. Therefore we first segmented the speaker-wise wave files containing the complete reading passage into phrases based on the phrase segmentations. Then we performed syllable-level forced alignment on the phrase wave files with an HTK-based (Young et al., 2015) American English alignment system trained with WSJ data. The resulting syllable units were then used for our first CNN training experiments. To ensure a balanced size of input for all five classes of accent, several additional speakers had to be manually segmented on the phrase level. We initially extracted data by 30 speakers of Canadian English, Cantonese, Mandarin, Russian and US American English accents. Their data amounts to 10.7, 13.2, 15.1, 14.7 and 10.5 min. of speech, respectively. In our learning experiments we considered the whole paragraph, individual phrases, syllables, disyllables and chunks of constant length. In preliminary training experiments with paragraphs and phrases we observed that the network did not converge. The problem is that the input pattern is very long, and at the same time the amount of training exemplars relatively small. Syllable-sized units showed much better convergence. However, since duration is not uniform across instances, shorter syllable patterns have to be padded with zeros. We also used disyllabic units which yielded somewhat better results than syllables. Finally, we extracted chunks of constant length, typically 550ms taken at a step of 150ms.

We used 20 MFCC coefficients calculated on a narrow band FFT spectrum, but for a window of 150 ms. The reason for using MFCCs for prosody scoring is based on their great robustness against noise. They also have been proven applicable to prosodic tasks like tone recognition (Tangwongsan et al., 2004, Ryant et al. 2014).

Our idea here is that the network should only pick up the longer term supra-segmental prosodic structures, not the segmental level. 150ms are close to mean syllabic durations. Fig.1 shows an example of a chunk of 550 ms with spectral representations (standard *Praat* spectrogram, spectrogram with window length=150ms/step=10ms, MFCC sequence, 20 coefficients with window length=150ms/step=10ms). The structure of the convolutional neural network adopted from Kumar Mandal (2017) features a 2D convolutional input layer employing 32 filters, kernel size of 2x2 and a relu activation function, operating on *number of feature vectors x 4 or 20* nodes, respectively – with the number of feature vectors depending on the step size. The convolutional layer is followed by max pooling with pooling size of 2x2, a dropout layer with a rate of 0.25 and a flatten layer. The next layer is densely connected with an output space of 128, followed by another dropout layer and a softmax output layer to yield probabilities in the range between 0 and 1 with five nodes for the five different accents *Canadian English (CD), Cantonese (CA), Mandarin (MA), Russian (RU), US-English (US)*. Subsequently we evaluated the probabilities given at the output nodes as a correlate of accent classification.

The neural network technology employed is based on the Python packages *Keras* and *Tensorflow* with *Librosa* as the library for audio file manipulation. The corpus was divided into 70% for training and 30% for testing, with training and testing sets containing *different speakers*. The system was programmed and tested inside the IDE *Eclipse*. Training was performed until the accuracy on the testing data did not improve any further.

3. Results of Experiments

Syllable-sized Units

As mentioned before, the first experiments investigated a network developed with a total of 30 speakers for each of the five accent categories (21 for training, and 9 for testing). We expected Canadian and US-American accents to be highly confusable and also anticipated similarities between Mandarin and Cantonese, both being closely related tone languages. The units investigated were syllables yielded by applying a forced alignment system to the phrase-size units. We compiled the resulting wave files into *numpy* archive files for the five accents. Since the input layer size of the network is fixed, we also fixed the *mfcc* vector sequence to have a maximum length of 35. Shorter sequences were padded with zeros. As a consequence, longer tokens – especially from Cantonese and Mandarin – had to be excluded. However, the amount of padding also captures in a sense the different speech rates in the accent classes.

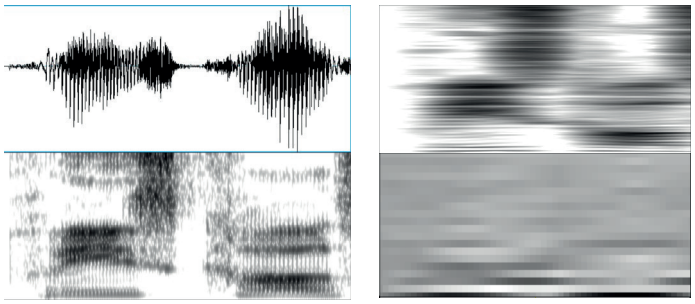


Fig.1: Example of chunk of duration 550 ms. Top left: Speech wave form, bottom left: Praat standard spectrogram, frequency range 0-8000Hz with window=5ms, step=2ms, top right: spectrogram with window=50ms, step=10 ms, bottom right: 20 MFCC coefficients with window=150ms, step=10 ms.

As expected, we found relatively high confusions between Canadian and US-American, but only a small overlap between Mandarin and Cantonese (see Table 1, train and test sets combined, MFCC features). If we accept these confusions as valid results - as one could argue that human listeners will also have a hard time distinguishing within the two pairs of utterances – we yield a recognition rate of about 70%. Pooling all the results for the foreign speakers also gives an interesting metric as to how native-like they sound. Although only anecdotal evidence could be gathered by listening to speakers with different recognition rates and we lacked perceptual ratings for the L2 speakers, there seemed to be a certain tendency for foreigners with lower node activation for their native accent to sound more

fluent and native-like than those with very high activation. This already points to a difficulty with respect to the best selection of speakers for forming a good data representation of their underlying accents. The more proficient learners become, the less typical their accent may sound. Therefore it seems advisable to incorporate speakers with a relatively strong accent for the network to form a more typical accent representation. In the absence of perceptual judgments another idea that we developed was to bootstrap the representation by building a core network which was then applied to hitherto unseen speakers as a rating device. Speakers with low accent recognition rates can then be exchanged for speakers with higher rates.

	recognized					
target	CD	CA	MA	RU	US	total
CD	1716	86	98	41	1167	3108
CA	157	1623	122	46	387	2335
MA	72	57	1633	50	536	2348
RU	140	53	266	1276	598	2333
US	144	46	72	40	3036	3338
total	2229	1865	2191	1453	5724	13462

Table 1: Confusion Matrix – syllable units. Results on train and test sets pooled. Abbreviations for native languages: Canadian English - CD, Cantonese-CA, Mandarin-MA, Russian-RU, US-English-US.

With respect to the longitudinal data gathered in XXZ, a study in which immigrants to Canada were surveyed one, six and nine years after their arrival to the country, the L2 learner groups (Mandarin and Slavic/Russian) were clearly different between themselves and from their Canadian English controls as to the network output node activations (Table 2). However, the residence time in years was only weakly negatively correlated with the respective accent's output node (Pearson's $r=-0.228$, $p<.001$ for speakers of Mandarin and $r=-0.054$, $p < .002$, for Slavic speakers). It must be stressed, however, that the three sub-groups do not contain the same speakers due to attrition from one up to nine years.

Finally, we look at the proportion correct as a function of the underlying syllable. Table 3 lists the ten syllables with the highest recognition rates, the one on the right the worst recognized. As can be seen, the best differentiated are all nouns, whereas the worst are mostly function words and unaccented syllables. This indicates that the network identifies accents best on syllables that are not reduced.

Chunks of equal duration

With the experience of the syllable-based units we wanted to investigate whether similar or better results could be achieved employing chunks of uniform length taken at a constant step from the original recording. The advantage of such an approach is that in the context of an unsupervised CAPT application one would not have to rely on the accuracy of the automatic syllable unit segmentation. It is also a common practice in accent classification tasks (Jiao et al., 2016). Furthermore, a constant chunk size also means a constant number of acoustic feature vectors which does not require zero padding.

As for the speaker data, we initially used the same speakers as with the syllable data, that is, only the speakers for which segmentations were available. However, subsequently we tested all remaining speakers with the resulting chunk-based network and identified those for which the network output the highest probabilities of belonging to their expected classes.

	recognized					
target	CD	CA	MA	RU	US	total
CD	879	348	241	243	988	2699
MA	489	529	808	332	783	2941
RU	612	427	657	462	1118	3276
total	1980	1304	1706	1037	2889	8916

Table 2: Confusion matrix-longitudinal data depending on the speaker group, all years pooled.

syllable	mean	SD	N
SLABS	.833	.202	176
SNAKE	.827	.198	178
CHEESE	.814	.204	177
BAGS	.804	.213	176
SNACK	.799	.217	178
SPOONS	.778	.223	173
THINGS	.764	.222	350
FRESH	.762	.223	177
PEAS	.757	.225	177
SMALL	.757	.223	176

syllable	mean	SD	N
TION	.622	.249	178
TO	.617	.215	352
OF	.609	.224	353
BE	.584	.218	178
WE	.581	.201	353
AT	.581	.222	178
WITH	.581	.220	178
A	.579	.210	534
THE	.572	.203	534
THEIR	.554	.204	178

Table 3: Ten syllables with the highest recognition rates (left), and the lowest (right).

This was done because we assumed that they were somewhat typical for the respective accents. The resulting speaker groups ("Best 30") for all five classes were employed for the ensuing experiments. The pre-processing removed silent pauses from the beginning and end of the paragraph sound files and examined the chunks for pauses contained. Such portions were not included in the training and test corpora. We also performed experiments with different chunk sizes, as well as window and step sizes for the calculation of MFCC vectors. Eventually we selected 550 ms as the chunk size – corresponding to approximately three syllables, 125ms for the window size and 30ms for the step. On the five-class problem we yielded typical accuracies of 73%.

In the final experiment we aimed to investigate whether some of the suboptimal results on the five-class network were to do with the small amount of data for the Canadian English and the Cantonese speakers which we had combined with 30 speakers for the other subgroups to yield roughly equal subsamples. We collected the maximum total of data for Mandarin in the GMU corpus of 115 speakers and combined them with 115 speakers of US English and the total amount of 76 speakers of Russian for chunk-sizes of 550ms. Table 4 lists the proportion correct depending on the target class as well as the difference between training and testing data. The lower accuracy for Russian might be due to the smaller number of speakers in the set. This result suggests that unseen speakers are more

poorly recognized than seen ones. However, if we concentrate only on the Mandarin speakers, there are four speakers in the test set that made it into the "Top 20" of speakers identified as having a Mandarin accent. Since the test set contains 30% of the speakers, the theoretical number would be six.

target	test set?	mean	SD	N
MA	no	.8812	.32353	10154
	yes	.6761	.46802	3736
	total	.8261	.37907	13890
RU	no	.6275	.48351	6821
	yes	.2547	.43581	2579
	total	.5252	.49939	9400
US	no	.8201	.38415	7909
	yes	.5458	.49797	3349
	total	.7385	.43947	11258

Table 4: Proportion correct depending on the target class as well as the difference between training and testing data, three-class problem. N is the number of chunks of duration 550ms.

Finally, we trained a two-class model with 115 US American English and Mandarin speakers each, thus exhausting the maximum number of Mandarin speakers available in the data base. The following table lists the proportion correct depending on L1 as well as the distinction between training and test data. Once again the training increases the probability of accent recognition, though less for Mandarin than for US English. If we look at the top 20 of speakers in the two-class model being identified as Mandarin it contains seven speakers who were not in the training set, one more than the expected 30%.

Courtesy of Ming Tu (Tu et al., 2018) we had perceptual ratings, but only for 30 of the Mandarin speakers. Though a Pearson's r of .15 between node activation "Mandarin" and the perceptual ratings (higher for less proficient speakers) points in the right direction, that result is not significant.

Table 5: *Proportion correct depending on the target class as well as the difference between training and testing data, two-class problem.*

target	test set?	mean	N	SD
MA	no	.8962	10154	.30502
	yes	.7768	3736	.41647
	total	.8641	13890	.34272
US	no	.7706	7909	.42045
	yes	.5626	3349	.49615
	total	.7087	11258	.45436

We listened in on some of the speakers of Mandarin with low "MA" scores. All of them were very fluent, though not necessarily with a native accent on the segmental level. On the other end of the spectrum, some US English speakers had very low "US" scores. One sounded more like a Russian and had major disfluencies, another one sounded native, but had several repairs in her reading. Other factors for low "US" scores we identified were poor recording quality and considerable background noise. In order to visualize the probability ratings alongside the audio file we wrote a program which collects all results for individual chunks for a speaker sound file and converts them to *Praat TextGrid* point tier format which can then be displayed with the audio data. Fig. 2 shows an example from speaker *mandarin27* with results for the two-class network (MA/US). The points mark the center of each chunk of 550ms. For a chunk in the middle of the figure with a relatively high probability "Mandarin" of 0.8 the extension of the underlying chunk is indicated by the selection in pink. This kind of display allows us to identify portions in the speech signal with conspicuous values. Speaker *mandarin27* yielded a relatively low average probability "Mandarin", but as can be seen, this does not apply to all his chunks.

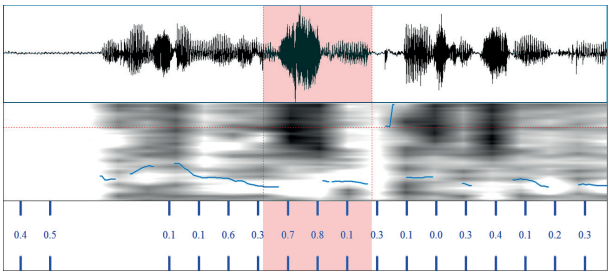


Fig. 2: Probability ratings associated with an utterance by Mandarin speaker 27.

4. Discussion and Conclusions

In this paper we presented results from a study employing a convolutional neural network for identifying accented Englishes. The idea is to employ scores from the network as a metric for evaluating the prosodic nativeness of a given speaker in CAPT system. Our results suggest that the accent recognition accuracy does not deteriorate when we move from syllable-sized units to units of uniform duration. Application of five-accent networks trained with the GMU accent corpus to accented speech from earlier studies only yielded limited success, due to considerable differences between the speech materials. It also must be taken into account that there are vast differences in fluency not only within the group of foreign speakers, but also in the groups of "native" speakers.

It might well be case that the network, besides performing some measurement of fluency also latches onto certain peculiarities in the recordings. Ideally speaking we would require perceptually rated data from L2 learners of a wider variety of proficiency levels. For the L1 group, a selection needs to be performed as to the suitability of the speakers, avoiding speakers with poor readings skills or other peculiarities who do not lend themselves as role models for L2 learning. In conclusion, the approach examined here appears promising. Besides the data quality and quantity problem discussed above, there are many variables that could be modified from different acoustic feature sets to varying the structure of the neural networks in terms of number and type of hidden layers. Although uniform chunks seemed to yield better results than syllables, it must be stated that the syllable segmentations were not manually corrected and it is possible, that zero-padding had an adverse effect. As the chunk-sized units can be back-traced to where they were taken from the master wave files, it is possible to examine areas of low

probability to see, what their acoustic properties are. As a first measure it seems desirable to have all speakers in the GMU corpus subset examined for acoustic quality, as well as reading performance.

5. Acknowledgements

This work was supported by DFG international travel grant Mi 625/30 to Mixdorff, funding a visit to the University of Western Australia. Thanks go to Steven H. Weinberger (George Mason University) for providing the accented database and additional materials, and to Ming Tu (Arizona State University) for supplying perceptual ratings of Mandarin speakers in the GMU accent corpus.

References

Boersma, Paul (2001). Praat, a system for doing phonetics by computer. *Glot International* **5:9/10**, 341-345.

Busà, M.G. (2008). Teaching prosody to Italian learners of English: Working towards a new approach. In: Taylor, C., Ecolingua: The Role of E-corpora in Translation, Language Learning and Testing. Trieste: EUT - Edizioni Università di Trieste, 113-126.

Cucchiarini, C., & Strik, H. (2017). Automatic speech recognition for second language pronunciation assessment and training. In O. Kang, R. I. Thomson & M. J. Murphy (Eds.) *The Routledge handbook of English pronunciation.*

Cucchiarini, C., Strik, H. & Boves, L. (2000a). Different aspects of expert pronunciation quality ratings and their relation to scores produced by speech recognition algorithm. *Speech Communication, 30*(2-3), 109-119. 10.1016/S0167-6393(99)00040-0

Cucchiarini, C., Nejjari, W. and Strik, H. (2014). My Pronunciation Coach: Computer-assisted English Pronunciation Training. In: Rias van den Doel & Laura Rupp (Eds.) Pronunciation Matters. Accents of English in the Netherlands and elsewhere. Amsterdam: VU University Press; pp. 45-68.

Eclipse: https://www.eclipse.org/downloads/

Eskenazi, M. (2009). An overview of spoken language technology for education. *Speech Communication*, 51(10): S. 832–844.

Hahn, L. (2004). Primary stress and intelligibility: Research to motivate the teaching of suprasegmentals. *TESOL Quarterly*, 38: 201-223.

Hinton, Geoffrey, et al. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups." Signal Processing Magazine, IEEE 29.6 (2012): 82-97.

Hu, W., Qian, Y., Soong, F.K., Wang, Y. (2015). Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers. *Speech Communication, 67*, 154-166. doi: 10.1016/j.specom.2014.12.008

Jiao, Yishan & Tu, Ming & Berisha, Visar & Liss, Julie. (2016). Accent Identification by Combining Deep Neural Networks and Recurrent Neural Networks Trained on Long and Short Term Features. 2388-2392. 10.21437/Interspeech.2016-1148.

Keras: https://keras.io/

Kumar Mandal, blog.manash.io, 5/8/2018.

Librosa: https://pypi.org/project/librosa/

Mixdorff, H. and Ingram, J. (2009). Prosodic Analysis of Foreign-Accented English. In Proceedings of Interspeech 2009, Brighton, England.

Mixdorff, H. and Munro, M. (2013). Quantifying and Evaluating the Impact of Prosodic Differences of Foreign-Accented English. Proceedings of Slate 2013, Grenoble, France.

Munro, M., Hansjörg Mixdorff and Tracy Derwing (2016). Longitudinal Acquisition of Prosodic Phenomena in L2 English. New Sounds 2016, Aarhus, Denmark.

Neumeyer, L., Franco, H., Digalakis, V. & Weintraub, M. (2000). Automatic scoring of pronunciation quality. *Speech Communication, 30*(2), 83–93.

Numpy: https://www.numpy.org/

Qian, X., Meng, H., Soong, F. (2012). The Use of DBN-HMMs for Mispronunciation Detection and Diagnosis in L2 English to Support Computer-Aided Pronunciation Training. *Proceedings of Interspeech 2012* (pp. 775-778), Portland, Oregon, USA.

Ryant, N., Slaney, M., Liberman, M., Shriberg, E., Yuan, J. (2014) Highly Accurate Mandarin Tone Classification In The Absence of Pitch Information. Proc. 7th International Conference on Speech Prosody 2014, 673-677.

Strik, H., Truong, K., de Wet, F., & Cucchiarini, C. (2009). Comparing different approaches for automatic pronunciation error detection. *Speech Communication, 51*(10), 845-852. doi: 10.1016/j.specom.2009.05.007

Tangwongsan, S, P. Po-Aramsri and R. Phoophuangpairoj, Highly Efficient and Effective Techniques for Thai Syllable Speech Recognition, in *Advances in Computer Science - {ASIAN} 2004, Higher-Level Decision Making, M.J. Maher (Editor)* 9th Asian Computing Science Conference, Chiang Mai, Thailand, 2004, pp. 259--270, 2004.

Tensorflow: https://www.tensorflow.org

Tu, M., Grabek, A., Liss, J., Berisha, V. (2018) Investigating the Role of L1 in Automatic Pronunciation Evaluation of L2 Speech. Proceedings Interspeech 2018, Hyderabad, India, 1636-1640.

Van Doremalen, J., Cucchiarini, C., & Strik, H. (2013). Automatic pronunciation error detection in non-native speech: the case of vowel errors in Dutch. *Journal of the Acoustical Society of America, 134,* 1336-1347. doi: 10.1121/1.4813304

Weinberger, S.(2015). *Speech Accent Archive*. George Mason University. http://accent.gmu.edu/ , accessed on 27/9/2018

Witt S. & Young, S. (2000). Phone-level pronunciation scoring and assessment for interactive language learning. *Speech Communication, 30*(2/3): 95-108. doi:10.1016/S0167-6393(99)00044-8

Young, Steve & Evermann, Gunnar & Gales, M.J.F. & Hain, Thomas & Kershaw, Dan & Liu, Xunying & Moore, Gareth & Odell, James & Ollason, Dave & Povey, Daniel & Ragni, Anton& Valtchev, Valtcho & Woodland, Philip & Zhang, Chao. (2015). The HTK Book (version 3.5a).

Experiments on error detection in morphological annotation

Emese K. Molnár[1,2], Andrea Dömötör[1,3]

[1]National Laboratory for Digital Heritage (Budapest)
[2]ELTE Doctoral School of Linguistics (Budapest)
[3]PPCU Doctoral School of Linguistics (Budapest)
mesii@student.elte.hu, domotor.andrea@btk.elte.hu

Abstract

The goal of our research was to reduce the amount of human annotation in our gold standard corpus project. In this paper we examine three methods (beside a baseline) which are language-independent, simple to implement, and suitable for finding annotation errors. The main goal is high recall: we are looking for a method that narrows down the range of tokens to be checked by a second annotator while covers as many errors as possible.The results of the three methods combined showed that by revising 29,96% of the tokens it is possible to reduce annotation errors to 1%. This means a significant decrease in human workload with the preservation of high annotation quality.

Keywords: annotation, morphology, gold standard, error detection

1. Introduction

A basic requirement for natural language processing is the availability of large and high-quality corpora that can be used as training and testing data for machine learning. However preparing a manually annotated and curated corpus requires a large amount of time and resource. Therefore our research investigates the possibilities to produce high quality annotations with less human resources. Based on the classic gold standard method every text should be annotated by (at least) two annotators and a third person curates them. But due to the fact that nowadays' automatic annotation tools are inherently quite accurate, we believe that manual annotation can be reduced.

In this paper we examine three methods (beside a baseline) which are language-independent, simple to implement, and suitable for finding annotation errors. The main goal is high recall: we are looking for a method that narrows down the range of tokens to be checked by a second annotator while covers as many errors as possible.

As we aim to produce a gold standard corpus we consider the first-round manual annotation necessary. Therefore our project focuses on finding the potentional errors in the annotation of the first annotators rather than detecting errors in the outputs of automatic text processing tools.

2. Motivation

Our research is motivated by a Hungarian gold standard corpus project. The aim of the project is to build a multi-level linguistically annotated reference corpus that will be suitable for both data-driven linguistic research, and as training and test data for natural language processing. It will contain morphological, syntactic (dependency), complex sentence, and noun phrase (NP) annotations. In the current phase we are working on the morphological annotations.

Currently the corpus contains ∼350.000 tokens with morphological annotations made by one annotator. The texts are from blogs, educational and cultural web sites, and 20th century Hungarian fiction. In the next phase of the workflow we plan to add academic, press and technical genres.

For the morphological annotation we used the tagset of emMorph (Váradi et al., 2018). The texts were preprocessed by the emtsv pipeline (Indig et al., 2019) and the annotators corrected the output of the pipeline manually. The emMorph annotation formalism contains the lemma, the POS-tag and the detailed morphological analysis of the token. The detailed analysis represents each morph of the token with their respective morphological features. This information lacks from the MSD annotation scheme of Universal Dependencies but can be relevant in morphologically rich languages such as Hungarian. However in this paper we focus only on the error detection in lemmatization and POS-tagging because the examined methods are not suitable for the detailed analysis.

The annotators corrected all components of the emtsv output using a self developed, Java-based annotation tool. The annotation tool is designed for the correction of the tokenization and morphological analysis. The tokens are displayed as buttons and by clicking on them the user can select from several options. They can merge a token with the previous or following token, split a token at any character, or rewrite a token if necessary (in case of typos).

The tools's interface displays all possible analyses of the selected token provided by the emtsv pipeline (Figure 1). The listed analysis options include the lemma, the detailed and the simple morphological analysis (POS-tag) of the token. A green tick indicates the analysis that was selected as correct by the POS-tagger module of emtsv. If the user disagrees with the analysis suggested by emtsv, it can be changed by clicking on another analysis. If the user does not agree with any of the offered analyses, they can provide a completely new analysis by filling in the blank fields of the last row. Unchecked tokens are displayed with red background, making the annotator's job easier. The an-

notation tool can request a new analysis for a token from emtsv anytime when the token or the sentence is changed in the course of the correction of tokenization. This feature of the tool makes the workflow much simpler, since tokenization and morphological annotation can be performed in one step.

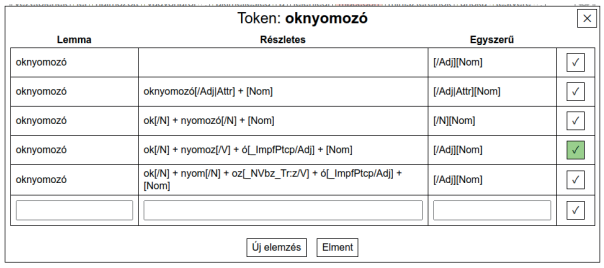


Figure 1: Annotation selection on the annotation tool

In order to make the annotation process faster, we have created a list of unambiguous words (i.e. those that can be analyzed only in one way in each case) based on Szeged Treebank (Vincze et al., 2010). The list was verified by linguists. After preprocessing, the annotation interface displays the words from the list as already checked tokens, so the annotator has no work to do with them. The list currently contains 31952 words, which usually covers one third of the texts.

The corpus is currently in XML format, but a simplified TSV format has been created for our error detection experiments. In the future we plan to publish the corpus in CoNLL-U+ format as well.

3. Methods

The main source of inspiration for our methods was the chapter titled *Iterative Enhancement* of the *Handbook of Linguistic Annotation* by Ide–Pustejovsky (Dickinson and Tufis, 2017). The chapter overviews several techniques for finding annotation errors, both in complete and in-progress corpora. The authors state that "most methods have not been tested with actual annotators" which is an additional motivation for our research.

Our corpus is an in-progress work therefore we examined methods that can be used with a relatively small set of annotated texts. Apart from an intuitive baseline we tested three methods. Our main goal is to decrease the number of annotations to revise (error candidates) while still cover major part of the errors. Therefore, our focus was achieving high recall rather than high precision.

3.1. Baseline: ambigous words and hapaxes as error candidates

The concept of our baseline method is that if a word occurs in the corpus several times with the same annotation than the annotation is probably correct. According to this concept every word that occurs in the corpus with different annotations is considered an error candidate. In addition, hapaxes should be considered error canidates too because in the case of these we cannot measure the consistency of the annotation. With this method we are likely to have many error candidates but will probably achieve high recall.

3.1.1. Human vs. machine annotations 1: human vs. the preprocessing tool

The handbook mentions several experiments based on the comparison of human and machine annotations. The main idea behind these methods is that the automatic text processing tools are trained to consistent behaviour therefore the differences between human and machine annotations may reveal inconsistencies in the human annotation.

In our first related experiment we compared the human annotations with the output of the emtsv pipeline that we used for preprocessing. This method relies on the high quality of the automatic text processing tool and presumes that most of the mistakes of the human annotation comes from the cases where the annotator changed the automatic annotation.

3.2. Human vs. machine annotations 2: human vs. an independent tool

In our other human vs. machine experiment we compared the human annotation with the output of an other automatic text processing tool which was independent of our annotation workflow. Possible choices were the UDPipe and Stanza modules integrated in the emtsv system, or HuSpaCy (Orosz et al., 2022). All these tools use the UDv2 tagset, therefore we needed to convert our annotations to this tagset. For the conversion we used emtsv's emmorph2ud2 module.

The tokens where the human annotation and the automatic annotation differed were considered error candidates. We compared lemmas and morphological analyses separately. In the latter case we considered both POS-tags and features because both are needed to define an emMorph tag unambigously.

The comparisons with UDPipe and Stanza, however, resulted in an unexpectedly large number of error candidates. This was caused by the incompatibility between the tagsets used by the text processing tools and the ones converted from emMorph. (The text processing tools displayed more features.) Therefore, at the results section we only show the results with HuSpaCy.

3.3. Invalid bigram method

The invalid bigram method (Květoň and Oliva, 2002) employs invalid bigrams to locate annotation errors. An invalid bigram is a POS-tag sequence that cannot occur in a corpus, and such bigrams are derived from the set of possible bigrams in a hand-cleaned subcorpus. The preparation of our hand-cleaned subcorpus is described in section 3.4.

As we used the same subcorpus for the extraction of valid bigrams and for the evaluation of the error detection methods we applied a cross-validation technique. In 10 iterations we divided both the original annotated and hand-cleaned subcorpora in 10 subparts, and used 9 subparts to extract valid bigrams and one subpart to search for error candidates. In this case all tokens that formed part of an invalid bigram were considered error candidates.

3.4. Test corpus

For the test corpus we selected 6 texts that contain 14147 tokens in total. The selection was made with the aim of including texts from as many annotators as possible and from both previously and more recently annotated texts. The latter criterion was necessary because of the changes in the annotation guidelines that occurred during the one year annotation process.

According to the classical gold standard method the selected texts were annotated by another annotator independently of the first annotator. The differences between the two annotations were examined in detail by the authors of this paper and the final annotations were decided by mutual agreement. We consider the resulting gold standard as a reference for the evaluation of the error detection methods. The list of errors was compiled based on the first annotations.

3.5. Error types in the test corpus

For a more detailed evaluation and accurate identification of error types we examined the errors of lemmas and POS-tags separately, although the errors of the different annotation levels partly overlap. In total, 502 errors were identified by comparing the first annotators and the gold standard versions. In 196 cases the lemma was wrong, and in 303 cases there was an error in the POS-tag.

The errors of each annotation level were categorised and divided into subtypes. A summary of the error types with examples and frequency values is presented in Tables 1–2.

In the case of lemmas (Table 1), we distinguished between errors in lemmatization (lem), misspellings of upper and lower case letters (lett), typos (typ), tokenization errors (tok), character errors in the text (cerr), and errors due to changes in the guidelines during the annotating process (schem). In the case of lemmatization errors the annotator defined the word structure incorrectly therefore a wrong lemma has been included in the annotation. The misspellings of upper and lower case letters occurred mainly in the case of proper names, as the emMorph module (Novák, 2014) consistently lower-cased some proper names which was not always corrected by the annotators. A typo was determined if a letter was wrong within a lemma. This was a result of changing the tokenization or correcting the token in the annotation interface. There were also some character errors in the text that affected punctuation marks and digits which were not corrected by the annotators. Changes of the guidelines refers to cases where the annotation scheme has changed during the annotation process, so the first annotator's version differs from the gold standard because it was made before the changes. An example for a change in the scheme is the annotation of titles. We decided to consider titles special noun groups therefore the last element of each title is always tagged as noun ([/N]) with the corresponding case tag, the rest of the title is tagged as [None] as not analysed, e.g. `Jojo[None] + nyuszi[/N] + ban[Ine]` ('in *Jojo rabbit*'). The lemmas of the words of a title should also be [None].

In the case of POS-tags (Table 2), a distinction was made between errors in POS tagging (lerr), errors in morphological features (morph), word errors due to wrong tokens in the text (word), errors in the syntax of the annotation (syn) and errors due to changes in the annotation scheme (schem). In the case of POS-tag errors, the annotator classified the token in the wrong part-of-speech category. However, in the case of incorrect morphological features, the part-of-speech was correct, but other morphological features displayed in the tag, such as case for nouns or number and person for verbs, were incorrect. Word errors occured when the annotators did not correct the spelling mistakes in the text. Furthermore, there were also errors in some elements of the syntax of the annotation tags. The changes in the annotation scheme also caused differences between the first annotator and the gold standard version.

LEMMA			
Error code	**Incorrect**	**Correct**	**Frequency**
lem	*szó – 'word' (noun)*	*szóval – 'so' (conjunction)*	70
lett	*nap – 'day/sun'*	*Nap – 'Sun' (star)*	15
typ	**Millenniumi – 'Millenial'*	*Milleniumi*	5
tok	**nem-megújuló – 'non-renewable'*	*nem – 'no'*	3
cerr	*15_000*	*15000*	7
schem	*Csillagok – 'Star' (from the title Star Wars)*	*[None]*	96
		Total:	**196**

Table 1: Overview of the error types of lemmatization

POS-TAG			
Error code	**Incorrect**	**Correct**	**Frequency**
lerr	*elhunyt [/V][Pst.NDef.3Sg] – 'died'*	*elhunyt [/N][Nom] – 'dead (person)'*	150
morph	*szereztek [/V][Pst.NDef.3Pl] – 'they obtained'*	*szereztek [/V][Pst.NDef.2Pl] – 'you obtain'*	40
word	**rossztanulókat [/N][Pl][Acc] – *'badstudents'*	*rossz [/Adj][Nom] – 'bad'*	14
syn	*Keralából /N][Ela] – 'from Kerala'*	*Keralából [/N][Ela]*	2
schem	*Fanny [/X]*	*Fanny [None]*	97
		Total:	**303**

Table 2: Overview of the error types of POS-tagging

4. Results

4.1. Evaluation of the performances of emtsv and the annotators

Before addressing the results of the methods described in section 3. we shortly summarize the performances of the emtsv pipeline and the annotators. These results can serve as benchmark for the evaluation of the examined methods.

In the emtsv-annotated version of our test corpus we found 989 (6,99%) errors in POS-tagging and 711 (5,03%) errors in lemmatization. We evaluated the performance of the annotators based on how many of these errors they managed to find and correct. The results are displayed in Table 3. The precision scores show that approximately 80-90% of the hand-made modifications resulted in a succesfull correction. (All tokens that were modified by the annotators were, in fact, erroneously annotated by emtsv, however, the modifications were not always correct.) The recall scores reveal that one third of the mistakes of emtsv remained undetected after the first round of manual annotation. In sum, manual annotation improved the annotation accuracy by 3-4%.

These results confirm the relevance of our study: in a workflow using a preprocessing tool with already high accuracy the standard double human annotation method seems extremely wasteful.

	Precision	Recall
Lemmas	81,50%	59,48%
POS-tags	91,23%	62,95%

Table 3: The performance of the annotators in finding emtsv's mistakes

	Number of error candidates	Found errors	Precision	Recall
Baseline	8796	247	2,81%	83,17%
emtsv vs. human	535	99	18,50%	33,33%
HuSpaCy vs. human	2714	147	5,42%	49,49%
Invalid bigram	967	130	13,44%	43,77%

Table 4: Results of the tested methods on POS-tags

4.2. Results of the tested methods

Although the invalid bigram method can be used for POS-tags only we evaluated the performances of all four methods on lemmas and POS-tags separately. The results are displayed in Tables 4–5. The overall best recall was achieved with the baseline method, however at significantly higher cost (with higher number of error candidates to revise by humans) compared to the other methods. This can also be due to the fact that the annotated corpus used by the baseline method is not particulalrly large (350.000 tokens).
On POS-tags (Table 4) the HuSpaCy vs. human method achieved the best result: comparing the annotations with the output of HuSpaCy we were able to detect almost half of the errors while the precentage of error candidates in the corpus remains only 20%.
Error detection in lemmatization (Table 5) seems to be a harder task. In case of the invalid bigram method the matches are only due to the overlapping of POS-tag and lemma errors, however this method surprisingly still outperforms the emtsv vs. human method. The winner was again HuSpaCy vs. human with a recall of 33,16%.
We also examined which error types were the methods the most successful with. The overviews of these are shown in Tables 6–7.
In the case of POS-tags (Table 6) both emtsv vs. human and invalid bigram methods detected all syntactic mistakes. This is not surprising since a syntactic mistake is only possible when the annotator inserts the tag manually, therefore it will obviously differ from emtsv. The invalid bigram method is equally foolproof for finding syntactic errors since the validated subcorpus (from which we extracted the list of valid bigrams) is very unlikely to contain syntactically incorrect tags. The invalid bigram method also seems efficient in finding the errors of morphological features.
In the case of lemmatization errors (Table 7) the HuSpaCy method was able to find most of the misspellings of upper and lower case letters. The emtsv method's best performance on lemmas was the detection of typos.
As seen each method excels in the detection of different error types therefore it seems worthwile combining them. We measured the overall performance of the 3 methods combined. In this experiment we did not differentiate between lemma and POS-tag errors but counted with erroneous tokens (tokens containing any erroneous annotation) instead. One reason of this change in evaluation is that the separate detection of lemma errors did not turn out very successful. Lemmatization errors, however often coincide with POS-tag errors, therefore the error candidates of POS-tags may detect lemma errors as well. Secondly, during the annotation workflow we annotate lemmatization and POS-tagging at the same time therefore the main goal of error detection in our project is indeed detecting erroneous tokens.
The overall results are shown in Table 8. The annotated test corpus contained altogether 401 (2.83%) erroneous tokens. The error detection methods combined gave 4238 error candidates and found 64,59% of the errors. This means that by revising 29,96% of the corpus it is possible to reduce the ratio of erroneous tokens to 1%.
The results seem even more promising when we review what kind of errors did remain undetected (Table 9). As seen the major part (63,38%) of remaining errors is due to the changes of the annotation guidelines. On the one hand, these errors only affect a smaller part of the corpus, and on the other, the changes are well known therefore it is possible to elaborate specific methods to detect them. This seems a worthwile step for the future of our research.
For more context to our results we reviewed the false positive error candidates looking for further errors that were not detected during the making of the test corpus (because the annotators agreed on them). We found 131 tokens that contained erroneous annotation(s). The distribution of the found error types is shown in Table 10.
These further found errors reveal that the expensive traditional double annotation method is not foolproof either. The tested automatic error detection methods achieved al-

	Number of error candidates	Found errors	Precision	Recall
Baseline	5099	160	3,14%	81,63%
emtsv vs. human	365	32	8,77%	16,33%
HuSpaCy vs. human	801	65	8,11%	33,16%
Invalid bigram	967	42	4,34%	21,43%

Table 5: Results of the tested methods on lemmas

	emtsv	HuSpaCy	Bigram
POS	38,00%	48,00%	48,00%
Morphological features	20,00%	55,0%	70,00%
Syntax	100%	50,0%	100%
Change of scheme	29,90%	50,52%	25,77%
Word error	21,43%	21,43%	21,43%

Table 6: The ratios of found POS-tag errors by type

	e-magyar	HuSpaCy	Bigram
Typo	60,0%	40,0%	20,0%
Character error	14,29%	42,86%	0%
lemmatization	31,43%	55,07%	38,57%
Upper-/lowercase	0%	86,67%	13,33%
Changes of shceme	6,25%	8,33%	12,50%
Tokenization	0%	0%	0%

Table 7: The ratios of found lemma errors by type

Error candidates		Erroneous tokens		Found errors		Remaining errors	
Number	Ratio	Number	Ratio	Number	Recall	Number	Ratio
4238	29,96%	401	2,83%	259	64,59%	142	1,00%

Table 8: Overall results of the tested methods combined

Error code	Number of remaining errors
schem	90
lerr	31
morph	6
lem	6
cerr	3
word	2
lett	2
typ	2

Table 9: Remaining errors by error types

Error code	Number of found errors
lerr	76
lem	20
schem	20
lett	9
morph	6

Table 10: Further found errors by error types

most the same accuracy while reduced the human work in the second round annotation to ∼30%.

5. Summary

The goal of our research was to reduce the amount of human annotation in our gold standard corpus project. We tested three different error detection methods, two of which is based on the comparison of human and automatic annotations. The manually annotated test corpus was compared to the output of the emtsv pipeline (used for preprocessing), and HuSpaCy (a text processing tool that is independent of our annotation workflow). The third examined method is called invalid bigram method and is based on detecting invalid POS-tag bigrams. The method uses a set of valid bigrams extracted from a validated subcorpus.

Although neither of the tested methods proved to be efficient enough on its own the more thorough examination of the found errors revealed that each method detected different error types. The results of the three methods combined showed that by revising 29,96% of the tokens it is possible to reduce annotation errors to 1%. This means a significant decrease in human workload with the preservation of high annotation quality.

The corpus and the annotated texts used for this research are available in our github repositories.[1]

References

Dickinson, Markus and Dan Tufis, 2017. Iterative enhancement. In Nancy Ide and James Pustejovsky (eds.), *Handbook of Linguistic Annotation*. Springer, pages 257–276.

Indig, Balázs, Bálint Sass, Eszter Simon, Iván Mittelholcz, Péter Kundráth, and Noémi Vadász, 2019. emtsv – Egy formátum mind felett. In Gábor Berend, Gábor Gosztolya, and Veronika Vincze (eds.), *XV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2019)*. Szeged: Szegedi Tudományegyetem Informatikai Tanszékcsoport.

Květoň, Pavel and Karel Oliva, 2002. Achieving an almost correct pos-tagged corpus. In Petr Sojka, Ivan Kopeček, and Karel Pala (eds.), *Text, Speech and Dialogue*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Novák, Attila, 2014. A New Form of Humor – Mapping Constraint-Based Computational Morphologies to a Finite-State Representation. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis (eds.), *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA). ACL Anthology Identifier: L14-1207.

Orosz, György, Zsolt Szántó, Péter Berkecz, Gergő Szabó, and Richárd Farkas, 2022. HuSpaCy: an industrial-strength Hungarian natural language processing toolkit.

Vincze, Veronika, Dóra Szauter, Attila Almási, György Móra, Zoltán Alexin, and János Csirik, 2010. Hungarian Dependency Treebank. In *Proceedings of LREC 2010*. Valletta, Malta: ELRA.

Váradi, Tamás, Eszter Simon, Bálint Sass, Iván Mittelholcz, Attila Novák, Balázs Indig, Richárd Farkas, and Veronika Vincze, 2018. E-magyar – A Digital Language Processing System. In Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga (eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA).

[1] Full corpus: `https://github.com/ELTE-DH/gold-standard`
Data of this research: `https://github.com/ELTE-DH/gold-standard-eval`

Do Arguments Migrate? Using NLP for Understanding Academia

Jürgen Neyer[1], Sassan Gholiagha[2], Mitja Sienknecht[3]

[1]European New School of Digital Studies (European University Viadrina)
neyer@europa-uni.de
[2]European New School of Digital Studies (European University Viadrina)
gholiagha@europa-uni.de
[3]European New School of Digital Studies (European University Viadrina)
sienknecht@europa-uni.de

Abstract

A crucial question for academia is the relevance of arguments for scientific progress. Are participants in academic debates open to the arguments and insights of other authors, even if they are embedded in competing research paradigms? Or is discursive openness limited to intra-paradigmatic debates? What are the conditions under which arguments are migrating inside and across paradigms? The paper presents the research design and first results from an ongoing research project which uses machine learning (ML) and natural language processing (NLP) to analyse a large corpus that combines thousands of research articles in International Relations (IR) scholarship. The project sets up the most extensive annotated text corpus available for international relations and trains an algorithm to recognise and qualify arguments according to their theoretical origin, supporting evidence and argumentative structure. It relies on an especially designed domain-level category system for the domain-level annotation and a simplified version of Toulmin's argumentation model for the argumentation-level annotation.

Keywords: NLP, ML, International Relations, Arguing, Academia, Politics

1. Introduction

Arguments are central in social science. Arguments are used to make sense of complex data, challenge assumptions, and develop theories. They are often specific to certain theories and help distinguish between competing theories. But while arguments are often assumed to be theory-specific, an open question in science is under what conditions arguments migrate inside and across paradigms. What kind of arguments have a significant probability of changing another's opinion, and to what extent can a systematic connection between reception intensity and specific features of scientific arguments be empirically proven? The paper presents the research design and first findings from a four-year research project to build a social science artificial intelligence (AI) lab for research-based teaching (SKILL).[1] Relying on computational linguistic and visual analysis of the corpus based on machine learning (ML) and natural language processing (NLP), the project aims to demonstrate the importance of arguments and how they are used in scholarly debates from the field of International Relations (IR) and political debates in the global realm. The results of the project and the interfaces and products developed as part of it can then be employed for both research and teaching.

To this end, the paper presents the theoretical foundations of the project (section 2), epistemological reflections (section 3), the data, the model used, and the methodological approach (section 4), as well as first findings in the conclusion (section 5).

2. Theory

Scientific discourse assumes that argumentative quality matters (Zangl and Zürn 1996, Müller 2004, critically Hanrieder 2011). Arguments are assumed to be assessed according to the merits of their scientific quality. Relevant standards include different features depending on scientific theoretical provenance. Positivist epistemologies emphasise the empirical verifiability of claims and the repeatability of lines of evidence (King, Keohane and Verba 1994). Constructivist epistemologies reject this claim and instead emphasise the subjectivity of observation and, thus, the impossibility of objectively testing claims about social facts (Berger and Luckmann 1966; Kratochwil and Ruggie 1986). Therefore, alternative standards of science are emphasised, such as the detailed and plausible reconstruction of meaning with the aim of making them comprehensible and thus understandable (see Jackson 2011 for an overview of different scientific logics for IR).

Regardless of the respective scientific theoretical orientation, theoretical reflections are, in both cases, endowed with additional plausibility when they are supported by empirical evidence. Both perspectives also share the idea that empirical data only become relevant through their explicit integration into a theoretical context. They furthermore both assume that theoretical perspectives gain traction to the degree that they are explained through an explicit exposition of their premises. The idea that quality matters for arguments to be considered seriously also applies to scientific policy advice. When scientists advise policymakers, they usually assume that their

[1] SKILL is funded by the German Ministry for Education and Research, the Brandenburg Ministry for Science and Culture, and the Thuringian Ministry for Science, Research and Art. It is chaired by Bernd Fröhlich, Katrin Girgensohn, Jürgen Neyer, and Benno Stein.

arguments will be considered if they comply with scientific standards.

However, the assumption of a high relevance of argumentation-specific features for their reception by other scientists and policymakers is not undisputed. Receptions within the scientific community are not only influenced by the quality of the arguments presented but also by their integration into established research networks (Risse, Wemheuer-Vogelaar and Havemann 2020) and sometimes even "citation cartels" (Teodorescu and Andrei 2013). Intellectually challenging positions that deviate from the majority opinion are easily ignored if they are not backed by particularly strong arguments and evidence while complying with lower standards is often good enough for arguments that replicate the mainstream. Thomas Kuhn has prominently pointed out that research programs have their own internal logic, selectively receiving content based on whether it fits into dominant paradigms (Kuhn 1962). Despite high formal quality, arguments would be easily ignored if they ignored dominant understandings of problems and solution strategies (paradigms) and followed unorthodox trajectories.

For policy advice, the assumption applies analogously that scientifically sound arguments are only received by policymakers if they can be reconciled with prevailing political calculations, i.e., are politically opportune (Böcher 2022). Luhmann's thesis of different societal functional systems, each with its own language codes and rationality criteria (Luhmann 1984), also suggests that the idea of a search for truth that integrates functional systems and is based on argumentation is at least optimistic: In science, knowledge is generated within the framework of disciplinary concepts and prevailing epistemological interests. It often sits squarely with the logic of politics in which solutions must be negotiated, and compromises will often be based on different values and interests. Science also involves a continuous critique and problematisation of findings, thus inevitably rejecting any conclusive certainty. This irrevocable uncertainty in science is, in turn, difficult to reconcile with the expectation that policymakers are able to make effective decisions that inspire consent and confidence (cf. Böcher 2022).

The tension between the thesis of an argumentation-based dynamic of scientific discourse, on the one hand, and the indications of non-scientific factors influencing the reception of arguments, on the other hand, gives rise to two interrelated questions. First, what is the significance of the quality of a scientific argument for its reception and the change of another's opinion? Second, to what extent can a systematic connection between reception intensity and specific quality features of scientific arguments be empirically proven?

3. Epistemology

The SKILL project addresses these questions by annotating and subsequently analysing a large corpus of academic articles. It develops an algorithm that can recognise and compare patterns of argumentation structures in the corpus. The algorithm may then be used on other corpora, such as debates within the United Nations or other international fora. The project follows an abductive approach, which is based on a combination of ML and NLP. It allows the combination of quantitative and qualitative methods and thus a "methodological twin-move of making *big data thick* and *thick data big*" (Adler-Nissen *et al.* 2021: 1, emphasis in original).

Abductive approaches to pattern recognition have been quite unusual for the social sciences. They have only recently started to gain some attention in the context of large data sets and have only been slowly considered by the social sciences. This immigration into a theory-driven discipline was triggered by the realisation that computer-based methods can unveil social patterns, which have since long been reflected upon but hardly ever been described empirically. The successes of research driven by big data have underlined that individual decisions often reflect broader social patterns rather than individual reflection (Nassehi 2019, Meyer-Schönberger/ Cukier 2013).

Social action is not only shaped by digitalisation but seems to be highly digitally structured and shaped by patterns of rule-compliant action. Pattern recognition procedures thus apply a methodology very much in line with an important logic of social action. The seemingly naive question of "what is?", which has often been rejected as unscientific up to now, moves to the centre in a recognition-oriented approach. Not the testing of hypotheses or the search for merely subjective meaning inherent in understanding-oriented approaches, but the identification, representation and analysis of regular social phenomena – such as arguments – become the goal of the research process.

4. Methodology and Data

This section provides an overview of our methodology, the models used, and the corpus that will be annotated. The section also provides a detailed description of the training process.

4.1. Argumentation model

We use a model of argumentation which builds on NLP methodology, which enables an algorithm to identify and classify arguments. The methodology holds that text can be made machine-readable by annotating individual sentences, i.e. using clearly defined categories to attach meaning to a text.

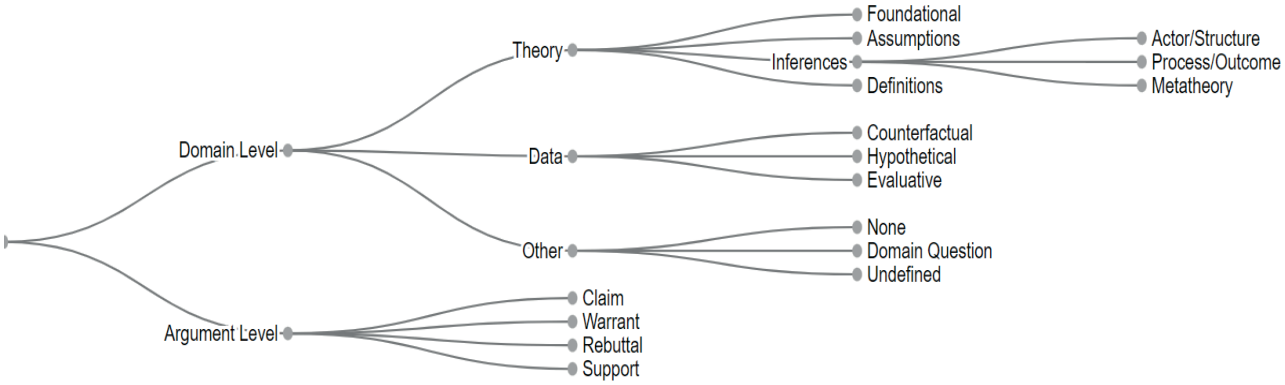


Fig. 1: Overview of categories for annotation. Tree design by Dora Kiesel, Bauhaus University Weimar

The methodology is composed of three main elements:

1. It starts from the assumption that the meaning of sentences can be assessed and understood without referencing the broader context in which they are embedded. Texts are thus decomposed into a set of sentences that are each annotated irrespective of their relationship with provisional or trailing sentences. The decomposition of texts is not unconditional, however. Provisional or trailing sentences are used as an additional resource for annotation if they provide important information without which sentences cannot be properly understood. The process of decomposing texts into sentences is also contextualised by adding relationships between sentences. Sentences that refer to each other and provide an explicit argumentative context are annotated as hanging together. For example, if sentence 1 contains a claim and sentence 2 lists the supporting evidence, then both sentences are annotated as relating to each other.
2. The annotation process works with a category tree that distinguishes between the domain and the argument level (see Figure 1). The domain level refers to propositions which make substantive claims about international politics, such as "war is wrong", "Russia has invaded Ukraine", or the like. In order to allow for a more detailed analysis, the model furthermore distinguishes between theory categories such as foundational, assumption, inferences, and definitions; data categories such as counterfactual, hypothetical, and evaluative; and other categories such as none, domain questions, and undefined. The distinction on the theory part of the category tree between foundational, assumptions, inferences, and definitions allows for considering theoretical contexts such as Realism, Constructivism, etc. Inferences can then be labelled as pertaining to either agents/structure or process/outcome. Additionally, statements could be labelled as providing a metatheoretical assessment of an inference. As a result of this category scheme, a detailed analytical framework emerges that allows the algorithm to search for specific arguments systematically and to relate them to theoretical conceptions.
3. Annotation on the argument level is concerned with the illocutionary aspects of a sentence. Sentences can imply an assertion, support of another claim, a contradiction, or an attack on another position. Annotating these attributes is important for guiding the algorithm in presenting arguments when they are to be set in a discursive context. A Realist debating with a constructivist would, for example, most likely use different concepts on the domain level (emphasising norms rather than interests) and opt on the argument level for a contradiction or an attack to undermine the thrust of a competing argument. Illocutionary annotations are undertaken independently of the material content of a sentence.

4.2. Corpus

The project aims to create and publish an annotated corpus comprising all open-access articles from the most important English-speaking political science journals dealing with international relations.[2] All sentences together will build on a corpus of approximately 800,000 annotated sentences, each with a specific domain meaning and a syntactic (illocutionary) meaning. In this process, subjective meaning

[2] The currently used text corpus comprises a total of 25 different scientific journals with a total of 1980 OpenAccess texts, which are available independently of institutional accesses. These are the American Journal of Political Science, British Journal of Politics and International Relations, Cooperation and Conflict, Ethics & International Affairs, European Journal of International Relations, European Journal of International Security, Foreign Affairs, Global Constitutionalism, Global Society, International Organization, International Security, International Studies Quarterly, International Theory, Journal of Common Market Studies, Journal of Conflict Resolution, Journal of Peace Research, Millennium: Journal of International Studies, Political Research Exchange, Politics and Governance, Politics & Society, Review of International Studies, Security Dialogue, Third World Quarterly, West European Politics, World Politics.

is quasi-reified by being assigned an objectified meaning. An annotated sentence is no longer merely an author's subjective opinion or a recipient's interpretation but becomes a datum with objective domain meaning, syntactic meaning, and a relation to another datum also with objectified domain and syntactic meaning.

This basic sum of annotated sentences represents the raw mass by means of which the algorithm begins to search for specific arguments and patterns of domain and argumentation attributes. With each additional analytic category added to its repertoire, its sensitivity to additional patterns increases, and with each additional text, its ability to process additional statements grows. The resulting dataset allows the algorithm to be trained to identify argumentative patterns from assumptions, processes, and outcomes of different theoretical provenance and to discriminate according to whether and with what kind of structure and evidence they are provided. The result is an instrument that can be used to interrogate texts across theories and time with respect to their argumentative structures and to generate statements about the conditions of their reception or rejection.

4.3. Training

The project invests much effort in the training of the annotators. Here, the training of the annotators (step 1) must be distinguished from training the algorithm (step 2) and from its subsequent independent learning and further data processing (step 3).[3]

The training of the annotators begins with the development of a so-called gold standard. A gold standard is a reference annotation used as a benchmark for annotator performance. In this gold standard, the trainers define a specific mode of annotating sentences with the aim of conveying the underlying principles to the annotators in such a way that they can understand and apply them autonomously. The practice of annotation is trained initially with four central texts characteristic of the four theoretical perspectives of neorealism, liberalism, constructivism and feminism.[4]

These texts are annotated by both student annotators and domain experts (the authors). The annotation process has two aims. First, develop a category system on the domain level that works across different theories, ontologies, and epistemologies (see Figure 1). Second: To train annotators in that category system, refine the category system, and reach a sufficient level of agreement with the gold standard (i.e. the annotation by the senior domain experts) and inter-annotator-agreement. Both are absolutely crucial for step 2.

In step 2, the annotators annotate the large corpus of IR journal articles. Here individual annotators are given different tasks of annotation, with the senior domain experts also annotating some of the corpus. Constant checks of gold standard comparison and inter-annotator reliability ensure sufficient annotation quality. In this step, the algorithm learns to identify arguments relating to theory-specific propositions, to tell, for example, an assumption from an empirical reference and to distinguish between different types of empirical references.

Step 3 of the training grants the algorithm access to the full-text corpus. In this process, the algorithm is set up for (semi-)autonomous annotation and machine learning. It will be closely guided by the annotators and monitored to see if the annotations comply with the standard developed in step 1. This third step leads to a large argumentative repertoire of the algorithm and, thus, significant usability. The repertoire should allow both the systematic search for arguments by users and infer statements about correlations of domain-level features and illocutionary arguments. This approach opens a promising way for answering the research question about the relevance of successful, i.e., persuasive arguments and their domain- and illocutionary features. At a later stage, the algorithm may then be applied to a larger corpus of IR journals or even different corpora, such as debates in the United Nations or the European Union.

This third step leads to a large argumentative repertoire of the algorithm and, thus, significant usability. The repertoire should allow both the systematic search for arguments by users and allow to infer statements about correlations of domain-level features and illocutionary arguments. This opens a promising way for answering the research question about the relevance of successful, i.e., persuasive arguments and their domain- and illocutionary features.

5. Conclusion

The approach taken here to research the relevance of arguments in scientific debates goes a qualitative step further than most previous social science projects. It looks for argumentative patterns in complex communicative acts. Not material reality, but scientific exchange and thus communication about reality is made the object of knowledge. Such a combination of AI/ML and NLP for social scientific reflection and its relevance for political reality has not yet been attempted in this way and to this extent.

Even though SKILL is still in an early phase, first substantial findings can already be reported. The training of the annotators and the implementation of the first annotation exercises on texts from International Relations have underlined the need for, and difficulty of, assigning subjectively meaningful interpretations to an objectifiable schema. This difficulty is first expressed in the definition of separable categories at the domain level. On the one hand, the categories must be specific enough to allow for a high degree of inter-annotator reliability. At the same time, they must be sufficiently general to apply to different theories. What becomes clear in this process is that the structure of

[3] At the time of writing we are in the final stages of step 1.

[4] Kenneth N. Waltz: The Emerging Structure of International Politics, *International Security.* Vol. 18, No. 2 (Fall, 1993), pp. 44-79; Robert D. Putnam, Diplomacy and Domestic Politics: The Logic of Two-Level Games, *International Organization*, Vol. 42, No. 3 (Summer, 1988), pp. 427-460; Finnemore, Martha; Sikkink, Kathryn (1998): International Norm Dynamics and Political Change. In *International Organization* 52 (4), pp. 887–917; Zalewski, Marysia (1995): 'Well, What is the Feminist Perspective on Bosnia?'. In *International Affairs* 71 (2), pp. 339–356.

arguments in scientific texts is far more complex than in other text genres, such as debate articles.

The difficulty of objectifying subjective meanings is also evident in annotators and domain experts working with subjective understandings of IR theories. Establishing an intersubjectively shared understanding thus requires not only mutual explanation but also a high degree of external understanding (Schütze *et al.* 1973). This presents one of the greatest challenges: Is it possible to develop a sufficiently intersubjectively shared understanding of theory without one of the existing interpretations claiming hegemonic status and thus marginalising equally valid interpretations? Or is it the case that the method of pattern recognition by necessity implies the setting of an exclusionary "gold standard"? Are ML and NLP thus necessarily establishing an algorithmic entity with a quasi-scientific "personality" that relies on specific interpretations of reality and will hardly ever be more objective than its annotators?

A final remark relates to the status of theory in a data-driven approach: Social science has, for many years, been dominated by theory. Good scientific work was only too often expected to start with theoretical reflections and use data only to illustrate its findings. Big data, ML and NLP, reverse this methodological bias. The seemingly naive question of "what is?", hitherto often rejected as unscientific, moves to the centre in a pattern-oriented approach. However, an approach to recognising patterns must not be misunderstood as an analytical or theoretical tabula rasa. Unfortunately, exaggerated and misguided misunderstandings of pattern recognition circulate in the literature.

Anderson, for example, fears that in the future digital data analysis will be able to do without researchers since machines could also independently develop the necessary expertise that would be needed in the algorithmic research process (Anderson 2008, Müller and Ritschel 2016: 5). Such fears are based on a misunderstanding of how algorithm-based pattern recognition works. Algorithms can only recognise meaningfully at all, i.e. distinguish relevant from irrelevant, if they have criteria that allow them to make this distinction. For example, an unguided search for patterns may allow the description of reality but will hardly allow any focused statements about scientifically relevant questions. Meaningful recognition, therefore, requires cognition-structuring analytical criteria. These criteria, in turn, cannot be drawn from a conceptual vacuum but must be anchored in theoretical discourses. Like any other social science question, a pattern recognition approach requires a thorough connection to theoretical discourses.

References

Adler-Nissen, R., Eggeling, K.A. and Wangen, P. (2021): Machine Anthropology: A View from International Relations. In *Big Data & Society* 8 (2), 1-6

Anderson, C. (2008): The End of Theory: The Data Deluge Makes the Scientific Method Obsolete. Available online at https://www.wired.com/2008/06/pb-theory/

Berger, P.L. and Luckmann, Thomas (1967 [1966]): *The Social Construction of Reality.* First Anchor Books Edition. New York: Anchor Books.

Böcher, Michael (2022): Wie funktioniert wissenschaftliche Politikberatung? in *Forschung und Lehre*, 02.06.2022, https://www.forschung-und-lehre.de/politik/wie-funktioniertwissenschaftliche-politikberatung-475

Hanrieder, Tine (2011): The false promise of the better argument. In *International Theory* 3 (3), 390–415.

Jackson, P. T. (2011): *The conduct of inquiry in international relations. Philosophy of science and its implications for the study of world politics*. London, New York: Routledge

King, G., Keohane, R.O. and Verba, Sidney (1994): *Designing social inquiry*. Princeton, N.J., Chichester: Princeton University Press.

Kuhn, T.S. (1962): *The structure of scientific revolutions*. Chicago: Chicago University Press.

Luhmann, N. (1984): *Soziale Systeme: Grundriss einer allgemeinen Theorie.* Frankfurt am Main: Suhrkamp.

Mayer-Schönberger, V. and Cukier (2013): *Big Data: Die Revolution, die unser Leben verändern wird.* 3rd edition München: Redline Verlag.

Müller, H. (2004): Arguing, Bargaining and All That: Communicative Action, Rationalist Theory and the Logic of Appropriateness in International Relations. In *European Journal of International Relations* 10 (3), 395–435.

Müller, T. and Ritschel, G (2016): Big Data als Theorieersatz? In T. Müller, G. Ritschel, A. Amberger, S. Böschen, R. Broemel, U. Busch et al. (eds.): Big Data als Theorieersatz. *Berliner Debatte Initia*l 4/2016. (2016) 4), 1–8.

Nassehi, A. (2019): *Muster: Theorie der digitalen Gesellschaft.* München: C.H. Beck.

Schütze, F., Meinefeld, W., Springer, W. and Weymann, A. (1973): Grundlagentheoretische Voraussetzungen methodisch kontrollierten Fremdverstehens. In Arbeitsgruppe Bielefelder Soziologen (Hrsg.): *Alltagswissen, Interaktion und gesellschaftliche Wirklichkeit* - Volume 2. Reinbek bei Hamburg: Rowohlt, 433–495.

Teodorescu, D. and Andrei, T. (2014): An examination of "citation circles" for social sciences journals in Eastern European countries. *Scientometrics* 99 (2), 209–231.

Zangl, B.; Zürn, M. (1996): Argumentatives Handeln bei internationalen Verhandlungen: Moderate Anmerkungen zur post-realistischen Debatte. In *Zeitschrift für Internationale Beziehungen* 3, 341-366

A Transfer Learning Approach for SDGs Classification of Sustainability Reports

Ata Nizamoglu[1,3], Lea Dahm[1,3], Talia Sari[1,3], Vera Schmitt[1], Salar Mohtaj[1,2], Sebastian Möller[1,2]

[1]Technische Universität Berlin, Berlin, Germany
[2]German Research Centre for Artificial Intelligence (DFKI), Labor Berlin, Germany

{nizamoglu | lea.dahm | t.sari}@campus.tu-berlin.de
{vera.schmitt | salar.mohtaj | sebastian.moeller}@tu-berlin.de

[3] The first three authors have contributed equally

Abstract

In 2015, The United Nations (UN) provided a blueprint for sustainable development in various domains. This Blueprint described 17 Sustainable Development Goals (SDGs), such as "No Poverty", "Zero Hunger", or "Gender Equality". Subsequently, many companies have started publishing yearly sustainability reports, explaining their efforts with respect to the SDGs. However, the manual assessment of these reports is an infeasible task, and the automatic processing of text documents is necessary to aggregate information about the distribution of SDGs throughout various domains. In this research, we have developed and measured the performance of various natural language processing models from classical to transfer learning-based models to identify the targeted SDG in sustainability reports. Hereby, transformer-based models show the best performance for this task, especially BERT-based models, such as RoBERTa. The results show, that the approach of automatically processing text documents to classify SDGs in various documents is feasible and can be used to aggregate information about which SDGs are covered by which companies and industry domains.

Keywords:

1. Introduction

Various types of organizations such as private entities and scientific and governmental institutions publish documents reporting about activities related to Sustainable Development Goals (SDGs). In 2015, all member states of the United Nations agreed upon the 2030 Agenda for Sustainable Development, which serves as a framework for addressing poverty, improving health and education, reducing inequality, promoting economic growth, and addressing the challenge of climate change (United Nations, 2015). The UN member states have defined 17 SDGs along with their corresponding targets that need to be achieved. As a result, the private sector has taken steps to address the SDGs by releasing annual sustainability reports, which detail the actions taken by companies to tackle the challenges related to the respective SDGs.

The information used to create the *Annual SDG* progress report comes from a variety of sources, including policy recommendations, sustainability reports, and progress reports. SDG experts analyze these sources and consolidate the information they contain in order to create the report (Guisiano et al., 2022b). However, identifying which SDGs are addressed in text documents published by both the private and public sector is a time-consuming task. This is particularly challenging as the number of documents addressing the SDGs increases each year, making manual scanning and classification a daunting task (Mhlanga et al., 2018). The use of machine learning models can aid in scanning large volumes of documents for content related to the SDGs, thus supporting the process of identifying and analyzing sustainability-related information on a large scale.

There have been attempts to address the infeasibility of manual assessment of the massive amount of SDG-related text documents. Most of the approaches focus on applying deep learning models to automatically classify SDGs in text documents, such as the *SDG-Meter* applying transfer learning models to map SDGs in progress reports (Guisiano et al., 2022b), and sustainability report classification based on the Open Source SDG (OSDG) Community Dataset (Angin et al., 2022), or SDG-oriented artifact detection in various types of text documents (Hajikhani and Suominen, 2022).

In this work, the focus is on analyzing various text sources (e.g. scientific publications and information from the UN website), which can vary greatly in terms of length, word count, and structure. There are no standards defining how to structure and describe any efforts made to address one or more of the 17 SDGs. Thus, the manual assessment of SDGs of different text sources is very challenging, and automated procedures are required to process the increasing amount of text sources and aggregate the activities and progress made by different industries, companies, and scientific and public institutions.

Therefore, in this paper different natural language processing (NLP)-based models will be applied to assess their performance in classifying the 17 SDGs in different text sources. For this purpose, a new dataset containing $41,351$ sentences from various text sources addressing SDGs has been gathered. The dataset has been used for training and testing the NLP models (mainly transformer-based mod-

els), such as BERT, RoBERTa, XLNet, and a stacking model under which we combined the predictions from the four best-performing models. Overall, RoBERTa achieves the best performance of $86,3\%$ F1, which is in line with the findings from (Guisiano et al., 2022b). The main contributions of this work are as follows:

1. Collect and scrape SDG-related text data from various text sources ($41,3k$ sentences),
2. Apply different data-preprocessing strategies for balancing the class distribution to improve the overall performance of teh transformer-based language models,
3. Analyzing the performance of state-of-the-art pretrained transformer models on the task of automated SDGs classification as a multi-class classification task.

The rest of the paper is organized as follows; Section 2. summarizes the state-of-the-art literature on the classification of SDGs in different domains. An overview of the dataset and the pre-processing steps are presented in Section 3., and the experimental setup and results are discussed in Sections 4. and 5., respectively. Finally, we conclude the paper and the system in Section 6..

2. Related Work

The number of documents reporting SDGs by companies and (international) organizations continues to increase, and new approaches to processing this information have been proposed. Hereby, NLP methods significantly contribute to developing solutions to automatically process large text documents reporting progress made with respect to SDGs. Recent advancements in deep learning for various NLP tasks have led to the development of large language models showing high performance for complex NLP tasks (Angin et al., 2022). Hereby, transformer-based methods show the most promising results in detecting SDGs in text documents, achieving a high-performance (Angin et al., 2022). For the processing of scientific reports, Smith et al. (2021) have assessed *Doc2vec* in combination with clustering (Le and Mikolov, 2014) to analyze similarities of SDGs in scientific research documents (Smith et al., 2021). Transformer-based models have been applied by Guisiano et al. (2021), who developed a tool based on the Bidirectional Encoder Representation from Transformer (BERT) model (Devlin et al., 2018) to facilitate faster processing and classification of the SDGs in text, by focusing on SDG 17 (Guisiano et al., 2022b). Yet, the UN emphasizes that the SDGs are interlinked and applying models to detect one SDG, will facilitate working on the other SDGs as well (United Nations, 2015). For example, improving health and education are fundamental elements to ending poverty and reducing inequality. (Smith et al., 2021) used NLP methods to analyze inter-dependencies between the goals, aiming to provide insight into overlaps in public conversation. The findings indicated that certain terms played a central role in addressing multiple SDGs. For instance, the term "gender" was found to be significant in discussions pertaining to both goal 5: gender equality, and goal 4: quality education. Another example is the term "development assistance", which was commonly referenced in relation to goal 2: zero hunger, goal 3: good health and well-being, goal 10: reduced inequalities, and goal 17: partnerships for the goals. The positive and negative correlation between indicators of the goals is analyzed in (Pradhan et al., 2017). This research assessed the degree to which the 231 indicators that comprise the 17 goals are complementary to each other, or are trade-offs. The findings of this study revealed that most indicators were considered to be synergistic within and across different SDGs. Nevertheless, certain indicators for particular goals were found to be contradictory to each other. While the automatic detection of SDGs in text documents does not enable a qualitative assessment of the impact of the mentioned SDGs or identify instances of "greenwashing" in the text, it can assist in consolidating information on which SDGs are being addressed by various companies or industry domains. This, in turn, can streamline further processing of the information. Thus, this research focuses on the multi-class classification of SDGs in sustainability reports. Specifically, the interrelated nature of the SDGs is treated as a multi-label task, where the classifier can assign sentences to one or more labels (Guisiano et al., 2022a).

3. Dataset

The subsequent section outlines the data collection procedure and the dataset utilized for training and evaluating NLP models for the task of automatically classifying SDGs.

3.1. Data Collection

To evaluate different NLP model performances with respect to the SDG classification task, a dataset was used containing 2219 sentences with the corresponding SDG labels[1]. The dataset covers two main sources: (1) scientific papers and sustainability reports from different companies, and (2) the SDG descriptions of the United Nations. From each resource, the SDG-related sentences have been extracted and aligned with the corresponding labels. The data is split into train and test sets, where the train set contains 37216 instances and the test set has 4135 instances. Some key statistics from the dataset are presented in Table 1.

Attribute	#
Number of instances (i.e., sentences)	41351
Max length of texts (in character)	1931
Min length of texts (in character)	11
The average length of texts (in character)	547

Table 1: Summary of statistics and frequency distribution of the dataset

Figure 1 highlights the difference between the length of texts in the dataset for each source. As depicted in the fig-

[1]The dataset and the related code can be found in the following GitHub Repository: https://github.com/ataniz/SDGs-Classification-of-Sustainability-Reports

ure, the sentences from scientific papers tend to be longer compared to sentences from the UN source.

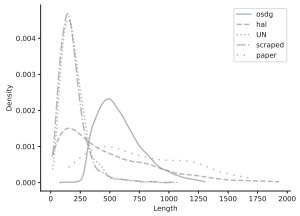


Figure 1: The distribution of sentence lengths from paper and UN sources

Furthermore, Figure 2 shows the frequency of how often different SDGs are represented in the dataset. Hereby, it is clearly visible that the dataset is not balanced, as the frequencies of the SDGs differ significantly. SDGs 5 and 4 are over-represented in the dataset, whereas SDGs 16 and 17 are underrepresented. Our dataset's imbalance can be attributed to the fact that it was collected from diverse sources across different countries. Survey studies conducted by (Kleespies and Dierkes, 2022) have indicated that the correlation between SDG counts tends to be highest in the environmental sector of various countries. The importance of each SDG goal for developing and developed countries, and for high- and low-income countries, is considered differently. For example, in some countries, where students receive an affordable and high-quality education, SDG 4 is considered far less important than other SDGs. A more detailed SDG rating graph according to various countries can be observed in (Sachs et al., 2022).

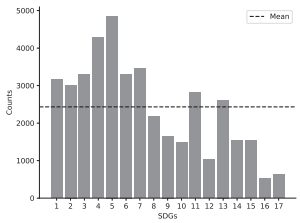


Figure 2: Number of instances per SDG in the dataset

In addition to the mentioned dataset, we scraped data from various sources in order to aid the model's generalization. In total, $41,351$ sentences were added to the training set and used in the training phase. We scraped almost $3,000$ sentences from sustainability reports, and also retrieved the remaining data from (Pukelis et al., 2020) and (Guisiano et al., 2022b)).

For scraping more data concerning sustainability reports two sources have been used: (1) `https://www.sustainability-reports.com/` (Reporting, 2002), and (2) `https://www.online-report.com/` (Nexar and Group, 2003), which aggregate corporate sustainability reports. Hereby, the respective pdfs concerning different sustainability reports have been downloaded first and the text was extracted using the python library *PyPDF2*. For the labeling task, a list of keywords associated with each SDG was compiled to label each sentence separately. A certain number of sentences has been manually verified, to assess how well the automatic labeling task was performed. Since our models are not for multi-label-dataset, we decided to assign sentences with multi-label to just a single label. For example, the sentence "More than 130 million women and girls around the world lack access to education, and women account for two-thirds of the 750 million adults without basic literacy skills" could be labeled as SDG 4 or SDG 5. Since the sentence focuses more on women's lack of access to education, this sentence has been labeled as SDG 5. Furthermore, additional data was used to extend and balance the dataset. For further balancing attempts, synthetic data generation approaches have been used, such as the python library *NLPaug*[2], to replace synonyms and back-translate text, which is also known as *round trip translation*. However, up-sampling approaches based on data augmentation did not improve the overall performance of the transformer-based models. The results in the following section are based on the original imbalanced dataset.

3.2. Pre-processing

The data has been scraped from different sources, which requires the application of text normalization as a pre-processing step to improve the generalization of applied NLP models.

Not only text-normalization has been applied but also further pre-processing steps such as:

1. spelling correction,
2. exclusion of non-English text,
3. duplicate sentence removal,
4. removing email addresses, hashtags, URLs, emojis and emoticons, footnote references, and HTML elements

Moreover, we performed normalization of non-Unicode characters, *noisy* characters, signs, and symbols (e.g., bullet points and hyphenated words quotes). The pre-processing steps were mainly done using Regular Expressions (regex) and the textacy[3] Python library. In addition to these pre-processing steps, we applied stop word removal, lowercasing, and stemming on the input text for classical models in our experiments.

To ensure an accurate evaluation of the model's performance, we split the dataset into different subsets for training, development, and testing. For classical models, we split the dataset with a 9:1 ratio of training to testing data. For transformer models, we used a split of 8:1:1 ratio of training, development, and testing data, respectively. This split allows us to fine-tune the transformer models on the development set and evaluate their performance on the test set, ensuring that the final results are robust and generalizable.

4. Experiments

In the following we will describe the experimental setup and the implementation of NLP models to classify the SGDs in text data, ranging from traditional NLP models to

[2] `https://pypi.org/project/nlpaug/`
[3] `https://pypi.org/project/textacy/`

state-of-the-art transformer-based techniques. As the baseline model, the Naive Bayes and a Support Vector Machine (SVM) model have been implemented, since they have shown promising results in text classification tasks in various domains (Luo, 2021; Xu, 2018). The input texts were converted into vectors based on the **Term Frequency - Inverse Document Frequency** (TF-IDF) weighting schema. In addition to the traditional NLP models, we also fine-tuned a number of pre-trained language models on the training and the scraped datasets. The pre-trained language models have outperformed the classical NLP models in many studies on text classification in different domains such as fake news detection (Mohtaj and Möller, 2022a). Transformer-based models such as BERT (base and large) (Devlin et al., 2018), XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019), and Ernie 2.0 (Sun et al., 2020) have been implemented. For these models, an additional dense layer has been added with an output size of 17 on top for the classification task.

We used *HuggingFace* (Wolf et al., 2020) and *Flair*(Akbik et al., 2019) Python libraries in *PyTorch* (Paszke et al., 2019) to implement the transformer-based models. Furthermore, a dropout layer has been added to prevent overfitting, and a linear layer and the softmax activation function have been used to fit the architectures to the multi-class classification task.

Regarding the hyper-parameters, we tested a range of values for different parameters including the *learning rate* (1e-6, 3e-6, 1e-5, 2e-5, 3e-5¡ 4e-5, and 5e-5), dropout probability (0, 0.1, 0.3, and 0.6), and mini-batch size (4, and 8). We used the *AdamW* (Loshchilov and Hutter, 2019) optimizer in all of the experiments and fine-tuned all of the models in 10 epochs.

Hereby, the learning rate of 1e-5 yielded the best F1-score. The dropout probability was set ¡y 0.1 and the mini-batch size to 4 in the final experiments since they showed the best performance compared to the other values.

5. Results

For the evaluation of the applied approaches, the evaluation criteria of accuracy, precision, recall, and F1-score have been used to compare the model performances. The macro F1-score values are presented in Table 2 for all models.

Classifier	F1-score
Naive Bayes	0.572
SVM	0.775
BERT-base	0.770
BERT-large	0.829
Ernie2.0	0.795
XLNet-large	0.836
RoBERTa-large	**0.863**

Table 2: The macro F1 score retrieved by different models

As it is highlighted in Table 2, almost all of the state-of-the-art transformer-based models could outperform Naive Bayes as a traditional model. However, the SVM model shows competitive results compared to the pre-trained models on the task. Among all models that we have implemented for the SDG classification task, *RoBERTa-large* yielded the best results with a meaningful margin compared to the second best model *XLNet-large*.

6. Discussion and Conclusion

In this research, the application of different NLP models for the SDG classification task has been explored. The data has been obtained from various text sources, such as scientific publications, the UN description of SDGs, and sustainability reports from different companies. NLP models such as SVM, Naive Bayes, and transformer-based models have been implemented to assess their performance in terms of the macro F1-score. Hereby, the transformer-based models, especially RoBERTa showed promising performance for the multi-class classification task, although other transformer-based models *XLNet*, *BERT*, and *Ernie 2.0* also achieved a sufficient performance for this task. The additional scraped data from sustainability reports $(41, 4k)$ improved the performance of the implemented models, but the up-sampling approaches did not improve the overall performance to balance differing frequencies of SDGs in the dataset.

Thus, further research is necessary to apply data preprocessing approaches to balance the dataset to achieve higher classification performance. Moreover, the embedding weights could be obtained from the *n* last layer of the pre-trained models, similar to the related research on text classification problems proposed by (Mohtaj and Möller, 2022b). Furthermore, it is necessary to evaluate the models in additional languages, using data from sustainability reports of companies and international organizations that are also published in other languages.

Overall, the results show that the application of NLP models for the automatic SDG classification task is feasible and can be implemented for the automatic processing of text documents. These approaches can be used to sort text documents according to their relevance concerning different SDGs and aggregate information about the relevancy of different SDGs in various industries and organizations.

Acknowledgment

We would like to express our very great appreciation to Charlott Jakob for providing the dataset and also to Lucy Grey-Gardner for her contribution implementing the NLP models.

References

Akbik, Alan, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf, 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019*.

Angin, Merih, Beyza Taşdemir, Cenk Arda Yılmaz, Gökcan Demiralp, Mert Atay, Pelin Angin, and Gökhan Dikmener, 2022. A roberta approach for automated processing of sustainability reports. *Sustainability*, 14(23):16139.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Guisiano, Jade Eva, Raja Chiky, and Jonathas de Mello, 2022a. SDG-Meter : a deep learning based tool for automatic text classification of the Sustainable Development Goals. In *ACIIDS :14th Asian Conference on Intelligent Information and Database Systems*.

Guisiano, Jade Eva, Raja Chiky, and Jonathas De Mello, 2022b. Sdg-meter: A deep learning based tool for automatic text classification of the sustainable development goals. In *ACIIDS: 14th Asian Conference on Intelligent Information and Database Systems*.

Hajikhani, Arash and Arho Suominen, 2022. Mapping the sustainable development goals (sdgs) in science, technology and innovation: application of machine learning in sdg-oriented artefact detection. *Scientometrics*:1–33.

Kleespies, M.W. and P.W. Dierkes, 2022. The importance of the sustainable development goals to students of environmental and sustainability studies—a global survey in 41 countries. *Humanit Soc Sci Commun 9, 218 (2022)*.

Le, Quoc and Tomas Mikolov, 2014. Distributed representations of sentences and documents. In *International conference on machine learning*. PMLR.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Loshchilov, Ilya and Frank Hutter, 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Luo, Xiaoyu, 2021. Efficient english text classification using selected machine learning techniques. *Alexandria Engineering Journal*, 60(3):3401–3409.

Mhlanga, Ruth, Uwe Gneiting, and Namit Agarwal, 2018. Walking the talk: Assessing companies' progress from sdg rhetoric to action.

Mohtaj, Salar and Sebastian Möller, 2022a. The impact of pre-processing on the performance of automated fake news detection. In *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer.

Mohtaj, Salar and Sebastian Möller, 2022b. On the importance of word embedding in automated harmful information detection. In *International Conference on Text, Speech, and Dialogue*. Springer.

Nexar and Message Group, 2003. Online reports database. `https://www.online-report.com/report-type/sustainability-report/`. Accessed: 2022-05-30.

Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al., 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Pradhan, Prajal, Luís Costa, Diego Rybski, Wolfgang Lucht, and Jürgen P. Kropp, 2017. A systematic study of sustainable development goal (sdg) interactions. *Earth's Future*, 5(11):1169–1179.

Pukelis, Lukas, Núria Bautista-Puig, Mykola Skrynik, and Vilius Stanciauskas, 2020. OSDG - open-source approach to classify text data by UN sustainable development goals (sdgsguisiano). *CoRR*, abs/2005.14569.

Reporting, International Corporate Environmental, 2002. The portal for sustainability reporting. `https://www.sustainability-reports.com/`. Accessed: 2022-05-30.

Sachs, J., C. Kroll, G. Lafortune, G. Fuller, and F. Woelm, 2022. *Sustainable Development Report 2022*. Cambridge: Cambridge University Press.

Smith, Thomas Bryan, Raffaele Vacca, Luca Mantegazza, and Ilaria Capua, 2021. Natural language processing and network analysis provide novel insights on policy and scientific discourse around sustainable development goals. *Scientific reports*, 11(1):1–10.

Sun, Yu, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang, 2020. Ernie 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34.

United Nations, 2015. The 17 goals — sustainable development. `https://sdgs.un.org/goals`. Accessed: 2022-05-30.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al., 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*.

Xu, Shuo, 2018. Bayesian naïve bayes classifiers to text classification. *Journal of Information Science*, 44(1):48–59.

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le, 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Token and Part-of-Speech Fusion for Pretraining of Transformers with Application in Automatic Cyberbullying Detection

Nor Saiful Azam Bin Nor Azmi[1], Michal Ptaszynski[1], Juuso Eronen[2], Karol Nowakowski[3], Fumito Masui[1]

[1]Text Information Processing Laboratory, Kitami Institute of Technology, Kitami, Japan
m2153308021@std.kitami-it.ac.jp, {michal,f-masui}@mail.kitami-it.ac.jp
[2]Prefectural University of Kumamoto, Kumamoto, Japan
[3]Tohoku University of Community Service and Science, Sakata, Japan
{eronen.juuso,nowakowski.karol.p}@gmail.com

Abstract

Studies on cyberbullying in general started around 2006 and have grown into one of the most popular eras of study. Traditionally, researchers used non-contextual machine learning (ML) models to detect cyberbullying such as TF-IDF combined with classic ML models, like SVM, but the growing popularity of neural networks, especially transformer models in various NLP tasks has encouraged researchers to use such more complex models also for cyberbullying detection. In this research, by using the transformer model ELECTRA, we were able to improve the performance of cyberbullying detection by embedding grammatical information into the model. In particular, we propose a novel method for pertaining tokens fused with Part-of-Speech (POS) information and then embedded into the ELECTRA model for the English language. We also compare several subtoken generation techniques, such as SentencePiece and WordPiece Tokenizers to find which methods properly segment the tokens and its grammatical information during the process of tokenization and pretraining.

Keywords: Automatic Cyberbullying Detection, Transformer, ELECTRA, SentencePiece, Parts-of-Speech, EDO2023

1. Introduction

Cyberbullying is defined as the abuse of open online communication by conveying harmful messages and sharing disturbing information about private individuals (Patchin and Hinduja, 2006). Cyberbullying is considered a type of harassment and includes personal threats, sexual remarks, pejorative labels, and false information (usually used to humiliate the victims). Cyberbullying messages are usually aimed at ridiculing someone's personality or appearance or spreading rumors about the victim (Ptaszynski et al., 2016). This usually leads victims to self-mutilation or even suicide (Hinduja and Patchin, 2010). Cyberbullying can frequently be seen in open Social Networking Services, such as Facebook, Twitter, and various forum-based websites, as they are in practice free from any jurisdiction to take legal action against the bullies.

In Japan, there have been many cyberbullying cases reported by the Parent-Teacher Association (PTA), which searches and monitors unofficial school websites and reports entries containing cyberbullying behavior. Unfortunately, manually searching and reading through the website is time-consuming and a psychological burden for PTA and Internet Patrol volunteers, especially with the increasing number of cyberbullying cases (Ptaszynski and Masui, 2018).

Studies on cyberbullying from the perspective of social sciences and psychology have been growing since 2006 (Patchin and Hinduja, 2006). While the first publications in automatic cyberbullying detection started to appear three years later, these studies proposed to automatically detect cyberbullying using Machine Learning (ML) to decrease human intervention (Ptaszynski et al., 2010). Recent studies have used more sophisticated neural language model architectures (Eronen et al., 2022b), some studies point out the lack of deeper linguistic patterns as an impediment to achieving satisfying results (Eronen et al., 2022a). Therefore, in research, as the first attempt of its kind, we fuse token and part-of-speech (POS) in a pretrained language model and report on the influence of POS information on automatic cyberbullying detection.

The paper is organized as follows: Section 2. covers related works. Section 3. presents the proposed method, and Section 4. discusses evaluation and analysis. Finally, Section 5. offers concluding remarks and addresses ethical considerations.

2. Background

In this section, we review the previous work related to our approach, namely, the pretrained language models, specifically Transformers, with a focus on the ELECTRA transformer model, as well as on Part-of-Speech tagging, and tokenization for pretrained language models, especially using SentencePiece Unigram tokenizer.

2.1. ELECTRA

ELECTRA, or Efficiently Learning an Encoder that Classifies Token Replacement Accurately (Clark et al., 2020), is a masked language model that is closely related to the Bidirectional Encoder Representations from Transformers (BERT) model (Devlin et al., 2018). BERT is a language model that has been designed to pretrain deep bidirectional representation from an unlabeled text by jointly conditioning on both left and right context in all layers. BERT randomly masked some of the tokens from the input, then predict the

original vocabulary id of the masked word. In contrast to ELECTRA, the model manipulates the input by replacing the token with [MASK] label and training the model to find plausible alternatives sampled from a small generator network. ELECTRA trains a discriminative model that predicts whether each token in the falsified input was replaced by a generator or not. ELECTRA models demonstrate that with the new way of pretraining, the task is more efficient than in previous Masked Language Models (MLM) because the task specifies overall input tokens rather than just the small subset that was masked out. As a result, given the same model size, data and computational prowess, the contextual representation trained by (Clark et al., 2020) approach performed significantly better than those learned by the classic BERT model. Based on these findings, we decided to use ELECTRA as the model of focus in this research.

2.2. Part-of-Speech Tagging

Part-of-Speech (POS) tagging, also referred to as morphological analysis or grammatical classification assigns verbs, adjectives, adverbs, nouns, etc. on designated words. Aiming to assign each word in the text with proper morphosyntactic tags in their context. It is one of the most important addressed areas and main building blocks in many Natural Language Processing (NLP) tasks, such as machine translation (Yang et al., 2019), word sense disambiguation (Pal et al., 2015), question answering (Nohria and Kaur, 2018), etc. With the increasing amount of data available, manually tagging POS to words is laborious, and time-consuming, especially if the dataset to be annotated contains millions of sentences and words. A tool performing POS tagging, or POS tagger, assigns POS information on words within a dataset (Zewdu and Yitagesu, 2022). Recent POS taggers annotate grammatical information automatically with very high, near-ideal accuracy[1], which makes them reliable tools to reducing the burden of manual tagging.

2.3. SentencePiece

SentencePiece is a language-independent tokenizer, subword generator, and detokenizer designed for neural-based text processing (Kudo and Richardson, 2018). In comparison, WordPiece which was popularized by BERT, starts its tokenizing from individual characters and merges them to form a larger token. SentencePiece however contains subword segmentation tools, where it is assumed the input is pre-tokenized into word sequences. SentencePiece is able to train subword models directly from raw sentences, which allows for making purely end-to-end that does not depend on any language-specific processing.

SentencePiece has two subword segmentation algorithms, byte-pair encoding (BPE), and Unigram language model, with the extension of direct training from raw sentences, which we also applied in this research. (Kudo and Richardson, 2018) In the training of Sentence Piece, the training and segmentation complexity of the unigram language model are heavily dependent on the size of input data. (Kudo and Richardson, 2018) discuss the application of SentencePiece in Neural Machine Translation from Japanese to English to a great extent, and show that SentencePiece is capable of proper segmentation of tokens and subtokens from raw data directly, and is fast enough to be applied in practice.

2.4. Previous research on token-POS fusion

In recent years there have also been applications of feature fusion for various ML-related tasks, such as automatic cyberbullying detection. For example, (Ptaszynski et al., 2019) proposed a method of detecting cyberbullying by applying a combinatorial algorithm resembling a brute-force search algorithm to automatically extract sophisticated sentence patterns and used a similar idea of fusing various types of features, including tokens with POS, in text classification of cyberbullying entries. The researchers used nine different feature sets and found that words/tokens and lemmas with part-of-speech information were the best of options for cyberbullying detection.

Similarly, (Ptaszynski et al., 2017) applied a fusion not only with tokens but also with semantic features and applied it in the cyberbullying detection. The detection was not based on harmful vocabulary lists but on the deep structure represented by both morphological and semantic information. The study used a combinatorial approach to obtain traditional word patterns, such as n-grams, as well as more sophisticated patterns with disjointed elements.

Additionally, in an initial study with simple Neural Networks, specifically Convolutional Neural Networks (CNN), (Eronen et al., 2021) used various combinations of grammatical information called Feature Density, to be processed with CNN. In the study, (Eronen et al., 2021) mention that using stopwords removal and POS tagging resulted in positive improvements for cyberbullying detection in Japanese.

Finally, some hints to further improvements and applications of token-POS fusion were laid out by (Eronen et al., 2022a), who trained Word2Vec Skip-Gram embeddings with encoded linguistic information. While POS did not perform well on SVM, the researchers showed that using word embedding on CNN did improve the performance. This opened the door to further applications, namely, that the embedding of fused tokens and POS features could also be performed with success on a more complex model such as the transformer model. (Eronen et al., 2022a)

3. Token and POS Fusion Method Description

In this section, we describe the method for the fusion of tokens with and Part-of-Speech used in this research.

3.1. Data Collection and Cleaning

We began by gathering dumped data from Wikipedia [2] and Book Corpus (Zhu et al., 2015) from the Internet. In order to imitate the similarities of the original ELECTRA models as closely as possible and make the later comparison fairer. However, the raw data contained unwanted information, such as pictures, links, and metadata embedded in XML and HTML formats. From the raw data, we eliminated all unwanted information and would be left with only text data.

[1] https://spacy.io/models/en

[2] https://dumps.wikimedia.org/

Word	POS	
Original Method	*to use*	VERB
Proposed Method	*to use*	δ

Table 1: Comparison between the original POS tagging method and the proposed method

Symbol	Part-of-Speech
α	PROPN
β	AUX
δ	VERB
ε	ADP
φ	ADJ
γ	ADV
η	CONJ
ι	CCONJ
κ	DET
λ	INTJ
μ	NOUN
π	NUM
ϑ	PART
ρ	PRON
σ	PUNCT
τ	SCONJ
υ	SYM
ω	X (Others)
ξ	SPACE

Table 2: POS to Greek symbol conversion list.

By using WikiExtractor [3] a program to clean the raw data collected from Wikipedia, we are able to clean the data from the unwanted information and leave only raw text data to be processed later on.

3.2. Token and Part-of-Speech Fusion Process

The main novel idea in this research is to help the language model take advantage of the POS information, by universally fusing POS information on the level of pretraining. To do this, we POS-tagged each word in the dataset. POS tagging was done using a POS tagger by SpaCy, which provides automatic Part-of-Speech tagging modules for many languages.

To seamlessly fuse Parts-of-Speech with tokens, we decided to use Greek symbols, in replacement of all original part-of-speech tags. Table 1 shows the difference between typical part-of-speech tags and proposed tags. The reason for this replacement is explained in section 3.3., below, where the results of a preliminary experiment show that using Greek symbols provided fewer incorrect subtokens.

Below, we show on a full sentence the final desired output where the words and Parts-of-Speech are fused together using Greek letter replacements.

Original Sentence:

"We will write a custom Essay on The History of Rice in Japan specifically for you"

POS-fused Sentence:

"Weρ willβ writeδ aκ customμ Essayα onε Theκ Historyμ ofε Riceα inε Japanα specificallyγ forε youρ"

[3] https://github.com/attardi/wikiextractor

Raw fused compound	dancingNOUN
Correct Token	dancing
Bad Token	dancingNOUN
Bad Token	NOUN
Correct Subtoken	##NOUN
Bad Subtoken	##ingNOUN

Table 3: Example of desired and undesired segmentation.

Type of Segmentation	Example
No space	centralADJ / ADJcentral
Underscore	is_AUX / AUX_is
Full-width underscore	recorded_VERB / VERB_recorded
Greek symbol	Japanα / αJapan

Table 4: Types of Segmentation

Table 2 below contains the list showing which part of speech is represented by which Greek character.

3.3. Tokenization

While the original ELECTRA uses WordPiece tokenizer for the tokenization process (Clark et al., 2020), we noticed that WordPiece does not tokenize the data as required. Therefore instead, we used the Unigram language model from the SentencePiece (Kudo and Richardson, 2018) tokenizer. We were expecting the word and its Part-of-Speech information to be segmented in a way where the former becomes the token and the latter becomes subtoken, respectively. SentencePiece Unigram was the method producing the fewest number of bad tokens.

Table 3 shows examples of desired and undesired segmentation outputs. In the table, we show examples of potential bad tokens and bad subtoken. We also show the desired correct token and subtoken segmentation, where the words and the POS tags have been properly separated during the tokenization process.

Since ELECTRA uses WordPiece tokenization similar to BERT model, it is a different format to SentencePiece Unigram. In order for ELECTRA to be able to tokenize the data, the format has to be converted into one that is compatible with ELECTRA.

3.4. Pretraining

Instead of reusing or re-training the original model of ELECTRA, we decided to train a new model from scratch, following all the settings that have been set for the original ELECTRA small in order to make comparisons as fair as possible during evaluation.

The training took seven days in total, running it on a single Nvidia GTX1080TI GPU with 11 GB of memory and 1481MHz Base Clock.

4. Evaluation Experiments

4.1. Preliminary Experiment: Selecting Optimal Fusion Settings

In the first preliminary experiment, we compared various fusion settings to select the optimal method of token and POS fusion for tokenization. Specifically, we wanted to

Type of Segmentation	Correct Subtoken	Supposed Subtoken	Bad Token	Bad Subtoken
SentencePiece (Unigram) (Part-of-Speech + Token)				
Full Width Underscore	8	0	0	0
No Space	1	11	0	424
Symbol (Greek)	15	14	0	0
Underscore	1	0	0	0
SentencePiece (Unigram) (Token + Part-of-Speech)				
Full Width Underscore	8	0	0	0
No Space	9	0	28	13
Symbol (Greek)	16	0	0	0
Underscore	8	0	0	0
WordPiece (Part-of-Speech + Token)				
Full Width Underscore	2	16	0	0
No Space	3	15	0	1155
Symbol (Greek)	1	16	0	1179
Underscore	2	16	0	1179
WordPiece (Token + Part-of-Speech)				
Full Width Underscore	2	16	0	0
No Space	15	18	463	654
Symbol (Greek)	14	17	464	653
Underscore	2	16	0	0

Table 5: Results of Tokenizer

know for which fusion settings the tokenizer most desirably separates tokens and Parts-of-Speech during tokenization. We used two available tokenizers applicable for ELECTRA, namely, WordPiece and SentencePiece Unigram.

4.1.1. Experiment Setup

In the experiments, we used a small dataset containing 5000 words in English, that have been taken from the large dataset (Wikipedia + book corpus) in order to shorten the processing time, the dataset has been constructed by counting words up to 5000 words, from the starts of the large dataset. Every word has been tagged with Part-of-Speech using a different type of segmentation technique shown in Table4.. Once it has produced a vocabulary file for each type of segmentation, we needed to convert it into ELECTRA compatible format.

The goal of the experiment was to select the method of fusion that would allow for Parts-of-Speech and tokens to be properly separated during the tokenization process.

In general, there can be two ways of fusing Parts-of-Speech to the tokens. One where POS tags appear in front of the words and another where they are at the end of the words, as in the examples below:

- Token(segmentation)Part-of-Speech (TOK_POS)
- Part-of-Speech(segmentation)Token (POS_TOK)

As shown above, in between the token and part-of-speech there is a "segmentation" element, which marks the separation of the token and part-of-speech, so that during the tokenization process, it is properly separated. We considered a few segmentation methods that we compared during this experiment to find which is the most suitable for the transformer models. Table 4 shows the different types of part-of-speech to token fusion, that has been used in the experiments.

In Table 3 we can observe the types of bad tokens and bad subtokens that were not acceptable in our case.

4.1.2. Results

After tokenizing a subset of the whole dataset, the result shown in Table 5 shows the analysis of the tokenization process, where it contains a number of correct subtokens, supposed subtokens, bad tokens, and bad subtoken. In the table, the correct subtoken refers to the amount of subtoken that is acceptable for this research. Supposed subtoken refers to the number of tokens that were supposed to be a subtoken, but appeared as tokens. Bad tokens and bad subtoken refers to the number of tokens and subtokens that is not acceptable for this experiment.

In Table 5, it can be observed that the most correct subtoken appeared for the Greek symbol with SentencePiece Unigram tokenization and token + Part-of-Speech arrangement method. While for WordPiece tokenization method the best option to consider is to choose "no space" + part-of-speech arrangement. While this method still contains a high number of bad tokens and bad subtokens, it still does have a large number of correct subtokens.

We found out that the optimal combination of tokens and part of speech is to fuse the Token with a Greek symbol without any separator where the Greek symbol is being replaced for every Part of speech. SentencePiece Unigram did not produce any bad tokens or bad subtokens. Therefore this combination was the combination that we used for the training of the whole datasets on ELECTRA from scratch.

Two models (ELECTRA_POS and ELECTRA_Vanilla for comparison) were pretrained with the dataset containing Wikipedia and BookCorpus with and without part of speech fused with tokens. Both of pretraining took roughly 7 days to complete. The result of the pretraining and later finetuning on a downstream task is discussed in section 4.2..

4.2. Evaluation Experiment: Automatic Cyberbullying Detection

Two models were then evaluated on the cyberbullying dataset to see the differences between the grammatical information-fused ELECTRA model and the original model. Table 6 shows the result of ELECTRA_POS and ELECTRA_Vanilla after being trained on cyberbullying dataset.

The cyberbullying dataset is in the English language and contains approximately 300 thousand tokens. The number of harmful samples was significantly smaller compared to the amount of non-harmful samples, which is around 7% of the whole cyberbullying dataset. This mimics the similarities of the real-life amount of profanity encountered on SNS. (Eronen et al., 2022b).

The results show that ELECTRA infused with POS information achieved slightly higher scores compared to the non-POS-fused ELECTRA model without grammatical information. Due to extra information added to the ELECTRA model, the highest improvement that has been recorded in this case was for the recall with 0.6209 compared to the non-POS-fused ELECTRA model with only 0.5878.

While the improvement was not large for other metrics, this research shows that, with additional grammatical information, the performance of the ELECTRA model can be improved.

Model	Acccuracy	Precision	Recall	F1 Positive	F1 Negative	F1 Macro
ELECTRA_POS	0.9530	0.5133	0.6209	0.5620	0.9752	0.7686
ELECTRA_Vanilla	0.9503	0.5133	0.5878	0.5480	0.9737	0.7609

Table 6: ELECTRA_POS and ELECTRA_Vanilla

5. Conclusions and Future Work

In this paper, we analyzed that, the best option to embed grammatical information into an ELECTRA model, was to replace all part-of-speech tags with Greek symbols, and used SentencePiece Unigram Tokenizer with the arrangement of TokenPOS for the tokenization process in order for the tokenizer to properly separated its part-of-speech tags. We also found that the fusion of grammatical information in a transformer model such as ELECTRA has the potential to improve the recall of cyberbullying detection. While the differences were not large compared to the original version of ELECTRA, the POS fusion method could be further improved by tuning various hyperparameters, such as the learning rate of the ELECTRA model. We also plan to investigate the influence of the length of the sentences on the results, since POS fusion causes every raw token to always contain two tokens (token and POS-subtoken), thus sentences longer than the maximum length of the model are always truncated, often by half, which could also influence the results.

References

Clark, Kevin, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning, 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. *CoRR*, abs/2003.10555.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Eronen, Juuso, Michal Ptaszynski, and Fumito Masui, 2022a. Comparing performance of different linguistically-backed word embeddings for cyberbullying detection. *arXiv preprint arXiv:2206.01950*.

Eronen, Juuso, Michal Ptaszynski, Fumito Masui, Masaki Arata, Gniewosz Leliwa, and Michal Wroczynski, 2022b. Transfer language selection for zero-shot cross-lingual abusive language detection. *Information Processing & Management*, 59(4):102981.

Eronen, Juuso, Michal Ptaszynski, Fumito Masui, Aleksander Smywiński-Pohl, Gniewosz Leliwa, and Michal Wroczynski, 2021. Improving classifier training efficiency for automatic cyberbullying detection with feature density. *Information Processing & Management*, 58(5):102616.

Hinduja, Sameer and Justin Patchin, 2010. Bullying, cyberbullying, and suicide. *Archives of suicide research : official journal of the International Academy for Suicide Research*, 14:206–21.

Kudo, Taku and John Richardson, 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *CoRR*, abs/1808.06226.

Nohria, Ankita and Harkiran Kaur, 2018. Evaluation of parsing techniques in natural language processing. *International Journal of Computer Trends and Technology*, 60:31–34.

Pal, Alok Ranjan, Anupam Munshi, and Diganta Saha, 2015. An approach to speed-up the word sense disambiguation procedure through sense filtering. *arXiv preprint arXiv:1610.06601*.

Patchin, Justin and Sameer Hinduja, 2006. Bullies move beyond the schoolyard a preliminary look at cyberbullying. *Youth Violence and Juvenile Justice*, 4:148–169.

Ptaszynski, Michal, Pawel Dybala, Tatsuaki Matsuba, Fumito Masui, Rafal Rzepka, and Kenji Araki, 2010. Machine learning and affect analysis against cyber-bullying. *the 36th AISB*:7–16.

Ptaszynski, Michal, Pawel Lempa, Fumito Masui, Yasutomo Kimura, Rafal Rzepka, Kenji Araki, Michal Wroczynski, and Gniewosz Leliwa, 2019. Brute-force sentence pattern extortion from harmful messages for cyberbullying detection. *Journal of the Association for Information Systems*, 20(8):1075–1127.

Ptaszynski, Michal, Fumito Masui, Yoko Nakajima, Yasutomo Kimura, Rafal Rzepka, and Kenji Araki, 2017. A method for detecting harmful entries on informal school websites using morphosemantic patterns. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 21(7):1189–1201.

Ptaszynski, Michal, Fumito Masui, Taisei Nitta, Suzuha Hatakeyama, Yasutomo Kimura, Rafal Rzepka, and Kenji Araki, 2016. Sustainable cyberbullying detection with category-maximized relevance of harmful phrases and double-filtered automatic optimization. *International Journal of Child-Computer Interaction*, 8:15–30.

Ptaszynski, Michal E and Fumito Masui, 2018. *Automatic cyberbullying detection: Emerging research and opportunities: Emerging research and opportunities*. IGI Global.

Yang, Xuewen, Yingru Liu, Dongliang Xie, Xin Wang, and Niranjan Balasubramanian, 2019. Latent part-of-speech sequences for neural machine translation. *CoRR*, abs/1908.11782.

Zewdu, Alebachew and Betselot Yitagesu, 2022. Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data*, 9.

Zhu, Yukun, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler, 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *CoRR*, abs/1506.06724.

A Comparative Study of Claim Extraction Techniques Leveraging Transformer-based Pre-Trained Models

Anouar Nouri[1], Salar Mohtaj[1,2], Sebastian Möller[1,2], Tilman Lesch[3]

[1]Technische Universität Berlin, Berlin, Germany
[2]German Research Centre for Artificial Intelligence (DFKI), Labor Berlin, Germany
[3]Deloitte Consulting GmbH, Berlin, Germany

anouar.nouri@campus.tu-berlin.de
{salar.mohtaj | sebastian.moeller}@tu-berlin.de
tlesch@deloitte.de

Abstract

The fast propagation of fake news and false content presented as truth is a dire problem we face more and more in today's world. It considerably threatens social, political, and economic stability. Hence the effort to battle misinformation with fact-checking becomes a very crucial endeavor. The first step toward the automated fact-checking of a news article is the extraction of check-worthy sentences (i.e., claims). Although a number of deep learning models based on recurrent neural networks (RNNs) have been developed for the task of claim extraction, the reached performance is limited since RNNs suffer from two principal constraints: the necessity of labeled data and short-term memory. In this paper, we combine different claim extraction techniques leveraging four transformer-based pre-trained models including T5, PEGASUS, BERT, and BART, and study their performance using two data sets, ClaimksKG, and NewsClaims. We extensively investigate the effectiveness of both zero-shot and fine-tuned summarization in capturing claims from the article text. Furthermore, we base a set of claim extraction techniques on sentiment analysis. Through a rigorous evaluation of the conducted claim extraction experiments using ROUGE and BERTScore metrics, in addition to ClaimBuster as a baseline, we prove the effectiveness of the proposed transformer-based pre-trained models in extracting claims.

Keywords: Claim Extraction, Transformer, Summarization, Sentiment Analysis

1. Introduction

The amount of textual data in cyberspace, be it news articles, blogs, scientific papers, or ebooks, is increasing exponentially day by day (Chiplunkar and Fukao, 2020), and while there is no doubt about its positive value for society and the individual, this exponential increase also introduces some dire problems. An important recent problem is the fast propagation of fake news and false facts presented as the truth, this phenomenon is also known as misinformation or disinformation (Lewandowsky and Van Der Linden, 2021). Because of the severity and the danger of misinformation, there have been many efforts to halt its propagation with fact-checking. The key step in the process of fact-checking an online article lies in the extraction of its check-worthy sentences, (i.e., the claims). This step is usually performed manually by a team of fact-checkers, who would spend hours meticulously scanning through articles (Ufarte-Ruiz et al., 2020). This approach is made up of tedious, slow work, and it's therefore ineffective to halt the rapid spread of fake news, thus creating an urgency for technologies that reliably automate the task of claim extraction.

Tools like ClaimBuster (Hassan et al., 2017) incorporating deep learning architectures like RNNs, provided promising results, and helped improve the process. However, they still need a large amount of human-labeled training data which makes traditional claim extraction models limited. In addition, RNNs are constrained by short-term memory and computational inefficiency that affect the performance of models based on this architecture (Deng and Liu, 2018).

The transformer architecture, introduced by Vaswani and colleagues (Vaswani et al., 2017) addresses these limitations and has the potential to alleviate the necessity of manual labeling while delivering state-of-the-art results. Transformer-based language models show promising results on a wide range of NLP tasks from sentiment analysis (Geetha and Karthika Renuka, 2021) to harmful information identification (Mohtaj and Möller, 2022) and vocabulary evaluation (Jacobsen et al., 2022). Our goal in this paper is to study and compare the performance of four transformer-based pre-trained models, namely, PEGASUS (Zhang et al., 2020a), T5 (Raffel et al., 2020), BERT (Devlin et al., 2019) and BART (Lewis et al., 2020)) on the task of claim extraction.

One task that transformer-based pre-trained models excelled at, is the task of summarization. We recognize the overlap between summarizing and claim extraction since both tasks aim to extract specific information from a text. Thus, we hypothesize whether summarization approaches leveraging transformer-based models can improve the performance of claim extraction. Based on the fact that false claims are usually subjective and provocative, loaded with extreme sentiment to attract the attention of the reader (Pennycook and Rand, 2021), and paired with

the observation that transformer-based models also achieve state-of-art results in the task of sentiment analysis (Alaparthi and Mishra, 2021) we draw a second hypothesis postulating the effectiveness of claim extraction techniques that employ transformer-based language models fine-tuned on the task of sentiment analysis.

The rest of the paper is organized as follows; Section 2. reviews recent, relevant research on the task of claim detection and claim extraction. In Section 3. we highlight the main concepts and models that have been used in the paper. We describe the conducted experiments, the data sets, as well as the obtained results in detail in Sections 4. and 5., respectively. Finally, we conclude the paper and highlight some ideas as future works in Section 6..

2. Related Work

The first functional and performant claim detection tool called ClaimBuster has been introduced in 2017 (Hassan et al., 2017). ClaimBuster uses a supervised learning model trained on a human labeled-data set to classify sentences into three groups, Non-Factual Sentence (NFS), Unimportant Factual Sentence (UFS), and Check-worthy Factual Sentence (CFS). The first iteration of ClaimBuster was trained with a Support Vector Machine (SVM) model that classified the sentences based on a set of features (e.g., sentiment). This evolved later to a Bi-directional Long Short-Term Memory (Bi-LSTM) model which delivered better results. We adopt the later version of ClaimBuster as our baseline model to establish a comparative benchmark with the pre-trained models presented in this study.

In 2019, researchers at the University of Copenhagen took a different approach to claim detection and introduced an automatic fact-checking system that employs RNNs with weak supervision to rank sentences based on their "check-worthiness" (Hansen et al., 2019). The model accomplishes this by using semantic word embedding as well as syntactic to represent each token of a sentence. The training is done on a weakly labeled data set, in which labels were determined by an already established check-worthiness ranking platform. Although the model outperforms the first iteration of ClaimBuster, it inherits the limitation of incorporating an RNN, including narrow context awareness.

This limitation is alleviated through the integration of state-of-art transformer architecture, as demonstrated through the evaluation done in (Prabhakar et al., 2020). The authors created a novel homogeneous data set extracted from the FEVER data set (Thorne et al., 2018) and Wikipedia, in an attempt to solve the problem of domain specificity. Additionally, the authors fine-tuned BERT and DistilBert (Sanh et al., 2019) with varying Data distribution on a binary classification task (if a sentence is a claim or not). The evaluation illustrates state-of-art results in claim detection.

All the above-mentioned research incorporated various approaches to tackle the task of claim extraction/detection from different angles with varying degrees of training. Generally, models built on transformer architectures and in particular BERT proved to be particularly effective. In this paper, we are making the next evolutionary step by studying different claim extraction techniques leveraging post-BERT state-of-the-art transformer-based pre-trained models in an attempt to eliminate the necessity of labeled data. We briefly highlight the main pre-trained models that have been used in this work in the next section.

3. Background

In this section, we discuss the main concepts and technologies that are used in the paper.

3.1. Claim Extraction

In this paper, the task of claim extraction consists of identifying the sentences that include verifiable assertions in a text corpus. An ideal NLP model for claim extraction would be inputted a text corpus and would output exclusively its claims.

3.2. Transformers

Transformers are introduced in (Vaswani et al., 2017). The architecture replaces the recurrence approach by extensively relying on an attention mechanism to form global relationships between input and output, enabling it to retain the memory of the entire text input far exceeding RNNs. Additionally, the transformer is computationally more efficient than the sequential design of RNNs, since it allows for more parallelization.

3.2.1. BERT

BERT (Devlin et al., 2019) was the first transformer-based language model, revolutionizing natural language understanding (NLU), and laying the blueprints for the next generation of language models (Greco et al., 2022). BERT was pre-trained on an enormous amount of unlabeled textual data to abstract deep bidirectional language representations that simultaneously attend to right and left context in all layers. It allows fine-tuning for downstream tasks with just one additional output layer, yet it achieves state-of-art results on a range of NLP tasks surpassing trained task-specific architectures.

3.2.2. T5

T5 refers to the Text-to-Text Transfer Transformer. The model was developed by researchers at Google after rigorous experimentation and empirical comparison of various transfer learning approaches and architectures to unify a range of NLP tasks under a "text-to-text" problem (Raffel et al., 2020). This means reformulating NLP tasks like summarization, question answering, sentiment classification, and language translation, to name a few, in such a way that the same model can tackle all of these tasks by taking a text as input and generating a novel text as output.

3.2.3. BART

BART stands for Bi-directional Auto-Regressive Transformers. The sequence-to-sequence model is a denoising autoencoder that is pre-trained to reconstruct a corrupted text (Lewis et al., 2020). This corruption (i.e., noise) is

accomplished by a randomized noising function that creates distinct types of noise in the input sequence to train the model on constructing correct and meaningful texts. BART incorporates a bi-directional encoder, similar to BERT, and a left-to-right (unidirectional) auto-regressive decoder, similar to GPT. The researchers behind BART described it as most effective when fine-tuned on text generation tasks, like QA and summarization.

3.2.4. PEGASUS

The last transformer-based model that we incorporate in our experimentation is PEGASUS, which stands for Pre-training with Extracted Gap-sentences for Abstractive Summarization (Zhang et al., 2020a). Similar to T5 and BART, PEGASUS makes use of both the transformer encoder and decoder. Unlike T5 and BART which are generally pre-trained for NLU, PEGASUS incorporates a novel pre-training objective, specifically designed to optimize the model's ability to reconstruct the input's principal information, and thus the ability to create linguistically sound abstractive summaries.

4. Methodology

In this section, we briefly describe the used data set to fine-tune and test the proposed models. Moreover, we discuss different experiments that have been conducted to test the validity of the established hypotheses.

4.1. Data set

To conduct the study, we selected two reputable and well-researched data sets, ClaimsKG (Tchechmedjiev et al., 2019) and NewsClaims (Reddy et al., 2021). ClaimsKG is a large structured database that includes 33261 claims extracted globally from various highly reputable fact-checking websites. NewsClaims is a novel benchmark data set created to assess the performance of models on extracting claims in addition to claim attributes. The version we incorporated in our experimentation consists of 143 news articles with 889 claims.

4.2. Baseline

We employ ClaimBuster (Hassan et al., 2017) as a baseline to better evaluate the effectiveness of our proposed approaches. The Bi-LSTM version of ClaimBuster which has been used as the baseline is a binary classifier (i.e., determines if a sentence is a claim or not). For training, we refactor our training data by dividing the input text into individual sentences and adding a label (e.g., 1 if a sentence is a claim and 0 if not). The trained models on ClaimsKG and NewsClaims data sets are used to identify claims from the test sets of these data sets, respectively.

4.3. Experiments

Considering different summarization approaches and a potential link between claim extraction and the task of sentiment analysis, we design a wide range of experiments to analyze the performance of different pre-trained models on the task. In this sub-section, we present these experiments in detail.

4.3.1. Abstractive Summarization

In this series of experiments, we employ *PEGASUS* and *T5* to test the effectiveness of abstractive summarization leveraging transformer-based models in extracting claims. *PEGASUS* was chosen since it was specifically designed and pre-trained for the task of abstractive summarization. As for *T5*, it does not only achieve state-of-the-art results in summarization, but it is also a dynamic model that solves a range of sequence generation and classification tasks.
The experiment is divided into two phases. The first phase is zero-shot abstractive summarization. We simply regard the task of claim extraction as an abstractive summarization task and summarize input texts from the test sets of *ClaimsKG* and *NewsClaims*. During this phase, we use three versions of PEGASUS and T5 as follows:

- The original pre-trained model
- The model fine-tuned on *CNN/DailyMail* data
- The model fine-tuned on *XSUM* data

In the second set of experiments, we fine-tune *T5* and *PEGASUS* using the training set of *ClaimsKG* while varying the data size (on 10%, 20% and 40%) and the training set of *NewsClaims* while varying the number of epochs (5, 15 and 25), here we utilize the entire training set since the data set is small.

4.3.2. Extractive Summarization

To achieve extractive summarization, we conduct the experiment using the BertExtractive pipeline (Miller, 2019). BertExtractive generates summaries by utilizing pre-trained BERT to create sentence embeddings, and k-means clustering algorithms to cluster the embeddings and select the top k closest sentences to the cluster's centroids as the summary-worthy sentences. In this approach, we compute the average number of claims per article in each data set and assign it to k (i.e., the number of extracted sentences per summary). The process of this experiment is illustrated in Figure 1 (a).

4.3.3. Query-Oriented Summarization

Abstractive and extractive summarization fall under the category of generic summarization. To cover a broader range of summarization models, in this experiment we use query-oriented summarization, in which the summary text is chosen based on a set of queries. To generate the query, we employ KeyBERT[1] to extract a set of keywords from input texts. Subsequently, we employ CTRLsum (He et al., 2020), a transformer-based model built on top of BART to generate controlled summaries based on the queries (i.e., keywords).

4.3.4. Sentiment Analysis

Based on the fact that claims can be subjective and provocative (Pennycook and Rand, 2021), we employ *Bert-Sentiment*[2], a pre-trained BERT model fine-tuned on scoring the sentiment of textual input.

[1] https://maartengr.github.io/KeyBERT
[2] https://huggingface.co/nlptown/bert-base-multilingual-uncased-sentiment

For each article, we simply compute the sentiment score of the sentences using the *Bert-Sentiment* model. As it is assumed that sentences with extreme sentiments (e.g., very positive or very negative) contain claims, we designed three different scenarios to extract claims. In the first scenario, we extract the top 3 negative sentences, while the top 3 positives have been extracted in the second scenario. Finally, we choose the top two positive and the top two negative sentences in the last scenario. The overall procedure of this experiment is shown in Figure 1 (b).

5. Results

In order to assess the efficacy of the various approaches utilized, we established the claims extracted from our test sets as the gold standard and employed both ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2020b) metrics to evaluate semantic and syntactic similarity. This section details the obtained results and provides the key findings.

5.1. Results on Summarization Experiments

The achieved performances of each model on *ClaimsKG* and *NewsClaims* data sets are presented in Tables 1 and 2, respectively. Almost all the proposed approaches outperform the baseline result from the *ClaimBuster* model in both data sets.

In general, the experiments based on fine-tuning abstractive summarization models show better performance results compared to the other experiments. The best overall model is the fine-tuned *PEGASUS-LARGE*. Fine-tuning proved very effective, especially on ClaimsKG. Regarding approaches that require no labeled data, leveraging BertExtractive for extractive summarization performed best on NewsClaims, whereas query-oriented summarization with KeyBERT and CTRLsum performed best on ClaimsKG.

5.2. Results on Sentiment Analysis Experiments

The performances of the sentiment analysis approaches on each data set are shown in Tables 3 and 4.

All three extraction variations outperform the baseline by a relatively small margin. Based on the experiments, BERT-sentiment (Neg) (i.e., employing BERT to extract the top negative sentences) achieves the best scores on the *ClaimsKG* and performs similarly to BERT-Sentiment (Neg&Pos) on the *NewsClaims* data set.

6. Conclusion

The aim of this paper was to improve the performance of claim extraction techniques to halt the rapid spread of misinformation. To this end, we propose a set of approaches leveraging four pre-trained transformer-based models (T5, PEGASUS, BERT, and BART), and investigate their performance on two claim extraction data sets. We conducted experiments on zero-shot extractive, abstractive, and query-oriented summarization, in addition to sentiment-analysis-based extraction.

Our results demonstrate that fine-tuning T5 and PEGASUS even on a small portion of the data set (e.g., 10% of ClaimsKG) drastically improves the model's performance. Following the success of PEGASUS on ClaimsKG, we release PEGASUS-ClaimsKG[3] a state-of-the-art transformer-based pre-trained model fine-tuned on extracting claims from fact-checking articles.

Future works can focus on designing and pre-training a transformer-based model tailored to the task of claim extraction. This will aid fact-checking immensely in halting the rapid spread of misinformation, mitigating its detrimental consequences on society.

Acknowledgment

The work presented in this paper has received funding from the German Federal Ministry for Economic Affairs and Climate Action (BMWK) through the project OpenGPT-X (project no. 68GX21007D).

References

Alaparthi, Shivaji and Manit Mishra, 2021. Bert: A sentiment analysis odyssey. *Journal of Marketing Analytics*, 9(2):118–126.

Chiplunkar, Niranjan N and Takanori Fukao, 2020. *Advances in Artificial Intelligence and Data Engineering: Select Proceedings of AIDE 2019*, volume 1133. Springer Nature.

Deng, Li and Yang Liu, 2018. A joint introduction to natural language processing and to deep learning. In *Deep learning in natural language processing*. Springer.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the ACL: NAACL-HLT 2019, Minneapolis, MN, USA*.

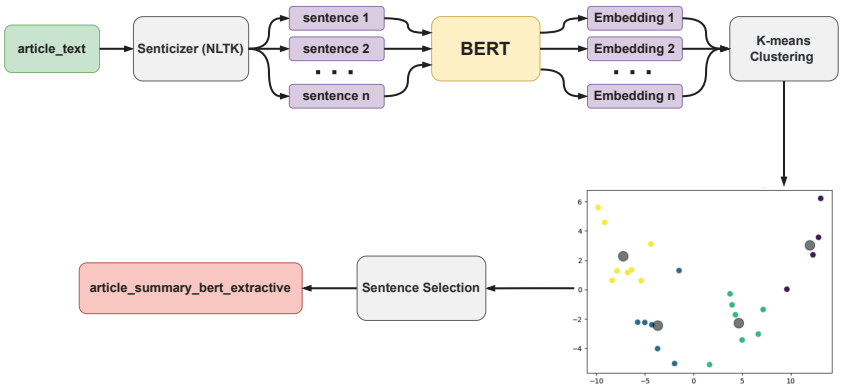
Geetha, M.P. and D. Karthika Renuka, 2021. Improving the performance of aspect based sentiment analysis using fine-tuned bert base uncased model. *International Journal of Intelligent Networks*, 2:64–69.

Greco, Candida Maria, Andrea Tagarelli, and Ester Zumpano, 2022. A comparison of transformer-based language models on nlp benchmarks. In *International Conference on Applications of Natural Language to Information Systems*. Springer.

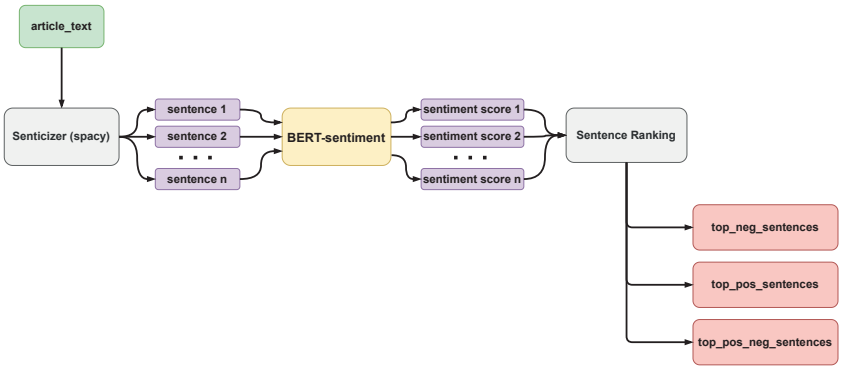
Hansen, Casper, Christian Hansen, Stephen Alstrup, Jakob Grue Simonsen, and Christina Lioma, 2019. Neural check-worthiness ranking with weak supervision: Finding sentences for fact-checking. In *Companion Proceedings of the 2019 World Wide Web Conference*.

Hassan, Naeemul, Fatma Arslan, Chengkai Li, and Mark Tremayne, 2017. Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*. ACM.

[3] https://huggingface.co/Cosmos/PEGASUS-ClaimsKG



(a)



(b)

Figure 1: The diagram of (a) zero-shot extractive summarization and (b) the sentiment analysis approaches

Experiment	Model	ROUGE	BERTScore
Baseline	ClaimBuster	0.150	0.542
Abstractive summarization (zero-shot)	T5-base	0.304	0.629
	T5-CNN/DailyMail	0.292	0.621
	T5-XSUM	0.264	0.564
	PEGASUS-LARGE	0.212	0.574
	PEGASUS-CNN/DailyMail	0.281	0.614
	PEGASUS-XSUM	0.333	0.667
Abstractive summarization (fine-tuning)	T5-BASE (10% of data)	0.713	0.856
	T5-BASE (20%)	0.724	0.863
	T5-BASE (40%)	0.731	0.865
	PEGASUS-LARGE (10%)	0.710	0.855
	PEGASUS-LARGE (20%)	0.718	0.859
	PEGASUS-LARGE (40%)	**0.733**	**0.866**
Extractive summarization	BertExtractive	0.356	0.638
Query-Oriented Summarization	CTRLsum (Keybert)	0.428	0.693

Table 1: *ROUGE* and *BERTScore* scores on the ClaimsKG data set

He, Junxian, Wojciech Kryscinski, Bryan McCann, Nazneen Fatema Rajani, and Caiming Xiong, 2020. Ctrlsum: Towards generic controllable text summarization. *CoRR*, abs/2012.04281.

Jacobsen, Anik, Salar Mohtaj, and Sebastian Möller, 2022. Mulve, A multi-language vocabulary evaluation data set. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France*. European Language Resources Association.

Lewandowsky, Stephan and Sander Van Der Linden, 2021. Countering misinformation and fake news through inoculation and prebunking. *European Review of Social Psychology*, 32(2):348–384.

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer, 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the ACL*

Experiment	Model	ROUGE	BERTScore
Baseline	ClaimBuster	0.206	0.505
Abstractive summarization (zero-shot)	T5-base	0.273	0.571
	T5-CNN/DailyMail	0.273	0.593
	T5-XSUM	0.165	0.576
	PEGASUS-LARGE	0.338	0.627
	PEGASUS-CNN/DailyMail	0.336	0.614
	PEGASUS-XSUM	0.196	0.586
Abstractive summarization (fine-tuning)	T5-BASE (5 Epochs)	0.373	0.654
	T5-BASE (15 Epochs)	0.385	0.664
	T5-BASE (25 Epochs)	**0.417**	0.663
	PEGASUS-LARGE (5 Epochs)	0.317	0.632
	PEGASUS-LARGE (15 Epochs)	0.405	**0.667**
	PEGASUS-LARGE (25 Epochs)	0.407	**0.667**
Extractive summarization	BertExtractive	0.378	0.646
Query-Oriented Summarization	CTRLsum (Keybert)	0.171	0.572

Table 2: *ROUGE* and *BERTScore* scores on the NewsClaims data set

Model	ROUGE	BERTScore
ClaimBuster	0.150	0.542
BERT-Sentiment (Neg)	**0.200**	**0.564**
BERT-Sentiment (Pos)	0.158	0.554
BERT-Sentiment (Neg&Pos)	0.175	0.558

Table 3: Results on the *ClaimsKG* data set based on sentiment analysis experiments

Model	ROUGE	BERTScore
ClaimBuster	0.206	0.505
BERT-Sentiment (Neg)	0.315	**0.604**
BERT-Sentiment (Pos)	0.258	0.581
BERT-Sentiment (Neg&Pos)	**0.322**	0.601

Table 4: Results on the *NewsClaims* data set based on sentiment analysis experiments

2020. Association for Computational Linguistics.

Lin, Chin-Yew, 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics.

Miller, Derek, 2019. Leveraging BERT for extractive text summarization on lectures. *CoRR*, abs/1906.04165.

Mohtaj, Salar and Sebastian Möller, 2022. On the importance of word embedding in automated harmful information detection. In *Text, Speech, and Dialogue - 25th International Conference, TSD 2022, Brno, Czech Republic*. Springer.

Pennycook, Gordon and David G Rand, 2021. The psychology of fake news. *Trends in cognitive sciences*, 25(5):388–402.

Prabhakar, Acharya Ashish, Salar Mohtaj, and Sebastian Möller, 2020. Claim extraction from text using transfer learning. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*. Indian Institute of Technology Patna, Patna, India: NLP Association of India (NLPAI).

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu, 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Reddy, Revanth Gangi, Sai Chinthakindi, Zhenhailong Wang, Yi R Fung, Kathryn S Conger, Ahmed S Elsayed, Martha Palmer, and Heng Ji, 2021. Newsclaims: A new benchmark for claim detection from news with background knowledge. *arXiv preprint arXiv:2112.08544*.

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf, 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Tchechmedjiev, Andon, Pavlos Fafalios, Katarina Boland, Malo Gasquet, Matthäus Zloch, Benjamin Zapilko, Stefan Dietze, and Konstantin Todorov, 2019. Claimskg: a knowledge graph of fact-checked claims. In *International Semantic Web Conference*. Springer.

Thorne, James, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal, 2018. FEVER: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA*. Association for Computational Linguistics.

Ufarte-Ruiz, María José, Belén Galletero-Campos, and Ana María López-Cepeda, 2020. Fact-checking, a public service value in the face of the hoaxes of the healthcare crisis. *Blanquerna School of Communication and International Relations*, 1(47):87–104.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Zhang, Jingqing, Yao Zhao, Mohammad Saleh, and Peter J.

Liu, 2020a. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*. PMLR.

Zhang, Tianyi, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi, 2020b. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

User Experience aspects of the WordNet-based digital asset search enhancement

Jędrzej Osiński[1], Daniel Rimoli[2]

[1,2]Department of Content & Experience Platforms
Wunderman Thompson Technology
{[1]jedrzej.osinski, [2]daniel.rimoli}@wundermanthompson.com

Abstract

The Princeton WordNet has been successfully used for years to support advanced semantic analysis in various domains. In this paper we propose and discuss how this tool can be looped into digital asset search engines to ensure quality of the results, but also to improve overall user experience (UX) allowing end-users to navigate across their collections easier and more consciously. We will also explain how this enhancement can also support users with limited domain knowledge and language competences, following diversity and inclusion assumptions of today's business applications.

Keywords: WordNet, ontology, digital assets, search engine.

1. Introduction and the solution context

The Princeton WordNet, as introduced in (Miller, 1995) and (Fellbaum, 1998), has been successfully applied in various scientific and commercial projects. The lexical database that allows to identify semantic relations between words can be used in complex reasoning algorithms to withdraw some more advanced conclusion based on communications in a natural language, e.g., in a security domain (Vetulani et al., 2010). It also inspired similar WordNet-like structures of different sizes and applications, e.g., PolNet (Vetulani, 2014), plWordNet (Dziob and Piasecki, 2018), or spatio-temporal ontology (Osiński, 2014).

The digital asset collections are important resources especially for Internet companies which prepare global campaigns on regular basis or use content management systems (CMS) to often refresh their Internet presence and respond quickly to market changes. These collections may contain hundreds of thousands of illustrations, images, photos. Also, they are not uploaded or edited by a single person, but usually by, sometimes geographically and culturally distributed, content-entry teams. The group of users (who do not upload assets but actively use them in their daily work to prepare campaigns, etc.) is even bigger, up to a few hundred people. Finally, the end-users who consume the content on the Internet are practically unlimited and often difficult to be categorized as a single target group.

On the other hand, the language we all speak individually, even if it is still English, is unique, as it reflects on our education level, social background, life experience, health condition and various psychological aspects (Pennebaker, 2003). We could even say the language we speak is almost as unique as our fingerprints.

Now the above leads to a significant challenge: how to make sure that assets named or tagged by one person will be successfully found by other people potentially using slightly different search queries (that reflect their own language and perception). Let us consider a simple example to illustrate it: suppose an editor uploads two images of cars and name them "limousine" and "auto" accordingly. Now an end-user looks for an image of a car for his online campaign. He starts to type different search queries: "car", "motorcar". Finally, he types in "auto" and gets one result. The limousine is not found. It is even more difficult if we consider different breeds of dogs. In that case two things can happen:

1. A user types in "dog" and focuses only on the results found. That means dozens of potentially even more appropriate images bought for the collection were never considered and there is a chance they will never be used.
2. A user spends a longer while trying different queries, searching for breeds of dogs over the internet and using their names as a search query.

Both above are not the most efficient approaches and so we proposed and implemented a digital asset search enhancement that allows a user to type a search query and active a WordNet-based extension to look for its synonyms, hyponyms (more specialized term), hypernym (more general terms) and meronyms (part of it) - see Fig.1.

The goal of this paper is present and discuss our plugin-like search extension which mainly focuses on improving users' ergonomics and daily efficiency. We understand these expectations as the ability to find the digital assets quicker, more intuitively, and with less prompts needed. The solution would also lower the level of language skills and expertise knowledge required to benefit from the domain-specific search features. All these we refer to as User Experience (UX) in the further parts of the text.

The next section describes technical aspects while sections 3 and 4 explain the use cases and their business and UX justifications.

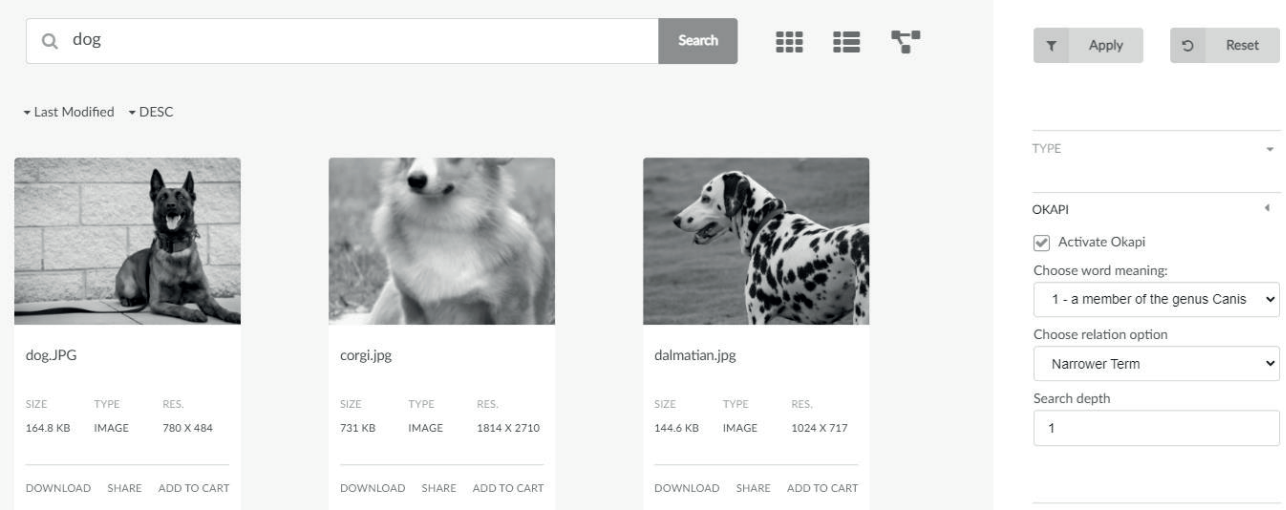


Figure 1: The search engine enhancement user panel (with the config on the right).

2. Solution implementation

2.1. High-level architecture

Okapi is a service that acts as an abstraction layer for the WordNet structure, exposing methods aimed at expanding a given search term, according to a choice of semantic relation. The interaction between Okapi and the actual WordNet database is achieved with the use of a third-party library called extJWNL (extJWNL, 2022). This library allows Okapi to query the WordNet database for all meanings (synsets) of a given word, as well as the textual definition ("gloss") and all semantic relations ("synonyms", "hyponym", "meronym", etc) for each of those meanings of the original word. The search for semantic relations also depends on the "search depth", which determines how far from the original word to traverse the WordNet tree structure to find results – deeper searches will yield more results, which are less closely related to the original word, while shallower searches will guarantee closer results, in lesser numbers. The library is also capable of working with a local instance of the WordNet database, which is then treated as a static database that can be optionally maintained and customized (e.g. extended by a solution-specific taxonomy branch, like branded products names) to the specific requirements of any application.

2.2. Implementation details

Okapi is composed of three layers – see Fig. 2:

1. WordNet Service: a standalone backend service written using the Kotlin language, that manipulates the Dictionary exposed by the extJWNL library and is responsible for the actual searches for words, retrieving whole "synset" entries from wordnet to be further processed by the next layer. This layer has no client-facing interface and is only used internally by Okapi.
2. HTTP Service: a Java service class which receives requests from clients (through one of the interfaces described below), communicates with WordNetService to obtain the necessary "synset" entries from the WordNet database and returns to clients lists of meanings or related words, ready to be used for extended searches.
3. Search Provider: an Adobe Experience Manager (AEM, 2022) integration layer, written in Java and HTML/JavaScript, that is used to extend AEM's native asset search to seamlessly incorporate semantic relations into search results. This is further detailed in section 2.4.

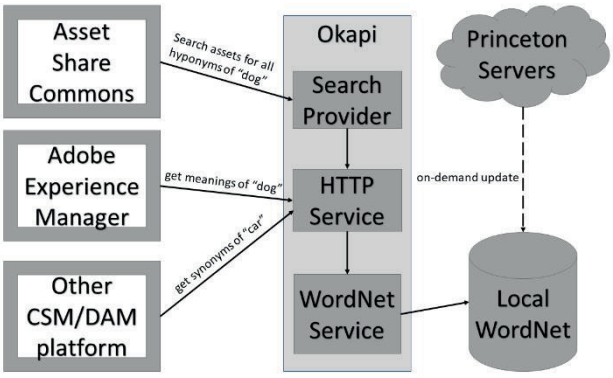


Figure 2: High level diagram of Okapi and example communications with external clients and servers

2.3. Client interface

The Okapi Service exposes two public methods and one REST API to allow client applications to retrieve meanings and semantic relations from the WordNet database in an easy and straight-forward manner:

- getGlosses: This public method is used to query all known meanings of a specific word. The only input parameter is the actual word to be used as a search term

and the output is a list of strings, each containing one known meaning of the word (synset definition), e.g.: passing "car" as the input parameter will yield in return meanings such as "a motor vehicle with four wheels; usually propelled by an internal combustion engine" and "a wheeled vehicle adapted to the rails of railroad", among others.

- getRelatedWords: Once the user has chosen one particular meaning of the word that was returned by the previous method (get Glosses), this method may be used to retrieve all the words that are related to that meaning, according to a choice of semantic relation, e.g.: passing "car" as an input parameter, selecting the second meaning listed above, choosing the "similar" semantic relation and indicating a search depth of "1" causes words such as "car", "railcar", "railway car" and "railroad car" to be returned in response.
- /apps/okapi/gloss: This REST API is an alternative interface to obtain the same results of the "getGlosses" method, to allow easier integration with different types of client applications.
- /apps/okapi/savanna: This REST API returns the full taxonomy of the asset collection, organized in the form of a tree, following the existing categories in the Wordnet database and including all the synsets for which there is at least one matching asset.

2.4. AEM integration

In addition to the general-use service described above, Okapi has already been successfully integrated as a plug-in feature within the Adobe Experience Manager content editing mode and in Asset Share Commons, Adobe's open-source asset share reference implementation built on AEM (Adobe, 2022). This allows asset searches performed in an AEM instance to optionally receive results based on extended semantic relations, while still keeping Okapi unaware of metadata or any specific attributes of the assets. This is also possible to integrate in similar way with other WordNet-like lexical databases, so from the architecture perspective Okapi can be simple extended across different languages and applications.

This integration is composed of two parts:

1. OkapiSearchProvider: implementation of the AEM SearchProvider interface that intercepts all asset searches, expands the provided search term according to the indicated semantic relation and includes in the results all the assets matching the extended search terms. Search queries must include, beyond the usual search term, a choice of semantic relation, the depth of the search in the WordNet tree and which meaning of the word to consider for the search.
2. Modified asset search page: this page, which is part of the AEM content editing mode, has been modified to allow users to visualize the list of meanings of a word and choose one of the supported options of semantic relation. When searching for assets while having Okapi activated, it will include the new values described above in the query parameters of the URL.

2.5. Further integrations

Due to its open nature, Okapi can also easily be integrated with any other asset management tools and services with little or no extra implementation. Any number of external services like CMSs, mobile apps, web pages, and others can communicate with the service through its multiple interfaces to boost their functionalities. Of course, other API libraries and WordNet-through search mechanisms can be also considered as an alternative approach, e.g., the query language introduced in (Kubis, 2016).

3. Use cases for WordNet relations

The semantic relations between words are the key feature of the WordNet what allows advanced analysis. They were explained carefully, e.g., in (Jurafsky and James, 2008) and (Saeed, 2003). Thus, in the following sub-sections we will not focus on these relations themselves but on the potential use cases they can solve and how they can improve the user experience of a search tool. We will place our examples in the e-commerce, business-to-business (B2C) context, however the reader is welcome to consider analogical scenarios in various further areas, where precise search is required to bypass the overwhelming amount of the information provided (check the discussion within (Vetulani and Osinski, 2017)).

3.1. Hyponymy

Hyponymy is a semantic relation between subtype and supertype, in other words it lists more specific terms that still fit within the supertype definition. Hyponyms for a car are different types of cars, like limousine, cabriolet or crossover. Allowing a user to display all the hyponyms for the search query provides much more valuable content to be displayed. Moreover, it makes the assets more accessible as users do not have to be domain experts anymore to search through the assets efficiently. For example, a user does not need to have knowledge of cynology to easily display dogs of various breeds. Moreover, this feature not only supports users, but also brings some educational benefits and assets collection awareness in general.

3.2. Hypernymy

Hypernymy is an opposite relationship that returns a supertype for a given term. This may be very useful when a potential business campaign refers to types of objects without very exact constraints. In that case a user can type in e.g., "tiger" and then easily extend the search using the user interface tab to display other species of big cats or moving further various carnivores without extra knowledge and deep consideration. In that case a user can put the effort into creative work instead of search query optimization.

3.3. Meronymy

Meronymy is a semantic relation between a part and a whole. It can be considered on different levels, e.g., the part is a physical element of the whole, or a member of a group (e.g., "singer" as a meronym of "choir") or a substance ("cellulose" as meronym of "paper"). In any of these, there is a huge variety of e-shop applications. If we consider a sport-related service and a user looking for a "bike", that person can immediately extend the search (or be automatically suggested to check) to see additional elements (meronyms) that are worth to be considered too (e.g., sprocket, chain, pedals, handlebar). As another example, the substance meronymy can be applied to support more conscious users to search for ecologic or more healthy products.

3.4. Similarity

Similarity relation can be calculated using various measures as reviewed in e.g. (Meng et al., 2013) and easily accessed via many libraries like in (Kubis, 2015). Again, this can be extremely helpful to the creative teams preparing digital campaigns as this relation may propose to users some unobvious but interesting assets categories. Thus, users could treat this UI option as a kind of an inspiration boost. Similarly, e-shop customers can be driven to find some similar products and offers even if they are not directly referring to an entered search query.

4. Additional benefits

In the previous section we introduced some use cases to present the solution application. However, its further benefits can be also looked at from wider perspectives and briefly summarized as follows:

- As described in section 2., the solution is platform independent and can be successfully re-used as an extension to various search engines and user interfaces.
- The taxonomy of an asset collection is built on the fly without any human input needed or advanced pre-processing. This is an important aspect as working on a proper and easy-to-navigate concepts taxonomy is a popular business challenge.
- The solution supports multilingual and culturally diverse companies and their target audience. The solution mitigates the chance for disambiguation when various phrases are used to describe similar concepts.
- Data accessibility. A user can successfully and efficiently find valuable assets even with limited vocabulary and limited domain knowledge. The learning aspect is an additional, positive side effect of this.
- The solution can be used to health-check the digital assets structure. The WordNet-based tree-like structure can be visualized graphically and so may be much easier to validate.

5. Comparison with visual auto-tagging systems

We have presented how the WordNet-based mechanism can be used to increase digital asset search accuracy. The alternative approach which is quite popular is to add many tags to assets based on automated image analysis that identifies some popular and key objects presented on the image assets. Currently one can choose between a few providers of such solution as services, including Microsoft's Azure Computer Vision (with early results presented in (Tran et al., 2016)), Amazon Web Services (AWS) or Google Cloud Vision. All of them are extensively used in many scientific and commercial applications, incl. solutions for visually impaired people (Naik, 2021) or human posture improvement (Ho and Ismail, 2021). However, although these services may seem out of the box and the default approach, our proposal has a few clear advantages that are worth highlighting:

- Visual tagging solutions have some known tech limitations (e.g. (Microsoft, 2022)), while our proposal reflects exact assets author's input. The content is not missed.
- Sometimes tens of visual tags are assigned to a single asset what increases the risk of losing precision and search results density. It is not the case in the proposed solution.
- Visual tagging systems have a limited taxonomy with a limited number of conceptual tags which is good for general purposes but may not be enough for specialized content. Our approach is based on 120.000 synsets so contains huge variety and range of language concepts and thus may be more efficient in managing content-rich and domain-specific assets.
- Visual tagging systems are designed to analyze images and partially videos. The proposed solution is independent from the data format and can be used to search for images, videos, sounds, documents, etc.
- Visual tagging systems work with visible physical objects only. The WordNet-based solution can be applicable for any element or idea (incl. objects that are invisible, abstract, technical, or emotions, etc.).
- The proposed solution is based on the free libraries. The Princeton WordNet is available for commercial use.

6. Conclusions

In this paper we presented a digital assets search enhancement. Currently Okapi is a working prototype, and the next stage is to implement it as a part of a commercial project, to collect and address feedback on its UX and performance during a long-term and full-scale business usage. The business use cases described above should be treated as easy-to-follow examples rather than complete and

extensive lists. In fact, the solution can be successfully applied in all domains wherever an extra search filtering is crucial functionality, like digitized historical archives, medical databases, museum artifacts records, and many more. Looking wider, this approach could be used also for various automated health-checks, including the assets (incl. documents) collection structure and taxonomy automated verification. Finally, mitigating cultural and linguistic differences when a distributed team manages huge collections of assets, as well as enhancing diversity and inclusion by supporting users for whom English is not a native language are also crucial values that should not be underestimated or omitted in the summary.

References

Adobe (2022): *Asset Share Commons: a modern, open-source asset share reference implementation built for adobe experience manager as a cloud service.* (https://opensource.adobe.com/asset-share-commons – access 2022-09-30)

AEM (2022): *Adobe Experience Manager: A powerhouse combo for your content and digital asset management needs.* (https://business.adobe.com/products/experience-manager/adobe-experience-manager.html – access 2022-09-30)

Agnieszka Dziob, Maciej Piasecki (2018): Implementation of the Verb Model in plWordNet 4.0. Proceedings of the 9th Global Wordnet Conference, Singapore, 8-12 January 2018.

extJWNL (2022): *Extended Java WordNet Library.* (https://github.com/extjwnl/extjwnl – access 2022-09-30)

Fellbaum, C. (Ed.). (1998). *WordNet: An Electronic Lexical Database*. MIT Press.

Ho, L.C. and Ismail, M.A. (2021): *Android Application for Posture Analysis using Tensorflow and Computer Vision*, 2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM), pp. 53-57.

Jurafsky, D., James, M. (2008): *Speech and Language Processing*. Financial Times Prentice Hall, 2008.

Kubis, M. (2015): *A semantic similarity measurement tool for WordNet-like databases*. Language and Technology Conference (pp. 155-168). Springer.

Kubis, M. (2016): *A Query Language for WordNet-like Lexical Databases*, International Journal of Intelligent Information and Database Systems, 9 (2), pp. 103–133, 2016, ISSN: 1751-5858.

Meng, L., Huang, R., & Gu, J. (2013). A review of semantic similarity measures in wordnet. International Journal of Hybrid Information Technology, 6(1), 1-12.

Microsoft (2022): *Computer Vision Documentation: Object detection.* (https://learn.microsoft.com/en-us/azure/cognitive-services/computer-vision/concept-object-detection – access 2022-09-30)

Miller, G.A. (1995). *WordNet: A Lexical Database for English*. Communications of the ACM Vol. 38, No. 11: 39-41.

Naik, K., Sawant, N., Kamat, G., Kandolkar, S., & Marchon, N. (2021). IRIS: An Application for the Visually Impaired Using Google Cloud API. In Advances in Signal and Data Processing (pp. 29-43). Springer, Singapore.

Osinski, J. (2014): The XCDC Relations as a Spatio-Temporal Ontology, [in:] Vetulani, Z., Mariani, J. (eds.): *Human Language Technology Challenges for Computer Science and Linguistics, 5th Language and Technology Conference, LTC 2011, Poznan, Poland, November 25-27, 2011, Revised Selected Papers*, Lecture Notes in Artificial Intelligence 8387, pp. 104-115, Springer.

Pennebaker, J. W., Mehl, M. R., & Niederhoffer, K. G. (2003). Psychological aspects of natural language use: Our words, our selves. Annual review of psychology, 54(1), 547-577.

Tran, K., He, X., Zhang, L., Sun, J., Carapcea, C., Thrasher, C., Buehler, C., & Sienkiewicz, C. (2016): *Rich Image Captioning in the Wild*, DeepVision workshop, Proceedings of CVPR 2016.

Saeed, J. (2003): *Semantics*, 2nd ed. Wiley-Blackwell.

Vetulani, Z., Marciniak, J., Obrebski, T., Vetulani, G., Dabrowski, A., Kubis, M., Osinski, J., Walkowska, J., Kubacki, P., Witalewski, K. (2010). Zasoby jezykowe i technologie przetwarzania tekstu. POLINT-112-SMS jako przyklad aplikacji z zakresu bezpieczenstwa publicznego (Language resources and text processing technologies. POLINT-112-SMS as example of public security application with language competence) (in Polish), Wydawnictwo Naukowe UAM, Poznan (ISBN 978-83-232-2155-5).

Vetulani, Z. (2014). PolNet – Polish WordNet. In: Vetulani, Z., Mariani, J. (eds) Human Language Technology Challenges for Computer Science and Linguistics. LTC 2011. Lecture Notes in Computer Science, vol 8387.

Vetulani, Z., Osinski, J. (2017): *Intelligent Information Bypass for More Efficient Emergency Management*, Computational Methods in Science and Technology, vol. 23 (2), pp. 105-123, PCSS Poznan.

Resources Creation of Bengali for SPPAS

Moumita Pakrashi[1], Brigitte Bigi[2], Shakuntala Mahanta[3]

[1]Centre for Linguistic Science and Technology, IIT Guwahati, Assam, India.
moumi176155103@iitg.ac.in
[2]Laboratoire Parole et Langage, CNRS, Aix-Marseille Univ., Aix-en-Provence, France.
brigitte.bigi@cnrs.fr
[3]Department of Humanities and Social Sciences, IIT Guwahati, Assam, India.
smahanta@iitg.ac.in

Abstract

The development of HLT tools inevitably involves the need for language resources. However, only a handful number of languages possess such resources for free. This paper presents the development of speech tools for the Bengali language. Particularly, this paper focuses on developing language resources of a tokenizer, an automatic speech system for predicting the pronunciation of the words and their segmentation in this low-resourced language. The newly created resources have been integrated into SPPAS software tool and distributed under the terms of public licenses.

Keywords: Human Language Technology, Automated Annotation, Less-Resourced Languages, linguistic resources

1. Introduction

The development of Human Language Technologies (HLT) tools is a way to break down language barriers. Only a handful of the approximately 7,000 languages of the world possess the linguistic resources required for implementing HLT technologies (Bigi, 2014). One needs to analyse a considerable amount of speech dataset to achieve reliable and consistent results from phonetic research. Even in languages where a huge speech dataset is available for research, analysing and annotating the data is a primary challenge. The basic problem is that analysing data (in the context of this paper, speech data) is very tedious and time-consuming task, even for a phonetician or a trained expert. The next problem is that differences in research objectives necessitate their corresponding analysis and annotation. In such a situation, automatic annotation of speech data becomes a primary requirement of phonetic research.

Currently, a number of tools are available for the purpose of automatically segmenting and aligning speech data with its corresponding transcription. Speech recognition engines like the open-source CSR-Engine Julius (Lee et al., 2001) or the licensed HTK Toolkit (Young and Young, 1993) can perform the task. Even some wrappers for HTK such as WebMaus - Automatic Segmentation and Labelling of Speech Signals over the Web (Kisler et al., 2017), P2FA - the Penn Phonetics Lab Forced Aligner (Yuan et al., 2008), and others[1] can make the task easier. Most of these tools require varying amounts of expertise in computer science - particularly those that require installing HTK to be able to operate, or they are not available across multiple platforms. But the SPPAS tool runs on multiple platforms, and the incorporation of a new language requires some bare minimum linguistic resources that can be easily handled by linguists (Bigi, 2015). Apart from the automated functions of phonetization, annotation in multiple formats (such as X-SAMPA, IPA), alignment and syllabification; prosodic analysis of utterances can also be performed using the Momel-INTSINT algorithm (Hirst and Espesser, 1993; Hirst, 2011) incorporated within SPPAS.

Bengali is one of the dominant languages in the Indian subcontinent that historically belongs to the Indo-Aryan (IA) family of languages. Spoken by almost 210 million people as their native or second language, it currently holds the seventh position among the world's languages[2]. Bengali is one of the official languages of India and the national language of Bangladesh. It is spoken primarily in Bangladesh and the Eastern Indian states of West Bengal, Tripura, parts of Assam. In this paper, we deal with the Standard Colloquial Bengali (SCB) variety which is spoken mostly in and around Kolkata. The Bengali script or Bangla alphabet (Bengali: বাংলা বর্ণমালা baŋla bɔrnamala) is the alphabet used to write the Bengali language. This writing system of Bengali has its origins in the Brahmi script, which is the source of all modern scripts of Indian languages (Klaiman and Lahiri, 2018).

This paper describes the process of implementing automatic annotation and analysis of Bengali speech using SPPAS software (Bigi, 2015). The SPPAS tool produces automatic segmentation, annotations, and analysis of a speech sound and its corresponding orthographic transcription. With that objective in mind, this paper describes the development of a corpus and

[1]For a list, see https://github.com/pettarin/forced-alignment-tools

[2]https://www.britannica.com/topic/Bengali-language

some language resources for Bengali. Such newly created linguistic resources were integrated into SPPAS for the multi-lingual automatic tokenizer (Bigi, 2014), the multi-lingual automatic speech system for predicting the pronunciation of words (Bigi, 2016) and for their segmentation (Bigi and Meunier, 2018).

2. Corpus description

In order to use standardised speech data of Bengali for creating necessary linguistic resources, we have used an open-source speech database created by Google to develop Text to Speech (TTS) systems. The data consists of audio recordings of short phrases/sentences, a pronunciation lexicon, and a phonology definition of Bengali[3]. All the data have been released under the Creative Commons Attribution 4.0 international license (CC-By-NC-4.0).

This set of data contains audio-transcript pairs. Audio recordings are in WAVE format. The accompanying line index.tsv file has the normalized transcript of the recorded audio and the ID of the corresponding audio file. The audio data was collected from a group of volunteers between the ages of 20 and 35. They were asked to read short sentences, each containing 5 - 20 words. The texts used for recording have been either extracted from Wikipedia or general websites or are declarative sentences created by native speakers of the language. The recording was conducted in a quiet environment: either a sound studio or a quiet room with a soundproof booth. Moreover, all audio files have passed through a QC process to ensure good audio quality, absence of background noise, and match between recorded audio and text transcript.

In order to convert each entry to consist of an audio file and its corresponding text transcript file, we used the "Fill in IPUs" automatic annotation of SPPAS to automatically detect the IPUs of each file. The result is a file indicating the time alignment of each sounding segment. This automatic annotation wasn't verified.

The corpus duration is 7291.82 seconds (2 hours) among 1366 audio files.

3. Phonetic description of Bengali

Bengali is an Eastern Indo-Aryan language. Phonemically, Bengali features 29 consonants and 7 vowels. However, the phonological alternations of Bengali vary greatly due to dialectal differences.

Among the corpus described in the previous section, 76 files were manually time aligned at the phoneme level. It represents 211.64 seconds including 36.9 of silences. Tables 3., and 1 indicate the phonemes both in SAMPA (Wells, 1997) and in the International Phonetic Alphabet, an example of a word and the number of occurrences in the manually aligned corpus. In this small part of the corpus, we observed 29 consonants and 11 vowels. We have also added those phonemes that had nil occurrence in our corpus such as ɖ, ɖʰ and nasal vowels primarily because of their prominent presence in Bengali vocabulary.

SAMPA	IPA	Example	Occ.
b	b	ব্রাত্য (*outcast*)	76
b_h	bʰ	ভদ্র (*polite*)	14
c	c	চকচকে (*shiny*)	14
c_h	cʰ	ছবি (*picture*)	40
d	d	দর (*rate*)	64
d_h	dʰ	ধনী (*the rich*)	10
d‘	ɖ	ডিম (*egg*)	0
d‘_h	ɖʰ	ঢালু (*slope*)	0
g	g	গণতন্ত্র (*democracy*)	30
g_h	gʰ	ঘটনা (*incident*)	12
k	k	কুটির (*cottage*)	140
k_h	kʰ	খেজুর (*dates*)	50
p	p	পার্থক্য (*difference*)	48
p_h	pʰ	ফেরা (*to return*)	10
t	t	তালিকা (*list*)	88
t_h	tʰ	থালা (*plate*)	20
t̀	ʈ	টোকা (*to copy*)	36
t_̀h	ʈʰ	ঠেকানো (*to prevent*)	2
dZ	d͡ʒ	জলন্ত (*burning*)	54
dZ_h	d͡ʒʰ	ঝাণ্ডা (*flag*)	0
f	f	ফান্ড (*fund*)	2
h	h	হাত (*hand*)	29
s	s	শামুক (*snail*)	6
S	ʃ	সহকর্মী (*colleague*)	88
m	m	মাস (*month*)	52
n	n	নিবন্ধ (*essay*)	115
N	ŋ	ধ্বংস (*destruction*)	4
l	l	লিপি (*script*)	78
r	r	রহস্য (*suspense*)	184
ṙ	ɽ	পড়া (*to study*)	4
j	j	হৃদয় (*heart*)	18
w	w	হওয়া (*to happen*)	2

a	a	আদর্শ (*principle*)	263
a~	ã	আঁকা (*to draw*)	2
e	e	এবার (*now*)	270
e~	ẽ	এঁকেবেঁকে (*twisted*)	6
i	i	ইচ্ছা (*wish*)	195
i~	ĩ	ইঁদুর (*rat*)	2
O	ɔ	অংশ (*part*)	70
O~	ɔ̃	পঁচা (*rotten*)	0
o	o	ওজন (*weight*)	194
o~	õ	ওঁৎ (*trap*)	0
u	u	উত্তর (*answer*)	87
u~	ũ	উঁচু (*high*)	2
{	æ	এক (*one*)	16
@	ə	জংশন (*junction*)	0

Table 1: Consonants and vowels of Bengali

[3]https://github.com/google/language-resources/tree/master/bn

4. Creating resources for HLT tools

4.1. Vocabulary, Pronunciation dictionary

The vocabulary list contained approximately 65000 lexical entries of Bengali, including many loan words written in English orthography. The dictionary entries provide a broad phonemic transcription of colloquial Bengali but of the Bangladeshi Standard Bengali. Therefore we *manually corrected* each of the lexical items in the list to suit our required Standard Indian variety of Bengali speech. This created a pronunciation dictionary of Bengali corresponding to the Standard colloquial Bengali speech variety of India.

4.2. Acoustic model

Acoustic models are Hidden Markov models (HMMs) created using the HTK Toolkit (Young and Young, 1993), version 3.4. HMM states are modelled by Gaussian mixture densities whose parameters are estimated using an expectation-maximization procedure. Acoustic models were trained from 16 bits, 16,000 Hz wav files for the corpus. The Mel-frequency cepstrum coefficients (MFCC) along with their first and second derivatives were extracted from the speech in the standard way (MFCC_D_N_Z_0). See (Bigi, 2012) for details. The training procedure is implemented into a Python script included in SPPAS.

The outcome of the training procedure is dependent on both: 1/ the availability of accurately annotated data; and 2/ on good initialization. The initialization of the models creates a prototype for each phoneme using time-aligned data. In the context of this study, this training stage has been switched off: it has been replaced by the use of phoneme prototypes already available in some other languages. The articulatory representations of phonemes are so similar across languages that phonemes can be considered as units which are independent of the underlying language (Schultz and Waibel, 2001). In SPPAS package, 10 acoustic models of the same type - i.e. same HMMs definition and the same MFCC parameters, are freely distributed with a public license so that the phoneme prototypes can be extracted and reused: English, French, Italian, Spanish, Catalan, German, Polish, Mandarin Chinese, Southern Min, Naija. To create an initial model for the Bengali language, most of the prototypes of English language were used, nasal vowels from French language and some from Southern Min, and Polish. The prototypes of noise and laughter were also added to the model to be automatically time-aligned. This approach enabled the acoustic model to be trained with the small amount of Bengali language speech data we collected (Le et al., 2008; Bigi et al., 2021).

5. HLT tools

5.1. Automatic tokenization, phonetization and forced-alignment

In recent years, the SPPAS software tool has been developed to produce annotations automatically. It proposes 23 automatic annotations of audio or video, including the ones for the alignment of recorded speech sounds with its phonetic annotation. The multilingual approaches that are proposed enabled us to make the automatic annotations of SPPAS available for the Bengali language. For this purpose, we created an archive containing the lexicon - a list of words, the pronunciation dictionary in HTK-ASCII format and the acoustic model; and we made it available on the SPPAS website. We also added their description in the SPPAS resources documentation. Figure 1 shows an example of the resulting automatic Text Normalization, Phonetization and Alignment of a Bengali speech segment when using SPPAS 4.7.

5.2. Experiments

Forced-alignment is the task of automatically positioning a sequence of phonemes in relation to a corresponding continuous speech signal. Given a speech utterance along with its phonetic representation, the goal is to generate a time-alignment between the speech signal and the phonetic representation.

Some experiments were conducted to evaluate the accuracy of the phoneme alignments. It was evaluated using the Unit Boundary Positioning Accuracy - UBPA that consists in the evaluation of the delta-times (in percentage) comparing manual phonemes boundaries with the automatically aligned ones. This *Delta* time is estimated on the beginning of each phoneme as: $Delta = T(auto) - T(manual)$ When *Delta* is a positive value, $T(auto) > T(manual)$ means that the automatic boundary is "in late".

SPPAS can perform the alignment either with julius or with hvite. The UBPA we report in this paper is estimated while using Julius CSR engine (Lee and Kawahara, 2009) but the results were also estimated with HVite command of the HTK 3.4 toolkit (Young and Young, 1993). Both are very close, so there's no need in this paper to report them both.

The UBPA was first estimated with a model we didn't train on data. This model is created from the prototypes of the phonemes extracted from models of different other languages. It resulted in an UBPA of 88.82% within a delta of 0.04 seconds, e.g. the automatic boundary is not more than 40 milliseconds before or after the manual one.

Another model was trained by using the prototypes as bootstrap and the transcribed data only. The UBPA of such model is 93.22% within a delta of 0.04 seconds.

Finally, we trained a model by using the prototypes as bootstraps, the transcribed data, and the manually time-aligned data. In order to estimate the UBPA, we used a leave-one-out algorithm by training 79 models and testing it on the single test sentence that had been excluded from training. This resulted in an UBPA of **93.98%** within a delta of 0.04 seconds.

Figure 2 show details about the differences between automatic boundaries and manual ones for vowels (including nasal vowels with almost nil occurrence in our

Figure 1: Example of result of the automatic annotations of Bengali - SPPAS 4.7

corpus); while Figure 3 shows this difference in the case of Bengali consonants. As shown in Figures 2 and 3, the automatic system nearly preserves the manually annotated duration measurements. For example, it allows to see that the end position of the vowel /{/ is correct, but the automatic system detects it lately. However, we can't comment much on nasal vowels because of very few occurrences.

6. Conclusion

This paper presents free linguistic resources created for the Bengali language. These were useful for creating HLT tools for Text Normalization (including a tokenizer), Phonetization and Alignment of automatic annotations for Bengali. These resources have been made available freely since SPPAS version 4.1. Developing an automatic syllabification system for Bengali will be the future focus of our work in this software. It will be based on the existing system already available for French, Italian and Polish (Bigi and Klessa, 2015).

References

Bigi, B., 2012. The SPPAS participation to Evalita 2011. In *Evaluation of Natural Language and Speech Tool for Italian*, volume 7689 of *Lecture Notes in Artificial Intelligence*. Springer Berlin Heidelberg, pages 312–321.

Bigi, B., 2014. A multilingual text normalization approach. *HLT for Computer Science and Linguistics, LNAI 8387*:515–526.

Bigi, B., 2015. SPPAS - Multi-lingual Approaches to the Automatic Annotation of Speech. *The Phonetician*, 111–112:54–69.

Bigi, B., 2016. A phonetization approach for the forced-alignment task in SPPAS. *Human Language Technology. Challenges for Computer Science and Linguistics, LNAI 9561*:515–526.

Bigi, B. and K. Klessa, 2015. Automatic Syllabification of Polish. In *Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*. Poznan, Poland.

Bigi, B. and C. Meunier, 2018. Automatic segmentation of spontaneous speech. *Revista de Estudos da Linguagem. International Thematic Issue: Speech Segmentation*, 26(4).

Bigi, B., A.-S. Oyelere, and B. Caron, 2021. Resources for automated speech segmentation of the african language naija (nigerian pidgin). *Human Language Technology. Challenges for Computer Science and Linguistics, LNAI 12598*:164–173.

Hirst, D.-J., 2011. The analysis by synthesis of speech melody: from data to models. *Journal of Speech Sciences*, 1(1):55–83.

Hirst, D.-J. and R. Espesser, 1993. Automatic modelling of fundamental frequency using a quadratic spline function. *Travaux de l'Institut de Phonétique d'Aix*, 15:71–85.

Kisler, T., Reichel U. D., and F. Schiel, 2017. Multilingual processing of speech via web services. *Computer Speech & Language*, 45:326–347.

Klaiman, M.H. and A. Lahiri, 2018. *Bengali*, chapter 23. Routledge, 3rd edition, pages 427–447.

Le, V.B., L. Besacier, S. Seng, B. Bigi, and T.N.D. Do, 2008. Recent advances in automatic speech recognition for vietnamese. In *International Workshop on Spoken Languages Technologies for Under-resourced languages*. Hanoi, Vietnam.

Lee, A. and T. Kawahara, 2009. Recent development of open-source speech recognition engine julius. In *Asia-Pacific Signal and Information Processing Association*. Annual Summit and Conference, International Organizing Committee.

Lee, A., T. Kawahara, and K. Shikano, 2001. Julius - an open source real-time large vocabulary recognition engine. In *European Conference on Speech Communication and Technology, EUROSPEECH*. Aalborg, Denmark.

Schultz, T. and A. Waibel, 2001. Language-independent and language-adaptive acoustic modeling for speech recognition. *Speech Communication*, 35(1):31–51.

Wells, J.C., 1997. Sampa computer readable phonetic alphabet. *Handbook of standards and resources for spoken language systems*, 4.

Young, S-J. and Sj Young, 1993. *The HTK hidden Markov model toolkit: Design and philosophy*. University of Cambridge, Department of Engineering.

Yuan, J., M. Liberman, et al., 2008. Speaker identification on the scotus corpus. *Journal of the Acoustical Society of America*, 123(5):3878.

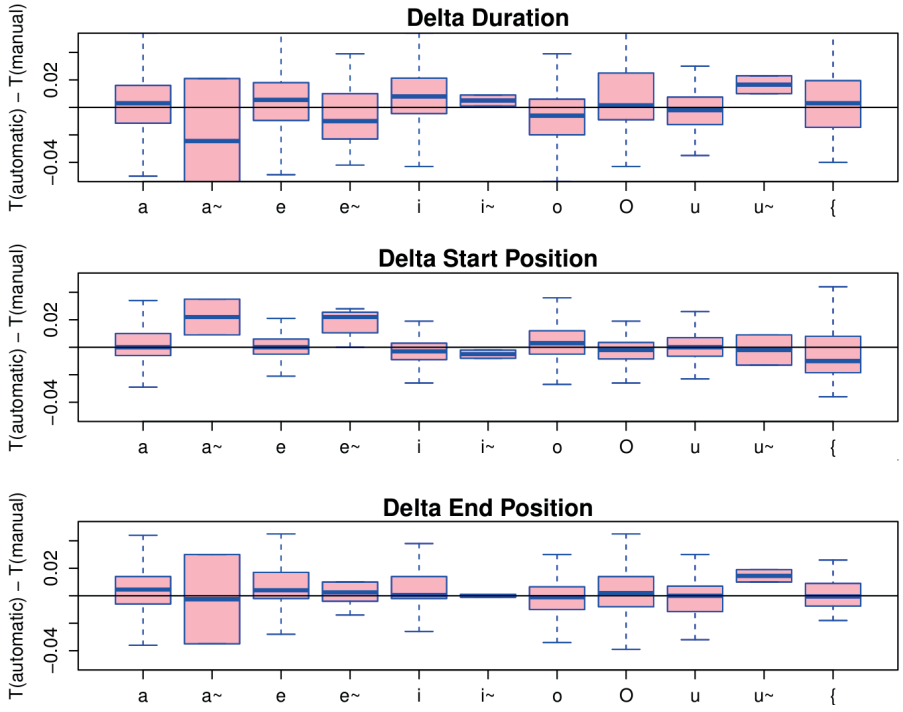


Figure 2: Detailed results of automatic alignment for vowels

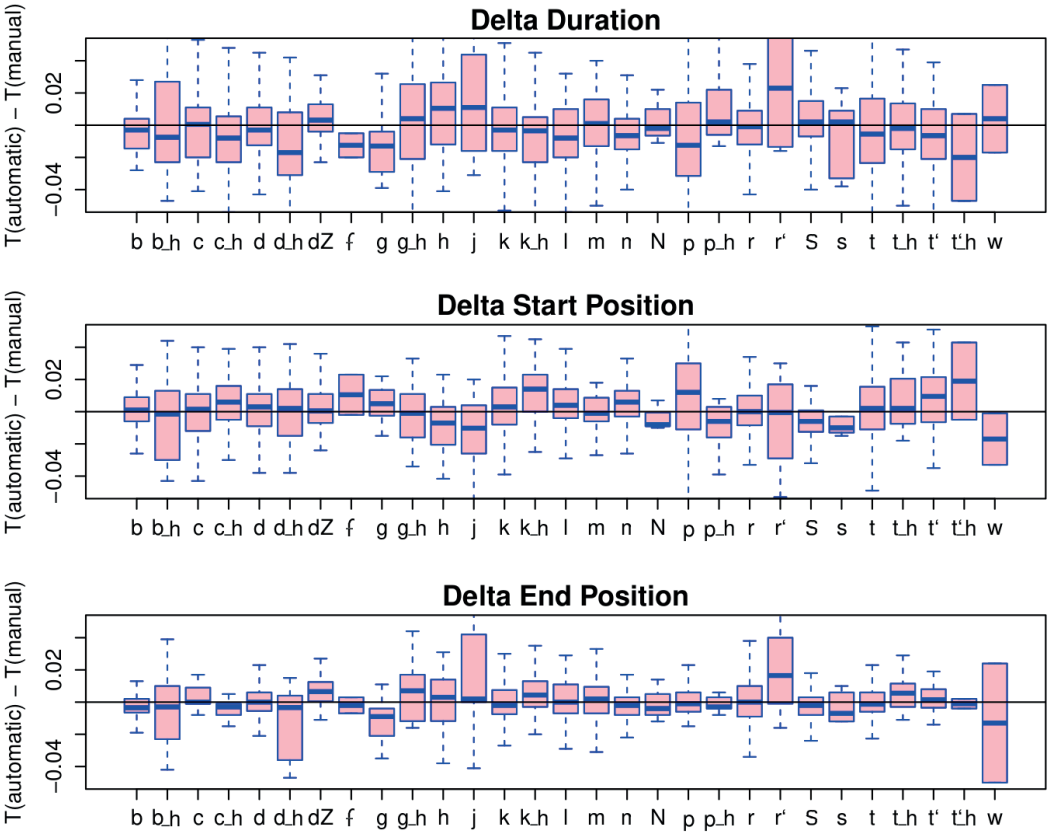


Figure 3: Detailed results of automatic alignment for consonants

Depression in the Times of COVID-19: A Machine Learning Analysis Based on the Profile of Mood States

Marco A. Palomino*, Rohan Allen*, Aditya Padmanabhan Varma†

*School of Engineering, Computing and Mathematics (SECaM), University of Plymouth
marco.palomino@plymouth.ac.uk; rohan.allen-13@students.plymouth.ac.uk
†Department of Computer Science and Engineering, Chalmers University of Technology
vaditya@chalmers.se

Abstract

As the COVID-19 pandemic continues to unfold, a parallel outbreak of fear and depression is also spreading around, impacting negatively on the well-being of the general public and health care workers alike. In an attempt to develop tools to expedite mental health diagnosis, we have looked into emotion analysis and recognition, as this has become indispensable to understand and mine opinions. We have produced a machine learning classifier capable of identifying one of the moods most commonly associated with COVID-19: depression. To analyse how moods and emotions conveyed about COVID-19 have changed in the public discourse over time, we have gathered two Twitter collections—one from 2020 and one from 2022. Our initial findings indicate that fear and depression remain attached to the COVID-19 discourse over the span of two years. Our insights can aid the design of strategic choices concerning the well-being of people in the UK and worldwide.

Keywords: Sentiment analysis, opinion mining, Twitter, machine learning, COVID-19, profile of mood states

1. Introduction

An outbreak of pneumonia reported in Wuhan, China, in December 2019, quickly spread worldwide, and the thousands of deaths caused by it led the *World Health Organization* (WHO) to declare a pandemic on 11 March 2020 (Ciotti et al., 2020). While adults, particularly men, were at greater risk of developing a serious illness as a consequence of such a *coronavirus* disease (*COVID-19*), studies have shown that the pandemic has affected women, young adults, and the unemployed the hardest in terms of mental health (Imran et al., 2020; Pfefferbaum and North, 2020; Benjamin et al., 2021; O'Connor et al., 2021). Regrettably, these groups also developed frequently both physiological and behavioural symptoms associated with distress (O'Connor et al., 2021; Shader, 2020).

Although a decrease in psychological well-being has been observed in the general public due to COVID-19, and higher levels of psychiatric symptoms have been found among health care workers (Vindegaard and Benros, 2020), those who have sought help have experienced serious delays in being treated (Papautsky et al., 2021). A major goal of our work in the long run is to develop tools to expedite mental health diagnosis. Thus, we would like to delve deeper into the subject of *emotion recognition* (Koolagudi and Rao, 2012) and *sentiment analysis* (Liu, 2012), as they have become indispensable to understand and mine opinions (Sun et al., 2017).

In April 2020, approximately a month after the first nationwide COVID-19 lockdown in the UK was announced, we launched an investigation on the emotions expressed on social media to understand the feelings of the general public. We concentrated on *Twitter* (Murthy, 2018), the microblogging platform. To assess the evolution of the emotions expressed about COVID-19 over time, we gathered a second collection of tweets in March 2022.

Then, we processed our two collections of tweets to extract insights into the feelings and emotions expressed by Twitter users. We expect the insights derived from our study to aid in the decision-making of strategic choices concerning the mental health of the population—especially, as a considerable amount of fear, sadness, and depression was conveyed on the tweets that we retrieved.

The reminder of this paper is organised as follows. We will summarise the related work in Section 2. Afterwards, we will describe the corpora that we used for our experiments in Section 3. We will also employ Section 3. to report on the implementation of our machine learning classifier. Section 4. will present our results and, finally, Section 5. will outline our conclusions.

2. Related Work

The availability of large language-based datasets has allowed us to improve the identification and understanding of mental health issues through the study of words (Tausczik and Pennebaker, 2010; Pennebaker et al., 2003). A great deal of research has demonstrated that word use is a reliable indicator of a person's psychological state (Chung and Pennebaker, 2011).

Recognising the emotions expressed by words in pieces of text has earned significance, as an alternative to assess the well-being of people—for example, when attempting to prevent suicide (Desmet and Hoste, 2013). Two of the most notable works on this field, which deserve careful consideration, are *Ekman's* basic emotion model (Ekman, 1992) and *Plutchik's* bipolar emotion model (Plutchik and Kellerman, 2013). Although Ekman's and Plutchik's are well-

regarded models, and we would like to look into them in the future, we will not pursue them in our current investigation. Ekman studied facial expressions, but facial recognition is beyond the scope of our project, as we do not have the equipment to pursue it.

Plutchik, on the other hand, considered eight basic, pairwise, contrasting emotions: joy vs. sadness, trust vs. disgust, fear vs. anger, and surprise vs. anticipation (Plutchik, 1980). Even though we plan to widen the range of emotions analysed by our classifiers as our research progresses, the lack of annotated training collections complicates any attempts to implement Plutchik's model.

As depression is a state of mood often associated with COVID-19 (Renaud-Charest et al., 2021; Johns et al., 2022), we wanted to have a classifier capable of detecting depression. Hence, we opted for the *Profile of Mood States* (POMS) (Norcross et al., 1984), which is a psychological test for assessing an individual's mood state (Berger and Motl, 2000). POMS was formulated by McNair *et al.* (McNair et al., 1971) and it contemplates depression. This made it particularly relevant to our work. We will elaborate below on the details of our implementation of POMS.

We were keen on testing approaches that depart from the traditional methods followed by sentiment analysis, which use lexicons and bag-of-words models (Rudkowsky et al., 2018). We are aware of the improvements reported by researchers who have worked with sequences of characters, without pre-processing the text that becomes the input of a *recurrent neural network* (RNN). For instance, Colnerič and Demšar (Colnerič and Demšar, 2018) implemented one of such approaches and used it to classify tweets into emotional categories.

Following Colnerič and Demšar's example, we have implemented our own POMS classifier. However, we only used characters that occurred in the training set 25 times or more, and we removed emoticons and other symbols that were not part of the tweets in our corpus.

3. Materials and Methods

As a testbed for our experiments, we gathered 409,761 tweets about COVID-19 on 22 April 2020, and we will refer to this corpus hereafter as the *2020 Corpus*. We chose 22 April 2020, because it was when the then UK Foreign Secretary, Dominic Raab, delivered a press briefing to address the Government's response to COVID-19, and he highlighted the use of a vaccine for the first time.

The British press started to cover news about a COVID-19 vaccine at the start of April 2020, when the first human trials began in Europe (Walsh, 2020). A significant investment was made on these trials; therefore, we assumed that the briefing on 22 April 2020 would spark off the discussion on Twitter. We thought this would be an ideal moment to capture tweets with a strong sentiment attached to them, either in the form of Government's criticism or concern for the prevailing situation. We expected the briefing to begin at around 16:30; hence, we started the retrieval of tweets a couple of hours prior to the beginning of the briefing, and kept it going for a couple of hours after the end of the briefing. To be precise, we captured our first tweet at 14:24:39, and the last one at 18:56:27.

Hashtag	Number of tweets
#covid19	238,432
#coronavirus	116,557
#stayhome	31,820
#covid_19	11,068
#socialdistancing	6,510
#covid-19	4,636
#covid2019	2,341
#flattenthecurve	2,124
#coronavirusoutbreak	2,058
#sarscov2	1,861
#virus	1,211

Table 1: Hashtags used to retrieve the 2020 Corpus.

To ensure that we were capturing information about COVID-19, we looked specifically for tweets comprising the hashtags listed in Table 1. Note that Table 1 also displays the number of tweets retrieved for each hashtag.

Figure 1 shows the number of tweets that we retrieved every 30 minutes. On average, we retrieved 81,952 tweets per hour between 14:24 and 19:24; yet, we retrieved more than 90,000 tweets per hour for the first three hours.

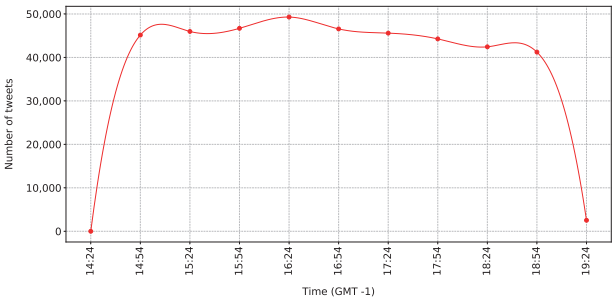


Figure 1: Distribution of tweets in the 2020 Corpus.

To assess how the emotion related to COVID-19 has changed from 2020 to 2022, we retrieved a second corpus, and we will refer to it as the *2022 Corpus*. We started the retrieval of this Corpus on 24 March 2022, because this date marked the second anniversary of the announcement of the first nationwide lockdown in the UK (Johnson, 2022). There was potential for the anniversary to spike the volume of the COVID-19 discourse on Twitter.

We planned to gather as many tweets as we did in 2020. However, COVID-19 seems to have lost popularity as a Twitter topic recently. Hence, we were unable to gather as many tweets as we did before. We collected tweets for five consecutive days, from 24 March 2022 to 29 March 2022—the first tweet was collected on 24 March 2022 at 23:30:05, and the last one on 29 March 2022 at 00:30:07. While the retrieval of the 2020 Corpus lasted 4 hours, 31 minutes, and 48 seconds, and gathered 427,639 tweets, the second one spanned over 4 days, 13 hours, and 2 seconds, and gathered only 265,108 tweets. The size of the 2022 Corpus is only 62% of the size of the 2020 Corpus. How-

Hashtag	Number of tweets
#covid	182,686
#covid19	125,434
#longcovid	34,565
#covidisnotover	25,010
#omicron	15,608
#coronavirus	9,487
#covid-19	8,443
#pandemic	8,429
#mask	6,627
#sarscov2	2,950
#stayhome	2,118
#virus	2,032
#vaccinated	1,320
#covidiots	1,038
#deltacron	283
#cases	273

Table 2: Hashtags used to gather the 2022 Corpus.

ever, it provides enough material to assess the evolution of the subject between 2020 and 2022.

To retrieve the 2022 Corpus, we looked for tweets comprising the hashtags listed in Table 2. We considered hashtags that were not part of the discourse in 2020 but have now emerged—for example, #longcovid, #omicron, #vaccinated, and #covidiots.

3.1. Profile of Mood States (POMS)

POMS is a test to measure an individual's mood (Curran et al., 1995). POMS is relevant to clinical and social psychology. POMS specifies 65 adjectives that are rated by the individual on a five-point scale. Each adjective contributes to one of seven categories: *anger*, *confusion*, *depression*, *fatigue*, *friendliness*, *tension*, and *vigour*.

Given that POMS can recognise depression, it became ideal for our work, as depression is commonly associated with COVID-19 (Renaud-Charest et al., 2021). Adjectives such as *unworthy*, *miserable* or *gloomy* used to describe a person's feelings contribute to classify her mood within the *depression* category (Mackenzie, B, 2022). Also, we removed *friendliness*, as Norcross *et al.* have found that the adjectives corresponding to it are too weak to ensure a valid classification (Norcross et al., 1984). We complemented the model with other adjectives suggested by the *BrianMac Sports Coach* website (Mackenzie, B, 2022). Table 3 shows the full list of adjectives that we employed to identify each of the mood states under consideration.

For the implementation of our classifier, we experimented with the following options: SVM (Noble, 2006), Naïve Bayes (Berrar, 2018), logistic regression (Kleinbaum et al., 2002), random forests (Biau, 2012)—the number of trees was selected using linear search—and *long short term memory* (LSTM) (Nowak et al., 2017).

To train our classifier, we used Colnerič and Demšar's training set, which is based on a corpus comprising 73 billion tweets annotated using distant supervision (Colnerič and Demšar, 2018). The corpus was collected between August 2008 and May 2015, and it is split into training (60%), validation (20%) and test (20%) sets. Colnerič and Demšar's corpus is considerably larger than other options, such as Mohammad and Kiritchenko's corpus (Mohammad and Kiritchenko, 2015). Unfortunately, the random forest was so slow that we were only able to build forests with a maximum of 100 trees. Training 100 trees using bi-grams took longer than a day on *Google Colab*.

Mood state	Adjectives
anger	angry, peeved, grouchy, spiteful, annoyed, resentful, bitter, ready to fight, deceived, furious, bad tempered, rebellious
confusion	forgetful, unable to concentrate, muddled, confused, bewildered, uncertain about things
depression	sorry for things done, unworthy, guilty, worthless, desperate, hopeless, helpless, lonely, terrified, discouraged, miserable, gloomy, sad, unhappy
fatigue	fatigued, exhausted, bushed, sluggish, worn out, weary, listless
tension	tense, panicky, anxious, shaky, on edge, uneasy, restless, nervous
vigour	active, energetic, full of pep, lively, vigorous, cheerful, carefree, alert

Table 3: Chosen adjectives for POMS.

Instead of pre-processing the tweets, we treated each of them as a sequence of characters, and pass such characters one by one into the RNN. The network's task was to combine the characters into a suitable representation and predict the moods expressed on it. The RNN had to learn which sequences of characters form words, since space was not treated differently from other characters. The benefit of this approach is that it does not require any pre-processing. If we were working with words, we would need a tokenizer first and then we would have to decide which morphological variations of the words are similar enough to consider them equivalent, which is what stemming and lemmatization do (Palomino and Aider, 2022). However, in our character setting approach, all those decisions were left to the neural network to figure out.

4. Results

A common metric to estimate the overall sentiment expressed towards a topic on social media is the *net sentiment rate* (NSR). While the NSR was developed for digital marketing, it has been successfully applied to other fields, for example, Palomino *et al.* (Palomino et al., 2016) have applied it to public health studies. The NSR is defined as the difference between the number of positive conversations—positive tweets—and the number of negative conversations—negative tweets—divided by the total number of conversations—total number of tweets:

$$NSR = \frac{Positive\ tweets - Negative\ tweets}{Total\ number\ of\ tweets}.$$

We selected *SentiStrength* (Thelwall et al., 2010) to determine the sentiment expressed on the tweets constituting our experimental corpus. SentiStrength estimates the strength of positive and negative sentiment in short texts, such as tweets, using methods to exploit the *de-facto* grammars and spelling styles of the informal communication that regularly takes place in social media, blogs and discussion forums (Thelwall et al., 2012).

The NSR values for both the 2020 Corpus and the 2022 Corpus are negative: -0.32% and -0.37%, respectively, reflecting the negative nature of the corpus as a whole. Figure 2 displays the percentages of positive, negative and neutral tweets in the 2020 Corpus and the 2022 Corpus, according to SentiStrength. Because the size of the 2020 Corpus differs from the size of the 2022 Corpus, Figure 2 displays percentages, as opposed to absolute values. As shown in Figure 2, the distribution of the polarities in the 2020 Corpus and the 2022 Corpus is very similar.

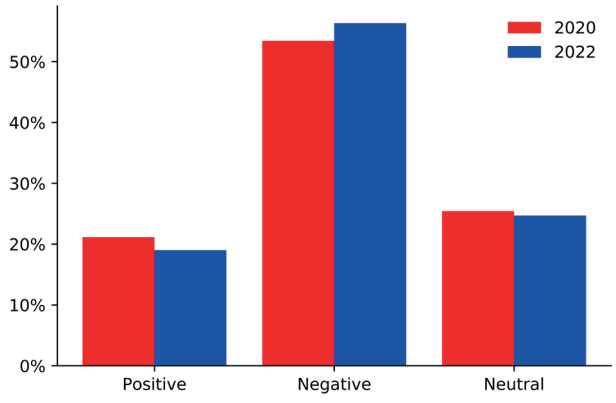


Figure 2: Polarity of the experimental corpus.

Using POMS, we assigned each tweet in the corpus a probability associated with each of the moods under consideration. Figure 3 displays the addition of the probabilities for each mood to occur in each of the tweets. Because the size of the 2020 Corpus differs from the size of the 2022 Corpus, Figure 3 displays percentages, instead of absolute values. Clearly, depression dominates the experimental corpora—the presence of the rest of the emotions appears minimal. Additionally, the percentages of all moods are utterly similar, despite of the year.

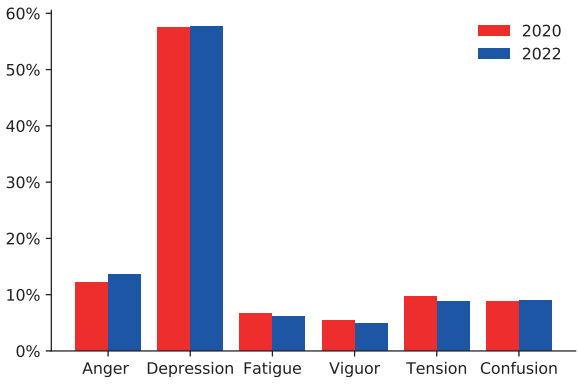


Figure 3: Moods according to POMS.

Research has found that the rates of depression have doubled since the pandemic began (Khubchandani et al., 2021; Seal et al., 2022). An increase in depression commonly accompanies large-scale disasters, whether natural or environmental (Galea et al., 2020).

5. Conclusions

We have applied machine learning to analyse tweets about COVID-19 recorded at two different points in time. Our analysis provides practical insights to aid in the decision-making of strategic choices concerning the wellbeing of the population. The advantages of using POMS to identify depression have been discussed, and we expect to start the search for other models that are applicable to expedite mental health diagnosis.

References

Benjamin, Asaf, Yael Kuperman, Noa Eren, Ron Rotkopf, Maya Amitai, Hagai Rossman, Smadar Shilo, Tomer Meir, Ayya Keshet, Orit Nuttman-Shwartz, et al., 2021. Stress-related Emotional and Behavioural Impact following the First COVID-19 Outbreak Peak. *Molecular psychiatry*, 26(11):6149–6158.

Berger, Bonnie G and Robert W Motl, 2000. Exercise and mood: A selective review and synthesis of research employing the profile of mood states. *Journal of Applied Sport Psychology*, 12(1):69–92.

Berrar, Daniel, 2018. Bayes' theorem and naive bayes classifier. *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, 403.

Biau, Gérard, 2012. Analysis of a random forests model. *The Journal of Machine Learning Research*, 13(1):1063–1095.

Chung, Cindy K and James W Pennebaker, 2011. Using computerized text analysis to assess threatening communications and behavior. *Threatening Communications and Behavior: Perspectives on the Pursuit of Public Figures*:3–32.

Ciotti, Marco, Massimo Ciccozzi, Alessandro Terrinoni, Wen-Can Jiang, Cheng-Bin Wang, and Sergio Bernardini, 2020. The COVID-19 Pandemic. *Critical Reviews in Clinical Laboratory Sciences*, 57(6):365–388.

Colnerič, Niko and Janez Demšar, 2018. Emotion Recognition on Twitter: Comparative Study and Training a Unison Model. *IEEE Transactions on Affective Computing*, 11(3):433–446.

Curran, Shelly L, Michael A Andrykowski, and Jamie L Studts, 1995. Short Form of the Profile of Mood States (POMS-SF): Psychometric Information. *Psychological assessment*, 7(1):80.

Desmet, Bart and VéRonique Hoste, 2013. Emotion Detection in Suicide Notes. *Expert Systems with Applications*, 40(16):6351–6358.

Ekman, Paul, 1992. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200.

Galea, Sandro, Raina M Merchant, and Nicole Lurie, 2020. The Mental Health Consequences of COVID-19 and Physical Distancing: The Need for Prevention and Early Intervention. *JAMA Internal Medicine*, 180(6):817–818.

Imran, Nazish, Muhammad Zeshan, and Zainab Pervaiz, 2020. Mental Health Considerations for Children & Adolescents in COVID-19 Pandemic. *Pakistan Journal of Medical Sciences*, 36(COVID19-S4):S67.

Johns, G, V Samuel, L Freemantle, J Lewis, and L Waddington, 2022. The Global Prevalence of Depression and Anxiety Among Doctors during the Covid-19 Pandemic: Systematic Review and Meta-Analysis. *Journal of Affective Disorders*, 298:431–441.

Johnson, Boris, 2022. Prime Minister's Statement on Coronavirus (COVID-19): 23 March 2020.

Khubchandani, Jagdish, Sushil Sharma, Fern J Webb, Michael J Wiblishauser, and Sharon L Bowman, 2021. Post-Lockdown Depression and Anxiety in the USA during the COVID-19 Pandemic. *Journal of Public Health*, 43(2):246–253.

Kleinbaum, David G, K Dietz, M Gail, Mitchel Klein, and Mitchell Klein, 2002. *Logistic Regression*. Springer.

Koolagudi, Shashidhar G and K Sreenivasa Rao, 2012. Emotion Recognition from Speech: A Review. *International journal of speech technology*, 15(2):99–117.

Liu, Bing, 2012. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.

Mackenzie, B, 2022. Scoring for POMS. `https://www.brianmac.co.uk/pomscoring.htm`.

McNair, Douglas M, Maurice Lorr, and Leo F Droppleman, 1971. *Manual Profile of Mood States*. Educational & Industrial Testing Service.

Mohammad, Saif M and Svetlana Kiritchenko, 2015. Using Hashtags to Capture Fine Emotion Categories from Tweets. *Computational Intelligence*, 31(2):301–326.

Murthy, Dhiraj, 2018. *Twitter*. Polity Press Cambridge.

Noble, William S, 2006. What is A Support Vector Machine? *Nature Biotechnology*, 24(12):1565–1567.

Norcross, John C, Edward Guadagnoli, and James O Prochaska, 1984. Factor Structure of the Profile of Mood States (POMS): Two Partial Replications. *Journal of Clinical Psychology*, 40(5):1270–1277.

Nowak, Jakub, Ahmet Taspinar, and Rafał Scherer, 2017. LSTM recurrent neural networks for short text and sentiment classification. In *International Conference on Artificial Intelligence and Soft Computing*. Springer.

O'Connor, Rory C, Karen Wetherall, Seonaid Cleare, Heather McClelland, Ambrose J Melson, Claire L Niedzwiedz, Ronan E O'Carroll, Daryl B O'Connor, Steve Platt, Elizabeth Scowcroft, et al., 2021. Mental health and well-being during the covid-19 pandemic: Longitudinal analyses of adults in the uk covid-19 mental health & wellbeing study. *The British Journal of Psychiatry*, 218(6):326–333.

Palomino, Marco, Tim Taylor, Ayse Göker, John Isaacs, and Sara Warber, 2016. The online dissemination of nature–health concepts: Lessons from sentiment analysis of social media relating to "nature-deficit disorder". *International Journal of Environmental Research and Public Health*, 13(1):142.

Palomino, Marco A. and Farida Aider, 2022. Evaluating the Effectiveness of Text Pre-Processing in Sentiment Analysis. *Applied Sciences*, 12(17).

Papautsky, Elizabeth Lerner, Dylan R Rice, Hana Ghoneima, Anna Laura W McKowen, Nicholas Anderson, Angie R Wootton, and Cindy Veldhuis, 2021. Characterizing Health Care Delays and Interruptions in the United States During the COVID-19 Pandemic: Internet-Based, Cross-sectional Survey Study. *Journal of Medical Internet Research*, 23(5):e25446.

Pennebaker, James W, Matthias R Mehl, and Kate G Niederhoffer, 2003. Psychological Aspects of Natural Language Use: Our Words, Our Selves. *Annual review of psychology*, 54(1):547–577.

Pfefferbaum, Betty and Carol S North, 2020. Mental Health and the COVID-19 Pandemic. *New England Journal of Medicine*, 383(6):510–512.

Plutchik, Robert, 1980. A general psychoevolutionary theory of emotion. In *Theories of Emotion*. Elsevier.

Plutchik, Robert and Henry Kellerman, 2013. *Theories of Emotion*, volume 1. Academic Press.

Renaud-Charest, Olivier, Leanna MW Lui, Sherry Eskander, Felicia Ceban, Roger Ho, Joshua D Di Vincenzo, Joshua D Rosenblat, Yena Lee, Mehala Subramaniapillai, and Roger S McIntyre, 2021. Onset and Frequency of Depression in Post-Covid-19 Syndrome: A Systematic Review. *Journal of Psychiatric Research*, 144.

Rudkowsky, Elena, Martin Haselmayer, Matthias Wastian, Marcelo Jenny, Štefan Emrich, and Michael Sedlmair, 2018. More than Bags of Words: Sentiment Analysis with Word Embeddings. *Communication Methods and Measures*, 12(2-3):140–157.

Seal, Adam, Andrew Schaffner, Suzanne Phelan, Hannah Brunner-Gaydos, Marilyn Tseng, Sarah Keadle, Julia Alber, Isabelle Kiteck, and Todd Hagobian, 2022. COVID-19 Pandemic and Stay-At-Home Mandates Promote Weight Gain in US Adults. *Obesity*, 30(1):240–248.

Shader, Richard I, 2020. COVID-19 and Depression. *Clinical Therapeutics*, 42(6):962–963.

Sun, Shiliang, Chen Luo, and Junyu Chen, 2017. A Review

of Natural Language Processing Techniques for Opinion Mining Systems. *Information Fusion*, 36.

Tausczik, Yla R and James W Pennebaker, 2010. The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. *Journal of language and social psychology*, 29(1):24–54.

Thelwall, Mike, Kevan Buckley, and Georgios Paltoglou, 2012. Sentiment Strength Detection for the Social Web. *Journal of the American Society for Information Science and Technology*, 63(1):163–173.

Thelwall, Mike, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas, 2010. Sentiment Strength Detection in Short Informal Text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558.

Vindegaard, Nina and Michael Eriksen Benros, 2020. COVID-19 Pandemic and Mental Health Consequences: Systematic Review of the Current Evidence. *Brain, Behavior, and Immunity*, 89:531–542.

Walsh, Fergus, 2020. Coronavirus: First Patients Injected in UK Vaccine Trial. BBC News Services.

Self-supervised Domain Adaptation of Statistical Language Models for Automatic Speech Recognition

Paweł Paściak[1],Danijel Koržinek[1,2], Dariusz Czerski[1,2]

[1]Silver Whisper
pawel.pasciak@silverwhisper.pl
[2]Polish-Japanese Academy of Information Technology
danijel.korzinek@pja.edu.pl
[2]Institute of Computer Science Polish Academy of Sciences
dariusz.czerski@ipipan.waw.pl

Abstract

Most modern automatic speech recognition systems rely on language modeling to manage their vocabulary. Both traditional multi-stage as well as end-to-end system often use a separate language model to fine tune the vocabulary that frequently evolves within a specific domain. Such models may rely on modern neural language models and transformer architectures, but statistical n-gram language models are still often preferred in most narrow domain situations. The problem with domain adaptation is the access to proper training material. This paper analyzes a self-supervised training approach where there is access to a large amount of un-annotated speech data and a pre-trained general purpose speech recognition model to obtain transcriptions to generate a custom domain-adapted model.

Keywords: automatic speech recognition, statistical language model, self-supervised training, domain adaptation

1. Introduction

The classical approach to Automatic Speech Recognition (ASR) relies on decomposing the overall process into modeling the acoustic and language context separately. Mathematically, this is described by the very popular Bayes-derived formula (Mori, 1997):

$$\arg\max_w P(w|O) = \arg\max_w [P(O|w) \cdot P(w)] \qquad (1)$$

where O denotes a sequence of acoustic observations, i.e. the input to the system, and w is the sequence of output tokens, i.e. words recognized by the system. The whole purpose of ASR is to find the sequence of words that maximizes the probability of those words being observed in the sequence, but given that the straightforward approach would be simply unfeasible to compute for any reasonably long sequence of words and observations, the process is inverted as product of the probability of each word being observed at each time step and the probability of the word sequence itself. The former is often referred to as the acoustic model (AM) and the latter is known as the language model (LM). Both of the components have a key role in obtaining the optimal output - the AM determines how the individual words match the observed acoustic sequence, but the LM makes sure the chosen words make sense from the point of view of the language being spoken. Oftentimes a sequence of audio observations can be explained using combinations of different words of varying lengths ordered in different sequences, but only some of them would make sense from the point of view of the language grammar (Mori, 1997).

This approach is common for all ASR systems starting from Hidden Markov Model (HMM) (Rabiner, 1989; Young et al., 2002) through more modern Weighted Finite-State Transducer (WFST) based solutions (Mohri et al., 2002; Povey et al., 2011). Recently end-to-end (E2E) deep neural-network based systems have been increasing in popularity (Babu et al., 2021; Radford et al., 2022). These systems try and circumvent the need to model the individual steps separately and combine the whole process in one large, integrated neural-network model. Such systems will often accept raw audio on input and output text directly without having a clear boundary between the stages required to reach from one modality to the other while keeping the same modeling power as the previous approaches. In fact, this flexibility is often considered an advantage, allowing for better adaptation, especially at scale.

Regardless, many of the E2E still use classic language modeling as an extra step to post-process the output. There are several reasons for this. An E2E system will often model a much smaller vocabulary than their classic counterparts. This is mostly due to the costly nature of fine-tuning such models, which will usually rely on a loss function that doesn't scale well with the the number of outputs (Graves et al., 2006). Many of the systems will be trained to output only letters or at most pieces of words. Given the size and accuracy of the models, they will still perform well, but given such an unconstrained output, mistakes are very likely to happen. The other reason is the ability to modify the vocabulary. Fine-tuning an E2E system is both time consuming and requires lots of expensive data. Updating a statistical LM is a much cheaper alternative, especially when updates are frequent or the domain is low-resource.

Developing an LM usually requires a large quantity of in-domain data. The training procedure is generally unsupervised, so no annotation is needed, but getting a decent amount of in-domain data often requires considerable cre-

ativity, which makes it impractical for automatic, long-term use. This paper deals with a scenario, where we have a small amount in-domain language data, but a large amount of unlabeled audio. This can occur in two scenarios: periodically updating a system that is in production use, or adapting a pre-existing system (e.g. available online) to a new domain. We discuss what is the optimal procedure for obtaining the LM while improving the overall ASR performance. The paper uses the term self-supervised training in the sense that the data used to train the model originates predominantly from the ASR system that will later use that model for further processing.

2. Theoretical Background

2.1. Statistical Language Models

A statistical language model is defined as a model that uses statistical methods to derive the probability $P(w)$ described in the introduction. Another popular method of language modeling are connectionist language models based on artificial neural networks, which include anything from simple feed-forward and recurrent topologies (Schwenk and Gauvain, 2002; Sundermeyer et al., 2012) to modern transformer-based variants like BERT (Devlin et al., 2018) and GPT (Brown et al., 2020). These models often perform better than their statistical counterparts, but the complexity and data required for their development make them unsuitable for the idea presented in this paper. Furthermore, the main advantage of such models is the ability to integrate them directly within the E2E architecture for ASR and one can argue that many of these architectures do include an LM of some variety. This still doesn't address the issues and usefulness of applying a statistical LM as a post-processing step as described in the introduction.

The most popular variant of a statistical LM is the N-gram model (Jelinek, 1998), defined by the formula:

$$P(w) \approx \prod_{i=1}^{L} P(w_i | w_{i-N+1} \dots w_{i-2}, w_{i-1}) \qquad (2)$$

where L is the length of the sequence and N is the length of the model context, also known as the model order. The most common order used in ASR is 3, but 4 is also common in some cases. The order, together with the size of the vocabulary is the main factor determining the size of the model, but also the amount of data required to train each model parameter with sufficient statistical significance. Given a small amount of training data, it would make little sense to train a high order model - a problem known as the curse of dimensionality (Bengio et al., 2000). There are several toolkits used to develop n-gram models, like SRILM (Stolcke, 2002) and KenLM (Heafield, 2011). The experiments described in this paper are based on the former toolkit.

A common method for comparing different LMs is the measure of perplexity (PPL) evaluated using the formula:

$$PPL = 10^{\frac{-\log P(w)}{L}} \qquad (3)$$

where L is the number of words used to compute the log-probability in the sequence. This value is a positive unconstrained value, with lower values denoting a better match of the model to the given test sequence.

A very common issue of N-gram models is the presence of out-of-vocabulary (OOV) words. Since the model requires a list of words to be defined during training, if a new word is presented during inference, this would cause an error, or more often the N-grams containing these words will simply be ignored in any computation. The presence of such words will, however, have a significant detrimental effect on the performance of transcription in the ASR. That is why this statistic is often provided in addition to PPL as a major contributor to model performance in ASR.

2.2. Model Pruning and Optimization

In its basic form, n-gram models require large amounts of training samples to properly approximate all the words and their contexts in a statistically significant manner. The problem becomes exponentially more complex with the increase in vocabulary size and order. To get around these issue, many smoothing and interpolation techniques were developed. In ASR one of the more popular methods is the Knesser-Ney discounting algorithm (Ney et al., 1994). With large enough training datasets, a model can easily become overwhelmingly large. Pruning a model discards the n-grams that contribute least to its performance in terms of PPL (Stolcke, 1998).

2.3. Model Mixtures

Model adaptation can be achieved by merging two datasets and training a model from their union, but the discrepancy in size can cause issues, especially when adapting a large, general-purpose model to a small, well-defined domain. A much more efficient method relies on combining two trained models with a weight (usually denoted as λ) assigned to each model's contribution in the final mix. Determining the weights is usually done by minimizing on an independent development dataset (Kneser and Steinbiss, 1993).

2.4. Transcription and ASR

The main goal, as well as one key step in this paper is the use of ASR to generate speech transcription. The process of determining the correct output sequence is often referred to as decoding. Both WFST-based as well as E2E (Kucsko and Lopez, 2022) systems rely on some form of beam search, with the possibility to use an LM to weigh individual hypotheses. Beam search works by generating a list of likely word sequences and pruning the least probable ones at each time step. The amount of pruning is determined by a parameter known as the beam width (Jelinek, 1998).

3. Adaptation Procedure

The procedure for creating an automated LM consists of several steps. The starting point is a set of three independent LMs and a set of audio files.

The LM named **big** is built using a variety of online sources and the National Corpus of Polish (NKJP) (Przepiórkowski

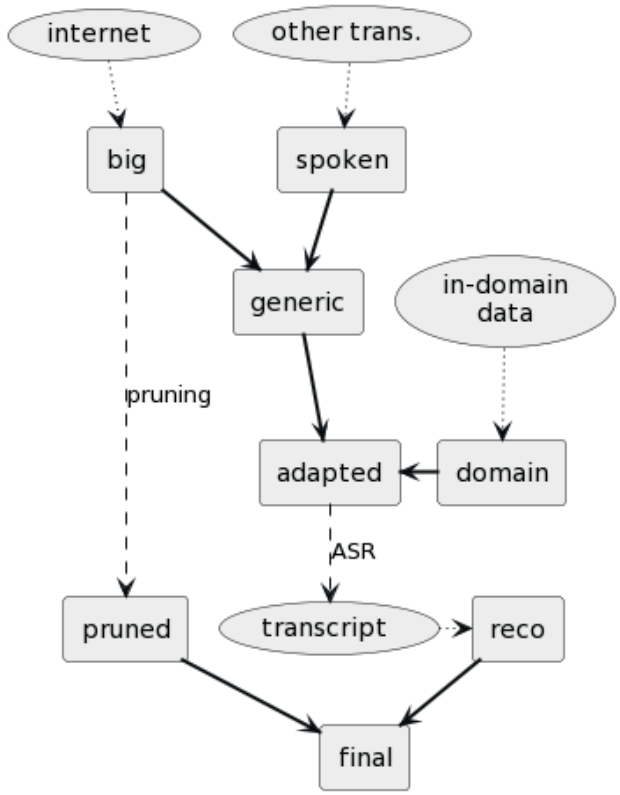


Figure 1: Diagram of adaptation procedure. Rectangles represent models and ovals represent data. Bold arrows represent model mixing operations.

et al., 2011). It includes a large vocabulary (over 4 million words), with the purpose of modeling primarily the language of cultural texts and official speeches. In our approach, this model is also used in a pruned version named **pruned**, which includes all unigrams and more useful higher order ngrams as outlined in section 2.2.

Next LM, titled **spoken**, is built on a collection of telephone conversations transcriptions. The purpose of this model is to include words, phrases and expressions present in spoken language (adjacency pairs, backchannels, fillers, self-corrections etc.). In general, this model does not have to directly relate to the language and vocabulary of the target domain, but it helps by presenting similar dialogues.

Finally, a small model titled **domain** is used to complement the vocabulary and phrases typical of the target subject area. The purpose of this model is to introduce the vocabulary that is specific to the domain. This model can also be used as a way to introduce rare words and named entities into the model.

The next step is to create two models using the mixing procedure mentioned in section 2.3.. In our experience, the mixing coefficient λ does not affect the performance of the approach and can be arbitrarily set to 0.5 (i.e. each model used equally), although tuning is possible in certain cases.

The LM named **generic** is a mixture of "big" and "spoken" models. For the purpose of the procedure it acts as the most accurate universal model, i.e. independent of the modeled domain. The model named **adapted** is made by mixing the "generic" and "domain" models, and is used by ASR to process the collection of audio files.

This results in a set of transcripts, which are used to create a model named **reco**. This model is highly correlated to the modeled domain and fairly small in size. To avoid overfitting to the domain, a final mix between the "pruned" and "reco" models yields the best model name **final**. In this step, the "big" model can be used instead of "pruned" and the vocabulary can be expanded, but here we propose an optimized solution for production use.

Model	1-gr	2-gr	3-gr	ARPA	graph
big	4.3M	45M	55M	1.01G	12.9G
pruned	4.3M	2.6M	1M	54M	1.37G
spoken	40K	457K	1.1M	15M	235M
generic	4.3M	46M	56M	1G	13.55G
domain	1.2K	3.3K	4K	64K	2.44M
adapted	4.3M	46M	56M	1.02G	13.55G
reco-10%	21K	191K	418K	5,7M	105M
reco-50%	46K	581K	1.6M	20M	315M
reco-100%	64K	931K	2.7M	33M	506M
final	64K	2M	3.2M	52M	706M
manual	3.7K	17.7K	26.8K	444K	11M
mix	64K	2.09M	3.25M	52M	707M

Table 1: Size- and performance-related features of developed LMs. For models explanation - see text. Meaning of columns labels: n-gr - number of n-grams for n = 1, 2, 3 (K = 10^3, M = 10^6); ARPA and graph - size of the raw model and WFST graph respectively in K (kilo-), M (mega-) or G (giga-) bytes.

4. Experiments

The experiments were carried out on a collection of telephone conversations related to the insurance domain. The "spoken" model used data from other telephone conversation domains (sales, tech support, etc.). The whole collection contains 15273 conversations with about 973 hours of audio, 933 of which is speech. An extra set of 124 conversation with 4.5 hours in length was manually transcribed for the purpose of measuring the WER. Transcriptions consist of 41035 words. All the conversation were collected as signle-channel audio sampled at 8 kHz, which required the data to be diarized and divided into segments before processing. A single acoustic model was used for all experiments. The model was prepared based on the call center recordings from other domains and expanded by other publicly available corpora. The domain data used to train the "domain" model consisted of only 7947 words and had a vocabulary of 1284 words. This included agent scripts and other call center knowledge base data. The "reco" model was tested on subsets including 10%, 50% and 100% of data, so 100, 500 and 1000 hours respectively. The final model was mixed using the best performing model.

Some features of the models described in section 3. are included in Table 1. All the language models were tested against the files in the test set and their result are presented

in Table 2.

For comparison, two extra models are added to Table 2. The "manual" model is the result of manual transcription of around 5 hours of speech from the same domain, whereas "mix" is a mix of the manual transcripts and various manually collected online sources and other domain data. Both of these models were relatively time consuming and expensive.

Model	RT	OOV	PPL	WER
big	14	129	1376.4	19.32
pruned	43	129	1730.9	20.82
spoken	40	803	323.2	26.66
generic	15	66	262.1	17.44
domain	62	2835	336.0	50.40
adapted	15	58	175.9	15.77
reco-10%	51	369	35.5	14.01
reco-50%	47	194	29.3	13.48
reco-100%	51	153	27.6	13.20
final	47	153	31.1	13.18
manual	47	1611	50.4	20.72
mix	47	137	28.6	12.62

Table 2: Experiments results - relative real time factor (RT), length of data versus processing time of ASR, out-of-vocabulary (OOV), perplexity (PPL) and ASR word error rate (WER) for the models explained in text for test set described in text.

5. Conclusion

This paper suggests a procedure for adapting a statistical model using a large collection of un-annotated speech recordings. This method is relevant because the increasing popularity of ASR technology requires adaptation to a constantly increasing number of domains and use-cases. Furthermore, the systems that are in production use require constant updating to new vocabulary and other changes in language. Traditionally, this process was performed manually and required many specialists to invest lots of time in developing processes for fine-tuning models. Recently, self-supervising methods are increasing in popularity (Baevski et al., 2020).

The method described in the paper allows to obtain a considerable improvement in performance of ASR at no cost of manual labor. The reduction from 17.44% WER from the previous best generic model to 13.18% WER is achieved completely automatically, which can be compared to the result of 12.62% WER that took a considerable investment in manual labor and time.

It is our hope this method can be applied both to updating commercial grade systems as well as support scientists working in low-resource domains. Even though the experiments presented in the paper refer to a specific domain in Polish language it is our belief that the procedure can easily be applied to other languages and domains. In other words, if there is general ASR model available for a given language, this method can be used to significantly improve the performance on a new, low-resource domain. In the future, we intend to expand the method to include adaptation of acoustic models and demonstrate the utility of the approach in other domains and ASR systems.

6. Acknowledgments

The research in this paper was funded by the National Center for Research and Development under the grant agreement POIR.01.01.01-00-1156/18.

References

Babu, Arun, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, et al., 2021. Xls-r: Self-supervised cross-lingual speech representation learning at scale. *arXiv preprint arXiv:2111.09296*.

Baevski, Alexei, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460.

Bengio, Yoshua, Réjean Ducharme, and Pascal Vincent, 2000. A neural probabilistic language model. In T. Leen, T. Dietterich, and V. Tresp (eds.), *Advances in Neural Information Processing Systems*, volume 13. MIT Press.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei, 2020. Language models are few-shot learners.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Graves, Alex, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*.

Heafield, Kenneth, 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*.

Jelinek, Frederick, 1998. *Statistical methods for speech recognition*. MIT press.

Kneser, Reinhard and Volker Steinbiss, 1993. On the dynamic adaptation of stochastic language models. In *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2. IEEE.

Kucsko, Georg and Jeremy Lopez, 2022. pyctcdecode: A fast and lightweight python-based ctc beam search

decoder for speech recognition. https://github.com/kensho-technologies/pyctcdecode.

Mohri, Mehryar, Fernando Pereira, and Michael Riley, 2002. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88.

Mori, Renato De, 1997. *Spoken dialogues with computers*. Academic Press, Inc.

Ney, Hermann, Ute Essen, and Reinhard Kneser, 1994. On structuring probabilistic dependences in stochastic language modelling. *Computer Speech & Language*, 8(1):1–38.

Povey, Daniel, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., 2011. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number CONF. IEEE Signal Processing Society.

Przepiórkowski, Adam, Mirosław Bańko, Rafał Górski, Barbara Lewandowska-Tomaszczyk, Marek Łaziński, and Piotr Pęzik, 2011. National corpus of polish. In *Proceedings of the 5th language & technology conference: Human language technologies as a challenge for computer science and linguistics*. Fundacja Uniwersytetu im. Adama Mickiewicza Poznań.

Rabiner, Lawrence R, 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Radford, Alec, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever, 2022. Robust speech recognition via large-scale weak supervision. *OpenAI Blog*.

Schwenk, Holger and Jean-Luc Gauvain, 2002. Connectionist language modeling for large vocabulary continuous speech recognition. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1. IEEE.

Stolcke, A., 1998. Entropy-based pruning of back-off language models. *Proc. Broadcast News Transcription and Understanding Workshop, 1998*.

Stolcke, A, 2002. Srilm-an extensible language modeling toolkit. In *Seventh international conference on spoken language processing*.

Sundermeyer, Martin, Ralf Schlüter, and Hermann Ney, 2012. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.

Young, Steve, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, et al., 2002. The htk book. *Cambridge university engineering department*, 3(175):12.

Comparative Analysis Of Speech-To-Text Systems For Ukrainian Dialects

Mariia Razno

University of Jena
mari.razno@gmail.com

Abstract

Automated Speech Recognition (ASR) or Speech to Text (STT) task still takes a leading position in discussions about tools for the investigation of the Ukrainian language and its practical usage in various tech products. In this research, was found the Ukrainian ASR model shows the best result on Ukrainian dialects. The WER and CER scores of every model were calculated for test data from 16 Ukrainian regions. The factors that influence dialect recognition by ASR models were compared.

Keywords: Speech to text, speech recognition, dialects, Ukrainian dialects

1. Introduction

Nowadays, Automated Speech Recognition (ASR) models and architectures are being created and developed at a rapid rate. This is happening for the sake of improving various tech products, to make them more human voice receptive. Big companies try to use this technology in almost all their products for navigation, guidance, management, casual talk, and many other purposes.

To make all those products and technologies available for the Ukrainian market and encourage the development of Ukrainian science, it is necessary to adapt those ASR models for Ukrainian speech. It is a challenging task, as Ukraine has a vibrant history of different events, and its political borders have changed during those periods. As a result, an enormous range of dialects was formed in those times. Collecting those dialects in audible and text forms is a separate and volumetric task. This fact makes Ukrainian ASR models' creation quite challenging to implement.

Thanks to the initiative and very talented Ukrainian Speech to Text community[1], the development and improvement of those models stand out. As a result, many state-of-the-art ASR models have already been adapted for the Ukrainian language. As Ukraine is already at the top of the technological innovations step, it is a great time to conduct a performance comparison of existing models for different dialects of Ukrainian language recognition.

This research aims to discover how models' recognition results vary across dialects and model architectures. It is also necessary to take into consideration the datasets that were used in model training. Presumably, dialects that are close in grammar, phonetics, and vocabulary features to specific model train datasets might show better recognition performance. It will be possible to expand the possibilities of existing models by knowing their strong and weak sides. This comparison will significantly boost Ukrainian Speech to Text and Text to Speech model improvements.

2. Ukrainian dialects

Differentiation of Ukrainian dialects concentrates mainly on the regional distribution of dialect similarities, such as cores of dialect and overlapping zones, which can be labeled according to a more or less slight variance of dialect between bordering locations. This research is based on audio recordings from 16 Ukrainian regions: Chernihiv, Ternopil, Poltava, Luhansk, Zakarpattya, Vinnytsia, Cherkasy, Sumy, Mykolayiv, Kyiv, Zhytomyr, Khmelnytskyi, Rivne, Lviv, Ivano-Frankivsk, and Volyn.

Nowadays, the most widespread dialect classifications are the northern and southern groups. The northern group includes the Chernihiv region. The southern group is also divided into two groups southeastern dialects and southwestern dialects. The Southwestern group includes Zhytomyr, Khmelnytskyi, Rivne, Lviv, Ivano-Frankivsk, Ternopil, Vinnytsia, Zakarpattya, and Volyn regions. The southeastern group includes Cherkasy, Sumy, Mykolayiv, Kyiv, Poltava, and Luhansk[2].

The fundamental difference between the northern and southern dialectal groups lies in the role of accentuation in the transmutation of the old vowels ě, o, and e to i sound (did from dědъ [an old man]; dim from domъ [house]; lid from ledъ [ice]). In the south, this change occurred independently of the accent (lis - lisý [forest]; dim - dimký); in the north, it took place only under the accent (lies - lesý; müst, muost – mostkí [bridge]). The same applies to the vowel 'a from the Common Slavic ę (in the northern group, when accented— 'a, ja: z'at' [son-in-law]; when unaccented -e: zetí [pl]) (Kubijovic, 1984).

Some of those dialects have just some minor differences in phonetics, morphology, and syntax. Still, at the same time, some of them contain entirely different features and are even based on different languages. For instance, a dialect of the Luhansk region had a significant influence from Russia, which is why it gained a lot of phonetic and vocabulary

[1] https://academictorrents.com/details/50f7a8e6157a9c2e38919afee0a11d8145e35556

[2] https://academictorrents.com/details/50f7a8e6157a9c2e38919afee0a11d8145e35556

features from the Russian language. In contrast, dialects of the Zakarpattya region are very similar in phonetics to Hungarian or Romanian languages (Moskalenko, 1962).

3. Ukrainian STT models

There are many End-to-end speech recognition models for the Ukrainian language. End-to-end (Jinyu Li 2022) is a system that directly maps a sequence of input acoustic features into a sequence of graphemes or words. For models' performance estimation, Word Error Rate (WER) and Character Error Rate (CER) were used (Dreuw et al., 2011). The word error rate can then be computed as follows:

$$WER = (S + D + I)/N,$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions, C is the number of correct words, N is the number of words in the reference (N=S+D+C). This value indicates the average number of errors per reference word. The lower the value, the better the performance of the ASR system with a WER of 0 being a perfect score. The WER can be estimated in percentages (0100%) or ranges from 0 to 1. CER is similar to Word Error Rate, but operates on characters instead of words. These are standard metrics of the automatic speech recognition system performance. Ukrainian models' WER is mostly estimated on Common Voice 6[3] and 10[4] datasets.

3.1. DeepSpeech

The DeepSpeech RNN model architecture (Hannun et al., 2014) was presented in 2020 by Mozilla Corporation. Ukrainian DeepSpeech model[5] is trained on a total of 1,230 hours from the Ukrainian Dataset at Academic torrents[6] and Common Voice Ukrainian 6. A big RNN model shows WER on the Common Voice 6 dataset, which equals 57%. This model has 47224861 parameters.

Ukrainian DeepSpeech model with Wiki LM is the same DeepSpeech model but with an additional Language Model that improves the transcription accuracy. LM is also called Scorer, it identifies probabilities of words occurring together. It contains phrases and words that have an additional boost during recognition. On the other hand, the Language model has a considerable size increasing model performance time.

LM contains texts from Ukrainian Wikipedia, and its WER on the Common Voice 6 dataset equals 30,65%. have an additional boost during the recognition process. On the other hand, the Language model has a big size increasing model performance time. LM contains texts from Ukrainian Wikipedia and its WER on the Common Voice 6 dataset equals 30,65%.

3.2. Wav2Vec

The Wav2Vec model (Baevski et al., 2020) was released in 2020 by MetaAI. The Ukrainian Wav2Vec model[7] is a finetuned version of the Facebook XLS-R Wav2Vec ASR model[8] trained on the Common Voice 8 dataset. This model uses CNN tokenizer and transformer to build contextualized representations. Creator suggests noting that the training test set was used as a validation set so that this model could have problems with generalization. The WER of this model calculated on the Common Voice 8 dataset equals 27,99%. Wav2Vec is a vast model that contains 317 million parameters.

3.3. Citrinet

The Citrinet is a convolutional Connectionist Temporal Classification (CTC) based automatic speech recognition model (Majumdar et al., 2021). Citrinet models were released by NVIDIA.

The Ukrainian Citrinet-1024 model by NVIDIA[9] is finetuned from the pre-trained Russian Citrinet-1024 model on Ukrainian speech data. It was trained on a validated Mozilla Common Voice Corpus 10.0 dataset (excluding dev and test data) comprising 69 hours of Ukrainian speech. Its WER equals 5,020%. This model has 141224337 parameters.

Ukrainian Citrinet-512 model[10] was trained on Mozilla Common Voice Corpus 10.0 and VOA Ukrainian datasets (~398h)[11]. The model is fine-tuned from the English Citrinet512 model[12]. This model has 35554833 parameters, reducing WER to 8,609% but significantly increasing processing time.

[3] https://huggingface.co/datasets/mozilla-foundation/common_voice_6_1
[4] https://huggingface.co/datasets/mozilla-foundation/common_voice_10_0
[5] https://huggingface.co/datasets/mozilla-foundation/common_voice_10_0
[6] https://huggingface.co/datasets/mozilla-foundation/common_voice_10_0
[7] https://huggingface.co/datasets/mozilla-foundation/common_voice_10_0
[8] https://huggingface.co/datasets/mozilla-foundation/common_voice_10_0
[9] https://huggingface.co/datasets/mozilla-foundation/common_voice_10_0
[10] https://huggingface.co/datasets/mozilla-foundation/common_voice_10_0
[11] https://huggingface.co/datasets/mozilla-foundation/common_voice_10_0
[12] https://huggingface.co/datasets/mozilla-foundation/common_voice_10_0

Model	Region							
	Cherkasy	Ternopil	Poltava	Luhansk	Zakarpattya	Vinnytsia	Chernihiv	Sumy
DeepSpeech	WER: 0.866 CER: 0.968	WER: 0.734 CER: 0.951	WER: 0.734 CER: 0.957	WER: 0.802 CER: 0.975	WER: 0.762 CER: 0.940	WER: 0.661 CER: 0.909	WER: 0.625 CER: 0.892	WER: 0.844 CER: 0.979
DeepSpeech with LM	WER: 0.948 CER: 0.990	WER: 0.879 CER: 0.968	WER: 0.836 CER: 0.948	WER: 0.931 CER: 0.988	WER: 0.919 CER: 0.976	WER: 0.779 CER: 0.953	WER: 0.791 CER: 0.916	WER: 0.921 CER: 0.985
Wav2Vec	WER: 0.369 CER: 0.798	WER: 0.386 CER: 0.775	WER: 0.343 CER: 0.738	WER: 0.319 CER: 0.770	WER: 0.391 CER: 0.789	WER: 0.305 CER: 0.724	WER: 0.331 CER: 0.678	WER: 0.312 CER: 0.822
Citrinet-1024	WER: 0.337 CER: 0.668	WER: 0.342 CER: 0.661	WER: 0.328 CER: 0.612	WER: 0.303 CER: 0.622	WER: 0.379 CER: 0.753	WER: 0.280 CER: 0.620	WER: 0.316 CER: 0.625	WER: 0.220 CER: 0.568
Citrinet-512	WER: 0.301 CER: 0.640	WER: 0.350 CER: 0.700	WER: 0.315 CER: 0.650	WER: 0.291 CER: 0.614	WER: 0.353 CER: 0.684	WER: 0.262 CER: 0.604	WER: 0.302 CER: 0.622	WER: 0.233 CER: 0.625
Squeezeformer-RNNT-ML	WER: 0.341 CER: 0.614	WER: 0.405 CER: 0.681	WER: 0.314 CER: 0.571	WER: 0.309 CER: 0.564	WER: 0.336 CER: 0.635	WER: 0.308 CER: 0.622	WER: 0.254 CER: 0.537	WER: 0.253 CER: 0.574
Squeezeformer-RNNT-XS	WER: 0.323 CER: 0.613	WER: 0.374 CER: 0.650	WER: 0.357 CER: 0.636	WER: 0.295 CER: 0.551	WER: 0.365 CER: 0.702	WER: 0.298 CER: 0.637	WER: 0.286 CER: 0.639	WER: 0.243 CER: 0.603
Squeezeformer-CTC-ML	WER: 0.309 CER: 0.621	WER: 0.330 CER: 0.666	WER: 0.292 CER: 0.573	WER: 0.256 CER: 0.560	WER: 0.338 CER: 0.667	WER: 0.239 CER: 0.540	WER: 0.274 CER: 0.614	WER: 0.214 CER: 0.591
Squeezeformer-CTC-SM	WER: 0.302 CER: 0.612	WER: 0.318 CER: 0.648	WER: 0.315 CER: 0.596	WER: 0.247 CER: 0.562	WER: 0.343 CER: 0.696	WER: 0.255 CER: 0.597	WER: 0.288 CER: 0.644	WER: 0.200 CER: 0.540
Squeezeformer-CTC-XS	WER: 0.318 CER: 0.642	WER: 0.343 CER: 0.690	WER: 0.330 CER: 0.672	WER: 0.270 CER: 0.570	WER: 0.355 CER: 0.719	WER: 0.273 CER: 0.660	WER: 0.333 CER: 0.674	WER: 0.244 CER: 0.623

Table 2. WER and CER results of experiments

Model	Region							
	Mykolayiv	Kyiv	Zhytomyr	Khmelnytskyi	Rivno	Lviv	Ivano-Frankivsk	Volyn
DeepSpeech	WER: 0.705 CER: 0.953	WER: 0.748 CER: 0.942	WER: 0.811 CER: 0.972	WER: 0.792 CER: 0.977	WER: 0.817 CER: 0.958	WER: 0.624 CER: 0.899	WER: 0.826 CER: 0.976	WER: 0.867 CER: 0.989
DeepSpeech with LM	WER: 0.905 CER: 0.979	WER: 0.900 CER: 0.962	WER: 0.917 CER: 0.985	WER: 0.937 CER: 0.987	WER: 0.924 CER: 0.959	WER: 0.814 CER: 0.916	WER: 0.954 CER: 1	WER: 0.966 CER: 0.993
Wav2Vec	WER: 0.295 CER: 0.719	WER: 0.240 CER: 0.614	WER: 0.360 CER: 0.777	WER: 0.312 CER: 0.748	WER: 0.250 CER: 0.667	WER: 0.214 CER: 0.620	WER: 0.358 CER: 0.800	WER: 0.424 CER: 0.864
Citrinet-1024	WER: 0.306 CER: 0.630	WER: 0.228 CER: 0.441	WER: 0.371 CER: 0.712	WER: 0.322 CER: 0.634	WER: 0.204 CER: 0.443	WER: 0.117 CER: 0.336	WER: 0.324 CER: 0.661	WER: 0.353 CER: 0.675
Citrinet-512	WER: 0.289 CER: 0.643	WER: 0.203 CER: 0.447	WER: 0.334 CER: 0.692	WER: 0.296 CER: 0.632	WER: 0.184 CER: 0.399	WER: 0.086 CER: 0.250	WER: 0.309 CER: 0.635	WER: 0.280 CER: 0.603
Squeezeformer-RNNT-ML	WER: 0.311 CER: 0.579	WER: 0.244 CER: 0.450	WER: 0.353 CER: 0.624	WER: 0.312 CER: 0.595	WER: 0.195 CER: 0.395	WER: 0.0893 CER: 0.2447	WER: 0.322 CER: 0.618	WER: 0.319 CER: 0.578
Squeezeformer-RNNT-XS	WER: 0.316 CER: 0.619	WER: 0.235 CER: 0.476	WER: 0.354 CER: 0.679	WER: 0.323 CER: 0.602	WER: 0.208 CER: 0.492	WER: 0.0896 CER: 0.2447	WER: 0.351 CER: 0.644	WER: 0.332 CER: 0.613
Squeezeformer-CTC-ML	WER: 0.272 CER: 0.578	WER: 0.217 CER: 0.454	WER: 0.318 CER: 0.633	WER: 0.260 CER: 0.549	WER: 0.184 CER: 0.402	WER: 0.077 CER: 0.219	WER: 0.288 CER: 0.615	WER: 0.274 CER: 0.558
Squeezeformer-CTC-SM	WER: 0.272 CER: 0.593	WER: 0.202 CER: 0.448	WER: 0.317 CER: 0.633	WER: 0.245 CER: 0.550	WER: 0.179 CER: 0.404	WER: 0.076 CER: 0.206	WER: 0.296 CER: 0.632	WER: 0.278 CER: 0.576
Squeezeformer-CTC-XS	WER: 0.278 CER: 0.610	WER: 0.214 CER: 0.448	WER: 0.333 CER: 0.687	WER: 0.282 CER: 0.603	WER: 0.190 CER: 0.433	WER: 0.089 CER: 0.270	WER: 0.317 CER: 0.651	WER: 0.299 CER: 0.613

Table 3. WER and CER results of experiments

3.4. Squeezeformer

The Squeezeformer architecture was presented in October 2022 as a new ASR model generation by NVIDIA. Models' evaluation is conducted on Mozilla Common Voice Corpus 10.0 dataset using metric WER. All Ukrainian Squeezeformer (Kim et al., 2022) models were trained on Mozilla Common Voice Corpus 10.0 and VOA Ukrainian datasets.

Ukrainian **Squeezeformer-RNNT-ML**[13] has 130293249 parameters and *Ukrainian* **Squeezeformer-RNNT-XS**[14] has 10066817. These models use Sequence Transduction with Recurrent Neural Networks.

Ukrainian **Squeezeformer-CTC-ML**[15], **Squeezeformer-CTC-SM** [15] **,** and **Squeezeformer-CTC -XS** [16] have 125025921, 28184961, and 9031953 parameters respectively. These models use Connectionist Temporal Classification (CTC) loss and decoding instead of RNNT/Transducer loss (Graves, 2012). (For WER values see Table 1.)

Model	WER
Ukrainian Squeezeformer-RNNT-ML	6,575%
Ukrainian Squeezeformer-RNNT-XS	8,814%
Ukrainian Squeezeformer-CTC-ML	6,632%
Ukrainian Squeezeformer-CTC-SM	7,557%
Ukrainian Squeezeformer-CTC-XS	12,17%

Table 1. WER of Ukrainian Squeezeformer ASR models.

4. Experiments

4.1. Testing Data

Data for the experiments include recordings for each dialect separated by regions listed above. These recordings are divided into smaller parts to get better model performance. Each dialect generally includes six audio files, 30-60 seconds each. The audio files include male and female voices. In addition, a table with ground truth manually annotated transcriptions was created to make a future comparison. The quality of audio files for each dialect is different, which was taken into account during testing. Those audio files were obtained from the Corpus of Ukrainian Dialects created by Lviv Polytechnic National University students.

4.2. Tests

The manually created transcripts for each audio segment were cleaned from punctuation and nonalphabetic symbols for more accurate WER and CER calculation[17]. DeepSpeech models' tests were run using the DeepSpeech Python package[18], and for Wav2Vec model transformers, the Python package[19] was used. Citrinet and Squeezeformer models' tests were conducted using the Nemo toolkit[20].

As a result of testing, a table that includes all obtained models' results for each dialect was created. WER and CER metrics were used to estimate the model's performance. A string with the model's result transcript was obtained for each audio segment of a particular dialect. The jiwer Python package[21] was used for WER and CER calculation. Cleaned manually created transcripts were taken as the ground truth and models' results for getting WER and CER values. After that, the mean WER and CER values for each dialect were calculated. Those results were put into the resulting general table for further analysis. (For resulting WER values of conducted experiments, see Table 2 and Table 3)

4.3. Results

The resulting table shows that the DeepSppech model with Language Model shows the worst results on every dialect. The mean WER score for all dialects is 98%. This model has shown the worst WER result on the Ivano-Frankivsk dialect, which equals 100%.

Squeezeformer-CTC-SM model shows the best result for the most significant number of dialects (8 regions): Cherkasy, Ternopil, Sumy, Kyiv, Zhytomyr, Khmelnytskyi, Rivne, and Lviv. This model's mean WER score for those regions is 22,5%. This model also has shown the best result among all models on audio data from the Lviv region, which equals 7%. The second most predictable model is SqueezeformerCTC-ML. This model works best for six regions: Mykolayiv, Ivano-Frankivsk, Poltava, Luhansk, Vinnytsia, and Volyn. This model showed the mean WER score that equals 55%. Also, the Squeezeformer-RNNT-ML model shows the best result for 2 Ukrainian regions: Zakarpattya and Chernihiv. Its mean WER for those two regions equals 60%. (For WER and CER values of obtained results, see Table 2 and Table 3.)

[13] https://huggingface.co/theodotus/stt_uk_squeezeformer_rnnt_ml

[14] https://huggingface.co/theodotus/stt_uk_squeezeformer_rnnt_xs#squeezeformer-rnnt-xs-uk-ua

[15] https://huggingface.co/theodotus/stt_uk_squeezeformer_ctc_ml

[15] https://huggingface.co/theodotus/stt_uk_squeezeformer_ctc_sm

[16] https://huggingface.co/theodotus/stt_uk_squeezeformer_ctc_xs

[17] https://huggingface.co/theodotus/stt_uk_squeezeformer_ctc_xs

[18] https://huggingface.co/theodotus/stt_uk_squeezeformer_ctc_xs

[19] https://huggingface.co/theodotus/stt_uk_squeezeformer_ctc_xs

[20] https://huggingface.co/theodotus/stt_uk_squeezeformer_ctc_xs

[21] https://huggingface.co/theodotus/stt_uk_squeezeformer_ctc_xs

5. Conclusions

DeepSpeech architecture is quite old-fashioned and very time-consuming. DeepSpeech with LM shows worse results than the DeepSpeech model due to an additional Scorer, which tried to match recognized words with existing vocabulary that contains boosted phrases. Since almost every dialect has specific morphology features that are not included in the Scorer Language Model, the worst results were obtained.

Squeezeformer-CTC-SM and Squeezeformer-CTC-ML models show the best results due to the variety of dialects in training datasets and the perfect number of parameters. The squeezeformer-RNNT-ML model was trained on the same datasets as Squeezeformer models but showed better results on audios where the speaker's pronunciation is not that good, which makes recognition more difficult. It can be stated that the modern architecture of ASR models shows the best recognition and performance results.

Dataset plays an essential role in the recognition process. Ukrainian Open Speech To Text Dataset was used to train 5 Ukrainian Squeezeformer models and the Citrinet-512 model. The Speech Recognition community of Ukraine created this dataset. The dataset consists of 188.31 GB of audio data. This dataset is based on speech data from Ukrainian most popular television channels. Also, it contains data from Mozilla Common Voice for the Ukrainian language. This dataset is more dialect diverse than other Mozilla Common Voice datasets.

The result analysis by region shows that dialect groups' specific characteristics do not play a defining role during recognition. In fact, the model that gives the best recognition results covers the regions from different dialect groups. Assuming that some regions have phonetics, morphology, and syntax similarities, that group's best WER and CER results should be obtained on one particular model testing.

In conclusion, the best model for dialects recognition is state-of-the-art Squeezeformer architecture. Medium-size models show the best results, so it is important not to overtrain the model using more parameters that increase processing time. In addition, it is better to train models on the dataset that includes a wider variety of speech from different country regions. Following these suggestions, other Ukrainian ASR model creations can be improved.

References

Baevski, A., Zhou, H., Mohamed, A. and Auli, M. (2020). Wav2vec 2.0: *A Framework for Self-Supervised Learning of Speech Representations*. arXiv:2006.11477 [cs.CL].

Kubijovic, V. (1984) *Encyclopedia of Ukraine vol. 1*, University of Toronto Press.

Dreuw, P. and H., Hermann, G. and N., Hermann. (2011). Confidence and Margin-Based MMI/MPE Discriminative Training for Offline Handwriting Recognition. IJDAR. 14. 273-288. 10.1007/s10032-011-0160-x.

Graves, A. (2012). *Sequence Transduction with Recurrent Neural Networks.* arXiv:1211.3711 [cs.NE].

Hannun, A., Case, C., Casper J., Catanzaro B. (2014). *Deep Speech: Scaling up end-to-end speech recognition*. arXiv:1412.5567 [cs.CL].

Jinyu, L. (2022) *Recent Advances in End-to-End Automatic Speech Recognition.* arXiv:2111.01690 [eess.AS].

Kim, S., Gholami, A., Shaw, A., Lee, N. et al. (2022). *Squeezeformer: An Efficient Transformer for Automatic Speech Recognition*. arXiv:2206.00888 [eess.AS].

Majumdar, S., Balam, J., Hrinchuk, O., et al. (2021). *Citrinet: Closing the Gap between Non-Autoregressive and Autoregressive End-to-End Models for Automatic Speech Recognition*. arXiv:2104.01721 [eess.AS].

Moskalenko, A. (1962). *Narys istorii ukrainskoi dialektolohii radianskyi period*. Odessa.

RPGs: small-scale register analysis using the taxonomy of discourse units

Aleksandra Rewerska[1], Antonina Świdurska[1]

[1]Adam Mickiewicz University, Poznań
rewerskaa@gmail.com, antonina.swidurska@gmail.com

Abstract

This article focuses on describing the spoken language register of RPG (role-playing games) in the Dungeons & Dragons 5e system. There are three purposes: (1) to verify whether the newly introduced taxonomy of discourse units (Biber 2020) would work well on another register; (2) whether the Role-playing Games register is dominated by the narratives about the past or future; (3) to find the most frequent purpose in the register. The study involves three people as a group going through a campaign. Two excerpts were separated from the same session, which was a part of a long-running Role-playing Game campaign. Further, the two excerpts are divided into discourse units and labelled according to the purposes fulfilled. They are then counted and analysed in terms of the most frequent occurrence, the degree of fulfilment and their appearance with each other. It is discovered that there is a dichotomy between players' purposes and their characters. Unusual uses of two purposes are found, as the most frequent one in the register. The taxonomy introduced by Biber works well for the study, but there seems to be a need for one more category to be added. Therefore, using "Performing" communicative goal for further, large-scale studies is proposed.

Keywords: conversation, discourse units, register analysis, RPG, speech genres, communicative purposes

1. Introduction

The focus of this study was the characterisation of the spoken language register of tabletop role-playing games, further in this article referred to as "RPGs". The purpose was to verify whether the newly introduced taxonomy of Discourse Units (Biber, 2020) would work well in describing the chosen register. In the initial characterisation of the register, the framework for Analysing Situational Characteristics proposed by Biber (2009) was followed. Moreover, the use of communicative goals was checked, particularly the number of them used per unit, as well as which one would be the most frequent.

Two fragments of an RPG session played in Polish based on the Dungeons & Dragons 5th edition system were analysed. The recording of the session is part of a larger campaign, happening in the same setting, among the exact same participants. Thereby, a small-scale analysis was made, from which the first conclusions were drawn.

In Section 2 the methodological framework used is shown. A brief description is given and some insight into the way of dealing with border cases is provided. Next, the material used for the analysis is described, and the steps taken in order to fit into the analytical approach adopted are explained. The last two Sections of this article are reserved for conclusions and their discussion.

2. Methodology

Section 2.1 provides the definitions of the crucial terms used. Most of further analysis uses the taxonomy proposed by Biber in the 2020' paper. It was developed with the thought of generalising possible communicative functions, which are presented in the Subsection 2.2.

In order to generally describe the register in question, "The framework for analysing situational characteristics", as proposed by Biber (2009, pp. 39-40) is followed. It can be seen in the description of the database in Section 3.

2.1. Definitions of the basic terms

Tabletop Role-playing Games (TRPGs): For the players of role-playing games, understanding of the term comes somewhat intuitively. Zagal and Deterding (2018) presented quite a lengthy chapter on definitions of role playing games. TRPGs are "play activities and objects revolving around the rule-structured creation and enactment of characters in a fictional world. Players usually individually create, enact, and govern the actions of characters, defining and pursuing their own goals, with great choice in what actions they can attempt." (Zagal and Deterding, 2018, pp. 35). Contrary to other people, including RPG handbook writers, Zagal and Detering do not put more weight on some aspects of the game rather than the others (see: Mearls and Crawford, 2014).

Register: The description of the register includes three main components: situational context, linguistic features, and the functional relationships between the first two components (Biber 2009). It is related to the specific situation of use, as well as specific communicative purposes.

Discourse Units: They are "coherent segments of talk with distinct communicative goals." (Biber 2020, pp. 4). For a piece of recording to be qualified as a discourse unit (DU), it must meet certain requirements: it must have a visible beginning and end, visible single main communicative purpose, be formed by a minimum of five utterances or one hundred words (Biber 2020).

2.2 Communicative purposes of the discourse units

The taxonomy developed by Biber (2020) consists of nine basic communication purposes. Since they are intended to be general, there are some border cases. For some of the purposes below, aside from a suitable explanation, a description of RPG-specific cases is given.

Situation-dependent commentary (SDC). It refers to commenting on people, events or objects that exist or are happening in a shared situational context in the situation currently happening. In the RPG-related register, this

purpose was assigned to all dice roll declarations, as well as describing game mechanics.

Joking around (JOK). This applies to a conversation whose purpose is to be humorous, including witty teasing or flirting.

Engaging in conflict (CON). This goal applies to any conflict, from debate through argument.

Figuring-Things-Out (FTO). It refers to a conversation through which one tries to consider options and plans, or find a solution to a problem.

Sharing feelings and evaluations (FEL). This includes talking about feelings, evaluations, opinions, beliefs, or sharing your own point of view.

Giving advice and instructions (ADV). It refers to giving instructions, advice, or suggestions to the other speaker.

Describing or explaining the past (PAS). This purpose relates to narrating past stories or describing the events of the past.

Describing or explaining the future (FUT). It includes descriptions and speculations about future events, as well as intentions, schemes or plans.

Describing or explaining (time-neutral) (DES). It is the explanation and description of facts, information, people, events where time is insignificant or indefinite. For the purposes of this project, the descriptions of what a player character does are also incorporated to this purpose, regardless of the grammatical tense. This includes any description of a character's own action in-game, such as "I stand and smoke," or "I walk contentedly ahead, holding a small cat in my hand, I smile and begin to say".

3. Dataset

The data consisted of two recordings in Polish, one around 6 and the other around 8 minutes in length. They were both taken from one meeting, during which the participants played a session of a campaign that started nearly two years prior. This was the first session which was recorded, yet participants were already familiar with the world they played in and the characters they portrayed. This session was therefore not arranged specifically for the study to be made. The intention of choosing an already established campaign was to minimise the risk of players acting unnatural and feeling insecure, contributing to the occurrence of an observer's paradox (Labov, 1980).

3.1 The participants

There were three people present at the session, as is usual for the game in question. They are all students in their twenties, native speakers of Polish. One of them is the Dungeon Master (Game Master, DM, GM) who acts as a narrator for the game, and the other two are players, who are responsible for the actions of their player characters. There are no people who are listening to them, yet everyone is aware of the voice recorder. The DM, as well as one of the players, are the authors of this paper.

Due to the characteristics of this particular register, the participants' utterances were split into two or more layers for each person – one is reserved for the in-character speech, and the second one is for out-of-character speech, which can include both the descriptions of one's character's actions, or comments completely unrelated to them. In case of the DM, there was a need for an additional layer for each extra character she impersonated.

3.2 The recordings and annotation

The recordings were taken during one meeting, and are a part of one session, hence the player characters, which are Charlene and Bernaar, stay the same. The DM impersonates three people: Netrand, the characters' supervisor; the shopkeeper and Ryker, who are episodic characters.

The recording from which the two excerpts were taken was made with a smartphone and then converted to .wav format. The excerpts were then selected and annotated using Praat software accordingly.

In the first recording, named "Shopping", the two player characters, as well as one non-player character (NPC), Netrand, are in a market place together. They are talking to a cheese seller. Netrand is the character who supervises the mission, yet, he is being impersonated by the game master, who wishes to let players deal with their own adventure. Thus, a situation arises in which the DM has to let the player make a decision, even though she plays an NPC in a position of authority who would normally make it.

Although there are only three people in the recording, there are four people in this scene. The game master, as is usual for TRPGs, controls more than one character at the time, being also the one responsible for describing the scene.

At the beginning of the second recording, named "Looking for Yoru", the player characters are alone in a bustling square. Charlene and Bernaar try to find a person named Yoru. The third and final member of the conversation appears after one of the players calls for him. Charlene decides that since she cannot find her friend, she will ask someone who is partying there, hoping for the best result. Ryker, who is the one she bumped into, does not seem to be in a condition to help her at all.

3.3. Working on data

Both recordings were annotated in Praat, with each layer corresponding to one of the roles that the subjects took in the game, or one of the participants' out of character speech, which did not necessarily mean out of game speech. The utterances were transcribed accordingly, and then converted into text files, where they were further processed to have the uniform form of a conversation script. An excerpt of the converted script is shown below:

Fragment of the script for the "Looking for Yoru" recording:
Char: Ej typie Yoru widziałeś? (Eng: Oi, dude, have you seen Yoru?) 0.776s
Ryker: Typiarę widziałem? Typiarę? (Eng: Dudess have I seen? Dudess?)
Char: Yoru, Yoru
P2: *laugh* 0.318s
Ryker: Typiary są wszędzie! (Eng: Dudesses are everywhere!)
Char: Typiary są wszędzie; a typ? (Eng: Dudesses are everywhere; what about dude?)

Having such looking scripts, we then proceeded to extract the subsequent topics that appeared in the conversational interaction. Each fragment was given a name that specified the goal of the conversation. Discourse Units from

"Shopping" excerpt were numbered DU 1.x, whereas from "Looking for Yoru" – DU 2.x.

The next step in the project was to label the subsequent DUs using the general taxonomy proposed by Biber (2020), described briefly in Section 2.2 of this article. The previously separated DUs were labelled, with each label also coded on a scale of 0-3 to reflect how much of the goal is met in a particular section: 0 means no presence of purpose, 1 indicates a minor goal, 2 indicates a major goal and 3 means that this purpose is dominant in the excerpt.

One DU may have several communication goals, but always only one dominant one (Biber et al., 2020). For this reason, all emerging goals were highlighted in each excerpt along with a number indicating the potency of the goal, but the purpose with a presence of 3 was always marked once. After this encoding, the fragments looked like DU 1.1:

DU 1.1: The supervisor brings up choosing the person for barter. FTO=3 (Figuring-Things-Out)

Netrand: No dobrze. (Eng: Very well, then.)
Czy któreś z was (Eng: Does any of you…?) 0.94s
Saramyrze, masz umiejętności (...) (Eng: Saramyr, do you have skills…) 1.42s
które wiążą się z targowaniem? (Eng: that involve bargaining?)
Bernaar: Tak. (Eng: Yes.) 0.37s
Myślę, że mógłbym spróbować. (Eng: I think I could give it a try.)
Charlene: Nie będę się kłócić. (Eng: I'm not going to argue.)
DM: No dobrze. (Eng: Very well, then.)

4. Results

The study was conducted on small fragments representing the register of tabletop role-playing games. First, the analysis of each fragment was carried out, and then they were looked at as a whole. It is important to remember that the conclusions drawn from such a small set of data are only preliminary, not final.

4.1 General Analysis

In the following Subsections the general tendencies are described. More detailed analyses of particular discourse units are in Sections 4.2-4.4.

4.1.1 "Shopping" analysis

This recording, which was 8 minutes and 14 seconds long, was divided into 13 discourse units, the last one of them was however excluded from analysis, as it was short and unfinished. From the remaining 12, each was labelled with an appropriate communicative purpose or set of purposes – the number of them varied between 1 and 3.

The purposes of engaging in conflict (CON), describing or explaining future/past (FUT/PAS) did not occur during this recording. The time-neutral describing or explaining (DES) was the most popular one in this dataset, followed by sharing feelings and evaluations (FEL) and figuring-things-out (FTO).

4.1.2 "Looking for Yoru" analysis

This recording was 5 minutes and 50 seconds long. It was divided into 11 discourse units. Similarly to the second recording, DES purpose was the most common one, occurring in more than half DUs. Joking around (JOK) and situation-dependent commentary (SDC) were the second most common purpose, closely followed by figuring-things-out (FTO). The rest of the purposes did not occur at all.

4.2 Hybrid discourse types

The discourse types characterised by not one dominant communicative purpose, but also a secondary one, are called hybrids. In the study, this was the case with 18 DUs from a total of 23.

In hybrid discourse types, five purposes were featured with another major one: **figuring-things-out** (in combination with giving advice and instructions, sharing feelings and evaluations, time-neutral describing or explaining, joking around); **sharing feelings and evaluations** (in combination with giving advice and instructions); **situation-dependent commentary** (in combination with joking around or time-neutral describing or explaining); **describing or explaining (time-neutral)** (in combination with giving advice and instructions, sharing feelings and evaluations, joking around, situation-dependent commentary, figuring-things-out) and **joking around** (in combination with figuring-things-out or time-neutral describing or explaining).

Contrary to what was denoted by Biber (2020) as the most frequent occurring, no hybrids with descriptions based on the future or the past (FUT/PAS) appeared in this register. Instead, the goal of joking around (JOK) proved to be an important element.

	SDC	JOK	CON	FTO	FEL	ADV	PAS	FUT	DES
SDC	1	3	-	-	-	-	-	-	2
JOK	-	-	-	1	-	-	-	-	1
CON	-	-	-	-	-	-	-	-	-
FTO	-	1	-	3	1	1	-	-	1
FEL	-	-	-	-	-	1	-	-	-
ADV	-	-	-	-	-	-	-	-	-
PAS	-	-	-	-	-	-	-	-	-
FUT	-	-	-	-	-	-	-	-	-
DES	2	1	-	-	1	1	-	-	1

Table 1:Primary and Secondary goals in hybrids

Table 1 describes the hybrid discourse types. Each row represents the primary communicative goal of a DU, while the columns are responsible for secondary goals. Tertiary goals are not included. Each number represents a number of occurrences of DUs in corresponding configuration. DUs with no secondary goal are marked as if they had the same primary and secondary goal.

4.2.1 Most prominent purposes

SDC

SDC, especially in the "Looking for Yoru" excerpt, appeared mainly as a hybrid with a secondary purpose of JOK or DES, but it also appeared in a unit with three fulfilled goals, namely SDC, JOK and DES.

From the very beginning of the “Looking for Yoru” excerpt, it is apparent that none of the participants is entirely serious, including the DM, who, despite her function, laughs and jokes with the players. There are a lot of situational comments, among them one regarding the bag of weed that appears in the DU 2.4 and also references to the mechanics of the game itself, in this situation related to the character skill test associated with smoking activity. All this contributes to the fact that the primary purpose of the situation is situational dependent commentary, which with the secondary purpose of joking around, also fulfils the purpose of time-neutral description.

Fragment of DU 2.4: The player decides to smoke instead of solving the problem. SDC: 3 JOK: 2 DES: 1

DM: I wyjmuje taki *laugh* woreczek (Eng: And he pulls out a kind of *laugh* little bag)

Ryker: Skręcimy! Skręćmy! (Eng: Let’s roll! Let’s roll!) (...) W prawo. (Eng: To the right.) (...)

P1: Keej, I go with that. (Eng: Okay, I go with that.) (...)

DM: To będzie constitution saving throw. (Eng: That’s gonna be a constitution saving throw.) (...)

FTO

The purpose of FTO tended to occur in triple hybrids, accompanied by ADV and FEL; while in dual primary purposes it always occurred with a secondary DES goal. Triple purposes with the primary goal of FTO occurred in the “Shopping” excerpt, in DUs following immediately after each other. In the first, it appeared in a combination of: FTO: 3 ADV: 2 FEL: 1, followed by: FTO: 3 FEL: 2 ADV: 1.

In the instances mentioned, it seems that the main purpose of the conversation is to determine what is to be purchased in the next scene. However, it turns out that one of the players, in fulfilment of the primary purpose, also decides to express her personal opinions and feelings. Admittedly, she does not do it directly, but instead makes meaningful sounds and sighs. Furthermore, the supervisor, played by the DM, exploits the primary purpose of FTO in order to give specific instructions (hence the use of ADV here) and convey mild annoyance at his employee's dissatisfaction. In those cases, the FTO becomes sort of an umbrella for hiding the other purposes.

DES

DUs with a primary DES function occurred mainly in dual goals, in “Shopping” excerpt, with a secondary JOK, SDC or FEL. Although one of the most important elements of the RPG register is precisely the narrative, introduced mainly by the DM, there was only one case in which a DU with a primary DES purpose also had two others, namely ADV and FEL. Unlike the FTO, which constituted an umbrella instance in the "Shopping" excerpt, that phenomenon does not occur here.

The DES does not represent a single purpose under which other, equally important purposes for the participants, are hidden. Rather, other goals are fulfilled in the course, interrupting the fulfilment of the DES purpose and replacing it with another. In DU 1.6, it can be seen that at the beginning, the DM focuses the participants’ attention back on the description she started before and tries to continue it. The description is interrupted by the NPC expressing his negative feelings, which in turn is interrupted by a return to the description and then moving on to decision-making. Thus, it can be seen that here the goals are more separable, with the DES being a primary purpose that provides a framework within which the others appear. It seems that DES is the axis to which participants return to keep the game moving forward.

4.3 DUs with a singular Discourse Type

Among the 23 separated DUs in the two recordings, only 5 of them had a single, major target assigned to them. It appeared in the case of **FTO** (appearing three times); **SDC** (appearing once); **DES** (appearing once).

DU with singular FTO purpose has already appeared in the article, in Subsection 3.3; that is DU 1.1, coming from the "Shopping" excerpt. Since the NPC in charge of the players' characters is present in the scene, he seems to naturally direct it, trying to determine who is to take over the bargaining function; the rest of the characters express their opinions on the matter, thus arriving at a solution of the problem. The excerpt is so atypical due to the lack of a separate target in the dynamic between the supervisor and the players. At this point, everyone is focused on one and the same goal, with emotion and the desire to give instructions coming only in subsequent units.

SDC as the main target appeared only once, in DU 2.7. It refers to a situation in which the main talking point is the substance used. Comments and reactions are made, only regarding the subject matter.

The one instance of a singular DES purpose in DUs, is in DU 2.8. It is a passage entirely focused on the description of the character sought by the player. The information given by her is repeated by the NPC in an amused manner. However, this time both the NPC and the player's character are focused on only one purpose, that is, the DES.

4.4 Influencing the game world

One of the ways to look at RPGs is to see them as a particular type of storytelling. The DM creates and describes the game world, but once the players are inside they get the power to change it through their characters’ actions. To do so, they use utterances with performative functions, which are spread throughout the whole session. Sometimes, even though description is not the dominant purpose of a DU, this influence can be seen, as portrayed in the examples below:

— “Stoję z boku z coraz wyżej uniesionymi brwiami, wyraźnie rozbawiony. (Eng: I’m standing to the side, raising eyebrows higher and higher, noticingly amused)” from DU 2.4

— “*inaudible* w każdym razie podchodzi. (Eng: Anyway, he approaches.)” from DU 1.6

— “Netrand się odsuwa bardzo tak- (...) I wyjmuje (...) papierosa. Tak patrząc porozumiewawczo. (...) Na zasadzie… (...) Więc wyjmuje drugiego. (...) Stoicie obok i będziecie palić. (Eng: Netrand steps back very– (...) And pulls out a cigarette. Looking to you. Like…(...) So he pulls out the second one (...) You two are standing aside and you will smoke.)” from DU 1.7.

Those quotes present something unique to this register: the ubiquitous performative function of utterances. It can be explained by the notion some literary critics use to help answer the question of “what literature does” (Culler, 2011).

It might be useful to think about RPG registers having that in mind.

Even though the performative function is applicable to particular utterances, there are some DUs where it can be noticed more than in others. The last quote presents a large part of DU 1.7. There, the DES purpose is the dominant one, yet it is noticeably different to the purpose present in some other DUs with DES. Here, instead of just describing, both DM and Player 2 take in-game actions. When the GM says "You two are standing aside and you will smoke.", it is not a description made for someone, but a way of making it happen.

DU 2.1 is one making this distinction more apparent. In the fragments below, two cases of the DES purpose can be observed.

Fragment of DU 2.1: The narrator explains why it is hard to find Yoru. DES: 3; SDC: 1

(...) DM: Kej, to widzisz że tych głośnych grup jest dosyć dużo, nie (Eng: Okay, so you see that there are quite a lot of these loud groups, yeah) (...)

DM: (...) Ale zaczynacie iść przez plac, nie, rozglądając się (Eng: But you walk through the square, yeah, looking around)

Here, DES has two functions. First, the DM describes the surroundings, but then she also uses description to influence the world and make players move. Although both those parts sound like description, their purpose is significantly different. As the used in this paper taxonomy developed by Biber (2020) is concerned with communicative goals, describing them by the same term seems unsuitable. Therefore, for future RPG related studies, we would suggest using a new term, "Performing" (PER) to describe the DUs where the performative goal is fulfilled. We speculate that this could be an important addition especially when describing combat situations, where the characters are in a constant state of "doing" instead of "talking".

5. Discussion

When talking about RPG registers, it is important to remember that the discourse happens on two levels. On the surface, there are the participants, "players", who are enjoying themselves in the game. On the second level, there are "characters", controlled by the participants; but they do not exist outside of the game world. The "players" and the "characters" have a different goal to begin with: the first ones want to have fun in the game world, while for the latter the game world is reality. This creates a dichotomy where the communicative goals might be different on both levels.

Contrary to our initial belief, there were no FUT, nor PAS purposes in the excerpts analysed, although we regarded the register as narrative. The speakers were exclusively focused on present in-game activities, did not narrate any past events, or talk about plans for the future. What's more, it has been found that purposes such as FTO and DES also serve other functions. The former becomes an umbrella premise that welds together the purpose of the player and the character, as well as the divergent purposes of individual characters, and hides this difference. The latter as the most common purpose also serves more functions. Not only does it confirm the existence of the narrative, it also provides a recurring structure for the ongoing story. Every time it is interrupted by other appearing purposes, it reappears again making the story move forward through the DM. However, it was present-oriented and expresses itself in two more ways. First, it meant describing the environment, especially (but not exclusively) by the DM. Secondly, it was related to performing in-game activities, using words only. As those purposes were visibly different to each other, we would propose using a new goal, "Performing" (PER), to depict the latter.

This small-scale analysis shows how different the RPG register is from others. The size of the dataset allowed us to look closely at each discourse unit, and find out what can be looked at in a bigger-scaled study. However, there is undeniably a need to investigate the subject further. The dataset missed a combat scene, which is regarded as the most important part of RPG by some of the players. As explained in Section 4, it might be especially useful to describe such scenes having a performative goal.

There is also no doubt that for a large-scale study the dataset could be more varied. It might be useful to look at different games and players, in various languages, since each RPG campaign is unique. Then, a comparison to different registers can be made, both to narrative ones and to casual conversations. Here, we used only the small, hand-annotated, dataset in order to draw the first conclusions and establish a framework.

References

Biber, D., & Conrad, S. (2009). Register, Genre, and Style. Cambridge University Press.

Biber, D., Egbert, J., Keller, D., & Wizner, S. (2021). Towards a taxonomy of conversational discourse types: An empirical corpus-based analysis. Journal of Pragmatics, 171, 20-35.

Boersma, P. & Weenink, D. (2022). Praat: doing phonetics by computer [Computer program]. Version 6.3.02, http://www.praat.org/ Accessed: 2.12.2022

Culler, J. (2011). Performative language. In: Culler, J., Literary Theory: A Very Short Introduction (2nd ed.), pp. 94-107.

Labov, W. (1980). Sociolinguistic Patterns. London, England: Blackwell.

Mearls, M. and Crawford J. (2014). Dungeons & Dragons player's handbook. Renton, WA: Wizards of the Coast.

Zagal, J. P. and Deterding, C. S. (2018). Definitions of Role-Playing Games. In: Role-Playing Game Studies: Transmedia Foundations. New York: Routledge, pp. 19-52.

SylLab: program for automatic sentiment analysis of poetry based on frequencies of phonetic units

Aleksandra Rykowska[1], Konrad Juszczyk[2]

[1]Jagiellonian University
aleksandra.rykowska@student.uj.edu.pl
[2]Adam Mickiewicz University
juszczyk@amu.edu.pl

Abstract

Computer stylometry is one of the basic methods of computer analysis of literature, mainly used in Poland to study prose. The paper proposes a new method of examining poetry by means of a computer program, aimed in automatic versological and rhythmic structure followed by the sentiment analysis. The theoretical part presents versological methods of analysing a poem and the significance of the characteristics of particular sounds of Polish language for the mood of words. For the purpose of the study, in which the above-mentioned linguistic aspects of the work play the key role, a program was written in Java language. The program is called SylLab and its workflow is based on Maria Dłuska's study. Main function of SylLab is a quantitative analysis of a given poem and a comparison with the most rhythmically similar works. The analysis is based on metrical characteristic of a poem, followed by a phonetic representation of verses. In contrast to word-based, bag-of-words approach, popular in nowadays stylometry, the program directly analyses both the metrics and phonetics, to establish similarities between poems and their sentiment.

Keywords: computer stylometry, computer poem analysis, sentiment analysis, emotional valence, semantic halo, **EDO 2023**

1. Introduction

Computational stylometry is one of the most rapidly growing fields in digital humanities. It is most often used to determine the authorship of texts. However, increasingly, various textual elements are also counted for other purposes, such as the automatic study of text semantics or the sentiment analysis.

Stylometry is based on counting text units relevant for its style. Statistically relevant results in stylometric studies are most often carried out on large samples, such as novels. Until now, short texts were rarely the subject of such research. This study changes this approach, and we propose applying stylometry analysis to shorter texts, namely the poems.

This paper presents a preliminary software model for a stylometric study of a Polish syllabic poem. The program calculates versological similarity among poems, which is an initial step in analysing the sentiment of a given work based on a quantitative study of the occurrence of palatalised and non-palatalised, as well as closed and open sounds in the work.

2. Previous research

Polish computational stylometry is thriving with the program *stylo* prepared by means of the R framework (Eder *et al.*, 2016). This program is used to analyse the most frequent words or n-grams in any text corpus. The program uses different types of distance measures, including Burrows' delta, cosine delta, Eder's delta, and others. However, this method provides good results only for the study of prose or longer dramas, as it has been shown that studies performed on poetry may not be statistically reliable (Eder, 2015, Eder and Rybicki, 2009).

Nevertheless, researchers have often turned to methods of examining words in poetic works to conduct stylometric analysis. This kind of analysis can be found in a study of Iranian (Razai and Kashanian, 2017), and English poetry (Kao and Jurafsky, 2012). The latter research uses the program PoetryANalyzer, which, in addition to basic word level methods, also analyses the rhyme structure of the poetry (converting the text from an orthographic to a phonological version), the alliterations, as well as consonances and assonances.

Sentiment analysis of selected contemporary English-language poems was also conducted using the aforementioned software (Kao and Jurafsky, 2015). However, the analysis conducted did not yield statistically significant results when analysing poetry from the Imagism strand. The research was also based on a word-based approach.

In an attempt to bypass the limitations of the bag-of-words method and the study of synsemantic sentence elements to analyse poetry, some attempts have been made to analyse poems on the basis of n-gram frequency for characters and metrical quantities (Forstal *et al.* 2011). However, this study showed poor separation of poets based on metrical n-gram features. Challenging this limitation, Ben Nagy used samples as small as ten verses, showing high efficiency in examining author style based on the metrical layer of a work (Nagy, 2021) in Latin poetry. These results were later confirmed by Plecháč, who described versification-based attribution as a reliable stylometric indicator (Plecháč, 2021).

Recent stylometric studies carried out with a versological and metrical factor prove that this approach to the study of poetry can be applied not only in authorship attribution. In the Russian tradition of literary studies, the theory of the semantic halo of metre has emerged, which states that the rhythmic flow of a poem somehow dictates its range of meaning (Trunin, 2017). This theory was often criticised due to the lack of empirical evidence for its veracity. This

situation changed after the application of stylometric methods supported by machine learning and deep learning. A semantic halo was proved for a corpus of Russian poems (Šeļa *et al.*, 2020). The authors predicted the presence of this phenomenon in any poetic tradition based on any versification system that allows for distinct and stable poetic form across time, a finding confirmed by a recent study of Czech, Russian and German poetry (Šeļa *et al.*, 2022). Findings from the article above strongly support the association between poetic meter and meaning, providing evidence for the theory of semantic halo.

The SylLab program, which is the subject of this article, was directly inspired by Maria Dłuska's interpretation of a poem by Kazimierz Wierzyński (2001). There, sentiment analysis is based on the juxtaposition of speech sounds ratios (palatalised vs. non-palatalised, and open vs. closed). The fact that the degree of palatalization of a given consonant can be indicative of its positive tone is supported by, among other things, experimental studies conducted with German speakers; however, using logatomes, which means the finding is relevant not only to German language (Körner, Rummer 2021).

3. Polish poem structure, versology and Polish consonant and vocative system

In the Polish literary tradition, there is a primary division of poetry according to the shape of the verses of a work, which includes three groups of poems – syllabic, syllabotonic, and tonic (Dłuska 2007). It is worth mentioning that this division is not able to encompass all the works that have been created, especially contemporary works, e.g., syntactic and anti-compositional verse. These types of poetry, however, are not the subject of the research presented in this paper.

In a syllabic poem, lines of one rhythmic measure are those with an equal number of syllables (Dłuska 2007). This type of poem is popular in Polish poetry since the time of Jan Kochanowski. It is worth noting that in this type of poem one accent is fixed according to the Polish rule of accentuation, where the penultimate syllable of the word is always accented. The syllabotonic system does not discard what Kochanowski had developed, but modifies it in its own way. It is due to the fact that in syllabotonic poems, we do not pay attention only to the number of syllables in each verse, but also to the arrangement of accents. According to Furmanik "the syllabotonic system thus involves a series of syllables in which not one but all accents are stabilised in terms of their number and place in the verse" (Furmanik, 1947). In tonic poem, on the other hand, neither the number of syllables in the verse, nor the distribution of accented syllables matters, but their equal number does (Furmanik, 1947).

3.1 Rhythmic structure of a syllabic poem

On the basis of the regularity in the occurrence of accented and unaccented syllables, we can distinguish different meters in the poem — repetitive arrangements of the same accentual feet. There are six types of meter in Polish ("s" means unstressed syllables and "S" stressed ones):

- trochaic meter: Ss | Ss | Ss | ..., consisting of trochees that have the scheme: ss,
- iambic meter: sS | sS | sS | ..., consisting of iambics which have the schema: sS,
- amphibrachic meter: sSs | sSs | sSs | sSs | ..., consisting of amphibrachs that have the schema: sSs,
- dactylic meter: sSs | sSs | sSs | ..., consisting of dactyls which have the scheme: sSs,
- anapestic meter: ssS | ssS | ssS | ..., consisting of anapests which have the schema: ssS,
- peonic meter: ssSs | ssSs | ssSs | ..., consisting of peons which have the schema: ssSs.

In poetry, it is widely accepted that the choice of rhythmic flow can affect the mood of a piece (Pszczołowska, 2008). For example, poems written in trochaic flow are not the most natural in our language, but they provide the text with a leaping rhythm that can indicate the positive mood of the piece.

3.2. Polish phonological system

Sentiment analysis in SylLab is based on the characterisation of Polish vowels and their quantitative ratios in the individual verses of a song. The program takes into account the opposition of opened/closed vowels and palatalised/non-palatalised consonants.

3.2.1 Polish vocal system

When considering the movements of the tongue in the oral cavity relative to the vertical axis, it can be observed that it is in its lowest position during the articulation of the vowel [a], while it rises upwards during the pronunciation of the other vowel sounds. The highest position of the tongue is observed during the articulation of [i]. The vowel chart represents the division of all vowels (Fig. 1).

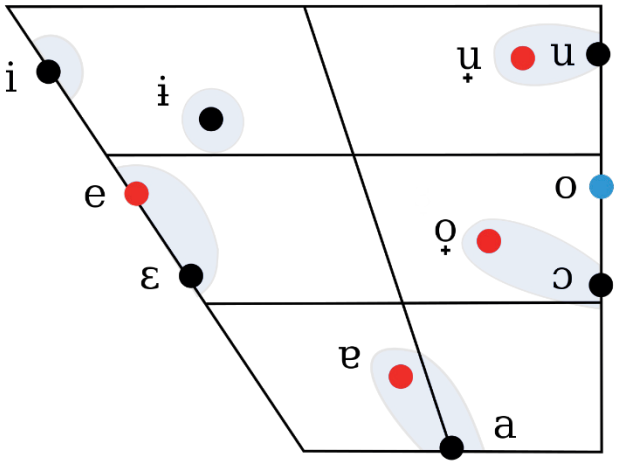


Fig 1. Polish vowel chart (Wiśniewski, 2007)

3.2.2 Polish consonant system and palatalisation

The movement of the tongue towards the hard palate is also essential during the articulation of consonants. In Polish, there is a group of dorsal consonants, which are characterised by the "accentuation of the centre of the tongue towards the palatum" (Ostaszewska, Tambor, 2000). These include: [ɕ], [ʑ], [t͡ɕ], [d͡ʑ], [ɲ], [j], [j̃], [c], [ɟ].

In addition to dorsal phonemes, there are also palatalised allophones in Polish, with "an additional articulatory movement — raising the middle part of the tongue towards the hard palate" (Ostaszewska and Tambor, 2000). These are the following consonants: [pʲ], [bʲ], [mʲ], [fʲ], [vʲ], [tʲ], [dʲ], [cʲ], [d͡zʲ], [nʲ], [sʲ], [zʲ], [t͡ʃʲ], [d͡ʒʲ], [ʃʲ], [ʒʲ], [xʲ], [rʲ], [lʲ], [wʲ]. Consonants in the articulation of which the tongue does not rise towards the hard palate, and in which there is thus no palatalisation, are classified as hard consonants.

4. Methods

The program presented in this paper is based on Maria Dłuska's (2001) interpretation of a poem by Kazimierz Wierzyński with the incipit *Gdzie nie posieją mnie....* The studied work is juxtaposed with a reference corpus consisting of two poems selected from a larger corpus in terms of versological similarity and similar time of composition (such action is motivated by the semantic halo theory). Then, individual vowels in each poem are counted, and their averages are presented in relation to the results obtained from the reference corpus.

The selection of works for the reference corpus is perhaps the most challenging task in the study. Therefore, an algorithm was established for carrying out the steps, shown in the diagram presented in Fig. 2.

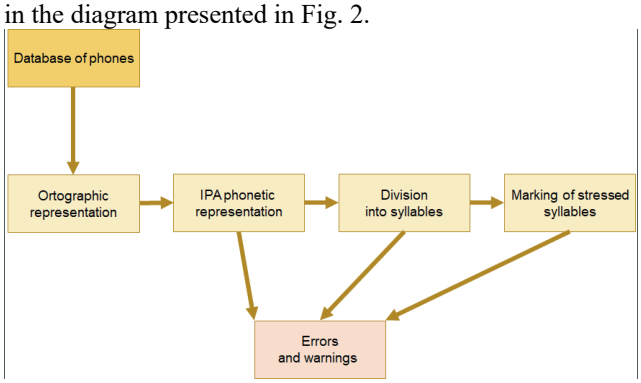


Fig. 2. SylLab's schema.

Each poem to be considered for the analysis must be annotated accordingly. The text is converted into its phonetic version (represented by the IPA alphabet) as shown in Fig. 3, then divided into syllables according to the principle of the sonority hierarchy (Śledzinski, 2016). Also, at this stage of the poem analysis, the software marks the accented syllables in blue which is represented in Fig. 4. Following the principle that in Polish, the accent is generally paroxytonic, the program marks the penultimate syllables of words as accented. There is also a list of exceptions, which includes words that do not have their accent - enclitics and proclitics (Wiśniewski, 2007). Violet is used to mark syllables accented laterally, i.e., optionally. From the syllabic representation of the piece, the program moves on to the marking of rhythmic feet in the verses. Each accented syllable is rearranged with a capital "S" and the unaccented ones are written as a lowercase "s". Side accents are also presented as capital letters but written in addition in round brackets "(S)" as shown in Fig. 5.

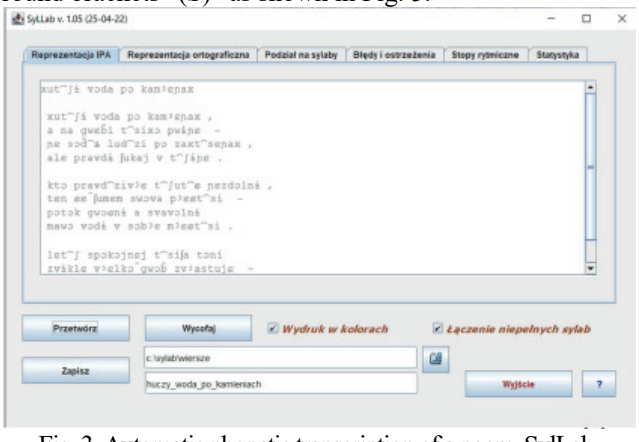


Fig. 3. Automatic phonetic transcription of a poem, SylLab.

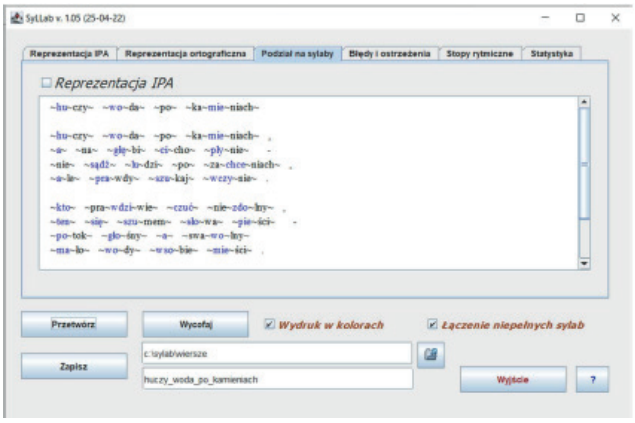


Fig. 4. Marked stressed syllables, SylLab.

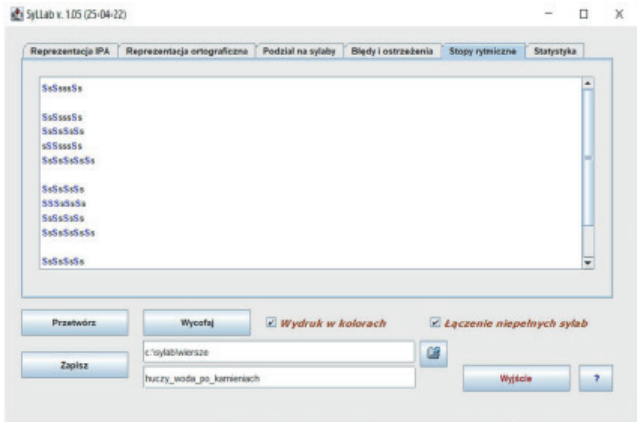


Fig. 5. Rhythmic structure of a poem, SylLab.

The last of the program's available options is a panel called 'Statistics'. Here, the program counts all occurrences of the given vowels and the occurrences of each vowel type in terms of their openness and consonants by their palatalisation. This information is directly taken into account in the study.

Fig. 6 shows how the program groups vowels and displays the number of occurrences of vowel categories concerned on the screen. Fig. 7 shows an inventory of frequency of all vowels, so that we can easily determine which vowels were counted consecutively by the program. In case of unforeseen errors or exceptions, the we can find the problematic line of the poem.

The statistical analysis shown in the screenshots of the program window presented in Fig. 6 and Fig. 7 is only a brief summary of the most important statistical counts of the vowels of the analysed work.

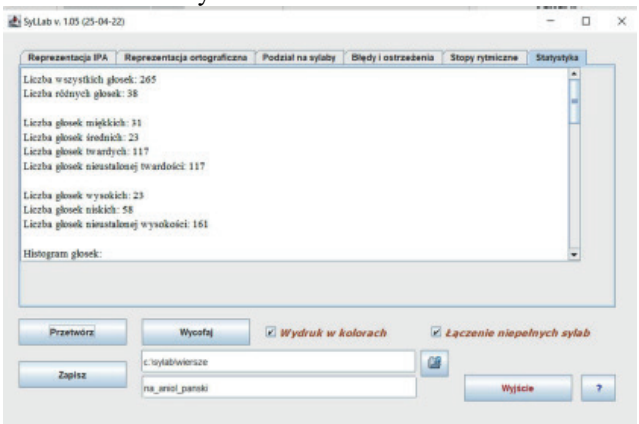


Fig. 6. Number of occurrences of the vowel categories, SylLab.

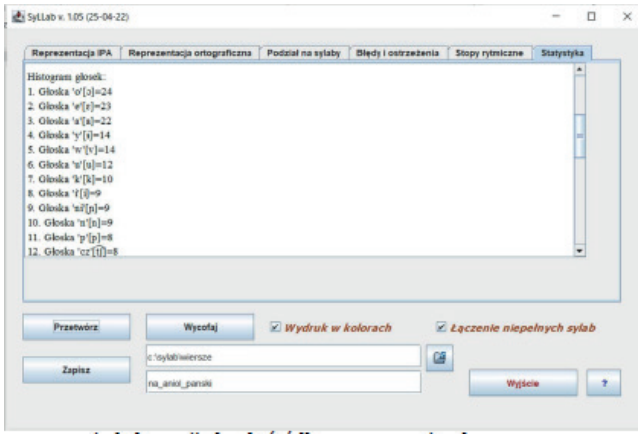


Fig. 7. An inventory of the frequency of all vowels, SylLab.

The exact data on the occurrences of specific vowels in each verse of the song and a summary of these calculations, together with the percentage of a given type of vowel in the whole song, is saved to a selected folder in a .csv (Excel-style) file.

5. Analysis of Wierzyński's poem

In order to carry out the study, 31 Polish poems were analysed using SylLab. The majority of them are regularly syllabic or syllabotonic and deviate slightly from regularity. The earliest poems included in the corpus were written in the era of Polish Romanticism, the latest in the 20th century. Several items in the corpus represent each literary era. All poems were also annotated manually in order to compare the results of the computer analysis and correct eventual errors.

The poems chosen as the reference corpus for the study of Kazimierz Wierzyński's work are *Na anioł pański* by Kazimierz Przerwa-Tetmajer and *Na szczytach* by Maria Konopnicka. These works are written in a regular nine-verse rhyme scheme, in which the rhythmic scheme can illustrate the rhyme scheme in the vast majority of the verses: sSsSsSsSs, i.e., they are written in an iambic four-stop.

Non-palatalised voicings make up 37% of the poem *Gdzie nie posieją mnie...*, whereas palatalised ones 22.3%. Open-mid and open vowels make up the same part of the work - 15.8%, while close vowels make up 9.1%. Poems by Maria Konopnicka and Kazimierz Przerwa-Tetmajer were also analysed. The results obtained for these two works are similar in many respects. The percentage share of non-palatalised phonemes in these two works is over 40% - for Konopnicka's poem, it is 45%, and for Tetmajer's, 43%. Palatalised allophones in both constitute about 15%.

What particularly distinguishes Wierzyński's piece is the significant difference between the occurrence of non-palatalised and palatalised sounds in the work. The non-palatalised ones are the most numerous of all the consonants in the work, which is natural for the Polish language, but in comparison with the works included in the reference corpus - there are fewer of them than in the works that do not represent such a joyful mood. In Wierzyński's work, non-palatalised sounds constitute only 36% of all sounds, which is as much as 8 percentage points less than the average in the works of Konopnicka and Przerwa-Tetmajer. To compare the magnitude of this difference - between the works *Na szczytach* and *Na anioł pański* this difference is only 2 percentage points. There is also a similar difference in the percentage share of palatalised sounds in the work. In Wierzyński's work, there are as many as 6 percentage points more palatalised allophones than in the reference corpus, indicating a definite accumulation of palatalised allophones in the work, which further results in the expression of its positive, joyful mood in the phonemic layer of the work. This difference between the songs from the reference corpus is only one percentage point. Smaller differences occur in the results for the openness of the vowels under study, although these too present themselves in favour of the thesis that in the stylometric analysis of the poem, we will obtain data indicative of the positive mood of the work. The percentage of open vowels in Wierzyński's poem is as much as three percentage points lower than in the corpus, and the number of occurrences of open-mid vowels is also significant. The above-presented results are summarized in Fig. 8 and Fig. 9.

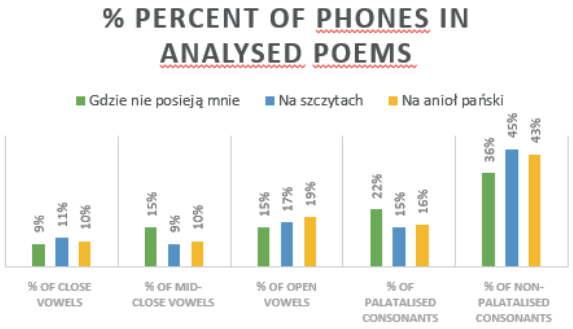


Fig. 8. Percent of phones in analysed poems.

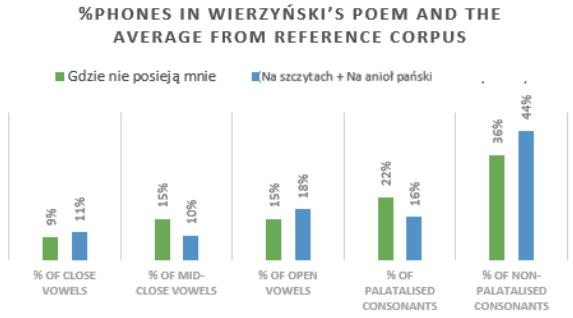


Fig. 9. Percent of phones in Wierzyński's poem and the average from reference corpus, own work

The significantly higher occurrences of palatalised-voiced words and thus the lower percentage of non-palatalised-voiced words in the poem are factors that make the poem perceived as happier, not only on the semantic level - the meanings of the words chosen - but also on the phonetic level, when perceiving the form of the work itself.

Thus, the study conducted confirms the results of Maria Dłuska's research; however, as addressed to much broaden number of poems, also significantly extends this research, and is a proof that such analysis could be, in large extent, automatized (which was not the case of Dłuska's work). The analysis of the statistical data obtained with the help of the SylLab program shows that the percentages of different categories of vowels supports the subjective readings and interpretations of the poem *Gdzie nie posieją mnie...* as well as the other poems under study.

6. Conclusions and future work

The analysis presented here is the beginning of a journey to a more accurate, and more automated stylometric analysis of Polish poems, taking into account the versological criterion. Even if this work should be treated as the initial step towards fully automated analysis, the results are very promising. The analysis should be enriched with further statistical procedures, and, as far as possible, the program should be developed on the basis of vector analysis methods (which have already begun to give positive results in the automatic comparison of the rhythmic turns of the poems).

Based on the semantic halo of meter theory, which would most likely also be confirmed in the Polish poems, one might risk saying that it is legitimate to compare poems of the same rhythmic structure with each other. The semantics would be imposed on these poems by the rhythmic structure, and differences in the sentiment of these poems can be observed on the basis of the relative juxtaposition of the different types of phones. .Moreover, differences in sentiment of these poems can be observed on the basis of relative juxtapositions of individual vowel types.

Future research should be carried out using a wider corpus of Polish poetry and juxtaposing other works to see if these methods are reliable. Although the performance of the SylLab program already seems satisfactory at this stage, the program itself is being successively improved as for the code efficiency and user interface.

References

Dłuska, M. (2001). "Kazimierz Wierzyński «Gdzie nie posieją mnie»". In: J. Prokop, and J. Sławiński, *Liryka polska. Interpretacje*, pp. 293–319. Gdańsk: słowo/obraz terytoria.

Dłuska, M. (2007). "Sylabizm". W *Wersyfikacja polska*, pp. 47–54. Lublin: Wydawnictwo Uniwersytetu Marii Curie-Skłodowskiej.

Eder, M., Rybicki, J. and Kestemont, M. (2016). "Stylometry with R: a package for computational text analysis". *R Journal* 8(1): 107-121. https://journal.r-project.org/archive/2016/RJ-2016-007/index.html

Eder, M. (2015). "Does Size Matter? Authorship Attribution, Short Samples, Big Problem". *Literary and Linguistic Computing*, *30*(2), 167–182. Retrieved from https://academic.oup.com/dsh/article/30/2/167/390738

Eder, M. and Rybicki, J. (2009). "PCA, Delta, JGAAP and Polish poetry of the 16th and the 17th centuries: who wrote the dirty stuff?" *Digital Humanities 2009: Conference Abstracts*. University of Maryland, College Park (MA), pp. 242-44.

Forstall, C. W., Jacobson, S. L., & Scheirer, W. J. (2011). "Evidence of intertextuality: investigating Paul the Deacon's Angustae Vitae". *Literary and Linguistic Computing*, *26*(3), 285–296. Retrieved from https://academic.oup.com/dsh/articlelookup/doi/10.1093/llc/fqr029

Furmanik, S. (1947). *Podstawy wersyfikacji polskiej : (nauka o wierszu polskim).* Wydawnictwo E. Kuthana.

Kao, J., and Jurafsky, D. (2012). "A Computational Analysis of Style, Affect, and Imagery in Contemporary Poetry". In *Proceedings of the NAACL-HLT 2012 Workshop on Computational Linguistics for Literature* pp. 8–17. Montreal, Canada: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/W12-2502

Kao T., and Jurafsky D. (2015). "A computational analysis of poetic style: Imagism and its influence on modern professional and amateur poetry". In *Linguistic Issues in Language Technology, Volume 12, 2015 - Literature Lifts up Computational Linguistics*. CSLI Publications.

Körner, Anita i Ralf Rummer. 2021. "Articulation contributes to valence sound symbolism". *Journal of Experimantal Psychology General*. DOI: 10.1037/xge0001124

Nagy, B. (2021). "Metre as a stylometric feature in Latin hexameter poetry". *Digital Scholarship in the Humanities* [advance articles]. doi: 10.1093/llc/fqaa043.

Ostaszewska, D., and Tambor J. (2000). *Fonetyka i fonologia współczesnego jezyka polskiego*. Warszawa: Wydawnictwo Naukowe PWN.

Plecháč P. (2021). *Versification and Authorship Attribution*. Institute of Czech Literature of the CAS.

Pszczołowska, L. (2008). "Comparative Slavic Metrics. Evolution of Aims and Methods of Investigation". *Stylistyka*, Vol. 17, pp. 347-359.

Rezaei S., and Kashanian N. (2017). "A Stylometric Analysis of Iranian Poets" *Theory and Practice in Language Studies,* Vol. 7, No. 1, pp. 55-64.

Śledziński, D. (2016). "Tworzenie reguł dla programu dzielącego tekst w języku polskim na sylaby". *Biluteyn Polskiego Towarzystwa Językoznawczego*, t. 72: 151-161.

Šeļa A, Orekhov B, and Leibov R. Weak. (2020). "Genres: Modeling Association Between Poetic Meter and Meaning in Russian Poetry". In: *CHR 2020: Workshop on Computational Humanities Research*. Amsterdam: CEUR-WS. pp. 12–31. Retrieved from: http://ceur-ws.org/Vol-2723/long35.pdf.

Šeļa A, Plecháč P., and Lassche A. (2022) "Semantics of European poetry is shaped by conservative forces: The relationship between poetic meter and meaning in accentual-syllabic verse". *PLoS ONE* 17(4): e0266556.

Trunin, M. (2017). "Towards the concept of semantic halo". *Studia Metrica et Poetica*, Vol. 4.2, pp. 41-66.

Wiśniewski, M. (2007). *Zarys fonetyki i fonologii współczesnego języka polskiego*. Toruń: Wydawnictwo Uniwersytetu Mikołaja Kopernika.

Utilizing Wikipedia for Retrieving Synonyms of Trade Security-related Technical Terms

Rafal Rzepka[1], Shinji Muraji[2] and Akihiko Obayashi[3]

[1]Faculty of Information Science and Technology, Hokkaido University, Japan
[2]Graduate School of Information Science and Technology, Hokkaido University, Japan
[3]Center for Innovation and Business Promotion, Hokkaido University, Japan
[1]rzepka@ist.hokudai.ac.jp, [2]shinjimuraji@ist.hokudai.ac.jp, [3]obayashi@mcip.hokudai.ac.jp

Abstract

Measuring semantic similarity of technical terms is not a trivial task especially for multi-word terminology. Matching term equivalents in documents for decision-making in fields like medicine or law requires high precision, therefore instead of using popular statistical methods, synonym lists are being manually created, which is the most certain but costly solution. Although many similarity-based methods have been proposed, we have not found any reports on how useful are human-made redirections included in Wikipedia pages. In this paper we report results of our experiments for discovering synonyms of terms related to trade security in Japanese language by using redirections and compare them with simple entity-linking approach. We perform a series of experiments using a synonym list prepared by the government specialists as the gold set. Strictness of the evaluation setup resulted in low scores, but we confirmed which heuristics have more potential than others. We discuss several findings which shed some light on how they could be utilized to solve the difficult task of extracting similar technical terms.

Keywords: Technical Terms, Wikipedia, Multiword Level Similarity, Trade Security, Export Control

1. Introduction

In recent years, trade security has gathered international attention not only on the level of governments or industries but also at universities. However, with some exceptions (Rzepka et al., 2021; Matsuzawa and Hayasaka, 2022) almost no NLP research deals with this topic. One of the biggest difficulties is the fact that except the legal jargon, the trade-security terminology covers various fields from nuclear physics through biology to engineering. Official vocabulary often differs from what the one that researchers use, and it becomes a problem when an artificial agent communicates with a user in order to navigate her or him to a proper passage of the regulatory text. Some terms can carry different weigh in terms of danger, therefore the usage of common similarity-based approaches becomes problematic. Hence, avoiding a loose fuzzy matching whenever possible can limit the number of agent's erroneous advices. In the case of Japanese language which does not use spaces, there is an additional problem of word segmentation which is not trivial with technical terms across several fields. When we tested a glossary list prepared by the CISTEC[1] organization (1,660 terms), Japanese morphological tool Janome has divided 72.5% (1,204/1,660) of all technical terms which are theoretically single semantic entities according to the glossary authors. On the other hand the Japanese government provides a list of official synonyms (or rather synonymical phrases) for no more than 6% of these terms. The research question that appears is "how could we automatically enrich such a list with a high confidence?" and in this research explores the possibility to utilize Wikipedia for discovering closely related terms. Our paper is structured as follows: in Section 2. we list some common approaches to discovering technical terms and measuring their similarities; in Section 3. we introduce eight approaches we tested; in Sections 4. and 5. we describe the experiments and their results. We discuss the results in Section 6. and conclude our paper with Section 7.

[1]https://www.cistec.or.jp/english/export/

2. Related Work

Technical terms have been approached in the field of natural language processing from different angles (Butters and Ciravegna, 2008). One obvious scenario is to discover them with their equivalents in other language to achieve higher quality machine translation. Bilingual corpora or datasets are utilized and matching the closest terms is the main target. For example, (Bollegala et al., 2015) propose prototype vector projection (PVP) which is a non-negative lower-dimensional vector projection method to help compiling large-scale bilingual dictionaries for technical domains. In the same paper they propose a method to learn a mapping between the feature spaces in the source and target language using partial least squares regression (PLSR) which requires only a small number of training instances to learn a cross-lingual similarity measure. The proposed method outperforms several other feature projection methods in biomedical term translation prediction tasks.

In the context of trade security, (Matsuzawa and Hayasaka, 2022) propose a method for associating technical documents and legal statements of export control in English and Japanese. Although the main goal is to find dissimilarities on the sentence level, the authors underline the importance

of proper matching technical terms which are crucial for avoiding misunderstanding of regulatory texts.

In their recent study, (Liwei, 2022) tackle the problem of technical term matching between English and Chinese patents. After testing many approaches including deep learning-based ones, they discovered that adapting C-value (Frantzi et al., 2000), a hybrid terminology extraction method combining linguistic rules and statistical theory, to specific domains, yields the best results. This newly proposed DC-value method combined with information entropy successfully extracted Chinese technical terms outperforming the original C-value method, the log-likelihood ratio method and the mutual information method (Church and Hanks, 1990).

Another relatively popular target is to find acronyms or abbreviations of technical terms. For example (Yagahara and Sato, 2020) automatically extract full forms from abbreviations by using word2vec for terminology expansion in the "image diagnosis"-related abstracts retrieved from PubMed. They determine the optimal word2vec parameters that ensure the highest accuracy, which was Skip-gram with 200 dimensions and 10 iterations achieving 74.3%. Although recognizing acronyms like "ldr" stating for "low dose rate" seem simple enough to use heuristics, errors like "bb" assigned to *biobreeding* instead of the correct *black blood*, underline the importance of context processing.

Simplifying the technical documents is another task where technical terms are important. The task is to identify them and replace with simple equivalents to make a document easier to comprehend for a layperson. (Abrahamsson et al., 2014) have improved an existing method for assessing difficulty of words in Swedish text. The difficulty of a word was assessed not only by measuring the frequency of the word in a general corpus, but also by measuring the frequency of substrings of words. By doing so they adapted the method to the compounding nature of Swedish, signaling that language specific approaches are important to develop bilingual thesauri.

Wikipedia is a valuable source for finding similar terms (Hwang et al., 2011). An early example of how to use inter-wiki links to extract named entities and rank synonyms is the work of (Bøhn and Nørvåg, 2010), who used, except heuristics, the frequencies of inter-wiki links which inspired our use of thresholds. More recent is an approach proposed by (Jagannatha et al., 2015), who use Wikipedia for automatic extraction of synonyms related to the biomedical domain. By using inter-wiki links, they extract the candidate synonyms (which are not technical terms) of an anchor-text in a Wikipedia page and the title of its corresponding linked page. They rank synonym candidates with word embedding and PRF (pseudo-relevance feedback). They found that PRF-based re-ranking outperforms word embedding based approach and a strong baseline using inter-wiki link frequency. Furthermore, their results showed that a hybrid method (namely Rank Score Combination), achieved the best results and upon this finding we also tested combinations of our implemented methods.

3. Tested Methods

3.1. Redirect-based Methods

In usual Wikipedia terminology, a redirect indicates a type of article that sends the reader to another article when there are different names for the same subject. For example, when "USA" is input in the search box, Wikipedia displays the page of "United States". In this research, we utilize redirects in a slightly different manner. Using the example of the "United States"-related page example, it contains a phrase "*scientific* force" as a linked string, and its link redirects to "Science and technology in the United States" page. It is not uncommon that the linked string and the title of the linked page are different, and the link in Wikipedia's HTML contains the title of the linked page[2]. For the purpose of this approach we assume that linked phrase and the title of the linked have similar meaning, and such pairs can form a thesaurus. However, this heuristic is not perfect due to ofter arbitrary way how the Wiki creators create such links. For instance, in the example above, only the word "scientific" is linked although the linked page is related to "scientific force". We assume that if a phrase is linked to a target page only once, there is a high probability that it is an unusual combination and it may cause noise. To confirm this hypothesis, we propose an additional method which collects pairs only if a phrase is linked two or more times to a give target title. We call these methods REDIRECTING and REDIRECTING WT (**W**ith **T**hreshold). To investigate the effectiveness of redirect's opposite direction, namely when the redirected page (here "Science and technology in the United States") sends back to the target word page ("United States" in our example), we add a pair of algorithms implementing this approach and name them REDIRECTED and REDIRECTED WT.

3.2. Inner-Link-based Method

In this method we use a Wikipedia page of the target word (if it exists), and assume that all linked words are related and probably synonymous. For example, in the Wiki page of "photodetector" (*hikari kenshutsu-ki*[3]), we can find inks to Japanese terms for "photomultiplier tube" or "solar cell". Similarly to the REDIRECTING and REDIRECTING WT methods, we add the same threshold and call the methods LINKING and LINKING WT, respectively. Furthermore, we also construct algorithms for checking the opposite direction ("photomultiplier tube" linking back to "photodetector") and call the additional methods LINKED and LINKED WT.

4. Experiment

In this section we explain how we tested the above-described heuristics by matching the results with expert-created thesaurus.

[2] `https://en.wikipedia.org/wiki/Science_and_technology_in_the_United_States` in this example

[3] `https://bit.ly/3j7ZBhP`

Approach	Precision	Recall	F-score
(1) REDIRECTING	0.0700 (14/200)	0.0927 (14/151)	0.0798
(2) REDIRECTING WT	0.1558 (12/77)	0.0795 (12/151)	**0.1053**
(3) REDIRECTED	0.2188 (7/32)	0.0464 (7/151)	0.0765
(4) REDIRECTED WT	0.2143 (3/14)	0.0199 (3/151)	0.0364
(1)+(3)	0.0806 (17/211)	0.1126 (17/151)	0.0939
(1)+(4)	0.0683 (14/205)	0.0927 (14/151)	0.0787
(2)+(3)	0.1868 (17/91)	0.1126 (17/151)	**0.1405**
(2)+(4)	0.1667 (14/84)	0.0927 (14/151)	0.1191

Table 1: Experimental results for the **redirect**-based approach and the combinations of its methods. Bold font is used for top F-scores in both single and hybrid approaches, highest **precision** scores are underlined. Numbers in brackets indicate number of terms matched with gold set / numbers of found synonyms (precision) and number of terms matched with gold set / number of all synonyms in the gold set (recall).

Approach	Precision	Recall	F-score
(5) LINKING	0.0033 (14/4305)	0.0927 (14/151)	0.0063
(6) LINKING WT	0.0040 (11/2720)	0.0728 (11/151)	0.0077
(7) LINKED	0.0033 (10/2995)	0.0662 (10/151)	0.0064
(8) LINKED WT	0.0103 (3/290)	0.0199 (3/151)	**0.0136**
(5)+(7)	0.0031 (14/4494)	0.0927 (14/151)	0.0060
(5)+(8)	0.0032 (14/4318)	0.0927 (14/151)	0.0063
(6)+(7)	0.0035 (11/3134)	0.0728 (11/151)	0.0067
(6)+(8)	0.0040 (11/2743)	0.0728 (11/151)	**0.0076**

Table 2: Experimental results for the **inner-links**-based approach and the combinations of its methods. Bold font is used for top F-scores in both single and hybrid approaches, highest **recall** scores are underlined. Numbers in brackets indicate number of terms matched with gold set / numbers of found synonyms (precision) and number of terms matched with gold set / number of all synonyms in the gold set (recall).

4.1. Data

Here we describe the data used for experiments – the source of links and the test dataset of synonyms.

4.1.1. Japanese Wikipedia

For the experiments we have downloaded latest dump of Japanese Wikipedia[4] with *wikiextractor* tool[5]. Redirects, linked phrases and target page titles have been extracted from HTML code with the BeautifulSoup library for Python.

4.1.2. Test Set

The gold set of term examples with their synonyms (*Yomikaehyou*) has been downloaded from the Export Control page of Japanese Ministry of Economy, Trade and Industry[6]. There are currently (as for January 25, 2023) 83 examples in the set. Because it contains sentences as "measuring equipment that uses linear variable differential transformers (LVDTs)", we removed all entries including verbs, as they are not technical terms but rather descriptions of their categories that cannot be precisely matched (77 is the number of terms after removing sentences). Most of the gold set terms have more than one synonym, for example "Solid-state cameras: CCD cameras, CMOS cameras". In some cases differences are in type of writings. For example term "photodetector" has three separate synonyms: "phototransistor", "photodiode" written in Chinese characters and "photodiode" written in katakana syllables used for loan words.

[4] https://dumps.wikimedia.org/jawiki/, version 20230101.

[5] https://github.com/attardi/wikiextractor

[6] https://www.meti.go.jp/policy/anpo/matrix_intro.html

4.2. Experimental Setup

We used every target word from the thesaurus described above and run the algorithms explained in Section 3.

5. Experimental Results

The results presented in Tables 1 and 2 show that redirect-based approach yields much better results than utilizing inner-links. The highest F-score for single methods is achieved by REDIRECTING WT but improved when this method is combined with REDIRECTED. When it comes to precision, also redirect functionality obtained better scores, but this time a single method (REDIRECTED) appeared to be higher than the best combination (REDIRECTING WT with REDIRECTED). On the other hand, overall scores of inner-links-based methods were minuscule with over 10 times lower F-score when compared to the redirect-based ones, meaning that recall has not improved the results as expected. None of the combinations scored higher than the single LINKED WT method, showing that implementing threshold removed many problematic synonym candidates.

Target Term	$Synonym_1$	$Synonym_2$	$Synonym_3$	$Synonym_4$	$Synonym_5$
asshuki (compressor)	*dendo kuuki asshuki (electric air compressor)*	*kuuki asshuki (air compressor)*	*konpuressaa (compresser)*	*eakonpuressaa (air compressor)*	***konpuressa (compressor)***
uran (uranium)	*U*	***uraniumu (uranium)***	*uran-235 (uranium-235)*		
genshiro atsuryoku youki (reactor pressure vessel)	*atsuryoku youki (pressure vessel)*	***genshiro youki (reactor vessel)***			
kotai satsuzou soshi (solid state image sensor)	*satsuzou soshi (image sensor)*	*imeeji sensaa (image sensor)*	*imeeji sensa (image sensor)*	*satsuei soshi (image sensor)*	
jikuuke (bearing)	***bearingu (bearing)***	*jikuu-ke (bearing)*	*rooraa bearingu (rolling-element bearing)*		
shuuseki kairo (integrated circuit)	***IC***	***LSI***	*chippu (chip)*	*IC chippu (IC chip)*	*VLSI*
shinkuu ponpu (vacuum pump)	***bakyuumu ponpu (vacuum pump)***	*kou-shinkuu ponpu (high vacuum pump)*			
tanso sen'i (carbon fiber)	***kaabon faibaa (carbon fiber)***	*kaabon (carbon)*	*kaabon-faibaa (carbon fiber)*	*tanso sen'i kyouka purasuchikku (carbon fiber reinforced plastics)*	*tanso-kei (carbon related)*
hakkou daioodo (light emitting diode)	***LED***	*furu karaa LED (full color LED)*	*LED-shiki (LED type)*	*aoiro hakkou daioodo (blue light emitting diode)*	*LED raito (LED light)*
hikari kenshutsu-ki (photodetector)	*hikari sensaa (light sensor)*	*kenshutsu-ki (detector)*	*hikari sensa (light sensor)*		
ben (valve)	***barubu (valve)***				
mujin koukuu-ki (unmanned aerial vehicle)	***UAV***	***doroon (drone)***	*mujin-ki (unmanned vehicle)*	*mujin teisatsu-ki (unmanned reconnaissance vehicle)*	*mujin (unmanned)*
rejisuto (resist)	***fotorejisuto (photoresist)***				

Table 3: Synonyms for target words extracted by the REDIRECTING WT method. Technical terms which exist in the gold set are marked with bold font. Due to the space constraints only up to five synonyms are given (10 out of total 54 have been truncated).

However, decreasing the number of candidates from 2,995 to 290 also lead to decreasing correct discoveries from 10 to 3.

6. Discussion

While we expected the inner-links methods too be weak as it treats all linked words as potentially related words, the methods based on the Wikipedia's redirect functionality, even if much better than inner-links, appeared to be far from perfect. Our assumption was that because Wikipedia creators use their knowledge to create meaningful links between pages, it will be possible to achieve a relatively high precision. As we deal with very specific expert knowledge which is not so widely represented in Wikipedia as, for instance, field of medicine or biology (Yang and Colavizza, 2022), the recall was not expected to be high. Moreover, the thesaurus used as the gold set is meant for experts, while Wikipedia is made by and targeted mostly by non-experts. This lead to the situation where a target word is very often redirected to more popular synonyms, while the gold set contains also less obvious equivalents. For example, in Table 3 which presents part of results of REDIRECTING WT method, popular synonyms of "integrated circuit" like "IC" or "LSI" are found, while in the thesaurus made by export control experts we can find synonyms like "monolithic IC" or "hybrid IC" written entirely in English. For certain, small size of the thesaurus and poor coverage of the terms in Wikipedia led to very low scores. Out of 77 terms in the gold set, only 21 had their pages in Wikipedia and the total number of gold thesaurus synonyms for these

terms with dedicated pages was 47. Of these, 14 were correctly extracted using the LINKING method, which means recall of 29.8% if only the terms with pages are considered. The lack of relevant content in both datasets seem to be a major problem, however it must be noted that our testing method is very strict. For example, when we showed full version of Table 3 to an export control expert, he ruled out only 10 out of 54 extracted synonyms as most probably improper to be included in the official list. If we had access to many experts we could perform more suitable evaluation experiment, unfortunately there are only few of them in Japan.

7. Conclusion

In this work we tested how inner-links and redirect functionality of Wikipedia can help to find synonyms of technical terms regarding export control regulations for the trade security. We discovered that although redirect-based methods yield much better results than inner-links, the expert-made thesaurus used for evaluation has too few overlaps with Wikipedia to achieve satisfactory F-score. However, a small evaluation performed by a single expert suggest that the tested methods have much bigger potential than the scores indicate.

8. Future Work

To improve the results, in the near future we are planning to implement similarity measures of linked pages and combine older approaches which utilize context clustering (Courseault Trumbach and Payne, 2007; Judea et al., 2014). We will also test various extraction methods and tools to enlarge the number of synonym candidates also for lay-person term equivalents similarly to the work performed by (Sandoval et al., 2019). By generating high-quality synonym candidates list we will aim to lessen the burden of experts who have to manually check the appropriateness of the technical terms. When the goal is achieved, we plan to extend the government-created thesaurus by finding all possible synonym candidates for the glossary published by the Japan Machinery Center for Trade and Investment.

9. Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 20K12556.

References

Abrahamsson, Emil, Timothy Forni, Maria Skeppstedt, and Maria Kvist, 2014. Medical text simplification using synonym replacement: Adapting assessment of word difficulty to a compounding language. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*.

Bøhn, Christian and Kjetil Nørvåg, 2010. Extracting named entities and synonyms from wikipedia. In *2010 24th IEEE International Conference on Advanced Information Networking and Applications*. IEEE.

Bollegala, Danushka, Georgios Kontonatsios, and Sophia Ananiadou, 2015. A cross-lingual similarity measure for detecting biomedical term translations. *PloS one*, 10(6):e0126196.

Butters, Jonathan and Fabio Ciravegna, 2008. Using similarity metrics for terminology recognition. In *LREC*.

Church, Kenneth and Patrick Hanks, 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Courseault Trumbach, Cherie and Dinah Payne, 2007. Identifying synonymous concepts in preparation for technology mining. *Journal of Information Science*, 33(6):660–677.

Frantzi, Katerina, Sophia Ananiadou, and Hideki Mima, 2000. Automatic recognition of multi-word terms:. the c-value/nc-value method. *International journal on digital libraries*, 3:115–130.

Hwang, Myunggwon, Do-Heon Jeong, Seungwoo Lee, and Hanmin Jung, 2011. Measuring similarities between technical terms based on wikipedia. In *International Conference on Internet of Things and on Cyber, Physical and Social Computing*.

Jagannatha, Abhyuday, Jinying Chen, and Hong Yu, 2015. Mining and ranking biomedical synonym candidates from Wikipedia. In *Proceedings of the sixth international workshop on health text mining and information analysis*.

Judea, Alex, Hinrich Schütze, and Sören Brügmann, 2014. Unsupervised training set generation for automatic acquisition of technical terminology in patents. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical Papers*.

Liwei, Zhang, 2022. Chinese technical terminology extraction based on dc-value and information entropy. *Scientific Reports*, 12(1):20044.

Matsuzawa, Keiichi and Mitsuo Hayasaka, 2022. Associating technical documents and export control laws using Japanese-English translation for regulatory compliant data access control (in Japanese). In *Proceedings of The 36th Annual Conference of the JSAI, 1D5-GS-11-03*.

Rzepka, Rafal, Daiki Shirafuji, and Akihiko Obayashi, 2021. Limits and challenges of embedding-based question answering in export control expert system. In *Proceedings of the 25th International Conference on Knowledge-Based and Intelligent Information & Engineering System*. Szczecin, Poland: Springer.

Sandoval, Antonio Moreno, Julia Díaz, Leonardo Campillos Llanos, and Teófilo Redondo, 2019. Biomedical term extraction: NLP techniques in computational medicine. *IJIMAI*, 5(4):51–59.

Yagahara, Ayako and Tetta Sato, 2020. Evaluation of the automatic full form retrieval method from abbreviation using word2vec for terminology expansion (in Japanese). *Nihon Hoshasen Gijutsu Gakkai Zasshi*, 76(11):1118–1124.

Yang, Puyu and Giovanni Colavizza, 2022. A map of science in Wikipedia. In *Companion Proceedings of the Web Conference 2022*.

Hard is the Task, the Samples are Few: A German Chiasmus Dataset

Felix Schneider[1], Sven Sickert[1], Phillip Brandes[2], Sophie Marshall[2], Joachim Denzler[1]

[1]Computer Vision Group, [2]Institut für Germanistische Literaturwissenschaft
Friedrich Schiller University Jena
{firstname}.{lastname}@uni-jena.de

[2]Eberhard Karls University Tübingen
phillip.brandes@uni-tuebingen.de

Abstract

In this work we present a novel German language dataset for the detection of the stylistic device called chiasmus collected from German dramas. The dataset includes phrases labeled as chiasmi, antimetaboles, semantically unrelated inversions, and various edge cases. The dataset was created by collecting examples from the GerDraCor dataset. We test different approaches for chiasmus detection on the samples and report an average precision of 0.74 for the best method. Additionally, we give an overview about related approaches and the current state of the research on chiasmus detection.

Keywords: chiasmus detection, antimetabole detection, stylistic device detection, stylometry, dataset

1. Introduction

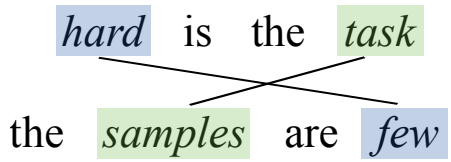


Figure 1: Example of a chiasmus

In this paper, we present a novel dataset containing annotated chiasmus examples. We make this dataset available online [1]. The chiasmus is a figure of speech which comprises an inversion of semantically related concepts, as depicted in Figure 1. This stylistic device can have various uses, such as expressing contrasting concepts. However, it is only sparsely used in literature, making it hard to gather enough instances to train a modern classifier or to conduct meaningful statistical analysis. Brandes et al. (2022) show that the automated detection of chiasmi can provide a useful tool in the field of literary studies. In their case, it is used to find differences between three genres of medieval texts.

The chiasmus dataset can also be used to test the general capabilities of language models. It is no trivial task to fully define from which point on inverted words are semantically related enough to be considered a chiasmus. Consistently detecting chiasmi is a task that requires a nuanced semantic knowledge of the underlying language included in whatever model is used for this task. Thus, this dataset is not only of interest for reseachers in the field of stylistic device detection, but also as a benchmak for the development of language models in general. The dataset consists of German chiasmi, antimetaboles and random inversions without semantic meaning. It was compiled by using a chiasmus detection method (Schneider et al., 2021) on the GerDraCor (Fischer, 2019) dataset while annotating the most highly rated chiasmus candidates. Furthermore, it is enriched by random inversions taken from the GerDraCor corpus.

In the following, we give an overview over related work on chiasmi in Section 2.. After that, we provide the details of the chiasmus dataset in Section 3.. In Section 4., we present and discuss the results of a baseline classifier on the dataset. Finally, Section 5. concludes the paper with a summary of our work.

2. Related Work

Research on chiasmi and their detection originates mainly from two areas of science. Many methodical contributions can be found in the area of computer science, while works from the field of literary studies focus on their historical occurence and meaning in the context of texts.

2.1. Computer Science

A first method to detect antimetaboles was introduced by Gawryjolek (2009). The author presents an approach which searches for repetition of words to find possible antimetaboles. While this approach uses no filtering steps, it can be considered the first step towards the automatic detection of these stylistic devices. Analogously, Java (2015) provides a method for finding general chiasmus candidates based on the inversion of syntax trees. This method also has no filtering step to remove random inversions – instead, the requirement for a full inversion of the syntax tree already narrows down the number of detected candidates. The method potentially misses candidates that can be found by looking at part-of-speech tag repetitions, but

[1] https://github.com/cvjena/chiasmus-annotations

its strong requirement also removes false positives. Lim (2016) also search for chiastic structures by locating repeating words without additional filtering. This approach is capable of finding also longer structures like *A B C ... C' B' A'*. As a limitation, it lacks the means to remove false positives. However, the method is suitable to find more deeply stacked chiastic structures, which can not be reliably detected by other methods. Especially with very deeply stacked structures, the likelihood of such often repeated random inversions should be lower than with the two-level chiastic structures presented in this work.

A method to detect antimetaboles using inversions of repeated lemmata was created by Dubremetz and Nivre (2018). In their work, they also searched for inverted repetitions of lemmata, but added a filtering step afterwards. It ranks the chiasticity of the sample based on a machine learning model, which uses several features. This approach is part of a series of works, where they first introduced a manual ranking system (Dubremetz and Nivre, 2015). Later on, they introduced a machine learning approach and added new sets of features (Dubremetz and Nivre, 2016). They test their approach on a dataset of antimetaboles in English created from the Europarl dataset. Schneider et al. (2021) extend this method to find general chiasmi using inversions of repeated part-of-speech tags instead of antimetaboles by inversions of repeated lemmata. They also rank them by a machine learning model. To cope with large amounts of false positives in their part-of-speech tag based approach, they add cosine distances between the word embeddings of the supporting tokens and lemma repetition information to their set of features.

2.2. Literary Studies

In literary studies, the chiasmus is studied as a stylistic device and as a wider structure in text. Welch (2020) gives a broad overview of chiasmus used in antique texts. The work covers a great timespan, beginning from sumero-akkadian literature, covering the Old and New Testament, until the era of ancient greek and latin texts. More recent texts are covered by Brandes et al. (2022). They analyze the use of chiasmus in Middle High German texts in the *Trois Matières*. In their work, they use automatic chiasmus detection techniques to compare the styles of different texts between genres, times and authors.

3. The Dataset

In the following, we describe our proposed dataset. Before we get to the details, we will summarize what a chiasmus is based on definitions found in the literature.

3.1. Chiasmus Definition

There are many definitions of the word chiasmus, which can include very informal descriptions by literary scholars. Those also include purely semantic chiasmi, that comprise opposing concepts, but are not represented by a certain syntactical structure (Welch, 2020). We acknowledge this diversity in the use of the term *chiasmus*. However, we need to set some constraints to operationalize the term for the use in a dataset and benchmark.

Type		Samples
chiasmus	(c)	31
antimetabole	(a)	39
negative examples	(x)	242
random negative examples	(xr)	4000
parallelism	(fp)	76
antithetic parallelism	(ap)	23
false antimetabole	(fa)	3
false chiasmus	(fc)	7
false antithetic parallelism	(fap)	2
false parallelism	(ffp)	2
synthetic parallelism	(fsp)	22

Table 1: The number of samples per class in our proposed chiasmus dataset.

In the context of this dataset a chiasmus is considered a cross-wise inversion of semantically related words which can be read as a stylistic device. We further define this as a cross-wise repetition of part-of-speech tags in an *A B B' A'* pattern. An example for this would be the sentence ***narrow** is the **world** and the **brain** is **wide***, comprising the supporting tokens *narrow, world – brain, wide* and the inverted part-of-speech tags *ADJ, NOUN – NOUN, ADJ*.

A special case of this chiasmus definition is the antimetabole, which consists of an inverted repetition of part-of-speech tags and lemmata. An example for this would be the sentence ***one** for **all**, **all** for **one***, with the repeated supporting tokens *one, all – all, one*. Further, we only consider chiasmi comprising two word pairs. More complex chiasmi like *A B C C' B' A'* patterns are not part of this work.

3.2. Chiasmus Annotations

The samples in our dataset are annotated in different manners. In additon to the base classes of chiasmus, antimetabole and inversion without special semantic meaning, we also include parallelisms. The parallelisms are instances, where the part-of-speech tags of all four supporting tokens are similar, leading to the *A A' A" A"'* structure matching the *A B A' B'* structure of a parallelism. There are also special cases, where the sample constitutes a chiasmus, but the main supporting tokens are not marked. Table 1 gives an overview over the different samples in the data. In the following, we quickly summarize and define the classes used in the dataset.

Chiasmus (c) represents the standard case of a chiasmus with repeating inverted part-of-speech tags, excluding antimetaboles.

Antimetabole (a) stands for antimetaboles, defined as an inverted repetition of lemmata.

Parallelism (fp) stands for parallelisms in the form of *A B A' B'*. Since the candidates are generated by searching for an *A B B A* pattern in the part-of-speech tags, also *A A A A* candidates are found, resulting in some parallelism examples.

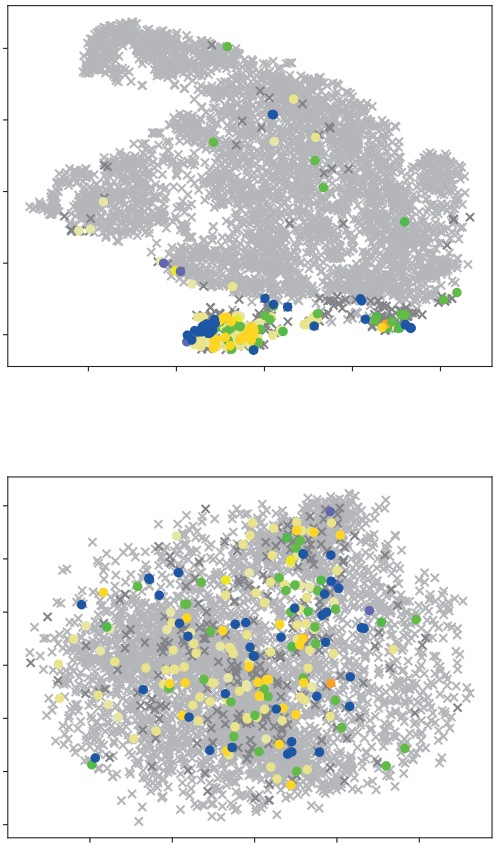


Figure 2: Plots of the T-SNE projected chiasmus datapoints. The upper plot shows the DLE features, the lower plot shows the features from the last hidden state of the DistilBERT model. Non-chiasmi are drawn as crosses, dark for annotated ones and bright for the random inversions. Chiastic samples and parallelisms are repesented as filled circles. The green colored dots are chiasmi, blue colored dots are antimetaboles and yellow colored dots are parallelisms. This figure is best viewed in color.

Synthetic parallelism (sp) stands for a form of parallelism, where the stylistic device gets its meaning by a related series of statements together. An example would be: *do the research, write the paper, submit the work*.

Antithetic parallelism (ap) describes a form of parallelism with a chiastic semantic meaning, representing a form of opposite.

Near misses (f.∗) are examples that contain a chiasmus or an antimetabole in their scope, but the supporting tokens detected by the algorithm are not the real supporting tokens of the chiasmus. An example would be: *narrow **is the** world, and **the** brain **is** wide*, which is a chiasmus with the real supporting tokens *narrow, world – brain, wide*, but the detected supporting tokens *is, the – the, is*.

Negative Examples (x) are all other part-of-speech tag inversions without special semantic meaning.

Randomly Chosen Negative Examples (xr) are the same as the other negative examples, but were randomly chosen and automatically annotated without human supervision.

For a more detailed descriptions of the above mentioned stylistic devices, we refer to works by (Braungart et al., 2010; Burdorf et al., 2007) and Ueding (1998).

3.3. Dataset Visualization

To give a visual overview of the dataset, we show in Figure 2 a scatterplot of a T-SNE (van der Maaten and Hinton, 2008) projection of the data. The features that were used for this projection were the DLE features by Schneider et al. (2021), which will be explained in Section 4.. The labels for our proposed dataset were created by annotating the top results from the application of the chiasmus detection algorithm by Schneider et al. (2021). It can be seen that many of the annotated negative examples lie close to the annotated positive examples, while the rest of the random inversions can easily be separated. This introduces examples which cannot be easily separated by the existing methods and need further research. On the other hand, it can be seen that the simple DistilBERT (Sanh et al., 2019) approach does not result in features that are easily pre-separated, even though it is already fine-tuned on the dataset.

3.4. Creation

The dataset was created on the basis of the experiments of Schneider et al. (2021). In their work, the authors searched for inverted repetitions of part-of-speech tags in texts from GerDraCor and then ranked the obtained results by different ranking methods. The top 100 results were then annotated by a domain expert. The dataset comprises these annotated results as well as 4000 randomly sampled inversions from the corpus, annotated as negative examples. Since the chiasmus is such a scarce phenomenon, this can be safely done. However, every example has an annotation indicating whether it is annotated by a domain expert or sampled randomly as a negative example.

3.5. Source and Dataset Format

The underlying data is drawn from the GerDraCor corpus, which comprises plays in the German language published between 1650 and 1940. As a result, some words might be spelled differently in the texts than they would be in modern German. Also, when taking into account the long timespan, some words may have gone through semantic changes (Schlechtweg et al., 2017; Koch, 2016).
The search window for the part-of-speech tag inversions for creating the dataset had a size of 30. That is, the distance from the first to the last supporting token spans at most 30 tokens. The phrases contained in the dataset have additionally 5 tokens before the first and after the last supporting token as context. The mean of the phrase length from the first to the last supporting token is 22 tokens, with a standard deviation of 6 tokens.

POS Tag	Percentage	Parallelisms
NOUN	45.7%	92.1%
PRON	34.3%	03.9%
VERB	13.6%	01.3%
PROPN	04.3%	01.3%
DET	01.4%	01.3%
ADJ	00.7%	00.0%

Table 2: Percentage of the different part-of-speech tags in the positive examples.

The data format is a JSON file. Every entry consists of seven different fields. The field *ids* contains the offset of the four supporting tokens in the source files, *cont_ids* describes the offset of the whole phrase. The different tokens are recorded in *tokens*, the lemmas in *lemmas* and the part-of-speech tags in *pos*. The results of the dependency parsing can be found in *dep*. Finally, the annotations are contained in *annotation*.

For tokenizing, lemmatizing, part-of-speech tagging, and for the dependency trees we used the spaCy library (Honnibal et al., 2020). However, the word embeddings which we used in the experiments were not created with spaCy, since the models included there only create embeddings for words contained in their dictionary. Instead, we used the German FastText (Bojanowski et al., 2017) model available on the FastText website (Grave et al., 2018).

3.6. Special Cases and Biases

One source of bias is that all positive examples were found by using a single algorithm with various sets of features, all trained on the same dataset. While every positive example was annotated manually, this means that positive examples which were not fitting the criteria in the ranking algorithm may be left out of the datset. The whole number of chiasmi in the GerDraCor corpus is not known. Thus, no quantitative statement can be made about this potential limitation.

Table 2 shows the distribution of different part-of-speech tags in the positive examples. We used positive annotations for chiasmi and antimetaboles, as well as the parallelisms. It can be seen that most chiasmi are based on repeated nouns, followed by pronouns. Determiners and especially adjectives make up the least part of the dataset. This may be a source of bias, since some better known chiasmus examples like ***narrow** is the **world** and the **brain** is **wide*** are also based on adjectives.

Another potential bias source is the annotation. Since the annotations were done by a single domain expert, metrics like inter-annotator agreement can not be reported.

3.7. Examples

In the following we show some examples of chiasmi and antimetaboles in the dataset. The examples are first given in German, followed by their English translation. The last line contains the class of the example. The examples were chosen since they carry obviously semantically related meaning between their main words and are thus very prototypical and unambiguous examples. The last three examples got very low scores with all feature combinations.

- O **Augen** ohne **Kopf**, o **Kopf** ohne **Augen**
  O **eyes** without **head**, o **head** without **eyes**
  Antimetabole
- **Dir** widert **Landluft**, **Seeluft** widert **dir**.
  You dislike **country air**, the **sea air** disgusts **you**.
  Chiasmus
- . . . der Menschen **belustigen mich** lange, eh sie **mich reizen**.
  . . . of the humans **amuse me** for long before they **irritate me**.
  Chiasmus
- **Ich** bin nicht, was ich **scheine**, und **scheine** auch nicht, was ich bin, Und wenn ich das wäre, was **ich** sein möchte
  I am not what I **seem** and do not **seem** what I am, and if I would be that, what **I** want to be
  Chiasmus
- Ja **ich** hab einen **Sohn** gequält, und ein **Sohn** mußte **mich** wieder quälen
  Yes **I** have tortured a **son**, and a **son** had to torture **me** again
  Chiasmus
- Meinst **du** damit etwa **mich**? Mein **ich** damit etwa **dich**?
  Do **you** happen to mean **me**? Do **I** happen to mean **you**?
  Antimetabole

4. Baseline Benchmark

For our baseline benchmark experiments, we use the approach by Schneider et al. (2021). Following their evaluation, we compare different feature sets presented in this work, including the features proposed by Dubremetz and Nivre (2018).

4.1. Methods

Chiasmus candidates are ranked by their chiasticity using a support vector machine (SVM) with an RBF kernel (Schölkopf and Smola, 2001). The regularization parameter for the SVM model is 1 with a maximum of 1000 iterations for the fitting of the model. The features are preprocessed to a mean of 0 and scaled to unit variance.

In the following we summarize the features. For a full explanation, we refer to the respective works:

Dubremetz features (D) include various basic features that are already useful for the detection of antimetaboles. These features include the usage of certain words like negations, the usage of punctuation, the repetition of words in certain parts of the example as well as the repetition of syntax tree tags. For a complete summary, please see the work of Dubremetz and Nivre (2018).

Lexical features (L) are binary features that indicate in a pairwise manner whether the main words constituting the chiasmus comprise repeating lemmata. They were proposed by Schneider et al. (2021).

	D	DL	DE	DLE	DistilBERT
full	0.49 ± 0.26	0.65 ± 0.32	0.73 ± 0.35	0.72 ± 0.35	0.10 ± 0.07
fp removed	0.61 ± 0.30	0.69 ± 0.33	0.74 ± 0.35	0.74 ± 0.35	0.07 ± 0.08

Table 3: Baseline results for the chiasmus detection. The values represent the mean average precision and their standard deviation.

Embedding features (E) describe the pairwise cosine distance of the word embeddings of the main words constituting the chiasmus and thereby indicate their relation. They were proposed by Schneider et al. (2021).

In addition, we also conducted experiments using a BERT-like language model (Devlin et al., 2019). We used *distilbert-base-german-cased*, a DistilBERT (Sanh et al., 2019) model trained on German texts, which is available for download on the huggingface website [2] with the pytorch implementation of the language model. We presented the data as one text string per example, as the tokens appear in the dataset, with the tokens separated by single spaces. The finetuning was conducted for 5 epochs with a batch size of 128, using the AdamW optimizer with a lerning rate of 0.001. We trained the model for 20 epochs with a weight decay of 0.01.

4.2. Results

Table 3 shows the results of the chiasmus detection. The experiments were conducted by using 5-fold cross-validation with 80% of the data used for training and 20% used for testing. For evaluation, we report the *average precision* metric. This information retrieval criterion describes the area under the precision-recall curve. This metric was chosen because the main interest of chiasmus detection is to extract chiasmi from a longer corpus instead of just classifying instances. The two experiments presented, *full* and *fp removed* show a different choice of samples. In *full*, all positive samples were annotated with *a, c, fa*, and *fc*. In *fp removed*, the positive examples were *a* and *c*, while *fa* and *fc* were removed from the dataset. It can be seen from the results that both sets of choices result in similar differences between average precision values. *D*, *DL*, *DE*, and *DLE* stand for different combinations of the features explained above. Please see Sec. 3. for an explanation of the acronyms. Additionally, the results of the DistilBERT experiment are included.

The combination of all features yields either the best or the second best results on our dataset. For the *full* experiment, the results deviate slightly. However, the standard deviation of the results is also very high, which implies that the small improvement of the *DL* combination over the *DLE* combination may be attributed to random noise. Since the repeated lemmas should also have very similar word embeddings, even if the tokens themselves differ, a lot of the information from the lemma repetition features is already included in the word embedding features. In comparison, the DistilBERT experiment performs worse in our case. As BERT-like models have shown superior performance in many NLP applications, this unexpected result needs further investigation.

To find out whether the improvements of the feature combinations compared to the baseline *D* were statistically significant, we ran a 5x2cv (Dietterich, 1998) evaluation on the full dataset. *DLE* showed an improvement in average precision from 0.2 to 0.51 with a p value of 0.06 compared to only using *D* features. *DL* increased average precision to 0.41 with a p value of 0.18, while *DE* also yielded 0.51 (like *DLE*), but with a higher p value of 0.09 instead of 0.06. While the similar average precision of the *DL* and the *DLE* combinations makes the lexical features look less useful, the lower p value of the *DLE* combination indicates with its lower *probability of the improvement being random* the importance of these features.

5. Conclusions

In this paper, we presented a dataset containing instances of chiasmi and antimetaboles, as well as parallelisms and part-of-speech tag inversions without special semantic meaning. The data is annotated in a way that opens up different possibilities on how to use it in the future, including which subset to use and what parts to exclude. It is difficult to define how exactly the supporting tokens of a chiasmus candidate need to be related to constitute a real chiasmus. Thus, chiasmus detection is a hard problem that needs further research.

At the same time, it is suitable to evaluate different kinds of language models on it. Our baseline experiments show that the task in principle is solvable. However, there is still much room for improvement with respect to performance. We hope that this dataset will encourage new research in that field and will be used to improve both the detection of chiasmi and the understanding of language models in general.

References

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov, 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Brandes, Phillip, Felix Schneider, and Sophie Marshall, 2022. Stilfiguren aus der Distanz gelesen. Zur automatischen Detektion von Wortstellungsfiguren und deren Nutzen für die qualitative Analyse.

Braungart, Georg, Harald Fricke, Klaus Grubmüller, Jan-Dirk Müller, Friedrich Vollhardt, and Klaus Weimar (eds.), 2010. *Reallexikon der deutschen Literaturwissenschaft*. Berlin, Boston: De Gruyter.

[2] https://huggingface.co/distilbert-base-german-cased

Burdorf, Dieter, Christoph Fasbender, and Burkhard Moennighoff, 2007. *Metzler Lexikon Literatur*.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics.

Dietterich, Thomas G., 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923.

Dubremetz, Marie and Joakim Nivre, 2015. Rhetorical figure detection: the case of chiasmus. In *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*. Denver, Colorado, USA: Association for Computational Linguistics.

Dubremetz, Marie and Joakim Nivre, 2016. Syntax Matters for Rhetorical Structure: The Case of Chiasmus. In *Proceedings of the Fifth Workshop on Computational Linguistics for Literature*. San Diego, California,USA: Association for Computational Linguistics.

Dubremetz, Marie and Joakim Nivre, 2018. Rhetorical figure detection: Chiasmus, epanaphora, epiphora. *Frontiers in Digital Humanities*, 5:10.

Fischer, Frank, 2019. Programmable corpora: Introducing dracor, an infrastructure for the research on european drama.

Gawryjolek, Jakub Jan, 2009. Automated annotation and visualization of rhetorical figures.

Grave, Edouard, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov, 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Honnibal, Matthew, Ines Montani, Sofie Van Landeghem, and Adriane Boyd, 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Java, James, 2015. *Characterization of Prose by Rhetorical Structure for Machine Learning Classification*. Ph.D. thesis, Nova Southeastern University.

Koch, Peter, 2016. *Meaning change and semantic shifts*. Berlin, Boston: De Gruyter Mouton, pages 21–66.

Lim, SeungJin, 2016. An algorithm for detection of chiastic structures in text databases. In *2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES)*.

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf, 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Schlechtweg, Dominik, Stefanie Eckmann, Enrico Santus, Sabine Schulte im Walde, and Daniel Hole, 2017. German in flux: Detecting metaphoric change via word entropy. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Vancouver, Canada: Association for Computational Linguistics.

Schneider, Felix, Björn Barz, Phillip Brandes, Sophie Marshall, and Joachim Denzler, 2021. Data-driven detection of general chiasmi using lexical and semantic features. In *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*. Punta Cana, Dominican Republic (online): Association for Computational Linguistics.

Schölkopf, Bernhard and Alexander J. Smola, 2001. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press.

Ueding, Gert, 1998. *Historisches Wörterbuch der Rhetorik*. Tübingen: Niemeyer.

van der Maaten, Laurens and Geoffrey E. Hinton, 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Welch, John W, 2020. *Chiasmus in antiquity*. Wipf and Stock Publishers.

Artificial Neural Networks based Baby Sign Language Recognition via Wearable Sensors

Emre Sevindik[1], Elif Ergin[2], Kübra Erat[3], Pınar Onay Durdu[4]

[1, 2, 3, 4] Department of Computer Engineering, Faculty of Engineering, Kocaeli University
{emre.sevindik99, elifergin2534}@gmail.com
{kubra.erat, pinar.onaydurdu}@kocaeli.edu.tr

Abstract

Baby sign language (BSL), is a non-verbal language, which is used to communicate between parents and their toddlers. Unlike the sign language used by hearing and speech-impaired individuals, BSL is used by hearing toddlers to communicate their emotions and desires before their verbally speaking period. In this study, a deep learning-based baby sign language recognition research has been conducted using a single Myo armband, which is a wearable technology with surface Electromyography (EMG) sensors and an Inertial Measurement Unit (IMU) sensor. The dataset containing signals from the sensors consisting of six words representing the basic needs of babies are classified using Artificial Neural Networks (ANN). The results show that the proposed method has the highest accuracy, which is 98.23%, when EMG and IMU sensors are used together. In addition, it is observed that only the IMU sensor data can also be sufficient for the selected BSL words' recognition.

Keywords: Baby Sign Language Recognition, Artificial Neural Network (ANN), Electromyography (EMG), Inertial Measurement Unit (IMU).

1. Introduction

Sign language is a non-verbal communication method for people who have speech and hearing impairments. It comprises hand gestures and body movements as a means for communication rather than the use of words. On the other hand, the use of sign language can facilitate communication among the parents and their toddlers before their verbal speaking period. Alternatively, this specific sign language is called baby sign language (BSL) (Baby Sign Language Inc., 2023). Different from regular sign languages, hearing infants and toddlers use BSL to communicate their desires and emotions to their parents quickly and easily. In addition, BSL has the potential to enhance children's mental development as well as improve their relationship with their parents (Nadgeri and Kumar, 2022).

Although sign languages help hearing-impaired people to communicate with each other, their communication with the rest of society is limited because hearing people are not trained in these languages. Software technologies including artificial intelligence and machine learning can be used to facilitate the integration of people who communicate with sign language into daily life (Papastratis et al., 2021). There are various studies on for automatic recognition of sign language for this purpose (Shakeel et al., 2020; Zia ur Rehman et al., 2018). However, implemented techniques are still limited (Tryon and Trejos, 2021).

The methods used for automatic sign language recognition (SLR) are classified as image-based and sensor-based methods (Gu et al., 2022). Vision-based methods generally use cameras and motion sensors to capture information in a 2-dimensional space. In some recent studies, Kinect (Nuzzi, et al., 2018) or Leap Motion (Yang et al., 2020) sensors are also used. However, in all of these vision-based technologies, there are various limitations due to occlusion which occurs when a foreign object blocks the camera vision or poor lighting conditions that cause inaccurate recognition results (Jane and Sasidgar, 2018). On the other hand, sensor-based methods typically use gloves with Inertial Measurement Units (IMU) (Kim et al, 2019) or non-invasive surface Electromyography (EMG or sEMG) (Vasconez et al., 2022) methods for the detection of gestural activities. EMG signals detect the muscle movements of the participants and allow to act on them. One of the technologies with wearable EMG sensors developed for this purpose is the Myo armband (Thalmic Labs, Canada). This device is also combined with IMU sensors, which makes the Myo armband superior to glove-based methods since they are less sensitive and uncomfortable to wear (Jane and Sasidhar, 2018). Signals detected through the Myo armband can be processed with machine learning methods and produce meaningful outputs.

There are various studies conducted by Myo armband for automatic SLR. It is seen that deep learning algorithms are commonly used in many of these studies (Briouza et al., 2021; Nadgeri and Kumar, 2022; Ozdemir et al., 2020; Shin et al., 2017; Vásconez et al., 2022; Yang and Zhang, 2019). However, Artificial Neural Networks (ANN) gain great importance in problems that do not have an algorithmic solution (Fatmi et al., 2019). With an ANN model, hidden patterns in the data can be detected and complex data can be processed by detecting the linear-nonlinear relationship between dependent-independent variables (Bangaru et al., 2021). Therefore, ANNs are widely used especially in sign language or gesture recognition studies (Fatmi et al. 2019; Jane and Sasidhar, 2018; Vásconez et al. 2022; Yang and Zhang, 2019).

In the scope of this study, a single Myo armband is used to record selected BSL words. EMG and IMU signals gathered are classified by an ANN model to achieve higher accuracy. In the model, data preprocessing, feature extraction, and classification steps are implemented. For preprocessing, bandpass filtering is applied. For feature extraction, five features which are Root Mean Square (RMS), Mean Absolute Value (MAV), Variance (VAR), Wavelet Length (WL), and Standard Deviation (SD) are extracted. For classification, a method based on ANN is implemented.

The rest of the paper is organized as follows. Section 2 summarizes a short literature review of the similar studies carried out. Section 3 describes the details of the implemented ANN based method. Section 4 gives the experimental results. Section 5 presents the conclusion, limitations and future work. Finally, there is also an acknowledgement section.

2. Related Works

There are various previous studies in the literature that deals with automatic sign language recognition (SLR) as well as automatic hand gesture recognition (HGR) which can also be considered as having a similar scope. However, this section presents only some recent examples which are considered related to the scope of this current work. These are grouped into two groups as SLR and HGR. Many of these studies use Myo armband or a similar sensor device to gather data. The common details of the studies are summarized in Table 1.

One of the recent prominent papers which directly relate to BSL recognition (BSLR) is conducted by Nadgeri and Kumar (2022). However, although this study exemplifies the automatic recognition of BSL as in this current work, it implements a vision based method because the data is gathered as BSL images. The study proposes a recognition model based on convolutional neural network (CNN) and long short-term memory (LSTM) network. The method is evaluated on the dataset consisted of 34 daily baby sign words created by the researchers. The study reports that Adam and SGD optimizers produce the accuracy results of 75% and 99% respectively based on that CNN-LSTM model.

On the other hand, in the scope of one of the sensor-based studies (Fatmi et al., 2019), the performance of various machine learning methods for American Sign Language (ASL) is evaluated and a system that produces voice and text outputs is developed. The researchers create their own data set by using Myo wearable armband and they collect data for selected 13 ASL gestures. EMG and IMU signals collected by Myo armband are processed by ANN, Support Vector Machines (SVM), and Hidden Markov Model (HMM) classifiers comparatively and ANN is reported to outperform the other two with 93.79 % accuracy.

Similar to Fatmi et al.'s (2019) study, Jane and Sasidhar (2018) develops a sign language interpreter (SLI) system to ease the life of hearing-impaired people. 48 words are selected from Signing Exact English (SEE-II) lexicon for the study. EMG and IMU data of these words are filtered using wavelet denoising techniques and they are segmented using Teager Kaiser Energy Operator (TKEO) thresholds. A total of 10 features including various time and frequency domain features are used for ANN classifier and it achieves a recognition rate of 97.12% accuracy.

In another study (Shin et al., 2017), sensor fusion technology and group-dependent Neural Network based model is implemented for the recognition of the selected words from the Korean Sign Language (KSL). In the study sign gestures are grouped into two as static and dynamic. EMG and IMU raw sensor data gathered by Myo armband are normalized by both z-score and min-max normalization and then zero-padding is used. As a result of

SLR Papers	Goal	Gestures	Sensor	Data	Method	Results
(Nadgeri and Kumar, 2022)	BSLR	34 words from BSL	Web camera	Image	A Model based on CNN and LSTM	Adam: 99% SGD: 75%
(Fatmi et al., 2019)	SLR	13 ASL gestures	Myo armband	EMG, IMU	ANN, SVM, and HMM	ANN: 93.79%, SVM: 85.56%, HMM: 85.90%
(Jane and Sasidhar, 2018)	SLI	48 SEE-II words	Myo armband	EMG, IMU	ANN	93.27%
(Shin et al., 2017)	SLR	30 words from KSL	Myo armband	EMG, IMU	CNN and LSTM	99.13% (without dropout) 98.1% (with dropout)
HGR Papers	**Goal**	**Gestures**	**Sensor**	**Data**	**Method**	**Results**
(Vásconez et al., 2022)	HGR	Static and dynamic 12 gestures	Myo armband, G-force	EMG, IMU	Deep Q-Network based on ANN	$97.50\% \pm 1.13\%$ (Static, C) $88.15\% \pm 2.84\%$ (Static, R) $98.95\% \pm 0.62\%$ (Dynamic, C) $90.47\% \pm 4.57\%$ (Dynamic, R)
(Briouza et al., 2021)	HGR	Ninapro's DB 2	-	EMG	CNN	Validation and training Acc: 78:84%, 89:79% (Ex B) 72:06%, 90:83% (Ex C) 88:07%, 94.04% (Ex D)
(Ozdemir et al., 2020)	HGR	7 gestures	BIOPAC MP36	sEMG	CNN based on ResNet	Acc: 99.59% F1-Score: 99.57%
(Yang and Zhang, 2019)	HGR	5 gestures	Myo armband	sEMG	ANN	96%

Table 1: Related works.

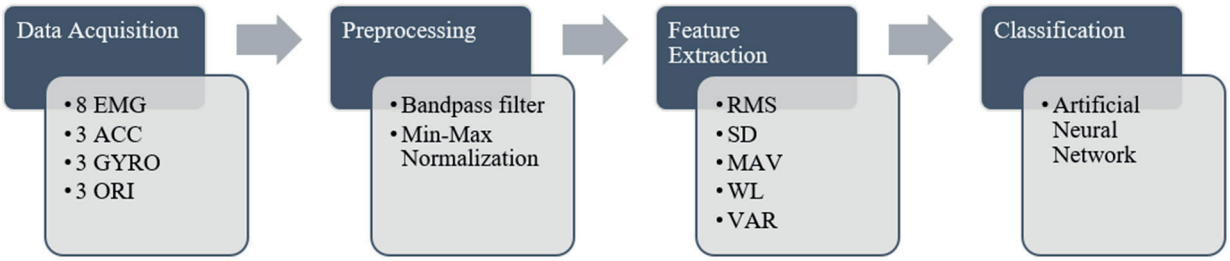


Fig. 1: The Flowchart of the methodology.

the study, recognition models based on each of CNN and LSTM are implemented and 99.13% accuracy of CNN without dropout and 98.1% of CNN with dropout are reported to be achieved.

Alternatively to SLR studies, there are various gesture recognition studies in which neural networks are implemented. One of the recent study (Vásconez et al., 2022) performs HGR by a Deep Q-Network, which is a reinforcement learning model, using EMG and IMU signals. The raw sensor data are collected by both Myo armband and G-force sensors. A feature vector composed of RMS, SD, Energy, MAV, and Absolute Envelope based on 11 static and dynamic gestures are used. Models created using both reinforcement and supervised learning algorithms are tested with 42 participants, and it is observed that the best classification and recognition results are obtained from Myo armband sensors with respect to G-force sensors. Classification and recognition performance are reported as 97.50%±1.13% and 88.15±2.84%, in static movements and 98.95%±0.62% and 90.47±4.57% in dynamic movements.

Briouza et al. (2021) proposes a new deep learning approach for classification of EMG signals without the need for feature selection. In this approach 8 layer architecture consists of 3 convolutional layer, 3 max pooling layer, a fully connected and a softmax layer is designed. Proposed model is tested on Ninapro Database 2 which includes raw EMG signals of 49 gestures gathered from 40 participants and the results has been reported regarding validation and training accuracy for Exercises B (78.84 %, 89.79 %), C (72.06 %, 90.83 %) and D (88.07, 94.04 %).

Similarly, Ozdemir et al. (2020) proposes a deep learning based model to improve the accuracy in hand gesture recognition. In the study, 4-channel surface EMG data is gathered from 30 participants while simulating 7 hand gestures. The collected sEMG data is segmented into sections with hand gestures and then spectrogram images created via STFT are used to train ResNet-based CNNarchitecture. The proposed CNN based model provides higher accuracy by achieving 99.59% test accuracy and 99.57% F1-Score.

In another EMG based gesture recognition study (Yang and Zhang, 2019), signals belonging to 5 different gestures pre-defined in the Myo armband are filtered with the fourth-order digital low-pass Butterworth filter, and then certain segments are taken with the Sliding Window technique. A total of 5 features, namely RMS, MAV, Slope Sign Changes, WL, and Hjorth parameters, are extracted from each segment and they are used as input for a feed-forward ANN model. It has been reported that 96.7% of the movements are recognized correctly before they are completed when they reach the threshold determined.

3. Methodology

In this study, an ANN-based method is implemented for the recognition of BSL. The steps followed for the study are presented in Fig. 1. As a first step in the study, a dataset is created by the use of Myo armband, since there is not any standard accessible dataset which complies with the purpose of the study (Nadgeri and Kumar, 2022). Afterward preprocessing, feature extraction, and classification operations are implemented to achieve a reasonable recognition accuracy.

3.1 Dataset Acquisition

The words for the dataset are selected from the gestures of BSL which is specifically designed for the communication of hearing toddlers with their parents before their speaking period. The selected words and their labels can be seen in Fig. 2 and they are as follows; Milk (0), Water (1), Sleep (2), Hungry (3), Eat (4), Drink (5). These words are selected according to whether these can be done by one hand only. EMG signals of selected words are collected from only one participant using the Myo armband. Data regarding each word is collected by repeating each gesture 10 times (Gharibo, 2021). Before each repetition, the participant is given a 30-second rest period and then he performs the following word's gesture. Both EMG and IMU sensors data are recorded.

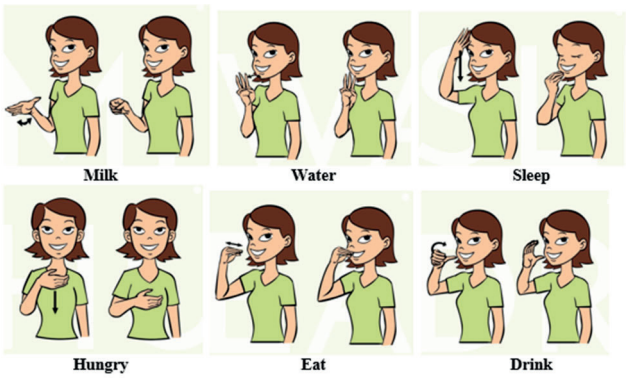


Fig. 2: Selected baby sign language words (Baby Sign Language Inc.)

3.1.1. Myo Armband

The Myo armband is a wearable device with 8 EMG sensors manufactured by Thalmic Labs. In addition, the armband consists of a 9-axis IMU which is used to detect arm movements. The IMU includes a 3-axis gyroscope, 3-

axis accelerometer and a 3-axis magnetometer. Myo SDK is required to develop software with Myo armband. In addition, the sample rate of EMG data recorded through the Myo armband is 200 Hz while IMU data is 50 Hz. Myo armband can be seen in Fig. 3.

Fig. 3: Myo armband (Thalmic Labs, Canada)

3.2. Preprocessing

In the data preprocessing step, a bandpass filter is used to filter the EMG signals. While EMG signals are distributed in the range of 0-500 Hz, they are dominant between 50-500 Hz (Gharibo, 2021). A bandpass filter is preferred when filtering EMG signals, since movement at low frequencies and noise at high frequencies may interfere (Robertson et al., 2014). Therefore, the fourth order bandpass Butterworth filter (Yang and Zhang, 2019) is used in this study. Since the signal to noise ratio of the IMU signals is at an acceptable level, filtering is not applied (Gharibo, 2021) to them.

IMU and filtered EMG signals are normalized between [0-1] with min-max normalization technique. The data size for 8 EMG sensors and 9 IMU sensors is as follows: [20673x17]. Then, a certain segment of the signals is taken and features are extracted from each segment by the sliding windows method. The window length of 300 and a step of 40 (Vásconez et al., 2022) is chosen. Image of unfiltered-filtered and filtered EMG data is given in Fig. 4.

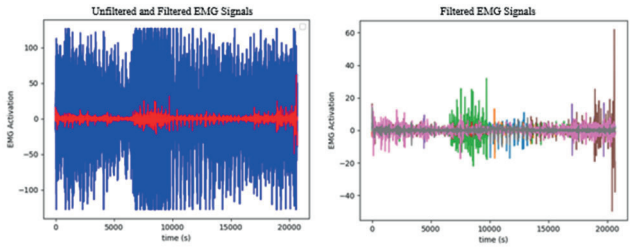


Fig. 4: (1) Unfiltered and Filtered signals, (2) Filtered signals

3.3. Feature Extraction

In the study, five different features that can be seen in Table 2 are extracted from each EMG and IMU signal segment obtained with the sliding window technique. These features are as follows; Root Mean Square (RMS), Mean Absolute Value (MAV), Variance (VAR), Wavelet Length (WL), and Standard Deviation (SD). And the parameters in the formulas of the features are as like: The value N is the size of the channel, X symbolizes channel information, and X_i is a data point in the channel (Yang and Zhang, 2019). Moreover, the data size and label matrix obtained in the last step are as follows respectively: [516x85], [516x1]. Here, [516x40] is data from EMG sensors, while [516x45] is data from IMU sensors.

3.4. Artificial Neural Network

ANN is a method that is developed inspired by models of sensory processing performed by the brain. In this way, by establishing an artificial network that imitates real neurons in the computer environment, it has become possible to teach these models to solve different problems from the domains of science, medicine, and engineering (Krogh, 2008).

Feature	Formula
RMS	$\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$
MAV	$\frac{1}{N}\sum_{i=1}^{N} \lvert x_i \rvert$
VAR	$\frac{1}{N-1}\sum_{i=1}^{N} (x_i - \bar{x})^2$
WL	$\sum_{i=1}^{N} \lvert x_i - x_{i-1} \rvert$
SD	$\sqrt{\frac{1}{N-1}\sum_{i=1}^{N} (x_i - \bar{x})^2}$

Table 2: Formulas of the features.

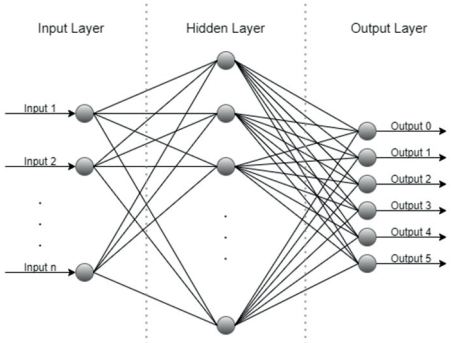


Fig. 5: Artificial neural network architecture

Multilayer Perceptron Architecture is used in this study. In this architecture, each Neural Network basically includes 3 layers, namely the input layer, hidden layer, and output layer (Gupta, 2013). The number of hidden layers can be zero or more, and the optimum number of perceptron and hidden layers can be found by trial and error (Fatmi et al., 2019). As seen in Fig. 5, a fully connected and three-layer feedforward perceptron neural network is designed in this study.

4. Results and Discussion

In this study, the performance of the proposed ANN model is tested according to accuracy, precision, recall and F1-Score values by comparing three different scenarios. These includes both EMG+IMU sensors data, only EMG sensors data, and only IMU sensors data respectively. The ANN

Hyperparameters	EMG + IMU	EMG	IMU
# of neuros in the Input Layer (n)	85	40	45
# of neuros in the Hidden Layer	45	23	25
Epoch	50		
Batch Size	32		
Activation Function	ReLu		
Output Layer	Sotfmax, # of neuros: 6 (0-5)		

Table 3: Proposed ANN architecture hyperparameters

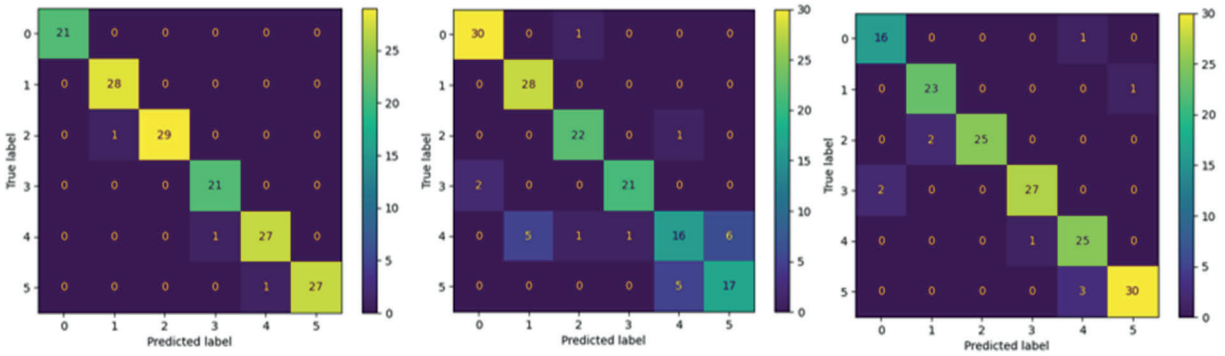


Fig. 6: Confusion matrices for EMG+IMU, EMG, and IMU, respectively.

architecture of the proposed method is given in Table 3 separately for each defined scenario. The ANN architecture created to classify the data received from sensors consists of an input layer, a hidden layer, and an output layer. Regarding the proposed ANN architecture that uses EMG+IMU sensors, there are 85 neurons in the input layer. This corresponds to 5 different features (attributes) extracted from the data received from 85 neurons, 8 EMG and 9 IMU sensors. The number of perceptron in the hidden layer is calculated using the Equation (1) (Fatmi et al., 2019) and is found to be 45. In the output layer, there are also 6 neurons representing the classes, which are Milk (0), Water (1), Sleep (2), Hungry (3), Eat (4), and Drink (5). Then again, the ANN architecture that uses only EMG sensors has 40 neurons in the input layer. These 40 neurons correspond to 5 different attributes extracted from the data received from 8 EMG sensors. The number of neurons in the hidden layer is calculated as 23. Finally, there are 45 neurons in the input layer of the proposed ANN architecture that uses only IMU sensors. These correspond to 9 IMU sensors and 5 different attributes, too. The number of neurons in the hidden layer is also calculated as 25 for this case.

$$\textit{Layer size} = (\#\ \textit{of classes} + \#\ \textit{of features})/2 \quad \text{Equation (1)}$$

The dataset used in the study is divided into a training dataset (70%) and a testing dataset (30%). Table 4 indicates accuracy, precision, recall, and F1-Score values obtained for three scenarios determined. Each value is tested 50 times and their average is taken. As can be seen from the Table 4, when both EMG and IMU sensors are used together, accuracy, precision, recall, and F1-Score values are 98.23%, 98.44%, 98.37%, and 98.36%, respectively. Similarly, these values are observed to obtain 88.80%, 88.62%, 88.64%, and 88.15% for only EMG sensors, while 96.05%, 96.14%, 96.00%, and 95.97% are obtained for only IMU sensors.

A graph of the accuracy obtained in each scenario is shown in Fig. 7. According to this graph, accuracy is 97.75%±2.24% when the data of the EMG+IMU sensors are used, 90.38%±7.69% when only the data of the EMG sensors are used, and finally, 95.18%±3.52% when only the data of the IMU sensors are used. In addition, confusion matrices of these three defined scenarios are also given in Fig. 6. The total recognition rate is found to be 98.07% for the EMG+IMU, %85.89 for only EMG, and 93.58%, for only IMU. While almost all of the words are recognized according to EMG+IMU and only IMU sensor data, it is seen that some words are rarely misclassified when recognized only with EMG data. That is the case when the word Eat (4) is classified as the word Drink (5) and vice versa. In addition the word Eat (4) is rarely classified as the word Water (1) as can be seen from the same confusion matrix. This is due to the gestural similarity of the representation of these words, as can be seen clearly in Fig. 2.

	EMG+IMU	EMG	IMU
Accuracy	98.23%	88.80%	96.05%
Precision	98.44%	88.62%	96.14%
Recall	98.37%	88.64%	96.00%
F1-Score	98.36%	88.15%	95.97%

Table 4: ANN results for Accuracy, Precision, Recall, and F1-Score.

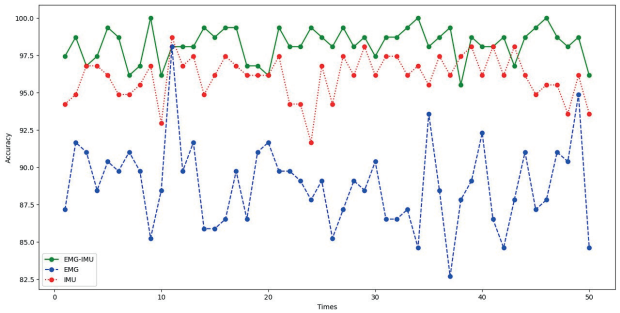


Fig. 7: Comparison of the accuracy by sensors

5. Conclusion

In this study, an ANN based method is proposed for BSLR using EMG and IMU sensors of the MYO armband. BSL words that can be done by one arm are selected for the dataset created. The data obtained are filtered using the 4th order bandpass Butterworth filter in the preprocessing stage. With the sliding windows technique, 5 features, namely RMS, MAV, VAR, WL, and SD, are extracted from the filtered data. Accuracy, F1-Score, precision, and recall values of classifications performed according to three different scenarios using ANN are calculated. The overall accuracy results for the proposed method based on, EMG+IMU, EMG, and IMU sensors data are found to be as 98.23%, 88.80%, and 96.05%, respectively. Similar to other studies (Bangaru et al., 2021; Vásconez et al., 2022), higher accuracy is observed when both EMG+IMU sensors were used together than sign language and gesture recognition studies using only EMG or only IMU sensor.

When the findings are analyzed, it is observed that using only IMU sensor data may also be sufficient for BSLR.

There are several limitations in the study. For instance, there is only one Myo armband so BLS words that can only be performed with one hand are included for the data set. In addition, the data is collected form only one participant. Therefore, as a future work it is aimed to carry out the experiment with more participants and with two devices to increase the number of data in the dataset. In addition, it is aimed to test with different classification algorithms by using different features, too.

6. Acknowledgment

We would like to thank Kocaeli University Scientific Research Projects Coordination Unit since this project is supported within the scope of the project numbered FMP-2021-2750. In addition, we would like to express our gratitude to Dr. Orhan Akbulut for his invaluable input and support throughout the research process. His insights and expertise were instrumental in shaping the direction of this project.

References

Baby Sign Language Inc. *Baby Sign Language Dictionary*. Retrieved from: (https://babysignlanguage.com/). Access Date: 28 January 2023.

Bangaru, S.S., Wang, C., Busam, S.A. and Aghazadeh, F. (2021). "ANN-Based Automated Scaffold Builder Activity Recognition through Wearable EMG and IMU Sensors". Automation in Construction 126:103653. doi:10.1016/j.autcon.2021.103653.

Briouza, S., Gritli, H., Khraief, N., Belghith, S. and Singh, D. (2021). "A Convolutional Neural Network-Based Architecture for EMG Signal Classification". In 2021 International Conference on Data Analytics for Business and Industry (ICDABI). Sakheer, Bahrain, pp. 107-112.

Fatmi, R., Rashad, S. and Integlia, R. (2019). "Comparing ANN, SVM, and HMM Based Machine Learning Methods for American Sign Language Recognition Using Wearable Motion Sensors". In 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC). Las Vegas, NV, USA, pp. 0290-97.

Gharibo, Jason S. (2021). "Data and Sensor Fusion Using FMG, SEMG and IMU Sensors for Upper Limb Prosthesis Control". Doctoral dissertation, The University of Westem Ontario (Canada).

Gu, Y., Sherrine, S., Weiyi, W., Xinya, Li., Jianan, Y., and Masahiro, T. (2022). "American Sign Language Alphabet Recognition Using Inertial Motion Capture System with Deep Learning". Inventions 7(4):112. doi: 10.3390/inventions7040112.

Gupta, N. (2013). "Artificial Neural Network". Network and Complex Systems", 3(1), pp. 24-28.

Jane, S.P.Y. and Sasidhar, S. (2018). "Sign Language Interpreter: Classification of Forearm EMG and IMU Signals for Signing Exact English". In 2018 IEEE 14th International Conference on Control and Automation (ICCA). Anchorage, AK, USA, pp. 947-952.

Krogh, A. (2008). "What Are Artificial Neural Networks?" Nature Biotechnology 26(2):195-97. doi: 10.1038/nbt1386.

Nadgeri, S., and Kumar, A. (2022). "Deep Learning Based Framework For Dynamic Baby Sign Language Recognition System". Indian Journal of Computer Science and Engineering 13(2):550-63. doi: 10.21817/indjcse/2022/v13i2/221302127.

Nuzzi, C., Pasinetti, S., Lancini, M., Docchio, F., Sansoni, G. (2018). Deep learning based machine vision: First steps towards a hand gesture recognition set up for collaborative robots. In Proceedings of the 2018 Workshop on Metrology for Industry 4.0 and IoT, Brescia, Italy, 16–18 April 2018; pp. 28–33.

Ozdemir, M.A., Kisa, D.H., Guren, O., Onan, A., and Akan, A., (2020). "EMG Based Hand Gesture Recognition Using Deep Learning". In 2020 Medical Technologies Congress (TIPTEKNO). Antalya, Turkey: IEEE, pp. 1-4.

Papastratis, I., Chatzikonstantinou, C., Konstantinidis, D., Dimitropoulos, K., and Daras, P., (2021). "Artificial Intelligence Technologies for Sign Language". Sensors 21(17):5843. doi: 10.3390/s21175843.

Robertson, D. G. E., Caldwell, G. E., Hamill, J., Kamen, G., and Whittlesey, S. (2013). "Research Methods in Biomechanics." Human Kinetics.

Shakeel, Z.M, So, S., Lingga, P., and Jeong, J.P. (2020). "MAST: Myo Armband Sign-Language Translator for Human Hand Activity Classification".In 2020 International Conference on Information and Communication Technology Convergence (ICTC). Jeju, Korea (South), pp. 494-499.

Shin, S., Baek, Y., Lee, J., Eun, Y., and Son, S.H. (2017). "Korean Sign Language Recognition Using EMG and IMU Sensors Based on Group-Dependent NN Models". In 2017 IEEE Symposium Series on Computational Intelligence (SSCI). Honolulu, HI, pp. 1-7.

Tryon, J., and Trejos, A.L. (2021). "Evaluating Convolutional Neural Networks as a Method of EEG–EMG Fusion". Frontiers in Neurorobotics 15:692183. doi: 10.3389/fnbot.2021.692183.

Vásconez, J.P., L.I. Barona López, Valdivieso Caraguay, Á.L., and Benalcázar, M.E. (2022). "Hand Gesture Recognition Using EMG-IMU Signals and Deep Q-Networks". Sensors 22(24):9613. doi: 10.3390/s22249613.

Yang, L., Chen, J. A., and Zhu, W. (2020). Dynamic hand gesture recognition based on a leap motion controller and two-layer bidirectional recurrent neural network. Sensors, 20(7), 2106.

Yang, K., and Zhang, Z. (2019). "Real-Time Pattern Recognition for Hand Gesture Based on ANN and Surface EMG". In 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC). Chongqing, China, pp. 799-802.

Zia ur Rehman, M., Waris, A., Gilani, S.O., Jochumsen, M., Niazi, I., Jamil, M., Farina, D., and Kamavuako, E. (2018). "Multiday EMG-Based Classification of Hand Motions with Deep Learning Techniques". Sensors 18(8):2497. doi: 10.3390/s18082497.

UzbekTagger: The rule-based POS tagger for Uzbek language

Maksud Sharipov[1], Elmurod Kuriyozov[1,2], Ollabergan Yuldashev[1], Og'abek Sobirov[1]

[1] Urgench State University, Department of Information Technologies; 14, Kh.Alimdjan str, Urgench,Uzbekistan;
{m.sharipov,elmurod1202,ollaberganyuldashov,sobirov_o}@urdu.uz
[2] Universidade da Coruña, CITIC; Campus de Elviña, A Coruña, 15071, Spain;
e.kuriyozov@udc.es

Abstract

This research paper presents a part-of-speech (POS) annotated dataset and tagger tool for the low-resource Uzbek language. The dataset includes 12 tags, which were used to develop a rule-based POS-tagger tool. The corpus text used in the annotation process was made sure to be balanced over 20 different fields in order to ensure its representativeness. Uzbek being an agglutinative language so the most of the words in an Uzbek sentence are formed by adding suffixes. This nature of it makes the POS-tagging task difficult to find the stems of words and the right part-of-speech they belong to. The methodology proposed in this research is the stemming of the words with an affix/suffix stripping approach including database of the stem forms of the words in the Uzbek language. The tagger tool was tested on the annotated dataset and showed high accuracy in identifying and tagging parts of speech in Uzbek text. This newly presented dataset and tagger tool can be used for a variety of natural language processing tasks such as language modeling, machine translation, and text-to-speech synthesis. The presented dataset is the first of its kind to be made publicly available for Uzbek, and the POS-tagger tool created can also be used as a pivot to use as a base for other closely-related Turkic languages.

Keywords: Uzbek language, part-of-speech, POS-tagger, dataset.

1. Introduction

Part-of-Speech (POS) tagging is a process of identifying and labeling the grammatical category of each word in a given text. POS tagging is a fundamental task in natural language processing (NLP) and is used in a wide range of applications such as text analysis, machine translation, language modeling, and information retrieval. It is also a key step in many other NLP tasks, such as syntactic parsing, named entity recognition, and sentiment analysis.

POS tagging has evolved from rule-based systems (Kupiec, 1992) to machine learning-based models(Awasthi et al., 2006; Constant & Sigogne, 2011), and now deep learning-based models (dos Santos & Zadrozny, 2014; Meftah & Semmar, 2018; Perez-Ortiz & Forcada, 2001). It is widely used in various applications and continues to be an active area of research in the field of NLP (Manning, 2011).

In this research, we present UzbekTagger - a Part-of-Speech (POS) tagger tool and an annotated dataset for the Uzbek language. Firstly, we carefully analysed the previous studies as well as the linguistic nature of the language under focus, and decided 12 POS tags to be used. Then, the rule-based POS-tagger tool for Uzbek, called UzbekTagger, was created in Python. The tool is based on stems and suffix-affix data and rules in our codebase, allowing for efficient and accurate tagging of given text in Uzbek. Lastly, we manually annotated a special Uzbek corpus which was balanced over 23 distinct fields with ~1K words each to ensure its representative nature. The POS-tagging guidelines of Universal Dependencies (version 2)[1] were followed during the creation of the tagger.

The POS tags proposed for Uzbek in this work are twelve main categories are following:

- **Open word classes:** noun, verb, adjective, numeral, adverb, pronoun;
- **Closed word classes:** auxiliary, conjunction, particle;
- **Intermediate words:** modal words, imitation words, interjection words.

All the identifier names of the proposed tags, their meaning and example Uzbek words are given in Table 1.

Id	Tag	Meaning	Uzbek examples
1	NOUN	OT (*Noun*)	olma (*apple*)
2	VERB	FE'L (*Verb*)	yugurmoq (*run*)
3	ADJ	SIFAT (*Adjective*)	ko'p (*many/much*)
4	NUM	SON (*Numeral*)	besh (*five*)
5	ADV	RAVISH (*Adverb*)	tez (*fast*)
6	PRON	OLMOSH (*Pronoun*)	bu (*this*)
7	AUX	KO'MAKCHI (*Auxiliary*)	bilan (*with*)
8	CONJ	BOG'LOVCHI (*Conjunction*)	va (*and*)
9	PART	YUKLAMA (*Particle*)	faqat (*only*)
10	MOD	MODAL (*Modal*)	darhaqiqat (*actually*)
11	IMIT	TAQLID (*Imitation*)	kuk-kuk (*imitation of a hen*)
12	INTJ	UNDOV (*Interjection*)	hoorah! (*when you win*)

Table 1. All the proposed POS Tags for the Uzbek language with their meaning and example words.

The main reason behind the rule-based POS-tagging choice in this work is because of the lack of labelled data big enough to feed the neural network models to expect a good accuracy results. In fact, the output of this tagger can be used as a source for modern POS neural network models.

[1] Universal Dependencies POS-tags and guidelines: https://universaldependencies.org/u/pos/

Apart from that, when POS tagging is applied to languages with rich morphology and agglutination, the rule-based approach is more effective in tagging unfamiliar words (Anbananthen et al., 2017).

Uzbek language. Uzbek is a Turkic language spoken by over 40 million people, primarily in Uzbekistan as an official and in neighboring countries as a second language.

The official script of the language is Latin, but the old Cyrillic script is still in use both in official and unofficial basis. The language is, like other Turkic languages in the same family, highly-agglutinative, with SOV word order and does not poses neither gender nor articles. It has been influenced by both the Persian and Russian languages due to historical and cultural interactions[2].

Despite its significant number of speakers, Uzbek is considered a low-resource language in the field of NLP. This is because there is a limited amount of labeled data and resources available for Uzbek language, making it difficult to develop and evaluate NLP models for this language.

The POS-tagger tool created in this research work was assessed using the new dataset, and the experiment results show that the tool has achieved at least 85% accuracy in every field, reaching almost 90% average accuracy for the overall dataset.

The lack of resources for the Uzbek language makes it a challenging task for NLP researchers, however, it also presents an opportunity to contribute to the field by developing NLP models for Uzbek and other low-resource languages. This research aims to provide a valuable resource for NLP tasks in Uzbek, as the presented dataset and the tagging tool, to our best knowledge, is the first of its kind for the low-resource Uzbek language.

2. Related Work

POS tagging has evolved over the years, starting with rule-based systems that relied on hand-written grammar rules to identify the POS of words (Voutilainen, 2003). These systems were limited in their accuracy and were not able to handle the complexity and variability of natural language.

With the advent of machine learning, statistical models were developed to automatically learn the POS tags from annotated corpora. These models, such as Hidden Markov Models (HMM) (Kupiec, 1992) and Conditional Random Fields (CRF) (Awasthi et al., 2006; Constant & Sigogne, 2011), have improved the accuracy of POS tagging.

With the recent advancements in deep learning, neural network-based models have been developed that have further improved the accuracy and efficiency of POS tagging, using both word-level (Meftah & Semmar, 2018; Perez-Ortiz & Forcada, 2001) and character level representations (dos Santos & Zadrozny, 2014) of POS tagging.

Related work in the field of NLP for the Uzbek language has primarily focused on the development of resources such as WordNet (K. A. Madatov et al., 2022), datasets for sentiment analysis (Kuriyozov et al., 2022; Matlatipov et al., 2022), as well as semantic evaluation (Salaev et al., 2022b). However, there has been a rapid growth on the development of NLP tools for Uzbek, such as stopword removal (K. Madatov et al., 2022), transliterator(Salaev et al., 2022a), and stemmer (Sharipov & Salaev, 2022; Sharipov & Yuldashov, 2022) recently.

A specific work about Uzbek POS-tagging by Abjalova and Iskandarov (Abdurashetona & Ismailovich, 2021) also propose 12 tags for the Uzbek parts of speech. Currently, the field of NLP is developing rapidly and playing an important role in solving problems in scientific, economic and cultural fields (Sharipov et al., 2022).

3. Methodology

This section is devoted to the methodological part of the research work, starting from the details of the tagging algorithm used, followed by the text normalization steps, all the way till the corpus creation and the annotation process.

3.1 Tagging algorithm

A specific algorithm was used to properly create the POS-tagger tool: Given Uzbek text is first tokenized into sentence and then word levels, then tokens(words) are searched from the dictionary of lemmas[3] from an existing previous work (Sharipov & Sobirov, 2022) and other available sources like Apertium package for Uzbek[4] . If the lemma is found, the word class corresponding to it is determined accordingly. In the case of a token (word) being found in a dictionary in more than one class, then the sequence of suffixes of this token (word) are taken and searched in the dictionary of suffixes.

If there are no suffixes in the word and there is a problem in determining which word group it belongs to, in that case, our proposed algorithm determines the category of the current word depending on the category of the words surrounding it (words that are coming after and before it). Let's have a look at the following example: "*Yaxshi ovqat yesang, yaxshi ishlaysan.*" (If you eat well, you work well). The first word *"yaxshi"* (good) in this sentence is an adjective, the second word *"yaxshi"*(well) is an adverb here. There are no suffixes at the word *"yaxshi"*, therefore, it is not possible to determine which category this word belongs to based on its suffixes, so we determine the class of the word using the neighboring words. If the word has more than one possible tag, then rule-based taggers use hand-written rules to identify the correct tag.

A dictionary of more than 80,000 Uzbek words was created alongside their 12 POS tags in an XML format were created as the main source of the UzbekTagger tool. Besides, special rules were developed to identify two words with the same tag. Some of the created rules the tagger contains are listed below as an example:

- IF previous word's POS is adjective THEN the current POS is noun;
- IF previous word's POS is adverb THEN current POS is verb;

[2] More about the Uzbek language: https://en.wikipedia.org/wiki/Uzbek_language

[3] As the definition of lemma varies among the existing NLP research works, the term in this paper is refer as the dictionary form of word.

[4] Apertium monolingual package for Uzbek: https://github.com/apertium/apertium-uzb

- IF next WORD takes yordamchi fel THEN current [with next] one gets the verb POS;
- IF previous WORD_SUF is egalik THEN current POS is noun;
- IF current WORD_SUF is verb_suffix THEN current POS is verb;
- IF current WORD_SUF is noun_suffix THEN current POS is noun;
- IF current WORD is bog`lovchi THEN previous and next POS is the same;

3.2. Normalization

During the creation of the tagger tool, we encountered several problems with text normalization. There are 29 letters and 1 apostrophe (’) in the Uzbek language’s official Latin script. Two of them are these letters: **o‘** and **g‘**, in texts, there are cases where these two letters’ sign are replaced by a apostrophe: o’ and g’, or completely different characters are used: o', o`, g', g`. In such cases, tokenizers tokenize incorrectly. Let’s tokenize this sentence: “O`qituvchi gapirdi”, tokens: “O”, “qituvchi”, “gapirdi”, but in the correct form this sentence must consist of two tokens: “O‘qituvchi”, “gapirdi”. In Uzbek, the apostrophe does not come after the letters o and g. Therefore, in solving this problem, we changed all the signs after o and g to (‘) [*similar to the number 6*], and in other cases to (’) [*similar to the number 9*]. Below are some examples:

o`rdak->o‘rdak, (duck)
g'ildirak->g‘ildirak, (wheel)
ta`lim->ta’lim, (education)

№	Category	Sentences	Words
1	Adabiyot (Literature)	76	999
2	Anatomiya (Anatomy)	60	1020
3	Biologiya (Biology)	87	1001
4	Botanika (Botany)	59	1014
5	Din tarixi (History of relegion)	67	1016
6	Dunyo (World)	74	1006
7	Fizika (Physics)	81	1008
8	Geografiya (Geography)	61	1002
9	Huquq (Law)	57	1014
10	Informatika (Informatics)	84	1005
11	Iqtisodiyot (Economy)	38	1027
12	Jamiyat (Society)	44	1003
13	Kimyo (Chemstry)	75	1002
14	Madaniyat (Culture)	72	1000
15	Matematika (Mathematics)	43	999
16	Ona tili (Mother tongue)	98	1012
17	Qishloq xo‘jaligi (Agriculture)	69	1006
18	Siyosat (Politics)	54	1305
19	Sport (Sports)	78	1008
20	Tarix (Hitory)	85	1005
21	Texnologiya (Technology)	74	1005
22	Tibbiyot (Medicine)	52	1013
23	Zoologiya (Zoology)	93	1012
	Total:	**1581**	**23482**

Table 2. Number of sentences and words per category in the created corpus.

3.3. Corpus annotation

One of the most important factors that show the true performance of a POS Tagger is the corpus it was used to asses. In particular, the size of the corpus, the reliability of the tagged corpus, and the diversity of the corpus have a great effect (Can et al., 2021).

Due to the lack of openly-available Uzbek corpus that is diverse enough and is equally balanced over the different fields, we developed a tagged corpus that is evenly distributed across different categories. The raw text was obtained from books openly available at the Republican Youth E-Library[5] and the category the text belongs to was assigned based on the field the book belongs to.

The tagged dataset contains 23 categories with total number of 1581 sentences made of 23482 words in total. An average of 1000 words were taken from almost all fields available. The detailed composition of the categories, and number of sentences taken are presented in Table 2.

The same POS-tags were used and the same guidelines as the POS-tagger tool were followed during the annotation process. Four annotators with an expert-level linguistic knowledge of Uzbek annotated the created corpus over the course of six months. Each sentence was assured to be annotated at least by two individuals to overcome the human error. The problem of sentences with conflicting tags was solved by a group discussion to choose the right tags.

The choice of so many POS tags for the annotation was a result of an effort to cover all possible word forms as much as possible. This way, the annotated text will avoid possible misconceptions among homonyms. For instance, in the field of biology, the word “tut” (*mulberry*) has to be a noun in the sense of a fruit, and in the field of sports, the word “tut” (*catch*) has to be a verb in the sense of an action.

4. Experimental results

For the experiments, we checked the performance of the created POS-tagger tool using the annotated dataset as a source of evaluation. The UzbekTagger tool, which was made as a Python library was fed with raw sentences taken from the annotated corpus, then the output from the tagger was compared with the manually annotated format of the same sentence.

As an additional mean of evaluation the authors also considered the category the sentence belongs to, so that the overall analysis allows to identify on which categories there is a need for more work. Accuracy was chosen as the main metric of evaluation.

To explain the comparison of the tagger output and the manual annotation, let us take an example sentence from the category of Informatics:

“Mantiqiy formulalar rostlik jadvallari yordamida izohlanadi.” (Logical formulas are interpreted using truth tables.)

The output of the UzbekTagger is as follows:

“Mantiqiy/NOUN formulalar/NOUN rostlik/NOUN jadvallari/NOUN yordamida/NOUN izohlanadi/VERB ./PUNCT”

In this example, the first word “Mantiqiy” [*Logical*] is not actually a NOUN, rather it should be an ADJ, so this case was counted as one mistake. The only condition is that if the

[5] The Republican Youth E-Library: https://kitob.uz

same word appears wrongly tagged more than once, it was still considered as one mistake.

The total mistakes were then calculated, and the detailed performance results over each category are reported in Table 3.

№	Category	Mistakes	Accuracy
1	Adabiyot (Literature)	126	87.40%
2	Anatomiya (Anatomy)	31	96.97%
3	Biologiya (Biology)	96	90.41%
4	Botanika (Botany)	32	96.85%
5	Din tarixi (History of relegion)	79	92.23%
6	Dunyo (World)	56	94.44%
7	Fizika (Physics)	86	91.47%
8	Geografiya (Geography)	178	82.24%
9	Huquq (Law)	157	84.52%
10	Informatika (Informatics)	115	88.56%
11	Iqtisodiyot (Economy)	96	90.66%
12	Jamiyat (Society)	92	90.83%
13	Kimyo (Chemstry)	100	90.00%
14	Madaniyat (Culture)	88	91.20%
15	Matematika (Mathematics)	82	91.80%
16	Ona tili (Mother tongue)	168	83.40%
17	Qishloq xo‘jaligi (Agriculture)	205	79.50%
18	Siyosat (Politics)	120	90.81%
19	Sport (Sports)	101	89.99%
20	Tarix (Hitory)	72	92.84%
21	Texnologiya (Technology)	118	88.26%
22	Tibbiyot (Medicine)	144	85.79%
23	Zoologiya (Zoology)	58	94.27%
Total:		**2400**	**89.78%**

Table 3. Number of mistakes and accuracy in each category

The results show that the POS-tagger tool performs with at least 83% accuracy in all categories of Uzbek text, with up to 97% accuracy in some others. This indicates that the tagger tool has already includes terminology from fields like Anatomy, Botany and Math, while the terminology has to be enriched for some other fields like Agricultur, Mother tounge, and Law.

5. Discussion

When solving the problem of tagging Uzbek language texts by word classes, there is a difficulty in determining the POS tag when the same words appear in sentences in different word classes. For example, in the sentence “*U juda qattiq ishlar edi*” [He/She was very hard-working], the word “*ishlar*” (work) is a verb, but in the sentence “*Kechagi bo‘lib o‘tgan ishlar yaxshi emas*” (Things happened yesterday were not good), the word “*ishlar*”(things) is a noun. When tagging words in the above case, it is necessary to make a conclusion by knowing which word class the word that comes before and after tagging a word belongs to.

According to the results of the experts' analysis in Table 3, Agriculture showed the lowest Accuracy of 79.50%, while Botany showed the highest Accuracy of 96.85%. The main reason for this kind of a trend happening in the performance can be explained by the scope of the terminology words included in the tagger’s stems dictionary, which has to be improved, especially for the fields the tagger is struggling with.

6. Conclusion and future work

In conclusion, we presented the first publicly available POS-tagged dataset with more than 1500 sentences, annotated using a balanced Uzbek corpus. Also, the first openly available rule-based Uzbek POS-tagger tool was introduced alongside the dataset that achieved high accuracy results when tested on the annotated dataset. The UzbekTagger tool achieved about 90% overall accuracy over the dataset with more than 20 fields.

In the future, the researchers plan to improve the performance of the POS tagger by incorporating machine learning and neural network techniques. Additionally, the researchers aim to expand the annotated dataset to include more data from different fields. Furthermore, the researchers plan to develop more sophisticated NLP tools for Uzbek, such as dependency parser using the POS-annotated dataset and the tagger tool which will provide more comprehensive NLP support for the Uzbek language.

Data availability.

The developed a Python-based POS-tagger tool for the Uzbek language is available to be installed and used via the Python Package Index (PyPi)[6]. Apart from that, both the source code of the POS-tagger tool and the annotated dataset files can be found at the project GitHub repository[7]. To our best knowledge, no open-source POS-tagger tool has been created or made publicly available.

Acknowledgements.

This research work was fully funded by the REP-25112021/113 - “UzUDT: Universal Dependencies Treebank and parser for natural language processing on the Uzbek language” subproject funded by The World Bank project “Modernizing Uzbekistan national innovation system” under the Ministry of Innovative Development of Uzbekistan.

Declarations.

The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

Abdurashetona, A. M., & Ismailovich, I. O. (2021). Methods of Tagging Part of Speech of Uzbek Language. *2021 6th International Conference on Computer Science and Engineering (UBMK)*, 82–85.

[6] https://pypi.org/project/UzbekTagger.

[7] https://github.com/MaksudSharipov/UzbekTokenizer.

Anbananthen, K. S. M., Krishnan, J. K., Sayeed, M. S., & Muniapan, P. (2017). Comparison of stochastic and rule-based POS tagging on Malay online text. *American Journal of Applied Sciences*, *14*(9), 843–851.

Awasthi, P., Rao, D., & Ravindran, B. (2006). Part of speech tagging and chunking with hmm and crf. *Proceedings of NLP Association of India (NLPAI) Machine Learning Contest 2006*.

Can, Ş., Karaoğlan, B., Kşla, T., & Metin, S. K. (2021). Using Word Embeddings in Turkish Part of Speech Tagging. *International Journal of Machine Learning and Computing*, *11*(5).

Constant, M., & Sigogne, A. (2011). MWU-aware part-of-speech tagging with a CRF model and lexical resources. *Proceedings of the Workshop on Multiword Expressions: From Parsing and Generation to the Real World*, 49–56.

dos Santos, C., & Zadrozny, B. (2014). Learning character-level representations for part-of-speech tagging. *International Conference on Machine Learning*, 1818–1826.

Kupiec, J. (1992). Robust part-of-speech tagging using a hidden Markov model. *Computer Speech & Language*, *6*(3), 225–242.

Kuriyozov, E., Matlatipov, S., Alonso, M. A., & Gómez-Rodr\'\iguez, C. (2022). Construction and evaluation of sentiment datasets for low-resource languages: The case of Uzbek. *Language and Technology Conference*, 232–243.

Madatov, K. A., Khujamov, D. J., & Boltayev, B. R. (2022). Creating of the Uzbek WordNet based on Turkish WordNet. *AIP Conference Proceedings*, *2432*(1), 60009.

Madatov, K., Bekchanov, S., & Vičič, J. (2022). *Automatic Detection of Stop Words for Texts in the Uzbek Language*. Preprints. https://doi.org/10.20944/preprints202204.0234.v1

Manning, C. D. (2011). Part-of-speech tagging from 97% to 100%: is it time for some linguistics? *International Conference on Intelligent Text Processing and Computational Linguistics*, 171–189.

Matlatipov, S., Rahimboeva, H., Rajabov, J., & Kuriyozov, E. (2022). Uzbek Sentiment Analysis Based on Local Restaurant Reviews. *CEUR Workshop Proceedings*, *3315*, 126–136. www.scopus.com

Meftah, S., & Semmar, N. (2018). A neural network model for part-of-speech tagging of social media texts. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Perez-Ortiz, J. A., & Forcada, M. L. (2001). Part-of-speech tagging with recurrent neural networks. *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*, *3*, 1588–1592.

Salaev, U., Kuriyozov, E., & Gómez-Rodr\'\iguez, C. (2022a). A machine transliteration tool between Uzbek alphabets. *CEUR Workshop Proceedings*, *3315*, 42–50. www.scopus.com

Salaev, U., Kuriyozov, E., & Gómez-Rodr\'\iguez, C. (2022b). SimRelUz: Similarity and Relatedness scores as a Semantic Evaluation dataset for Uzbek language. *ArXiv Preprint ArXiv:2205.06072*.

Salaev, U., Kuriyozov, E., & Gómez-Rodr\'\iguez, C. (2022c). SimRelUz: Similarity and Relatedness scores as a Semantic Evaluation Dataset for Uzbek Language. *1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages, SIGUL 2022 - Held in Conjunction with the International Conference on Language Resources and Evaluation, LREC 2022 - Proceedings*, 199–206. www.scopus.com

Sharipov, M., Mattiev, J., Sobirov, J., & Baltayev, R. (2022). Creating a morphological and syntactic tagged corpus for the Uzbek language. *ArXiv Preprint ArXiv:2210.15234*.

Sharipov, M., & Salaev, U. (2022). Uzbek affix finite state machine for stemming. *ArXiv Preprint ArXiv:2205.10078*.

Sharipov, M., & Sobirov, O. (2022). Development of a rule-based lemmatization algorithm through Finite State Machine for Uzbek language. *ArXiv Preprint ArXiv:2210.16006*.

Sharipov, M., & Yuldashov, O. (2022). UzbekStemmer: Development of a Rule-Based Stemming Algorithm for Uzbek Language. *ArXiv Preprint ArXiv:2210.16011*.

Voutilainen, A. (2003). *Part-of-speech tagging* (Vol. 219). The Oxford handbook of computational linguistics.

Strategies for creating corpora and language resources for under-resourced South African indigenous languages

Nomsa Skosana[1], Respect Mlambo[1], Muzi Matfunjwa[1]

[1]South African Centre for Digital Language Resources (SADiLaR), North-West University, Potchefstroom, South Africa

{Nomsa.Skosana, Respect.Mlambo, Muzi.Matfunjwa}@nwu.ac.za

Abstract

Amongst the eleven official languages of South Africa, nine are indigenous and are perceived as under-resourced, particularly in the field of Human Language Technology (HLT). Since 1994 positive strides have been taken by various entities to develop the indigenous languages by creating open access HLT tools. However, there have been some impediments in the development of these languages such as the availability of corpora which plays a crucial role in their development. Therefore, this article proposes strategies for developing corpora and language resources for South African indigenous languages. The five recommended strategies are creation of a universal data bank, using the web or crawling websites, reusing corpora, adopting tools from sister languages, and collaboration between stakeholders in the HLT field. These strategies are suitable for the South African context as they point to where rich corpora exist. If these strategies are correctly implemented, they could help in creating and availing corpora for developing the language resources for the under-resourced indigenous languages.

Keywords: Corpora, Language resources, Human language technology, Under-resourced languages, Indigenous languages

1. Introduction

South Africa is a multilingual country that has eleven official languages namely Sepedi, Sesotho, Setswana, Siswati, Tshivenda, Xitsonga, Afrikaans, English, isiNdebele, isiXhosa and isiZulu (South African Constitution, 1996). Except for Afrikaans and English, the other nine languages are indigenous languages which are perceived as under-resourced. Not all of the South African languages are at the same level in terms of available corpora and language resources. Lack of corpora and resources for the indigenous languages have been a challenge in developing Human Language Technology (HLT) resources (Eiselen and Puttkammer, 2014). The major obstacle that has led to insufficient development and creation of relevant resources for indigenous languages is the lack of appropriate terminology in contemporary fields (Ngcobo and Nomdebevana, 2010; Mlambo et al., 2021).

The creation of corpora that will be used to develop relevant resources for under-resourced South African indigenous languages is essential in the HLT field. Such development is in line with the constitutional mandate of promoting the status of indigenous languages and advancing their use in contemporary fields (Mlambo et al., 2021). In the past decades, the South African government has been engaged in funding entities that intend to develop the indigenous languages of the country. The main aims of the entities were to collect annotated corpora and create language tools. Despite such vigorous efforts, studies have shown that up to date the indigenous languages are still under-resourced (Taljard and Bosch, 2006; Bosch et al., 2008; Ngcobo and Nomdebevana, 2010; Eiselen and Puttkammer, 2014; Van Niekerk et al., 2017; Skosana and Mlambo, 2021; Mlambo et al., 2021). This article proposes five strategies for creating suitable corpora and resources for the indigenous South African languages. These languages are resource-scarce due to a lack of corpora required for developing HLT tools; therefore, it is critical to devise strategies to address this issue so that these languages can benefit from these tools.

2. Related work

Maseko et al. (2010) investigated the internet's potential role in promoting the use and status of indigenous languages. The study showed that the internet can play an important role in the economic, social, and educational development of indigenous languages. It was stated that resources in these languages are available, but their visibility on the internet is minimal. Hence, the available resources merely serve as a symbol rather than an instrumental role in promoting indigenous languages. The researchers asserted that the government, communities, non-governmental organisations, academic institutions, and the media industry are important stakeholders in increasing the visibility of indigenous languages on the internet. Such initiative will not only make indigenous languages visible, but it will also help to develop corpora and technologies as developers will find it easier to crawl the information.

Getao and Miriti (2006) discussed steps to create an application that was used to build a Kiswahili corpus. The first step was to develop an initial model that was used to determine and categorise documents which were downloaded from the web into their specific languages. They were obtained using a search engine, which returned a set of pages with links which were collected and stored after a search engine query. The documents were then processed and converted into appropriate formats, such as htm, html and txt files. The automatic processing of the documents included the removal of sentences containing digits, brackets, and sentences which consisted of less than five words, as such sentences are grammatically incorrect in Kiswahili. Following these steps, the documents were then uploaded to the corpus and utilised to augment the Kiswahili wordlists. As a result, a corpus of 5 million

tokens was built in less than 24 hours. It was proposed that the methods used to build the Kiswahili corpus can also be applied to other languages.

El-Haj et al. (2015) identified three paradigms for creating Arabic resources. These were crowdsourcing, translating existing datasets, and a manual approach. Selected Arabic students used Wikipedia and Arabic newspaper websites to crowdsource. They chose sentences from the downloaded documents that were deemed as a worthy corpus after quality control had been conducted. A total of 18 264 tokens were collected from this exercise to develop an Arabic corpus. Machine translation was used as a second strategy for the creation of Arabic resources in which 17 340 English sentences were translated from the National Institute of Standards using Google Translate API. The third strategy involved collecting a corpus manually. Twelve fluent participants in English and Arabic were selected to translate, summarise, and validate 100 documents extracted from English Wikinews. Even though this strategy is mostly expensive, it is effective in creating a high quality corpus. While crowdsourcing is quick and less expensive, a corpus created through this method is not of a high quality.

Wanjawa et al. (2022) explored how corpora for Swahili, Dholuo and Luhya as low-resource languages could be collected. The authors collected speech and text data that was intended to alleviate the gap in the availability of data meant for Machine Learning tasks and Natural Language Processing. A collaboration between entities was used to collect data in which researchers gathered data from many sources such as communities and schools. The joint work of collecting the corpus also included the media and publishers to obtain various types of data in the languages. The collaborative work between researchers resulted in the creation of Kencorpus which consisted of 5594 items, with 4442 texts of 5.6 million words and 1152 speech files of 117 hours. The created Kencorpus for three languages was considered essential in availing a corpus for low-resource languages.

Juan (2015) investigated the use of corpora in closely related languages. The study presented strategies for developing an Automatic Speech Recognition (ASR) system for Iban which is a resource-scarce language, using corpora from Malay which is a well-resourced language. These languages are closely related because they belong to the same language family, with similar phonological description and writing system. A semi-supervised method for constructing a pronunciation dictionary and cross-lingual strategies to improve acoustic models trained with very little training data from Malay was created. The processes significantly improved the performance of Iban ASR, which demonstrated that corpora from a closely related language can be used to build ASR for another language. An alternative method known as zero-shot ASR method that uses Malay corpora was also proposed for transcribing Iban speech. It was ascertained that the difficulties in developing ASR systems for under-resourced languages such as Iban have grown due to a lack of training corpora and pronunciation dictionaries.

From the literature reviewed, the studies have shown several strategies that were used for creating corpora for other resource-scarce languages. It has been found that various strategies are employed to create corpora and resources for these languages. However, these techniques have not been explored in South African indigenous languages. Therefore, this study fills the gap by providing viable methods for creating suitable corpora and resources for the indigenous South African languages.

3. Discussion of strategies

This section discusses five strategies for creating suitable corpora and resources for the indigenous South African languages.

3.1. Web crawling

Web crawling is used recursively to download complete documents or data from websites, then converting them to plain text, tokenising and creating frequency lists for its database (Scannell, 2007). This method is significant for gathering corpora from websites, which are constantly growing (Singh and Varnica, 2014). In South Africa, several websites such as Storybooks South Africa https://global-asp.github.io/storybooks-southafrica/stories/nr/, Jehovah's Witnesses https://www.jw.org/en/, The Presidency Republic of South Africa https://www.thepresidency.gov.za/, and Parliament of the Republic South Africa https://www.parliament.gov.za/ have information in all indigenous languages that can be used to build large corpora. Newspaper websites that contain monolingual information in indigenous languages such as Nthavela https://www.nthavela.co.za/, Ilanga https://ilanganews.co.za/, Isolezwe https://www.isolezwe.co.za/, Seipone Madireng https://seiponemadireng.co.za/about-us, and I'solezwe lesiXhosa https://www.isolezwelesixhosa.co.za/ can also be used for crawling. The web crawling strategy facilitates the collection of corpora from multidimensional fields. Such corpora in indigenous languages are essential as they represent various usages of the languages and could play a vital role in creating and developing HLT tools that are efficient.

3.2. Universal data storage

Creating an open access universal data storage system that can be used to keep corpora in such a way that they can be retrieved and used in the future can help to overcome the lack of corpora for developing HLT tools for South African indigenous languages. The system could function as a platform for various translation agencies, government departments, standardisation sectors, and communities to share their indigenous language corpora, which will aid in the development of HLT tools for these languages. This system will be sustainable for these languages, as there are large texts available in South African languages as noted by Van Niekerk et al. (2017). Piperidis (2012) presented a similar platform (META-SHARE) that worked for the development of corpora that were used to create and develop HLT tools for European languages. The purpose of META-SHARE is to establish an infrastructure of interlinked repositories in which language resources and corpora are found and accessed in a universal platform (Piperidis, 2012). Similarly, the South African Universal Data Storage can be seen as a system of networked repositories that will be viewed as a national unified space in which language stakeholders can share their corpora. Such a system would not be limited to existing data only but the new and emerging language corpora, tools, and systems that are needed for the evaluation of existing and

the creation of new HLT tools would be targeted as well. The shared corpora will also be available for research purposes, which will help to improve the corpora's quality. The already established infrastructure, the South African Centre for Digital Language Resources (SADiLaR), is better positioned for this task as it has open access repositories which could be used for depositing indigenous language corpora. The corpora deposited in the Universal Data Storage system can be used to develop HLT tools for indigenous languages for noncommercial use. SADiLaR will collaborate with developers to use the corpora under a clear licensing agreement.

3.3. Reusing existing corpora

Developing HLT tools for resource-scarce languages, such as the South African indigenous languages, has always been hampered by a scarcity of high-quality training corpora. The causes of a lack of high-quality training corpora vary but may include limited access to technology for developing the language, a smaller number of speakers, or a lack of urgency for collecting and creating required corpora where the second language is well-resourced (Liu et al., 2022). Exploiting or recycling existing corpora from closely related languages with better resources can be seen as a strategy that can aid in the development of corpora and language resources for less-resourced languages. Among South Africa's nine indigenous languages, four are related Nguni languages (isiZulu, isiXhosa, isiNdebele, and Siswati) and three are related Sotho languages (Setswana, Sesotho, and Sepedi) (Roux, 2001). Nguni languages are written conjunctively, whereas Sotho languages are written disjunctively (Prinsloo and De Schryver, 2002). When compared to other related languages in the Nguni language group, isiZulu has more corpora and resources developed. The Autshumato Machine Translation Web Service, Google Translate, ZulMorph, and isiZulu.net are some of the HLT tools available in isiZulu that are not available in other Nguni languages, except for isiXhosa in Google Translate. Similarly, Sesotho and Sepedi as Sotho languages are available in Google Translate, but Setswana is not. The developers of these tools typically use a phased approach in selecting languages based on the availability of training data (Skosana and Mlambo, 2021). Therefore, this study proposes that the corpora used to develop such tools in isiZulu as a Nguni language and Sesotho and Sepedi as Sotho languages can be reused as a foundation to develop more or similar technologies for the other related languages. In the South African context, reusing existing corpora from closely related languages is also a useful practical method to overcome the scarcity of high-quality training corpora and resources for under-resourced indigenous languages.

3.4. Adopting tools from sister languages

In the context of South Africa, some indigenous languages are more developed than others. Among the Nguni languages, isiZulu takes the lead with the most language processing tools, followed by isiXhosa while Siswati and isiNdebele have fewer language tools. For example, an open access machine translation tool known as Autshumato Machine Translation Web Service https://mt.nwu.ac.za/ can translate from English to isiZulu while isiXhosa, Siswati and isiNdebele are not supported by the tool (Skosana and Mlambo, 2021). This developed machine translating tool can be adapted to the sister languages as they have a similar agglutinating morphological structure and share semantic relations to a large extent. Another language tool that can be adapted to the above-mentioned sister languages is ZulMorph. This is a morphological analyser for isiZulu which was created using Xerox finite state tools lexc and xfst. It can analyse words from their surface form to their base form (Pretorius and Bosch, 2018), meaning that a complete lexical item is used as an input and an automated morphological analysis of the word is then obtained as an output. To demonstrate that this tool can be adapted to Siswati, the Siswati words angifuni kudla (I do not want to eat) were accurately analysed by ZulMorph in Table 1 as angifuni > /a/ negation prefix, /ngi-/ subject concord first person, /-fun-/ verb root, /-i/ terminative vowel negation. Kudla > /ku-/ basic prefix class 15, /-dl-/ verb root, /-a/ terminative vowel or /ku-/ subject concord class 15, /-dl-/ verb root, /-a/ terminative vowel.

Fig. 1: Siswati words correctly analysed by ZulMorph

The similarity in the morphological structure and semantic relations between the languages can easily afford the adaptation of the ZulMorph tool or lead to a creation of a Siswati finite state morphological analyser possibly named “SwatMorph”. Heeringa et al. (2015) endorse that the use of already developed tools between languages of the same group makes it simpler for developers to adapt a tool to a sister language other than to create it from scratch. Therefore, the strategy to adopt existing language tools and adapting them to support sister languages is a viable approach in minimising the gap between indigenous South African languages.

3.5. Collaboration between stakeholders in the HLT field

The field of HLT encompasses a wide range of activities with the goal of allowing humans to communicate with machines using natural languages (Cole, 1997). Rashel (2011) defines HLT as a field that combines computer science, linguistics, and psychology to provide a variety of applications that allow users to interact with computer-assisted devices using their native languages. From these notions, it is clear that HLT is a multidisciplinary field that focuses on getting computers or software to do useful things with natural language, whether written or spoken. This shows that to develop HLT tools requires a significant collaborative effort from a diverse group of professionals. In South Africa one of the most serious issues affecting language development is a lack of collaboration among various stakeholders (Van Huyssteen, 1999). For example, the Pan South African Language Board was established to promote and create conditions for the development and use of all official languages in South Africa (Beukes, 2009). This board serves all South African official languages, but each language sector within the board is responsible for its own work and development. Their nine lexicography and terminology development units for the indigenous languages operate in isolation as they are based at different tertiary institutions across South Africa. According to Brand South Africa (2007) these units were established in 2001, with the primary goal of compiling monolingual explanatory dictionaries and other products to aid in language development. In the indigenous languages, only isiZulu and isiXhosa were able to publish a few volumes of monolingual dictionaries (Brand South Africa, 2007). If these units had been able to collaborate, they could have shared ideas and information on the compilation of various dictionaries and published in all 11 official languages simultaneously. Like in HLT, linguists, engineers, computer scientists, language boards, translation agencies, universities and communities are required to collaborate, which is currently lacking to vanquish the challenges of training corpora and language resources for indigenous languages. Linguists are required for a better understanding of the structure of human language, and engineers and computer scientists are needed to create complex systems for machines as well as to create and program the architectures of the tools (Bodomo, 2006). The development of HLT technologies for under-resourced languages would not be possible without collaboration between these diverse groups of professionals. Therefore, fruitful collaboration among stakeholders in the HLT field is critical in overcoming some of the major obstacles to the development of HLT tools for South African indigenous languages.

4. Conclusion

This paper presented five strategies, namely web crawling, universal data storage, reusing existing corpora, adopting tools from sister languages and collaboration between stakeholders which can be used to create corpora and develop resources for South African indigenous languages. These strategies have revealed that there are opportunities and possibilities for these languages to grow in the HLT field. The strategies are suitable for the South African context as they point to where rich corpora exist. If these strategies are correctly implemented, they could help in creating and availing corpora for developing the language resources for the under-resourced indigenous languages.

References

Beukes, A.M. (2009). Language policy incongruity and African languages in post-apartheid South Africa. In: *Language Matters*, 40 (1), pp. 35–55.

Bodomo, A. (2006). Human language technology in multilingual perspectives: Institutions and applications. In: *International Journal of Technology and Human Interaction,* 2 (1), pp. i-vi.

Bosch, S.E., Pretorius, L. and Fleisch, A. (2008). Experimental bootstrapping of morphological analyses for Nguni languages. In: *Nordic Journal of African Studies,* 17 (2), pp. 66–88.

Brand South Africa. (2007). *Pan South African Language Board.* Retrieved from: https://brandsouthafrica.com/111322/languageboard/ Access date: September 24, 2022.

Cole, R. (1997). *Survey of the state of art in human language technology.* Cambridge: Cambridge University Press.

El-Haj, M., Kruschwitz, U. and Fox, C. (2015). Creating language resources for under-resourced languages: Methodologies, and experiments with Arabic. In: *Language Resources & Evaluation*, 49, pp. 549–580.

Eiselen, E. and Puttkammer, M. (2014). *Developing text resources for ten South African languages. 9th Language Resources and Evaluation International Conference, Reykjavik, Iceland, May 26–31, 2014.*

Getao, K. and Miriti, E. (2006). Automatic construction of a Kiswahili corpus from the world wide web. In: Williams, D. and Daryamureeba, V. (Eds), *Measuring computing research excellence and vitality,* Volume 1, Fountain: Kampala, ISBN: 13-978-9970-02-592-3, pp. 209-219.

Heeringa, W., De Wet, F. and Van Huyssteen, G.B. (2015). Afrikaans and Dutch as closely-related languages: A comparison to West Germanic languages and Dutch dialects. *Stellenbosch Papers in Linguistics Plus*, 47, pp. 1-18.

Ilanga. Retrieved from: https://ilanganews.co.za/ Access date: November 20, 2022.

I'solezwe lesiXhosa. Retrieved from: https://www.isolezwelesixhosa.co.za/ Access date: November 20, 2022.

Isolezwe. Retrieved from: https://www.isolezwe.co.za/ Access date: November 20, 2022.

Jehovah's Witnesses. Retrieved from: https://www.jw.org/en/ Access date: November 20, 2022.

Juan, S.F.S. (2015). *Exploiting resources from closely-related languages for automatic speech recognition in low-resource languages from Malaysia.* Grenoble: Université Grenoble Alpes.

Liu, Z., Richardson, C., Hatcher Jr, R. and Prud'hommeaux, E. (2022). *Not always about you: Prioritizing community needs when developing endangered language technology. 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, May 22–27, 2022.*

Maseko, P., Nosilela, B., Sam, M., Terzoli, A. and Dalvit, L. (2010). The role of the web in the promotion of African languages. *Alternation,* 17 (1), pp. 312–327.
Mlambo, R., Skosana, N. and Matfunjwa, M. (2021). The extraction of terminology list using ParaConc for creating a quadrilingual dictionary. *Southern African Linguistics and Applied Language Studies,* 39 (1), pp. 82–91.
Ngcobo, M.N. and Nomdebevana, N. (2010). The role of spoken language corpora in the intellectualisation of indigenous languages in South Africa. *Alternation,* 17 (1), pp. 186–206.
Nthavela. Retrieved from: https://www.nthavela.co.za/ Access date: November 20, 2022.
Parliament of the Republic South Africa. Retrieved from: https://www.parliament.gov.za/ Access date: November 20, 2022.
Piperidis, S. (2012). *The META-SHARE language resources sharing infrastructure: Principles, challenges, solutions. Eighth International Conference on Language Resources and Evaluation, Istanbul, Turkey, May 21–27, 2012.*
Pretorius, L. and Bosch, S. (2018). ZulMorph: Finite state morphological analyser for Zulu (Version 20190103) [Software]. Web demo. Retrieved from: https://portal.sadilar.org/FiniteState/demo/zulmorph/. Access date: November 11, 2022.
Prinsloo, D.J., De Schryver, G.M. (2002). Towards an 11x11 array for the degree of conjunctivism/disjunctivism of the South African languages. *Nordic Journal of African Studies,* 11 (2), pp. 249-265.
Rashel, M.M. (2011). Introducing language technology and computational linguistics in Bangladesh. In: *International Journal of English Linguistics,* 1 (1), pp. 179–186.
Roux, J.C. (2001). HLT development in an African context: Planning for the next decade in South Africa. Retrieved from: https://www.academia.edu/45429052/HLT_development_in_an_African_context_planning_for_the_next_decade_in_South_Africa Access date: September 24, 2022.
Scannell, K. (2007). *The Crúbadán Project: Corpus building for under-resourced languages. Building and exploring web corpora. 3rd Web as Corpus Workshop, Louvain-la-Neuve, Belgium, May 15–16, 2007.*
Seipone Madireng. Retrieved from: https://seiponemadireng.co.za/about-us Access date: November 20, 2022.
Singh, M and Varnica, B. (2014). Web crawler: Extracting the web data. In: *International Journal of Computer Trends and Technology,* 13 (3), pp. 132–137.
Skosana, N.J. and Mlambo, R. (2021). A brief study of the Autshumato Machine Translation Web Service for South African languages. *Literator,* 42 (1), a1766.
South African Constitution. (1996). Constitution of the Republic of South Africa. Retrieved from: https://www.justice.gov.za/legislation/constitution/ Access date: October 20, 2022.
Storybooks South Africa. Retrieved from: https://global-asp.github.io/storybooks-southafrica/stories/nr/ Access date: November 20, 2022.
Taljard, E. and Bosch, S.E. (2006). A comparison of approaches to word class tagging: Disjunctively vs. conjunctively written Bantu languages. *Nordic Journal of African Studies,* 15 (4), pp. 428–442.
The Presidency Republic of South Africa. Retrieved from: https://www.thepresidency.gov.za/ Access date: November 20, 2022.
Van Huyssteen, L. (1999). Problems regarding term creation in the South African African languages, with special reference to Zulu. In: *South African Journal of African Languages,* 19 (3), pp. 179–187.
Van Niekerk, D., Van Heerden, C., Davel, M., Kleynhans, N., Kjartansson, O., Jansche, M. and Ha, L. (2017). 'Rapid development of TTS corpora for four South African languages.' In Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017.
Wanjawa, B., Wanzare, L., Indede, F., McOnyango, O., Ombui, E. and Muchemi, L. (2022). Kencorpus: A Kenyan language corpus of Swahili, Dholuo and Luhya for natural language processing tasks. arXiv preprint arXiv:2208.12081, pp. 1-16.

Sentiment Analysis of Polish Online News Covering Controversial Topics – – Comparison Between Lexicon and Statistical Approaches

Joanna Szwoch[1], Mateusz Staszkow[2], Rafal Rzepka[3], Kenji Araki[3]

[1]Graduate School of Information Science and Technology, Hokkaido University,
joannaeleonora.szwoch.u0@elms.hokudai.ac.jp
[2]Mateusz Staszkow Software Development, mateuszstasz@gmail.com
[3]Faculty of Information Science and Technology, Hokkaido University, {rzepka, araki}@ist.hokudai.ac.jp

Abstract

In this paper we present results of a sentiment analysis performed on a part of the Polish Online News Corpus which consists of articles covering controversial topics in Poland. We perform both lexicon-based and statistical experiments and compare the obtained results for this task. The main goal of our comparative work is to discover whether supposedly objective online news contain any emotive content and how different the results are, depending on the approach. Our study found no meaningful differences in emotional load at the word level, but the multilingual language model trained with the Natural Language Inference objective has partially confirmed its existence. We believe this work can be used for further research on media polarization in Polish language.

Keywords: Polish language, online news, sentiment analysis

1. Introduction

The Internet is a place full of content generated by people. To a large extent, people's views and decisions that follow them are shaped by the content they encounter online (Burbach et al., 2020). Moreover, in present days, traditional media such as newspapers, magazines, radio or even TV are slowly fading away, in favour of news outlets available on the Internet. News providers, according to the highest journalistic standards, are supposed to deliver pure information without any bias. However, there are examples where this condition is not met (Elejalde et al., 2018). In this work we want to investigate if two popular online news providers in Poland are following the aforementioned standards.

News outlets which are considered biased, tend to present information in favour of a certain side, whether it is a political party or a social movement. Their promotion of a specific perspective can lead to a spread of confusing or inaccurate information, causing mistrust and divisions within a society (Aggarwal et al., 2020).

Sentiment analysis and emotion detection are ways to determine the emotions hidden in text and to gain insights on how people feel about certain topics (Nandwani and Verma, 2021). It can help to detect fake and biased news as well as misinformation. In general, the problem of including opinions and emotions in contemporary media spotlights the importance of critical thinking and literacy in current digital age (Machete and Turpin, 2020).

The goal of sentiment analysis is to identify subjective information within a text, such as opinions or emotions. There are two approaches to this task, namely a rule-based and a statistical-based one. The former one relies on manually-crafted rules, including dictionaries and corpora to classify text into categories. Three classes, that is positive, negative and neutral are examples of such categorization, but they can also consist of various emotions. On the other hand, the latter approach can be divided into three methods – supervised, unsupervised and self-supervised. Both approaches have strong points and limitations depending on the type of input data and the use case.

Polish language is rather neglected when it comes to the number of studies on sentiment analysis. In comparison with other languages, there is a relatively small number of available resources. It is an obstacle for researchers wishing to delve into this topic. Despite this challenge, in this paper we experiment with two different approaches to opinion mining in Polish language. We also discuss the pros and cons of the utilized techniques and potential future directions in this field.

The rest of this paper is structured as follows. In Section 2 we mention prior works on sentiment analysis, including different approaches to the task depending on the length of text and language. Section 3 describes utilized datasets and Section 4 outlines methods applied in our experiments. In Section 5 we analyze obtained results and in Section 6 we present points open for discussion. Section 7 focuses on conclusions which can be drawn from the conducted experiments and our future work plans.

2. Related works

Opinion mining became a fairly popular branch of natural language studies, which resulted in a bountiful of works covering the topic (Mäntylä et al., 2016; Cui et al., 2023). Majority of studies relate to short inputs such as tweets or reviews. In our work we will focus on examining emotions in news, which is a fairly uncommon approach.

One of the examples of lexicon-based approaches is a study which performed sentiment analysis on English BBC news (Shirsat et al., 2017). Bing sentiment dictionary allowed

classification of positive, negative or neutral content. Such approach showed that in different news categories information tends to be polarized rather than impartial. Another popular lexicon-based technique includes VADER[1]. It is prepared for English vocabulary, but a common practice is to translate this source to any desired language, as in the study of sentiment analysis of health-related online news in Brazilian Portuguese (Mello et al., 2022). Other examples of dictionaries include WordStat Sentiment Dictionary, SentiWordNet, AFINN, Loughran & McDonald Financial Sentiment Dictionary (L&M), Lexicoder Sentiment Dictionary (Tomanek, 2014). Out of these, only SentiWordNet includes a dedicated Polish version – plWordNet (Kocoń et al., 2017).

Sentiment analysis conducted on online financial texts indicates that RoBERTa-based model outperformed results obtained by SVM, Logistic Regression or Naive Bayes (Zhao et al., 2020). Moreover, opinion mining of Italian reviews indicated that machine and deep learning approaches such as BERT perform better in general, however, lexicon-based methods are recommended for small datasets with limited resources (Catelli et al., 2022).

Polish language studies include sentiment analysis of conversation transcripts with a social robot (Probierz and Galuszka, 2022). For this type of long text, lexicon-based approach was applied. Two different dictionaries – English NRC (Mohammad and Turney, 2013) and Polish plWordNet were used for this task. Both of them allow to associate words with 8 different types of emotions. However, due to the limited number of analyzed conversations, the variation between results was not observed.

On the other hand, multilingual BERT model yielded good results in sentiment analysis of news in another Slavic language, namely Slovenian (Pelicon et al., 2020). Moreover, this zero-shot learning approach was also used on Croatian language for the same task and it outperformed the results of other classifying models.

However, no similar experiments were conducted on Polish online news.

3. Datasets

In this section we describe the datasets used for sentiment analysis of news covering controversial topics in Polish language.

3.1. NAWL – The Nencki Affective Word List

The Nencki Affective Word List (NAWL) is a database consisting of 2,902 Polish words such as nouns, verbs, and adjectives (Riegel et al., 2015). All of them are rated for emotional valence, arousal, and imageability. Furthermore, measures of frequency, grammatical class, and number of letters are also included in this set. NAWL is a Polish version of the Berlin Affective Word List - Reloaded (Võ et al., 2009), which was created for the same purpose, but in German language. This dictionary describes each word with a point value in terms of 5 categories: anger, disgust, fear, happiness and sadness. Additionally, neutral and unknown classes are included for words that do not have a dominant category. The higher the value, the stronger the word expresses the particular emotion.

NAWL is used in our non-statistical approach to sentiment analysis of words which appeared in examined online news. We chose this lexicon due to its accessibility and freeware licence.

3.2. Polish online news articles covering controversial topics

In this paper we used the Polish Online News Corpus to retrieve articles covering contemporary controversial topics in Poland (Szwoch et al., 2022). The original dataset consists of more than 200,000 online news articles in Polish language between years 2019 and 2021 from two major TV news broadcasters – TVP Info and TVN 24. As our work primarily focuses on performing sentiment analysis of articles related to subjects considered controversial, we extracted only relevant examples from the corpus. The topics were chosen arbitrarily and are related to contemporary topics which appear in Polish media. However, recent study on the political leaning discovery was an inspiration to consider topics such as EU and abortion (Baran et al., 2022).

The following subjects were selected as controversial ones in Polish society:

- Abortion: in Poland, it is only legal when pregnant woman's life is in danger or in case the pregnancy is a result of rape or incest. In 2021, it became illegal in case the fetus is severely malformed[2].
- Church: the Roman Catholic Church is the dominant religion in Poland.
- Constitution: recent constitution changes resulted in financial penalties from the European Union (EU)[3].
- Presidential elections: last ones took place in 2020[4].
- EU: Poland is a part of the EU. Citizens' attitude towards Poland's membership in the EU varies in time[5].
- Independence March: it is an annual event that takes place in Warsaw on November 11th to commemorate regaining independence by Poland[6].
- LGBT parade: the Equality Parade is an annual event that takes place in several cities in Poland to celebrate the LGBT community and promote tolerance and equality[7].
- Refugees: the Polish-Belarusian border has become a site of increased activity in recent years, as refugees and migrants try to enter the EU via Poland[8].

To select news articles on controversial topics, we filtered them using keywords that are listed in Table 1. We attempted to choose a non-biased wording. The keyword

[1] https://github.com/rafjaa/LeIA

[2] Dziennik Ustaw (Journal of Laws) Dz.U.2022.1575

[3] Order ECLI:EU:C:2021:878, Court of Justice of EU

[4] https://wybory.gov.pl/index

[5] https://www.cbos.pl/PL/trendy/trendy.php?trend_parametr=stosunek_do_integracji_UE

[6] https://marszniepodleglosci.pl/

[7] http://www.paradarownosci.eu/

[8] https://www.strazgraniczna.pl/pl/aktualnosci/9689,Nielegalne-przekroczenia-granicy-z-Bialorusia-w-2021-r.html

"strike" for abortion was chosen due to its connection to the Polish Women's Strike movement. To avoid repetitions of articles between categories, keywords from one subject were banned in others. Moreover, we limited website category to "Poland" to focus on matters within the country.

Topic	Keywords
Abortion	abortion, strike
Church	church, catholic, priest
Constitution	constitution, court, law, judge, tribunal
Presidential elections	elections
EU	union, european, cjeu
Independence March	independence, march, day
LGBT parade	lgbt, parade
Refugees	refugee, border, migrant

Table 1: Keywords used for selecting articles concerning each controversial topic

Table 2 shows the distribution of articles between topics. Analyses showed that the number of words in articles is resembling normal distribution. We decided to remove the outliers that were outside the three standard deviation distances from the mean length.

Topic	Number of articles	
	TVN	TVP
Abortion	62	143
Church	166	47
Constitution	147	92
Elections	234	205
EU	260	155
Independence March	46	86
LGBT parade	29	34
Refugees	124	147

Table 2: Number of articles in each topic per news outlet

4. Methods

This Section describes all methods of analyses performed in this work.

4.1. Lexicon-based classification with NAWL dictionary on articles covering controversial topics

In our first experiment sentiment analysis is conducted by matching words from selected articles, lemmatized with *SpacyPL*[9] library, with pre-defined NAWL sentiment dictionary consisting of 6 classes. All words within an article are a subject of the analysis. The sentiment scores are aggregated to determine the sentiment of each topic per news provider. Two-tailed t-test is performed to compare the results between news providers depending on the topic and check if the differences are statistically significant.

4.2. Zero-shot classification on NAWL dictionary

In the second step we perform zero-shot classification of NAWL dictionary words with XLM-RoBERTa Large model[10], fine-tuned on XNLI multilingual dataset[11]. We decided to use zero-shot approach as it yielded promising results for Slavic languages.

4.3. Zero-shot classification on articles covering controversial topics

Last experiment is a zero-shot classification on full article texts within each controversial topic. Six labels were provided in English, the same as in NAWL dictionary. Articles have been truncated to BERT-based models' maximum capacity of 512 tokens. Obtained results were used to perform two-tailed t-test for any statistically significant differences between groups.

5. Results

In this section we present obtained results of performed experiments.

5.1. Lexicon-based classification with NAWL dictionary on articles covering controversial topics

Results of lexicon-based sentiment analysis on the article content are shown in the Table 4.
T-test carried out on all emotions showed that at 5% level of significance, we do not reject the null hypothesis on no difference between means for any categories, meaning that the differences between groups proved to be insignificant.

5.2. Zero-shot classification on NAWL dictionary

Results of RoBERTa classification on NAWL dictionary is shown in Table 3.

Emotion	NAWL	RoBERTa-XNLI
Anger	13.26	7.58
Disgust	6.50	39.51
Fear	22.06	2.841
Happiness	19.89	4.33
Neutral	29.63	42.35
Sadness	8.66	3.38

Table 3: Percentage of words in each emotion category, based on NAWL dictionary and RoBERTa-XNLI classification

Percentages showed in the Table 3 indicate that RoBERTa based model differently classifies most of the groups. It is worth mentioning that there are more words classified by RoBERTa as "Neutral" than in the original dictionary. Moreover, there is also a noticeable difference in a number of words associated with emotions "Disgust", "Fear" and "Happiness". The first one is almost 8 times bigger than in the original dictionary. The other two are respectively almost eleven and five times smaller than the lexicon.

5.3. Zero-shot classification on articles covering controversial topics

Results of lexicon-based sentiment analysis on article content are shown in the Table 5.

[9] https://spacy.io/models/pl

[10] https://huggingface.co/xlm-roberta-large

[11] https://huggingface.co/joeddav/xlm-roberta-large-xnli

Emotion	Abortion		Church		Constitution		Elections		EU		Independence March		LGBT		Refugees	
	TVN	TVP	TVN	TVP	TVN	TVP	TVN	TVP	TVN	TVP	TVN	TVP	TVN	TVP	TVN	TVP
Anger	13.88	10.40	5.81	5.84	4.68	4.76	4.57	5.12	4.38	4.30	5.65	5.66	6.22	5.87	5.34	5.32
Disgust	1.37	0.63	1.39	1.47	0.15	0.11	0.17	0.51	0.09	0.09	0.23	0.21	2.58	0.26	0.37	1.06
Fear	34.22	34.74	29.55	29.29	64.39	64.97	49.56	45.40	64.38	61.98	27.30	31.08	30.90	25.77	20.21	23.71
Happiness	13.13	15.43	16.67	17.43	6.53	6.06	16.47	16.98	5.45	6.09	31.68	29.22	19.31	22.70	14.46	15.31
Neutral	33.92	36.11	40.54	39.97	23.12	22.93	26.63	29.04	24.06	25.90	29.15	29.48	35.19	41.07	56.17	51.14
Sadness	3.49	2.69	6.04	6.00	1.14	1.17	2.59	2.94	1.65	1.64	5.99	4.36	5.79	4.34	3.45	3.46

Table 4: Percentage of articles with dominant emotion per news outlet – based on NAWL classification

Emotion	Abortion		Church		Constitution		Elections		EU		Independence March		LGBT		Refugees	
	TVN	TVP	TVN	TVP	TVN	TVP	TVN	TVP	TVN	TVP	TVN	TVP	TVN	TVP	TVN	TVP
Anger	11.29	18.88	15.06	19.15	11.49	7.69	16.87	18.29	11.15	14.84	8.70	12.79	**3.45**	**20.59**	4.84	19.05
Disgust	11.29	16.08	10.84	10.64	7.43	12.09	7.23	10.98	8.46	14.19	6.52	15.12	**10.34**	**26.47**	11.29	14.29
Fear	30.65	30.77	29.52	25.53	**29.05**	**42.86**	34.94	35.37	27.69	29.68	36.96	38.37	**34.48**	**17.65**	33.87	36.05
Happiness	**16.13**	**5.59**	**17.47**	**8.51**	**21.62**	**13.19**	9.64	7.32	**26.54**	**11.61**	19.57	16.28	17.24	5.88	19.35	6.12
Neutral	17.74	14.69	15.66	14.89	15.54	13.19	19.28	17.07	16.92	14.19	**19.57**	**8.14**	20.69	26.47	18.55	11.56
Sadness	12.90	13.99	**11.45**	**21.28**	14.86	10.99	12.05	10.98	**9.23**	**15.48**	8.70	9.30	**13.79**	**2.94**	12.10	12.93

Table 5: Percentage of articles with dominant emotion per news outlet – based on XLM-RoBERTa Large fine-tuned on XNLI classification

After performing the t-test for all emotions, we noticed that at 5% level of significance, we reject the null hypothesis on no difference between means only for "Abortion" and "Church" categories. For better visibility of this outcome we display the counts of classified examples with the regard to news outlet for "Abortion" category on Figure 1.

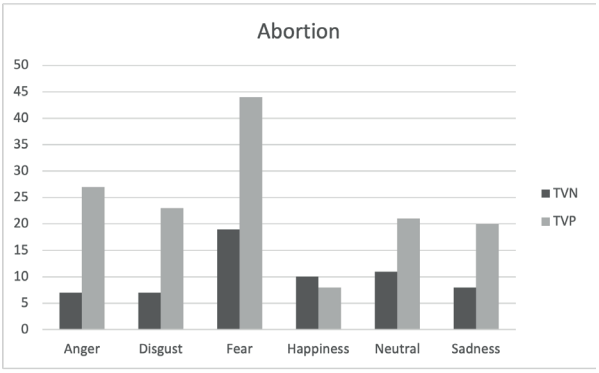


Figure 1: Number of articles classified by RoBERTa-XNLI from category "Abortion"

6. Discussion

Results of error analysis of the second experiment are showed in Table 6. Up to ten words with highest classification probability are displayed (their English translations are presented in brackets). There is a subtle difference in meaning between "Disgust" and "Anger" classes for words like foolishness or absurd. However, five misclassifications which occurred in "Happiness" category related words hold a completely opposite emotional appeal.

Additionally, a study on how to measure controversy is worth taking into consideration. Moreover, the differences in our results are caused by different level classification –

Emotion	Misclassified words
Disgust	głupota, przekleństwo, absurd, tort, malkontent, niezgodny, niewygodny, wścibski, szemrać, zniechecony (foolishness, curse, absurd, cake, malcontent, inconsistent, uncomfortable, nosy, murmur, discouraged)
Happiness	zadufany, moralista, współczuć, omen, lufcik (complacent, moralist, to sympathize, omen, backlash)

Table 6: Examples of words misclassified by RoBERTa-XNLI

word-level for lexicon-based method and article-level for language model. The length of the articles could have had an influence on results for the former method, in contrast to the latter one. Dictionaries are simpler and faster in use, but have problems with long texts and detection of negations, sarcasm or irony. Also, according to (Sun et al., 2019), in case of long texts which cannot be processed by BERT, we should consider truncating the articles so that only the beginning and the end are left.

7. Conclusion and Future Work

In this work we performed sentiment analysis on a subset of the Polish Online News Corpus which focused on controversial topics. We checked if there are any differences in sentiment of particular topics between news providers. For this task, we compared a lexicon-based and a statistical method of classification. Our work showed that there were no significant differences of emotional appeal on a word-level analysis. However, they were observed in case of results of zero-shot classification between analyzed news outlets for "Abortion" and "Church" categories. In the former topic, three times more words occurring in TVN articles were classified as happy in comparison to TVP. For the latter news category, a classifier showed that TVP had al-

most twice more words describing sadness and at the same time two times less expression of happiness compared with TVN.

Our work is a contribution to quantitative studies on sentiment analysis of Polish news articles. So far, most works were qualitative or based only on short pieces of texts such as tweets or opinions from rating websites, but not on online news. In the future we plan to take into account more controversial topics and analyze the dynamics of emotional load in time by considering publication dates of articles. In case of a lexicon-based approach, we intend to examine overall emotional load of the articles instead of focusing on a dominant sentiment and to run corresponding experiments on other dictionaries available in Polish. Moreover, we aim at classifying the emotional overtone of articles by human annotators. The dataset will be narrowed down to pairs of corresponding news from both media outlets and articles will be chunked into sentences to allow more accurate annotation. This will enable further comparison of zero-shot classification as well as other models results. Additionally, annotators will be asked to state whether selected sentences indicate political leaning. New sentence-level, annotated dataset will be used for research on bias and polarization studies of Polish media.

References

Aggarwal, Swati, Tushar Sinha, Yash Kukreti, and Siddarth Shikhar, 2020. Media bias detection and bias short term impact assessment. *Array*, 6:100025.

Baran, Joanna, Michał Kajstura, Maciej Ziolkowski, and Krzysztof Rajda, 2022. Does Twitter know your political views? POLiTweets dataset and semi-automatic method for political leaning discovery. In *Proceedings of the LREC 2022 workshop on Natural Language Processing for Political Sciences*.

Burbach, Laura, Patrick Halbach, Martina Ziefle, and André Calero Valdez, 2020. Opinion formation on the internet: The influence of personality, network structure, and content on sharing messages online. *Frontiers in Artificial Intelligence*, 3.

Catelli, Rosario, Serena Pelosi, and Massimo Esposito, 2022. Lexicon-based vs. BERT-based sentiment analysis: A comparative study in Italian. *Electronics*, 11:374.

Cui, Jingfeng, Zhaoxia Wang, Seng Ho, and Erik Cambria, 2023. Survey on sentiment analysis: evolution of research methods and topics. *Artificial intelligence review*:1–42.

Elejalde, Erick, Leo Ferres, and Eelco Herder, 2018. On the nature of real and perceived bias in the mainstream media. *PLoS ONE*, 13.

Kocoń, Jan, Maciej Piasecki, and Monika Zaśko-Zielińska, 2017. PlWordNet as a Basis for Large Emotive Lexicons of Polish. In *Human language technologies as a challenge for computer science and linguistics: 8th Language Technology Conference*.

Machete, Paul and Marita Turpin, 2020. The use of critical thinking to identify fake news: A systematic literature review. *Responsible Design, Implementation and Use of Information and Communication Technology*, 12067:235 – 246.

Mello, Caio, Gullal Cheema, and Gaurish Thakkar, 2022. Combining sentiment analysis classifiers to explore multilingual news articles covering London 2012 and Rio 2016 Olympics. *International Journal of Digital Humanities*:1–27.

Mohammad, Saif and Peter Turney, 2013. NRC emotion lexicon. *National Research Council of Canada*:1–234.

Mäntylä, Mika, Daniel Graziotin, and Miikka Kuutila, 2016. The evolution of sentiment analysis - a review of research topics, venues, and top cited papers. *Computer Science Review*, 27.

Nandwani, Pansy and Rupali Verma, 2021. A review on sentiment analysis and emotion detection from text. *Social Network Analysis and Mining*, 11.

Pelicon, Andraž, Marko Pranjić, Dragana Miljkovic, Blaž Škrlj, and Senja Pollak, 2020. Zero-shot learning for cross-lingual news sentiment classification. *Applied Sciences*, 10:5993.

Probierz, Eryka and Adam Galuszka, 2022. Emotion detection based on sentiment analysis: An example of a social robots on short and long texts conversation. *European Research Studies Journal*, XXV:135–144.

Riegel, Monika, Małgorzata Wierzba, Marek Wypych, Łukasz Żurawski, Katarzyna Jednoróg, Anna Grabowska, and Artur Marchewka, 2015. Nencki affective word list (NAWL): the cultural adaptation of the berlin affective word list–reloaded (BAWL-R) for Polish. *Behavior Research Methods*.

Shirsat, Vishal S., Rajkumar S. Jagdale, and S. N. Deshmukh, 2017. Document level sentiment analysis from news articles. In *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*.

Sun, Chi, Xipeng Qiu, Yige Xu, and Xuanjing Huang, 2019. How to fine-tune bert for text classification? In Maosong Sun, Xuanjing Huang, Heng Ji, Zhiyuan Liu, and Yang Liu (eds.), *Chinese Computational Linguistics*.

Szwoch, Joanna, Mateusz Staszkow, Rafal Rzepka, and Kenji Araki, 2022. Creation of Polish online news corpus for political polarization studies. In *Proceedings of the LREC 2022 workshop on Natural Language Processing for Political Sciences*.

Tomanek, Krzysztof, 2014. *Sentiment analysis: history and development of the method within CAQDAS (in Polish: Analiza sentymentu: historia i rozwój metody w ramach CAQDAS*.

Võ, Melissa, Markus Conrad, Lars Kuchinke, Karolina Urton, Markus Hofmann, and Arthur Jacobs, 2009. The berlin affective word list reloaded (bawl-r). *Behavior research methods*, 41:534–8.

Zhao, Lingyun, Lin Li, and Xinhao Zheng, 2020. A BERT based sentiment analysis and key entity detection approach for online financial texts. *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*:1233–1238.

Detection of depression on social networks using transformers and ensembles

Ilija Tavchioski[1], Marko Robnik-Šikonja[1], Senja Pollak[2]

[1] University of Ljubljana, Faculty of Computer and Information Sciences
Večna pot 113, 1000 Ljubljana, Slovenia
[2] Jozef Stefan Institute
Jamova Cesta 39, 1000 Ljubljana, Slovenia

Abstract

As the impact of technology on our lives is increasing, we witness increased use of social media that became an essential tool not only for communication but also for sharing information with community about our thoughts and feelings. This can be observed also for people with mental health disorders such as depression where they use social media for expressing their thoughts and asking for help. This opens a possibility to automatically process social media posts and detect signs of depression. We build several large pre-trained language model based classifiers for depression detection from social media posts. Besides fine-tuning BERT, RoBERTA, BERTweet, and mentalBERT were also construct two types of ensembles. We analyze the performance of our models on two data sets of posts from social platforms Reddit and Twitter, and investigate also the performance of transfer learning across the two data sets. The results show that transformer ensembles improve over the single transformer-based classifiers.

Keywords: Transformers, Depression detection, Ensembles

1. Introduction

Depression is one of the most widespread mental health disorders. According to the World Health Organization, it has the second highest number of affected people after anxiety, with 284 million cases worldwide (James et al., 2018). Depression is affecting people's behaviour, mood, and feelings, but also affects their productivity at work and their relationships. If left untreated the depression can lead to serious consequences such as suicide, which is the case for 800,000 people every year (Salas-Zárate et al., 2022). The bright side is the fact that the depression is treatable and its early detection is beneficial in the success of the treatment.

During the past decade, the use of social media rapidly grew, and social platforms digitized our society and human interactions. In many cases, people use social media to express their thoughts and feelings and also to share important moments of their lives. Marriott and Buchanan (Marriott and Buchanan, 2014) showed that the expression of a person's personality online is very similar or even identical to its expression offline, hence presenting a possibility to infer knowledge of people's personalities along with mental health-related issues.

We apply natural language processing (NLP) methods to social media posts, aiming to detect signs of depression and improve an early stage depression detection of depression, which is highly beneficial for its successful treatment. We build a number of transformer based classifiers and their ensembles and analyse classifiers' performance on two depression data sets. Besides direct assessment of depression in each dataset we also test cross-data set transfer.

The paper is structured into six sections. We present the related work in Section 2.. In Section 3., we describe the problem and datasets. The methodology is presented in Section 4., while the experiments results are described in Section 5.. Conclusions and plans for future work are presented in Section 6..

2. Related work

Several NLP researchers have tried to detect depression and related mental issues from social media, foremost from Twitter but also Reddit. (”Coppersmith et al., 2016) tried to detect suicide intentions in tweets using logistic regression with character n-gram features. (Almouzini et al., 2019) checked depression in both English and Arabic social media posts by extracting sparse features and constructing vector representations. They tested several classification methods such as Random Forest, Naive Bayes, AdaBoost and linear SVM. Later deep learning methods, such as bidirectional LSTM neural networks with attention (Zhou et al., 2016) produced better performance. For Reddit, (Trifan et al., 2020) proposed a SVM classifier with stochastic gradient descent using TF-IDF weighted feature vectors. Recently, the transformer-based architectures become the primary deep learning method for text analysis. (Sivamanikandan et al., 2022) used several transformer-based methods such as RoBERTa, ALBERT, and DistilBERT and achieved decent results on multi-class data sets where each class corresponds to the level of depression expressed in the post. Some researchers combined transformer-based methods with a combination of TF-IDF weighted vector representations and knowledge-graph based embeddings (Tavchioski et al., 2022a). A similar shared task aims to classify users as early as possible based on their history on Reddit (Parapar, 2022); one of the proposed approaches (Tavchioski et al., 2022b) constructed document embeddings using sentence-BERT (Reimers and Gurevych, 2019) and used logistic regression for classification.

3. Problem description and data sets

In this section, we define the depression detection problem and and two datasets used in our experiments along with their the distributions of class values (see Table 3.2. for Reddit and Table 3.2. for Twitter dataset).

3.1. Problem description

The problem is defined as follows. We are given a set of social media posts $D = \{d_1, d_2, ..., d_n\}$ along with their respective labels $L = \{l_1, l_2, ..., l_n\}$, where the labels correspond to the level of depression signs present in posts. The goal is to train a prediction model that will label new posts as accurately as possible.

3.2. Data description

We used two English datasets in experiments. The Reddit dataset (Kayalvizhi et al., 2022) is composed of posts from the Reddit social platform, mostly from subreddits like "r/stress", "r/loneliness", but also from "r/Anxiety", "r/depression" etc. The second dataset is composed mosty of posts from the social platform Twitter (Hu, 2021). The Twitter dataset contains short posts, some unrelated to depression. The Reddit dataset contains longer longer expressing persons' feelings in a deeper way. The Reddit posts were annotated by two domain specialists into three classes corresponding to the level of depression signs in the post.

- **Level 0 (Not depressed)** - there are no signs of depression in the post; the statements in the post are either irrelevant concerning depression or are related to giving help or motivation to people with depression.
- **Level 1 (Moderate)** - a post contains moderate signs of depression. These posts are related to change of feelings, but they show signs of improvement and hope.
- **Level 2 (Severe)** - these posts contain severe signs of depression. They are often related to serious suicide thoughts, disorder conditions or past suicide attempts.

Data set	Training	Validation	Test
Level 0	1659 (22%)	312 (23%)	2306 (51%)
Level 1	5140 (68%)	879 (66%)	1830 (41%)
Level 2	758 (10%)	143 (11%)	360 (8%)
Total	7557	1334	4496

Table 1: Reddit depression detection label distribution.

The Twitter data set contains four depression labels corresponding to different levels of depresion signs. *Level 0* corresponds to the *Level 0* from the previous dataset, i.e. Not depressed. *Level 2* and *Level 3* corresponds to *Level 1* and *Level 2* of the Reddit dataset, respectively, expressing Moderate and Severe depression. The Twitter dataset *Level 1* is similar to *Level 2* label, where statements reflect a slight change of person's feelings, but do not drastically affect the person's mood. We call this label *Change of feelings*.

Data set	Training	Validation	Test
Level 0	19095 (41%)	4118 (41%)	3989 (40%)
Level 1	2408 (5%)	507 (5%)	537 (5%)
Level 2	20069 (43%)	4272 (42%)	4328 (43%)
Level 3	6931 (15%)	1498 (14%)	1539 (15%)
Total	46167	10045	10016

Table 2: Twitter depression detection label distribution.

4. Depression prediction models

In this section, we present the constructed models, starting with baseline methods, and followed by transformer and ensemble models, We end the presentation with cross/domain transfer models.

4.1. Baseline methods

For comparison of our models' performances we apply three baseline models.

Majority classifier returns the most frequent label in the training set.

TF-IDF method uses the logistic regression classifier from *scikit-learn* (Pedregosa et al., 2011) library with the default parameters using TF-IDF weighted feature extracted from the posts.

Doc2Vec We used doc2vec method (Le and Mikolov, 2014) for construction of document embeddings for the posts and applied the logistic regression classifier with default parameters.

4.2. Transformer-based models

We tested several models based on large pre-trained language models similar to BERT (Devlin et al., 2019) with a *softmax* layer on top of the final hidden vector corresponding to the *CLS* token with *L* nodes, where *L* corresponds to the number of labels in the dataset. The last *softmax* layer returns label probabilities and the label with the highest probability is returned as prediction. The hyper-parameters used for fine-tuning were chosen based on the validation set. The loss function used was the cross-entropy. The details for each of the for BERT-like models are presented below.

BERT The BERT base model (Devlin et al., 2019) was pre-trained on the English Wikipedia (2,500 million words) and the BookCorpus (800 million words) (Zhu et al., 2015) on two tasks. In the masked language modelling task 15% of the words were masked and the goal was to predict them. The second pre-training task is the next sentence prediction where given two sentences the goal is to predict is the second is the successor of the first. The model is composed of 14 stacked encoder blocks with 12 self-attention heads and the the vector representation of 768 dimensions. The total number of parameters is 110 million. We used the model and its *tokenizer* (an algorithm for converting the text into a sequence of tokens) from the *HuggingFace* framework, named *bert-base-cased*.

RoBERTa (Liu et al., 2019) is a BERT-like model with 24 encoder blocks, with vector representation of 1024

and 16 self-attention heads, with a total of 355 million parameters. It is pre-trained on larger corpus; in addition to English Wikipedia and BookCorpus, the models was also pre-trained on data from CC-News, OpenWebText, and Stories. RoBERTa does not use the next sentence prediction task in pre-training and adjusts the first task with dynamic masking, where the masking pattern is different for each input sequence. The model was pre-trained for longer and with larger batch sizes. We used the *HuggingFace* implementation, named *roberta-base*.

mentalBERT While BERT and RoBERTa are general models, pre-trained on general texts, for depression detection we used the mentalBERT model (Ji et al., 2021) which may be better suitable, since it is additionally pre-trained on mental health related data from the Reddit social platform. The model has the same architecture as the BERT model and the same tasks for pre-training, but the authors adjusted the BERT model by continuing the pre-training process with the new data from Reddit. The pretraining data of this model does not overlap with our Reddit dataset. It was was collected from subreddits "r/SuicideWatch", "r/offmychest", "r/Anxiety", "r/mentalhealth", "r/bipolar", and, "r/mentalillness/", around two years prior to the collection of our Reddit dataset. The model was taken from *HuggingFace* where it is named *mental/mental-bert-base-uncased*.

BERTweet As we also use a dataset from the Twitter social network, we also tested the BERTweet model (Nguyen et al., 2020) which uses the same pre-training approach as RoBERTa and the same architecture as BERT. The main difference is that the model is pre-trained solely on Twitter data composed 850 million tweets in English. We used the *HuggingFace* [1] models named *vinai/bertweet-base*.

4.3. Transformer ensembles

We combined several standalone BERT-based models from above and formed different ensembles. We experimented with four combinations of baseline BERT models (G-general model, M-mentalBERT, T-BERTweet). As general models (pre-trained on general texts), i.e. RoBERTa and BERT, we selected the one with better performance which in most cases is RoBERTa. The combinations are named as follows.

- **GMT** - RoBERTa/BERT, mentalBERT and BERTweet.
- **GT** - RoBERTa/BERT and BERTweet.
- **GM** - RoBERTa/BERT and mentalBERT.
- **MT** - mentalBERT and BERTweet.

We combined the ensemble member predictions in two ways.

Averaging ensembles (AE) calculate the output as the average of the models' probabilities.

Bayesian ensembles (BE) (Pirš and Štrumbelj, 2019) applies Bayesian framework and outputs the label with the highest probability.

[1] https://huggingface.co/

4.4. Cross-data set transfer

We have datasets from two different platforms (Twitter and Reddit), with similar purpose but different in several aspects. To test the knowledge transfer between the datasets, we first fine-tuned BERT models on one data set, and then additionally on the second one, using the final model for the prediction. To align the labels in the datasets we merged two classes in the Twitter data set (*Label 1* and *Label 2*), since they are the most similar and only 5% of the data is labeled with *Label 1*.

5. Experiments and results

In this section, we first present the evaluation metrics and selection of hyper-parameters, followed by the results on the test sets. We used the validation set to select the best hyper-parameters. For each setting, we repeated the experiments three times and took the mean of the results.

5.1. Evaluation metrics

The following metrics were used:

- **Precision** - represents the percentage of correctly predicted instances out of all positive predictions.
- **Recall** - represents the percentage of correctly predicted instances out of all instances that were labeled as positive.
- **F1-score** is a harmonic mean of the precision and recall.

$$F_1 = \frac{2 \times precision \times recall}{precision + recall}$$

- **Accuracy** - represents the classification accuracy of predictions.

Since both problems are multi-class classification problems, we used weighted average of the scores for each class according to its number of instances.

5.2. Hyper-parameter selection

We tuned the hyper-parameters of all models on the validation set. We considered the following hyper-parameters.

- **Batch size (BS)** – $BS \in \{8, 16, 32\}$
- **Learning Rate (LR)** – $LR \in \{10^{-3}, 10^{-4}, 5 \cdot 10^{-5}, 10^{-5}\}$
- **Number of epochs (NE)** – $NE \in \{5, 10, 15\}$

The obtained hyper-parameters are presented in Table 5.2..

Finally, we fine-tuned the models using both the training and validation data and the same hyper-parameters were used for the transfer learning and ensembles.

5.3. Results

For each setting, we conducted two types of experiments, one without transfer learning from the other dataset (the first line for each model), and the other with transfer learning (the second line—in italics—for each model). Results are presented in Tables (5.3.1. and 5.3.2.), for Reddit and Twitter, respectively.

Model	Dataset	BS	NE	LR
BERT	Reddit	32	15	$5 \cdot 10^{-5}$
RoBERTa	Reddit	8	15	10^{-5}
mentalBERT	Reddit	32	15	10^{-4}
BERTweet	Reddit	16	15	$5 \cdot 10^{-5}$
BERT	Twitter	32	15	$5 \cdot 10^{-5}$
RoBERTa	Twitter	32	15	$5 \cdot 10^{-5}$
mentalBERT	Twitter	32	10	$5 \cdot 10^{-5}$
BERTweet	Twitter	16	5	$5 \cdot 10^{-5}$

Table 3: The values of hyper-parameters used for BERT models, based on the validation set. We tuned batch size (BS), learning rate (LR) and number of epoch (NE).

	Model	Acc	F1	SD
Baselines	majority	0.513	0.348	0.0000
	doc2vec	0.459	0.459	0.0051
	TF-IDF	0.519	0.576	0.0000
Transformers	mentalBERT	0.577	0.577	0.0050
		0.569	*0.565*	*0.0079*
	RoBERTa	0.557	0.563	0.0035
		0.532	*0.537*	*0.0110*
	BERT	0.559	0.561	0.0057
		0.564	*0.559*	*0.0051*
	BERTweet	0.560	0.561	0.0072
		0.557	*0.553*	*0.0068*
AE	GMT	**0.592**	**0.592**	**0.0024**
		0.586	*0.580*	*0.0040*
	GM	0.592	0.591	0.0031
		0.583	*0.575*	*0.0057*
	MT	0.584	0.579	0.0057
		0.579	*0.571*	*0.0093*
	GT	0.579	0.580	0.0030
		0.577	*0.569*	*0.0021*
BE	GMT	0.588	0.590	0.0037
		0.564	*0.567*	*0.0063*
	GM	0.545	0.556	0.0045
		0.496	*0.513*	*0.0209*
	MT	0.525	0.541	0.0125
		0.525	*0.537*	*0.0051*
	GT	0.568	0.571	0.0086
		0.565	*0.562*	*0.0069*

Table 4: Results of baseline models, standalone transformers, averaging ensembles (AE) and Bayesian ensembles (BE) on the Reddit dataset. For each setting, the results with transfer learning from Twitter data set are in *italics* and the results without transfer learning are without italics.

5.3.1. Reddit result

In Table 5.3.1., we present the results on the Reddit dataset introduced in Section 3.2.. Not surprisingly, the lowest performance methods were the baselines, the majority classifier, doc2vec and TF-IDF with logistic regression. The standalone transformers are much more successful, with mentalBERT (pretrained on Reddit data) being the best in this group, followed by RoBERTa, BERT and BERTweet. For majority of ensemble models, RoBERTa is used as the general model, except in the transfer learning experiments using averaging and Bayesian ensembles, where BERT performed better. For Bayesian ensembles, we can see that the GMT (RoBERTa, mentalBERT and BERTweet) and GT (RoBERTa and BERTweet) improved the results in comparison to the single BERT models, but the other combinations (MT (mentalBERT and BERTweet) and GM (RoBERTa and mentalBERT)) did not. Finally, the averaging ensembles were consistent in improving the results where the GMT (RoBERTa, mentalBERT and BERTweet) ensemble had the highest performance with F1-score of 0.592 and standard deviation of 0.0024. Example of the improvement can be the following Reddit post "*My dad had to explain to me that high school parties are a real thing. : I graduated 2 years ago and I genuinely thought high school parties were only from TV because I was never invited to or told about one. I'm not going to college either so to my knowledge, college parties don't exist either. I was going to commuter college but dropped out. I can't afford to go back for the forseeable future either. Social isolation is really doin it to me.*", where the RoBERTa model assessed the post with moderate level of depression while the ensemble GMT classified the post with the Level 0 which is the correct label. Regarding the transfer learning from the Twitter dataset, we can see that it did not lead to improved results. In general, the standard deviations are small, and the ensembles mostly further reduce the variance compared to standalone models.

The Reddit data set was part of the shared task (Kayalvizhi et al., 2022) organized by ACL and the best scores achieved in the competitions varied from 0.60 to 0.64 which testifies of decent performance our models exhibit. Note that the presented results here are based on the performance from the development set instead of the test set due to the missing gold labels of the unavailable official test set. The development and the test set were composed in a similar manner. Although many of the methods that were used are transformer-based, our ensemble methods introduce new combinations of transformer-based models such as combining mentalBERT and BERTweet. The code for the experiments can be find at `https://gitlab.com/teletton/diploma`.

5.3.2. Twitter results

Table 5.3.2. presents the results on the Twitter dataset. Similarly to Twitter, the baseline methods lag behind standalone transformers, and ensembles. In standalone BERT-like models, RoBERTa considerably outperformed other models, followed by BERTweet (pretrained on Twitter data), BERT and mentalBERT. The Bayesian ensembles show lower performance in comparison to the standalone BERT methods. As was the case in the Reddit dataset, the averaging ensembles showed the best performance with the GMT ensemble (RoBERTa, mentalBERT and BERTweet) giving the weighted F1-score of 0.859. Again in comparison with the RoBERTa for example "texas at night is creepy", the RoBERTa model asigned the Level 1 class while the GMT model assigned the Level 2 class which

	Model	Acc	F1	SD
Baselines	majority	0.416	0.245	0.0000
	doc2vec	0.683	0.680	0.0009
	TF-IDF	0.730	0.728	0.0000
Transformers	mentalBERT	0.831	0.831	0.0023
		0.848	*0.848*	*0.0004*
	RoBERTa	0.852	0.852	0.0009
		0.865	*0.866*	*0.0020*
	BERT	0.831	0.831	0.0008
		0.846	*0.846*	*0.0003*
	BERTweet	0.849	0.849	0.0008
		0.860	*0.860*	*0.0043*
AE	GMT	0.858	0.859	0.0018
		0.871	*0.871*	*0.0005*
	GM	0.851	0.851	0.0017
		0.857	*0.857*	*0.0024*
	MT	0.839	0.839	0.0029
		0.853	*0.853*	*0.0013*
	GT	0.858	0.858	0.0014
		0.873	***0.873***	***0.0005***
BE	GMT	0.849	0.849	0.0064
		0.857	*0.858*	*0.0025*
	GM	0.853	0.854	0.0003
		0.866	*0.866*	*0.0016*
	MT	0.835	0.835	0.0054
		0.849	*0.849*	*0.0005*
	GT	0.844	0.844	0.0020
		0.854	*0.854*	*0.0017*

Table 5: Results of baseline models, simple transformers, averaging ensembles (AE) and Bayesian ensembles (BE) for the Twitter data set. For each setting, the results with transfer learning from Twitter data set are in *italics* and the results without transfer learning are without italics.

is the correct one. In contrast to the previous dataset, the transfer learning did improve the results for almost all methods and the highest performing model was the GT (RoBERTa and BERTweet) averaging ensemble with transfer learning, obtaining the weighted F1-score of 0.873.

6. Conclusion and future work

The results show that fine-tuning large pre-trained language models can be successfully used to predict depression levels from social media. They outperform standard baselines such as majority classifier and logistic regression with doc2vec and TF-IDF weighted features. The use of models pre-trained on specific domain data turned out to be better compared to transfer from other domain for standalone BERT models. The mentalBERT model and BERTweet model were successful for datasets from the domain where they were pretrained, showing benefit of such domain specific pretraining.

The Bayesian ensembles did not surpass much simpler averaging ensembles and even, in some cases, produced lower results than standalone BERT models. This may be due to the fact that all members of ensembles were relatively similar and Bayesian ensembles did not get a chance to learn and exploit their differences.

We also note the difference in the performance of transfer learning in the results of the two data sets, where transfer learning improves the results on the Twitter data set but not on the Reddit dataset. One possible explanation may be that the Twitter data set is much larger and more general, which can have a strong impact on fine-tuning with the Reddit data set. On the other hand, the Reddit data set is much smaller with longer posts and can enrich the model with knowledge about different types of expressing feelings which can occur in tweets. The second explanation may be that due to merging of the labels, the models had easier task to determining the labels since there is one less class.

Our methods could be further improved in several ways. First, the input size is limited and the model truncates longer input posts which can lead to loss of information. In future work, we will consider the models with larger inputs such as Longformer (Beltagy et al., 2020). Also, we could try to enriching our set of ensemble models with other BERT-based models.

7. Acknowledgments

The research was supported by the Slovene Research Agency through research core funding no. P6-0411 and P2-103, as well as project no. J6-2581.

References

Almouzini, Salma, Maher Khemakhem, and Asem Alageel, 2019. Detecting arabic depressed users from twitter data. *Procedia Computer Science*, 163:257–265.

Beltagy, Iz, Matthew E. Peters, and Arman Cohan, 2020. Longformer: The long-document transformer.

"Coppersmith, Glen, Kim Ngo, Ryan Leary, and Anthony" Wood, 2016. "exploratory analysis of social media prior to a suicide attempt". In *"Proceedings of the Third Workshop on Computational Linguistics and Clinical Psychology"*.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics.

Hu, Nico, 2021. Depression social media dataset. Visited on 08/10/2022.

James, Spencer L, Degu Abate, Kalkidan Hassen Abate, Solomon M Abay, Cristiana Abbafati, Nooshin Abbasi, Hedayat Abbastabar, Foad Abd-Allah, Jemal Abdela, Ahmed Abdelalim, et al., 2018. Global, regional, and national incidence, prevalence, and years lived with disability for 354 diseases and injuries for 195 countries and territories, 1990–2017: a systematic analysis for the global burden of disease study 2017. *The Lancet*, 392(10159):1789–1858.

Ji, Shaoxiong, Tianlin Zhang, Luna Ansari, Jie Fu, Prayag Tiwari, and Erik Cambria, 2021. Mentalbert: Publicly available pretrained language models for mental healthcare.

Kayalvizhi, S., D. Thenmozhi, B. R. Chakravarthi, and Jerin Mahibha C., 2022. Findings of the shared task on Detecting Signs of Depression from Social Media. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*. Association for Computational Linguistics.

Le, Quoc V. and Tomás Mikolov, 2014. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, 2019. Roberta: A robustly optimized bert pretraining approach.

Marriott, Tamsin and Tom Buchanan, 2014. The true self online: Personality correlates of preference for self-expression online, and observer ratings of personality online and offline. *Computers in Human Behavior*, 32:171–177.

Nguyen, Dat Quoc, Thanh Vu, and Anh Tuan Nguyen, 2020. Bertweet: A pre-trained language model for english tweets.

Parapar, Javier, 2022. erisk 2022: Early risk prediction on the internet. Visited on 15/17/2022.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pirš, Gregor and Erik Štrumbelj, 2019. Bayesian combination of probabilistic classifiers using multivariate normal mixtures. *Journal of Machine Learning Research*, 20(51):1–18.

Reimers, Nils and Iryna Gurevych, 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Salas-Zárate, Rafael, Giner Alor-Hernández, María del Pilar Salas-Zárate, Mario Andrés Paredes-Valverde, Maritza Bustos-López, and José Luis Sánchez-Cervantes, 2022. Detecting depression signs on social media: a systematic literature review. In *Healthcare*, volume 10.

Sivamanikandan, S, V Santhosh, N Sanjaykumar, C Jerin Mahibha, and Thenmozhi Durairaj, 2022. scubeMSEC@LT-EDI-ACL2022: Detection of depression using transformer models. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*.

Tavchioski, Ilija, Boshko Koloski, Blaž Škrlj, and Senja Pollak, 2022a. E8-ijs@ lt-edi-acl2022-bert, automl and knowledge-graph backed detection of depression. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*.

Tavchioski, Ilija, Blaž Škrlj, Senja Pollak, and Boshko Koloski, 2022b. Early detection of depression with linear models using hand-crafted and contextual features. *Working Notes of CLEF*:5–8.

Trifan, Alina, Rui Antunes, Sérgio Matos, and Jose Luís Oliveira, 2020. Understanding depression from psycholinguistic patterns in social media texts. In *European Conference on Information Retrieval*. Springer.

Zhou, Peng, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu, 2016. Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling.

Zhu, Yukun, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler, 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision (ICCV)*.

New Bulgarian Resources for Studying Deception and Detecting Disinformation

Irina Temnikova[1], Silvia Gargova[1], Ruslana Margova[1], Veneta Kireva[1], Ivo Dzhumerov[2], Tsvetelina Stefanova[1] and Hristiana Krasteva [2]

[1]Disinformation Detection Research Group, Big Data for Smart Society Institute (GATE), Sofia, Bulgaria

[2]University of Plovdiv "Paisii Hilendarski"

irina.temnikova@gate-ai.eu, silvia.gargova@gate-ai.eu, ruslana.margova@gate-ai.eu, venkireva80@gmail.com, i.dzhumerov@gmail.com, tsveti.stefanov@gmail.com, hnikolaeva@uni-plovdiv.bg

Abstract

Automatically detecting disinformation is an important Natural Language Processing (NLP) task whose results can assist journalists and the general public. The European Commission defines "disinformation" as "false or misleading content that is spread with an intention to deceive". Deception and thus disinformation can be identified by the presence of (psycho)linguistic markers, but some lower-resourced languages (e.g. Bulgarian) lack sufficient linguistic and psycholinguistic research on this topic, lists of such markers and suitable datasets. This article introduces the first ever resources for studying and detecting deception and disinformation in Bulgarian (some of which can be adapted to other languages). The resources can benefit linguists, psycholinguists and NLP researchers, are accessible on Zenodo (subject to legal conditions) and include: 1) an extended hierarchical classification of linguistic markers signalling deception; 2) lists of Bulgarian expressions for recognizing some of the linguistic markers; 3) four large Bulgarian social media datasets on topics related to deception, not fact-checked, but automatically annotated with the markers; 4) Python scripts to automatically collect, clean, anonymize, and annotate new Bulgarian texts. The datasets can be used to build machine learning methods or study potential deception. The article describes the methods of collecting and processing the datasets and linguistic markers, and presents some statistics.

Keywords: datasets, resources, disinformation, deception, Bulgarian

1. Introduction

Nowadays, detecting untrue content becomes an increasingly important task for the general public, journalists, and Natural Language Processing (NLP) researchers. Untrue textual content includes texts written by humans with the intention to mislead or unknowingly spread untrue statements, as well as automatically generated false texts commonly known as "textual deepfakes". Although there has been considerable research on detecting untrue (also called "fake") texts in English (e.g. Pérez-Rosas et al., 2017; Alam et al., 2021), significantly less attention has been given to other languages, particularly the under-resourced ones. One specific type of untrue texts, written by humans, is *disinformation*. The Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee, and the Committee of the Regions on the European Democracy Action Plan[1] defines *disinformation* as "false or misleading content that is spread *with an intention to deceive* or secure economic or political gain and which may cause public harm". This *intention to deceive* distinguishes disinformation from misinformation (unintentionally spread false information) and makes it a type of deception. Recent existing NLP research on disinformation does not take into consideration the intention to deceive, but only the fakeness of the texts and the potential harm they may cause (Alam et al., 2021). The fakeness of the text is determined most often on the basis of comparison with fact-checked claims, stored in special international or local websites and databases, or by comparing the text's claims with specific ontologies. However, untrue statements that are not covered by fact-checking databases or that cannot be checked against existing ontologies (for example if such ontologies do not exist) may remain undetected. One alternative method to identify untrue statements is to recognize textual disinformation by the specific linguistic markers of deception it may contain.

While there is a lot of research on the linguistic and psycholinguistic characteristics of deceitful language (e.g. Zhou et al., 2004; Vrij, 2000), most of it is about English. There are also NLP methods using linguistic and psycholinguistic features, among many others, to detect fake content (e.g. Pérez-Rosas et al., 2017). Most often (Pérez-Rosas et al., 2017; Shrestha et al., 2020) such features are taken from the Linguistic Inquiry and Word Count (LIWC[2]) dictionaries and tools, which are available for a limited number of languages and do not include Bulgarian. Additionally, the most recent LIWC versions are not available for free.

This article addresses several gaps in Bulgarian, such as the absence of deception, disinformation, and textual deepfakes datasets, and of linguistic and psycholinguistic resources and studies on the Bulgarian markers of deception. It also contributes to the very limited research on the language of deceit in Bulgarian (Kitanova, 2019). While the messages included in these datasets have not been fact-checked, they are a much larger alternative (over 118,000

[1] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM:2020:790:FIN&qid=1607079 662423 .

[2] https://www.liwc.app/.

messages) to the existing Bulgarian social media datasets for detecting fake texts. The latter (Nakov et al., 2021) currently contain up to 3,700 messages, with very few of them annotated as containing fake information.

The present version of the new resources includes: 1) an extended (mostly language-independent) hierarchical classification of linguistic markers of deception, with three markers that may signal deepfake texts; 2) 38 thematically divided lists with a total of 618 Bulgarian expressions for matching some of the marker categories in texts; 3) four social media datasets (three from Twitter and one from Telegram) consisting of 118,570 messages mostly on the topics of Covid-19 and deception, not fact-checked, but automatically annotated with marker categories and expression lists; and 4) over 20 well-documented Python scripts for annotating other Bulgarian datasets. The social media datasets were collected via keywords search (from Twitter) and from public groups and channels on specific topics (from Telegram). Some of the keywords were taken from famous controversial Bulgarian cases involving accusations of deception, and the messages collected in this way can thus be used for linguistic studies of social media texts on these topics. The resources could also have contributions for other languages - the expression lists could be translated, and the hierarchical classification and Python scripts could be applied to other datasets parsed with the Python packages *CLASSLA*[3] or *Stanza*[4] after language-specific adaptations. The next sections provide details about: the related research on fake content and deception detection in Bulgarian and other languages (Section 2), the linguistic markers of deception (Section 3), the methods of collection, preprocessing, and statistics of the social media datasets (Section 4), and the Python scripts (Section 5). Section 6 explains the legal restrictions of access to the resources, Section 7 discusses their limitations, and Section 8 provides the conclusions.

2. Related work

Since this article concerns detecting disinformation as a type of deception, the research works most closely related to ours are Bulgarian-language datasets on the topic of fake text detection, as well as datasets and language resources (such as word lists and lexicons) in other languages that were developed for detecting deception. While there are other important topics related to the detection of fake content, such as propaganda, bias, and framing, the definitions of these terms only partially overlap with the European Commission's definition of "disinformation". As they are not directly relevant to the focus of this article, they will not be discussed in this section.

The existing Bulgarian-language social media datasets that were built for detecting fake texts include the Twitter datasets from the 2021 Workshop on NLP for Internet Freedom (NLP4IF)[5] of approximately 3700 tweets (with very few of them annotated as "fake"), which completely overlap with the CLEF2021 CheckThat! Lab dataset (Nakov et al., 2021) and mostly with the CLEF2022 CheckThat! Lab dataset[6]. Additionally, there is the PHEME[7] project dataset that focused on rumours about the Bulgarian bank crisis, which includes 952 tweets. Apart from these, there are two news article datasets - the first includes 481 news articles that were annotated for toxicity (Dinkov et al., 2019), and the second includes almost 12,300 news articles, out of which 6,382 are fake stories (Hardalov et al., 2016). However, neither of these datasets is annotated with linguistic markers of deception.

There are deception detection datasets (frequently multimodal) for several languages, including English (Pérez-Rosas et al., 2015, de Ruiter and Kachergis, 2018), Polish (Rubikowski and Wawer, 2013), Chinese (Zhou and Sung, 2008), Italian (Fornaciari and Poesio, 2012), and Russian (Litvinova et al., 2017). These datasets contain texts collected from online games (de Ruiter and Kachergis, 2018), fake reviews (Rubikowski and Wawer, 2013), or court proceedings (Fornaciari and Poesio, 2012, Pérez-Rosas et al., 2015). Some of these datasets have the advantage of containing texts that are most likely to be truly deceptive (such as those from court proceedings). However, there is currently no such dataset available for Bulgarian, and it can be challenging to find truly deceptive texts.

Finally, there are some studies on the specific linguistic characteristics of Bulgarian fake news, including two classes of grammatical constructions (Getsov, 2018; Margova, 2022), as well as specific lexical expressions (Eftimova, 2018; Margova, 2021). However, to our knowledge, there are no other collections of linguistic markers focused on detecting deception in Bulgarian.

3. Linguistic markers signalling deception

The linguistic markers of deception have been collected on the basis of extensive literature review (Rubin et al., 2017; Ekman, 1985; Zhou et al., 2004; Addawood et al., 2019; Bond and Lee, 2005; etc.) mainly from studies on deception in English, as well as research on the linguistic characteristics of fake news in Bulgarian. There are also three language markers found to often signal texts generated by GPT-2, GPT-3[8] and Megatron (Xu et al., 2020) (for deepfake detection). Over 300 markers were originally collected, manually refined, and grouped into 97 fine-grained and 18 coarse-grained categories. The resulting classification, available in an .xlsx format, is somewhat similar to the one proposed by Zhou et al. (2004) and adopts some of their categories. Some (coarser-grained) categories are accompanied by possible methods of automatic detection, reflecting the characteristics of the Bulgarian language, and sometimes include a reference to specially created expression lists. The version of the datasets introduced in this article has been annotated with only a subset of the categories for three reasons: 1) some markers require NLP tools or language resources that do not exist for

[3] https://pypi.org/project/classla/.

[4] https://stanfordnlp.github.io/stanza/.

[5] http://www.netcopia.net/nlp4if/2021/index.html.

[6] https://sites.google.com/view/clef2022-checkthat.

[7] https://www.pheme.eu/.

[8] https://www.sigmoid.com/blogs/gpt-3-all-you-need-to-know-about-the-ai-language-model/.

Coarse-grained category	Fine-grained categories and Word lists for look-up
1. Sentiment/emotions/affect	1.1. Negative emotions; 1.2. Positive emotions; 1.3. Lack of emotions; 1.4. Inciting fear; 1.5. Replace the polarity sign of emotions; **1.6. Attention-attracting expressions (list of)**
2. Negation	**Count of negative expressions**
3.Vocabulary/expressivity	**3.1. Lexical diversity/richness(type/token ratio; content words diversity); 3.2. Emotiveness/expressivity (Zhou et al. (2004)); 3.3. Redundancy (Zhou et al. (2004);** 3.4. Simple vocabulary
4. Perceptual information	**4.1. Visual inform.; 4.2. Sounds; 4.3. Smells; 4.4. Taste; 4.5. Tactile inform.; 4.6. Physical sensations - Lists of verbs and expressions (4.1-4.6)**
5. Unbalanced/Lack of perceptual information	**5.1. Temporal information (list of Temporal expressions); 5.2. (List of) Spatial information; 5.3. Causation (list of Cause-effect expressions); 5.4. Numbers**
6. Senseless or overbalanced text	6.1. Too many/irrelevant details; 6.2. Unnecessary connections
7. Details/specificity	7.1. Lack of details/memory; 7.2. Missing links; 7.3. Avoidance of discrediting information (also lack of details); **7.4. Lack of past details (past tense verbs)**
8. Grammaticality and fluency	8.1. Coherence; 8.2. Syntax; 8.3. Ungrammatical/ linguistically incompetent
9. Length and complexity	**9.1. Sentence length; 9.2. Text/message length**; 9.3. Cognitive complexity; 9.4. Reticent (less talking, but also linked to fewer details)
10. Cognitive operations	**List of cognitive operations verbs**
11. Clarity	11.1. Ambiguity (lexical, syntactic), Unclarity; 11.2. Using expert terms; 11.3. Contradiction; **11.4. List of Generalization words** (also linked to distance)
12. Nervousness (betraying emotions)	12.1. Arousal; 12.2. Nervousness/tension
13. Depersonalisation/distance (also "non-immediacy")	**13.1. Self-references; 13.2. Group references; 13.3. Use of pronouns; 13.4. Passive voice** (also sign of complexity); 13.5. Formal speech; 13.6. Automatic and template phrasing; **13.7. List of volitional words**
14. Plausibility, credibility	**Lists of words: Verbs for hesitation/doubt; Expressions for doubt, unclear quantity/content; Verbs for confidence; Expressions of confidence**
15. Subjectivity	**Subjectivity expressions in the word lists**
16. Untypical way of writing for that specific person	Needs analysis of several messages of the same author.
17. Specific Part-of-speech	**17.1. Number of verbs; 17.2. Number of Nouns and Noun Phrases; 17.3. Number of modifiers**
18. Social media specific	**18.1. Hashtags; 18.2. Mentions; 18.3 Links**

Table 1: Coarse-grained, fine-grained categories and methods of their detection.

Bulgarian; 2) such tools or resources exist, but their quality is too low; 3) some categories of markers require a different type of data. For example, Category 16 requires several messages from the same author, whereas the current datasets contain separate messages from many different authors. Such data collection was motivated by our desire to prevent author identity reconstruction in accordance with European laws. The methods used to annotate the current datasets mostly involve calculating text length normalised counts of various linguistic elements (e.g. number of passives, adjectives, or locations) in texts processed by a syntactic parser and a Named Entity Recognizer (NER). For some categories (such as *perceptual*, *temporal*, and *spatial information*) the method includes performing an automatic look-up for specific Bulgarian expressions, contained in the specially created expression lists. These lists of expressions (e.g. verbs, synonyms of the verb "to see", sensation expressions used for "feeling cold", "warmth", and different kinds of smells) have been collected by Bulgarian linguists after a search revealed that there was no existing resource containing all the necessary expressions. An additional list of "attention-attracting" expressions has also been compiled. These expressions were manually identified by a Bulgarian linguist in tweets related to Covid-19. Some examples include: "обрат" (in English: "sudden change"), "ВНИМАНИЕ, ВНИМАНИЕ" (in English: "ATTENTION, ATTENTION"), "Тревога" (in English: "Alarm"), "Важно съобщение" (in English: "Important message"). Table 1 shows the 18 coarse grained categories, their fine grained subcategories, and the respective expression lists. The categories shown in **bold** are those that have been annotated in the four datasets (explanations can be found in Section 4). The final resource contains 38 lists with a total of 618 expressions.

4. Social media datasets

There are four datasets - three from Twitter (separated by topic) and one from Telegram. All the datasets have been cleaned, language-filtered, and anonymized. The Twitter datasets were collected via the Twitter API with academic access and thus cannot be used for commercial purposes. The three Twitter topics are: Covid-19, general lies and manipulation, and 17 famous Bulgarian political cases, in which politicians were accused (and sometimes proven) of lying. The tweets were collected between 1 January 2020 and 30 June 2022. Several keyword combinations in Bulgarian, mostly using Cyrillic letters, were used for each of the Twitter datasets, and are also shared with the set of resources (see Table 2 for examples).

The Telegram dataset was collected from nine public groups and channels, using the Telegram desktop version after checking the Telegram requirements in terms of data collection and dataset release. The topics are Covid-19, politics, and general news, and the time period is from 1 January 2021 to 30 June 2022.

Platform, topic	Keywords examples
Twitter **topic**: lies, deception, manipulation, deceivers	(лъжа OR лъжи OR лицемерие OR лъжат OR излъга OR измама OR измамници OR измами OR лъжец OR лъжци) (фалшиви OR fakenews OR невярно OR неверни OR подвеждащи OR подвеждащо OR неистини) lang:bg -is:retweet (Манипулация OR манипулира OR стъкмистика OR крие OR далавераджия OR далавери OR далавера) lang:bg -is:retweet"
Twitter **topic**: vaccinated members of Parliament; guest houses; dams	(ваксиниран депутат) OR (ваксинирани депутати) lang:bg -is:retweet (язовири премиер) OR (язовири прокуратура) OR (язовири прокуратурата) lang:bg -is:retweet (апартаментгейт OR (къща за гости) OR (къщи за гости)) lang:bg -is:retweet
Twitter **topic**: coronavirus, pandemic	Covid OR коронавирус OR Covid19 OR Covid-19 OR Covid_19 ... -is:reply -is:retweet (Корона OR корона OR Corona OR пандемия OR пандемията OR Spikevax OR SARS-CoV-2 OR бустерна доза) lang:bg isreply -is:retweet

Table 2: Twitter keywords examples.

All datasets were then cleaned of duplicates, and non-Bulgarian messages were removed using FastText[9] - previously determined to be the best tool for Bulgarian language identification for our datasets (Gargova et al., 2022). Table 3 shows the final number of messages, which were left in the datasets after cleaning and language filtering. The datasets were subsequently anonymized in several automatic (with ad-hoc Python scripts) and manual review rounds to comply with the General Data Protection Regulation (GDPR) and prevent user identities from being reconstructed. All data collection, preprocessing, and release processes were conducted in active consultations with lawyers specialised in Bulgarian and European law.

Platform, topics	Num. of soc. media msgs
Twitter - lies and manipulation (general) **Time period:** 1 January 2020 – 30 June 2022	32518
Twitter - 17 famous cases of suspected lies from Bulgarian media **Time period**: 1 January 2020 – 30 June 2022	15850
Twitter - Covid-19 **Time period:** 1 January 2020 – 30 June 2022	61411
Telegram **Topics:** Covid-19; 1 Bulgarian political party; 1 news media **Time period**: 1 January 2021 – 30 June 2022	8791

Table 3: Dataset details.

Finally, the datasets were automatically annotated with most of the categories (shown in **bold** in Table 1) of the markers of deception, described in Section 3, and by using the Python scripts, described in Section 5. As most categories represent counts of items, we normalized them to account for message length differences. This was achieved by dividing the counts by the message length, which was determined by the number of words in each message. Figure 1 shows an example of a message with some of its annotations and its translation into English in a mock format. The original format of the annotated messages is horizontal, in a spreadsheet with many columns, which is thus inappropriate for this article.

In order to detect most of the annotated categories of markers, the texts were syntactically parsed with *CLASSLA,* and the named entities were annotated with the Bulgarian *CLASSLA* and *Slavic BERT*[10] NERs*,* which produced the best, but slightly different, results. We plan to continue to update the datasets with annotations of new categories, when NLP tools of satisfactory quality appear.

The datasets are supplied with detailed information, which describes the contents of each column of values. Many of these values (e.g. message and sentence lengths, number of passives, number of specific Part-Of-Speech (POS) and Named Entities (NE) tags) can also be used for general research purposes, other than for deception detection.

Text	"Covid-19 не е особено опасна инфекция и никога не е била третирана като такава…" К. Ангелов 28.04.2021 г. English: *"Covid-19 is not a particularly dangerous infection and has never been treated as such…" K. Angelov, 28.04.2021.*	
Markers	**words_message**	18
	negation_norm_count	0,111111111111111
	count_upos_all	{'PROPN': 0.16666666666666666, 'ADP': 0.05555555555555555, 'NOUN': 0.1111111111111111, 'ADV': 0.1111111111111111, 'ADJ': 0.1111111111111111, 'VERB': 0.05555555555555555, 'CCONJ': 0.05555555555555555, 'AUX': 0.16666666666666666, 'DET': 0.05555555555555555, 'X': 0.0, 'PRON': 0.0, 'NUM': 0.0, 'PART': 0.1111111111111111, 'INTJ': 0.0, 'SCONJ': 0.0, 'PUNCT': 0.2222222222222222}
	passive_voice_count	0,0769230769230769
	ner_types_count_bert	{'PER': 0.05555555555555555, 'LOC': 0.0, 'ORG': 0.0, 'PRO': 0.05555555555555555, 'EVT': 0.0}
	ner_types_count_classla	{'PER': 0.05555555555555555, 'LOC': 0.0, 'ORG': 0.0, 'OTH': 0.0}

Figure 1: Example of an annotated message.

5. Python scripts

The resources also contain over twenty well-documented Python scripts, which can be used for cleaning, anonymizing, and annotating new datasets in Bulgarian with the linguistic markers, provided that the texts have been parsed with *CLASSLA*. *CLASSLA* is a fork of the Python NLP package *Stanza*, which supports 66 languages and has been adapted to Slovenian, Croatian, Serbian, Macedonian, and Bulgarian. The Python scripts can also be used to annotate datasets in other languages supported by *CLASSLA* or *Stanza*, provided that the necessary language-specific adaptations to the scripts and lists of expressions are made.

[9] https://fasttext.cc/docs/en/language-identification.html.

[10] https://github.com/deeppavlov/Slavic-BERT-NER.

6. Access to the resources and legal and ethical considerations

The resources can be accessed on the public platform Zenodo[11] via *restricted access,* subject to users agreeing to detailed legal conditions. The conditions are in line with the TRACES project's Data Management Plan[12] and were drafted after consultations with lawyers in accordance with several European laws (including the GDPR, the Artificial Intelligence Act, and the Charter of Fundamental Rights of the European Union). For example, the resources cannot be used for user profiling, in court proceedings, or for government surveillance. Additionally, the Twitter datasets cannot be used for commercial purposes.

Table 4 lists the Zenodo links where the resources can be found.

Resource	Link
Hierarchical classification of deception markers and expression lists	https://zenodo.org/record/7656905
Python scripts	https://zenodo.org/record/7657029
Datasets	
Twitter dataset on Famous Bulgarian Political Cases of Suspected Lies	https://zenodo.org/record/7614357
Twitter dataset on Covid-19	https://zenodo.org/record/7614247
Twitter dataset on general lies and manipulation	https://zenodo.org/record/7614318
Telegram dataset	https://zenodo.org/record/7614294

Table 4: Zenodo links to the resources.

The research described in this article follows the Ethical Code of the authors' host institution, the Big Data for Smart Society Institute (GATE)[13].

7. Limitations

While providing several contributions, the resources introduced in this article have the following limitations:

- The linguistic markers (such as the number of passives, and adjectives) can characterise any text. Only in specific amounts and combinations, which also vary by text type, can they signal potential deception. For more information on how to use the markers to identify deception, please consult the instructions on Zenodo.
- The social media messages collected with keywords related to 'lies', 'manipulation', and famous Bulgarian cases of suspected deception may not contain deception but merely discuss such cases. To address this issue, a new version of the datasets has been created (see below for details).
- The social media messages in the datasets presented in this article have not been checked for truthfulness and disinformation. However, a subset of the datasets was subsequently manually annotated for truthfulness and disinformation by Bulgarian journalists and will be published in future work.
- It is possible that the automatic annotations contain some errors due to the errors produced by the NLP tools used for pre-processing the datasets (*CLASSLA* and *Slavic Bert NER*). No manual evaluation of the annotations was done due to the size of the datasets.

8. Conclusions

This article presents the first version of a new set of resources that fills several gaps in different research fields (linguistics, psycholinguistics, and NLP) for Bulgarian. The resources can be used to study and automatically detect deception and disinformation in Bulgarian through linguistic and psycholinguistic markers of deception. They also provide a larger alternative to the currently existing Bulgarian social media datasets for fake news detection. Some of the resources can be adapted to other languages. The resources are released for free, but under specific legal conditions. In future work, we will publish a subset of these datasets labelled with truthfulness and disinformation by journalists, and we will continue to annotate the datasets with new categories. Additionally, the resources will be uploaded to other platforms to increase their reusability.

Acknowledgements

The resources presented in this article, as part of the research project TRACES[14], have indirectly received funding from the European Union's Horizon 2020 research and innovation action programme, via the AI4Media Open Call #1 issued and executed under the AI4Media project (Grant Agreement no. 951911). The article reflects only the authors' view. The work has been also supported by the GATE project, funded by Operational Programme Science and Education for Smart Growth under Grant Agreement No. BG05M2OP001-1.003-0002- C01.

The authors are grateful to the multiple LTC'23 reviewers, to Alison Carminke for proofreading the article, and to Milena Dobreva for providing comments.

References

Addawood, A., Badawy, A., Lerman, K., Ferrara, E. (2019). Linguistic cues to deception: Identifying political trolls on social media. In *Proceedings of the international AAAI conference on web and social media* (Vol. 13, pp. 15-25).

Alam, F., Cresci, S., Chakraborty, T., Silvestri, F., Dimitrov, D., Martino, G.D.S., Shaar, S., Firooz, H., Nakov, P., (2021). A survey on multimodal disinformation detection. *arXiv preprint arXiv:2103.12541*.

Bond, G. D., and Lee, A. Y. (2005). Language of lies in prison: Linguistic classification of prisoners' truthful and deceptive natural language. *Applied Cognitive Psychology*, *19*(3), 313-329.

[11] The full list of Zenodo links can be accessed on the project's website (https://traces.gate-ai.eu/), or by contacting the first two authors.

[12] https://traces.gate-ai.eu/?p=611.

[13] https://gate-ai.eu/.

[14] https://traces.gate-ai.eu/.

de Ruiter, B. and Kachergis, G. (2018). The mafiascum dataset: A large text corpus for deception detection. arXiv preprint arXiv:1811.07851.

Dinkov, Y., Koychev, I., Nakov, P. (2019). Detecting toxicity in news articles: Application to Bulgarian. *arXiv preprint arXiv:1908.09785.*

Ekman, P. (1985). Telling lies: Clues to deceit in the marketplace, marriage, and politics. New York, NY: Norton.

Eftimova, A. (2018). Lexical means which indicate the (in)credibility of media contents (a psycholinguistic experiment). In: *Език и литература*, 2018 - vol. 1-2, pp. 143-153.

Fornaciari, T. and Poesio, M. (2012). Decour: a corpus of deceptive statements in Italian courts. In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12), pp. 1585-1590.

Gargova, S., Temnikova, I., Dzhumerov, I., & Nikolaeva, H. (2022). Evaluation of Off-the-Shelf Language Identification Tools on Bulgarian Social Media Posts. In Proceedings of the 5th International Conference on Computational Linguistics in Bulgaria (CLIB 2022), pp. 152-161.

Getsov, A. (2018). Appositive Word Groups as an Instrument for Creating Fake News. In: *21st Century Media and Communications*, 2018 - vol. 2, issue 1, pp. 80-84.

Hardalov, M., Koychev, I., Nakov, P. (2016). In search of credible news. In I*nternational conference on Artificial intelligence: methodology, systems, and applications,* pp. 172-180. Springer.

Kitanova, M. (2019). Axiology of lies in Bulgarian language. In *Axiological Investigation into the Slavic Languages* (pp. 120-132).

Litvinova, T., Ryzhkova, E., Litvinova, O., Larin, E., Lyell, J., Seredin, P. (2017). Building a corpus of "real" texts for deception detection. In *Proceedings of the International Conference IMS-2017,* pp. 110-115.

Margova, R. (2021). Here is where* fake news can hide. Contemporary Linguistics, no. 2, 109-118, "St. Kliment Ohridski" University Publishing House, Sofia.

Margova, R. (2022). Problems with the renarrative and the perfect for constatation in press and online journalism. International Jubilee Conference of the Institute for Bulgarian Language "Prof. Lyubomir Andreychin" (Bulgarian Academy of Sciences). Sofia, Bulgaria.

Nakov, P., Alam, F., Shaar, S., Da San Martino, G., Zhang, Y. (2021). COVID-19 in Bulgarian social media: Factuality, harmfulness, propaganda, and framing. *International Conference on Recent Advances in Natural Language Processing* (RANLP 2021), pp. 997-1009.

Pérez-Rosas, V., Abouelenien, M., Mihalcea, R., Burzo, M. (2015). Deception detection using real-life trial data. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pp. 59-66.

Pérez-Rosas, V., Kleinberg, B., Lefevre, A., & Mihalcea, R. (2017). Automatic detection of fake news. *arXiv preprint arXiv:1708.07104*.

Rubikowski, M. and Wawer, A. (2013). The scent of deception: Recognizing fake perfume reviews in Polish. In *Intelligent Information Systems Symposium,* pp. 45-49. Springer, Berlin, Heidelberg.

Rubin, V. L. (2017). Deception detection and rumor debunking for social media. In *The SAGE handbook of social media research methods*, p. 342. Sage.

Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H. (2017). "Fake news detection on social media: A data mining perspective." ACM SIGKDD explorations newsletter 19.1, pp. 22-36.

Shrestha, A., Spezzano, F., Joy, A. (2020). Detecting fake news spreaders in social networks via linguistic and personality features. In *Working Notes of CLEF 2020-Conference and Labs of the Evaluation Forum*.

Vrij, A. (2000). Detecting lies and deceit: The psychology of lying and implications for professional practice. Wiley.

Xu, P., Patwary, M., Shoeybi, M., Puri, R., Fung, P., Anandkumar, A., Catanzaro, B. (2020). MEGATRON-CNTRL: Controllable story generation with external knowledge using large-scale language models. *arXiv preprint arXiv:2010.00840.*

Zhou, L. and Sung, Y. (2008). Cues to deception in online chinese groups. In *Proceedings of the 41st Hawaii International Conference on System Sciences*. (pp. 146–46).

Zhou, L., Burgoon, J. K., Nunamaker, J. F., Twitchell, D. (2004). Automating linguistics-based cues for detecting deception in text-based asynchronous computer-mediated communications. Group decision and negotiation, 13 (1), 81-106.

CroSentiNews 2.0: A Sentence-Level News Sentiment Corpus

Gaurish Thakkar*, Nives Mikelic Preradović*, Marko Tadić*

*Faculty of Humanities and Social Sciences, University of Zagreb, Zagreb 10000
gthakkar@m.ffzg.hr
nmikelic@m.ffzg.hr
marko.tadic@ffzg.h

Abstract

This article presents a sentence-level sentiment dataset for the Croatian news domain. In addition to the 3K annotated texts already present, our dataset contains 14.5K annotated sentence occurrences that have been tagged with 5 classes. We provide baseline scores in addition to the annotation process and inter-annotator agreement.

1. Introduction

The objective of sentiment analysis (SA) is to categorize the orientation of the author's text (Cardie, 2014). The SA approaches have been analyzed for a variety of text types, including reviews Kumar et al. (2019); Ejaz et al. (2017), news articles (Balahur et al., 2010), and social media (Rosenthal et al., 2017). Prior work (Pang et al., 2008, 2002) in sentiment analysis primarily focused on document-level analysis. Since then, the research community has shifted its focus to fine-grained analysis (Narayanan et al., 2009; Nguyen and Shirai, 2015; Wang et al., 2016, 2017). Although there is a significant amount of research on high-resource languages, the same cannot be said for low-resource ones. In addition, the absence of annotated data limits the applicability of untested models because they cannot be evaluated on a test set. This study presents an enhanced version of the existing sentiment annotation dataset for Croatian sentences.

SentiNews 1.0's sentiment corpus served as the basis for the initial dataset utilized to annotate SentiNews 2.0. Croatian and Slovene language annotations can be found in the SentinNews 1.0 dataset. Croatian is not annotated at the sentence or paragraph level in the SentiNews sentiment corpus, which is annotated at the document level. But it includes news on a range of subjects. This study discusses adding a sentence-level annotation layer to the SentiNews corpus; we think that such a dataset would be addition to the linguistic resources for low-resource languages like Croatian, in addition to providing a crucial building block for developing a sentiment classifier. The entire package will be made available to the public upon acceptance of this paper.

The following are the paper's main contributions: (1) We provide CroSentiNews 2.0, a large dataset for sentiment analysis in Croatian news articles that has been manually annotated. (2) We perform an extensive evaluation of the dataset using a pre-trained transformer model. To improve individual classification performance, we employ the multi-task architecture (Thakkar et al., 2021), which makes use of a multi-level dataset.

The paper is structured as follows: In Section 2, we review the related work in sentiment analysis and data annotation. In Section 3, we describe our overall annotation strategy and summarize the dataset. In Section 4, we describe the details of the experiments. We report the results in Section 5. Before concluding, we provide a summary and future work in Section 6.

2. Related work

Below, we note the related work on existing datasets and sentiment modelling approaches in Croatian.

Agić et al. (2010) made one of the first attempts at Croatian sentiment classification, developing a rule-based model for the automatic detection of general sentiment and polarity phrases in Croatian text from the finance domain. Using the text from the game reviews, Rotim and Šnajder (2017) compiled a dataset with sentiment marked with three-class label and compared the performance of the Support Vector Machine (SVM) classification algorithm on two additional Twitter-derived short-text Croatian datasets.

Babić et al. (2022) utilized a sentiment lexicon in Croatian to annotate a training dataset for the classifier. A dataset for stance, claim, and sentiment for Croatian was presented by Bošnjak and Karan (2019). Using topic modelling and word-emotion lexicons, Pandur et al. (2020) studied sentiments and emotions in crisis communication in the news connected to the COVID-19 epidemic. Pelicon et al. (2020) compiled a sentiment-annotated dataset for Croatian and conducted experiments in zero-shot settings. Using a multilingual Twitter sentiment dataset (Mozetič et al., 2016), Robnik-Šikonja et al. (2021); Ptiček (2021) evaluated state-of-the-art approaches to the cross-lingual transfer of sentiment prediction models for 13 European languages, including Croatian.

Earlier attempts at SA typically depended on rules and lexicons (Hutto and Gilbert, 2014; Baccianella et al., 2010). A majority of the recently proposed work (Sun et al., 2019) relies on fine-tuning the BERT(Devlin et al., 2019) for downstream tasks. Continual pre-training (Gururangan et al., 2020) using unlabelled data has also yielded great results for domain-specific sentiment classification.

3. Annotations

CroSentiNews 2.0 is built using the Croatian Sentiment Dataset (Pelicon et al., 2020), which was compiled

in accordance with the guidelines from Bučar et al. (2016). The collection is comprised of news articles from the website of 24sata, the major media organization in Croatia. The news text includes topics such as automotive news, health, culinary content, and lifestyle advice, in addition to daily news. The dataset's statistics are as follows:

- 2,025 documents;
- 12,032 paragraphs;
- 25,074 sentences.

The documents were initially tagged on a five-point Likert (Likert, 1932) scale (very negative, negative, neutral, positive, very-positive) and subsequently projected onto three-class labels (negative, neutral, positive). Contrary to its Slovenian counterpart, the Croatian SentiNews 1.0 contains documents with sentiment labels but lacks paragraph- and sentence-level annotations. Due to a large number of sentences, we opted to leave the annotation of sentiment at the paragraph level for future work. To prepare the sentences for annotations, we performed sentence tokenization on the whole document. All the sentences were divided into nine groups such that no annotator received a part of the document. Each group consisted of two or more annotators, with an average of approximately 2,114 sentences. Before presenting the sentences to the annotators the text was pre-annotated using an existing sentiment classifier (Thakkar et al., 2021). Thus reducing the sentiment annotation problem to simply label correction if the labels didn't match the text. All the annotators were native speakers of Croatian enrolled in undergraduate courses in linguistics. The total numbers of annotators at the beginning of the crowdsourcing process were 20 but only 16 completed the whole task.

3.1. Annotation guidelines

Following the guidelines outlined in (Mohammad, 2016) and (Pelicon et al., 2020), we produced annotation guidelines detailing the complete annotation technique. Annotators were presented with five labels of sentiment: 1—negative, 2—neutral, 3—positive, 4—mixed, and 5—other/sarcasm. Each label was illustrated with an example. The overall annotation was conducted using the INCEpTION (Klie et al., 2018) tool and a detailed user manual was provided. No user was allowed to view documents from groups other than the ones assigned to him. The documents were marked as complete using the locking mechanism that freezes the annotation for a document. Eventually, statistics of complete documents were derived for the locked documents. We measured the reliability of the agreement using the Fleiss Kappa (Fleiss et al., 1971) score across multiple groups. The scores range from moderate (0.41-0.60) to substantial (0.61-0.80) levels of annotator agreement.

3.2. Data statistics

Out of all the sentences, only 19k instances were tagged by at least one annotator. Out of which only 14.5k had a total agreement. We filtered out the sentences not having any labels. Thus, the corpus is comprised of 1,988 unique non-empty documents and 14,5k sentiment-labelled phrases. Moreover, there were 428 cases of mixed language and 73 instances of sarcasm, but we did not include these in our studies. Instances marked by two annotators who did not agree with the label were likewise eliminated. The final label for a sentence was determined by the majority vote of the sentence's annotators. The distribution of labels for the CroSentiNews 2.0 and SentiNews dataset is depicted in Table 1. The SentiNews 1.0 dataset utilized in our tests is displayed in table 4 .

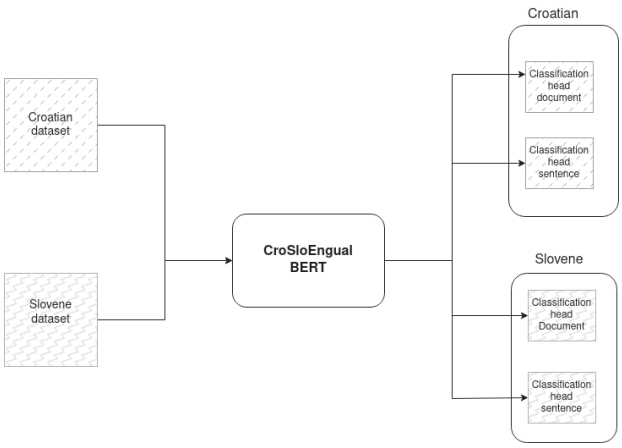


Figure 1: Multi-task training setup using Croatian and Slovene datasets.

4. Experiments

We employed two experimental setups to perform bench-marking on the newly compiled dataset namely single-task fine-tuning and multi-task fine-tuning. In a single-task fine-tuning setup (Devlin et al., 2019), a classification head is added on top of the pre-trained model and all the parameters are jointly trained using the dataset from the downstream task. We modified the model presented in Thakkar et al. (2021) for the multi-task learning (MTL) setup by eliminating the paragraph classification head, as we lacked paragraph-level annotations. Still employing multi-level labels, the approach is restricted to sentence- and document-level labels. The models were trained using the following combinations of datasets.

1. HR_Document_STL - The model consists of a single classification head that is trained end-to-end using a Croatian document-level supervised dataset.

2. HR_Sentence_STL - Similar to the previous scenario, except that the model is trained with sentence-level annotations in only Croatian.

3. HR_Sentence+SL_Sentence - The model consists of a single classification head trained on sentences from datasets in Croatian and Slovene.

4. HR_Document+SL_Sentence_MTL - Based on the MTL model from Thakkar et al. (2021) with no

Language	Level	Total Instances	Positive	Negative	Neutral
Croatian	Document	1,988	321	450	1,217
	Sentence	14,570	3,265	3,353	3,265

Table 1: Distribution of CroSentiNews 2.0 dataset

Language	Level	Total Instances	Positive	Negative	Neutral
Slovene	Document	10,417	1,665	3,337	5,418
	Paragraph	86,803	14,270	23,265	49,268
	Sentence	161,291	26,679	44,014	90,598

Table 2: Distribution of SentiNews 1.0 dataset

# of classification heads	Train set	Test set			
		Croatian		Slovene	
		Document	Sentence	Document	Sentence
3 head (Thakkar et al., 2021)	SL MTL	50.07		**74.86**	**69.40**
3 head (Thakkar et al., 2021)	SL+HR MTL	63.86		74.21	69.21
1 head	HR_Doc - STL	59.24	60.67	49.54	46.87
1 head	HR_Sent - STL	55.00	77.72	54.99	56.29
1 head	HR_Sent+SL_Sent	57.71	72.24	63.15	64.28
2 head	HR_Doc+SL_Sent - MTL	64.46	65.15	68.22	67.77
2 head	HR_Doc+HR_Sent - MTL	**65.60**	**79.65**	55.83	56.97

Table 3: Results of experiments. The first two rows contain values from the original paper.

paragraph-level classification head. The model consists of two classification heads, one for document-level and another for sentence-level classification. The dataset used for training comprises Croatian documents and Slovene sentences.

5. HR_Document+HR_Sentence_MTL - Similar to the preceding scenario, but limited to Croatian and without dataset inter-mixing.

4.1. Model training

We utilized the CroSloEngual BERT (Ulčar and Robnik-Šikonja, 2020) as the pre-trained language model in all trials. All datasets were divided into 80-20 splits for training and testing. For the single-task training, a population-based hyper-parameter search on 10% of the training set yielded the values for the classification of Croatian documents and sentences depicted in Table 4. The trial was terminated early when there was no discernible improvement in the evaluation loss. We utilized a minibatch size of 32, a learning rate of 2e-05, and a hidden state dropout of 0.3 for the MTL. Five epochs were required for the system to converge.

4.2. Results

As the datasets are skewed, macro-F1 is used as the metric of evaluation. Table 2 displays the macro-F1 scores for various combinations of datasets. For the sake of comparison, we also offer the macro-F1 classification of Slovene documents and sentences. We repeated the experiment with 5 seeds for each dataset and reported the mean.

	Parameters	Values
STL	learning rate_document	2.8e-05
	learning rate_sentence	3.9e-05
	weight decay_document	0.15
	batch size	16
MTL	weight decay STL_sentence	0.28
	learning rate	2e-05
	weight decay	0.0
	batch size	32
	hidden state dropout	0.3

Table 4: Hyper-parameters and their values.

The most effective strategy for document-level and sentence-level categorization employed HR_document and HR_sentence-level annotations that were trained in an MTL-like fashion. We believe the model's ability to learn from the same text and language, but for different tasks, has led to considerable advances.

We discovered that the Slovene classification scores did not improve when coupled with Croatian training data in almost every instance. On the other hand, training Croatian (documents) and Slovene (sentences) mixed datasets with different degrees of annotations using MTL architecture enhanced the performance of the Croatian document classification task. This suggests that using data from a typologically similar language in an MTL setup could be advantageous when the source language contains fewer instances (<2,000 or less for Croatian). When a large number of cases are available, however (>14.5k), combin-

ing with Slovene affects the performance of sentence-level categorization in Croatian. In terms of scores, the BERT-based sentence-level classification model performed second best due to a large number of annotated examples. During training using the mix of Slovene and Croatian sentence-level examples, we discovered that this combination required the most time to converge. This may be related to the magnitude of the combined dataset, but it did not result in any notable gains over the BERT-based baseline.

5. Conclusion

In this paper, we introduced the CroSentiNews 2.0 dataset for Croatian. The dataset is derived from an existing dataset with document-level sentiment annotations and is augmented with a sentence-level annotation layer. We have discussed our annotation technique and presented the statistics of the final dataset. In addition, we have conducted studies employing single-task and multi-task frameworks and published their respective performance results. In the future, we want to incorporate other Slavic and non-Slavic language datasets utilizing MTL configuration.

6. Ethical consideration

We work on a sentiment classification task which is a standard NLP problem. Based on our experiments, we do not see any major ethical concerns with our work. We would like to note that, depending on their source, news articles and their annotators can be politically biased.

References

Agić, Željko, Nikola Ljubešić, and Marko Tadić, 2010. Towards sentiment analysis of financial texts in Croatian. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA).

Babić, Karlo, Milan Petrović, Slobodan Beliga, Sanda Martinčić-Ipšić, Andrzej Jarynowski, and Ana Meštrović, 2022. Covid-19-related communication on twitter: Analysis of the croatian and polish attitudes. In Xin-She Yang, Simon Sherratt, Nilanjan Dey, and Amit Joshi (eds.), *Proceedings of Sixth International Congress on Information and Communication Technology*. Singapore: Springer Singapore.

Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani, 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA).

Balahur, Alexandra, Ralf Steinberger, Mijail Kabadjov, Vanni Zavarella, Erik van der Goot, Matina Halkia, Bruno Pouliquen, and Jenya Belyaeva, 2010. Sentiment analysis in the news. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA).

Bošnjak, Mihaela and Mladen Karan, 2019. Data set for stance and sentiment analysis from user comments on Croatian news. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*. Florence, Italy: Association for Computational Linguistics.

Bučar, Jože, Janez Povh, and Martin Žnidaršič, 2016. Sentiment classification of the slovenian news texts. In Robert Burduk, Konrad Jackowski, Marek Kurzyński, Michał Woźniak, and Andrzej Żołnierek (eds.), *Proceedings of the 9th International Conference on Computer Recognition Systems CORES 2015*. Cham: Springer International Publishing.

Cardie, Claire, 2014. Sentiment analysis and opinion mining bing liu (university of illinois at chicago) morgan & claypool (synthesis lectures on human language technologies, edited by graeme hirst, 5(1)), 2012, 167 pp; paperbound, ISBN 978-1-60845-884-4. *Comput. Linguistics*, 40(2):511–513.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics.

Ejaz, Afshan, Zakia Turabee, Maria Rahim, and Shakeel Khoja, 2017. Opinion mining approaches on amazon product reviews: A comparative study. In *2017 International Conference on Information and Communication Technologies (ICICT)*.

Fleiss, J.L. et al., 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Gururangan, Suchin, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith, 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics.

Hutto, C. and Eric Gilbert, 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1):216–225.

Klie, Jan-Christoph, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych, 2018. The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*. Association

for Computational Linguistics. Event Title: The 27th International Conference on Computational Linguistics (COLING 2018).

Kumar, HM, BS Harish, and HK Darshan, 2019. Sentiment analysis on imdb movie reviews using hybrid feature extraction method. *International Journal of Interactive Multimedia & Artificial Intelligence*, 5(5).

Likert, R, 1932. A technique for the measurement of attitudes. *Archives of Psychology*, 22 140:55.

Mohammad, Saif, 2016. A practical guide to sentiment annotation: Challenges and solutions. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. San Diego, California: Association for Computational Linguistics.

Mozetič, Igor, Miha Grčar, and Jasmina Smailović, 2016. Multilingual twitter sentiment classification: The role of human annotators. *PLOS ONE*, 11(5):1–26.

Narayanan, Ramanathan, Bing Liu, and Alok Choudhary, 2009. Sentiment analysis of conditional sentences. In *Proceedings of the 2009 conference on empirical methods in natural language processing*.

Nguyen, Thien Hai and Kiyoaki Shirai, 2015. Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis. In *Proceedings of the 2015 conference on empirical methods in natural language processing*.

Pandur, Maja Buhin, Jasminka Dobša, Slobodan Beliga, and Ana Meštrović, 2020. Topic modelling and sentiment analysis of covid-19 related news on croatian internet portal. *Information Society*, 2020:5–9.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan, 2002. Thumbs up? sentiment classification using machine learning techniques. In *EMNLP*.

Pang, Bo, Lillian Lee, et al., 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in information retrieval*, 2(1–2):1–135.

Pelicon, Andraž, Marko Pranjić, Dragana Miljković, Blaž Škrlj, and Senja Pollak, 2020. Zero-shot learning for cross-lingual news sentiment classification. *Applied Sciences*, 10(17).

Ptiček, Martina, 2021. How good bert based models are in sentiment analysis of croatian tweets: comparison of four multilingual berts:175–182.

Robnik-Šikonja, Marko, Kristjan Reba, and Igor Mozetič, 2021. Cross-lingual transfer of sentiment classifiers. *Slovenščina 2.0: empirical, applied and interdisciplinary research*, 9(1):1–25.

Rosenthal, Sara, Noura Farra, and Preslav Nakov, 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics.

Rotim, Leon and Jan Šnajder, 2017. Comparison of short-text sentiment analysis methods for Croatian. In *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing*. Valencia, Spain: Association for Computational Linguistics.

Sun, Chi, Luyao Huang, and Xipeng Qiu, 2019. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics.

Thakkar, Gaurish, Nives Mikelić, and Tadić Marko, 2021. Multi-task learning for cross-lingual sentiment analysis. In *Proceedings of the 2nd International Workshop on Cross-lingual Event-centric Open Analytics co-located with the 30th The Web Conference (WWW 2021*, volume 2829.

Ulčar, M. and M. Robnik-Šikonja, 2020. FinEst BERT and CroSloEngual BERT: less is more in multilingual models. In P Sojka, I Kopeček, K Pala, and A Horák (eds.), *Text, Speech, and Dialogue TSD 2020*, volume 12284 of *Lecture Notes in Computer Science*. Springer.

Wang, Bo, Maria Liakata, Arkaitz Zubiaga, and Rob Procter, 2017. TDParse: Multi-target-specific sentiment recognition on Twitter. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics.

Wang, Yequan, Minlie Huang, Xiaoyan Zhu, and Li Zhao, 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics.

Kazakh-Uzbek Machine Translation on the Base of Complete Set of Endings Model

U. Tukeyev, G. Akhmet, N. Gabdullina, A. Turganbayeva, T. Balabekova.

Al-Farabi Kazakh National University, Almaty, Kazakhstan
ualsher.tukeyev@gmail.com, gulstan.akhmet@gmail.com, gabdullinanargiza7@gmail.com,
turganbayeva16@gmail.com, tolganaybalabekova@gmail.com

Abstract

Currently, for most Turkic languages, there are difficulties in the field of machine translation due to the lack of parallel corpora for neural machine translation. In Turkic languages, sentence syntax is similar. Therefore, in this work propose an alternative way of machine translation for low resource Turkic languages based on a complete set of endings (CSE) morphology model. In this methodology, the machine translation based on tables of morphological relations between Turkic languages pair. During the research work, tables of morphological analysis were created for the endings of the Uzbek language, tables of correspondence between Kazakh and Uzbek endings, correspondence tables of stems and stop words for the Kazakh and Uzbek languages. Then, based on the CSE model, a Kazakh-Uzbek machine translation algorithm was developed. In addition, on the collected linguistic resources, the Kazakh-Uzbek machine translation program was tested, and the results of the experiments showed the possibility of the proposed machine translation technology for the Turkic languages. The scientific contribution of this work is the development of a technology for the machine translation of the Kazakh-Uzbek pair based on the relational (tabular) models of morphological CSE model as alternative way for low resource Turkic languages.

Keywords: Uzbek language, Kazakh-Uzbek, machine translation, morphology model, morphological analysis

1. Introduction

The rapid growth of data has led to the need for systems that can quickly translate it into other languages. In response, practical research has begun to emerge in the field of machine translation or automatic translation. Most importantly, machine translation research has been a pivotal moment in the emergence of computational linguistics.

Currently, data-driven neural networks improve the quality of translation every year. However, machine translation still has many challenges (Koehn and Knowles, 2017). Such challenges reflected in Turkic languages, one of which is the Kazakh-Uzbek pair.

The area stretches from the Kolyma River in the northeast to the eastern coast of the Mediterranean Sea in the southwest, where more than 200 million people speak Turkic languages. The most common Turkic languages are Turkish, Azerbaijani, Uzbek, Kazakh, Uyghur, Tatar, Turkmen, Kyrgyz, and Bashkir. The level of comprehension of native speakers of Turkic languages varies and depends on vocabulary, not grammar rules. Anyone who speaks any of the native languages of the Turkic language can easily master another language of this group because their grammar is similar. Uzbek and Kazakh, like other Turkic languages, are agglutinative, which means that word formation is carried out by agglutination, that is, by adding affixes to the root or stem of the word, each of which has a unique grammatical meaning. Since both languages belong to Turkic languages group, they are similar in grammatical features.

State-of-art of machine translation is neural machine translation, which require big volume of parallel corpora for good quality of translation.

However, most of Turkic languages are low resource languages because have not satisfactory volume of parallel corpora. Kazakh and Uzbek are low-resource languages for machine translation because lack Kazakh-Uzbek parallel corpora. For this reason, in this work, the morphological model CSE (Complete Set of Endings) is used to perform Kazakh-Uzbek machine translation, which based on relational models of representing of morphologies of languages and its correspondences. This approach does not require parallel corpora.

The motivation of this research is a hypothesis that there is possibility to create machine translation technology on the base of new CSE morphology model for Kazakh-Uzbek pair as example for low resource Turkic languages.

2. Related Works

The article (Turaeva, 2020a) explains the errors and shortcomings of machine translation, the inclusion of the Uzbek language in Google Translate, the problems of translating the system into the Uzbek language, the role of the Uzbek language in the world community, examples of errors. They will also discuss the quality of Uzbek content in the Uzbek language, the lack of a spell check function in programs, the function of processing text in Cyrillic and Latin alphabets.

One of the most widely used automatic translation systems today is Google Translate (Google Translate, 2023). For now, Google Translate supports more than 120 languages. In December 2014, Uzbek and Kazakh languages were added to Google Translate. This is, of course, encouraging, but the Uzbek and Kazakh translation of machine translation has a number of disadvantages (Turaeva, 2020b). This article introduces several types of Google Translate machine translation errors, like unrelated words in sentences and misspellings in words.

Recently, research in the field of Uzbek language processing has intensified. In (Ismailov et al., 2016), a comparative study of stemming algorithms for the Uzbek

language was carried out based on which a proposal was made to choose a stemming model for the Uzbek language. In the paper (Abdurakhmonova and Tuliyev, 2018), morphological analysis was studied using a finite state transducer for Uzbek-English machine translation. A model for the morphological analysis of nouns in the Uzbek language has been developed (Khamroeva, 2022). In (Ismailov et al., 2021), studies of statistical machine translation for translation from Uzbek into English were carried out. In (Aripov et al., 2018), an ontological model of grammatical rules was developed on the example of nouns in the Uzbek and Kazakh languages. A linguistic resource of endings for the Uzbek language was developed (Matlatipov et al., 2020), experiments were carried out with stemming words of Uzbek texts using this linguistic resource of endings. In (Tukeyev et al., 2020), a model and algorithms for morphological segmentation for Turkic languages were studied, based on a morphological model of complete sets of endings, which are necessary for preprocessing for neural machine translation of Turkic languages.

Currently, the field of machine translation with Uzbek-Kazakh language pairs is not fully studied, and open works are rare.

3. Method of machine translation based on the complete set of endings morphology model

In this section, we present a set of linguistic resources necessary for the Kazakh-Uzbek machine translation based on the CSE morphology model. First, since the Kazakh and Uzbek languages have a similar syntactic structure of sentence, we use the hypothesis to translate word-to-word. Therefore, we use CSE morphology models of Kazakh and Uzbek, create correspondence table of endings of Kazakh to endings of Uzbek. Each word of source sentence separated on a stem and an ending. For received stem and ending of source language word, correspondence stem and ending of target language are searching via correspondence tables of stems and endings of these source and target languages. Then the received stem and ending of the target language are joined as a target word.

Therefore, the new machine translation technology based above idea include next steps:

- inferring complete set of endings of Kazakh and Uzbek;
- create the morphological analysis tables of the complete endings of Kazakh and Uzbek;
- create correspondence tables of endings, stems and stop words of Kazakh and Uzbek;
- develop algorithm for Kazakh-Uzbek machine translation based on the CSE model;
- create software based on developed algorithm.

3.1. Inferring complete set of endings of the Uzbek language

Let's consider the scheme of derivation of combinations of possible placements of basic affix types on example of nominal stems (Tukeyev, 2015). The set of affixes to the nominal stems of words in the Kazakh language has four types: - plural affixes (denoted by K); - possessive affixes (denoted by T); - case affixes (denoted by C); - personal affixes (denoted by J). The stem will be denoted by S. The number of placements is determined by the formula (1):

$$A_{nk} = n!/(n-k)! \quad (1)$$

Total possible placement options of one types of endings - 4, two types - 6, three types - 4, four types - 1. Thus, the total number of possible placements of types of nominal base endings are 15. Inferring of complete set of endings of Kazakh is presented in (Tukeyev and Karibayeva, 2020).

The Table 1 shows the type and numbers of noun endings in the Uzbek language for one-type endings.

suffix type	suffixes	number of endings
K	-lar	1
T	-im, m,-ing,-ng, -i, -si, -imiz, -miz, ingiz, -ngiz, -niki	11
C	-ning, -ga, -ka, -qa, -ni, -dan, -da	7
J	-man, -san, -miz, -siz, -dir, -dirlar	6

Table 1: Endings of one-type endings.

The placements of two types of endings are (KT, KC, KJ, TC, TJ, and CJ). Consider the CJ endings number as example.

Number of endings in placements CJ: C*J= 24. The number of endings of the CJ endings placements presents in Table 2.

Examples	Endings type C	Endings type J		Number
		singular	plural	
	-ning – no use -ga, -ka (-qa no use) -ni – no use -da -dan	-man -san -dir	-miz -siz -dirlar	18+6=**24**
ona-	-ga -da -dan	-man -san -dir	-miz -siz -dirlar	3*6=18
kapalak-	-ka	-man -san -dir	-miz -siz -dirlar	6

Table 2: Number of endings of the CJ placement.

Three and four affixes type placements of nouns were considered in the same. The number of endings for Uzbek nominal base words – 339.

Inferring of endings for Uzbek verb base words presents below on the example of transitive present tense.

Hozirgi zamon davom fe'li (present continuous tense in Uzbek language). A verb of this type of tense is formed by combining three different endings with the root. The first method of forming a present continuous tense verb is presents in Table 3 for singular and plural.

In addition, in accordance with the tenses of the verbs, the forms of the negative and question were considered.

The number of Uzbek endings for verb words – 295.

The system of participle endings includes the following types:

-participle's base affixes (R) – 12 (-gan, -qan, -kan, -ayotgan, -yotgan, -adigan, - ydigan, -ar, -r, -mas, -ajak, -mish);

-case affixes (C);

-plural affixes (K);

-personal affixes (J);

-possessive affixes (T).

examples		suffix	1 person	2 person	2 person (respect)	3 person	Number of endings
After consonant	kel-	-yap	-man -miz	-san - siz	-siz -sizlar	-di -dilar	4
After vowel	o'qi-						**4**

Table 3: Number of endings of the present continuous tense.

Then, having considered possible variants of affix types sequences (participle's base affixes for all variants is the same) on the semantic permissibility: - from one type affixes permissible sequences: RK, RT, RC, RJ; - from two type affixes permissible sequences: RKT, RTC, RCJ, RKC, RKJ; - from three type affixes permissible sequences: RKTC, RKCJ; - from four type affixes permissible sequences.

Thus, the quantity of permissible types of the endings of participles is 11. Using the combinatorial procedure for deriving endings described above, below are data on their number for participles and voices.

The number of Uzbek participles endings – 1344. The number of voices endings – 252.

The number of endings for verb base words – 1916.

The total number of endings for Uzbek – 2255.

3.2. Morphological analysis on the complete set of endings for the Uzbek language

In accordance with the machine translation scheme Kazakh-Uzbek pair based on the CSE model, the next step is the morphological analysis of the endings of the Uzbek language. For the morphological analysis needs to create a table of morphological description of endings.

To begin with, let's consider the morphological description of noun endings in the Uzbek language.

During morphological description were used next signs: the noun - <NB>, verb - <VB>, possessive ending - <pos>, case ending - <per>, plural ending - <pl>, etc.

In Table 4 shows example of morphological description of the one-type noun endings in the Uzbek language.

#	Ending	Morphological description	Ending's type
1	lar	<NB>*lar<pl>	K
2	im	<NB>*im<pos><sg><p1>	T
3	m	<NB>*m<pos><sg><p1>	T
4	imiz	<NB>*imiz<pos><pl><p1>	T
5	miz	<NB>*miz<pos><pl><p1>	T
6	ing	<NB>*ing<pos><sg><p2>	T
7	ng	<NB>*ng<pos><sg><p2>	T
8	ingiz	<NB>*ingiz<pos><sg><frm><p2>	T
9	ngiz	<NB>*ngiz<pos><sg><frm><p2>	T
10	i	<NB>*i<pos><sg><p3>	T
11	si	<NB>*si<pos><sg><p3>	T
12	niki	<NB>*niki<pos><sg><p3>	T

Table 4: Morphological description of the one-type noun endings.

All the 2255 Uzbek language endings were described on this basis.

3.3. Creation of correspondence tables of endings, stems and stop words of Kazakh-Uzbek pair

This section discusses the creation of correspondence tables of endings, compiled according to the descriptions of the Kazakh and Uzbek morphological analysis.

The correspondence table of endings of nominal stems for Kazakh and Uzbek words is presented in Table 5. Since the number of correspondences of endings is large, as an example, we have given a table of correspondences of the plural and possessive endings.

Kazakh-endıngs	Kazakh-morph description	Uzbek-morph description	Uzbek-endings
дар	<NB>*дар<pl>	<NB>*lar<pl>	lar
дер	<NB>*дер<pl>	<NB>*lar<pl>	lar
лар	<NB>*лар<pl>	<NB>*lar<pl>	lar
лер	<NB>*лер<pl>	<NB>*lar<pl>	lar
тар	<NB>*тер<pl>	<NB>*lar<pl>	lar
тер	<NB>*тер<pl>	<NB>*lar<pl>	lar
м	<NB>*м<pos><sg><p1>	<NB>*m<pos><sg><p1>	m
ң	<NB>*ң<pos><sg><p2>	<NB>*ng<pos><sg><p2>	ng
ңыз	<NB>*ңыз<pos><sg><p2><frm>	<NB>*ngiz<pos><sg><p2><frm>	ngiz
сы	<NB>*сы<pos><sg><p3>	<NB>*si<pos><sg><p3>	si

Table 5: The correspondence table of Kazakh-Uzbek KT endings (segment).

According to this approach, a correspondence table is compiled for all endings. That is, in the correspondence table, the Uzbek language endings correspond to the Kazakh endings. In addition, according to proposed scheme machine translation, it is needed to create a correspondence table of stems and stop words in Kazakh and Uzbek.

3.4. Development algorithm for Kazakh-Uzbek machine translation based on the CSE model

Here is a description of a machine translation algorithm based on the CSE morphology model.

1. The source word is taken from a sentence of the source Kazakh text.

2. The source Kazakh word is compared with a list of Kazakh stop words.

If the source Kazakh word is found in the Kazakh stop-word, then it is considered as a stop-word, and the stop-word in the Uzbek language corresponding to the Kazakh word is found in the Kazakh-Uzbek stop-word correspondence table. Go to step1.

If no matches are found in the Kazakh stop word list, the next step is doing.
3. The source Kazakh word is divided to the stem and the ending by using of the stemming algorithm with stems-lexicon (NLP-KAZNU/Stemming algorithm with stems-lexicon, 2023; Tukeyev et al., 2021).
4. By the stem of Kazakh word will find the stem of Uzbek word using of Kazakh-Uzbek stems correspondence table.
If no matches are found in the Kazakh stems list, the stem itself will be provided, i.e., Kazakh-stem.
5. The corresponding translation of Kazakh ending to Uzbek ending will found in the Kazakh-Uzbek endings correspondence table.
6. Received Uzbek-stem and Uzbek-ending combined and presented as a translation of the source Kazakh-word.
7. Concatenate received Uzbek-word for current Uzbek-sentence.
8. If Source Kazakh sentence does not end, then Go to step1. Otherwise, next step.
9. End.

3.5 Creating software based on the algorithm

The algorithm is implemented in the Python programming language (NLP-KAZNU/Kazakh-Uzbek machine translation, 2023). The program accepts four files at the entrance. They are:

-Correspondence table of Kazakh- Uzbek endings ("qaz-uz-tab.xlsx");

-Correspondence of Kazakh- Uzbek stems ("qaz- uz -stems.xlsx");

-Correspondence of Kazakh- Uzbek stop words ("qaz- uz -stopwords.xlsx");

-Text in the Kazakh language ("text-qaz.txt").

4. Experiments and Results

In this paper, we have proposed a new way of the Kazakh-Uzbek pair machine translation using the morphological model CSE.

In testing the operation of the machine translation algorithm based on the model of a complete set of Kazakh and Uzbek endings, a case consisting of 45 parallel sentences in the Kazakh and Uzbek languages was used. In Kazakh is used Cyrillic Kazakh alphabet and in Uzbek is used Latin Uzbek alphabet. Results for the Kazakh-Uzbek pair given in Table 6.

Source text in Kazak	Translated text by developed program	Correct translation of text
Электроника – ғылым мен техниканың вакуумда, газда, сұйықта, қатты дене мен плазмада, сондай-ақ олардың бір-бірімен жанасу шекарасында байқалатын электрондық және йондық құбылыстарды зерттеуге және оларды қолдануға арналған саласы.	elektronika – ilm-fan va texnikasining vakuumda , gazda , suyuqlikda , ***bu qiyin tanasi*** va plazmada , ***sondai -oq*** ularning bir - biri bilan aloqa qilish chegarasida ***kuzatilganayotgan*** elektron va iondik hodisalarni o'rganishqa va ulardi ***foydalanishqa*** bag'ishlangan sohasi.	Elektronika - ilm-fan va texnikasining vakuumda , gazda , suyuqlikda , ***qattiq tana*** va plazmada , ***shuningdek*** ularning bir-biri bilan aloqa qilish chegaralarida ***kuzatiladigan*** elektron va ion hodisalarini o'rganishga va ularni ***qo'llashga*** bag'ishlangan sohasi .
Оның физикалық электроника және техникалық электроника деп аталатын басты екі саласы бар.	uning fizik elektronika va ***texnikasilik*** elektronika deb ***boboiladigan*** asosiy ikki sohasi bor .	Uning fizik elektronika va ***texnik*** elektronika deb ***ataladigan*** asosiy ikki sohasi bor.

Table 6: Example of Kazakh-Uzbek machine translation (segment).

The metrics TER (Translation Error Rate), WER (Word Error Rate) and BLEU (Bilingual Evaluation Understudy) were used to evaluate the results of the machine translation based on the model of a complete set of Kazakh and Uzbek endings. The TER metric proposed by Snover in 2006 (Snover et al., 2006). TER is defined as the minimum number of edits needed to change a hypothesis so that it exactly matches to the reference. A TER score is a value in the range of 0-1, but is frequently presented as a percentage, where lower is better. A high TER score suggests that a translation will require more post-editing.

WER or Levenshtein weighted distance, allows measuring the distance between a machine translation and an exemplary translation in the same way as we measure the distance between a dictionary word and a word with a typo (counting whole words as characters, not letters). In fact, WER measures the minimum number of changes that need to be made to get a reference translation from the result of the MT's work (Koehn, 2010). At the same time, WER can consider different variants of the reference translation with different word order.

The BLEU metric is currently the most popular in the modern assessment of MT. Allows to consider not only the accuracy of the translation of individual words, but also chains of words (N-grams).Results of experiments presents in Table 7.

Language pair	TER	WER	BLEU
Kazakh- Uzbek	0.36	0.32	49%

Table 7: Result of experiment.

It is believed that the lower the TER value, the higher the translation quality.

5. Discussion

Although the results of a formal evaluation of machine translation using the proposed technology show a high percentage of errors, the translation itself turned out to be very close in meaning to the translation template, and the value of the metric BLEU is quite high. As show in Table 6, different words between translated text and correct translation (highlighted in bold and italics) maybe in cases: 1) a word is not in Kazakh stems list; 2) a translation word is a synonym of word in template. The second case significantly increases the metrics TER and

WER, although the meaning remains the same. Also, the quality of translation is affected by the stage of word stemming, the quality of which is about 97% for the Kazakh language (Tukeyev et. al., 2021). This shows that the use of the proposed technology is possible. Moreover, the proposed machine translation technology does not need parallel corpora for training as a neural model. Of course, this method must be used if there is not enough parallel corpora for the language pair machine translation.

6. Conclusions and Future Works

The technology of machine translation of the Kazakh Uzbek pair based on a complete set of endings morphology model has been created in the article. That is the main scientific contribution of this work. In addition, for new machine translation technology were created several new resources suitable for both languages. These are a correspondence table of the complete set of endings of the Kazakh and Uzbek languages, based on morphology descriptions of endings, correspondence tables of stems and stop words of the Kazakh and Uzbek languages. Morphological analysis in the Kazakh and Uzbek languages is made for all possible endings of classes of nouns and verb words. Moreover, the result of an experiment with a new technology of machine translation using the created linguistic resources showed good results.

In the future, it is possible to increase the percentage of accuracy of this result by increasing the number of corresponding stems in the Kazakh-Uzbek pair and adjusting the correspondence table of Kazakh-Uzbek endings in both languages and is planned to use this technology for the Kazakh to Uzbek speech machine translation by cascade scheme.

References

Abdurakhmonova, N., Tuliyev, U. (2018). Morphological analysis by finite state transducer for Uzbek-English machine translation. International Journal of Comparative Literature and Translation Studies, 3, pp. 59-66.

Aripov, M., Sharipbay, A., Abdurakhmonova, N., Razakhova B. (2018). Ontologyof grammar rules as example of noun of Uzbek and Kazakh languages. Abstract of the VI International Conference Modern Problems of Applied Mathematics and Information Technology - Al-Khorezmiy, pp. 37–38, Tashkent, Uzbekistan.

Google Translate. https://translate.google.com/ Access date: February 2, 2023

Ismailov A., Jalil M. M. A., Abdullah Z. and Rahim N. H. A. (2016). "A comparative study of stemming algorithms for use with the Uzbek language," 3rd International Conference on Computer and Information Sciences (ICCOINS), doi: 10.1109/ICCOINS.2016.7783180, pp. 7-12.

Ismailov, A.S., Shamsiyeva, G., Abdurakhmonova, N. (2021). Statistical machine translation proposal for Uzbek to English. "Science and Education" Scientific Journal, Volume 2 Issue 12, pp. 212-219.

Khamroeva, S. M. (2022). Finite state machine model of nouns for uzbek language morphological analyzer Computer processing of Turkic languages. X International Conference: Proceedings, Nur-Sultan: IE "Bulatov A.Zh.", pp. 153-164.

Koehn, P., and Knowles, R. (2017). Six challenges for neural machine translation. Proceedings of the first workshop on neural machine translation, Vancouver, pp. 28–39

Koehn, P. (2010). Statistical Machine Translation. Cambridge, UK.

Matlatipov, S., Tukeyev, U., Aripov, M. (2020). Towards the Uzbek Language Endings as a Language Resource. In: Hernes M., Wojtkiewicz K., Szczerbicki E. (eds) Advances in Computational Collective Intelligence. ICCCI 2020. Communications in Computer and Information Science, vol 1287, pp.729-740. Springer, Cham, https://doi.org/10.1007/978-3-030-63119-2_59

NLP-KAZNU/Kazakh-Uzbek machine translation. https://github.com/NLP-KazNU/Kazakh-Uzbek-machine-translation-on-the-base-of-CSE-model Access date: February 2, 2023

NLP-KAZNU/Stemming algorithm with stems-lexicon. https://github.com/NLP-KazNU/Stemming_algorithm_with_stems-lexicon_according_to_the_CSE_morphology_model Access date: February 2, 2023

Tukeyev, U. A. (2015). Automaton models of the morphology analysis and the completeness of the endings of the Kazakh language. Proceedings of the international conference Uzbek languages processing TURKLANG-2015, pp. 91- 100, Kazan. Tatarstan. in Russian, September 17–19.

Tukeyev U., Karibayeva A., Zhumanov Zh. (2020). Morphological Segmentation Method for Turkic Language Neural Machine Translation. Cogent Engineering, Volume 7,- Issue 1 https://doi.org/10.1080/23311916.2020.1856500

Tukeyev, U., Karibayeva, A. (2020). Inferring the Complete Set of Kazakh Endings as a Language Resource. In: Hernes M., Wojtkiewicz K., Szczerbicki E. (eds) Advances in Computational Collective Intelligence. ICCCI 2020. Communications in Computer and Information Science, vol 1287, pp.741-751. Springer, Cham https://doi.org/10.1007/978-3-030-63119-2_60

Tukeyev U., Karibayeva A., Turganbayeva A., Amirova D. (2021). Universal Programs for Stemming, Segmentation, Morphological Analysis of Uzbek Words // In: Nguyen N.T., Iliadis L., Maglogiannis I., Trawiński B. (eds) Computational Collective Intelligence. ICCCI 2021. Lecture Notes in Computer Science, – Springer, Cham, 2021. – vol 12876. – pp. 643–654.

Turaeva, G.Kh. (2020). Problems of machine translation when translating into Uzbek // Universum: technical sciences: electronic scientific magazine, 10(79), URL: https://7universum.com/ru/tech/archive/item/10816

Snover M, Dorr B, Schwartz R, Micciulla L, Makhoul J. (2006). A study of translation edit rate with targeted human annotation. In: Proceedings of the 7th conference of the association for machine translation of the Americas (AMTA 2006). Visions for the Future of Machine Translation, Cambridge, Massachusetts, USA, pp 223–231.

IndicSumm: Summarization Resource Creation for Eight Indian Languages

Sireesha Vakada[1], Anudeep Ch[1], Mounika Marreddy[1], Radhika Mamidi[1]

[1]IIIT Hyderabad (iiit.ac.in)
lakshmi.sireesha@research.iiit.ac.in, anudeepch528@gmail.com, mounika.marreddy@research.iiit.ac.in, radhika.mamidi@research.iiit.ac.in

Abstract

Recent works in summarization have focused on creating reliable resources for high-resource languages like English. However, the creation of dedicated resources is rarely seen for low-resource Indian languages. In this paper, we create summarization resources for Indian languages (Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Tamil, Telugu) by introducing ISummCorp (Indic Summarization Corpora) and IndicSumm (Indic Language Summarization Models). ISummCorp is a highly abstractive summarization dataset extracted from the Times Of India (TOI). It is manually annotated by experts across eight Indian languages. Human and intrinsic evaluations demonstrate the high quality, abstraction, and compactness of ISummCorp. IndicSumm is a set of diverse monolingual and multilingual models based on ISummCorp. We refined IndicSumm, by finetuning the sophisticated, multilingual pre-trained mT5 model. With ISummCorp, we show that a language can perform better in a monolingual environment when trained with enough monolingual data than in a multilingual finetuning scenario. To investigate the potential of monolingual models, we finetune mT5 using ISummCorp in both monolingual and multilingual situations and achieved better performance in a monolingual setting. Furthermore, we compare IndicSumm to other finetuned summarization models and achieve state-of-the-art results.

Keywords: Summarization, Indian languages, multilingual models, monolingual models, low-resource

1. Introduction

Text summarization is one of the most focused areas of the Natural Language Processing (NLP) community. Summarization provides a brief synopsis of the input text containing the key information. It presents several hurdles regarding high-quality resources, models, and accurate text generation. The scientific community has shown great interest in summarization due to the expansion of digital data and deep-learning innovations over the past few years (Marreddy et al., 2022; and Grail et al., 2021). Several extractive and abstractive techniques were proposed that use machine learning and deep-learning techniques (Sefid and Giles, 2022; and Gupta et al., 2022). However, high-quality, meticulously extracted, sizable, annotated datasets are required to benefit fully from these deep learning techniques. Recently, the NLP community has contributed large-scale multilingual datasets towards summarization in different languages (Varab and Schluter, 2021). However, datasets on such a large scale must be appropriately retrieved and evaluated. If not, the models trained on these datasets will likely be at stake.

High-resource languages like English have many datasets to train effective models. On the other hand, low-resource languages currently benefit from high-resource languages when trained in multilingual settings (Hasan et al., 2021). It is crucial to explore the potential of these low-resource languages when trained alone and with an appropriate amount of language-specific data. Here, we try to develop monolingual and multilingual models and datasets for a few low-resource languages in the Indian subcontinent.

This work aims to contribute resources towards eight Indian languages (Bengali, Gujarati, Marathi, Kannada, Tamil, Telugu, Hindi, and Malayalam) by providing new benchmark datasets and models. Here, we introduce ISummCorp: a manually annotated multilingual abstractive summarization dataset sourced from TOI, and IndicSumm: a set of monolingual and multilingual models finetuned on ISummCorp.

IndicSumm is a professionally annotated dataset consisting of about 376k article-summary pairs from eight Indian languages. It is one of the largest summarization datasets and the first publicly available dataset for a few languages (Malayalam, Kannada). With the help of ISummCorp, we build IndicSumm models finetuned on a pretrained multilingual model mT5 (Xue et al., 2020). By finetuning the models in a monolingual setting, we want to demonstrate that any language with enough data in a monolingual setting outperforms the multilingual finetuning strategy. We are the first to present monolingual summarization models trained in multiple Indian languages. We later train a multilingual model on all eight languages to create a generic summarization model specific to Indian languages. We also compare the performance of IndicSumm with the existing multilingual summarization models and achieve state-of-art results on our dataset. We publicly make available all the resources related to IndicSumm and ISummCorp[1].

2. ISummCorp

ISummCorp is one of the largest multilingual summarization datasets spread across eight Indian languages: Hindi (hi), Tamil (ta), Telugu (te), Bengali (bn), Gujarati (gu), Marathi (ma), Malayalam (ml), and Kannada (kn).

2.1. Why ISummCorp?

Unlike English, very few attempts have been made to the Indian languages in summarization. Additionally, due to the

[1]https://github.com/ sireeshasummarization/is

structural and morphological differences across languages, models developed using English datasets cannot be translated or extended to Indian languages. Interestingly, a few multilingual datasets include Indian languages as part of their vast corpora, like XL-Sum (Hasan et al., 2021) and MassiveSumm (Varab and Schluter, 2021). Other monolingual datasets are also created for Indian languages (Urlana et al., 2022; and Marreddy et al., 2021), but we restrict our discussion towards multilingual datasets because of the availability of their huge corpora.

Features (sent refers to sentences here*)	**Bengali (Bn)**	**Gujarati (gu)**	**Marathi (mr)**	**Kannada (kn)**	**Malayalam (ml)**	**Hindi (hi)**	**Tamil (ta)**	**Telugu (te)**
Total #samples	32093	25787	45988	82866	15198	123526	12297	38269
Avg length of document(sent)	24.66	23.01	21.89	21.36	18.88	17.35	18.07	19.24
Avg #tokens in a document	397.0	389.75	317.38	278.30	254.45	384.71	289.32	251.48
Avg length of summary(sent)	3.07	2.36	2.32	2.29	1.811	2.91	1.66	2.97
Avg #tokens in a summary	40.11	38.63	33.42	29.21	20.73	62.82	21.61	29.71
Total vocabulary (document)	7.7M	5.6M	9M	14.7M	2.75M	22M	2.2M	6.2M
Total vocabulary (summary)	1.1M	840k	1.3M	2.2M	291k	5.8M	244k	992k

Table 1: ISummCorp dataset statistics.

XL-Sum covers seven Indian languages, whereas MassiveSumm covers around ten Indian languages from various websites. XL-Sum is a single-sentence summary, whereas the MassiveSumm extraction process is different for different websites and unpredictable. (Urlana et al., 2022) pointed out various flaws present in XL-sum and MassiveSumm datasets which might account the flaws for a large portion of the respective datasets. We further compare the performance of the XL-Sum model on our dataset. We are not considering MassiveSumm for comparison because of the complete unavailability of data and the unreliable nature of the summaries.

We have developed ISummCorp, a high-quality summarization dataset for Indian languages, to address the lack of standard and reliable data in this area. ISummCorp is very meticulously extracted based on the structure of the article. Moreover, unlike the existing datasets, we carefully filtered the dataset with various heuristics that will be discussed in the coming sections. We hope ISummCorp will benefit the field of NLP for Indian languages by providing high-quality datasets for researchers to use.

2.2. Data Collection

ISummCorp is extracted from a single source, the Times Of India (TOI). TOI is one of India's most used and reliable news websites. TOI publishes stories on its website from diverse subjects such as science, technology, politics, current events, economics, finance, and health. For the news to reach every part of the country, TOI started publishing online news articles in native languages through various websites such as Samyam (Telugu)[2], Navabharat Times (Hindi)[3], VijayKarnataka (Kannada)[4], Samayam (Malayalam)[5], Samayam (Tamil)[6], Maharashtra Times (Marathi)[7], Eisamay (Bengali)[8], iamgujarat (Gujarati)[9].

2.3. ISummCorp Creation and Pre-processing

After a deep analysis of the extraction processes, we found that TOI news article websites turn out to be of trustworthy structure and content. To collect the data from TOI websites, we have created domain-specific crawlers to extract article-summary pairs effectively. The structure of the article page starts with a headline, followed by a short paragraph that concisely describes the content present in the article, i.e., what we consider to be a summary. The short paragraph is then followed by the actual input article. (Please refer for an example article page [10].)

We used Polyglot-tokenizer[11] for tokenization at both sentence and word levels. For ISummCorp, our summary extraction process has two tiers: 1) filtering during extraction and 2) evaluation after the extraction process. In the first phase, basic pre-processing such as removing any URLs, non-Unicode characters, and unwanted text, is removed. For the second phase, we designed various heuristics to filter out the noisy data. One of the heuristics is removing articles of tiny size (or) if the summary length is too small or large relative to the article size. We eliminate the articles that do not obey the heuristics. These heuristics help us create a more uniform and consistent dataset. We report all the statistics related to ISummCorp in Table 1.

	Intrinsic Evaluation		Human Evaluation			
Lang	C-ratio	Abstractivity	Consistency	R&C	Fluency	Coherence
bn	0.189	23.5%	0.98	4.3	4.7	4.7
gu	0.127	20.65%	0.98	4.6	4.8	4.6
hi	0.23	8.75%	0.93	4.4	4.9	4.5
kn	0.132	11.92%	0.94	4.5	4.9	4.6
ml	0.113	21.2%	0.97	4.7	4.7	4.7
mr	0.121	14.6%	0.98	4.5	4.9	4.6
ta	0.112	33.4%	0.97	4.4	4.8	4.5
te	0.17	32.3%	0.99	4.7	4.7	4.8

Figure 1. Quality Analysis of ISummCorp for different languages. Here, C-ratio describes the Compression ratio of the article, and R&C describes the Relevance and Coverage metric. Abstractivity is the percentage of novel 1-grams in the summary. The Consistency metric is rated out of 1, and the remaining human evaluation metrics are rated out of 5.

3. Quality Analysis of ISummCorp

When attempting to create a dataset on such a huge scale, its quality must be ensured. We follow a two-step process to ensure that the summaries extracted are reliable and concise. We first automatically evaluate ISummCorp based on compression and abstractivity introduced by (Bommasani and Cardie, 2020). Later, we manually

[2] https://telugu.samayam.com/
[3] https://navbharattimes.indiatimes.com/
[4] https://vijaykarnataka.com/
[5] https://malayalam.samayam.com/
[6] https://tamil.samayam.com/
[7] https://maharashtratimes.com/
[8] https://tamil.samayam.com/
[9] https://maharashtratimes.com/
[10] https:shorturl.at/uvMOW
[11] https://pypi.org/project/ polyglot-tokenizer

evaluate our dataset based on Consistency, Relevance, Fluency, and Coherence. Figure 1 showcases our dataset's intrinsic and manual evaluation.

3.1. Intrinsic Evaluation

Compression reveals how condensed a summary is relative to the input article. Including novel vocabulary in the summary while training an abstractive summarization model is crucial. We measure this abstractiveness of our dataset based on novel n-grams.

Lang Setting →	Monolingual			Multilingual		
Languages ↓	R-1	R-2	R-L	R-1	R-2	R-L
Bengali (bn)	39.38	28.1	37.66	32.73	19.32	31.55
Gujarati (gu)	33.22	22.97	31.67	30.5	22.56	30.49
Hindi (hi)	65.14	55.77	62.63	57.84	25.11	57.69
Kannada (kn)	59.39	52.13	58.3	53.91	45.52	53.88
Malayalam (ml)	26.18	15.33	25.04	21.18	18.80	21.03
Marathi (mr)	59.16	52.04	58.36	51.67	21.67	50.63
Tamil (ta)	44.89	27.69	41.57	37.53	23.86	37.32
Telugu (te)	36.95	22.46	35.88	32.95	19.68	32.93

Table 2: Comparison of IndicSumm monolingual and IndicSumm multilingual models

3.2. Manual Evaluation

Evaluating a dataset by intrinsic measure serves no justice for abstractive summarization. So, we manually evaluate our dataset by crowd-sourcing. We randomly sampled our dataset from each language and assigned it to professional annotators who are also native speakers. Each annotator is asked to rate 100 articles, and two annotators are chosen for each language.

Each article is rated based on four factors: Consistency, Relevance and Coverage, Fluency, and Coherence. Consistency is rated on a binary scale, and the remaining factors follow a 1-5 scale for evaluation. Below, we briefly describe the four factors as part of annotation guidelines.

- **Consistency (Yes/No):** Does the summary convey the whole essence of the article?
- **Relevance and Coverage (1-5):** The summary of the article must be significant and concise. It should contain all the important aspects and named entities in the article. Any text unrelated to the article or repetitive information should be considered redundant.
- **Fluency (1-5):** Fluency is one of the most subjective factors in text generation. Under fluency, we try to maintain the rightness of grammatical and syntactical aspects of the summary. Furthermore, the summary has understandable by the majority of natives.
- **Coherence (1-5):** When we try to train an abstractive model, we must refrain from inputting a summary where the sentences are disjoint. The reader should find some continuity or relevance within the paragraph which comes under coherence.

We try to provide the human-evaluated results of ISummCorp in Figure 1.

4. Experimental Setup

We discussed the uniqueness and the generic capability of the ISummCorp dataset in the previous sections. Finetuning transformer-based and sequence-to-sequence-based models have achieved state-of-the-art results on many abstractive summarization datasets. Recently, a wide variety of pre-trained models for multiple languages has come. We choose to finetune the mT5 model (Xue et al., 2020) to create IndicSumm. mT5 is a multilingual variant of T5 model pre-trained on mC4 dataset (Xue et al., 2020) trained on 101 different languages. We chose mT5 because of its massive coverage over our eight Indian languages and its unique training objective for all downstream tasks. Here, we discuss in detail the finetuning process of mT5 (Xue et al., 2020) in both monolingual and multilingual settings.

4.1. Monolingual IndicSumm

In this study, we trained eight different mT5 models, each for a language. The mT5 model consisted of 8 blocks, with two layers of encoder and decoder settings in each block. The hidden and filter sizes were set to 512 and 1024, respectively, with six attention heads. We applied a dropout rate of 0.1 and used the AdamW (Loshchilov and Hutter, 2017) optimizer with a maximum learning rate of 3E-4. The batch size was set to 2048 tokens. The training was performed on 4 NVIDIA GeForce GTX 1080Ti GPUs and took approximately five days for 12 epochs. The tokenizer used was Google's pre-trained mT5 tokenizer, which covers 101 languages and has a vocabulary of 250k words. Due to computational limitations, we had to reduce the input and output to 512 and 50 tokens, respectively. We reserved 20% of the data for testing and 10% for validation and used the remaining 70% to train the model. We followed a similar training strategy (in terms of hyperparameters) as of (Lample and Conneau, 2019).

4.2. Multilingual IndicSumm

This model is created for the multilingual setting by giving all the article-summary pairs from eight languages as input. The input batch size, learning rate, and optimizer remained as the monolingual models. The multilingual model setting is the same as the monolingual setting. However, for each batch, the input article-summary pairs are from multiple languages, unlike in a monolingual setting. The multilingual model took five days to train 12 epochs. The tokenizer used was bert-base-multilingual-cased, which is pretrained on 104 languages and has a vocabulary size of 110,000.

5. Results and Analysis

Here, we conduct an empirical evaluation of different summarization techniques and IndicSumm when tested on ISummCorp. We first try to analyze the performance of IndicSumm monolingual and multilingual models. Later, we compare the performance of IndicSumm with standard baselines and then with recently successful summarization models.

5.1. Analysis of IndicSumm

Our main objective behind finetuning monolingual models is to explore the potential of low-resource languages. Table 2 reports the F1-score of ROUGE (Lin, 2004) metric for the monolingual and multilingual models finetuned. As observed, all the monolingual models outperform the multilingual model for all languages. Our main observation is that low-resource languages can outperform the

multilingual strategy when trained separately with enough resources. However, the multilingual and monolingual model scores are competitive for the languages Gujarati, Marathi, and Tamil. This can be accounted for relatively fewer training samples of the languages. Hindi and Kannada languages have showcased higher ROUGE scores of all the languages, which might account for their large number of training samples. Figure 2 demonstrates a sample example from ISummCorp and the predicted summary.

Methodology	Random			LEAD3			LexRank			IndicSumm		
Languages↓ / Metric→	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Bengali (bn)	14.96	5.08	13.72	27.99	16.3	26.03	20.37	10.66	16.09	39.38	28.1	37.66
Gujarati (gu)	14.70	4.84	13.33	24.27	12.62	22.32	22.49	9.75	16.62	33.22	22.97	31.67
Hindi (hi)	22.58	9.07	19.10	42.05	29.13	34.09	34.79	20.38	25.73	65.14	55.77	62.63
Kannada (kn)	16.72	6.70	15.85	50.08	42.20	49.49	32.61	21.68	27.18	59.39	52.13	58.3
Malayalam (ml)	15.14	4.71	14.26	20.74	7.74	19.53	22.65	11.55	18.75	26.18	15.33	25.04
Marathi (mr)	14.98	6.03	14.28	32.26	22.42	31.17	30.01	19.20	24.48	59.16	52.04	58.36
Tamil (ta)	16.19	3.79	14.95	29.73	14.24	27.57	23.64	9.59	18.42	44.89	27.69	41.57
Telugu (te)	15.61	3.23	14.84	31.57	17.52	31.43	25.00	10.81	20.06	36.95	22.46	35.88

Table 3: Comparison of IndicSumm with few baselines

Methodology	XL-Sum mT5			IndicBART-IndicSS			IndicBART-XLSum			IndicSumm (Multilingual)			IndicSumm (Monolingual)		
Languages↓ / Metric→	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Bengali(bn)	20.80	9.73	20.00	0.24	0.02	0.22	19.7	10.35	18.34	32.73	19.32	31.55	39.38	28.1	37.66
Gujarati(gu)	20.24	9.01	18.87	0.57	0.04	0.46	16.88	7.19	15.56	30.5	22.56	30.49	33.22	22.97	31.67
Hindi(hi)	33.57	17.55	27.89	24.15	9.96	20.31	30.85	17.71	27.43	57.84	25.11	57.69	65.14	55.77	62.63
Kannada(kn)	4.8	0.06	4.87	1.84	0.01	1.63	33.80	24.35	32.30	53.91	45.52	53.88	59.39	52.13	58.3
Malayalam(ml)	4.70	0.04	4.70	0.94	0.007	0.7	14.67	6.04	13.6	21.18	18.80	21.03	26.18	15.33	25.04
Marathi(mr)	32.91	21.77	31.50	16.17	7.06	15.37	27.15	18.53	26.06	51.67	21.67	50.63	59.16	52.04	58.36
Tamil(ta)	25.11	11.95	27.09	0.95	0.06	0.67	22.29	10.01	20.54	37.53	23.86	37.32	44.89	27.69	41.57
Telugu(te)	26.02	10.27	24.90	2.52	0.02	1.91	19.57	9.22	18.31	32.95	19.68	32.93	36.95	22.46	35.88

Table 4: Comparison of IndicSumm models with the existing multilingual models

5.2. Baselines

Here, we compare IndicSumm monolingual setting with a few standard baseline techniques. We considered three baseline systems: Random (selecting k random sentences to be the summary), LEAD-k: a simple intuition of choosing the first K sentences (or) tokens is considered to be the summary, and LEXRANK (Erkan and Radev, 2004): a graph-based approach that ranks sentences based on the weights of a TF-IDF graph generated from the input.

Table 3 shows the ROUGE scores for the baselines compared with IndicSumm. According to the table 3, IndicSumm outperforms all the established baselines in terms of all the ROUGE metrics. Of all the baselines, LEAD-k tends to outperform the Random and LexRank models. Random baseline performed similar ROUGE scores for all the datasets irrespective of their abstractiveness.

5.3. Comparison with other pre-trained models

We compare the performance of IndicSumm monolingual models with the existing finetuned multilingual models. We compare IndicSumm with the extensively trained multilingual transformer models such as the XL-Sum mT5 model (Hasan et al., 2021) and IndicBART variants (Dabre et al., 2021).

XL-Sum mT5: (Hasan et al., 2021) have released a multilingual variant of T5 model finetuned on XL-sum dataset. XL-sum dataset supports 44 languages trained over 1 million article-summary pairs.

IndicBART variants: IndicBART (Dabre et al., 2021) is a multilingual pre-trained sequence-to-sequence model trained specifically on Indian languages based on mBART architecture. IndicBART evaluates the summarization task by finetuning IndicSentenceSummarization and XL-sum datasets. We compare our models with IndicBART-XLSum[12] model and MultiIndicSentenceSummarization [13]. Table 4 showcases the performance of different finetuned models on ISummCorp. From the table 4, we observe that IndicSumm outperformed the existing multilingual models in all aspects. We observe that the IndicBART-XLSum performed better than the XL-Sum mT5 for most languages except Malayalam and Kannada. We also observe that both LexRank and XL-Sum mT5 performed similarly. However, the inferior performance of XL-Sum mT5 on Kannada and Malayalam datasets is due to the absence of these language datasets in the XL-Sum. Also, IndicSumm multilingual model has outperformed all the existing baseline and multilingual models. We should be able to deduce how critical it is to have a suitable summarising dataset from the outcomes of IndicBARTIndicSentenceSummarization.

[12] https://huggingface.co/ai4bharat/IndicBART-XLSum

[13] https://huggingface.co/ai4bharat/MultiIndicSentenceSummarization

Article	https://telugu.samayam.com/sports/cricket/news/india-vs-sri-lanka-virat-kohli-becomes-6th-indian-batter-to-score-8000-test-runs/articleshow/89989203.cms
Actual summary	టెస్టుల్లో ఇప్పటి వరకూ ఐదుగురు భారత క్రికెటర్లు మాత్రమే 8వేల పరుగుల మార్క్ కి చేరుకోగా.. తాజాగా విరాట్ కోహ్లీ ఈ రికార్డ్ లో చోటు సాధించాడు. So far only five Indian cricketers have reached the 8000 run mark in Tests. And Virat Kohli has recently achieved this record.
Predicted summary	విరాట్ కోహ్లీ టెస్టుల్లో మరో అరుదైన మైలురాయిని అందుకున్నాడు. కెరీర్ లో 100వ టెస్టులో 8వేల పరుగుల మార్క్ ని అందుకున్న రెండో క్రికెటర్ గా విరాట్ కోహ్లీ నిలిచాడు. Virat Kohli has achieved another rare milestone in Tests. He became the second cricketer to reach the 8000-run mark in the 100th Test of his career.

Figure 2: A sample article-summary pair and predicted summary from ISummCorp

Otherwise, no matter how many hyperparameters are pre-trained into a model, the outcome from finetuning is always substandard.

5.4. Scope of Extension

We used ISummCorp to create monolingual and multilingual summarization models in this work. However, with ISummCorp extracted from a single source in different languages, it can be used to draw parallel data (based on similar articles from different languages) and create a cross-lingual summarization dataset of around 56 Indic language combinations. This further helps in advancements of cross-lingual transfer learning of Indian languages. In addition, ISummCorp can also be used to produce domain-specific data for specialized applications. By leveraging the flexibility of our dataset, researchers can tailor the data to their specific needs and use it to advance state of the art in domain-specific summarization.

6. Conclusion

Indian languages are resource-poor, relative to English, regarding available datasets, feature representations, and machine learning models. We tried to bridge the gap by creating Indian language-specific datasets and models. We developed ISummCorp, a high-quality and standard largescale multilingual dataset comprising around 376k samples from eight Indian languages. We also present IndicSumm, a set of Indic language-based summarization models created to promote the development of Natural Language Generation for Indic languages. We are the first to develop monolingual summarization models for Indian languages, to enhance the performance of summarization tasks. We also explore the potential of low-resource monolingual models by training them with enough data. In conclusion, we hope that the IndicSumm and ISummCorp resources will be helpful to the research.

We reused publicly available information from TOI website to create summarization datasets in eight Indian languages. We have reviewed the privacy policy of the website [14]. We do not foresee any harmful uses of using the data from the TOI website.

[14]https://timesofindia.indiatimes.com/ privacy-policy/cookiepolicy/80245266.cms

7. References

Bommasani, Rishi and Claire Cardie, 2020. Intrinsic evaluation of summarization datasets. In *Proceedings of the 2020 Conference on (EMNLP)*.

Chowdhury, Radia Rayan, Mir Tafseer Nayeem, Tahsin Tasnim Mim, Md Chowdhury, Saifur Rahman, and Taufiqul Jannat, 2021. Unsupervised abstractive summarization of bengali text documents.

Dabre, Raj, Himani Shrotriya, Anoop Kunchukuttan, Ratish Puduppully, Mitesh M Khapra, and Pratyush Kumar, 2021. Indicbart: A pre-trained model for natural language generation of indic languages. *arXiv:2109.02903*.

Erkan, Günes and Dragomir R Radev, 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *JAIR*, 22:457–479.

Grail, Quentin, Julien Perez, and Eric Gaussier, 2021. Globalizing bert-based transformer architectures for long document summarization. In *Proceedings of the 16th conference of EACL: Main volume*.

Gupta, Anushka, Diksha Chugh, Rahul Katarya, et al., 2022. Automated news summarization using transformers. In *SAC*. Springer, pages 249–259.

Hasan, Tahmid, Abhik Bhattacharjee, Md Saiful Islam, Kazi Samin, Yuan-Fang Li, Yong-Bin Kang, M Sohel Rahman, and Rifat Shahriyar, 2021. Xl-sum: Largescale multilingual abstractive summarization for 44 languages. *arXiv preprint arXiv:2106.13822*.

Lample, Guillaume and Alexis Conneau, 2019. Crosslingual language model pretraining. a*rXiv:1901.07291*.

Marreddy, M., Oota, S.R., Vakada, L.S., Chinni, V.C. and Mamidi, R., 2022. Am I a Resource-Poor Language? Data Sets, Embeddings, Models and Analysis for four different NLP Tasks in Telugu Language. *ACM, TALLIP*, *22*(1), pp.1-34.

Lin, Chin-Yew, 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*.

Sefid, Athar and C Lee Giles, 2022. Scibertsum: Extractive summarization for scientific documents. In *International Workshop on Document Analysis Systems*. Springer.

Urlana, Ashok, Nirmal Surange, Pavan Baswani, Priyanka Ravva, and Manish Shrivastava, 2022. Tesum: Humangenerated abstractive summarization corpus for telugu. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*.

Varab, Daniel and Natalie Schluter, 2021. Massivesumm: a very large-scale, very multilingual, news summarisation dataset. In *Proceedings of the 2021 EMNLP.*

Xue, Linting, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel, 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv:2010.11934*.

Marreddy M, Oota SR, Vakada LS, Chinni VC, Mamidi R. Multi-Task Text Classification using Graph Convolutional Networks for Large-Scale Low Resource Language. In2022 (IJCNN) 2022 Jul 18 (pp. 1-8) IEEE

Challenges and a New Paradigm Frontier of Human Language Technology Applications in Business Management, Business Communication in Organizations and Society

Zygmunt Vetulani[1], Peter Odrakiewicz[2]

[1]Adam Mickiewicz University, Poznań
vetulani@amu.edu.pl
[2]Academy of Management, USA N.Y.
p.odrakiewicz@gmail.com

Abstract

In the world of high technologies, until recently, it seemed that globalization tendencies were the dominant trend. Globalization and present challenges of deglobalization occur to be an underestimated phenomenon. Recently, we have been observing changes in this area. In the business world, communication, both interpersonal and among institutions, has always played a key role. While the globalization model facilitate business communication, moving away from this model generates new challenges for communication technologies, among which Human Language Technologies (HLTs) and AI play a special role. Regardless of which of these trends turns out to be dominant, communication in a natural language will play a key role at all levels of management for a long time, including language interaction between humans and their technological environment. In order to better face globalization and recent deglobalization related challenges, a thorough analysis of practical needs is crucial. With this paper we intend to open discussion on problems related to the use of the modern state-of-the-art business communication technologies, the vision of future needs, emerging ideas and novel solutions involving AI-based HLTs. It is worth to address these developments make resources more evenly allocated, especially to the under resourced and developing parts of the globe.

Keywords: human language technologies, business management, globalization, deglobalization, communication challenges.

1. Introduction

Businesses, like other organizations require effective communication to operate efficiently and meet their objectives. International business firms require effective communication at a number of levels. A company must communicate with its work force, customers, suppliers and host government officials. (Ferraro, 1996)

In today's globalized labor market, business, enterprises and multinational corporations demand some communication and intercultural communication competences while employing graduates. (Irving, 2008).

2. Globalization Challenges

The effects of globalization are being experienced around the world. The interconnectedness and interdependency of cultures is felt within and between the organizations, cultures, and societies of the world. One of the strongest evidences of globalization in this day is the interdependency of global economies (ibid.).

The idea of globalization is a cultural product and accompanies the development of technology that is subordinated to the vision of a human-friendly environment. The fact that the development of technology does not keep up with the vision leads to perturbations in the progress of globalization and it is a historical phenomenon. We know it because it has repeatedly accompanied the collapse of empires in favor of the decentralization of central power, and consequently the emergence of various trends in the processes of deglobalization. The accompanying effect was communication problems, which can be called "Tower of Babel syndrome". Nowadays, there is an alternative in the form of the development of communication technologies to deal with challenges more effectively instead of the rebirth of globalization processes.

The current development of alternative communication technologies allows for a more appropriate handling of the complexity of communication and management processes, enabling a better response to the challenges of de-globalization, and the emerging trends of limited protectionist globalization that meet the goals of individual leading systems, states and corporate players.

3. Deglobalization Challenges

With the increasing challenges of globalization, new growing trend in deglobalization, disruption of international and business related travel due to post pandemic health and safety related issues, world security political challenges and financial constraints, many organization are facing the increasingly interwoven global economy. Organizations must decide on the most effective modes of communication in business and society. It may involve more advanced digitalized tools including the use of HLT (Human Language Technologies) in a new paradigm of Artificial Intelligence (AI) era.

According to Irwin (2020), the COVID-19 pandemic is driving the world economy to retreat from global economic integration (Baldwin and Evenett 2020). Also the authors of this paper note emerging trends of deglobalization (see Chapter 2, above) and coexisting with them at the same time, although somewhat weaker, trends in re-emerging new limited sectoral globalization proposed by major global economic and military powers with

Contents

leading corporations at the forefront of this changes. Policymakers and business leaders are now questioning whether global supply chains have been stretched too far. In an environment where alliances are uncertain and international cooperation is absent, they are also asking whether they should reduce their economic interdependence. National security and public health concerns are providing new rationales for protectionism, especially for medical gear and food, and an emphasis on domestic sourcing. This retreat will not mark the end of globalisation, a process that has reached a historically high level. But globalisation can be reversed, at least partially. The Great Recession of 2008–2010 marked a historic turning point in the degree of global economic integration. Now, in response to the current health and economic crisis, policymakers appear poised to take deliberate steps to reinforce the movement toward deglobalization noted. These steps threaten to slow or reverse the economic growth delivered by globalisation. Even worse, new restrictions on trade could proliferate and inflict damage that could take decades to reverse (ibid).

In contrast to the above opinion (by Irwin), the authors of this paper state that both deglobalization and the post-covid induced changes do not have to necessarily inflict such a damage as described by some authors (recently by Irwin, ibid.).

If we embrace new HLTs in business advances with new developments in AI technologies, with their balanced use with the humanistic vision of society incorporating sustainable management practices, strengthening education, research and development will lead to new innovation and bring long awaited solutions. This may include, but is not limited to, production of cleaner energy, development of more efficient and less polluting logistic chains, cleaner environment and just society for all its members.

4. Business Communication and Intercultural Challenges

Due to the communication complexities, as mentioned above, the way leaders and organizations approach business management and business communication across various managerial processes have dramatically changed. While digital and intercultural competence was once viewed as something only necessary for those engaged in business communication departments and direct international relations, today organizations face a need to equip the majority of their leaders and staff in effective digital and intercultural communication competence.

Business English in its nature in a special way is a tool of choice in automatic computer processing of business documentation. It can be viewed as an intellectual bridge in communication management in times of digitalized and transformation corporate changes and challenges (Odrakiewicz 2009).

How can AI and advanced use of HLT play a role in the following areas described below?

The ability of an organization to identify, maintain, and strategically build interorganizational networks for the mutual benefit of network members and for the reduction of dysfunctional relational outcomes has been the underlying strategy of many organizations in globalizing their operations (Gronroos 1994). The success of business relationships over the long run are contingent on each partner's investment in each network relationship as well as the partners' ability to communicate effectively throughout the duration of the relationship (Mohr and Nevin 1990). The competitive importance of strong relationships has generated an underlying paradigm shift in marketing and organizational strategy that has redefined the exchange process (Gronroos 1994).

National culture encompasses the values, beliefs, and assumptions that define a distinct way of life of a group of people and is based on the fundamental concepts imparted in early individual development (Hofstede 1980, 1991).

According to Magala (2005) *"it is not just the food that is being genetically manipulated but also organisations, (...) in a cultural way."*

In his book Cross-Cultural Competence Magala (ibid.) states that cross-cultural management is a crucial challenge for the successful development of international business, yet it is often badly understood and poorly implemented. Misunderstandings arise because culture affects both individuals and organizations, from our core values to our table manners. Yet, attempts to understand, explain and interpret these differences have often been hidden by a welter of conflicting theories and paradigms, as well as biases and stereotypes. Magala developed an innovative approach to learning and teaching cultural sensitivity via the exploration of a number of classic film masterpieces.

When studying cultural competences of business firms, we can focus on specific antecedents or conditions which explain variances in cultural competences. On the other hand, various dimensions of cultural competence can be assumed to influence business performance, including both financial and non-financial aspects.

Meeting the demands of globalization is vitally important. Educating for intercultural competence must take a holistic and multi-dimensional approach that focuses both on the often intrapersonal cognitive and affective levels as well as the relational and behavioral levels of interpersonal experience. Universities and business colleges must strive to compete to meet the growing and changing needs and demands of globalization by preparing faculties and students with intercultural competences and skills. (Irving 2008).

5. Diversity and Interorganizational Challenges

Globally competent managers are critical for the future success of all multinational organizations. To address this HR need, many strategic HR departments have initiated global leadership development programs to groom future leaders. There are two inherent assumptions in all global leadership programs. First, that "global competence" can be defined in terms of developmental dimensions. Second, that, once defined, these dimensions can be developed through global experience (Caligiuri, P. and Di Santo, V. 2021).

Diversity in the workplace is a phenomenon of great strategic and operational consequences for both public and private organizations with the expanding globalizations of businesses. The strategic use of language and actions constitutes the communication foundation for organizational change and development (DeLuca and McDowell, 1992; Jackson and Alvarez, 1992).

The benefits of diversity in a global economy are very well recognized (Allen et al. 2008). Managing a diverse workforce is a business imperative yet challenges continue to exist. Organizational members might not recognize the impact they have on others, and how to build their inter-cultural competences. As a consequence the need to develop an instrument that will identify organizational members' skill deficiency in diversity inclusion is essential. If the organizational goal of embedding an inclusive environment is at odds with the values, behaviors, attitudes and feelings of its employees, then the goal will not be fully achieved and the bottom line of the business will be impacted (ibid.)

Dealing with issues of diversity in the workplace - whether they be theoretical or pragmatic - presents a challenge to the management scholar for at least two reasons. First, the concept of diversity is still evolving. Workplace diversity has been seen as a relatively unidimensional construct that dictates the creation of organizational strategies for "managing" or "valuing" it (Jackson and Alvarez, 1992). But examining this diversity requires a multidimensional lens - one which sees it as much an issue of organizational change as it does as a problem of individual differences; as much as a challenge for human resource policy. Second, the meaning of diversity in an organization depends largely on the organizational actors in power and their perspective on diversity, be it humanistic, legal, economic, or marxist. This happens because – as Ranson, Hinings, and Greenwood (1980) observed – an organization is composed of a number of groups divided by alternative conceptions, value preferences, and sectional interests. The analytical focus becomes the relations of power which enable some organizational members to constitute and recreate organizational structures according to their provinces of meaning.

In recent conceptualizations, diversity encompasses the idea of multiculturalism, which involves increasing the consciousness and appreciation of differences associated with the heritage, characteristics, and values of many different groups, as well as respecting the uniqueness of each individual (Morrison, 1992). This interpretation of diversity draws attention to a number of attributes, including but not limited to gender and race (Jackson & Alvarez, 1992), and implies that organizational systems and norms must learn to accommodate a wide range of workers.

Effective communication is one of the key processes in an organization. Its success is not only highly interrelated to the abilities of senders and/or receivers but also to the internal and external environment of an organization as well as to the social network wherein the communication takes place. Communication within an organization is therewith always influenced and sometimes even determined by the corporate vision, the business mission, the corporate culture and of course by corporate guidelines and policies. Therefore, organizational communications has to be a vital part of strategic management – not least as communication processes themselves vitally influence strategic management as well.

According to Bedell and Landesberg (2008), business people who may be confused over the value of public relations – is it hype or helpful? – should consider what Sir Ken Robinson (1950–2020) has learned over more than 30 years in the field. Robinson, president of Robinson Group Communications in Hampden Township, says that PR is a form of organizational communication that hinges on content and credibility. If a business isn't believable in its relationships with the news media and the public, it will falter. Robinson honed his communication skills as a spokesman for Pennsylvania's corrections system in the 1970s and during the Camp Hill prison riots in 1989. He discovered that the news media expect an organization to be organized in its approach to communication but are restive when they feel it's being too controlling. That's the inherent tension in media relations. What should a business person looking to improve his or her communications focus on? "Recognize that communication is a continuing process," Robinson said, "and understands yourself and what you are trying to accomplish" (ibid.).

If in business situation we say nothing, we communicate a lot. Therefore, with regard to this form of miscommunication one can find a lot of complaints in many companies-complaints about being not informed about the strategy, mission and the goals of the company. Especially in post-covid times of unprecedented change, the management has to make clear what their position is about challenges which all employees are aware of in their daily life. Employees and customers expect from business honest information about the why and especially the how of identifying, evaluating, and solving problems along with internal as well as external problems and developments. Human Language Technologies can be definitely part of the improved business communication solution inside the firm and when communicating with customers and society as well with feedback reception.

Communication always takes place in cross-functional directions. Therefore, there has to be ways for employees and the lower or middle management to communicate upwards, acwards, feedbacking and across lateral planes in order to give vital feedback in order to make necessary decision.

Communication behaviour is the way one speaks, listens, reads, and writes. Managers and supervisors give instruction, communicate decisions, discuss problems, and solicit feedback, and so on through speaking, listening, writing, and reading. Their behaviour in speaking, listening, and writing can be easily observed by others and can also affect others' behaviour immediately. For example, a supervisor may ask a worker to change his or her job performance in either a relationship-oriented manner or in a task-oriented manner. (Zhao and Parks 1995).

In a task-oriented manner, the supervisor might say, "What are you doing? Do it this way, or you will not be allowed to work here." By contrast, in a relationship-oriented manner, the supervisor might say, "I think this may be a better way to do this job. We've found that it is easier to do this way and that it also saves time. Why don't you try this way and see if it works any better."

The task-oriented approach communicates instruction to a worker without respect and empathy, so the worker feels bullied into complying and develops a strong dislike to the supervisor. However, the relationship-oriented approach communicates instruction to a worker with respect and empathy, so the worker feels treated as a responsible and intelligent person who will cooperate if given good ideas. Consequently, this worker likes his or her supervisor, works harder, and cooperates well with not only the

supervisor but also co-workers. Therefore, managers' and supervisors' communication behaviour is closely related to their management success. Communication behaviour is even more important in managing an intercultural workforce and in handling international business (ibid.).

In addition to the above overview of various challenges for business communication of intercultural and inter-organizational nature, Xxxxxxx and Yyyyy notice other challenges then those triggered by the Covid 19 pandemic, wars or human migrations at a large scale

It follows, that due to globalization and new technologies, present-day firms operate in a totally different mode than before. Increasing competition and rapid technological development have led corporations to focus on their core competence and outsource other activities to those countries and places where it is considered not only the most economical but also most safe. Near shoring, utilizing various technological advances and even bringing back production from not so safe presently countries from overseas may be becoming the new norm. Consequently, businesses are increasingly dependent on interorganizational cooperation and networks, as well as disruptive short innovation cycles, employing design thinking in management and innovation, crossovers, and citizen empowerment due to the extensive communication via distance internet supported communication utilizing various synchronous, meta-synchronous and asynchronous platforms, Internet of Things in business and management, use of Artificial Intelligence and Human Language Technologies all playing a vital role in various business and managerial communication processes and in creating of the new business and economic model paradigm.

5. Technological, Public Security and Post-Covid Challenges

The significant challenges of post-pandemic, emergency management and security supply chain disruptions require new ideas and solutions especially in fields of HLT-Human Language Technologies, Artificial Intelligence (AI) and Internet of Things (IoT) (Vetulani and Osiński, 2017; Moosavi et al., 2022).

The COVID-19 pandemic has made a significant and long term impact on various supply chains (SCs). All around the world, the COVID-19 pandemic affects different dimensions of SCs, including but not limited to finance, lead time, demand changes, and production performance. There is an urgent need to respond to this grand challenge. The catastrophic impact of the COVID-19 pandemic prompted scholars to develop innovative SC disruption management strategies and disseminate them via numerous scientific articles. However, there is still a lack of systematic literature survey studies that aim to identify promising SC disruption management strategies through the bibliometric, network, and thematic analyses (ibid.). Soon after the pandemic Covid 19, it has become clear that the world had entered a period if global crisis on a scale not seen for more than 70 years and this presented new hitherto unknown challenges for humanity, especially for higher education sector (Vetulani and Juskowiak 2022).

6. Discussion – challenges, technological and social answers to existing issues

Due to globalization, recent trends of deglobalization and a new technology, present-day firms and society, especially in the post-covid arena, operate in a totally different mode than before. Some people, in particular, technologically excluded working-poor vulnerable members of the population, seniors, may become not only technologically marginalized, but this may also lead to their social isolation. Increasing competition and rapid technological development including existing and growing use of Human Language Technologies have led firms and corporations to focus on their core competence and outsource other activities to those countries and places where it is considered most economical. Consequently, businesses are increasingly dependent on join work using various web-based and interorganizational cooperation and networks, as well as undergoing innovation through the destruction of the old office traditional communication structures. Many workers especially those employed in traditional production industries and sectors, as those dependent on heavy use of labour and use of fossil fuels including mining of coal, oil and gas, may become displaced as a side effect of the transformation to green energy industries.

Taking into consideration the present and future challenges and possible coming changes it is worth to strengthen and support the new modes and models of communication in business and society including possible support coming from HLTs in the era of AI.

HLTs have a significant potential in business management, also in training workers into new economy, environment friendly jobs and in transforming entire industries and sectors into the new greener and more environmentally friendly economy. Communication in business both interpersonal and among institutions, has always played a key role. While the globalization model assuming unification seems to facilitate business communication, moving away from this model generates new challenges for communication technologies, among which Human Language Technologies (HLTs) and AI play a special role. Regardless of which of these trends turns out to be dominant in the short and medium term, communication in a natural language will play a key role at all levels of management for a long time, including language interaction between humans and their technological environment (human-machine communication). In order to better face the globalization and de-globalization related challenges, a thorough analysis of practical needs is crucial.

How can we advance HLTs applications to make business management communication processes human friendly and more socially inclusive ?

7. Conclusion and frontiers of the future development and research

The future discussion should not be limited to problems related to the use of the modern state-of-the-art business communication technologies, but welcomes discussion on vision of the future needs, emerging ideas and novel solutions involving Human Language Technologies. It is

worth to address and effectively manage these developments and changes to make resources more evenly allocated, especially to the under developed and developing parts of the globe.

8. Recommendations

In the opinion of the authors there is a role for the governments to support the dissemination of tools and technologies supporting communication not only in business and administration but first and foremost in the fields of healthcare, social and support services addressed to seniors, war victims, migrants and underprivileged people. There is a definite role of HLT in provision of prescreening services in healthcare delivery. Supporting medical service delivery communication in multilingual and multicultural millieu, is essential for improving the life conditions in the situation and scarcity of medical services. Such a support for the dissemination is essential for innovativeness of society that is in fact the leading indicator of the progress. It can aid in combating communication exclusion for small linguistic and cultural communities.

References

Allen, R., Dawson, G., Wheatley, K., and White, C. (2008). Perceived diversity and organizational performance. In *Employee Relations* 30 (1) 2008.

Bedell, D. and Landesberg, P. (2008). Communication matters, whether it's with employees, customers or community, *Central Penn Business Journal,* 10/3/2008, Vol. 24 Issue 41.

Baldwin, R and Evenett, S.J. (2020). *COVID-19 and Trade Policy: Why Turning Inward Won't Work*. VoxEU.org eBook, CEPR Press. https://cepr.org/voxeu/columns/pandemic-adds-momentum-deglobalisation-trend

Caligiuri, P. and Di Santo, V. (2021). Global Competence: What Is It, and Can It Be Developed Through Global Assignments?, pp. 27-35.

DeLuca, J.M. and McDowell, R.N. (1992). Managing diversity: A strategic 'grass-roots' approach. In S. E. Jackson & Associates (Eds.), *Diversity in the workplace* (pp. 227-247). New York: The Guilford Press.

Ferraro, G.P. (1996). The need for linguistic proficiency in global business. In. *Contingencies and Crisis Management*, 16(3), pp 143-153.

Gronroos, C. (1994). From Marketing Mix to Relationship Marketing: Towards a Paradigm Shift in Marketing, *Management Decision,* 32 (2), pp. 4-20.

Hofstede, G. (1980). *Culture's Consequences.* Beverly Hills. CA: Sage Publications.

Irving, J.A. (2008). Educating global leaders: Exploring intercultural competence in leadership education. In: *Journal of International Business and Cultural Studies.* pp. 1-14. https://www.aabri.com/manuscripts/09392.pdf.

Irwin, D A (2020). *Trade Policy Disaster: Lessons from the 1930s*, MIT Press. https://cepr.org/voxeu/columns/pandemic-adds-momentum-deglobalisation-trend

Jackson, S. E. and Alvarez, E. B. (1992). Working through diversity as a strategic imperative. In S.E. Jackson & Associates (Eds.), *Diversity in the workplace* (pp. 13-29). New York: The Guilford Press.

Joutsenvirta, M. and Uusitalo, L., (2009). Cultural Competences: An Important Resource in the Industry–NGO Dialog. In: *Journal of Business Ethics*, **91**, Springer (2010), pp.379–390.

Magala, S. (2005). *Cross-Cultural Competence*. Taylor and Francis, London.

Mohr, J. and Nevin, J.R. (1990) Communication Strategies in Marketing Channels: A Theoretical Perspective. *Journal of Marketing,* 54, pp. 36-51. http://dx.doi.org/10.2307/1251758

Moosavi, J., Fathollahi-Fard, A.M. and Dulebenets, M.A., (2022). Supply chain disruption during the COVID-19 pandemic: Recognizing potential disruption management strategies., *International Journal of Disaster Risk Reduction*, Elsevier, Volume 75, 1 June 2022.

Morrison, A. (1992). *The new leaders.* San Francisco: Jossey-Bass Publishers.

Odrakiewicz P. (2009). Busines English as an intellectual bridge-management of the syncretic case study, organizational changes in the management of education and blended learning for students of Business English in an intercultural world. In: *Organization and Management* no 3(7) 2009, Wyd. Politechnika Śląska.

Ranson, S., Hinings, B. and Greenwood, R. (1980). The Structuring of Organizational Structures. In: *Administrative Science Quarterly*, vol. 25, pp. 1-17. https://doi.org/10.2307/2392223

Vetulani, Z. and Osiński, J. (2017). Intelligent Information Bypass for More Efficient Emergency Management, In: *Computational Methods in Science and Technology* 23(2), pp. 105–123. https://doi.org/10.12921/cmst.2017.0000019

Vetulani, J. and Juskowiak, E. (2022). COVID 19 – A New Challenge for Academic Teaching. In: *INTED2022 Proceedings,* pp. 246-255. https://doi.org/10.21125/inted.2022.0129

Witherspoon, P.D. and Wohiert. K.L. (1996). An Approach to Developing Communication Strategies for Enhancing Organizational Diversity. In: *International Journal of Business Communication,* Volume 33 Issue 4, October 1996, p. 375.

Zhao, J.J., Parks, C. (1995) Self-Assessment of Com munication Behavior: An Experiential Learning Exercise for Intercultural Business Success. In: *Business and Professional Communication Quarterly.* SAGE Publishing. https://journals.sagepub.com/doi/abs/10.1177/108056999505800106?journalCode=bcqd

Improving Performance of Affect Analysis System by Expanding Affect Lexicon

Lu Wang[1], Michal Ptaszynski[1], Pawel Dybala[2], Yuki Urabe[3], Rafal Rzepka[4], Fumito Masui[1]

[1]Text Information Processing Laboratory, Kitami Institute of Technology, Kitami, Japan
m2153308016@std.kitami-it.ac.jp, {michal,f-masui}@mail.kitami-it.ac.jp
[2]Jagiellonian University, Krakow, Poland, [3] Independent Researcher
{paweldybala1,yuki.urabe.1011}@gmail.com
[4]Langauge Media Laboratory, Hokkaido University, Sapporo, Japan
rzepka@ist.hokudai.ac.jp

Abstract

In this paper, we conducted research to improve the performance of ML-Ask affect analysis system for Japanese by expanding its affect lexicon. To expand the affect lexicon, we proposed a method to add and integrate a new dictionary of emotive expressions into the database of the original system. We investigated the differences between the emotion types in the new dictionary and the emotion types in the original affect lexicon used in the system. We proposed a method for disambiguating the out-of-vocabulary (OOV) emotive expressions in the new dictionary of emotive expressions. The method was used to prepare example sentences with the OOV emotive expressions and a survey was then sent to native speakers of Japanese in the form of a questionnaire. The results obtained from the questionnaire were analyzed to determine the correct emotion types to unify the OOV emotive expressions. We incorporated the new disambiguated emotive expressions into the affect lexicon of the affect analysis system and evaluated the performance of the system.

Keywords: Affect Analysis, Emotive Expressions, Affect Lexicon, EDO 2023

1. Introduction

As social networking services (SNS) have become popular in recent years, Internet users have been posting opinions, feelings, reviews, and criticisms about things, services, products, and events on the Internet. By accurately detecting, efficiently collecting, and further analyzing user emotions and other information contained in these Internet postings, it will be possible to automatically collect public opinions about these things, services, products, and events, and to make efficient use of the information to improve those products and services. For this reason, along with the spread of social media, research on the automatic extraction and collection of sentiment information from text data contained in these media has gained in popularity. An example is ML-Ask[1] (Ptaszynski, 2009; Ptaszynski et al., 2017), an affect analysis system for textual input in Japanese.

However, the affect lexicon contained in ML-Ask was constructed based on the Dictionary of Emotive Expressions (Nakamura, 1993), which was created in the 1990s, and thus many contemporary emotive expressions have not yet been included. Furthermore, the database has not been updated for many years, and it is still unable to respond to new emotive expressions that have started to be used since the spread of the Internet. To solve this problem, it is necessary to collect new emotive expressions and expand the database.

Natural language processing methods for affect analysis include lexicon-based methods that use pre-prepared lexicons of emotive expressions, rule-based methods that use lists of rules for emotive expressions extracted from emotive sentences, and machine learning-based methods that use machine learning algorithms to automatically learn rules.

Lexicon-based and rule-based methods use a pre-built database, which contains a lexicon of words, phrases, and sentence patterns that express positive and negative emotions such as “happy” and “sad,” etc. When analyzing a new input sentence, this lexicon is used as a reference. If an input sentence contains one of these words, the sentiment score is set to either negative or positive, and the sentence is annotated with the type of emotion expressed by the word (e.g., “happy” to “joy”, “afraid” to “fear”, etc.).

From a technical point of view, methods using lexicons and predefined rules, unlike machine learning-based methods, do not use complex computational algorithms and perform simple pattern matching, thus they have the advantage of being able to analyze large numbers of documents in a short time. However, they also have some disadvantages. One is not being able to analyze new terms that are not in the lexicon because the analysis is performed using only the limited data in the lexicon. Another is not being able to properly process idiomatic phrases and emotive expressions consisting of multiple words distributed in a sentence, whose nuances are ambiguous depending on the wider context.

On the other hand, machine learning-based methods learn from a large amount of text data and calculate sentiment scores, and have the advantage of being more accurate than rule-based and lexicon-based methods if a large amount of data can be prepared. However, it is an expensive method, as it requires collecting such large data sets, performing data cleaning, accurate expert label annotation, and post-processing. In addition, when using machine learning, it

[1]https://github.com/ptaszynski/mlask

has been pointed out that the more classification classes there are, the less accurate the classifier will become (Gupta et al., 2014). Furthermore, people often express multiple emotions in natural speech, even in one sentence or a single utterance, or in writing, etc. Therefore even a classifier that supports multiple classes can output only one final class, making it difficult to analyze crowded and complex emotional states.

From the above, it can be seen that lexicon-based and rule-based affect analysis methods are still highly applicable. However, the performance of such methods is highly dependent on the lists of words and rules contained in lexicons, and it is necessary to add new expressions to the original affect lexicon to ensure and improve system performance. Therefore, the purpose of this study was to propose a method of expansion when adding emotive expressions from a new widely available dictionary of emotive expressions to the original affect lexicon.

2. Adding New Dictionary of Emotive Expressions

2.1. Differences in emotion categories in dictionaries

To update and expand the database of emotive expressions in ML-Ask affect analysis system, we decided to add and integrate a new dictionary of emotive expressions, namely, Hiejima's "A Short Dictionary of feelings and Emotions in English and Japanese" (Hejima, 1995). In this dictionary, Japanese emotive expressions are classified into eight types: expressions of joy, expressions of love, expressions of anger, expressions of suffering, expressions of sadness, expressions of blame, expressions of enjoyment, and expressions of surprise. On the other hand, the dictionary used in the database of ML-Ask is Nakamura's dictionary (Nakamura, 1993), which contains ten emotion categories: joy, fondness, relief, gloom, dislike, anger, fear, shame, excitement, and surprise. At first glance, there are similar types of emotion: joy and expressions of joy, fondness and expressions of love, anger and expressions of anger, gloom and expressions of sadness, and surprise and expressions of surprise. In Nakamura's dictionary, however, the type of joy includes not only expressions of joy, but also expressions of enjoyment. For a simple example, "enjoyable ("tanoshii" in Japanese)" can be found in both "joy" and "expressions of enjoyment". Therefore, to more precisely classify the emotive expressions, we had to verify how many existing emotive expressions there are in each emotion type in Hiejima's dictionary, and how those two dictionaries align together regarding the understanding of emotion categories.

2.2. Checking for existing emotive expressions

Emotive expressions that were not found in the original dictionary are classified as out-of-vocabulary (OOV) expressions.

As a result, the numbers and percentages of overlapping emotion expressions per category/type are the highest for expressions of joy, expressions of suffering, and expressions of sadness, and are higher than that for OOV expressions. Originally, we would have classified them into the type with the highest percentage, but after looking at these results, we decided that this was inappropriate because we found that almost all of the expressions were OOV in other emotion types. All OOV expressions had to be processed in some other way.

2.3. Processing OOV expressions

2.3.1. Basic Concept

Sakai et al. (2019) designed a questionnaire survey on the type of emotion expressed by each emoticon generated by their automatic emoticon generation algorithm (Sakai et al., 2019). In our research, to determine the most appropriate type of emotion expressed by the emotive expressions of uncertain emotion types from Hiejima's dictionary, we decided to prepare a similar questionnaire survey, containing – not emoticons, but rather – example sentences containing those OOV emotive expressions and conduct the questionnaire among native speakers of Japanese. First, for each OOV word, 10 example sentences containing it were prepared. In the questionnaire, respondents were asked to choose the type of emotion expressed in the example sentences.

However, as we found, there were 642 OOV expressions overall. If all of these OOV expressions were to be processed using the above method, 6420 example sentences would need to have been first prepared. This would not only be burdensome in terms of preparing the dataset of example sentences but would also require a considerable amount of time to complete the questionnaire survey by the participants, resulting in fatigue and errors. Since the less time it takes to process the OOV expressions, the better, we needed to reduce the number of example sentences to 5 (since the maximum number of example sentences for a word in Hiejima's dictionary is 5) and reduce the number of OOV expressions in some other way to make the study realizable in the given time constraints.

2.3.2. Reducing number of OOV expressions

We again studied Hiejima's dictionary and found that most of the Japanese emotive expressions in Hiejima's dictionary are grouped into smaller groups of 4 synonymous emotive expressions. Specifically, in this dictionary, the main expression is highlighted in bold and is followed by a single parenthesis, inside of which there are 3 (very rarely 2 or 4) emotive expressions that have a similar meaning to the main emotive expression. In other words, there are typically 4 synonyms in each group. Knowing that there are synonyms, we first had to process the synonyms and reconfirm the existing emotive expressions before reducing the number of OOV expressions. Also, some expressions were duplicated, so they had to be dealt with as well. The number and percentages of existing emotive expressions in each emotion type in Hiejima's dictionary after processing the synonyms and duplicate expressions were shown in Table 1 above.

Emotion categories		Nakamura's dictionary										
		Joy	Fondness	Relief	Gloom	Dislike	Anger	Fear	Shame	Excitement	Surprise	OOV
Hiejima's dictionary	Joy	115 (82.73%)	4 (2.88%)			4 (2.88%)				4 (2.88%)		12 (8.63%)
	Love	4 (2.45%)	59 (36.20%)	11 (6.75%)	8 (4.91%)			4 (2.45%)		1 (0.61%)		76 (46.63%)
	Anger					39 (22.16%)	61 (34.66%)	3 (1.70%)		13 (7.39%)		60 (34.09%)
	Suffering		1 (0.68%)		4 (2.72%)	101 (68.71%)	2 (1.36%)	9 (6.12%)		10 (6.80%)		20 (13.61%)
	Sadness				36 (31.03%)	59 (50.86%)		1 (0.86%)		4 (3.45%)		16 (13.79%)
	Blame	4 (2.26%)				27 (15.25%)	8 (4.52%)	4 (2.26%)	4 (2.26%)			130 (73.45%)
	Enjoyment	83 (64.34%)		6 (4.65%)		4 (3.10%)						36 (27.91%)
	Surprise					4 (5.00%)		4 (5.00%)			40 (50.00%)	32 (40.00%)

Table 1: Number and percentages of existing emotive expressions after processing synonyms and duplicate expressions.

As shown in Table 1, there were a total of 382 OOV expressions. The number of OOV expressions was reduced by a little less than half compared to the number of that before. Furthermore, since there were synonyms among the OOV expressions, if the emotion type of only the main emotive expression was determined for a group of emotive expressions, the emotion types of the other synonyms can be also assumed to belong to the same emotion type as the main emotive expression. In this way, after eliminating the synonyms and duplicates, the number of OOV expressions that must be processed decreased to 93. Thus, the number of OOV expressions was reduced to almost a quarter, which considerably shortened the process of preparing the example sentence dataset and the questionnaire survey.

2.4. Questionnaire survey

2.4.1. YACIS Large-scale Japanese Blog Corpus

To process the OOV emotive expressions in Hiejima's dictionary in the questionnaire survey, we first had to determine the source text data for the example sentences. Obviously, it is best to use raw Japanese sentences to process Japanese expressions. In this case, we used the YACIS blog corpus, which is currently the largest Japanese blog corpus with 5.6 billion words (Maciejewski et al., 2010; Ptaszynski et al., 2012). The YACIS corpus, made in 2010, was collected from the Ameba blogs and when it was created it included roughly one-third of the Ameba blog. Therefore, we considered it to be suitable for application to this study.

2.4.2. Preparation of Examples for Questionnaire

In preparation for the questionnaire, we needed to collect proper examples. Firstly, the example sentences were collected from Hiejima's dictionary, which already contains some examples for each emotive expression. If this was not enough, we used example sentences from the YACIS corpus, and if this was still not enough, we searched for sentences on Twitter containing the specific expression.

However, if the sentences extracted from YACIS corpus or Twitter were too long or too short, it could cause additional bias in the responses. To extract example sentences from the YACIS corpus, we first needed to determine how long the sentences should be. We aimed at a length similar to the examples from the dictionary.

To specify the optimal sentence length, the example sentences from Hiejima's dictionary were segmented into words using Mecab (Kudo, 2005), and the number of words in each sentence was calculated and used as the length of the sentence. Then the mean and standard deviation of the example sentence lengths in the dictionary were calculated to be 10.96 and 2.80, respectively. The mean plus or minus standard deviation was used to determine the final optimal range of sentence lengths to be extracted from the YACIS corpus. Based on the calculation, the range of sentence lengths should be from 8 to 13 words. Finally, we extracted the example sentences from the YACIS corpus of the specific lengths using regular expressions.

2.4.3. Questionnaire Setup

The setup of the questionnaire was designed as follows. Firstly, 465 example sentences were prepared for 93 OOV emotion words (five sentences per word). All the example sentences were distributed into four sets on average, and four sets of questionnaires were created. Respondents were asked to respond to each sentence in terms of the type of emotion expressed in the sentence.

The questionnaire was administered in a multiple-choice format. However, if all the emotion types used in the database of ML-Ask were used as options, the respondents would be overloaded, and this would affect the quality of their responses, so only four options were given. Specifically, the questionnaire responses were given four options: (1) the original emotion type (treated as the potential correct answer), (2) a similar emotion type, (3) the opposite emotion type, and (4) other.

The similar and opposite emotion types were selected by referring to ML-Ask emotion types mapped onto the two-dimensional emotion model (Russell, 1980), as in (Ptaszynski et al., 2017). Since the labels of the emotion types in the original database include Japanese characters that are not usually used (e.g., "iya", or "takaburi" in Japanese), instead of using just those labels we referred to the database of ML-Ask and used easy-to-understand ex-

pressions as the options. Thus, for example, if the correct answer was "happy" the options were (1) happy, (2) like, (3) dislike, and (4) other. In addition, we randomized the options from (1) to (3) for each example sentence so that the correct option would not be easily guessed by the respondents . In case no emotion type matched the respondent's idea, the option "other" was given, and the respondent was asked to describe the emotion type that matched his/her idea. Thus, four sets of questionnaires were created and surveyed.

2.4.4. Initial results of questionnaire

The following is a description of the responses submitted by the respondents participating in the survey. Before responding to questions regarding the emotion types, we first asked them to indicate their nationality, gender, and age. To investigate the emotion types of Japanese texts, we will focus on responses from Japanese respondents only. In the future, we aim to analyze the responses of non-Japanese respondents as well to see the differences in how Japanese language learners of various nationalities perceive Japanese emotive expressions. As a preliminary survey, we aimed for a minimum of 10 respondents in each set, but the number of respondents dropped to 8 in the 1st set, and then to 3 in the 4th set. The percentage of male respondents was higher than that of female respondents in each set, thus we would like to adjust the percentage of female respondents to be more equal in the future to increase statistical reliability and investigate the differences in the perception of emotive expression between male and female respondents. The statistics on respondents per set are shown in Table 2.

Furthermore, in the initial questionnaire run, the respondents commented that there were few options for the emotion labels and that the objective descriptions made it difficult to determine the type of emotion. Therefore, we had to improve the emotion labels, provide additional explanations about the questionnaire, change a few difficult-to-understand example sentences, and conduct the questionnaire once more.

2.4.5. Improvement of the questionnaire

To improve the labels of the emotion types, three easy-to-understand expressions were selected by referring to the emotive expression database of ML-Ask, and the emotion type of ML-Ask to which they belonged was added in front of them. Thus, for example, if the potential correct emotion type was joy, the choices were (1) [joy] (enjoyment, happiness, fun), (2) [like] (love, dear, addicted to), (3) [dislike] (aversion, disgust, unpleasant), and (4) others. The example sentences used in the questionnaire were extracted from dictionaries and blogs (YACIS corpus), therefore many sentences directly expressed the speaker's emotions. However, other sentences were descriptions of a person's behavior or short sentences with a narrow range of context. Since it was difficult to directly judge the emotion type of these sentences, we asked the participants to simply answer which emotion they felt is referred to when they read this sentence, rather than from the standpoint of the speaker expressing it. Then, several difficult-to-understand example sentences were changed. In this way, four sets of questionnaires [2] were improved and re-surveyed.

2.4.6. Final results and summary of both surveys

All four sets of questionnaires in the 2nd survey were answered by university students in their 20s. The information for each set of respondents is shown in Table 3.

We had to integrate the results obtained from both surveys. Specifically, first, the emotion labels from the previous survey were changed to the current labels. Then, all of the "other" responses were initially processed by ML-Ask affect analysis system to fish out those emotion labels that could be unified automatically. Responses that couldn't be processed by the system were considered outliers. In this way, the percentage of each emotion type for each emotive expression could be calculated.

The survey results show that most of the emotive expressions tend to have a high percentage of association with two emotion types. However, there still remained expressions with more than one or two responses, for which the emotion label was ambiguous. To disambiguate those emotion labels, we applied a method similar to the one proposed by Ptaszynski et al. (Ptaszynski et al., 2020a; Ptaszynski et al., 2020b), for automatic estimation of ambiguity in the meaning of emoticons, with the assumption that it may be possible to estimate ambiguity in the meaning of whole texts as well. However, since there was no emotive expression with the highest percentage of the label "other" in the current survey, we determined that there was no need to estimate the ambiguity of the emotive expressions in this case. To determine the emotion type to which an emotive expression belongs, we had to determine a threshold for the percentage of emotion types. This time, to cover all emotive expressions as much as possible, each emotive expression was classified into the emotion type with the highest percentage if the highest percentage of the emotion type was 67.5% or higher. If the highest percentage of the emotion type is less than 67.5%, the emotive expression is classified into the emotion type with the first and the second highest percentage. Thus, we were able to classify all of the emotive expressions except one emotive expression whose sum of the first and second highest percentage was less than 67.5%.

3. Evaluation Experiments

To confirm the coverage of Hiejima's dictionary and the overall performance of the integrated system, we conducted 2 evaluation experiments. To conduct evaluation experiments, it was first necessary to create test data. In these evaluation experiments, two sets of test data created in previous studies were combined and used. Specifically, we used the dataset used by Sakai et al. (Sakai et al., 2019) for the affect analysis of emoticons and the dataset created

[2] https://forms.gle/PDhbWR7E9rVGjXP69
https://forms.gle/zb5uJTHjMyognej9A
https://forms.gle/jHh3q35DrcwmveBfA
https://forms.gle/uEPGxxBZukZukEJ26

set	Number of Respondents	Age (generations/%)					Percentage by gender	
		10s	20s	30s	40s	50s	Male	Female
1	8	12.5%	75%	0%	0%	12.5%	87.5%	12.5%
2	6	16.7%	66.7%	0%	0%	16.7%	83.3%	16.7%
3	4	0%	100%	0%	0%	0%	100%	0%
4	3	0%	100%	0%	0%	0%	100%	0%

Table 2: Statistics on respondents per set in the 1st survey.

set	Number of Respondents	Age (generations/%)					Percentage by gender	
		10s	20s	30s	40s	50s	Male	Female
1	7	0%	100%	0%	0%	0%	100%	0%
2	10	0%	100%	0%	0%	0%	90%	10%
3	12	0%	100%	0%	0%	0%	83.3%	16.7%
4	15	0%	100%	0%	0%	0%	100%	0%

Table 3: Statistics on respondents per set in the 2nd survey.

by Ptaszynski et al. (Ptaszynski et al., 2017) for the development of ML-Ask. The test data contained 280 emotive and non-emotive sentences labeled with 10 emotion types, which were treated as correct data for evaluation.

3.1. Experiment 1: Coverage of Hiejima's Dictionary

3.1.1. Experimental setup

The setup for the evaluation experiment was as follows. First, we created a list of candidate emotive expressions using only the emotive expressions from Hiejima's dictionary. Next, we added each of the prepared lists to the affect lexicon of ML-Ask, evaluated them on the test data using the ML-Ask baseline model, and checked for changes in results.

Furthermore, when checking the accuracy of the affect analysis, we first checked whether the output matched the correct data completely (exact match rate). However, since it is impractical to obtain the exact match rate when multiple emotion types are expressed in a single sentence from the correct data, we also checked whether at least one emotion type per sentence was detected (partial match rate) in addition to the exact match rate. In addition to the specific emotion type, we also used Russell's two-dimensional model of emotion (Russell, 1980), which is incorporated in ML-Ask, to determine whether the emotion expressed in the sentence was positive or negative, that is, the probability that the polarity of the emotion matched, as well as the probability that the activation dimension of the emotion was consistent. In addition, we also checked the probability that the system incorrectly extracted emotions from sentences that did not express emotions. The results were shown in Table 4.

3.1.2. Results and Discussion of Experiment 1

The results in Table 4 show that all indicators in this experiment were significantly reduced compared to the ML-Ask baseline. This suggests that the coverage of Hiejima's dictionaries is low since it is also consistent with the fact that there are about 1100 emotive expressions in Hiejima's dictionary while there are about 1000 more emotive expressions in Nakamura's dictionary. In other words, there is a significant difference between the emotive expressions in Hiejima's dictionary and those in the database of ML-Ask. Therefore, we expected that integrating Hiejima's dictionary into ML-Ask will improve the accuracy rate. Consequently, we integrated the Hiejima dictionary into ML-Ask and conducted an additional experiment.

3.2. Experiment 2: Expansion of the Affect Lexicon

In the additional experiment, we added emotive expressions from Hiejima's dictionary to the database of ML-Ask by subtracting duplicated emotive expressions. The results of the re-evaluation after the dictionary was integrated are shown in Table 4.

3.2.1. Results and Discussion of Experiment 2

In the results of Experiment 2, the partial match rate and the polarity match rate were improved by 2% pt. (percentage points), and 1% pt., respectively. However, the exact match rate decreased by 4% pt., while the polarity match rate also decreased, but only by about 1% pt. The incorrect emotion extraction rate was unchanged from that of the ML-Ask baseline model, suggesting that the improvement was successful.

An investigation of the reasons for the decrease in the exact match rate revealed that emotive expressions were added to some of the analysis results, and the number of emotion types analyzed was also increased. This suggests that the more emotive expressions there are in the database, the lower the percentage of exact matches.

The experiment also suggests that other available dictionaries should also be added to improve the affect analysis system. Moreover, experiment results show the limitations of such manually collected dictionaries. Therefore,

	ML-Ask baseline	Hiejima's Dict.	Integration
partial match rate	66%	19%	68%
exact match rate	40%	11%	36%
polarity dimension match rate	65%	21%	64%
activity dimension match rate	58%	18%	59%
incorrect extraction rate	3%	2%	3%

Table 4: Results of experiments 1 and 2.

a method should be developed to collect emotive expressions not only manually but also automatically. In addition, the test data for the evaluation experiment needs to be increased to show a more diverse spectrum of how emotions are expressed.

4. Conclusions and Future Work

In this paper, we expanded the affect lexicon of a lexicon-based affect analysis system by extracting emotive expressions from the new dictionary of emotive expressions and adding them to the system manually.

According to the results of the experiments, we can conclude that we succeeded in improving the performance of the system, although only by a small amount (2% pt.).

In the future, we plan to integrate another available dictionary of emotive expressions to improve the performance of the system even further. We also plan to propose a method of automatic expansion of the affect lexicon.

References

Gupta, Maya R, Samy Bengio, and Jason Weston, 2014. Training highly multiclass classifiers. *The Journal of Machine Learning Research*, 15(1):1461–1492.

Hejima, Ichiro, 1995. *A Short Dictionary of feelings and Emotions in English and Japanese*. Tokyodo Shuppan.

Kudo, Taku, 2005. Mecab: Yet another part-of-speech and morphological analyzer. *http://mecab. sourceforge. net/*.

Maciejewski, Jacek, Michal Ptaszynski, and Pawel Dybala, 2010. Developing a large-scale corpus for natural language processing and emotion processing research in japanese. In *Proceedings of the International Workshop on Modern Science and Technology (IWMST)*.

Nakamura, Akira, 1993. *Kanjo hyogen jiten [Dictionary of Emotive Expressions] (in Japanese)*. Tokyo: Tokyodo Publishing.

Ptaszynski, Michal, 2009. Affecting corpora: Experiments with automatic affect annotation system-a case study of the 2channel forum. In *Proceedings of The Conference of the Pacific Association for Computational Linguistics 2009 (PACLING-09), Hokkaido University, Sapporo, Japan, September 1-4*.

Ptaszynski, Michal, Pawel Dybala, Rafal Rzepka, Kenji Araki, and Fumito Masui, 2017. Ml-ask: Open source affect analysis software for textual input in japanese. *Journal of Open Research Software*, 5(1):16.

Ptaszynski, Michal, Pawel Dybala, Rafal Rzepka, Kenji Araki, and Yoshio Momouchi, 2012. Yacis: A five-billion-word corpus of japanese blogs fully annotated with syntactic and affective information. In *Proceedings of the AISB/IACAP world congress*.

Ptaszynski, Michal, Fumito Masui, and Naoto Ishii, 2020a. Automatically estimating meaning ambiguity of emoticons. In *Biologically Inspired Cognitive Architectures 2019: Proceedings of the Tenth Annual Meeting of the BICA Society 10*. Springer.

Ptaszynski, Michal, Fumito Masui, and Naoto Ishii, 2020b. A method for automatic estimation of meaning ambiguity of emoticons based on their linguistic expressibility. *Cognitive Systems Research*, 59:103–113.

Russell, James A, 1980. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161.

Sakai, Tomoaki, Michal Ptaszynski, and Fumito Masui, 2019. Study on potential of automatic emotion generation based on emoticon parts. *The 33rd Annual Conference of the Japanese Society for Artificial Intelligence*, JSAI2019:2E4OS903–2E4OS903.

Translation Memory Principle in Neural Machine Translation: A Multilingual and Multidirectional Comparison

Yaling Wang[1], Bartholomäus Wloka[2], Yves lepage[1]

[1] Waseda University, Graduate School of Information, Production and Systems
yaling.wang@moegi.waseda.jp yves.lepage@waseda.jp
[2] UniversityofVienna,CentreforTranslationStudies
bartholomaeus.wloka@univie.ac.at

Abstract

Neural Machine Translation (NMT) has made significant progress in recent years, while Translation Memories (TMs) have long been used by translators as a tool to suggest similar translations. Integrating TM into NMT has also been proved to improve translation quality. We propose a simple method to leverage the advantage of TM in NMT without altering the model architecture. We retrieve similar sentences covering a source sentence and use them to enrich the input of an NMT system. Our results show that our method can outperform a baseline model in some cases, which shows that similar sentences offer more contextual information than a baseline model without retrieval. While conducting extensive experiments with this approach, we found significant differences depending on language pairs and translation directions. We present our findings and suggest possible reasons for these differences.

1. Introduction

Machine Translation (MT) has been steadily increasing in quality, especially after the emergence of Neural Machine Translation (NMT). NMT aims to build a single, extensive neural network that reads a sentence and outputs a correct translation, unlike previous translation approaches, which integrated many sub-components, tuned separately. NMT systems are usually based on the encoder-decoder model. An encoder reads and encodes a source sentence into a vector representation, and a decoder then outputs a translation from the vector representation. This boosted the quality of machine translation significantly and has increased the fluency of machine translated sentences. These advantages however are outweighed by (a) the poor interpretability of NMT – errors are difficult to interpret, i.e., to trace back to the training data – and (b) its high computational cost.

Translation Memories (TMs) are tools used by human translators. They contain parallel sentence pairs of high quality translations. Given a sentence to translate, a TM retrieves the most similar sentence in the source language that contains large common or similar parts. The corresponding sentence in the target language is returned to the translator. In this way, the translator only needs to modify the unmatched parts to complete the translation. This results in a variant of Post-Edited Machine Translation (PEMT). A main advantage of this approach is that it ensures consistency and interpretability ((a) above) across translations because common or similar parts in sentences can easily be identified.

It is natural to think that interpretability, as offered by TMs, would be beneficial to NMT.

2. Related Work

Past research (Federico et al., 2012) already proposed to combine the advantages of TM (mainly for interpretability, (a) above) with MT (efficiency). Practical methods have been proposed to achieve closer integration with NMT. For example, an additional encoder can be added to an NMT architecture specifically for TM matches (Cao and Xiong, 2018). The decoding algorithm can be modified to incorporate retrieved strings (Gu et al., 2018). An easy-to-implement TM–NMT integration has been proposed by (Bulté and Tezcan, 2019): they concatenate the target-language side of matches retrieved from a TM with the sentence to translate. This only involves data pre-processing and augmentation, and is thus compatible with different NMT architectures. Retrieval can be extended to include semantically related translations, thanks to the use of distributed representations (Xu et al., 2020). (He et al., 2021) presented a fast and accurate approach for TM-based NMT which can be applied to general translation tasks besides TM-specialized tasks. This led to efficient training and inference. Also, the parameters are effectively optimized through a novel training criterion. All of the approaches above were shown to lead to a significant increase in the quality of MT outputs.

3. Method

In this paper, we propose a method that utilizes the TM principle in conjunction with an NMT system without altering the model architecture. This makes this approach applicable to any neural network architecture by leveraging pre-trained models. Figure 1 illustrates this method.

Suppose that we want to translate a sentence from English to German. First, we retrieve English sentences from the parallel aligned data and obtain similar English sentences which cover the input English sentence. For instance, from the source sentence: '*I want to go to school.*', we can retrieve the two following sentences that cover it: '***I want to go to*** *hospital.*' and '*This is a beautiful* ***school.***'. Their corresponding German translations are obtained from the parallel data: '*Ich will ins Krankenhaus.*' and '*Das ist eine schöne Schule.*' In this manner, in addition to just the source sentence, we acquire German-English sentence pairs with sentences in the source lan-

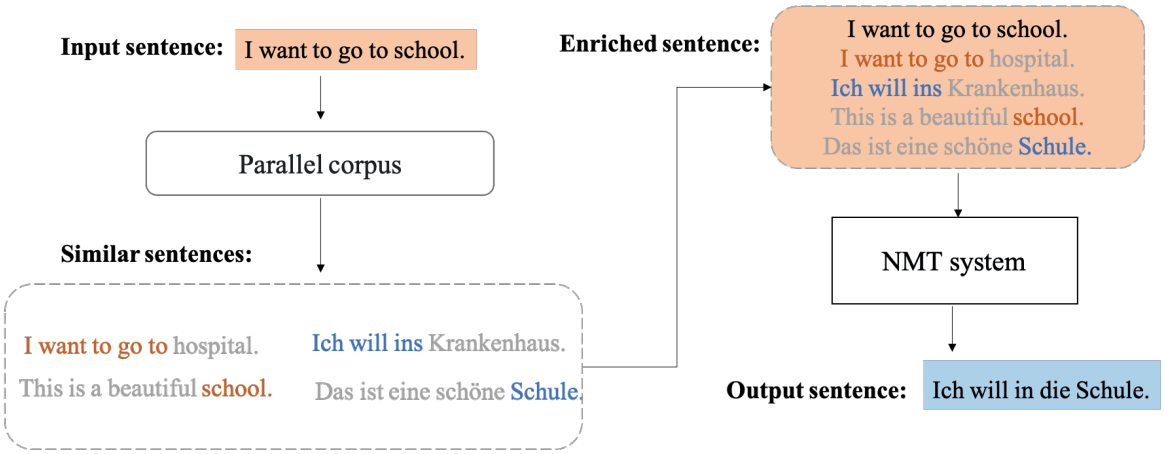


Figure 1: Overview of translation, based on retrieval: In the upper left corner, we show the input sentence in the target language. This sentence is the basis for the extraction of similar sentences, i.e. the sentences that cover the input in the lower left corner. The enriched sentence, shown in the upper right corner is the input to the NMT system and generates the improved translation, shown on the lower right.

guage similar to the input sentence, and their equivalents in the target language.

By extension, this provides us with sentences that cover the desired target sentence. The principle of translation memory claims that the German sentences should also be similar to the German translation of the English sentence. We use such translation pairs to enrich the input of an NMT system, i.e., we use them as a way of enriching the input sentence. We expect a boost in translation accuracy because the similar sentences with their translations should offer information that is useful during translation. The advantage of this method goes beyond the improvement in quality, it adds interpretability ((a) in Section 1.), thanks to traceability, and the potential to make the system less computationally intensive ((b) in Section 1.), alleviating the issues of NMT mentioned above.

4. Enrichment Schemes

To generate enriched sentences with retrieved sentences, there are different possibilities. We define several parameters for the enrichment of the input sentence. Basically, three aspects characterize an enrichment scheme: common parts, language side, and order in the case both language sides are used.

Common parts The following pieces of information extracted from the retrieved sentences can be used to enrich the source sentence.

- the common parts, i.e., only the parts which are common to the retrieved and input sentence (‘*I want to go*’ and ‘*school.*’ in the above example);
- the whole sentences retrieved, with markers to identify the common parts, in the source or the target language (e.g., ‘<cov>*I want to go to*</cov> *hospital.*’);
- the whole sentences retrieved, without any marker (e.g., ‘*I want to go to hospital.*’).

Language side One can choose to enrich the input sentence with:

- the sentences retrieved in the source language only;
- the corresponding translations in the target language only;
- both the retrieved sentences in the source language and their corresponding translations in the target language.

Order of similar sentences In the last case above, i.e., when both language sides are chosen, we can imagine several ways of combining the information in the source and target sides. We propose the following four ways:

- all source sentences followed by all target sentences;
- all target sentences followed by all source sentences;
- each source sentence followed by its corresponding target sentence for all pairs of sentences;
- each target sentence followed by the source sentence it corresponds to, for all pairs of sentences.

There are consequently $3 \times (2 + 4) = 18$ possible enrichment schemes for one given input sentence.

Table 1 gives some possible enriched sentences for the input sentence ‘*I want to go to school.*’ and the retrieved similar sentence pairs: ‘***I want to go to*** *hospital.*’, ‘***Ich will ins*** *Krankenhaus.*’ and ‘*This is a beautiful* ***school.***’, ‘*Das ist eine schöne* ***Schule.***’. Table 2 shows examples of translation results obtained using different enrichment schemes.

5. Experimental Setup

5.1. Datasets

We use the parallel corpus Multi30k (Elliott et al., 2016) as our parallel corpus. It contains multilingual image descriptions for the task of multilingual multimodal retrieval. We use the German, English and French parts in our experiments, i.e., 30,000 sentences in each language, as the name of the resource says.

All sentences are tokenized using SentencePiece (Kudo and Richardson, 2018) with subword units, i.e., byte-pair-encoding (BPE). We randomly divide the data

Common parts	Language side	Order	Enriched sentence
common parts	both	each source sentence followed by each target sentence	*I want to go to school. I want to go to Ich will ins school. Schule.*
common parts	both	all source sentences followed by all target sentences	*I want to go to school. I want to go to school. Ich will ins Schule.*
common parts	target	-	*I want to go to school. Ich will ins Schule.*
sentence without markers	target	-	*I want to go to school. Ich will ins Krankenhaus. Das ist eine schöne Schule.*
sentence with markers	target	-	*I want to go to school.* <cov> *Ich will ins* </cov> *Krankenhaus. Das ist eine schöne* <cov> *Schule.* </cov>

Table 1: Examples for several possible enrichment schemes

Reference	**Scheme**	**Translation**	**BLEU**
a man **in** ~~a cluttered~~ **office** ~~is using~~ the ~~tele~~phone .	source + sent.	a man talking on the phone **in** an **office** setting .	9.42
a man ~~in a cluttered~~ **office** is ~~using~~ the ~~tele~~phone .	target + sent.	a man is talking on the phone in an **office** setting .	8.91
a man in a cluttered office ~~is using~~ **the** ~~tele~~**phone** .	target + ngram	a man is talking on **the phone** in a cluttered office .	27.09
a man **in a** ~~cluttered~~ **office** is ~~using~~ ~~the tele~~**phone** .	target + sent._mark	a man is talking on a cellphone **in a** messy **office** .	10.60
a man in a cluttered office ~~is using~~ **the** ~~tele~~**phone** .	both + sent._mark + target_source	a man is talking on **the phone** in a cluttered office .	27.09

Table 2: Examples for translation results by different enrichment schemes. Stroke out text in the reference does not appear in the translation; boxed text is inserted text by comparison to the reference; bold text denotes a shift. The source sentence for this example is: "ein mann telefoniert in einem unaufgeräumten büro")

set into three parts: training set (80%), validation set (10%), and test set (10%).

5.2. Evaluation

We assess the results of the experiments by computing BLEU (Papineni et al., 2002) scores on the test set. In addition, we compute CHRF (Popović, 2015) and Translation Error Rate (TER) (Snover et al., 2006) scores. CHRF computes character n-gram F-scores between the candidate translation and the reference. TER measures the number of edits required to change a system output into one of the references. We report all above scores in the range of 0 to 100. For BLEU and CHRF, the higher the better. On the contrary, for TER, the lower, the better.

6. Results and Discussion

In order to present and discuss the various results in terms of language combination and translation direction, we briefly present the differences in results between the models and the impact of the size of the initial corpus used for the training in the following subsections.

6.1. Different Models

In our proposed method, one of the advantages is that we can use any NMT model easily without altering the architecture, as we only pre-process the input before using the NMT model. In our experiments, we compare LSTM (Bahdanau et al., 2015), Transformer (Vaswani et al., 2017) and mBART (Liu et al., 2020) models. mBART is trained on large-scale monolingual corpora in many languages. In our proposal, the enriched sentence contains words in both source and target languages, so we think that a multilingual model should be suitable for our method. The results are shown in Table 3.

LSTM or Transformer The Transformer model performs much better than the LSTM model, which proves again the efficiency of the self-attention mechanism.

With or without enrichment The last two rows in Table 3 are for our method, where the retrieved information is given to the model in addition to the source sentence. For both LSTM and Transformer models, our method outperforms the baseline model without enrichment. The addition of the retrieved information is thus proven to be effective and improves translation accuracy.

System	Trained or fine-tuned on Multi30k	Enrichment	BLEU	CHRF	TER
mBART	No	No	5.2 ± 0.7	25.0 ± 0.9	89.7 ± 2.0
LSTM	Yes	No	37.1 ± 1.5	55.4 ± 1.1	46.8 ± 1.4
Transformer	Yes	No	38.6 ± 1.5	57.8 ± 1.2	43.7 ± 1.4
LSTM (Ours)	Yes	Yes	38.1 ± 1.5	56.0 ± 1.1	47.1 ± 1.5
Transformer (Ours)	Yes	Yes	$\mathbf{39.7} \pm 1.5$	$\mathbf{58.2} \pm 1.2$	$\mathbf{43.5} \pm 1.3$

Table 3: Results for different models on test set (de → fr)

mBART The pre-trained mBART model without fine-tuning does not outperform a Transformer model trained from scratch. This indicates that fine-tuning is necessary when using pre-trained models.

6.2. Different Sizes of Corpora

To simulate an extremely low-resourced situation, we run experiments with different sizes for the training set. In the baseline model, the input data is just the sentence to translate, while in our method, the input is the sentence to translate with similar sentences, or common parts shared by similar sentences in the source or the target languages. Table 4 shows the results.

When the training size is $30,000 \times 80\% = 24,000$ sentence pairs, our method (*with* enrichment) outperforms the baseline model (input sentence only). But, when dividing the training data size by half, our method fails to outperform the baseline model. We presume that the reason is in the poor performance of the retrieval phase when working with insufficient data. The optimistic interpretation is that our method starts working from a relatively small amount of data: around 25,000 sentence pairs.

6.3. Multilingual and Multidirectional Comparison

Our initial experimental setup included all enrichment schemes on different translation tasks in all translation directions between German, English and French, resulting in 18 experimental runs in 6 different translation directions for each scheme. We conducted all possible experiments and examined the performance of all enrichment schemes. Due to space limitations we present a selection of 3 representative schemes in Table 5.

The most striking results are that in the English–French direction, all three enrichment schemes outperform the baseline model. In the German–French translation direction, two enrichment schemes perform better than the baseline.

For the same translation task, for example, from French to English, the scheme '*common part, both sides, each source sentence followed by each target sentence*' performs similarly compared with the baseline model. However, the scheme '*common part, target*' performs much worse than the baseline.

The first general observation is that the BLEU scores tend to be higher for most combinations that involve English. The directions from English to another language also tend to produce higher scores. We believe that this is due to a bias of the models used towards the English language. Translation from French to other languages seem to benefit the least from the enrichment, while English and German generally benefit from it. We believe that this is an interesting starting point of debate for the significance of these models towards individual language pairs, or the potential grouping of languages into categories that benefit more or less from these approaches.

7. Conclusion

In this paper, we proposed to integrate TM with NMT by enriching an input sentence with results from the retrieval of similar sentences. We enriched the input of an NMT system with information from such retrieved sentences using different schemes, and compared the translation results. When comparing different NMT architecture, the efficiency of the Transformer model was shown. Also, when using pre-trained models, our experiments showed that fine-tuning is necessary. The results of translation using our enrichment schemes show that, for some directions and some language pairs, our proposed method can perform better than a standard NMT system without enrichment.

References

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio, 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the Third International Conference on Learning Representations (ICLR 2015)*. Computational and Biological Learning Society.

Bulté, Bram and Arda Tezcan, 2019. Neural fuzzy repair: Integrating fuzzy matches into neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: ACL.

Cao, Qian and Deyi Xiong, 2018. Encoding gated translation memory into neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: ACL.

Elliott, Desmond, Stella Frank, Khalil Sima'an, and Lucia Specia, 2016. Multi30K: Multilingual English-German image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*. Berlin, Germany: ACL.

Federico, Marcello, Alessandro Cattelan, and Marco Trombetti, 2012. Measuring user productivity in machine translation enhanced computer assisted translation. In *Proceedings of the 10th Conference of the Association for Machine Translation in the Americas: Research Papers*.

Training Size	Baseline			Ours		
	BLEU	CHRF	TER	BLEU	CHRF	TER
24,000	38.6 ± 1.5	57.8 ± 1.2	43.7 ± 1.4	**39.7** ± 1.5	**58.2** ± 1.2	**43.5** ± 1.3
12,000	**33.8** ± 1.4	**52.7** ± 1.1	**50.2** ± 1.3	31.0 ± 1.3	50.0 ± 1.0	52.0 ± 1.2
6,000	**29.0** ± 1.3	**47.0** ± 1.1	**56.3** ± 1.3	26.7 ± 1.2	45.2 ± 1.0	58.2 ± 1.3
3,000	**20.7** ± 1.1	**39.0** ± 1.0	**66.2** ± 1.3	19.4 ± 1.0	37.5 ± 0.9	67.4 ± 1.1

Table 4: Results for different sizes of corpus (language pair: de → fr)

Translation task	Enrichment scheme	Baseline	Ours
de → en	common both source target	37.6 ± 1.5	36.2 ± 1.5
	common target		37.1 ± 1.5
	sentence mark target		36.8 ± 1.4
de → fr	common both source target	38.6 ± 1.5	**39.6** ± 1.5
	common target		**39.2** ± 1.5
	sentence mark target		37.9 ± 1.4
en → de	common both source target	34.5 ± 1.6	34.1 ± 1.6
	common target		33.4 ± 1.5
	sentence mark target		33.4 ± 1.5
en → fr	common both source target	52.8 ± 1.7	**53.2** ± 1.7
	common target		**53.7** ± 1.6
	sentence mark target		**53.3** ± 1.7
fr → de	common both source target	30.0 ± 1.5	30.1 ± 1.5
	common target		28.6 ± 1.4
	sentence mark target		28.8 ± 1.4
fr → en	common both source target	47.1 ± 1.6	42.6 ± 1.6
	common target		47.1 ± 1.6
	sentence mark target		46.6 ± 1.6

Table 5: Translation results for enrichment schemes in different translation tasks in BLEU score. The enrichment scheme *common both source target* uses only common parts on both sides, and each source part followed by each target. The *common target* scheme uses only common parts on the target side. The *sentence mark target* scheme uses sentence with markers on the target side.)

Gu, Jiatao, Yong Wang, Kyunghyun Cho, and Victor O.K. Li, 2018. Search engine guided neural machine translation. In *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*. AAAI press.

He, Qiuxiang, Guoping Huang, Qu Cui, Li Li, and Lemao Liu, 2021. Fast and accurate neural machine translation with translation memory. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*. ACL.

Kudo, Taku and John Richardson, 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: ACL.

Liu, Yinhan, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer, 2020. Multilingual Denoising Pre-training for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu, 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. ACL.

Popović, Maja, 2015. CHRF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: ACL.

Snover, Matthew, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul, 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*. Cambridge, Massachusetts, USA: Association for Machine Translation in the Americas.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Xu, Jitao, Josep Crego, and Jean Senellart, 2020. Boosting neural machine translation with similar translations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: ACL.

Solving Sentence Analogies by Using Embedding Spaces Combined with a Vector-to-Sequence Decoder or by Fine-Tuning Pre-trained Language Models

Liyan Wang, Zhicheng Pan, Haotong Wang, Xinbo Zhao, Yves Lepage

IPS, Waseda University
2-7 Hibikino, Kitakyushu, 808-0135, Japan
{wangliyan0905@toki., panzhicheng@toki., wanghaotong0925@toki., zhao.symbol@fuji., yves.lepage@}waseda.jp

Abstract

We address the task of solving sentence analogies, i.e., generating a sentence in analogy with three other given sentences. To this end, we use pre-trained word or sentence embedding models or fine-tune several pre-trained language models. Our training data consist of several thousands of semantico-formal analogies between short English sentences. Our experiments show that fine-tuning the large-scale language model GPT-2 performs better than other methods. However, the use a word embedding model to compute vector sentence representations combined with the use of a vector-to-sequence decoder, is shown to deliver reasonably competitive results in accuracy. For large pre-trained language models, the increase in number of parameters might not be worth the slight increase in performance observed, when compared with simpler models.

Keywords: Artificial Intelligence, Sentence Analogy, Pre-Trained Models

1. Introduction

Research on linguistic analogies, from strings of characters to sentences, has explored their potential in various tasks (e.g., machine translation) by casting the problem as reasoning analogies built up from given knowledge. Concretely, solving sentence analogies is the task of finding or generating the sentence D which satisfies the analogical equation $A : B :: C : D$ where A, B, and C are given and D is the unknown. The following sentence analogy

$$\begin{array}{l}\textit{I'm very glad}\\ \textit{to see you.}\end{array} : \begin{array}{l}\textit{I'm very pleased}\\ \textit{to meet you.}\end{array} :: \begin{array}{l}\textit{I'm happy}\\ \textit{to see you.}\end{array} : x$$

$$\Rightarrow \quad x = \begin{array}{l}\textit{I'm delighted}\\ \textit{to meet you.}\end{array}$$

is called a semantico-formal analogy because it combines the level of form:

$$\textit{see} : \textit{meet} :: \textit{see} : x \quad \Rightarrow \quad x = \textit{meet}$$

with the level of meaning:

$$\textit{glad} : \textit{pleased} :: \textit{happy} : x \quad \Rightarrow \quad x = \textit{delighted.}$$

Sentence-level analogies are far more difficult to solve than word analogies. Semantic analogies between words can be tackled by using word embedding models, because word embeddings capture specific types of relational similarity. The differences between word vectors express these relations Mikolov et al. (2013a); Pennington et al. (2014); Vylomova et al. (2016); Allen and Hospedales (2019). Analogies have indeed been used to assess the quality of word embeddings. For that purpose, test sets like the Google analogy test set Mikolov et al. (2013b) or BATS Gladkova et al. (2016) have been released.

2. Previous Work and Proposed Method

To solve sentence analogies, early research considered sentences as sequences of words or characters Nagao (1984); Lepage and Peralta (2004). This ignores the semantics of sentences. As a first step towards taking meaning into account, Lepage (2019) proposed an approach that basically considers sentences as sequences of words, but allows for analogies between words at the semantic level by using word embeddings. For that reason, such analogies are called semantico-formal analogies. A step further, Wang and Lepage (2020) proposed a method to generate the solution of a sentence analogical equation that uses a fully connected network to learn a mapping between the vector of the sentences given in an analogical equation and the vector of the sentence solution of the equation. They then use a pre-trained decoder to decode the solution vector back to a sentence.

Here, we propose to fine-tune pre-trained language models to directly perform the task of solving an analogical equation between sentences in an end-to-end manner.

Given an analogical equation $A : B :: C : D$, we concatenate the sentence quadruple as a text sequence according to the format:

$$\texttt{<A>}A\texttt{</A><B>}B\texttt{</B><C>}C\texttt{</C><D>}D\texttt{</D>}$$

where <X> and </X> are the boundary tokens identifying the term X of the analogy. We hypothesize that the ability of solving sentence analogies, i.e., generating solutions to analogical equations, can be learned through language modeling of sequences in the above format. The learning mechanism of language models should provide a way to glean information relevant for analogy from such format. For instance, the BERT-like architectures can predict masked words at any possible position in a sequence by

	analogies	unique sentences	words/ sent.	chars/ sent.
train	4,486	4,548	6.46	24.94
valid	560	1,216	6.25	24.06
test	561	1,168	6.26	24.09
all	5,607	5,124	6.46	24.95

Table 1: Statistics on the dataset of analogies.

conditioning their representations on the surrounding context. We follow Devlin et al. (2019) in randomly masking 15% of the words in a sequence, including boundary tokens. As for traditional language models (e.g., GPT-2), the entire sequence of analogy is generated in an autoregressive manner, where each word is estimated conditional on the previous words.

At the test time, the fine-tuned models are expected to generate the fourth sentence D given three sentences which implicitly encapsulate certain analogical transformations, i.e., $A : B :: C : x \quad \Rightarrow \quad x = D$. To this end, the input query for each test analogy is formatted in different ways pertaining to the model architecture. In the case of using a GPT-2 model, the three known sentences with their boundary tokens and the beginning boundary token of D are concatenated as the input. The model predicts the fourth sentence from left to right until the token `</D>` is generated. For BERT-like models, the words in D are masked out in sequences. The definition of formal analogy between strings in Lepage (2001) implies the knowledge of the length of the solution: $|D| = |B| - |A| + |C|$, where $|X|$ denotes the length of X in words. Thus, for each test analogy, the unknown sequence is replaced by a sequence of mask tokens of length $|B| - |A| + |C|$.

3. Experiments

3.1. Datasets

We use the set of semantico-formal sentence analogies released in Lepage (2019)[1]. It comprises 5,607 semantico-formal analogies between sentences extracted from the English part of the Tatoeba corpus. We divide the entire set into 80%, 10%, 10% for training, validation, and testing. Some statistics are shown in Table 1.

3.2. Using Embedding Spaces Combined with a Vector-to-Sequence Decoder

We perform experiments with the sentence analogy solver proposed in Wang and Lepage (2020) and test new configurations. This method consists in encoding sentences into vectors, solving the analogy at the level of vectors and decoding a vector back into a sentence, i.e., the candidate solution of the analogy.

We try two methods to obtain embedding vectors for sentences. The first one uses fastText word embeddings Grave et al. (2018) (simple vector summation to get a sentence vector) and the other one uses the Sentence-BERT (SBERT) Reimers and Gurevych (2019) model.

The analogy solver is a fully connected network that learns the mapping relations between three known vectors and one expected vector. The network comprises four linear layers, each hidden layer has 512 neurons with the Leaky Rectified Linear Unit (LeakyReLU) as the activation function. The loss function is mean square error. The network is trained on the semantico-formal analogy set mentioned above. It should be noted that the dimensionality of the vectors differ for fastText (300 dimensions) and SBERT (768 dimensions). For this reason, the vector analogy solver for fastText has 0.8 M parameters and it has 1.3 M parameters for SBERT.

To transform sentence embedding vectors back into sequences of words, we use a vector-to-sequence model similar to the decoder part of the RNN Encoder-Decoder. The parameters are as follows: embedding dropout of 0.4, number of layers of 1, batch size of 128, learning rate of 0.001. The optimizer is Adam. This decoder is trained on sentences from the English part of the Tatoeba corpus.

3.3. Fine-Tuning Pre-trained Language Models

We perform experiments in fine-tuning pre-trained language models for the task of predicting the fourth term of an analogical equation, based on the first three terms. This is an end-to-end configuration that takes a text consisting of three sentences marked with tags (for A, B and C) as input, and outputs the candidate solution of the analogy. We experiment with BERT Devlin et al. (2019), RoBERTa Liu et al. (2019), GPT-2 Radford et al. (2019) and ELECTRA Clark et al. (2020). Following the methodology proposed in the paper that introduced it, we initialize the ELECTRA model with the same number of parameters as BERT (110 M). We were unable to use the large or extra-large models of GPT-2, and used the medium model (345 M), which induces a maximal use of our GPU power. For tokenization, we use pretrained tokenizers of language models and extend them with eight boundary tokens which will not be split. We fine-tune these language models with a batch size of 8 for up to 100 epochs. In order to tailor ELECTRA, which is pre-trained as a discriminator for token replacement, to the analogy task, we fine-tune the model for masked language modeling (as BERT). To avoid over-fitting, we apply early stopping with a patience of 3 epochs, i.e., the training procedure automatically stops if there is no improvement on validation loss for 3 epochs. The learning rate is 5×10^{-5}. The optimizer is the default optimizer for these language models, AdamW.

3.4. Evaluation

We assess the results of the experiments by comparing the output candidate sentence to the reference sentence for each sentence analogy in the test set. We use three different metrics: accuracy, BERTScore and edit distance. Accuracy gives a global view, while BERTScore and edit distance provide a more refined view, with the difference that the

[1] http://lepage-lab.ips.waseda.ac.jp/en/projects/kakenhi-kiban-c-18k11447/, see tab Experimental results.

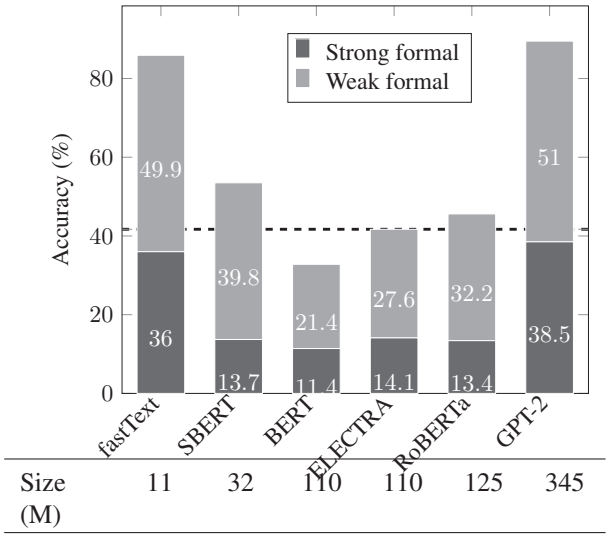


Figure 1: Accuracy results in two cases: weak and strong formal analogy. The dashed line indicates the percentage of strong formal analogies in the test set (41.7%). fastText and SBERT mean vecsolver+vec2seq for these two embedding models.

edit distance measures differences at the formal level, while BERTScore is a measure of semantic similarity.

The accuracy is the proportion of the number of predicted sentences that exactly match the actual sentences in the test set over the total number of sentences in the test set. The higher the better.

The Levenshtein distance is the minimal number of operations necessary to change one string into another one, with insertion, deletion and substitution as basic operations Levenshtein (1966). The lower, the better.

The BERTScore Zhang et al. (2020) basically aligns a candidate sentence and a reference sentence by cosine similarity between word embeddings. Its values range between 0 and 1 and the higher, the better.

4. Results and Discussion

4.1. Results

Table 2 gives the results that we obtained with all the different methods and models presented above. The fine-tuned GPT-2 model achieves 89.5% accuracy, i.e., in 9 cases out of 10, the model outputs the exact solution of the analogy, without even the smallest difference. The score of 0.68 for edit distance in characters indicates that, on average, the difference between the candidate sentence and the reference sentence is less than one character. As 90% of the candidate sentences are exact, this means that the remaining 10% of the sentences differ by an average of $10 \times 0.68 \simeq 7$ characters, i.e., one word and a half ($10 \times 0.15 = 1.5$ word). As expected, ELECTRA outperforms the BERT model of the same size by a 9% improvement in accuracy and achieves comparable performance to RoBERTa with less computation.

In Figure 1, we further examine the performance of inferring two categories of test analogies, which differ in the strength of the encapsulated relationship. We call *strong formal analogies,* analogies that follow a strict formal analogical relationship, with all the words in the reference answer appearing in the known sentences. We call *weak formal analogies* the other cases, which capture semantic similarity by using unseen words. The two best models (GPT-2, a large transformer-based language model, and vecsolver+vec2seq (fastText), a simple network combined with a small RNN-based decoder model) are proficient at learning the regularity in formal analogies, with only two to three errors ($(41.7\% - 36.0\%) \times 561 \simeq 3$) in strong formal analogies. Despite having access to information cues on the length of the answer, BERT-like decoders struggle to solve strong formal analogies.

Table 3 lists some examples where all learning models have trouble generating exactly the correct answers. In general, the GPT-2 model predicts answer sequences that are syntactically sound and have certain semantic overlap with the references. The vecsolver+vec2seq models can achieve performance competitive with the large language model, with a few words deviation from the references, but sometimes with strange word distributions. BERT-like language models struggle with solving sentence analogies, even reconstructing identical words within a quadruple. They exhibit the problematic phenomenon of generating repetitive words.

Next, we explore the performance of fine-tuned language models in solving analogies with the help of knowing a various number of preceding words in D. Table 4 shows the variations in accuracy of language models, when knowing the first 1 to 3 words. They perform better when they have access to the correct first words. In particular, the accuracy score of the BERT model starting with two cue words is significantly higher than that of the model generated from scratch, almost twice as high. Also, the gap between BERT-like models and the GPT-2 model is relatively reduced as the number of cue words increased.

4.2. Discussion

A sentence analogy contains four sentences, and we have nearly 6,000 sentence analogies. Hence we should theoretically have 24,000 sentences. However, we found that the dataset contains just over four thousand unique sentences. We hypothesize that an insufficient number of non-repeating sentences can affect the effectiveness of the fine-tuning. We expect a larger sentence analogy data set to induce higher performance. However, preliminary experiments in performing data augmentation by using the eight equivalent forms of an analogy did not yet lead to decisive conclusions. This remains to be further studied.

The method using an analogy solver on fastText embeddings combined with a decoder comes second in accuracy. It achieves a score of 85.9, only 3.6 percents behind the GPT-2 model. The two models achieve a very similar BERTScore at 0.996 and 0.995, supposedly with no statistically significant difference. The interpretation of the BERTScore tells that the candidate sentences supposedly have an almost same meaning as the reference sentences.

Method	Model	Size (M)	Accuracy (%)	Edit distance (chars)	Edit distance (words)	BERTScore
vecsolver+vec2seq Wang and Lepage (2020)	fastText	11	85.9	0.93	0.22	0.995
	SBERT	32	53.5	3.58	0.97	0.978
Fine-tuning (this paper)	BERT	110	32.8	5.47	1.50	0.949
	ELECTRA	110	41.7	4.56	1.29	0.960
	RoBERTa	125	45.6	4.25	1.24	0.960
	GPT-2	345	**89.5**	**0.68**	**0.15**	**0.996**

Table 2: Experiment results in sentence analogy resolution. The sizes of the models are in millions of parameters.

she was advised by him to give up smoking . : ***she was advised by him to give up drinking .*** :: ***she advised him to give up smoking .*** : *x*	
x= ***she advised him to give up drinking .***	
vecsolver+vec2seq (fastText)	****she advised him to give** in **drinking .***
vecsolver+vec2seq (SBERT)	***she advised him to** stop **.***
BERT	**he **advised up** ... ing **.***
ELECTRA	****she advised** to **to** to **up drinking .***
RoBERTa	****she advised him** him **to** to **drinking** drinking*
GPT-2	***she advised him to give up** smoking **.***
how many plates do you want ? : ***how many pens do you have ?*** :: ***how many apples do you want ?*** : *x*	
x= ***how many pencils do you have ?***	
vecsolver+vec2seq (fastText)	***how many** kids **do you have ?***
vecsolver+vec2seq (SBERT)	****how many** are **you have ?***
BERT	****how** have **many have** pencil **pencils ?***
ELECTRA	****how many** have have **you have ?***
RoBERTa	**what **do you have** any moneys*
GPT-2	***how many** bags **do you have ?***

Table 3: Examples of incorrect answers. The bold-faced words in the answers denote words that match the references. Ungrammatical sentences are denoted by *.

Figure 2, gives a comparison in sizes. It shows that the vecsolver+vec2seq (fastText) model (11 M) is more than 30 times smaller than the GPT-2 model (345 M). This obviously raises the question of the amount of data necessary to gain one percent of accuracy. If we recall the contemporary concern about green computing, our task exhibits a case where it is certainly worth examining the trade-off between the size of the models and their performance.

Now, due to the limitations of the computing power at our disposal, the medium GPT-2 model with 345 M parameters was used in our experiments. The large model with 762 M parameters and the extra-large model with 1,542 M parameters are supposed to capture and acquire even more language features for higher performance on downstream tasks. To repeat the above, the question is open concerning the cost at which any percent of accuracy would be gained by using these models.

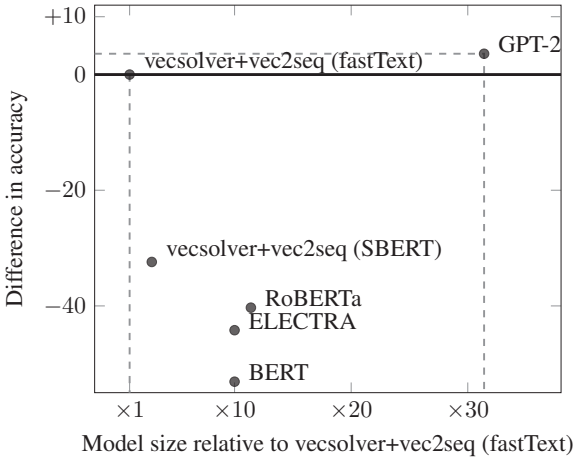


Figure 2: Accuracy versus model size. The reference is the vecsolver+vec2seq (fastText) model (size 11 M). Fine-tuning a GPT-2 model, which is more than 30 times larger than the reference, only improves accuracy by 3.6 points.

5. Conclusion

In this paper, we proposed to solve sentence analogy by using embedding spaces combined with a vector-to-sequence decoder, or by fine-tuning pre-trained language models. Although the large-scale GPT-2 pre-trained language model performs best in accuracy and formal similarity as measured by edit distances, it is worth questioning whether such a large model is really necessary, as the second model, which uses a fastText vector analogy solver combined with a vec2seq decoder, and is more than 30 times smaller, comes close behind in accuracy, and is indistinguishable

Model	# known words of D						
	0		1		2		3
BERT	32.8	39.8	(+7.0)	62.9	(+30.1)	70.9	(+38.1)
ELECTRA	41.7	51.0	(+9.3)	67.0	(+16.0)	72.6	(+21.6)
RoBERTa	45.6	59.7	(+14.1)	74.7	(+29.1)	79.1	(+33.5)
GPT-2	89.5	90.0	(+0.5)	87.5	(−2.0)	91.4	(+1.9)

Table 4: Accuracy results (%) of language models in solving analogies with none, and 1 to 3 word answer hints. The values enclosed in parentheses represent the gain (+) or drop (−) in accuracy scores from Table. 2.

in semantic similarity as measured by BERTScore.

Acknowledgment. The work reported in this paper has been supported in part by a grant for research (Kakenhi C) from the Japanese Society for the Promotion of Science (JSPS), n° 21K12038 "Theoretically founded algorithms for the automatic production of analogy tests in NLP".

References

Allen, Carl and Timothy Hospedales, 2019. Analogies explained: Towards understanding word embeddings. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*.

Clark, Kevin, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning, 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1 (Long and Short Papers). Minneapolis, Minnesota.

Gladkova, Anna, Aleksandr Drozd, and Satoshi Matsuoka, 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL-HLT SRW*. San Diego, California.

Grave, Edouard, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov, 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*. Miyazaki, Japan.

Lepage, Yves, 2001. Analogy and formal languages. *Electronic Notes in Theoretical Computer Science*, 53:180 – 191. Proceedings of the joint meeting of the 6th Conference on Formal Grammar and the 7th Conference on Mathematics of Language.

Lepage, Yves, 2019. Semantico-formal resolution of analogies between sentences. In *Proceedings of the 9th Language & Technology Conference – Human Language Technologies as a Challenge for Computer Science and Linguistics*.

Lepage, Yves and Guilhem Peralta, 2004. Using paradigm tables to generate new utterances similar to those existing in linguistic resources. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*. Lisbon, Portugal.

Levenshtein, V.I., 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics-doklady*, 10(8):707–710.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean, 2013a. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, volume 2. NY, USA.

Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig, 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia.

Nagao, Makoto, 1984. A framework of a mechanical translation between japanese and english by analogy principle. *Artificial and human intelligence*:351–354.

Pennington, Jeffrey, Richard Socher, and Christopher D Manning, 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing*.

Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever, 2019. Language models are unsupervised multitask learners. Technical report, OpenIA.

Reimers, Nils and Iryna Gurevych, 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. Hong Kong, China.

Vylomova, Ekaterina, Laura Rimell, Trevor Cohn, and Timothy Baldwin, 2016. Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Berlin, Germany.

Wang, Liyan and Yves Lepage, 2020. Vector-to-sequence models for sentence analogies. In *Proceedings of the 2020 International Conference on Advanced Computer Science and Information Systems*.

Zhang, Tianyi, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi, 2020. BERTScore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Exploring synonymy relation between multi-word terms in distributional semantic models

Yizhe Wang[1], Béatrice Daille[2], Nabil Hathout[3]

[1] University of Science and Technology of China
[2]LN2S, CNRS & University of Nantes
[3]CLLE, CNRS & University of Toulouse2

Abstract

Terminology describes the knowledge structure of a domain through the relationships between its terms. However, relations between multi-word terms (MWTs) are often underrepresented in terminology resources. Moreover, most of the work on this issue concerns the relations between simple terms (STs). In this paper, we explore the ability of distributional semantic models (DSMs) to capture synonymy between MWTs by lexical substitution based and analogy based methods. We evaluated our methods on the English and French MWTs of the environmental domain. Our experiments show that the results obtained using analogy in static word embeddings are globally better than the ones obtained using lexical substitution in pre-trained contextual models.

Keywords:

1. Introduction

The demand for structured terminological resources is strong, especially for extracting and acquiring information from texts. Terminology resources gather the terms of a domain and describe the relations that exist between them, such as synonymy. While multi-word terms (MWTs) are widely represented in terminology, the relationships between them are often missing. Synonymy is an important relation in terminology. It has been the subject of several studies in which a variety of methods have been proposed, including methods based on syntactic patterns, multilingual methods and distributional methods. However, most of these studies concern single terms (STs) and very few focus on the acquisition of synonymy between MWTs. Works on synonymy between MWTs in the literature often explore the internal structure of MWTs using different types of linguistic information, especially semantic information. In this paper, we explore the ability of distributional semantic models (DSMs) to capture synonymy between MWTs in the environmental domain in English and French. Our study focuses on nominal MWTs composed of two lexical words (i.e., biterms). Two methods are proposed. The first is based on lexical substitution using a masked language model (MLM). The second captures synonymy through analogy between STs and MWTs representations in a FastText (Bojanowski et al., 2017) model. Both methods are tested on two datasets of English and French MWT synonyms of the environment domain extracted from the IATE translation dictionary.

Section 2. presents related works on the acquisition of relations between words and terms. Our methods are introduced in Section 3.. Section 4. presents the resources we used to create our data set and to perform the experiments. Section 5. describes the implementation of the methods. We present and discuss the results obtained in Sections 6.. Section 7. concludes the article and presents future avenues of research.

2. Relates work

We propose two methods for identifying synonymy between MWTs: lexical substitution and analogy. Lexical substitution is a task that aims at predicting candidate words that can replace a target word in a given context. In recent studies, lexical substitution has often been used to acquire semantic relations and is usually performed using masked language models (MLMs). For example, Schick and Schütze (2020) and Arefyev et al. (2020) use Transformer models to test the ability of these models to capture lexical relations between words from the general domain without any task-specific optimization. Their results show that BERT is able to capture relational semantic properties and that most of the returned substitutes are synonyms and co-hyponyms when the masked word is a noun. This observation confirms that of Ferret (2021). The way we use lexical substitution is close to the ones presented in these works. However, our method differs in several respects. We seek to identify lexical semantic relations between MWTs in French (in addition to English) in the environmental domain, whereas the work presented focuses on relations between single words in English in the general domain. Moreover, we use contexts extracted from corpora and not patterns that express these relations as Schick and Schütze (2020) do. In addition, we use a conditioning strategy that allows us to provide the model with additional information about the masked word, but in a different way than Arefyev et al. (2020) (cf. Section 3.).

Analogy is a method we use for detecting whether two pairs of words are in the same relation. The study of Mikolov et al. (2013) shows that analogy is able to capture linguistic relations in vector space models and that the identification of these relations can be estimated by the offset between their distributional representations ($V_a - V_b \approx V_c - V_d$ for an analogical quadruplet $a : b :: c : d$). In line with Mikolov et al. (2013), many studies have focused on the ability of analogy to capture various lexical, encyclopedic,

or specialized domain relations (Gladkova et al., 2016; Chen et al., 2018; Wohlgenannt et al., 2019). While most studies focus on word analogy between single terms, Chaudhri et al. (2022) focus on solving analogous equations between single and multi-word terms in the biology domain in English in order to capture domain-specific relationships like *a type of*. The study of Paullada et al. (2020) also focuses on analogy between STs and MWTs in the biomedical domain. Their objective is to acquire domain-specific relations, such as gene-disease relations. The authors created a DSM from a corpus of sentences extracted from the biomedical literature and annotated with syntactic dependencies. They show that embeddings that incorporate syntactic information do improve the resolution of biomedical analogy equations.

Our study differs from the ones we have just presented in several respects. As we already pointed out, we are working on the identification of terminological relations between MWTs in English and French in the environmental domain. We are interested in classical lexical semantic relations between MWTs and not in domain-specific ones. Like Paullada et al. (2020), we use vector offset instead of seq2seq and seq2vec models as do Chaudhri et al. (2022) to solve analogy equations. However, the model we use is different from that of Paullada et al. (2020) because we use a FastText model where the MWTs and their components are represented in the same vector space.

3. Methods

In this section, we describe in detail our methods for acquiring synonymy between MWTs in the environmental domain.

3.1. Lexical substitution

Our first method is lexical substitution using MLMs. MLMs are models trained to predict which tokens are likely to replace a special token `<mask>`. They can thus easily be used to acquire synonymy between MWTs. Let MWT_1 and MWT_2 be two MWTs with the same syntactic structure, such that MWT_1 contains the lexical words W_1 and W_3 and MWT_2 contains W_2 and W_3. We assume that MWT_1 and MWT_2 have a compositional meaning. Therefore, W_3 contributes identically to the meaning of MWT_1 and MWT_2 and as a consequence, the relationship between W_1 and W_2 is preserved between MWT_1 and MWT_2. Let S_1 be a context of MWT_1 and S_2 a context of MWT_2. Let k_1 be the rank of W_1 among the MLM predictions for the query obtained by masking W_2 in S_2. Let k_2 be the rank of W_2 among the MLM predictions for the query obtained by masking W_1 in S_1. Let N be the number of neighbors that we consider to be close enough. If $k_1 < N$ or if $k_2 < N$, we predict (*i*) that W_1 and W_2 are synonymous and (*ii*) that MWT_1 and MWT_2 are probably synonymous.

The method can be illustrated with the following example. The context S_1 is used to create the query Q_1 whose $N = 10$ first answers contain the other word (*protection*). This allows the method to predict that the relation (synonymy) between the two TS also exists between the two MWTs.

MWT pair: forest preservation ; forest protection
M_1: preservation
M_2: protection
Target relation: synonymy
Masked context S_1: financial support for the $< mask >$ of forests will be a major topic at the conference
N = 10
Observation: *protection* appears at rank 2 in the list of predictions for query Q_1
Conclusion: *forest preservation* and *forest protection* are synonyms

In our study, we compare "basic" MLM queries and conditioned MLM queries. Zhou et al. (2019) observe that MLMs produce candidates that can be semantically very different from the masked word while being perfectly suited to the context. To solve this problem, we adopt the conditioning method proposed by Qiang et al. (2019). The method uses queries composed of the concatenation of the original context (where the target word is not masked) and the masked context (where the target word is masked).

3.2. Analogy

The second method is based on analogy in static word embeddings. The detection of relations between MWTs by analogy follows from the observation that if $W_1 : MW_2 :: MWT_1 : MWT_2$ is a proportional analogy then the relation between W_1 and W_2 is the same as the one between MWT_1 and MWT_2. The analogy function *3CosADD* (Mikolov et al., 2013) can be used to solve analogy equations in DSMs. For example, if we choose MWT_2 as the unknown, then we seek to estimate the distance between the representation of MWT_2 and the expected vector $V_{\text{expected}} = V_{MWT_1} - V_{W_1} + V_{W_2}$. Each quadruplet produces two analogy equations taking respectively MWT_1 or MWT_2 as the unknown. The final result is the average of the rank of the unknown MWT in the predictions for both equations. The following example illustrates the method:

Quadruplet: *dry : wet : dry climate : humid climate*
Known relationship: antonymy between *dry* and *humid*
Analogy equations:
equation_1: *dry : wet :: dry climate : ?*;
equation_2: *dry : wet :: ? : humid climate*
Number of neighbors considered close: 5
Observation: the expected MWT for both equations are found in the first 5 predictions
Conclusion: *dry climate* and *humid climate* are antonyms

4. Data

Corpus. We used the English and French monolingual PANACEA Environment corpora (ELRA-W00653 and ELRA-W0065) which were built in the framework of the PANACEA project[1]. These corpora are more heterogeneous than typical specialized ones which normally contain specialized texts only because the environmental domain is heterogeneous in nature Bernier-Colborne (2016).

[1] `http://www.panacea-lr.eu/en/info-for-researchers/data-sets/monolingual-corpora`

DicoEnviro. DiCoEnviro[2] is a multilingual dictionary of environmental terms developed by the Observatoire de linguistique Sens-Texte (OLST)[3]. It describes the meaning and linguistic properties (especially the lexical-semantic ones) of terms belonging to various sub-domains of the environment domain.

IATE. IATE[4] (Interactive Terminology for Europe) is an EU translation terminology resource that contains synonymy relations between terms. It is a rich resource from which datasets can easily be extracted.

Data sets. Our datasets are created from IATE. We extracted 786 pairs of synonymous English biterms (we will call this set Data_en in the following) like *climate conference* : *climate summit* and 928 pairs in French (we will call this second set Data_fr in what follows) like *analyse du risque*: *risk study*. We manually validated the synonymy relation between the biterms in the extracted pairs. In order to further select our data, we performed an analysis of the pairs extracted from IATE. We observed that more than 85% of the pairs of synonymous biterms share one lexical word. We used the following subsets of Data_en and Data_fr in our experiments: Data_MLM_en (510 pairs) and Data_MLM_fr (563 pairs) are made up of MWT pairs that have the same pattern and share one lexical element while Data_FastText_en (431 pairs) and Data_FastText_fr (599 pairs) contain MWT pairs of frequency higher than 5 in PANACEA.

5. Experiments

We use the MRR score and the precision at Top1, Top5, and Top10 to evaluate the quality of methods for all our experiments.

$$\text{MRR} = \frac{1}{|W|}\sum_{i=1}^{|W|}\frac{1}{Rank_i}$$

where $|W|$ is the number of queries and $Rank_i$ is the rank of the first correct answer for the i-th query. The closer the MRR score is to 1, the better the model performs.

$$\text{Precision} = \frac{n}{|W|}$$

where n is the number of queries that produce a correct result among the answers at Top1, Top5, or Top10 and where $|W|$ is the total number of queries.

5.1. Lexical substitution

Data sets. The lexical substitution experiments were performed using Data_MLM_en and Data_MLM_fr. We removed the pairs of biterms in which one of the lexical items is not included in the vocabulary of the model because out-of-vocabulary words are divided into several wordpieces, which makes the identification of the possible substitutes for the target word difficult. Moreover, we need contexts to build the queries submitted to the MLM model. For each MWT in the dataset, we extracted 100 contexts from the corresponding PANACEA corpus. Only contexts that meet some of the quality criteria of good contexts proposed by Kilgarriff et al. (2008) were selected. Note that some MWTs appear less than 100 times in PANACEA. Overall, the data used for the lexical substitution experiment in English consists of 317 term pairs and 24,265 contexts (referred to as Test_MLM_en). For French, the data used consists of 385 term pairs and 24 404 contexts (referred to as Test_MLM_fr).

Models. We conducted the experiments of synonymy acquisition by lexical substitution using the BERT-large-uncased model for English and CamemBERT-large for French.

Vocabulary. In these experiments, we only count as candidates the substitutes that are simple terms. As the MWTs were extracted from IATE, we considered using as a reference the simple terms that appear in this same bank. However, the number of single terms in IATE proved to be too small. For that reason, we used a larger vocabulary consisting of the 818 English and 784 French lexical units that appear in the MWTs of Data_en and Data_fr and that are part of the vocabulary of the models.

5.2. Analogy

Data sets. The data used for the analogy experiments are quadruplets $W_1 : W_2 :: MWT_1 : MWT_2$ such that MWT_1 contains W_1; MWT_2 contains W_2; MWT_1 and MWT_2 share a word W_3; W_1 and W_2 are synonyms. For each of the two languages, two datasets were created using the biterms pairs in Data_Fasttext_en and Data_Fasttext_fr and the synonymy relations in DiCoEnviro and IATE. These datasets will be referred to as Quad_IATE_en (9 quadruplets) and Quad_Envi_en (33 quadruplets) for English and Quad_IATE_fr (20 quadruplets) and Quad_Envi_fr for French (63 quadruplets).

Models. The representations of the MWT should not be computed by composition from the representations of their constituents because the analogy equation would then always be trivially true. Therefore, we use FastText models for the acquisition of synonyms by analogy because can include independent representations for MWTs and their constituents within the same vector space. To compute these representations, we first annotated the corpus so that MWTs and their constituents are indexed separately. For example, a MWT such as *cold air* produces the three tokens: `air`, `cold`, and `air_cold`. We have also forced the model not to split the words into character n-grams by setting the `maxn` parameter to 0.

Vocabulary. The task being the acquisition of synonymy between MWT, the rank of the candidate solutions of the analogical equation is computed with respect to a vocabulary composed of all the nominal biterms in IATE which appear at least 5 times in the PANACEA corpus (5 465

[2] `http://olst.ling.umontreal.ca/cgi-bin/DiCoEnviro/search_enviro.cgi`
[3] `http://olst.ling.umontreal.ca`
[4] `https://iate.europa.eu/`

biterms for English and 5 002 biterms for French).

6. Results and discussions

The results of the lexical substitution and analogy experiments are presented in Tables 1 and 2. Overall, the two methods perform similarly on the English and French datasets.

Method	MRR	P1	P5	P10
Data_MLM_en without conditionning	0.304	0.186	0.433	0.551
Data_MLM_en with conditionning	**0.443**	**0.315**	**0.589**	**0.689**
Data_MLM_fr without conditionning	0.302	0.189	0.416	0.532
Data_MLM_fr with conditionning	0.374	0.253	0.502	0.613

Table 1: MRR score and precision at Top1, Top5 and Top10 of the lexical substitution methods using MLM queries without and with conditioning

Method	MRR	P1	P5	P10
Quad_IATE_en	0.733	0.612	**0.889**	0.889
Quad_Envi_en	0.698	**0.727**	0.83	**0.909**
Quad_IATE_fr	**0.744**	0.650	0.875	0.900
Quad_Envi_fr	0.624	0.548	0.723	0.746

Table 2: MRR score and precision at Top1, Top5 and Top10 of the analogy method using FastText models

Table 1 shows that lexical substitution results are improved by query conditioning. The MRR scores increase from 0.304 to 0.443 for English biterms and from 0.302 to 0.374 for French biterms. Precision is also improved. Query conditioning improves synonymy acquisition in English more than in French. This could be due to the fact that we used different MLMs for the two languages. A second possible reason could be that MWT contexts in English are less informative than those in French, which could be roughly estimated by the length of the contexts: on average, English queries contain 30 words while French ones contain 35 words. Moreover, we also checked in both languages that short queries benefit more from the conditional strategy than long ones. A qualitative analysis of the first 10 predictions of 100 randomly selected queries with conditioning shows that most of the predictions are semantically similar to the masked word. Most of them are synonyms and variants, including derivational ones. These results are in line with the observations of Ferret (2021) and Arefyev et al. (2020). For most queries where the expected term does not appear in the Top10 predictions, some of its synonyms do. For example, when *habitation* 'house' is masked in a context of *habitation individuelle* 'individual house', the expected term *maison* 'house' only appears at rank 71, but its synonym *logement* 'housing' appears at rank 2.

Table 2 shows that analogy captures synonymy between MWTs effectively. The best MRR score of 0.744 is obtained for Quad_IATE. We can also see that the quadruplets constructed using relations between simple terms from IATE give the best results. These good numbers could be explained by the fact that the synonymy relations between MWTs and the simple terms that compose these quadruplets come from the same source. Remember that IATE is a translation dictionary while DiCoEnvio is a terminology database. We conducted a qualitative analysis similar to the one we did for lexical substitution. We examined the first 5 predictions of all queries whose unknown is MWT_2. We observed that MWT_2 is present in the first five candidates for more than 70% of the quadruplets. When MWT_2 does not appear among the first five candidates, we found in most cases one of its synonyms among the candidates. For example, for the quadruplet *effet:incidence::effet sur l'environnement:incidence sur l'environnement* 'effect:impact::environmental effect:environmental impact', the unknown term *incidence sur l'environnement* 'impact on the environment' is at rank 3 698, but the first prediction, *incidence environnementale* 'environmental impact', is a derivational variant of the target MWT. In addition, we also noticed that the frequency of MWTs in the corpus also has an impact on the results. For queries where the unknown term and its synonym do not appear in the first five predictions, the frequency of MWT_1 or/and MWT_2 is often less than 10.

The main difference between the lexical substitution and analogy methods is that MLM predictions are highly context dependent unlike the predictions based on FastText models. Moreover, FastText representations are built on a small specialized corpus, while BERT models are pre-trained on a large variety of corpora. The differences also arise from the fact that BERT is queried at the occurrence level, whereas FastText representations are based on a set of occurrences. Our analogy-based method is better suited for synonymy identification. This observation is consistent with that of Peters et al. (2018) who show that contextual language models underperform compared to static models on analogy-based semantic relation identification tasks.

It should also be noted that the analogy method gets more information than the lexical substitution method because it is provided with the two related simple terms, which is not the case for the latter. The better results obtained with analogy also suggest that semantic composition in MWTs is better captured by FastText models than it is by MLMs. These observations are consistent with the conclusion of Hupkes et al. (2020) that Transformer models have a low level of compositional generalization.

7. Conclusion

In this paper, we explored the capacity of DSMs to capture synonymy between biterms of the environment domain

in English and French. We performed experiments using MLMs for the lexical substitution method and using static FastText models for the analogy method. The results of the experiments show that overall, both methods perform well in both languages. However, the analogy methods outperform the lexical substitution methods. The analogy method obtains an MRR score of 0.744 on the French dataset extracted from IATE. Our results also suggest that semantic composition is better grasped by static dives; conversely, the level of compositional generalization of Transformer models seems to be lower. Overall, this study is one of the first attempts to identify synonymy between MWTs in a specialized domain, the environment, by exploring DSMs. This work also provides a roadmap for the application of DSMs to the terminology structuring task. The future step of this work is to increase the performance of the lexical substitution method by improving the quality of the contexts. We also plan to use generative models like GPT-3 (Brown et al., 2020) instead of MLMs to perform the lexical substitution task.

8. Acknowledgements

This work has been funded by the French National Research Agency (ANR) under the ADDICTE (ANR-17-CE23-0001) and the DELICES (ANR-19-CE38-0005) projects

References

Arefyev, Nikolay, Boris Sheludko, Alexander Podolskiy, and Alexander Panchenko, 2020. A comparative study of lexical substitution approaches based on neural language models. *arXiv preprint arXiv:2006.00031*.

Bernier-Colborne, Gabriel, 2016. *Aide à l'identification de relations lexicales au moyen de la sémantique distributionnelle et son application à un corpus bilingue du domaine de l'environnement*. Ph.D. thesis, Université de Montréal, Montréal, Canada.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov, 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei, 2020. Language Models are Few-Shot Learners. In H Larochelle, M Ranzato, R Hadsell, M F Balcan, and H Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc.

Chaudhri, Vinay K, Justin Xu, Han Lin Aung, and Sajana Weerawardhena, 2022. *A Corpus of Biology Analogy Questions as a Challenge for Explainable AI*. Cham: Springer International Publishing, pages 327–337.

Chen, Zhiwei, Zhe He, Xiuwen Liu, and Jiang Bian, 2018. Evaluating semantic relations in neural word embeddings with biomedical and general domain knowledge bases. *BMC medical informatics and decision making*, 18(2):53–68.

Ferret, Olivier, 2021. Exploration des relations sémantiques sous-jacentes aux plongements contextuels de mots (exploring semantic relations underlying contextual word embeddings). In *Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale*. Lille, France: ATALA.

Gladkova, Anna, Aleksandr Drozd, and Satoshi Matsuoka, 2016. Analogy-based detection of morphological and semantic relations with word embeddings: What works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*. San Diego, California.

Hupkes, Dieuwke, Verna Dankers, Mathijs Mul, and Elia Bruni, 2020. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795.

Kilgarriff, Adam, Miloš Husák, Katy McAdam, Michael Rundell, and Pavel Rychlý, 2008. Gdex: Automatically finding good dictionary examples in a corpus. In *Proceedings of the XIII EURALEX international congress*. Barcelona, Spain: Documenta Universitaria.

Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig, 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*. Atlanta, Georgia: Association for Computational Linguistics.

Paullada, Amandalynne, Bethany Percha, and Trevor Cohen, 2020. Improving biomedical analogical retrieval with embedding of structural dependencies. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*. Online: Association for Computational Linguistics.

Peters, Matthew E., Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih, 2018. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics.

Qiang, Jipeng, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu, 2019. A simple bert-based approach for lexical simplification. *ArXiv*, abs/1907.06226.

Schick, Timo and Hinrich Schütze, 2020. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *Proceedings of the*

AAAI Conference on Artificial Intelligence, volume 34. New York, USA: AAAI Press.

Wohlgenannt, Gerhard, Ekaterina Chernyak, Dmitry Ilvovsky, Ariadna Barinova, and Dmitry Mouromtsev, 2019. Relation extraction datasets in the digital humanities domain and their evaluation with word embeddings. *arXiv preprint arXiv:1903.01284*.

Zhou, Wangchunshu, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou, 2019. Bert-based lexical substitution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics.

Exploring the Synergies between Technology and Socio-Cultural Approaches in Computer-Assisted Language Learning for Less Commonly Taught languages

Liang Xu[1], Elaine Uí Dhonnchadha[2], Monica Ward[1]

[1]Dublin City University, Dublin, Ireland
liang.xu6@mail.dcu.ie, monica.ward@dcu.ie
[2]Trinity College Dublin, Dublin, Ireland
uidhonne@tcd.ie

Abstract

Language learning is a complex task and to be successful, it involves a range of cognitive processes. Intelligent Computer Assisted Language Learning systems can be useful for enhancing the effectiveness and efficiency of both teacher-led instruction and student learning. Digital technologies are routinely employed for commonly taught languages but are less frequently used in the Less Commonly Taught Language (LCTL) context. In this study, we investigate methods for encouraging the learning and teaching of LCTLs, particularly indigenous and endangered languages. Our research blends language learning pedagogy, socio-cultural theory, and Digital Game-Based Language learning (DGBLL) in the form of games and Virtual Reality. Language learning materials are enhanced with the aid of Natural Language Processing techniques. We present a DGBLL system designed to promote language learning and student engagement. The system has been tested successfully in primary school classrooms, with positive feedback from both students and teachers.

Keywords: Computer-Assisted Language Learning · Less Commonly Taught Languages · Digital-Game-Based Language Learning

1. Introduction

Computer-Assisted Language Learning (CALL) research is heavily focused on the most commonly taught languages, particularly English. This is not surprising as there are around 1.5 billion English language learners in the world (Council, 2016). This means that most of the CALL resources developed are for learners of English. Consequently, there are fewer resources for Less Commonly Taught Languages (LCTLs) (Ward, 2015). In fact, a language can be an official language of a country and yet be a LCTL internationally. This is the case for the Irish language.

The concept of learning languages as a by-product of playing games is known as Digital Game-Based Language Learning (DGBLL). Furthermore, research (Lan, 2020; Lin and Lan, 2015) has shown that Virtual Reality (VR) has the capability to improve language learning. CALL involves the use of technology in the language learning process. It encompasses DGBLL and VR language learning but is broader in reach. Natural Language Processing (NLP) technologies facilitate human language interactions for tasks such as natural language understanding, generation and inference. NLP resources have the potential to contribute to CALL (Ward, 2019), but this potential remains largely under-utilised. In this project *Cipher*, DGBLL is used as the bridge between CALL and NLP to develop a game-based and VR-enhanced language learning application. The game's theme, centred around fairy tales, stories, and myths, is selected to actively engage and motivate learners. By choosing familiar fairy tales at the lower levels of language proficiency, we can build on the learner's knowledge from their L1. At the more advanced levels, we use folk tales and mythology which are engaging and can be made culture-specific and reflect the theme of "reconnecting to the spirit of the language" (see section 2.2.1).

An educational and entertaining game such as ours is particularly important in the context of Irish society, where Irish is a minority and threatened indigenous language. In this context, many L2 learners lack extracurricular exposure to the language. This game provides the opportunity to improve certain language skills while engaging with the language in a fun and pedagogically beneficial way. Additionally, the project adheres to robust software engineering practices making it language-independent so that the game can be adapted to other LCLTs.

2. Background

2.1. CALL for Less Commonly Taught Languages and endangered languages

In recent years, there has been a surge in the popularity of language learning applications such as Duolingo, Memrise, and BabbleAR. Blake (2011) suggests that CALL is the way forward for the learning and teaching of the more commonly taught languages. However, fewer CALL resources are available for LCTLs. For these languages, it is not easy to assemble the human and financial resources to develop engaging, high-quality language learning resources (Ward, 2015). Furthermore, motivation is a big challenge for learners of LCTLs and increasing motivation is important in LCTL CALL research. LCTL CALL researchers are creative (e.g., Millour and Fort, 2020) and aim to leverage existing technology and resources and adapt them to their own LCTL (e.g., Purgina et al., 2017).

Endangered languages have further challenges when it comes to CALL including the lack of printed and online resources in the language, dialectal issues, lack of societal support, lack of quality language documentation, lack of an active speaker community or native speakers, as well as a lack of competent linguists and teachers (Ward, 2015). Many learners of LCTLs, particularly those of endangered languages, are deprived of the benefits of CALL such as

accessibility to learning materials that is both easier and more cost-effective. Furthermore, it is more challenging for learners to access authentic language resources. For example, simply wanting to hear their languages being spoken can be an issue as learners of the languages may not have native speakers in their vicinity and some of these languages may not even have many speakers remaining (Ward, 2015). As a result, learners face more difficulties in learning these languages, leading to a reduction in the number of language learners and a corresponding decrease in demand for CALL, creating a vicious cycle.

To address this issue, our research focuses on digital resources for the learning and teaching of LCTLs. From a computational resources' perspective, the emphasis is on low-resourced languages, while from a socio-cultural standpoint, the focus is on indigenous and endangered languages.

2.2. Indigenous language revitalisation in the modern world

2.2.1. Philosophy: reconnecting to the spirit of the language

More and more languages are disappearing all over the world (UNESCO, 2022). The reason for this can be attributed in part to the dwindling number of individuals who speak the language. Indigenous people, especially younger generations, are reluctant to learn the language and their parents are sometimes reluctant to pass it on to them. For example, Napier and Whiskeyjack (2021) hold the view that there is a disconnect between the Cree language (an indigenous language in Canada) and the Cree people. Their research explored the reasons for the disconnection and provided some solutions to improve the situation of the language. This is the theory of reconnecting to the spirit of the language. The steps of reconnection are history, harms and healing. According to their research, the spirit of languages relies on the history of the land, languages and laws. As the findings are useful for the language revitalisation of the Cree language, it is possible that the theory may apply to other indigenous languages. In our project, we leverage the theory by incorporating the language, lore and mythology within the Irish context in an immersive environment.

2.2.2. Indigenous language learning in a digital age

Previous studies on indigenous language revitalisation show that digital technology plays an important role in the survival of these languages due to the advantage of digital technology being able to record, preserve, analyse, manipulate, and transmit languages in numerous ways (Galla, 2018). Digital technology has been applied to the field of indigenous language revitalisation in various forms including social media, apps, and VR (Galla, 2018) (Outakoski et al., 2018). According to Galla's (2016) survey of at least 47 indigenous languages in 2009, digital technology has a positive impact on indigenous language education. Furthermore, research has demonstrated that indigenous youth actively use digital technology, which can help their reconnection to their heritage and languages. With active engagement and interaction with digital technology, indigenous youth are gaining increasing language exposure consciously and unconsciously. Furthermore, emerging technologies like VR allow young learners to have immersive experiences which reconnect them to the land they were born in and where the languages reside, helping them reconnect to the spirit of the language. Digital technology facilitates more efficient, cost-effective language documentation and material development through various media formats (e.g., image, audio, and video) (Outakoski et al., 2018).

Digital ancestral knowledge is essential to indigenous knowledge and education (Wemigwan, 2016). Furthermore, digitised indigenous knowledge has the potential to engage a broader audience, allowing individuals from various regions to participate in the preservation and development of the language beyond their own community. For example, our research project benefited from the digitalisation of archived indigenous language materials (project Dúchas[1]), which enabled us to repurpose the materials for DGBLL.

2.2.3. Irish cultural heritage and Dúchas resource

The National Folklore Collection (NFC) is a valuable cultural repository for Ireland. Its aims are to collect, preserve and disseminate the oral tradition of Ireland (Daly, 2010). The Dúchas project (Meitheal, 2022) has been running since 2012 (Ó Cleircín et al., 2014) and its goal is to digitise historical documents. The Schools' Collection (TSC) from NFC (e.g., 450k pages) has been scanned and indexed in the project and a considerable number of texts have been transcribed (e.g., 40k pages) from the collection (Raghallaigh et al., 2022). The Schools' Collection consists of essays gathered from over 50,000 schoolchildren from 5,000 schools in Ireland from 1937-1939. The schoolchildren wrote about folklore, mythology and local traditions and in all 740,000 pages were collected. These texts provide a unique insight into Irish life and the Irish language at that time.

While the availability of this resource is highly valuable and the perusal of the children's texts is captivating, the application of NLP techniques to these handwritten texts presents a challenge. Fortunately, there is a national crowdsourcing initiative (Meitheal, 2022) that leverages support from the community to transcribe these handwritten texts into a digital format. To date, 75% of the Irish texts have been digitally transcribed (Meitheal, 2022). These texts were subsequently reviewed by the Dúchas project team for quality assurance purposes.

2.2.4. Irish language technology

Although Irish is a less-resourced language in terms of NLP resources, there are two NLP tools that have been particularly useful in this project: the Irish POS tagger (Uí Dhonnchadha and Van Genabith, 2006) and Gramadóir (Scannell, 2005). The Irish POS tagger annotates a text with part-of-speech tags and lemmas. It uses Parole morpho-syntactic tags (Monachini and Calzolari, 1999) and XML Corpus Encoding Standard (Ide et al., 2000). Irish has a rich inflectional morphology and a rule-based approach is used throughout for the POS tagging, chunking and parsing tools. A rule-based approach rather than a machine learning approach was essential as there were no annotated corpora available for this first POS tagger for Irish. Rule-based approaches are particularly important for low-resourced

[1] www.dúchas.ie

languages. While the Irish POS tagger was primarily used to supply the *Cipher* engine with POS-tagged texts, it was also helpful in the initial classification of texts so that players would be shown a text appropriate for their level of Irish. As an Irish spelling and grammar checker, *Gramadóir* is used to identify misspelt or grammatically erroneous words in Irish text. Within the NLP pipeline of *Cipher*, *Gramadóir* is employed to spellcheck manually revised texts.

2.3. Motivation

The intrinsic/extrinsic motivation theory is built based on the self-determination theory (Ueno, 2005) which suggests human behaviours are self-determined (Deci and Ryan, 1985). Intrinsic motivation is driven by the rewards or punishments that people may receive from the activity itself while extrinsic motivation is driven by external rewards or punishments (Ueno, 2005). Theodoropoulos and Antoniou (2022) have found that games, particularly those in VR, have been used to increase cultural awareness and heritage appreciation due to the highly immersive nature of VR experiences. This project intends to explore the potential of VR as a means of reconnecting to the spirit of the language and thereby increasing learners' internal motivation. Meanwhile, game rewards provide external motivation for learners, as demonstrated in language learning applications like Duolingo.

2.3.1. Motivation in the Irish context

Although Irish is one of the three official languages of Ireland, it is only spoken daily by 1.5% of the population outside of the education system (CSO, 216). With some exceptions, Irish is a mandatory subject in both primary and post-primary education. However, many students lack the motivation to study the language, and this can make Irish lessons seem like drudgery rather than an enjoyable experience for some students. The teachers, who mostly are not native speakers, often must shoulder the responsibility of Irish language education as many parents are unable to help their children with Irish homework. Until recently, there were very few interactive resources for teaching Irish in the school context. While there are now some online resources linked to specific textbooks, they still remain few in number. As Sanacore (2007) outlined, for reluctant learners in general there is a need to provide challenging learning activities, offer student choice and provide opportunities for more active learning. A DGBLL app for Irish presents a potential means of providing these activities and opportunities.

2.4. Digital educational games

Dixon et al. (2022) mentioned that the efficacy of the DGBLL approach is heightened in games designed for entertainment as opposed to those designed for educational purposes. Nevertheless, entertaining games are less likely to provide a language option for minority languages due to a lack of commercial incentives. Educational applications incorporating gamification elements exist outside of the aforementioned classifications. Nonetheless, Dixon *et al*. (2022) included Duolingo in the research for DGBLL as Duolingo is an "edutainment" app with many game-like features and complex game mechanics. The *Cipher* game described in this paper primarily focuses on reading and word awareness, and leverages NLP tools and resources to make the game more language appropriate and variable. Focus on form is important in language learning and NLP resources can help in this regard (e.g., Meurer et al., 2010).

2.5. VR

Lan (2020) notes that VR's immersion and interaction features are critical factors that render it beneficial to both educators and learners. CALL researchers have made some effort to investigate the pedagogical perspectives of VR for language learning despite its original purposes (Lin and Lan, 2015). Lan (2020) suggests that various elements must be taken into account when applying VR to language learning, including learners' language proficiency and language acquisition process.

3. Research Questions

The following research questions (RQ) are primarily focused on three strands: technology, pedagogy, and socio-cultural context. These questions came about through an analysis of the associated research around CALL for LCTLs and from the progress and plan described in Sections 4 and 5.

- RQ1 How can existing game resources for dominant languages be repurposed for low-resourced languages for CALL?
- RQ2 How can specific technology (i.e., NLP, VR) be leveraged to enhance CALL resources?
- RQ3 How can language learning pedagogy be incorporated into DGBLL?
- RQ4 How can socio-cultural approaches be integrated into CALL for indigenous languages?

4. Progress

4.1. Repurpose game resources

This section aims to contribute to research question RQ1. Irish is an under-resourced language. If it is possible to repurpose resources from dominant languages (e.g., English), it will provide many language opportunities for under-resourced languages. In line with this objective, the present study successfully adapted a computer game in English to serve as a tool for Irish language education through the DGBLL concept. Furthermore, language independence is a design feature of this project, aimed at ensuring the adaptability of our work to other LCTLs.

The educational game introduced in this paper is based on the game *Cipher* (Xu and Chamberlain), which was designed for detecting errors in English text through the idea of games-with-a-purpose and crowdsourcing. The informal feedback from players indicated that *Cipher* has the potential for facilitating language learning. This provided the foundation for the further development of *Cipher*.

Based on the original *Cipher* game, the *Cipher* engine was enhanced to enable the creation of *Cipher* games in other languages. *Cipher: Faoi Gheasa* (*Cipher: under a spell*) was designed to encourage students, especially young learners, to learn Irish. To emphasise language learning purposes, new storyline and game elements were added to the game, to encourage "noticing" (spelling) of vocabulary, reading and writing. The game presents a magical world in which an evil spirit casts spells on ancient legends in which many ancient spirits dwell in order to make people forget

the ancient spirits and the past. In the game, the player needs to complete certain "tasks" before an evil spell can be lifted from the tales and ancient spirits can be saved. Game elements (e.g., spells, power-ups and ancient spirits) help to transform language tasks (including noticing, reading and writing) into interesting game tasks. (See Fig 1 for some screenshots of *Cipher*)

Fig 1: *Cipher* screenshots

Ideally, the player needs to have some degree of Irish knowledge (beginner level). However, with game elements like power-ups, we found that players can play and enjoy the game even if they do not have previous knowledge of Irish. During a user experience study in a primary school in Ireland where most students study Irish as a subject, there were a few students who did not study Irish but were still able to play the game and even enjoy it. Therefore, for some students who did not study Irish, the game became a language exposure experience.

4.2. Repurpose language resources

In relation to research questions RQ2 and RQ4, this section provides an overview of the process of repurposing stories from The Schools Collection (TSC) of the Irish NFC for use in *Cipher – Faoi Gheasa*, a DGBLL app for Irish. As part of the Dúchas Project (Ó Cleircín et al., 2014) approximately 450k pages were scanned, indexed and transcribed, and the transcribed text and metadata were made freely available online. This resource provided a rich source of reading material for the game. The aims of this game are to promote language awareness, vocabulary learning, reading and writing, all of which are known to be important in language learning. In the game, players are asked to spot "enchanted" words that have been selected randomly in the texts and players must also identify the type of spell (cipher) that was used. Moreover, research has shown that folklore and indigenous cultural elements can help learners reconnect to the language's spirit, which encourages indigenous language learning (Napier and Whiskeyjack, 2021).

There was a multi-stage process in the development of texts for *Cipher - Faoi Gheasa*. A variety of NLP tools were used to process these texts, in a series of cleaning, updating and annotation steps, into a format that can be used in the game by modern language learners. In step 1, TSC was searched to find suitable texts for the game. It was important that interesting and accessible stories were used, and that they have a magical or supernatural element. The metadata for each of the stories was useful in this regard. Once the original stories had been found, step 2 involved downloading the digitally transcribed stories and their metadata from TSC. There were several files related to each story including the actual digital text and the related metadata. In step 3, these texts were then manually reviewed and obvious changes were made to the text to convert it to modern standard Irish since these stories were written down almost a century ago by schoolchildren and some adjustments to spelling and grammar were deemed necessary.

Once the text had been adjusted, the text was then checked by *Gramadóir* for any remaining spelling errors in step 4. Once these errors had been corrected, in step 5, the cleaned text was then processed by the Irish POS tagger (Uí Dhonnchadha and Van Genabith, 2006). The POS-tagged texts were then processed by a chunker and noun phrase checker and additional errors spotted were then corrected. It is essential that only deliberately inserted errors (i.e., spells) are found in the text – other types of errors would be confusing for the players and interfere with the learning process. The XML POS-tagged file was then passed to the *Cipher* engine to produce *Cipher - Faoi Gheasa*. The POS tags include the gender of each noun in the text. This enables the *Cipher* engine to display masculine and feminine words in different colours in the game where red words are feminine and blue words are masculine.

The next stage in the processing is to determine the text difficulty level, i.e., whether a text is suitable for a beginner or more advanced learner and for progression purposes in the game. This involved the development of text complexity measures for Irish (Uí Dhonnchadha et al., 2022). Lexical diversity, lexical frequency and grammatical complexity were calculated using the POS-tagged and lemmatised text. This guided the objective classification of texts based on complexity scores and enables the *Cipher* engine to present texts of an appropriate level to the player. Further details pertaining to the NLP integration within *Cipher* are expounded upon in (Ward et al., 2022).

5. Results

5.1. Technical results

Existing and new NLP resources for Irish were leveraged successfully to provide the text resources for the *Cipher*. This involved a combination of automated and semi-automated processes. The *Cipher* development team was able to reuse the existing handwritten texts transcribed in digital format after they had been amended and adapted with the *Cipher* NLP pipeline. As the game is intended for language learning, all POS-tagged text is manually verified. The *Cipher* engine has been re-designed to be language-independent. It can accept annotated XML texts in any language and new language-specific or culture-specific ciphers can also be designed and implemented. The interface language has also been parameterised.

5.2. User experience study

The first school to use the app was an Irish medium primary school where students are taught through the medium of Irish and generally have higher levels of Irish than in English medium schools. The app was tested by approximately 20 students using Android tablets. Informal feedback from the students and teachers was positive. The second school was an English medium primary school and the students used laptops to play the game. The game was tested by more than 150 students across nine classes, who

either played the app individually or in pairs. Feedback from the students and teachers was positive. Further details on the user experience study are available in the study (Xu et al., 2022).

6. Work-in-Progress

6.1. Curriculum development

This section is intended to explore research question RQ3.

6.1.1. Game extension

The first phase of the research project was game adaptation as a proof of concept. If an Irish story is presented to students in the game, can they find the "disrupted" words and ciphers? Does the game work and is it suitable for primary school students? Moving further, the next phase is to focus on the pedagogy and make sure the game not only can encourage young learners to interact with Irish learning materials in a fun way but also improves their language skills in a standardised way. In phase 1, the game is mostly based on reading. To align with the curriculum, vocabulary learning and writing tasks are added to the game. The vocabulary part, also called "letter bricks", will pre-teach some words so that learners will learn some vocabulary before the reading tasks. These words are related to the reading material they will be given later in the game. The writing task, also called "word bricks", will be presented after the reading task. The writing task is also based on the reading material. This might sound slightly dull but both vocabulary learning and writing tasks are designed with game elements to make them engaging. The entire game is centred around the themes of steganography and mythology. This part of the work has already been implemented in the game and is ready to test in schools once the curriculum design is completed. (See Fig 2)

Fig 2: Vocabulary learning task in the game

6.1.2. Curriculum design

In game-based learning applications, gaming alone does not make learning happen, it is the pedagogy that supports learning in the game. We are working with primary school teachers on aligning *Cipher* with the primary school curriculum for Irish. We are designing the language learning content in the game in a way that is pedagogically appropriate. We are using stories and current versions of Irish myths to make them more appealing and suitable for gameplay and language learning. The thematic approach of the Irish curriculum might help with scaling the game or providing a framework around to build stories. Through the main theme of myths, there are sub-themes of the Irish curriculum (e.g., weather, clothes, family). We intend to add challenges to the game for tests, which are appropriate for children with a certain level (e.g.,4th grade).

6.2. VR

VR technology will be employed to provide enhanced exposure to the folklore and mythology of the target language with the aim of enhancing culture and language awareness among users (See Fig 3). This exposure will help learners reconnect to the spirit of the language, thereby augmenting their intrinsic motivation. By building the system on top of an existing language-independent game resource *Cipher*, user acceptance risk and implementation processes can be greatly reduced. This segment of the study will contribute to addressing research questions RQ2 and RQ4.

Fig 3: *Cipher* VR blueprint

7. Conclusion

While developing CALL for less-resourced languages is more challenging, it is not impossible. This paper demonstrates that a structured and creative use of existing technology and resources that are pedagogically suitable and appropriate as well as engaging for language learners can be used to good effect to develop CALL for LCTLs. There are future improvements planned and research will continue into this novel approach within our framework represented above.

Acknowledgement

This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. Our gratitude extends to Tianlong Huang for providing support in game development.

References

Blake, R. J. (2011). *Current Trends in Online Language Learning.* Annual Review of Applied Linguistics, 31, pp. 19–35.

Council, B. (2016). *English in numbers.* Retrieved from: https://www.britishcouncil.cn/en/EnglishGreat/numbers. (last access date: 2023-02-20)

CSO. (216). *Press Statement Census 2016 Results Profile 10 - Education, Skills and the Irish Language.* Retrieved from:https://www.cso.ie/en/csolatestnews/pressreleases/2017pressreleases/pressstatementcensus2016resultsprofile10-educationskillsandtheirishlanguage. (last access date: 2023-02-20)

Daly, M. E. (2010). *'The State Papers of a forgotten and neglected people'; the National Folklore Collection and the writing of Irish history.* Béaloideas, pp. 61–79.

Deci, E. L., and Ryan, R. M. (1985). *The General Causality Orientations Scale: Self-determination in Personality.* Journal of Research in Personality, 19 (2), pp. 109–134.

Dixon, D. H., Dixon, T. and Jordan, E. (2022). *Second Language (L2) Gains through Digital Game-Based Language Learning (DGBLL): A Meta-Analysis.* Language Learning & Technology, 26(1), pp. 1-25.

Galla, C. K. (2016). *Indigenous language revitalization, promotion, and education: function of digital technology.* Computer Assisted Language Learning, 29 (7), pp. 1137–1151.

Galla, C. K. (2018). *Digital Realities of Indigenous Language Revitalization: A Look at Hawaiian Language Technology in the Modern World.* Language and Literacy, 20 (3), pp. 100– 120.

Ide, N., Bonhomme, P., & Romary, L. (2000). XCES: *An XML-based Encoding Standard for Linguistic Corpora.* Proceedings of the Second International Language Resources and Evaluation Conference. Paris: European Language Resources Association.

Lan, Y.-J. (2020). *Immersion into virtual reality for language learning.* In Psychology of Learning and Motivation, pp. 1–26. Elsevier.

Lin, T.-J., & Lan, Y.-J. (2015). *Language Learning in Virtual Reality Environments: Past, Present, and Future.* Journal of Educational Technology & Society, 18 (4), pp. 486–497.

Meitheal. (2022). *Meitheal Dúchas.ie Volunteer Transcription Project.* Retrieved from: https://www.duchas.ie/en/meitheal. (last access date: 2023-02-20)

Meurers, D., Ziai, R., Amaral, L., Boyd, A., Dimitrov, A., Metcalf, V. and Ott, N. (2010). *Enhancing Authentic Web Pages for Language Learners*. NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications, Los Angeles, California.

Millour, A. and Fort, K. (2020). *Text Corpora and the Challenge of Newly Written Languages.* 1st Joint SLTU and CCURL Workshop (SLTU-CCURL 2020).

Monachini, M., and Calzolari, N. (1999). *Standardization in the Lexicon.* Syntactic Wordclass Tagging, Volume 9, Springer-Dordrecht, pp. 149-174.

Napier, K. and Whiskeyjack, L. (2021). *wahkotowin: Reconnecting to the Spirit of nêhiyawêwin (Cree Language)*. Engaged Scholar Journal, 7(1), pp. 1-24.

Ó Cleircín, G., Bale, A., & Ó Raghallaigh, B. (2014). *Dúchas. ie: ré nua i stair Chnuasach Bhéaloideas Éireann.* Béaloideas, pp. 85-99.

Outakoski, H., Cocq, C., & Steggo, P. (2018). *Strengthening Indigenous Languages in the Digital Age: Social Media–Supported Learning in Sápmi.* Media International Australia, 169(1), pp. 21-31.

Purgina, M., Mozgovoy, M., & Ward, M. (2017). *MALL with WordBricks–building correct sentences brick by brick.* CALL in a climate of change: adapting to turbulent global conditions–short papers from EUROCALL, pp. 254-259.

Raghallaigh, B. Ó., Palandri, A., & Mac Cárthaigh, C. (2022, June). *Handwritten Text Recognition (HTR) for Irish-Language Folklore.* In Proceedings of the 4th Celtic Language Technology Workshop within LREC2022, pp. 121-126.

Sanacore, J. (2007). *Needed: Critics of Literacy Education with a More Inclusive Perspective.* International Journal of Progressive Education, 3 (1), pp. 29–43.

Scannell, K. (2005). *An Gramadóir.*

UNESCO. (2022). *A decade to prevent the disappearance of 3,000 languages.* Retrieved from: https://www.iesalc.unesco.org/en/2022/02/21/a-decade-to-prevent-the-disappearance-of-3000-languages. (last access date: 2023-02-20)

Uí Dhonnchadha, E. and Van Genabith, J. (2006). *A Part-of-speech tagger for Irish using Finite-State Morphology and Constraint Grammar Disambiguation.* LREC 2006, pp. 2241-2244.

Uí Dhonnchadha, E., Ward, M., and Xu, L. (2022). *Cipher–Faoi Gheasa: A Game-with-a-Purpose for Irish.* Proceedings of the 4th Celtic Language Technology Workshop within LREC2022, pp. 77–84.

Ueno, J. (2005). *An Analysis of Leamer Motivation of Less Commonly Taught Languages.*

Ward, M. (2015). *CALL and less commonly taught languages: Challenges and opportunities.* Research-publishing. net.

Ward, M. (2018). *Qualitative research in less commonly taught and endangered language CALL.* Language Learning & Technology, 22 (2), pp. 116–132.

Ward, M. (2019). *Joining the blocks together – an NLP pipeline for CALL development.* CALL and Complexity, pp. 397.

Ward, M., Xu, L., and Uí Dhonnchadha, E. (2022). *How NLP Can Strengthen Digital Game-Based Language Learning Resources for Less Resourced Languages.* Proceedings of the 9th Workshop on Games and Natural Language Processing within the 13th Language Resources and Evaluation Conference, pp. 40–48.

Wemigwans, J. (2016). *A Digital Bundle: Exploring the Impact of Indigenous Knowledge Online Through FourDirectionsTeachings.com* (Doctoral dissertation). University of Toronto (Canada).

Xu, L., & Chamberlain, J. (2020). *Cipher: A Prototype Game-with-a-Purpose for Detecting Errors in Text.* Workshop on Games and Natural Language Processing, pp. 17–25.

Xu, L., Uí Dhonnchadha, E. and Ward, M. (2022). *User Experience Study of "Cipher: Faoi Gheasa", A Digital Educational Game for Language Learning and Student Engagement.* Proceedings of the 2nd Workshop on Games Systems, pp. 5–8.

Improving Hate Speech Detection with Self-Attention Mechanism and Multi-Task Learning

Nicolas Zampieri, Irina Illina, Dominique Fohr

Lorraine University, CNRS, Inria, Loria, F-54000 Nancy, France

Abstract

Hate speech detection is a challenging task of natural language processing. Recently, some works have focused on the use of multiword expressions for hate speech detection. In this paper, we propose to use an auxiliary task to improve hate speech detection: multiword expression identification. Our proposed system, based on multi-task with self-attention, outperforms an MWE-based features state-of-the-art system on four hate speech corpora.

Keywords: hate speech, detection, neural networks, self-attention, multi-task

1. Introduction

Social media have an important place in today's society, in particular thanks to their forms of communication which are intended to be instantaneous and uncensored. Social networks make possible to communicate an idea, a thought, or any other form of the message whether it is harmful or not. Millions of messages are posted every day: e.g. Twitter with around 500 million posts every day (Bendler et al., 2014). Each social media has its own definitions of unwanted content. Hate speech is part of the unwanted content of social media and is punished by several countries. Automatic detection of hateful contents is essential due to the huge amount of posts on social media.

According to the Committee of Ministers of the Council of Europe, hate speech is "*any types of expression that incite, promote, spread or justify violence, hatred or discrimination against a person or group of persons, or that denigrates them, by reason of their real or attributed personal characteristics or status such as race, color, religion etc.*"[1].

Hate Speech Detection (HSD) is a challenging task in the field of natural language processing. The nature of social media posts makes it difficult to detect hate speech, especially in Twitter posts (tweets). Indeed, tweets consist of short texts (maximum of 280 characters) that often employ non-standard syntax, and can contain misspellings, abbreviations, or even non-texts (e.g., emojis, images, and URLs). Annotate hate speech corpus is time consuming and expensive, so there are only a few annotated corpora.

Nowadays, state-of-the-art systems in this field are based on Deep Neural Networks (DNN). Chakrabarty et al. (2019) studied the impact of self-attention and contextual-attention. Kapil et al. (2020) explored multi-task learning, in parallel on five hate speech corpora. Awal et al. (2021) developed the AngryBERT system, which was trained on HSD and sentiment classification tasks.

In this article, we propose to incorporate syntactic and semantic information in a DNN-based system to improve HSD. Syntactic and semantic information will be learned from the Multiword Expression (MWE) identification task. MWE is a group of words (more than two lexemes) that express some form of idiosyncrasy: lexical, morphological, syntactic, semantic, and/or statistic (Baldwin and Kim, 2010) (e.g., *shut up*, *break a leg*, *black and white*). An MWE can be idiomatic or noun compound, and can have several meanings if we consider the word-by-word meaning of the MWE or if we take the meaning of the words composing an MWE as a single lexical unit. For example, *break a leg* could mean *good luck* when it is an idiom MWE and depends on the context of the sentence.

Multiword expressions and HSD have been the subject of some recent studies. Ptaszynski et al. (2017) proposed the use of morphosemantic patterns, such as part-of-speech and semantic role. Stankovic et al. (2020) extended a Serbian lexicon of abusive language with special attention to MWEs and proposed to exploit it to create an abusive corpus for the Serbian language. Zampieri et al. (2021) developed a DNN-based system that uses MWE features. MWE features have been integrated into a DNN-based system that utilizes MWE categories. Zampieri et al. (2022) compared the impact of two MWE identification systems: the first is based on a lexicon and the second is based on DNN. These works have shown that MWEs are helpful for the HSD task.

In this article, we propose a new HSD-system based on MWE which outperforms the system proposed by Zampieri et al. (2022). Our system uses self-attention mechanism and multi-task learning. The advantage of our system is to learn MWE and hate speech features thanks to a self-attention layer. Compared to Chakrabarty et al. (2019), we use multi-head self-attention. In contrast with Kapil et al. (2020), where several corpora were used for the same task, we use two different tasks.

2. Methodology

Zampieri et al. (2021) and Zampieri et al. (2022) showed that system using MWE features outperforms system without MWE features on the HSD task. In this current work, we pursue this idea in the framework of multi-task learning.

[1] https://www.coe.int/en/web/freedom-expression/hate-speech

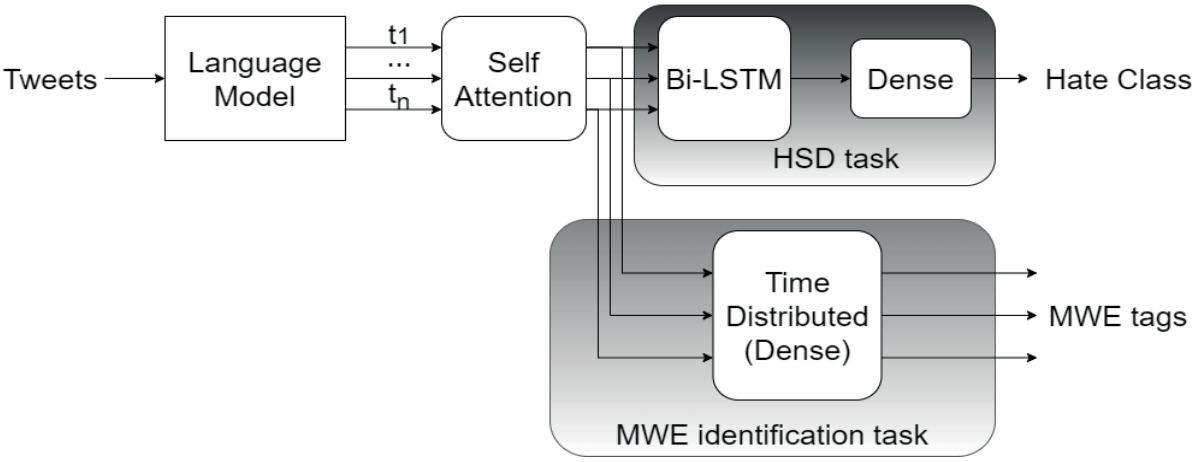


Fig. 1: Proposed HSD system based on multitask learning with self-attention.

The two tasks are MWE identification and HSD. Moreover, we propose to share, between the two tasks, a multi-head self-attention layer proposed by Vaswani et al. (2017) in order to learn attention simultaneously from both tasks. We believe that attention to MWEs could help the system distinguish hate from non-hate speech.

Figure 1 shows the architecture of our proposed HSD system based on multi-task learning and self-attention. The self-attention layer is there to learn representations considering the two tasks. Our system uses the contextual token embeddings provided by the outputs of a pre-trained language model like BERT-based models (Delvin et al., 2019). These embeddings are given as input to a self-attention layer. For the HSD task, we utilize a bidirectional long short-term memory layer followed by a dense layer. The final prediction is made from the output of specialized dense HSD task layer. For the MWE identification task, a dense time-distributed layer is used, and the outputs are formatted as "BIOo": each lexical unit is tagged "B" if it is at the start of an MWE, "I" if it is inside an MWE, "O" if it does not belong to an MWE. The "o" tag has the same meaning as the "O" tags, but the word is nested in an enclosing MWE.

We compare our approach with a baseline system trained only on the HSD task without self-attention. To perform the training of the multi-task system, we need a HSD corpus annotated in terms of MWEs. Since no such corpus exists, we utilize the predictions provided by the deep neural network MWE identification system of Zampieri et al. (2022). Compared to the work of Zampieri et al. (2022), where MWE features are used at the input of the DNN-based system, we design a multi-task approach that consists of hate speech detection and MWEs identification by using a self-attention mechanism.

3. Experimental Setup

In this section, we describe hate speech corpora and system configuration.

3.1. Datasets

Waseem and Hovy (2016) corpus (**Waseem**) contains 16,919 tweets annotated in three classes: sexist, racist, and neither. We recovered 10,807 tweets because some tweets have been removed from social media. We focus on HSD task, so we combine the sexist and the racist classes into single class: hate class. Tweets labeled as ''neither'' are considered to belong to the non-hate class. The corpus contains 73% and 27% of non-hateful and hateful tweets, respectively.

Davidson et al. (2017) corpus (**Davidson**) is a corpus annotated in terms of hate speech, offensive speech, or neither. The corpus contains 24,802 tweets: 76% are offensive, 7.4% hateful, and 16.6% neither. We do not merge offensive and hate speech classes, as the corpus is designed to distinguish offensive content from hateful content.

Founta et al. (2018) corpus (**Founta**) contains 100k tweets, annotated in four classes: hateful, abusive, normal, and spam. Our experiments focus on HSD, so we remove spam tweets, and we keep around 86k tweets. The corpus contains 63% normal, 31% abusive, and 6% hateful tweets. As in the Davidson dataset, we do not aggregate abusive and hateful tweets under the same label.

Basile et al. (2019) corpus (**HatEval**) is a balanced corpus annotated in hate and non-hate speech: 42% hateful and 58% non-hateful tweets. It contains 13k tweets and is partitioned into training, development, and test sets with 9k, 1k, and 3k tweets, respectively. It is provided by the SemEval2019 shared task 5.

For Waseem, Davidson, and Founta datasets, we utilize 60%, 20%, and 20% for **training**, **validation**, and **testing sets**, respectively. For the HatEval corpus, we use the standard corpus partition of the SemEval shared task 5. We apply the same preprocessing as in Zampieri et al. (2022): we remove mentions, hashtags, URLs, and we replace emojis with readable text (e.g., ♥ → *:heart:*).

3.2. MWE identification system

To annotate the MWEs on the four tweet corpora, we use the transformer-based system proposed by Liu *et al.* (2021). Indeed, Zampieri et al. (2022) showed that this MWE identification system achieves better performance than a lexicon-based approach. In our study, we apply the same configuration of the MWE identification system as in Zampieri et al. (2022). We use this MWE identification system to automatically annotate hate speech corpora in terms of MWEs. The MWE identification system tagged about 4k, 9k, 10k and 46k MWEs in Waseem, HatEval, Davidson, and Founta training sets, respectively.

HSD Systems	#Head	Binary classification		Ternary classification		Average
		Waseem	HatEval	Davidson	Founta	
Zampieri et al. (2022)	-	81.9 (±0.6)	64.6 (±1.1)	73.9 (±1.4)	74.0 (±0.7)	73.5
BERTweet embeddings						
Single task (baseline)	-	82.8 (±0.5)	61.1 (±4.8)	71.0 (±0.6)	73.7 (±1.0)	72.2
Single task	2	84.5 (±0.8)	61.5 (±2.3)	72.0 (±3.2)	74.0 (±0.9)	73.0
	4	84.3 (±2.0)	**64.4** (±3.7)	73.2 (±2.3)	**74.1** (±1.1)	74.0
Multi-task	2	**85.5** (±2.2)	64.2 (±1.8)	**74.2** (±1.0)	73.5 (±0.4)	**74.5**
	4	85.1 (±0.7)	63.3 (±4.6)	73.6 (±2.2)	**74.1** (±1.1)	74.0
HateBERT embeddings						
Single task (baseline)	-	81.6 (±1.2)	63.1 (±3.6)	71.9 (±3.5)	74.3 (±0.6)	72.7
Single task	2	82.4 (±0.6)	61.0 (±2.4)	72.8 (±3.1)	73.8 (±1.7)	72.5
	4	82.7 (±1.6)	60.4 (±2.9)	73.2 (±1.8)	74.4 (±0.7)	72.7
Multi-task	2	83.2 (±0.8)	63.7 (±2.6)	**75.1** (±2.0)	**74.6** (±0.3)	**74.2**
	4	**83.5** (±2.3)	**65.0** (±1.7)	73.3 (±3.1)	73.4 (±1.2)	73.8

Table 1: Median macro-F1 of HSD and standard deviation of 5 runs. The column #Head represents the number of heads for the attention layer. The results that are significantly better than the ''baseline'' systems are underlined. The *Average* column represents the average of median macro-F1 on the four corpora and the significant improvement is computed by merging all predictions.

Note that in this article, we do not evaluate our proposed multi-task system on the MWE identification task because the corpora used are not labeled in terms of MWE.

3.3. Hyperparameters of HSD system

To generate contextual token embeddings, we use state-of-the-art transformers-based models trained on tweets or hateful data: the BERTweet-base model (Nguyen et al., 2020), the HateBERT model (Caselli et al., 2021), and the fBERT model (Sarkar et al., 2021). The BERTweet model is trained on tweets. The HateBERT model is trained on Reddit comments that potentially contain abusive or hateful speech. The fBERT model is a BERT-based model fine-tuned on offensive tweets. Sarkar et al. (2021) showed that the fBERT model outperforms the HateBERT model. However, in our preliminary experiments, we found that the fBERT embeddings achieves lower performance compared to the two other models. So, in this article, we are experimenting with the BERTweet and the HateBERT embeddings.

For the MWE identification task, we use a ''IO'' tagging scheme with two labels: if a word belongs to an MWE, then it is tagged by ''I'', otherwise it is tagged by ''O''. Concerning the HSD task, we use a bidirectional long short-term memory layer with 128 neurons and followed by a dense layer. The output size of the bidirectional long short-term memory is 256.

3.4. Evaluation Metrics

We evaluate our models in terms of macro-average F1. It is the average of the F1 scores of all classes. We compute the median macro-F1 score over 5 runs. We use a matched pairs test with a 5% risk (Gillick and Cox, 1989) to determine if there is a significant improvement compared to the baseline system.

4. Results

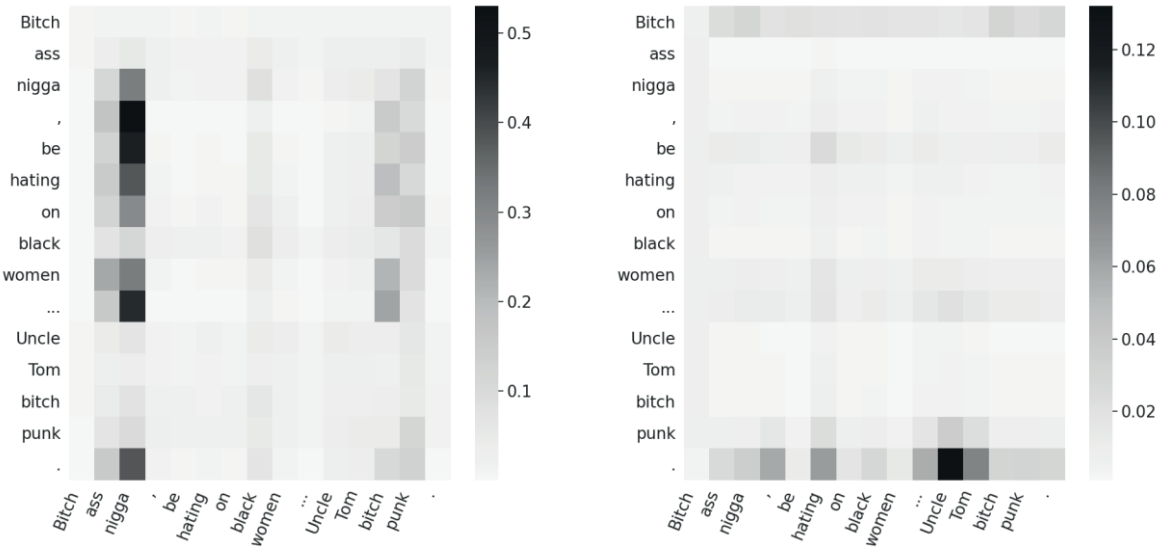
The goal of our experiments is to improve the performance of the HSD task using the MWE identification task. We study the effect of the self-attention mechanism on the HSD task. Moreover, we assess the multi-task approach using two different contextual token embeddings. We compare our new approach with the approach proposed by Zampieri et al. (2022) as they obtained good performance for the hate detection task and they also used MWEs.

Table 1 shows that our approach based on self-attention with multi-task learning outperforms the Zampieri et al. (2022) system for all datasets. Our best configuration of HSD system with multi-task learning, two heads of self-attention and BERTweet embeddings improves the average score by 1% relative compared to the Zampieri et al. (2022) HSD system (74.5% versus 73.5%). The best improvement is achieved for Waseem test corpus with an increase of 3.6% relative (85.5% versus 81.9%).

4.1. Impact of the self-attention mechanism

For the BERTweet embeddings and the single task approach, we find that using 2 attention heads does not significantly improve the average score compared to Zampieri et al. (2022) system. Using 4 attention heads, the average macro-F1 score is significantly better than baseline: 74.0% versus 72.2%. This improvement is observed in three corpora: Waseem, HatEval and Davidson. Using more than 4 self-attention heads does not provide any further improvement and it is not shown here.

Regarding the use of HateBERT embeddings, we do not observe an improvement using the self-attention mechanism compared to the baseline: the baseline achieves 72.7% versus 72.5% and 72.7% using 2 and 4 attention heads, respectively. It can be due to the fact that there is a mismatch between the training HateBERT embeddings (on Reddit) and testing on tweets.



(a) Weights of the first head of self-attention (b) Weights of the second head of self-attention

Fig. 2: Attention weights of the multi-task system with two heads (a and b) and the BERTweet embeddings. Example from the Davidson development set: ''*Bitch ass nigga, be hating on black women... Uncle Tom bitch punk.*''. Two MWEs are detected: *Bitch ass nigga* and *Uncle Tom*.

4.2. Impact of the multi-task learning

Table 1 shows that in the case of BERTweet embeddings, the multi-task system significantly outperforms the baseline system: the baseline reaches 72.7% of the average macro-F1 score, compared to 74.5% and 74.0% using 2 and 4 attention heads with multi-task learning, respectively. This is the case for 3 corpora. However, multi-task systems do not outperform single task with 4 attention heads. Using HateBERT embeddings, all multi-task configurations significantly outperform single-task systems: multi-task systems obtained 74.2% and 73.8% of the average macro-F1 scores compared to 72.5% and 72.7% obtained by single-task systems using 2 and 4 self-attention heads, respectively.

It is important to note that multi-task performance is obtained using an automatic MWE tagging system (used only during training). As Zampieri et al. (2022), this confirms that MWEs are helpful for the HSD task. For the two studied embeddings, the best performance is achieved by the multi-task system with 2 attention heads: 74.5% and 74.2% using BERTweet and HateBERT, respectively.

For further analysis, Figure 2 provides an example of the weights of the multi-task system with two self-attention heads. The example is extracted from the Davidson development set. We observe in some samples that each self-attention head often focuses on one task: the first head (2a) tends to specialize on harmful words (*ass*, *nigga* and *bitch*) and the second (2b) on MWEs (*Uncle Tom*).

4.4. Limitations

One limitation of our approach is the fact that it requires both MWE and hate/non-hate annotations of the data. To the best of our knowledge, such corpus does not exist. Therefore, in this work, we used an automatic MWE annotation system. The performance of the multi-task system may depend on the accuracy of this automatic MWE annotation system. As no corpus annotated in terms of both MWE and hate speech is available, we cannot fine-tune the BERTweet or HateBERT models for multi-task learning.

5. Conclusions

In this work, we investigated the impact of the self-attention mechanism and the multi-task learning for the hate speech detection. The two tasks that we want to investigated are the MWE identification task and the hate speech detection task. We carried out our experiments on four corpora and using two contextual embeddings: BERTweet and HateBERT. We observed that multi-task system significantly outperforms the baseline single task system. The best performance is obtained using the multi-task system with two attention heads.

For future work, we would like to take advantage of multi-task and multi-corpus approaches: MWE-annotated corpus and hate-speech-annotated corpus can be used simultaneously to train the system.

6. Acknowledgements

Experiments presented in this article were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations.

References

Awal, M.R., Cao, R., Lee, R.KW., Mitrović, S. (2021). AngryBERT: Joint Learning Target and Emotion for Hate Speech Detection. In *Advances in Knowledge Discovery and Data Mining*. PAKDD Lecture Notes in Computer Science, Vol. 12712. Springer International Publishing.

Baldwin, T., and Kim, S. N. (2010). Multiword expressions. *Handbook of Natural Language Processing*, pp. 267–292. CRC Press, Taylor and Francis Group, Boca Raton, 2nd edition.

Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Rangel Pardo, F. M., and Sanguinetti, M. (2019). SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 54–63. Association for Computational Linguistics.

Bendler, J.T., Brandt, T. L., Wagner, S. and Neumann, D. (2014). Investigating Crime-to-Twitter Relationships in Urban Environments - Facilitating a Virtual Neighborhood Watch. *European Conference on Information Systems*.

Caselli, T., Basile, V., Mitrović, J., and Granitzer, M. (2021). HateBERT: Retraining BERT for Abusive Language Detection in English. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pp. 17–25. Association for Computational Linguistics.

Chakrabarty, T., Gupta, T., and Muresan, S. (2019). Pay "Attention" to your Context when Classifying Abusive Language. In *Proceedings of the Third Workshop on Abusive Language Online*, pp. 70–79. Association for Computational Linguistics.

Davidson, T., Warmsley, D., Macy, M. W., and Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. *CoRR, abs/1703.04009*. http://arxiv.org/abs/1703.04009

Davidson, T., Warmsley, D., Macy, M., and Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, pp. 512–515.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1, pp. 4171–4186.

Founta, A., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G., and Kourtellis, N. (2018). Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 12.

Gillick, L., and Cox, S. J. (1989). Some statistical issues in the comparison of speech recognition algorithms. *International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 532–535.

Kapil, P., and Ekbal, A., (2020). A deep neural network based multi-task learning approach to hate speech detection. *Knowledge-Based Systems*, Vol. 210.

Liu, N. F., Hershcovich, D., Kranzlein, M., and Schneider, N. (2021). Lexical Semantic Recognition. *Proceedings of the 17th Workshop on Multiword Expressions (MWE 2021)*, pp. 49–56. Association for Computational Linguistics.

Nguyen, D. Q., Vu, T., and Tuan Nguyen, A. (2020). BERTweet: A pre-trained language model for English Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 9–14. Association for Computational Linguistics.

Ptaszynski, M., Masui, F., Nakajima, Y., Kimura, Y., Rzepka, R., and Araki, K. (2017). A Method for Detecting Harmful Entries on Informal School Websites Using Morphosemantic Patterns. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 21, No. 7, pp. 1189–1201.

Sakar, D., Zampieri, M., Ranasinghe, T., and Ororbia, A. (2021). fBERT: A Neural Transformer for Identifying Offensive Content. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*, pp. 1792–1798. Association for Computational Linguistics.

Savary, A., Cordeiro, S., and Ramisch, C. (2019, August). Without lexicons, multiword expression identification will never fly: A position statement. In *Proceedings of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019)*, pp. 79–91. Association for Computational Linguistics.

Stanković, R., Mitrović, J., Jokić, D., and Krstev, C. (2020). Multi-word Expressions for Abusive Speech Detection in Serbian. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*, pp. 74–84. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is All you Need. In *Advances in Neural Information Processing Systems.* NeurIPS, Vol. 30, pp. 5998–6008.

Waseem, Z., and Hovy, D. (2016). Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pp. 88–93. Association for Computational Linguistics.

Zampieri, N., Illina, I., and Fohr, D. (2021). Multiword Expression Features for Automatic Hate Speech Detection. Dans E. Métais, F. Meziane, H. Horacek, and E. Kapetanios (Éd.), *Natural Language Processing and Information Systems*, pp. 156–164. Springer International Publishing.

Zampieri, N., Ramisch, C., Illina, I., and Fohr, D. (2022). Identification of Multiword Expressions in Tweets for Hate Speech Detection. In *Proceedings of the 13th Language Resources and Evaluation Conference*, pp. 202–210. European Language Resources Association.

Zhang, Z., Robinson, D., and Tepper, J. (2018). Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. Dans A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, M. Alam (Éd.), *The Semantic Web*, pp. 745–760. Springer International Publishing.

Utilizing BERT with Auxiliary Sentences Generation to Improve Accuracy of Japanese Aspect-based Sentiment Analysis Task

Yiyang Zhang[1], Masashi Takeshita[1], Rafal Rzepka[2], Kenji Araki[2]

[1]Graduate School of Information Science and Technology, Hokkaido University
[2]Faculty of Information Science and Technology, Hokkaido University
Email: {yiyang.zhang, takeshita.masashi, rzepka, araki}@ist.hokudai.ac.jp

Abstract

Aspect-based Sentiment Analysis (ABSA) is an sentiment analysis task to determine the polarity of opinions regarding a particular aspect of a text. ABSA is a fine-grained sentiment analysis that can handle more detailed and useful information, hence it continues to attract attention in the NLP field. Although many studies have focused on English as the target language, there are still few studies focused on Japanese. Therefore, in this study, we aim to improve the accuracy of the Aspect Extraction (AE) subtask and the Aspect Sentiment Classification (ASC) subtask of the Japanese ABSA task. We improved the method proposed for the English ABSA task, which is to automatically generate auxiliary sentences and then fine-tune the BERT with sentence pairs that combine target sentences and the generated auxiliary sentences, and applied it to the Japanese ABSA task. As a result of the performance evaluation experiment based on a Japanese ABSA dataset called chABSA, compared to the results of the previous study, the F1-score improved by 7.54 points in the AE subtask and the accuracy improved by 3.1 points in the ASC subtask, confirming the effectiveness of the proposed method.

Keywords: Aspect-Based Sentiment Analysis, Aspect Extraction, Aspect Category Detection, Aspect Sentiment Classification

1. Introduction

Aspect-Based Sentiment Analysis (ABSA) is the task that determines the sentiment polarity towards a specific aspect in a sentence. For example, in the sentence "The sushi at this restaurant is good, but the ramen is bad", there are two aspects of the object "this restaurant": "sushi" and "ramen". Since the speaker evaluates "sushi" as "delicious," he has a "positive" feeling toward the aspect of "sushi," while he evaluates "ramen" as "bad," he has a "negative" feeling toward the aspect of "ramen. ABSA is expected to be used in many situations because it can perform sentiment analysis at a finer granularity than document-level sentiment analysis and sentence-level sentiment analysis, and can provide a relatively large amount of information. For example, if a company wants to analyze how consumers feel about each aspect of a product that the company has released, it is necessary to conduct an aspect-level sentiment analysis. The ABSA task consists of two main subtasks: the first is Aspect Extraction (AE), which extracts aspects contained in sentences. The second task is Aspect Sentiment Classification (ASC), which identifies the sentiment polarity of each aspect. In this paper, we focus on both of these two subtasks in ABSA task.

While there have been many studies on ABSA with English as the target language, to the best of authors knowledge there are only three studies on ABSA with Japanese as the target language, and it can be expected to achieve higher accuracy by using better methods.

This study aims to achieve a higher accuracy in Aspect Extraction (AE) subtask and Aspect Sentiment Classification (ASC) subtask than previous ABSA studies using Japanese as the target language. We improved the method Sun et al. (2019) proposed for English, which automatically generates auxiliary sentences and fine-tuning the BERT model with sentence pairs combining the target sentences and the auxiliary sentences, and applied it to Japanese. We conducted an experiment based on a Japanese ABSA dataset called chABSA. Eventually, in both AE and ASC subtasks of ABSA, we achieved better results than previous studies based on the same dataset. We also studied how to apply this method to the ASC subtask. This paper is based on our findings described in technical reports written in Japanese (Zhang et al., 2022) (Zhang et al., 2023).

2. Related work

Early ABSA research focused primarily on feature engineering (Wagner et al., 2014) (Kiritchenko et al., 2014) and deep learning techniques (Nguyen and Shirai, 2015) (Wang et al., 2016) (Tang et al., 2016) (Wang et al., 2017) (Ma et al., 2018) (Liu et al., 2018) suitable for ABSA task. Recently, pre-trained language models like ELMo (Peters et al., 2018), GPT (Radford et al., 2018), BERT (Devlin et al., 2019) have shown the ability to reduce the complexity of feature engineering. In particular, since BERT was pre-trained in the task of next sentence prediction, it performs well in tasks that require an understanding sentence-sentence relationships, such as the QA and NLI. Thus, Sun et al. (2019) proposed a method that utilizes this advantage of BERT and achieved new state-of-the-art results on English ABSA datasets.

In the studies using Japanese as the target language, Akai and Atsumi (2019) applied a neural network model using the self-attention mechanism used in a previous study using English as the target language to Japanese, and conducted experiments on sentences in the KNB corpus (Hashimoto et al., 2011) with emotion tags for reputation information. The accuracy of the sentiment classification reached 85%. Miura et al. (2020) proposed a self-attention neural network

model incorporating BERT, which consists of an aspect category classification network and an aspect sentiment analysis network. Experimental results show that the model achieves up to 85.6% accuracy in the AE subtask and up to 70.68% of F1-score in the ASC subtask.

However, there are two main problems in the aforementioned Miura et al. (2020) study using Japanese as the target language. In the Miura et al. (2020)'s method, each sentence is input directly into BERT for fine-tuning, which does not take advantage of the feature of BERT that allows it to perform better on sentence pair classification problems such as QA and NLI tasks. The second is the way to evaluate the performance of the model in the AE subtask. In the AE subtask of the Miura et al. (2020)'s study, the performance was evaluated by changing the dropout rate on test data, but we think it is more appropriate to find the dropout rate that achieves the best accuracy on validation data and evaluate the model performance using that dropout rate. We hypothesize that this method of performance evaluation is more appropriate.

In order to solve the above problems, this study aims to find a more suitable fine-tuning method for BERT, and evaluate the performance of the model in the AE subtask using a more appropriate performance evaluation method, and eventually achieve better results in the aspect category detection task, i.e., the AE subtask, and the ASC subtask achieved by Miura et al. (2020).

3. Propose method

In this study, we apply the method proposed by Sun et al. (2019), which automatically generates auxiliary sentences and then fine-tunes BERT with sentence pairs generated by combining the target sentence and the auxiliary sentence, which we think is a more suitable fine-tuning method for BERT, to Japanese. We improved the method of constructing auxiliary sentences in order to find a more suitable fine-tuning method for BERT than the original method, and because of some characteristics of Japanese, we also studied how to apply this method for the ASC subtask.

The following sections describe the BERT model, auxiliary sentence generation methods, and the BERT-pair model, which is fine-tuned by sentence pairs.

3.1. BERT

In this study, we conduct experiments using the pre-trained Japanese BERT-base model[1] provided by Tohoku University, which is pre-trained on the Japanese version of Wikipedia.

3.2. Methods of auxiliary sentences generation

3.2.1. Aspect Extraction (AE) subtask

The methods for generating auxiliary sentences in AE subtasks is shown in **Table 1**. Each of the auxiliary sentence generation methods is described below in detail.

- QA: In the Question Answering method, auxiliary sentences are generated in the form of questions, and the system is asked to answer those questions. The specific method is to generate auxiliary sentences by combining the aspect category and the sentence "が含まれていますか" (Does it contain [aspect category]).
- NLI: Natural Language Inference method generates auxiliary sentences using only aspect categories.

Table 1: Method of auxiliary sentences generation in AE subtask

Method	Auxiliary sentence	Output
QA	(aspect category) が含まれていますか (Does it contain [aspect category])	Yes/No
NLI	(aspect category)	Yes/No

3.2.2. Aspect Sentiment Classification (ASC) subtask

The auxiliary sentence generation methods in the ASC subtask are shown in **Table 2**. Each of the auxiliary sentence generation methods is described below.

- QA-M: M stands for Multi, meaning that the classifications are not binary but multiple. This method generates auxiliary sentences in the form of questions, and then let the system to answer those questions. Specifically, the method is to generate auxiliary sentences by combining an aspect, i.e., the object whose sentiment polarity is to be judged, with the sentence "についてどう思いますか" (What do you think of [aspect]?)
- QA-M-Improve: Our improved auxiliary sentence construction method based on QA-M. The original QA-M auxiliary sentence construction method of Sun et al. (2019) sets the question to "What do you think of" (についてどう思いますか), but we believe that there is a more appropriate question sentence, and we assume that "に対する感情極性は何ですか" (What is the sentiment polarity towards) is a direct question about the sentiment polarity, and therefore, the BERT model can be guided to obtain more correct results and higher accuracy can be achieved.
- NLI-M: NLI means Natural Language Inference, and the meaning of M is same as QA-M. In this method, auxiliary sentences are generated using only the aspect.
- QA-B: Here, B stands for Binary, meaning that the class to be classified is binary ("Yes" and "No"). The method generates three auxiliary sentences by combining the aspect with three phrases, "に対する感情極性は肯定的" (the polarity of [aspect] is positive), "に対する感情極性は否定的" (the polarity of [aspect] is negative), and "に対する感情極性は中立的" (the polarity of [aspect] is neutral) respectively. When validating with test data, the sentence with the highest probability of "Yes" among the three sentences is selected, and the polarity indicated by that sentence is the polarity determined by the model. When expressing sentiment polarity, there are two types of expressions in Japanese: Kanji and Katakana. Therefore, in order to verify whether the Kanji expression: "肯定的" (positive), "否定的" (negative), "中立的" (neu-

[1] https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking

tral) or the Katakana expression: “ポジティブ” (positive), “ネガティブ” (negative), “ニュートラル” (neutral) can construct auxiliary sentences that can achieve higher accuracy, we conducted an experiment by generating auxiliary sentences expressing sentiment polarity in each of them.

- NLI-B: This method generates three auxiliary sentences by combining the aspect and three phrases “肯定的” (positive), “否定的” (negative), and “中立的” (neutral) with a hyphen, respectively. Everything else is the same as QA-B.

Table 2: Method of auxiliary sentences generation in ASC subtask

Method	Auxiliary sentence	Output
QA-M	(aspect) についてどう思いますか (What do you think of [aspect])	Pos/Neg/Neu
QA-M-Improve	(aspect) に対する感情極性は何ですか (What is the sentiment polarity towards [aspect])	Pos/Neg/Neu
NLI-M	(aspect)	Pos/Neg/Neu
QA-B	・(aspect) に対する感情極性は肯定的/ポジティブだ (the polarity of [aspect] is positive) ・(aspect) に対する感情極性は否定的/ネガティブだ (the polarity of [aspect] is negative) ・(aspect) に対する感情極性は中立的/ニュートラルだ (the polarity of [aspect] is neutral)	Yes/No
NLI-B	・(aspect)-肯定的/ポジティブ ([aspect]-positive) ・(aspect)-否定的/ネガティブ ([aspect]-negative) ・(aspect)-中立的/ニュートラル ([aspect]-neutral)	Yes/No

3.3. BERT-pair model

After generating auxiliary sentences, for each target sentence, we combine the target sentence and the token “[SEP]” and the auxiliary sentence as a new sentence, and use the new sentences to fine-tune the BERT. The resulting BERT is then named with the name “BERT-pair” combined with the names of the respective auxiliary sentence generation methods.

4. Evaluation

This section describes the dataset, hyperparameters, evaluation index, and experimental results for the experiment in the AE subtask and the experiment in the ASC subtask.

4.1. Dataset

In this study, we use a dataset called chABSA[2]. chABSA dataset is a dataset based on securities reports of listed companies (fiscal year 2016), and contains the sentiment polarity information of “Positive”, “Negative”, and “Neutral” for each aspect included in a sentence. Another ABSA dataset for Japanese is Rakuten Travel: Review Aspect and Sentiment Tagged Corpus (Nakayama et al., 2022) by Rakuten, but it is not used in this study because it is not publicly available for free.

Similar to Miura et al. (2020)’s study, this study deals with 15 aspect categories in this dataset, consisting of combinations of three entities (company, business, product) and five attribute (sales, profit, amount, price, cost) (e.g., company#sales, business#amount). However, since the number of aspects belonging to the aspect category “company#price” is zero, the actual number of aspect categories is 14, and the total amount of sentences is 1,077, the total number of aspects is 2,079.

4.1.1. Aspect Extraction (AE) subtask

Since the proposed method expands each sentence to 14 sentences depending on the number of aspect categories handled, the total number of data in the actual data set is 15,078 (14 times 1,077). First, this dataset is divided into training, validation, and test data in a 7:1:2 ratio for experimentation. After adjusting the dropout rate to achieve the highest accuracy on the validation data, the validation data is put into the training data for experimentation. Thus, the final experiment divides this data set into 12,062 training data and 3,016 validation data at a ratio of 4:1.

4.1.2. Aspect Sentiment Classification (ASC) subtask

In the proposed auxiliary sentence generation methods QA-M, QA-M-Improve, and NLI-M, one auxiliary sentence is generated for each aspect, resulting in 2,079 pieces of data, which are divided into 1,663 training data and 416 test data in a 4:1 ratio for the experiments. In contrast, the proposed auxiliary sentence generation methods QA-B and NLI-B generate three auxiliary sentences for each aspect, resulting in 6,237 sentences in total, which are divided into 4,989 training data and 1,248 test data at a ratio of 4:1 for the experiments.

4.2. Hyperparameters

4.2.1. Aspect Extraction (AE) subtask

In the AE subtask experiments, the hyperparameters were set as shown in **Table 3**.

Here, the dropout rate was adjusted using the validation data. Since the highest F1-score was obtained in the validation data when the dropout rate was set to 0 for all models, we set the dropout rate to 0.

4.2.2. Aspect Sentiment Classification (ASC) subtask

In the experiments in the ASC subtask, the hyperparameters were set almost the same as in the AE subtask except

[2] https://github.com/chakki-works/chABSA-dataset

Table 3: Hyperparameters in AE subtask

Hyperparameter	Value
Learning rate	2e-5
Epoch	3
Batch size	16
Dropout rate	0
random_state (pandas.DataFrame.sample)	4,040
Random seed (Pytorch)	2,020

for two differences. First, we set the dropout rate to 0.1. Second, we set the batch to 16 for the M-type models, i.e. the models whose output is multi, and to 48 (three times 16) for the B-type models, i.e. the models whose output is binary. The reason for this is that the auxiliary sentences generation methods with "B" generate three sentences for each sentence and each aspect contained in that sentence, corresponding to the three sentiment polarities, and we want BERT to learn these three sentences as one coherent unit.

4.3. Evaluation index

As in the study by Miura et al. (2020), in the AE subtask, we use the F1-score, which is the harmonic mean index of the precision and recall, and in the ASC subtask, we use the accuracy as the evaluation index.

4.4. Experimental results

4.4.1. Aspect Extraction (AE) subtask

The experimental results in the AE subtask are shown in **Table 4**. "BERT-pair-NLI", fine-tuned with the auxiliary sentences generated by the NLI auxiliary sentence generation method, outperformed all the results achieved by the models of the existing methods and exceeded all the other proposed models, obtaining the highest precision, recall, and F1-score.

Table 4: Experimental results in the AE subtask. Underlined numbers indicate better results than in previous studies; bold numbers indicate the best results.

Model	Dropout	Precision	Recall	F1-Score
Miura et al. (2020)	0	0.7530	0.6531	0.6985
	0.2	0.7545	0.6667	0.7068
	0.5	0.7313	0.6737	0.7009
BERT-pair-QA	0	0.8094	0.7282	0.7666
BERT-pair-NLI	0	**0.8321**	**0.7379**	**0.7822**

4.4.2. Aspect Sentiment Classification (ASC) subtask

The experimental results in the ASC subtask are shown in **Table 5**. The "BERT-pair-QA-M-Improve" model, which we fine-tuned with auxiliary sentences generated by QA-M-Improve, an improved version of QA-M, outperformed the existing methods and achieved the highest accuracy among all methods.

The M-type models, i.e. the models whose output is multi, obtained a overall higher accuracy than the B-type models, i.e. the models whose output is binary. The models using Kanji expression obtained a higher accuracy than the models using Katakana expression.

Table 5: Experimental results in the ASC subtask. Katakana means expressing sentiment polarity in Katakana expression. Kanji means expressing sentiment polarity in Kanji expression. Underlined numbers indicate better results than in previous studies; bold numbers indicate the best results.

Model	Accuracy
Miura et al. (2020)	85.6
BERT-pair-QA-M	87.0
BERT-pair-QA-M-Improve	**88.7**
BERT-pair-NLI-M	87.5
BERT-pair-QA-B (Katakana)	82.2
BERT-pair-QA-B (Kanji)	82.9
BERT-pair-NLI-B (Katakana)	81.7
BERT-pair-NLI-B (Kanji)	82.0

5. Discussion

This section discusses the experimental results of the AE and ASC subtasks, respectively.

5.1. Aspect Extraction (AE) subtask

The experimental results confirm that the proposed method achieves better accuracy than the existing methods. The first reason is that the proposed method expands each sentence according to the number of aspect categories in the dataset (14 in this study), which expands the dataset, increases the number of sentences used for training, and increases the training data for the model. The second reason is that BERT has been pre-trained on the "next sentence prediction" task, which captures the sentence-sentence relationships better and thus performs better on sentence pair classification problems such as the QA and NLI tasks.

The reason why "BERT-pair-NLI" obtained higher F1-scores than "BERT-pair-QA" is still unknown and we will investigate this further in the future.

5.2. Aspect Sentiment Classification (ASC) subtask

The experimental results confirm that the proposed method achieves better accuracy than the existing methods. The reason for this is the same as in the AE subtask, and the only difference is that the proposed method expands each sentence according to the number of aspects in the dataset.

Furthermore, the results showed that the models using the Kanji expression ("肯定的", "否定的", and "中立的") achieved higher accuracy and higher F1-score than the models using the Katakana expression ("ポジティブ", "ネガティブ", and "ニュートラル") when expressing sentiment polarity in auxiliary sentences. The reason for this is that Japanese BERT was pre-trained on the Japanese version of Wikipedia, and in this corpus, the Kanji expression are more often used to express sentiment polarity than the Katakana expression, therefore the model captures the meaning better with Kanji expression than Katakana expression when expressing sentiment polarity.

The reason the "BERT-pair-QA-M-Improve" model

achieved better results than the "BERT-pair-QA-M" model is probably that the auxiliary sentence the former model used asked a more direct question about the sentiment polarity, and thus induced BERT to yield more correct results. These results confirm that higher accuracy can be obtained by constructing more appropriate auxiliary sentences.

6. Conclusion

In this study, we improved the method originally proposed for English language, which automatically generates auxiliary sentences and fine-tuned the BERT with sentence pairs that combine target sentences and auxiliary sentences, to the aspect category detection subtask, i.e., the AE subtask, and the ASC subtask of the Japanese ABSA task. Experimental results on the chABSA dataset showed that the F1-score in the AE subtask improved by 7.54 points over the previous study, and the accuracy in the ASC subtask improved by 3.1 points over the previous study, confirming the effectiveness of the proposed method. Furthermore, in the experiment on the ASC subtask, the result that the accuracy of the "BERT-pair-QA-M-Improve" model is 1.7 points higher than the "BERT-pair-QA-M" model indicates that higher accuracy can be achieved if more appropriate auxiliary sentences are constructed. In the future, we plan to search for better methods to further improve the accuracy of the Japanese ABSA task.

References

Akai, Ryuichi and Masayasu Atsumi, 2019. Application of aspect-based sentiment analysis using self-attention mechanism to japanese sentences. In *The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019*.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*.

Hashimoto, Chikara, Sadao Kurohashi, Daisuke Kawahara, Keiji Shinzato, and Masaaki Nagata, 2011. Construction of a blog corpus with syntactic, anaphoric, and sentiment annotations. *Journal of Natural Language Processing*, 18(2):175–201.

Kiritchenko, Svetlana, Xiaodan Zhu, Colin Cherry, and Saif Mohammad, 2014. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th international workshop on SemEval*.

Liu, Fei, Trevor Cohn, and Timothy Baldwin, 2018. Recurrent entity networks with delayed memory update for targeted aspect-based sentiment analysis. In *Proceedings of NAACL-HLT*.

Ma, Yukun, Haiyun Peng, and Erik Cambria, 2018. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Miura, Yoshihide, Ryuichi Akai, and Masayasu Atsumi, 2020. Self-attention neural network for sentiment analysis of multiple aspects in sentences. In *The 34th Annual Conference of the Japanese Society for Artificial Intelligence, 2020*.

Nakayama, Yuki, Koji Murakami, Gautam Kumar, Sudha Bhingardive, and Ikuko Hardaway, 2022. A large-scale japanese dataset for aspect-based sentiment analysis. In *Proceedings of the 13th LREC*.

Nguyen, Thien Hai and Kiyoaki Shirai, 2015. Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis. In *Proceedings of the 2015 EMNLP*.

Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer, 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*.

Radford, Alec, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al., 2018. Improving language understanding by generative pre-training.

Sun, Chi, Luyao Huang, and Xipeng Qiu, 2019. Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. In *Proceedings of NAACL-HLT*.

Tang, Duyu, Bing Qin, and Ting Liu, 2016. Aspect level sentiment classification with deep memory network. In *Proceedings of the 2016 EMNLP*.

Wagner, Joachim, Piyush Arora, Santiago Cortes, Utsab Barman, Dasha Bogdanova, Jennifer Foster, and Lamia Tounsi, 2014. Dcu: Aspect-based polarity classification for semeval task 4. In *SemEval@ COLING*.

Wang, Bo, Maria Liakata, Arkaitz Zubiaga, and Rob Procter, 2017. Tdparse: Multi-target-specific sentiment recognition on twitter. In *Proceedings of the 15th EACL: Volume 1, Long Papers*.

Wang, Yequan, Minlie Huang, Xiaoyan Zhu, and Li Zhao, 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 EMNLP*.

Zhang, Yiyang, Rafal Rzepka, and Kenji Araki, 2022. Using bert and auxiliary sentences generation to improve accuracy of japanese aspect-based sentiment analysis. In *70th Special Interest Group on Language Sense processing Engineering (SIG-LSE)*. The Japanese Society for Artificial Intelligence.

Zhang, Yiyang, Masashi Takeshita, Rafal Rzepka, and Kenji Araki, 2023. Utilizing bert with auxiliary sentences generation to improve accuracy of aspect category detection in japanese aspect-based sentiment analysis. In *NLP2023*. The Association for Natural Language Processing.

Large Language Models and the future of the Localization Industry

Andrzej Zydroń[1], Rafał Jaworski[2], Szymon Kaczmarek[3]

[1]CIO XTM International
azydron@xtm.cloud
[2]Head of AI NLP, XTM International
rjaworski@xtm.cloud
[3]AI NLP expert, XTM International
skaczmarek@xtm.cloud

Abstract

Transformer based large language models (LLMs), such as Google PaLM and LaMDA, Microsoft MT-NLG, Meta OPT-175B and OpenAI GPT-3.5 have the potential to revolutionize the Localization Industry by providing more accurate and efficient human quality translation. These models, which are trained on vast amounts of data, are able to capture the nuances and complexities of language, allowing them to produce translations that are more natural and accurate than those produced by Neural Machine Translation (NMT). LLMs also have significant additional advantages over NMT systems, mainly that it is significantly faster to produce custom models based on prior examples (e.g. translation memory), can cope significantly better with very long sentences and allow for translation between typically 150+ language pairs. The ability to produce human or near human quality translation will mean a significant shift towards machine translation (MT) post-editing as the main role for translators in the future. All of this signifies a big change in how translation workflows will look like in future. It also has profound repercussions for existing NMT providers are the barriers for entry are significant and available only to very large enterprises who have the resource to train and run LLM systems

Keywords: HLT4BM, Large Language Model, LLM, Machine Translation, MT, Neural Machine Translation

1. Introduction

Neural Machine Translation (NMT) burst onto the language technology scene in 2017 with a substantial improvement in translation quality over previous statistical machine translation (SMT) efforts. NMT showed that given enough data it is possible to build an effective Machine Translation (MT) system. Nevertheless there were drawbacks/weaknesses with NMT systems: training could take weeks/months depending on the amount of data being processed. In addition NMT systems were not very good at handling long sentences, nor at coping with words that had not been encountered during training, so called out of vocabulary words (OVW).

2. LSTM

The breakthrough algorithm that NMT systems are based on is LSTM (Long Short Term Memory). LSTM based systems are not capable of multi-threading which was an additional drawback leading to long training times. The Long Short-Term Memory (LSTM) algorithm is a type of recurrent neural network (RNN). It is a type of artificial neural network that is designed to remember long-term dependencies in data. The LSTM algorithm is used to learn the context of a sentence and to generate a translation that is more accurate than traditional statistical machine translation (SMT) systems. The LSTM algorithm is able to learn the context of a sentence by using a memory cell that stores information from previous words in the sentence. This allows the NMT system to generate a translation that is more accurate and natural-sounding than a translation generated by an SMT system.

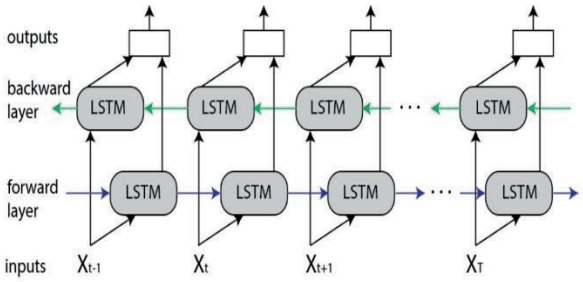


Fig. 1: LSTM Algorithm illustrated source

2.1. LSTM drawbacks

The main weaknesses of the LSTM algorithm used by NMT systems are its inability to capture long-term dependencies, its difficulty in dealing with rare words, and its lack of interpretability. Long-term dependencies are important for understanding the context of a sentence, but LSTM networks struggle to capture them due to their limited memory. Rare words are also difficult for LSTM networks to process, as they lack the data to accurately represent them. Finally, LSTMs due to their sequential architecture take a really long time to train and they are prone to overfitting.

2.2. NMT system issues

NMT systems have several production issues. In addition to the drawbacks inherent with the LSTM algorithm, such as

Contents

difficulty in coping with long sentences, very long training times, inability to multi-thread training, inability to cope with OVWs. It is impossible to retrain an NMT engine on the fly with new input. In addition LSTM training is not amenable to multi-threading which is one of the reasons for long training times.

3. Large Language Models

The advent of Large Language Models started around the same time as NMT systems but were designed initially for a different purpose: the initial idea behind LLM systems was to create a language model that could understand natural language and be used for a variety of tasks, such as question answering, sentiment analysis, and text classification.

3.1 Transformers

The key to this was the discovery of a new deep learning model called Transformer used by LLM systems to process natural language. It is based on a neural network architecture that uses attention mechanisms to learn the relationships between words in a sentence. The Transformer is capable of understanding the context of a sentence and can be used to generate text, perform machine translation, and answer questions. It is a powerful tool for natural language processing and has been used in many applications.``

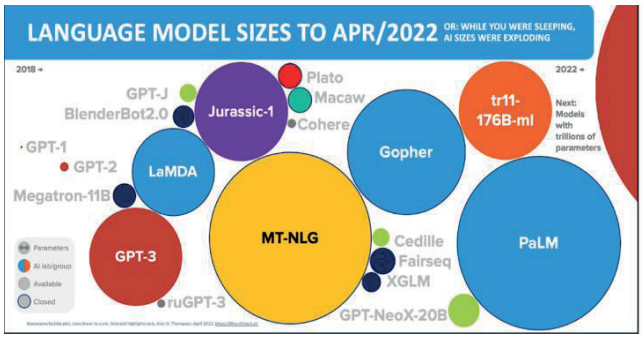
The Transformer model has many advantages over traditional methods of natural language processing. First, thanks to its self-attention mechanism it is able to capture long-term dependencies in language, allowing it to better understand the context of a sentence. Second, thanks to its parallel architecture it is able to process large amounts of data quickly and accurately, making it ideal for applications such as machine translation. Transformer training is also very amenable to multi-threading which speeds up model training times. Finally, they can be easily fine-tuned to perform many language related tasks. This makes it a powerful tool for LLM systems, as it can quickly and accurately process natural language data.

This work initially resulted in the creation of the Transformer encoder-decoder model by Google Research (Vaswani et al.) which was designed for machine translation and achieved at the time state-of-the-art results. A year later researchers from Google AI Language introduced ERT - Bidirectional Encoder Representations from Transformers (Devlin et al.). It was designed to be a deep learning-based language model that could be used to pre-train a model on a large corpus of text, allowing it to learn the language structure and then be fine-tuned for specific tasks. BERT is a deep learning model that uses an encoder-only transformer architecture to learn the context of a given text. It has been tested on a variety of natural language understanding tasks (Wang et al.) and achieved state-of-the-art results. This has enabled the development of more accurate and efficient language models, which have been used in a variety of applications such as natural language processing, machine translation, and question answering. Researchers from FacebookAI introduced RoBERTa - which was trained for longer and on more data. Batch size and BPE vocabulary was also increased. (Liu et al.). RoBERTa outperformed BERT in all individual tasks on the General Language Understanding Evaluation (GLUE) benchmark. The BERT model had a problem of the continuously growing size of the pretrained language models which resulted in longer training times and substantial memory usage. In 2019 Google Research introduced *A Lite BERT* (ALBERT) (Lan et al.) architecture. The base model of ALBERT is only slightly worse than BERT having 9 times fewer parameters. The biggest ALBERT model outperformed BERT with 5 times fewer parameters. In 2019 Generalized Autoregressive Pre Training for Language Understanding (XLNet) (Yang et al.) was introduced that combines the bidirectional capability of BERT with the autoregressive technology of Transformer-XL (Yang et al.). In 2020 Microsoft Research proposed Decoding-enhanced BERT with Disentangled Attention, (DeBERTa) and surpassed RoBERTA in GLUE benchmark.

Decoderonly transformer models have gained significant attention in recent years for their ability to generate high-quality text. These models are trained to only focus on the decoding phase of the transformer architecture, which allows them to generate text with a high level of coherence and fluency.

OpenAI demonstrated that an unsupervised pre-trained and then supervised fine-tined model can perform very well on natural language understanding tasks (Radford et al.). Later newer versions of this model were presented. GPT-2 with 1542M parameters (Radford et al.) and significantly larger GPT-3 with 175B parameters. The latter achieved promising results on NLP tasks while not being fine-tuned on any specific task.

In 2022 yet again Google Research introduced a new model: Scaling Language Modeling with Pathways (PaLM) (Chowdhery et al.) with over 540B parameters.



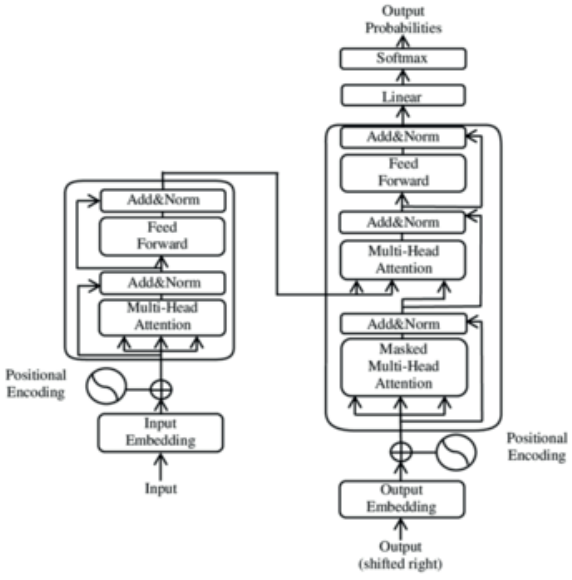


Fig. 2: LLM State Apr/2022

3.2 Training data

LLM systems inherently need very large volumes of data for training. OpenAI's latest GPT-3 model is trained on a massive dataset of over 45TB of text data. This data includes a wide variety of sources, such as books, articles, webpages, and social media posts. The data is also diverse in terms of topics, ranging from news and politics to science and technology. GPT-3 is also trained on a variety of tasks, such as language translation, question answering, and summarization. This allows GPT-3 to learn from a wide range of sources and tasks, giving it a better understanding of language and how to use it. The results are particularly impressive. GPT-3 can produce programming code if required in a variety of languages in addition to translation between some 150+ languages.

On 30th November 2022 OpenAI released ChatGPT along with GPT-3.5. The latest improvements have included semi-supervised training on the original GPT-3 LLM itself which has resulted in further improvements to the quality of the output. ChatGPT can maintain a full 'session' conversation.

3.3 LLM development trajectory

As evidenced recently with the release of GPT-3.5 as part of ChatGPT on 30th November 2022, LLM development is still very dynamic with constantly improving quality and without any sign of plateauing:

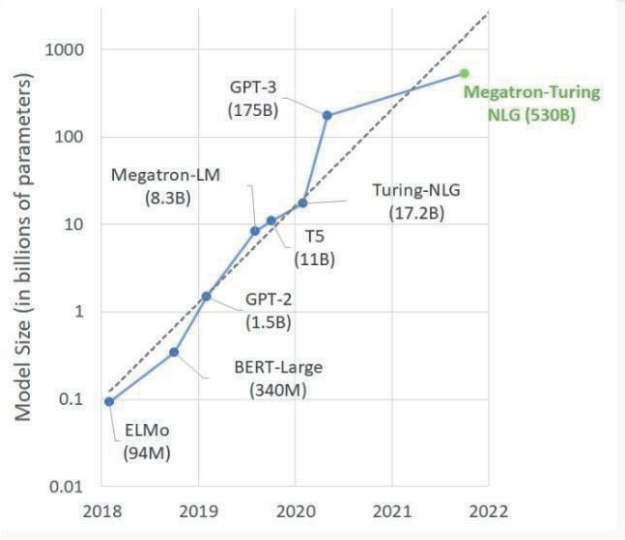


Fig. 4. LLM improvement trajectory

3.3 LLM and Translation

Of particular note is the ability of LLM systems such as GPT-3.5 to translate between 150+ human languages. The results are very impressive, equaling human translation quality in most cases that we have experimented with so far. Whereas NMT systems work on language pairs LLM systems do not have this limitation and can translate between any of the available languages that they have been trained on. It is also relatively easy to create custom models based on new training data. LLMs have therefore removed some of the key limitations of NMT systems, namely the ability to cope with long sentences, provide relatively fast and easy feedback to produce new custom models and the ability to remove limitations relating to OVW instances.

The drawbacks of LLMs is that they require a vast amount of computer resources to train (between $10 - $14 million) and a vast computer farm to run. This means that only very large enterprises such as OpenAI, Amazon, Google, Meta or IBM can train and offer such models as a service. So far only OpenAI has provided easy and transparent access to the general public to their system.

Initial tests run on a standard set of 2000 segments of technical literature relating to networking control software on the following language pairs: EN-DE, EN-SV, EN-ES, EN-FR, EN-PL, EN-EL, EN-JA show near human quality, but with the ability up quickly and constantly update the language models with supervised or control data feedback.

4. LLMs and the Localization Industry

TheWhat does this mean for the future of the Localization Industry? We can see a direct transition from NMT to further and more extensive adoption of MT. The improvement in quality, the constantly improving output, the ease of creating custom models and the ability to cope with very long sentences all remove previous barriers to wider adoption of NMT. Human translation is rarely perfect 100% of the time. Human translators make errors, both factual and grammatical when translating so LLM based MT offers the capability to improve translation quality in general.

The main issue is then around MT post-editing and how to assess the quality for each translated segment and where to direct the attention of post-editors. Fortunately MT engines are able to provide an assessment score of their own translation: the degree to which the software itself is confident of its own quality. This is due to the fact that for MT systems in general the software always attempts to create fluent output even if it is not sure of the accuracy of the translation – MT software tends to suffer from the Dunning-Kruger effect.

In our opinion LLM based MT will become the dominant starting norm for most translation projects within the next 2 – 5 years with the main emphasis on post-editing driven by cognitive software that highlights segments that require review. The need for highlighting is very important as it is very easy to miss MT errors when overall quality seems to be very high. There is a general 80/20 rule for human beings in general that if something appears to work 80% of the time then attentiveness drops off sharply resulting in true errors not being noticed.

It is our view that LLM based MT should lead to an overall reduction of around 20% – 50% in the overall cost of translation.

References

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). *Attention is all you need*. Google Research.

Gers, Felix. (2001). *Long Short-Term Memory in Recurrent Neural Networks* PhD Thesis. http://www.felixgers.de/papers/phd.pdf

Thompson, Alan D. 2023 *LLM Models current state.* https://lifearchitect.ai/models/

Dunning, David and Kruger, Justin, (1999*), Unskilled and Unaware of It: How Difficulties in Recognizing One's Own Incompetence Lead to Inflated Self-Assessments* , Journal of Personality and Social Psychology.

Brown, Tom B., et al. “Language Models are Few-Shot Learners.” *arXiv:2005.14165 [cs.CL]*, 2020. https://doi.org/10.48550/arXiv.2005.14165.

Devlin, Jacob, et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” *arXiv:1810.04805*, 2018. https://doi.org/10.48550/arXiv.1810.04805.

Lan, Zhenzhong, et al. “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations.” *arXiv:1909.11942 [cs.CL]*, 2019. https://doi.org/10.48550/arXiv.1909.11942.

Liu, Yinhan, et al. “RoBERTa: A Robustly Optimized BERT Pretraining Approach.” *arXiv:1907.11692 [cs.CL]*, 2019. https://doi.org/10.48550/arXiv.1907.11692.

Radford, Alec, et al. “Improving Language Understanding by Generative Pre-Training.” 2018.

Radford, Alec, et al. “Language Models are Unsupervised Multitask Learners.” 2019.

Vaswani, Ashish, et al. “Attention Is All You Need.” *arXiv:1706.03762*, 2017. https://doi.org/10.48550/arXiv.1706.03762.

Wang, Alex, et al. “GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding.” *arXiv:1804.07461 [cs.CL]*, 2018. https://doi.org/10.48550/arXiv.1906.05474.

Yang, Zhilin, et al. “Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context.” *arXiv:1901.02860 [cs.LG]*, 2019. https://doi.org/10.48550/arXiv.1901.02860.

Yang, Zhilin, et al. “XLNet: Generalized Autoregressive Pretraining for Language Understanding.” *arXiv:1906.08237 [cs.CL]*, 2019. https://doi.org/10.48550/arXiv.1906.08237

Author index

Dr. Patrick Paroubek is a senior CNRS Reseach Engineer with a 25 year long experience in Natural Language Processing. He holds a PhD in Computer Science from Université Pierre & Marie Curie (Paris 6) and an "Habilitation a Diriger les Recherches" from Université Paris-Sud. After working on programming languages during his PhD., he shifted progressively to the study of natural language following a post-doctorate internship at University Pennsylvania, in Prof. A.K. Joshi's laboratory. As a member of LIMSI-CNRS, he has been working in syntax, dialog systems, NLP evaluation (contributing to the organization of 15 evaluation campaigns), corpora, opinion mining and sentiment analysis, as well as scientometrics and scientific reporting quality. He has published over 100 scientific papers on these issues. He also contributed to several international initiatives related to the evaluation of NLP technologies, notably during the FLaReNet and Meta-NET European projects.

Prof. Dr. Zygmunt Vetulani, initiator and head of Department of Computer Linguistics and Artificial Intelligence (since 1.4.1993 to 31.12.2020), is a full professor in computer science (since 2006). He studied mathematics (1973, M.Sc.) and French philology (1982, M.A.) at Adam Mickiewicz University, Poznań (AMU). He received his PhD in mathematics from University of Warsaw (1977) and habilitation in computer linguistics from AMU (1990). He was Alexander von Humbold Foundation Fellow in 1987-1989 (Universität Bielefeld) and in 2018 (DFKI Saarbrücken). His main research activities during the last 30 years are in the field of natural language computer understanding and related topics. He served at several occasions the EC and Polish Government as human language technologies expert. He has published over 130 research contributions in computer science, mathematical logic and linguistics (papers, books, language resources and tools). He was the initiator (1995) and the main organizer of Language and Technology Conferences (until now). www.vetulani.home.amu.edu.pl

ISBN 978-83-232-4176-8 (Print)
ISBN 978-83-232-4177-5 (PDF)