



HAL
open science

Proceedings of the International Conference on Natural Language Processing, Signal and Speech Processing

Kamel Smaïli, Mohammed Diouri, Khalid Benzakour

► **To cite this version:**

Kamel Smaïli, Mohammed Diouri, Khalid Benzakour. Proceedings of the International Conference on Natural Language Processing, Signal and Speech Processing. 2017, 978-9954-99-758-1. hal-03349724

HAL Id: hal-03349724

<https://hal.science/hal-03349724v1>

Submitted on 22 Sep 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**INTERNATIONAL CONFERENCE ON NATURAL
LANGUAGE, SIGNAL AND SPEECH PROCESSING**

DECEMBER 05 and 06


**Casablanca 2017
Morocco**



Sponsor

ISGA

INSTITUT SUPERIEUR D'INGENIERIE & DES AFFAIRES

 **ISGA Campus Casablanca**
392, Road of EL Jadida, Casablanca, Morocco

Edited by Pr Kamel Smaïli, Dr Mohammed Diouri and Dr Khalid Benzakour

ICNLSSP 2017

International Conference on Natural Language Processing, Signal
and Speech Processing

Casablanca 5-6 December, 2017



Copyright 2017 International Science and General Applications (ISGA)

<http://icnlssp.isga.ma>

All rights reserved

Editors: Pr Kamel Smaïli, Dr Mohammed Diouri and Dr Khalid Benzakour

Legal deposit: 2017MO4568 / ISBN: 978-9954-99-758-1 / ISSN : 2351-8715

Presentation of ISGA

ISGA: Institut Supérieur d'Ingénierie et des Affaires is a private higher education group created in 1981. It is made up of five campuses spread over Morocco: Rabat, Casablanca, Marrakech, Fes and El Jadida. The campuses are postgraduate schools in Management and Engineering. ISGA has a research activity that is handled by the CRI.

The CRI (ISGA Research Center) deals with all the research and development of the group. The objective of the CRI is to:

- To give the conditions for the researchers and students to develop a good quality research.
- To encourage the researchers and the students to produce high level research articles.
- To participate to the organization of international conferences
- To fund the best students for a research stay of some months in international labs

The CRI is composed of the students enrolled in PhD program and permanent teachers and researchers. Thanks to the different collaborations, the CRI welcomes several researchers from different labs of:

- University Hassan II.
- CRAN (Centre de Recherche en Automatique de Nancy).
- LORIA (Laboratoire Lorrain de Recherche en Informatique et ses Applications).
- Doctoral school of UBS (Université de Bretagne Sud).
- Doctoral school of Senghor.

Twice a year, the CRI organizes the doctoral days, one for each research department. This is an opportunity for all the PhD students to present their work progress. Supervisors, industrial partners and renowned research guests participate to these events. A report of the thesis is presented and a schedule for the PhD defense is discussed with the most advanced students. More information concerning the CRI are provided at <http://www.isga.ma> ("Research & Development").

The CRI proposes an interdisciplinary journal (ISGA: International Science and General Applications <http://www.journal.isga.ma/>) whose primary objective is to fulfill the need for discussions of basic and applied research results. This journal meets the needs of professionals and researchers working in several scientific topics. Its aim is to bridge the gap between theoretical research and practical applications with potential real-world use. ISGA journal launches twice per year special issues since the topic of science is large. The editorial policy and the technical content of the Journal are the responsibility of the Editors. The next number will include the 10 best papers of ICNLSSP 2017 <http://www.icnlsp.ma>.

Welcome Message of the ICNLSSP 2017 chairs

On behalf of the Organizing Committee, we would like to welcome you to ICNLSSP 2017 in Casablanca, Morocco.

ISGA organizes several conferences related to different domains, this edition is dedicated to Natural language, Signal and Speech Processing (ICNLSSP).

The objective of ICNLSSP 2017 is to create a synergy between different areas related to language processing: Speech Recognition, Social Network, Opinion mining, Images, Videos,... The conference highlights new approaches related to the language from basic theories to applications. ICNLSSP is an international conference dedicated to Natural Language Processing and Speech recognition. It is a technical conference not only proposing new researches on the concerned topics but also permits exchanging ideas between researchers from the world that will be very useful for PhD students and developers in this area.

Four very well known keynote speakers will present their authoritative views on the state of the art of the topics of ICNLSSP. An expert of NLP in Arabic will present a tutorial on an unavoidable NLP platform for Arabic.

Concerning the origin of the submissions, 25% of the authors are from France, 20% are from USA, 15% are from Tunisia, 9% are from Morocco, 9% are from Algeria, 5% are from Netherlands and 17% are from other places.

The selection criteria were similar to those used for Large conferences. Between two and five reviewers, having strong experience, were assigned to each paper. The majority of the papers (71%) had 3 reviewers, 16% of the papers had 4 reviewers, 7% of the papers had 5 reviewers and 6% had 2 reviewers. Each paper has been weighted, the best one get a score of 35 and the last one -13. All the papers between 35 and 14 have been kept for publication and oral presentation.

Finally, we want to thank all of you who have sent your articles to the conference. We would like to thank also the committee program who has done a a good work on revising all the papers that helped us to have an excellent program. Our thanks also for Abdellah Lalaoui Hassani for the design of the website and the different exchanges with authors and for Hanane Laguesir for the design of the covers, badges, ...

See you in Casablanca.

Prof Kamel Smaïli

(PC chair)

Dr Mohammed Diouri

(Co-chair)

Dr Khalid Benzakour

(OC chair)

Program committee - chair Pr K. Smaïli

- Frédéric Béchet (Professor, University Aix-Marseille, France)
- Laurent Besacier (Professor, University of Grenoble, France)
- Khalid Choukri (Executive director of the European Language Resources Association (ELRA))
- Mona Diab (Associate Professor, George Washington University)
- Yannick Estève (Professor, University of Le Mans, France)
- Dominique Fohr (researcher, CNRS, France)
- Emmanuel Vincent (Researcher Inria Lorraine)
- Jean-Paul Haton (Professor emeritus, University Lorraine, France)
- Salma Jamoussi (Assistant Professor, University of Sfax, Tunisia)
- Denis Jovet (Researcher, INRIA Lorraine, France)
- David Langlois (Assistant Professor, Lorraine University, France)
- Chiraz Latiri (Professor, University of Tunis, Tunisia)
- Yves Lepage (Professor, University Waseda, Japan)
- Mikolaj Leszcuk (Assistant Professor, Poland)
- Khalifa Mansouri (Professor, University Hassan II)
- Odile Mella (Assistant professor, University of Lorraine)
- Franck Poirier (Professor, University of Bretagne sud)
- Fatiha Sadat (Associate Professor, UQAM, Canada)
- Karim Bouzoubaa (Professor, the Mohammed V University in Rabat, Morocco)
- Christophe Servan (Research Engineer chez Systran)
- Khaled Shaalan (Professor, The British University in Dubai, UAE)
- Olivier Siohan (Researcher, Google, USA)
- Yahya Slimani (Professor, University of Tunis, Tunisia)
- Kamel Smaili (Professor, University Lorraine, France)
- Juan-Manual Torrès (Assistant Professor, University of Avignon, France)
- Imed Zitouni (Researcher, Microsoft, USA)

Organization committee - chair Dr K. Benzakour

- Khalid BENZAKOUR (Director General, Director of Research and Development)
- Nabil CHERKAOUI (Educational Director)
- Imane CHLYAH (Responsible of studies)
- Dounia ENNAJY (Responsible of the information system)
- Said HARCHI (Technical director)
- Abdellah LALAOUI HASSANI (Director of the Information System)

- Hanane LAGUESIR (Digital Brand Manager)
- Siham RAIS (Assistant to the Director of Studies)

Subreviewers

We would like to thank the following people who accepted in a last minute to reviews articles for ICNLSSP 2017.

- Dr Salima Harrat (ESI - Algeria)
- Dr Karima Meftouh (University of Annaba - Algeria)

Keynotes and Tutorial

ISGA invited to ICNLSSP 2017 high-profile talks from USA, Japan, France and Morocco. The first day, two plenary talks are given by Professor Jean-Paul Haton (Institut Universitaire de France) and by Professor Yves Lepage (University of Waseda - Japan). The second day, Plenary talks are given by researchers Dr Olivier Siohan from Google (USA) and Dr Imed Zitouni (from Microsoft (USA)). At the end of the second day, a tutorial is given by Professor Karim Bouzoubaa from University of Mohammed V (Morocco).



Jean-Paul Haton
Professor emiritus

Biography

Jean-Paul Haton is Emeritus Professor in Computer Science at Université de Lorraine, Nancy, France. He is a senior member of the Institut Universitaire de France where he created the first chair in computer science. Jean-Paul Haton has been Director of the French National Project on Man-Machine Communication from 1981 to 1993, and Research Director at INRIA from 1988 to 1993. His research interests relate to Artificial Intelligence and Man-Machine Communication, especially in the fields of automatic speech recognition and understanding, speech training, signal interpretation, knowledge-based systems, and robotics. He has supervised or co-supervised more than 100 PhD theses in these fields. He authored or co-authored about 300 articles and books. Jean-Paul Haton is a Fellow of IEEE, a Fellow of International Pattern Recognition Society, IAPR and a Fellow of the European Artificial Intelligence Association, ECCAI. He served as chairman of AFIA (French Association for Artificial Intelligence) until 1994 and of ASTI, the French federation of associations for information processing. He is Vice-president of Académie Lorraine des sciences and he was awarded a Doctorate Honoris Causa from the University of Genève, Switzerland.

Keynote1: Artificial Intelligence: past, present and future

During the last decades, Artificial Intelligence (AI) has experienced impressive successes in various domains: games (chess, Go, poker...), robotics, perception (speech, vision), autonomous vehicles. The term AI is now frequently cited by the media, and it is known by the public. But despite these successes, AI is still far from reaching human intelligence in many domains. AI systems are based on two alternative approaches, based either on knowledge-based reasoning engines, or on biologically inspired neural networks. Besides, in several domains, including automatic speech recognition (ASR), and pattern recognition and image understanding, the role of statistical model is often of paramount importance. An example is the one of Hidden Markov Models that are present in all present ASR systems. Recently, approaches based on neural networks have been highly successful in almost domains of AI, under the form of

Deep Neural Networks (DNN). Hybrid models including neural networks have obtained the best results since a long time in ASR or character recognition, but these new DNN models are characterized by the fact that they have a large number of layers, much higher than previous networks, thanks to powerful deep learning algorithms and the availability of big data files. In this talk, we will present the different approaches and models of AI together with applicative examples in various areas, including ASR. The future of AI will also be explored



Yves Lepage
Professor

Biography

Professor Yves Lepage received his Ph.D. degree from GETA, Grenoble university, France. He worked for ATR labs, Japan, as an invited researcher and a senior researcher until 2006. He joined Waseda University, graduate school of Information, Production and Systems in April 2010. He is a member of the Information Processing Society of Japan, the Japanese Natural Language Processing Association, and the French Natural Language Processing Association, ATALA. He was editor-in-chief of the French journal on Natural Language Processing, TAL, from 2008 to 2016.

Keynote2: Automatic production of quasi-parallel corpora for machine translation

Abstract

This talk will address the problem of data scarcity in building machine translation systems in the data-oriented approach. It will show how to automatically produce quasi-parallel data from unrelated monolingual data to be added to a basic training corpus, so as to increase translation accuracy, as measured by BLEU, in statistical machine translation between Chinese and Japanese.

Index Terms: Machine translation, Quasi-parallel data

The problem: scarcity of bilingual aligned data

This talk will address the problem of data scarcity in building machine translation systems in the data-oriented approach. The work reported concerns statistical machine translation (SMT). The language pair addressed is Chinese—Japanese. Individually, Chinese and Japanese are relatively well-documented languages with efficient segmenters, morphological analysers, parsers, etc. However, the language pair itself suffers from a lack of bilingual corpora. Only recently have two large corpora been released in the domain of scientific and technological domain been released, but neither of them is freely accessible for download. The lack of data is thus an acute problem for this particular language pair.

Different possible solutions to augment the size of parallel corpora have been proposed in the past. They range from the manual creation of data to the automatic extraction of comparable corpora, with attempts at creating bilingual data from monolingual data [1, 2, 3]. In statistical machine translation, where the translation table is crucial, directly augmenting the data in the translation table has also been proposed [4]. All these methods may solve the problem of data scarcity to some extent and lead to slight increases in BLEU points improvement in different language pairs when used in addition to existing training data.

The method: automatic generation of quasi-parallel data from unrelated unaligned monolingual data

Here, we propose to follow a path which has been described as 'hallucinating' linguistic data [5]. It consists in creating synthetic data, parallel sentences, from unrelated unaligned monolingual data [6], not necessarily comparable corpora. The 'hallucinated' data are added to an existing basic training corpus to train an SMT system. We will show that this may lead to variable improvements, as measured by BLEU, ranging from less than a half point on difficult tasks, to several points in other tasks, depending on the experimental conditions [7].

The method is based on the use of a well-know operation: proportional analogy. The goal here is to produce a relatively important number of small variations in the training data. The variations are found in monolingual data and should thus be characteristic of each language [8]. We show that the introduction of these small variations increases the size of the translation tables and that the new phrases are actually used and contribute to translation accuracy [9].

The method consists in several steps. The first step is to produce a representation of the small variations by collecting analogical clusters independently in each language, from independently collected monolingual data. We will explain the method and introduce some tools to produce such analogical clusters, which have been publicly released. The second step is to align the small variations across the two languages of the language pair considered [10]. We adopt a relatively simple method which consists in computing a similarity coefficient between analogical clusters relying on a bilingual dictionary. The third and the fourth steps consist in producing variations in the basic training corpus and avoid over-generation due to analogy. This achieved by the use of filtering techniques [9].

Results and analysis: quasi-parallel rather than parallel, but grammaticality correct data

The talk will present various experiments conducted over several years in different settings. It will stress two important points that can be made on the method.

First, several experiments tend to show that the quality of the alignment of the produced sentences is not so crucial. What seems to be crucial is the grammaticality of the sentences produced. For that, different configurations and various methods have been tested so as to automatically ensure a very high level of grammaticality or semantic correctness. We will mention the different methods that may be used and present those that led to the best results, in particular the N-sequence filtering method.

Second, the relationship or rather the absence of relation between the basic training data and the monolingual data seems to be important. Monolingual data from the same domain or the same collection of texts do not seem to conduct to significant improvements. Thorough experiments still need to be conducted to confirm this impression, but it seems that variations unseen in the basic training data, i.e., rather from the general language, are necessary to obtain improvements in translation accuracy.

Acknowledgements

Part of the work reported here was supported by a JSPS Grant, Number 15K00317 (Kakenhi C), entitled Language productivity: efficient extraction of productive analogical clusters and their evaluation using statistical machine translation.

References

1. A. Klementiev, A. Irvine, C. Callison-Burch, and D. Yarowsky, "Toward statistical machine translation without parallel corpora," in Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, ser. EACL '12. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 130–140. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2380816.2380835>
2. J. Sun, Y. Qing, and Y. Lepage, "An iterative method to identify parallel sentences from non-parallel corpora," in Proceedings of the 6th Language & Technology Conference (LTC'13), Z. Vetulani, Ed. Poznań: Fundacja uniwersytetu im. Adama Mickiewicza, December 2013, pp. 238–242.
3. C. Chu, T. Nakazawa, and S. Kurohashi, "Chinese–Japanese parallel sentence extraction from quasi-comparable corpora," in Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL'13), Aug 2013, pp. 34–42.
4. J. Luo, A. Max, and Y. Lepage, "Using the productivity of language is rewarding for small data: Populating SMT phrase table by analogy," in Proceedings of the 6th Language & Technology Conference (LTC'13), Z. Vetulani, Ed. Poznań: Fundacja uniwersytetu im. Adama Mickiewicza, December 2013, pp. 147–151.
5. A. Irvine and C. Callison-Burch, "Hallucinating phrase translation for low resource MT," in Proceedings of the Eighteenth Conference on Computational Natural Language Learning. Ann Arbor, Michigan: Association for computational linguistics, June 2014, pp. 160–170.
6. W. Yang and Y. Lepage, "Inflating a training corpus for SMT by using unrelated unaligned monolingual data," in Advances in Natural Language Processing: Proceedings of the 9th conference on language processing (PolTAL 2014), A. Przepiórkowski and M. Ogrodniczuk, Eds., vol. LNAI 8686. Warsaw, Poland: Springer, September 2014, pp. 236–248.
7. H. Wang, W. Yang, and Y. Lepage, "Improved Chinese-Japanese phrase-based MT quality using extended quasi-parallel corpus," in Proceedings of the 2014 IEEE International Conference on Progress in Informatics and Computing (PIC 2014), Y. Wang, X. Li, and H. Cai, Eds. IEEE Computer Society Press, May 2014, pp. 6–10.
8. H. Wang and W. Yang and Y. Lepage "Sentence generation by analogy: towards the construction of a quasi-parallel corpus for Chinese-Japanese," in Proceedings of the 20th Annual Meeting of the Japanese Association for Natural Language Processing, Sapporo, March 2014, pp. 900–903.
9. W. Yang, H. Shen, and Y. Lepage, "Inflating a small parallel corpus into a large quasi-parallel corpus using monolingual data for Chinese–Japanese machine translation," Journal of Information Processing, vol. 25, pp. 88–99, 2017.
10. W. Yang, H. Wang, and Y. Lepage, "Deduction of translation relations between new short sentences in Chinese and Japanese using analogical associations," International Journal of Advanced Intelligence, vol. 6, no. 1, pp. 13–34, 2014.



Olivier Siohan
Research Scientist Google

Biography

Doctor Olivier Siohan is a Research Scientist working on speech and language technologies at Google Inc. His major research interests are speech and speaker recognition with a special focus on acoustic modeling and noise robustness with deep neural networks. From 2003 to 2007, he was a Research Staff Member at IBM T. J. Watson Research Center working on large vocabulary speech recognition. Prior to that, he was a Member of Technical Staff at Lucent Technologies - Bell Labs where he led a broadcast news speech recognition project. He also worked on speaker recognition at AT&T Labs from 1996 to 1998, and held a post-doc position at AT&T Bell Laboratories in 1996. He is currently serving his second term as a member of the IEEE Speech and Language Technical Committee and is a former associate editor of Pattern Recognition Letters.

Keynote3: Advances in Acoustic Modeling for Speech Recognition at Google

In the past decade, speech recognition technology underwent some profound changes driven by access to large computational resources combined with the availability of massive training sets for both acoustic and language modeling.

Statistical speech recognition, as defined in the mid 70's, treated the speech signal as the output of a generative stochastic process. This led to the development of acoustic models based on hidden Markov models (HMM) to represent the temporal variability of the speech signal, with Gaussian mixture densities (GMM) to represent the statistical distribution of speech features within an HMM state. For the next four decades, GMM-HMMs, which could be estimated in a fully data-driven manner from a transcribed audio corpus and easily adapted to new acoustic environments, became the dominant approach for acoustic modeling. Combined with N-gram language models estimated from large amount of text, HMMs have been the foundation of the first dictation systems as well as the first voice assistants such as Siri or Google Voice Search.

In the past several years however, acoustic models based on GMM-HMMs have been progressively replaced by deep neural networks (DNN). Rather than attempting to model the statistical distribution of the speech signal associated to HMM state, DNNs attempt to directly predict the HMM state associated to a short sequence of acoustic features. While a similar approach was developed in the mid 90's, it failed to outperform GMM-HMMs. This was mostly attributed to the compute power available at that time, which did not enable the use of deeper network with a large number of HMM states that made this technology successful. Following the success of DNNs, new classes of recurrent neural networks such as the long short term memory networks (LSTM) have also been developed to explicitly model the temporal variability of the speech signal, bringing additional performance improvements.

More recently, neural network based acoustic models have progressively evolved from being designed to predict the phonetic class associated with a short temporal slice of the acoustic

signal, to sequence-to-sequence models optimized to directly predict an entire sequence of words from the entire audio signal. This latest family of models brings significant changes in speech recognition technologies, in some cases alleviating the need for human-derived lexicon and text normalization grammars, in favor of a fully integrated end-to-end recognition system.

In this talk, we will describe those technological changes and illustrate the related performance improvements they brought on various speech products and services developed at Google, including some advances in far field recognition developed for Google Home and speech recognition for YouTube.



Imed Zitouni
Research Manager Microsoft

Biography

Imed Zitouni is a Principal Research Manager of the Conversation Understanding Sciences group at Microsoft AI+R, in charge of dialog systems and language understanding technology for the digital assistant Cortana. Prior to joining Microsoft in 2012, Imed was a Senior Researcher at IBM for almost a decade, where he led several Multilingual NLP projects, including Arabic NLP, informatics extraction, semantic role labeling, language modeling and machine translation. Prior to IBM, Imed was a researcher at Bell Labs, Lucent Technologies, for almost half dozen years working on speech recognition, language modeling and spoken dialog systems. Imed received his M.Sc. and Ph.D. from the University-of-Nancy¹ in France. He also obtained a MEng degree in computer science from ENSI in Tunisia. Imed is a senior member of IEEE, served as a member of the IEEE Speech and Language Processing Technical Committee, and is the associate editor of IEEE Trans. on Audio, Speech and Language Processing as well as TAL-LIP ACM journals. He is also the Information Officer of the ACL SIG on Semitic-Languages and served as chair as well as reviewing-committee-member of several conferences and journals. Imed is the author/co-author of two books as well as more than 100 patents and scientific papers. His research interest is in the area of Natural Language Processing, including dialog systems, language understanding and Information Retrieval.

Keynote4: Conversational Semantic Search: looking beyond web search, Q&A and dialog systems

Voice-controlled intelligent personal assistants, such as Cortana, Google Now, Siri and Alexa are increasingly becoming a part of users' daily lives, especially on mobile devices. They allow for a radical change in information access, not only in voice control and touch gestures but also in longer sessions and dialogues preserving context, necessitating to evaluate their effectiveness at the task or session level. The first part of this talk presents innovative approaches that evaluate different tasks in voice-activated intelligent assistants. These approaches use implicit feedback from users to predict whether they are satisfied with the intelligent assistant as well as its components, i.e., speech recognition and intent classification. Using these techniques, we can potentially evaluate and compare different tasks within and across intelligent assistants according to the predicted user satisfaction rates. This is characterized by an automatic scheme of categorizing user-system interaction into task-independent dialog actions, e.g., the user is commanding, selecting, or confirming an action. We use the action sequence in a session to predict user satisfaction and the quality of speech recognition and intent classification. The second part of the talk considers the new golden age that has opening-up for conversation systems that are at the heart of intelligent assistants. The recent influx of Deep Learning approaches that remove the burden of input featurization and dialogue-state design. Sequence-to-Sequence and Information Retrieval methods that make it easy to stand-up shallow chatbots that – despite their lack of understanding – are able to converse in convincingly natural language. Search En-

gines that are attempting to incorporate elements of dialogue state-tracking and deliver actions, e.g. deep links to Apps, to users. The confluence of these approaches offers the potential for exciting new ways of building and training conversation systems, repackaging statistical dialogue management in forms that non-experts can use, and thus deliver on some of the promises of Conversational AI.



Karim Bouzoubaa
Professor

Biography

Karim Bouzoubaa is a full professor of computer science in the Mohammadia School of Engineers at the Mohammed V University in Rabat. He has published two books and over 70 journal and conference papers on all aspects of intelligent systems and Arabic language processing. Karim Bouzoubaa holds a MSc and a PhD from Laval University in Canada in the artificial intelligence and multi-agent systems fields. He contributed in the release of Amine platform for the development of intelligent systems. Since 2006, he is focusing on the Arabic NLP field. His experience on that field has been expressed in many directions: heading a research group in his institution, reviewing papers, organizing local and international events, being invited as a speaker as well as a visiting professor, participating in the creation of a local Arabic NLP association as well as participating in local, regional and international projects.

Tutorial: Safar framework for Arabic NLP

This tutorial has two parts. The first one is a theoretical presentation of the Safar framework. In the context of NLP, we discuss the advantages of exploiting a framework when it comes to implement a NLP solution instead of relying from scratch on the use of a specific programming language. We also explain the differences between a toolkit, a platform and a framework using a software engineering perspective and present advantages and disadvantages of the available ones in the case of the Arabic language. Finally, we present the architecture of Safar and explain how it could easily be used and exploited to implement different Arabic NLP problems.

The second part of the tutorial is practical-oriented. Attendants are invited to bring their own computer and learn how to use Safar framework. At the beginning, we explain how non-programming users (such as linguists) can use the framework via dedicated web applications. For programming users, we start with basic levels and needs of an NLP application calling morphological and syntactic functions. Then, we demonstrate how to call utilities functions (such as removing stop words) and resource-oriented functions. Finally, we show how to build pipelines, taking the most known examples such as preprocessing algorithms for ML algorithms, etc.

Employing Context-Independent GMMs to Flat Start Context-Dependent CTC Acoustic Models

Mohamed Elfeky, Parisa Haghani, SeungJi Lee, Eugene Weinstein, Pedro Moreno

Google Inc., USA

{mgelfeky, parisah, leesj, weinstein, pedro}@google.com

Abstract

Context-Dependent (CD) CTC acoustic models have been shown to outperform Context-Independent (CI) CTC models. Although a temporal alignment of speech signals is not essential to the core CTC training algorithm, a *bootstrapping* acoustic model is still needed for producing the CD phone inventory used as the output of the CTC model. Previous work has shown that forced alignments using an acoustic model trained to optimize conventional alignment-based criteria, e.g., a CD-GMM or a cross-entropy trained CD-DNN, can be used for this purpose. However, both types of models take several days to train, increasing the end-to-end training time of a CD-CTC model. More recently, it was shown that a bidirectional CI-CTC model, whose training takes less time than a CD-GMM or CD-DNN model, could be used as a bootstrapping model. In this paper, we investigate using a CI-GMM that takes only a few hours to train, to generate the target CD inventory. We show that the CD-CTC model bootstrapped using this technique performs at similar accuracy to the ones bootstrapped from computationally-more-expensive and slower-to-train models. Furthermore, our evaluation results show that a CD-CTC model bootstrapped using a CI-GMM’s alignments is significantly more accurate than its bidirectional CI-CTC alignment bootstrapped counterpart, and thus suggest that CI-GMM alignments are more accurate.

Index Terms: Flat start, CTC, LSTM RNN, GMM, acoustic modeling

1. Introduction

Deep neural network (DNN) architectures are currently at the heart of most modern algorithms for acoustic modeling in automatic speech recognition. Acoustic model training generally assumes the presence of labeled training data, that is speech recordings annotated with ground-truth transcripts. In the classical training scenario, training transcripts, after conversion into the phonetic units that are the outputs of the acoustic model, are aligned to the speech signal using an algorithm such as Viterbi forced alignment. Providing such an alignment enables the acoustic model, implemented as a multi-way classifier, to receive pairs of feature vectors and corresponding phonetic label as training examples.

In recent years, recurrent neural network (RNN) architectures such as long short term memory (LSTM) [1, 2] have been shown to perform better than conventional feed-forward neural networks. In the basic RNN approach, the network receives the sequence of feature vectors and attempts to label each feature vector with the correct phonetic label by using the information about the feature vector itself, as well as the recurrent “state” in the network’s memory. However, these networks have no concept of trying to learn the actual sequence of labels that must be

produced to generate a correct transcript.

Recently, connectionist temporal classification (CTC) [3] techniques have become adopted to remedy this limitation of standard RNNs. CTC models attempt to learn the sequence of labels that is required to produce the correct transcript, but do not attempt to model the label that should be given to a specific feature vector, nor do they attempt to output the labels in a way that is aligned temporally with the speech signal. Because of this characteristic of CTC training, this type of algorithm is notable in that an alignment of speech features to phonetic labels is no longer necessary in the core training algorithm of the model. However, there are several reasons why an alignment is still beneficial. First, while training transcripts are generally provided on the word level, obtaining a forced alignment allows for the correct phonetic transcript to be selected among multiple alternative valid pronunciations. Second, in the training of a context-dependent (CD) CTC acoustic model, it is still necessary to learn the CD phone inventory, e.g., by way of a tree-clustering algorithm which needs alignments. And third, using alignment information during CTC model training allows for the training to be constrained in a way such that phonetic labels are generated within a certain time window of the feature vectors corresponding to those labels [4].

Training alignments may be obtained by forced alignment using any acoustic model previously trained to optimize conventional alignment-based criteria, e.g., Gaussian mixture model (GMM), as well as RNN models such as LSTM and CLDNN [5] trained without using the CTC algorithm. We refer to this process of training a model from an existing alignments as *bootstrapping*. Model recognition accuracy (such as word error rates) is typically used to judge whether a bootstrap model will produce good alignments. This is partly due to the absence of universally accepted and easy-to-compute measures of alignment quality, as well as the absence of large current speech corpora annotated with ground-truth phonetic segment labels. However, the authors are not aware of published work showing conclusively that recognition quality and forced alignment quality are positively correlated. In fact there is some evidence to the contrary [6]. Additionally, training a sophisticated context-dependent acoustic model that can achieve good recognition quality from scratch is a complex and time- and resource-consuming multi-step process. So it is appealing to consider techniques that do not require such complexity and resources. We refer to the process of building a model from scratch as *flat starting*. Flat starting is necessary in many settings, e.g., training a new system for the first time for a new language.

In [7], the authors showed that a CI-CTC bidirectional LSTM (CI-CTC-BLSTM) model can effectively provide alignments for training a unidirectional CD-CTC-LSTM model that outperforms a previous best non-CTC CLDNN model. In this

paper, we propose a much simpler bootstrap/alignment technique based on a CI-GMM. Not only can it be trained quickly and without using large quantities of computational resources, but we will demonstrate that such a bootstrap model can be used to produce a final CTC model of superior quality.

The remainder of the paper is organized as follows. In Section 2, we will present a brief overview of the CTC training technique and GMMs. In Section 3, we outline our proposed technique of flat starting a CD-CTC model using a CI-GMM, describing the details of generating the target CD phone inventory in Section 3.1. We present and discuss evaluation results in Section 4 and finally conclude the paper in Section 5.

2. Background

2.1. Connectionist Temporal Classification

Connectionist Temporal Classification (CTC) [8] is a method for training recurrent neural networks, in which the model is trained on just a sequence of outputs and temporal alignment to the inputs is not required. Since the sequence of inputs (e.g., acoustic feature vectors) is usually longer than the sequence of outputs (e.g., phonemes) the network is permitted to output a special "blank" output. During training, the network is continuously integrating over all possible alignments and tries to optimize the total likelihood of all possible labelings of an input sequence with its target sequence.

However, lack of temporal alignment supervision can result in the recurrent model (which is capable of keeping state for use in the future) to arbitrarily delay outputting a non-blank label until it has observed a sufficiently large future context. In practice, as shown in [4], a CTC model delays outputs compared to a conventional model trained with temporal alignments. This characteristic is especially undesirable when using the acoustic model as part of a live streaming speech recognition service. In [4] the authors present a method for preventing such delays: during training, the set of search paths in the forward-backward algorithm is restricted to those that do not exceed a certain threshold between the CTC model output and the original ground truth alignments. Their reported experimental results show that this technique is effective in reducing the CTC model's output delay without causing regressions in the accuracy of the final sequence-trained CTC model.

It is well known that acoustic models predicting context dependent (CD) phonetic states result in more accurate speech recognition results compared to their context independent counter-parts. This is true for CTC models as well, and as such a bootstrapping model that can be used for generating the target CD inventory is needed. Earlier work [9] relied on the existence of a conventionally trained CD-DNN (or LSTM) for that. More recently the authors in [7] have shown that a bidirectional CI-CTC model can achieve the same goal. The intuition is that in bidirectional training, the model will try to delay outputs in both directions, thus the final delay during inference will not be in just one direction and the model will be able to generate sufficiently accurate time-alignments to be used during CD inventory generation. We will discuss the details of generating the CD phone inventory in each case in Section 3.1.

2.2. Gaussian Mixture Model

Gaussian mixture model (GMM) is a probabilistic model, defined as a weighted sum of Gaussian density functions. In speech recognition, GMM is used in conjunction to Hidden Markov Model (HMM) to model acoustic posterior probability,

$Pr(x|z)$, where z denotes a phonetic state and x is an acoustic feature vector. Then, $Pr(x|z)$ becomes

$$Pr(x|z) = \sum_{i=1}^M \lambda_i \mathcal{N}(x; \mu_i, \Sigma_i) \quad (1)$$

where \mathcal{N} is a Gaussian distribution with mean μ and covariance Σ , and M is the number of Gaussian components in the model.

When training a GMM the iterative Expectation Maximization (EM) algorithm can be repeated many times in phases and between each phase the complexity of the model or features can be changed. For example during the first phase, CI states are modeled, corresponding to a small M , which limits the number of parameters learned during this phase and allowing for faster training times. In follow-up phases, more complex features, for example discriminatively trained features such as linear discriminant analysis (LDA) features, may be used. Much larger CD states may replace the simple CI states and the number of mixture components (M) could be increased, all resulting in an increase in the model's accuracy and refined alignments.

3. CI-GMM to CD-CTC

Before our proposed flat starting technique, there were two options to bootstrap a delay-constrained CD-CTC model from scratch:

- Train a CD-DNN model from scratch, which takes more than 7 days including the time needed to build the CD phone inventory.
- Train a CI-CTC-BLSTM model, which takes 3 days to build.

Our proposed flat starting technique, outlined below, cuts the time to build such bootstrapping model for delay-constrained CD-CTC training to an average of *6 hours*. This is a significant reduction in this time-consuming multi-step process.

1. We train a CI-GMM that predicts 3-state HMM context independent phonemes. As the acoustic features for training this model, we use simple perceptual linear predictive features (PLPs) with deltas and delta-deltas. The standard Expectation Maximization (EM) algorithm is used for training this model.
2. We use the trained CI-GMM to force align the training data. From forced-alignments we generate the context-dependent (CD) phone inventory that will be used as the target outputs of the CTC model. Section 3.1 presents more details on this step.
3. We train a delay-constrained CD-CTC model that uses the alignments generated by the CI-GMM as ground-truth alignments for constraining CTC output delay. To achieve this, we compile the CD inventory model as a context dependency transducer that maps CD phones to CI phones. Using this transducer we map the forced-alignments to the newly generated CD inventory and use the technique introduced in [4] to constrain the CTC output delay.

3.1. Generating the Context Dependent Phone Models

As previously mentioned, it has been shown that unidirectional CD phone CTC models outperform unidirectional CI phone CTC models. Furthermore, since the underlying architecture

of CTC models is an LSTM, target CD phones instead of CD states are preferred [10]. Previous work [4] has shown how to generate the target CD inventory using a conventional 3-state CD inventory. For our proposed technique, we need to generate the target CD inventory using a 3-state CI-GMM. First, we force align the training data using the CI-GMM. This will produce 3-state CI alignments. We map all the 3 states of a CI alignment to the corresponding phoneme and collect acoustic features for that phoneme in its phonetic context (the usual left and right context). For each phoneme we then use the standard iterative decision tree building algorithm by Young et al. [11]. At each node of the tree a set of binary phonetic questions on the context are asked (for example whether the right context is a fricative). Each question divides the data into two disjoint sets and for each set a Gaussian model based on the corresponding features is computed. The question with the greatest likelihood gain is chosen for splitting and the process is continued until there are not enough observations at a node to divide it further.

As the acoustic features, we use the usual 40-dimensional log filterbank energy features. However, following the reasoning from [10] we take one representative feature vector from each segment of the alignment instead of all features. This is because in case of LSTMs we are interested in modeling the trajectories of the acoustic features and not their piecewise stationary periods. So for each segment of the alignment, we take the feature vector from the middle frame. Since we are using three state alignments, we concatenate three such feature vectors for each phoneme sample.

Figure 1 shows a schematic view of how the CD phone models are generated using different bootstrapping models. The vertical rectangles show the acoustic feature vectors (in our case 40-dimensional log filter banks). For simplicity, we have assumed that the 3-state CD-DNN model and the 3-state CI-GMMs generate similar alignments time-wise, but that is not true in experimentation. In all cases, only one feature vector is taken from each alignment segment. For 3-state models, the 3 segments of the same phoneme are grouped together and their corresponding feature vectors are concatenated. For the 1-state CI bidirectional CTC model, the generated alignments are very spiky, i.e., many frames are aligned to the “blank” target while phone outputs get aligned to only a few frames.

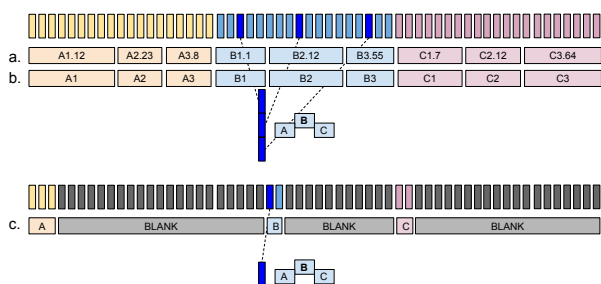


Figure 1: Phoneme sample from alignments generated by: a. 3-state CD-DNN model, b. 3-state CI-GMM, and c. 1-state CI bidirectional CTC model

4. Experimental Results

Our experiments were performed in five languages: Arabic, Farsi, Greek, Korean and Spanish. Each language has a training corpus of around 3M anonymized user utterances (approximately 2,000 hours), and a testing set of 25K anonymized and

manually transcribed utterances (approximately 12.5 hours). For each language, two test sets of similar size are available: one consisting of voice-search (VS) utterances, and the other of dictation (DT) utterances. For evaluation purposes, we measure word error rates (WERs) on the test sets. We use a five-LSTM-layers architecture with CTC training running for five days for each evaluated model.

To evaluate our proposed technique, we compare the performance (WER) of the CD-CTC model when bootstrapped using the proposed flat start CI-GMM versus two other bootstrapping models:

- An existing non-CTC trained CD-DNN model. This is considered the bootstrapping baseline to compare against.
- Flat start CI-CTC-BLSTM [7]. To the authors’ knowledge, this is the only known other flat starting technique for CD-CTC training.

It is important to reiterate that a CD-DNN model is not always available, as e.g., when training a model for a new language, and hence the need for flat starting. Of course, one can start from scratch and train that CD-DNN model to use for bootstrapping, but this is a complex and time- and resource-consuming process. In our experiments, we assume that these DNN models already exist for the five languages, and that they are the best performing non-CTC-trained DNN models.

The left subtable of Table 1 shows the results when using delay-constrained [4] CTC training. As expected, the baseline bootstrapping using an existing CD-DNN model outperforms both flat starting techniques. This is clearly due to the better training alignments generated from a previously well-trained model rather than from flat starting. However, when flat starting is inevitable, Table 1 shows that our proposed CI-GMM significantly outperforms CI-CTC-BLSTM by huge margins (ranging from 5.3% to 27.2% absolute points). Again, this is due to the better alignment we get from our CI-GMM technique, as we will illustrate later. Moreover, the inevitable quality losses from flat start CI-GMM bootstrapping is acceptable (ranging from 0.6% to 4.8% absolute points).

Our justification of the results in Table 1 has been about the training alignments produced by the bootstrapping model. To illustrate this further, Figure 2 visualizes the alignments of the same Arabic utterance as produced by the two flat starting techniques. Observe how our proposed CI-GMM aligned the utterance almost perfectly, whereas the CI-CTC-BLSTM is off. For example, the second word starts exactly at time 1.060s, and the CI-GMM aligned it perfectly, whereas the CI-CTC-BLSTM aligned it at time 0.87s. The same for the other two words.

To further investigate why the flat start CI-CTC-BLSTM bootstrapping performance is inferior to that of CI-GMM bootstrapping, we conducted two more experiments. First, we trained unconstrained CTC models using the same training alignments generated by CI-CTC-BLSTM for two of the five languages, which are shown in the right subtable of Table 1. Although the performance improved significantly over the constrained version (as expected [4]), it is still worse than the constrained version of our proposed flat start CI-GMM bootstrapping model. This implies that our proposed CI-GMM flat starting technique outperforms the CI-CTC-BLSTM technique even for unconstrained CTC training. Second, we trained CTC models without using the training alignments generated by the bootstrapping model, but rather let the CTC trainer optimize over all possible alignments [4]. In other words, the bootstrapping

Table 1: Absolute word error rates (WER) of CD-CTC models bootstrapped using different techniques

Test set		CD-DNN	Constrained CI-CTC-BLSTM	CI-GMM	Unconstrained CI-CTC-BLSTM
Arabic	DT	25.5	35.4	30.1	30.7
	VS	22.6	32.5	26.3	27.6
Farsi	DT	32.7	40.5	34.4	-
	VS	28.0	35.2	29.7	-
Greek	DT	19.9	30.4	20.6	28.5
	VS	25.0	35.1	25.6	32.8
Korean	DT	13.9	35.7	14.9	-
	VS	19.6	43.9	21.9	-
Spanish	DT	16.6	29.5	21.1	-
	VS	15.6	47.6	20.4	-

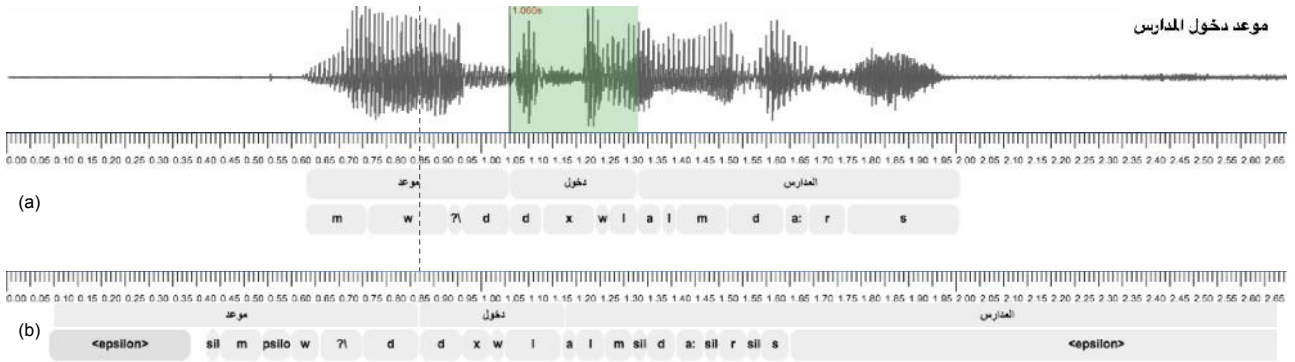


Figure 2: Alignments of an Arabic utterance produced by (a) CI-GMM and (b) CI-CTC-BLSTM flat starting techniques

model is only used to generate the CD phone inventory. The results for two of the languages are shown in Table 2. With respect to the CI-CTC-BLSTM bootstrapping, the performance has improved significantly over the constrained/unconstrained versions. The fact that ignoring alignments provided by CI-CTC-BLSTM bootstrapping yields substantially better final model quality serves as further evidence that these alignments are not of consistently good quality. Although in this experiment we don't see as big a quality difference as in Table 1, it is still generally the case that our proposed CI-GMM flat starting technique outperforms CI-CTC-BLSTM one. Nevertheless, the baseline bootstrapping using an existing CD-DNN model (Table 1) still outperforms both flat starting techniques.

Table 2: Absolute word error rates (WER) of CD-CTC models bootstrapped (CD phone inventory only) using different techniques

Test set		CI-CTC-BLSTM	CI-GMM
Greek	DT	24.2	25.6
	VS	28.1	27.7
Spanish	DT	20.9	17.0
	VS	20.9	15.8

Finally, let us compare the two flat starting techniques in terms of the time needed by each. For CI-CTC-BLSTM, we found that training it for less than 3 days will not produce a good enough model to be used for bootstrapping the CD-CTC model. Therefore, all the results above are obtained after training the CI-CTC-BLSTM model for 3 days. On the other hand, the average running time to obtain the CI-GMM was 6 hours.

This is undoubtedly a drastic reduction in end-to-end training time. Had we chosen to start from scratch and train a CD-DNN model, it would have taken 7-8 days to obtain the best model. Hence, a flat starting technique is preferable, and our proposed one is superior in time and performance.

5. Conclusion

In this paper, we investigated the possibility of using a CI-GMM for bootstrapping state-of-the-art CD-CTC acoustic models. CI-GMMs have the benefit of being very easy and fast (only a few hours) to train. This makes them a very attractive alternative to previous expensive bootstrapping models. Our extensive experiments are carried over several languages using both delay-constrained and unconstrained training, and without using the bootstrapped alignments at all. They show that the recognition accuracy of a CD-CTC model bootstrapped using our proposed technique is close to that of the one bootstrapped from a non-CTC trained model; and is significantly better than that of the one bootstrapped from a bidirectional CI-CTC model. Our results emphasize that time-accurate alignments are a necessary requirement of the bootstrapping model for the resulting CD-CTC model to produce accurate recognition results.

6. Acknowledgment

The authors would like to thank Olivier Siohan, Kanishka Rao and Haşim Sak for their valuable comments and discussions.

7. References

- [1] H. Sak, A. Senior, and F. Beaufays, “Long short-term memory recurrent neural network architectures for large scale acoustic modeling,” in *Interspeech*, 2014.
- [2] H. Sak, O. Vinyals, G. Heigold, A. Senior, E. McDermott, R. Monga, and M. Mao, “Sequence discriminative distributed training of long short-term memory recurrent neural networks,” in *Interspeech*, 2014.
- [3] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006.
- [4] A. Senior, H. Sak, F. de Chaumont Quitry, T. N. Sainath, and K. Rao, “Acoustic modelling with CD-CTC-SMBR LSTM RNNs,” in *ASRU*, 2015.
- [5] T. Sainath, O. Vinyals, A. Senior, and H. Sak, “Convolutional, long short-term memory, fully connected deep neural networks,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2015.
- [6] J. M. Kessens and H. Strik, “On automatic phonetic transcription quality: lower word error rates do not guarantee better transcriptions,” *Computer Speech & Language*, vol. 18, no. 2, pp. 123–141, 2004.
- [7] K. Rao, A. Senior, and H. Sak, “Flat start training of CD-CTC-SMBR LSTM RNN acoustic models,” in *ICASSP*, 2016.
- [8] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks,” in *Proceedings of the 23rd International Conference on Machine Learning*. ACM, 2006, pp. 369–376.
- [9] H. Sak, A. Senior, K. Rao, and F. Beaufays, “Fast and accurate recurrent neural network acoustic models for speech recognition,” in *INTERSPEECH*. ISCA, 2015, pp. 1468–1472.
- [10] A. Senior, H. Sak, and I. Shafran, “Context dependent phone models for LSTM RNN acoustic modelling,” in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, April 2015, pp. 4585–4589.
- [11] S. Young, J. Odell, and P. Woodland, “Tree-based state tying for high accuracy acoustic modelling,” in *Proc. ARPA Human Language Technology Workshop*, 1994.

About vocabulary adaptation for automatic speech recognition of video data

D. Jouvét^{1,2,3}, D. Langlois^{1,2}, M.A. Menacer¹, D. Fohr^{1,2,3}, O. Mella^{1,2,3}, K. Smaïli^{1,2}

¹ Université de Lorraine, LORIA, UMR 7503, Villers-lès-Nancy, F-54600, France

² CNRS, LORIA, UMR 7503, Villers-lès-Nancy, F-54600, France

³ Inria, Villers-lès-Nancy, F-54600, France

{ denis.jouvet, david.langlois, mohamed-amine.menacer, dominique.fohr,
odile.mella, kamel.smaili }@loria.fr

Abstract

This paper discusses the adaptation of vocabularies for automatic speech recognition. The context is the transcriptions of videos in French, English and Arabic. Baseline automatic speech recognition systems have been developed using available data. However, the available text data, including the GigaWord corpora from LDC, are getting quite old with respect to recent videos that are to be transcribed. The paper presents the collection of recent textual data from internet for updating the speech recognition vocabularies and training the language models, as well as the elaboration of development data sets necessary for the vocabulary selection process. The paper also compares the coverage of the training data collected from internet, and of the GigaWord data, with finite size vocabularies made of the most frequent words. Finally, the paper presents and discusses the amount of out-of-vocabulary word occurrences, before and after update of the vocabularies, for the three languages.

Index Terms: Speech recognition, vocabulary, vocabulary adaptation, vocabulary selection.

1. Introduction

The vocabulary is one of the key components of an automatic speech recognition (ASR) system. It needs to be adequate with respect to the considered speech recognition task, and this is usually achieved through a training or adaptation process. That is the object of this paper, which discusses the adaptation of vocabularies for automatic speech transcription of videos in French, English and Arabic, for AMIS (Access Multilingual Information opinionS) project¹. AMIS project aims at helping users to access information from videos that are in a foreign language, that is to understand the main ideas of the video. The best way to do that, is to summarize the video for having access to the essential information. Therefore, AMIS focuses on the most relevant information in videos by summarizing and translating it to the user. Obviously, the process starts by an automatic transcription of the audio channel.

Baseline ASR systems used at the beginning of the project have been developed from available corpora. For what concerns the linguistic part, that means that the vocabularies and the associated language models have been elaborated from quite old text data. Consequently, the vocabularies are somewhat outdated, and they are not relevant for a proper processing of person names and locations that have recently emerged in the news. Besides the fact that out-of-vocabulary (OOV) words affect speech recognition performance (in average, each out-of-vocabulary word produces 1.2 errors [1]), names of persons and of locations convey a very important and useful information for understanding the content of the videos. One way to cope with

this aspect is to collect large amounts of text data over the web, that correspond to about the same time period as that of the videos to be processed, and build new speech recognition vocabularies from this new text data.

Unknown words are also problematic in natural language processing, for example for syntactic parsing and for machine translation. Several papers have investigated the handling of unknown words [2], including the use of a probabilistic model for guessing base forms [3] in English and Finnish, and a morphological guesser for lemmatization in Arabic [4]. However, such approaches for dealing with written texts are not applicable to speech recognition.

With respect to speech recognition, several approaches have been developed in the past for elaborating vocabularies that are adequate for a given task. When a single text corpus is available, and when this corpus is homogeneous, the selection method is straightforward, it simply consists in selecting the most frequent words in the training corpus. However, since many years, the selection is done from numerous and heterogeneous corpora, which differ strongly in term of source or content (e.g., various radio or TV channels, journals, speech transcripts, ...), time period, and size (from a few million words up to more than several hundred million words). In such case, it is not suitable to concatenate all the text corpora and just select the most frequent words. A frequent word in a small corpus, thus interesting to select, may end up with a small frequency in the concatenated data, and would thus not be selected.

When dealing with a heterogeneous set of text corpora, various selection methods have been proposed that rely on the unigram distribution of the words in each sub-corpus. A conventional approach consists in finding the linear combination of the unigrams associated to each sub-corpus, that matches the best with the unigram distribution of some development set [5], and [6]; then the words having the largest unigram values (according to the combined unigram distribution) are selected. The combination parameters are obtained through an expectation-maximization process. Selection approaches based on neural networks have also been investigated [7]. It should be noted that all these techniques require the availability of a development set, representative of the task, for optimizing the unigram combination weights.

This paper investigates the selection of speech recognition vocabularies in French, English and Arabic, for the automatic transcription of videos in AMIS project. It is organized as follows. Section 2 presents the baseline speech recognition systems. Section 3 describes the collection of the textual data over internet. Section 4 presents an analysis of the collected data, with a comparison to the GigaWord data sets. Finally, Section 5 details the selection of speech recognition vocabularies and discusses some evaluation results.

¹ <http://deustotechlife.deusto.es/amis/project/>

2. Baseline ASR systems

The speech recognition systems are based on the KALDI speech recognition toolkit [8].

Acoustic modeling is based on Deep Neural Networks (DNN), as such modeling provides the best performance [9]. The DNN has an input layer of 440 neurons (11 frames of 40 coefficients each), 6 hidden layers of 2048 neurons each, and the output layer has about 4000 neurons, corresponding to the number of shared densities of the initial GMM-based speech recognition system. The classical n-gram approach is used for language modeling.

Table 1. *Some characteristics related to linguistic aspects of the baseline ASR systems.*

	French	English	Arabic
Text training data (number of word occurrences)	1,620 M	155 M	1,000 M
Vocabulary size (number of words)	97 k	150 k	95 k
Number of pronunciation variants per word.	2.1	1.1	5.1

Table 1 presents some characteristics of the baseline ASR systems, with respect to some linguistic aspects. Vocabulary sizes vary from about 100 k words (for French and Arabic) to 150 k words (for English). The average number of pronunciations variants vary from 1.1 for English, to 2.1 for French, and 5.1 for Arabic. In French, most of the pronunciation variants are due to the optional mute-e at the end of many words, and to possible liaison consonants with following words starting by a vowel. In Arabic, the larger number of pronunciation variants is due to the absence of diacritic marks, which indicate short vowels, in the spelling of the vocabulary words.

3. Web textual data

As the vocabularies in the baseline ASR systems have been defined according to available text corpora, that are rather old, the vocabularies are somewhat outdated, and they do not properly reflect the names of persons and locations observed in the recently collected videos of AMIS project. To update the vocabularies, new text data has been collected over the internet, in a period matching the period of the videos. This section also describes the elaboration of the test and development sets.

3.1. Training corpus

A few newspaper, radio and TV web sites in French, English and Arabic have been selected for collecting text data. A script was used to crawl web pages from the given sites over several months. The period over which text data was collected, was the same for the three languages.

A preprocessing has been applied on the raw text data collected from the various web sites. It mainly consists in removing useless data (e.g., date tags, hour tags, some keywords such as "view image", "download", ...), long non-Arabic text in Arabic web pages, ... Moreover, all duplicated sentences were also removed. About 80% of the amount of collected data is thus ignored. The amount of word occurrences available per language, after this preprocessing, is reported in

Table 2. Note that during this preprocessing, all capital letters have been kept.

Table 2. *Amount of word occurrences per language for the web training data, and for the GigaWord data.*

Language	Web data	GigaWord
French	1.9 G	0.8 G
English	2.9 G	4.1 G
Arabic	0.7 G	1.1 G

3.2. Test corpus

The videos processed in AMIS project have been collected from Youtube. They correspond to various channels such as Alarabiya, Alquds, BBC, EnnaharTV, Euronews, France24, RT, SkynewsArabia... For most of the videos, Youtube provides short descriptions which correspond to the content of the video, and thus contain names of persons and locations occurring in the videos. Such data has been collected for all AMIS videos (a few thousand videos per language). This data is used as a test data set for evaluating the percentage of occurrences of out-of-vocabulary words. Table 3 indicates the amount of word occurrences in the development sets, for each language. An example of YouTube description is available in the top part of Figure 1.

3.3. Development corpus

On Euronews web site, one can find descriptions of Euronews videos. Such descriptions generally provide detailed information on the content of the video, which in some cases, is rather similar to a transcription of its content.

Independently of the collection of videos for AMIS project, another set of about 8000 videos, in Arabic language, were collected from Euronews web site. Cross language links available in the Euronews descriptions make possible to collect also the descriptions in French and in English for those videos. This led to about 8000 text descriptions in French, in English and in Arabic. This data set, which is not associated to AMIS videos, but comes from a similar period was used as a development set for the selection of the new vocabularies.

Among the videos collected in AMIS project, a part of them corresponds to Euronews. Hence, you can find in the top part of Figure 1 an example of a Euronews description (long description), along with the YouTube description (which is much shorter, four lines only).

Table 3. *Amount of word occurrences per language in development and test sets.*

	French	English	Arabic
Development set	1500 k	1720 k	1240 k
Test set	250 k	280 k	70 k

4. Analysis of the collected web data

An analysis of the data was carried out. For each data set collected from internet (i.e., French, English and Arabic), the frequency of occurrences of the words has been analyzed. The same analysis was applied on the GigaWord corpora available from the LDC (French [10], English [11], and Arabic [12]).

Euronews fra 06CurVMod74 AVmedium(WebM)[1.37] Isral renonce - librer des prisonniers palestiniens
 hash: #occupiedterritories
 title: Israël renonce à libérer des prisonniers palestiniens
 YouTube keywords: euronews, world, Politique, Israël, Politique Palestine.

Les autorités palestiniennes déplorent la décision d'Israël de ne pas libérer un groupe de prisonniers. Cette décision a été annoncée ce jeudi par les dirigeants israéliens à l'issue d'une rencontre avec les négociateurs palestiniens.
 Cette libération devait être la quatrième du genre. Elle s'inscrivait dans le cadre du processus de paix.
 "Israël a toujours cherché à s'acquiescer de ses engagements dans le processus de paix, tout en se trouvant des excuses, estime Jihad el-Quawami, habitant d'Hé...

En fait, chacun renvoie la responsabilité sur l'autre. Et au final, c'est le processus de paix qui en pâtit.
 Les pourparlers ont été engagés l'été dernier pour une durée de 9 mois. Ils sont donc entrés dans leur dernier mois. Et rien n'indique qu'ils vont aboutir, au grand dam des Américains, gagnés par une certaine lassitude.
 Le secrétaire d'Etat John Kerry, en première ligne dans ce dossier, a rappelé aux Israéliens et aux Palestiniens que cet état euh à faire des compromis le chef de la diplomatie américaine s'est, malgré tout, dit optimiste sur une poursuite du dialogue.

ASR_V02

01:04 - S01: le secrétaire d'état de jeunes qui lui en première ligne dans

ASR_V02 * S01: Le secrétaire d'Etat John Kerry en première ligne dans

ASR_V01

01:04 - S01: le secrétaire d'état de jeunes qui lui en première ligne dans

01:09 - S01: Le secrétaire d'Etat John Kerry en première ligne dans ce dossier a rappelé aux Israéliens et aux Palestiniens que cet état euh à faire des compromis le chef de la diplomatie américaine s'est, malgré tout, dit optimiste sur une poursuite du dialogue.

Figure 1: Display of speech recognition results achieved with the old and new vocabularies. Both speech recognition results (ASR-V01 corresponding to the baseline ASR, and ASR-V02 corresponding to ASR with the updated vocabulary) are displayed as synchronized subtitles (bottom-left) and in separate frames (bottom-right). For helping checking recognition performance, when available, the YouTube and Euronews description are also displayed (middle part).

As a result, Figure 2 displays, for each corpus, the coverage of the word occurrences with respect to the most frequent word tokens of the corresponding corpus. For example, for English with data collected over internet the 100,000 most frequent words cover about 96.6% of the 2,880 million word occurrences of the English data; whereas for Arabic, the 100,000 most frequent words cover only 92.7% of the 690 million word occurrences of the Arabic data. Solid lines correspond to the data collected on internet. Dotted lines correspond to the GigaWord corpora.

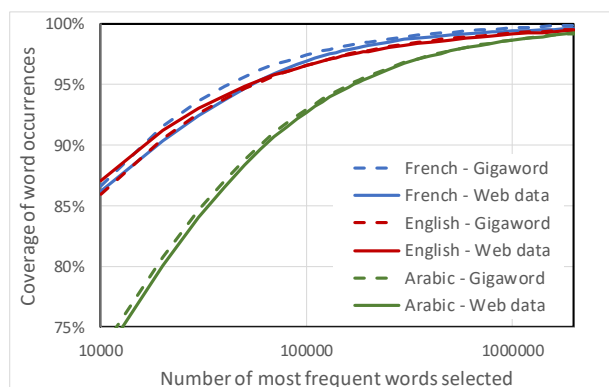


Figure 2: Coverage of text data with respect to the most frequent words (of each data set).

On the figure, the 4 curves corresponding to “French (gigaword corpus)”, “French (internet data)”, “English

(gigaword corpus)”, and “English (internet data)” are very similar. For each language, internet data and GigaWord data leads to very similar results. The figure also shows that to reach a given coverage, much more words are needed in Arabic than in French and English languages, probably due to the morphological richness of Arabic.

5. Updated vocabularies

The new text data collected over internet is used here as text material from which new ASR vocabularies are selected for transcribing AMIS videos. Results are then analyzed mainly in terms of percentage of out-of-vocabulary words in the test sets for the different languages and different vocabulary sizes.

5.1. Selection of vocabulary words

The selection process relies on the conventional approach. First a unigram is estimated on each subset of the training corpus, corresponding to a radio channel, a TV channel, a journal, etc. For example, for the French language, the various subsets correspond to Euronews, France 24, France Inter, Le Monde, Le Figaro, L’Humanité, and so on. Overall, there are about 30 subsets for the French language. A similar splitting, according to web site, is done also for English and Arabic data, leading to 22 subsets for English and 29 subsets for Arabic.

Once unigrams models are trained on each subset, they are linearly combined to make a global unigram. The weights of the linear combination are estimated with an Estimation-Maximization (EM) algorithm to match as best as possible the unigram estimated on the development data. The objective

function that the E.M. algorithm optimizes is the Kullback-Leibler distance between the unigram distribution corresponding to the linear combination of the unigrams estimated on each sub-corpus, and the unigram distribution estimated on the development corpus.

On the French data, the two largest combination weights and the associated sub-corpus are the following: 0.876 for Euronews, and 0.106 for France24. All the other weights are below 0.01. A similar behavior is observed for the other languages. The large weight obtained for the Euronews channel may be due to the fact that the development set is made of descriptions of Euronews videos.

Finally, the selected vocabulary corresponds to the words that have the largest probability in the combined unigram. For each language, four vocabularies have been extracted corresponding respectively to the 100 k, 200 k, 400 k and 800 k most probable words.

5.2. Analysis of results

The best analysis that could be carried out requires a manual transcription of a large subset of AMIS videos. As such transcription is not available for AMIS videos, we have used the text data corresponding to the Youtube descriptions as test sets. On the test sets, we evaluated the amount of out-of-vocabulary words for the various vocabularies: baseline ASR vocabulary, and new vocabularies (100 k, 200 k, 400 k, and 800 k words). Results for the 3 languages are reported in Table 5. For comparison purpose, Table 4 reports the percentages of out-of-vocabulary words on the development sets.

Table 4. *Percentage of out-of-vocabulary words in the development sets for each language and vocabulary. Sizes of baseline vocabularies are specified in Table 1.*

	French	English	Arabic
Nb. words	51 k	64 k	129 k
Nb. occurrences	1500 k	1720 k	1240 k
Baseline (95 to 150 k)	1.8%	7.2%	17.4%
New 100 k	0.4%	1.1%	5.5%
New 200 k	0.1%	0.4%	3.1%
New 400 k	0.1%	0.3%	1.5%
New 800 k	0.1%	0.3%	0.2%

Table 5. *Percentage of out-of-vocabulary words in the test sets for each language and vocabulary. . Sizes of baseline vocabularies are specified in Table 1.*

	French	English	Arabic
Nb. words	20 k	21 k	20 k
Nb. occurrences	250 k	280 k	70 k
Baseline (95 to 150 k)	1.8%	5.5%	16.4%
New 100 k	0.8%	3.3%	6.8%
New 200 k	0.4%	2.7%	4.5%
New 400 k	0.2%	1.9%	3.1%
New 800 k	0.2%	1.5%	2.0%

As can be seen on these tables, the percentage of out-of-vocabulary words is much lower with the new vocabularies than with the old ones. The same behavior is observed on the development and on the test sets. In all cases, increasing the size of the vocabularies significantly reduces the percentage of out-of-vocabulary words. For example, for the English data, on the test set, the OOV rate was reduced from 5.5% with the baseline vocabulary (150 k words) to 3.3% with the new 100 k word vocabulary, and then to 2.7%, 1.9% and 1.5% respectively with the 200 k, 400 k and 800 k vocabularies.

Comparing the languages, the OOV rates are smaller for the French data than for the English data. The largest OOV rates are observed for the Arabic language. The large OOV rate on Arabic data was also observed in other studies related to statistical modeling of Arabic [13] and [14].

To check the benefit of the new vocabularies, they have been used for a new transcription of AMIS videos. The two speech recognition results obtained with the old vocabulary, and with the new vocabulary (100 k words), are displayed as simultaneous subtitles. Figure 1 provides a typical example of the recovery of names of persons thanks to the new vocabularies. “John Kerry” was not present in the old French vocabulary, and thus the corresponding occurrence was replaced by a sequence of short words which are acoustically close. As the person name is missing in the old vocabulary, the corresponding transcription (cf. line ASR-V01 in Figure 1) gets difficult to understand, and an important information (the name “John Kerry”) is missing; such behavior will also impact the machine translation process. With the new vocabularies, this problem is overcome.

6. Conclusion

This paper has investigated the problem of out-of-vocabulary word in the transcription of videos in French, English and Arabic. A large part of out-of-vocabulary words concerns names of persons and locations, which convey an important information for understanding the content of videos. To elaborate speech recognition vocabularies that are adequate for the transcription of the videos, large amount of data has been collected over internet in a period matching the period of the videos. This data collected over internet has been compared to the well-known GigaWord corpora, available from LDC. The behavior (coverage) of the frequent words of each corpus, is similar between the data collected over the web and the GigaWord data. Nevertheless, the comparison shows that much more (frequent) words are needed in Arabic than in French or English to achieve a similar coverage of the word occurrences.

The collected data has been used to elaborate updated vocabularies in French, English and Arabic. Different sizes have been considered from 100 k words up to 800 k words. Noticeable reductions in the OOV rates are observed when the vocabulary size increases. The smallest OOV rates are observed on French data, and the largest ones on Arabic data.

7. Acknowledgements

Part of this work was supported by the Chist-Era AMIS (Access Multilingual Information opinionS) project. Also, some experiments presented in this paper have been 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 (see <https://www.grid5000.fr>).

8. References

- [1] Rosenfeld, R., "Optimizing lexical and ngram coverage via judicious use of linguistic data", *EUROSPEECH'95, 4th European Conf. on Speech Communication and Technology*, pp. 1763-1766, Madrid, Spain, 1995.
- [2] Attia, M., Foster, J., Hogan, D., Roux, J. L., Tounsi, L., & Van Genabith, J., "Handling unknown words in statistical latent-variable parsing models for Arabic, English and French", *NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pp. 67-75, 2010.
- [3] Lindén, K., "A probabilistic model for guessing base forms of new words by analogy", *Computational Linguistics and Intelligent Text Processing*, pp. 106-116, 2008.
- [4] Attia, M., Samih, Y., Shaalan, K. F., & Van Genabith, J., "The Floating Arabic Dictionary: An Automatic Method for Updating a Lexical Database through the Detection and Lemmatization of Unknown Words". *COLING*, pp. 83-96, 2012.
- [5] Venkataraman, A., and Wang, W., "Techniques for effective vocabulary selection", *INTERSPEECH'2003, 8th European Conf. on Speech Communication and Technology*, pp. 245-248, Geneva, Switzerland, 2003.
- [6] Allauzen, A., and Gauvain, J.-L., "Automatic building of the vocabulary of a speech transcription system", In French "Construction automatique du vocabulaire d'un système de transcription", *JEP'2004, Journées d'Etudes sur la Parole*, Fès, Maroc, 2004.
- [7] Jouvet, D., and Langlois, D., "A machine learning based approach for vocabulary selection for speech transcription". *TSD'2013, Int. Conf. on Text, Speech and Dialogue*, Pilsen, Czech Republic, 2013.
- [8] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al., "The kaldi speech recognition toolkit", *ASRU'2011, IEEE Workshop on Automatic Speech Recognition and Understanding*, Waikoloa, HI, USA, 2011.
- [9] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups", *IEEE Signal Processing Magazine*, 29(6):82-97, 2012.
- [10] Graff, D., Mendonça, A, and DiPersio, D.. "French Gigaword Third Edition LDC2011T10". DVD. Philadelphia: Linguistic Data Consortium, 2011.
- [11] Graff, D., and Cieri, C., "English Gigaword LDC2003T05". Web Download. Philadelphia: Linguistic Data Consortium, 2003.
- [12] Parker, R., et al., "Arabic Gigaword Fifth Edition LDC2011T11". Web Download. Philadelphia: Linguistic Data Consortium, 2011.
- [13] Meftouh, K., Smaili, K., & Laskri, M. T., "Comparative Study of Arabic and French Statistical Language Models". *ICAART*, pp. 156-160, 2009.
- [14] Meftouh, K., Tayeb Laskri, M., & Smaili, K., "Modeling Arabic Language using statistical methods". *Arabian Journal for Science and Engineering*, 35(2), 69, 2010.

Building an ASR System for a Low-resource Language Through the Adaptation of a High-resource Language ASR System: Preliminary Results

Odette Scharenborg¹, Francesco Ciannella², Shruti Palaskar², Alan Black², Florian Metzger², Lucas Ondel³, Mark Hasegawa-Johnson⁴

¹ Centre for Language Studies, Radboud University, Nijmegen, The Netherlands

² Carnegie Mellon University, Pittsburgh, PA, USA

³ Brno University of Technology, Brno, Czech Republic

⁴ University of Illinois at Urbana-Champaign, Champaign, IL, USA

o.scharenborg@let.ru.nl

Abstract

For many languages in the world, not enough (annotated) speech data is available to train an ASR system. We here propose a new three-step method to build an ASR system for such a low-resource language, and test four measures to improve the system's success. In the first step, we build a phone recognition system on a high-resource language. In the second step, missing low-resource language acoustic units are created through extrapolation from acoustic units present in the high-resource language. In the third step, iteratively, the adapted model is used to create a phone transcription of the low-resource language, after which the model is retrained using the resulting self-labelled phone sequences to improve the acoustic phone units of the low-resource language. Four measures are investigated to determine which self-labelled transcriptions are 'good enough' to retrain the adaptation model, and improve the quality of the phone speech tokens and subsequent phone transcriptions: TTS and decoding accuracy to capture acoustic information, a translation retrieval task to capture semantic information, and a combination of these three. The results showed that in order to train acoustic units using self-labelled data, training utterances are preferably needed that capture multiple aspects of the speech signal.

Index Terms: Low-resource language, Automatic speech recognition, Adaptation, Linguistic knowledge

1. Introduction

Automatic speech recognition technologies require a large amount of annotated data for a system to work reasonably well. However, for many languages in the world, not enough speech data is available, or these lack the annotations needed to train an ASR system. In fact, it is estimated that for only about 1% of the world languages the minimum amount of data that is needed to train an ASR is available [1]. In order to build an ASR system for such a low-resource language, one cannot simply use a system trained for a different, even if related, language, as cross-language ASR typically performs quite poorly [2]. Different languages have different phone inventories, and even phones transcribed with the same IPA symbol are produced slightly differently in different languages [3].

Recently, different approaches have been proposed to build ASR systems for such low-resource languages. One strand of research focuses on discovering the linguistic units of the low-resource language from the raw speech data, while assuming no other information about the language is available, and using these to build ASR systems (the Zero-resource approach) [4]-[15]. Another strand of research focuses on building ASR

systems using speech data from multiple languages, thus trying to create universal or cross-linguistic ASR systems [16]-[19].

However, most of the world's languages have been investigated by field linguists, meaning that some information about the language typically is available. We here propose a method to adapt an ASR system for a high-resource language using linguistic information of the low-resource language to build an ASR system for that low-resource language. In addition to some unlabelled speech data (in line with the Zero-resource approach), we assume that a 'description' of the phone(me) inventory of the language is available, e.g., obtained from a field linguist. A second assumption is that enough annotated speech material of a related high-resource language is available to build an ASR system for that related high-resource language. Note, however, that the here-proposed system does not rely on having a high-resource related language; in principle, the approach presented here could work for any language pair. Experiments in cross-language ASR adaptation tend to report that adaptation between related languages is more successful than adaptation among unrelated languages [17], though many other factors seem to be equally important, including similarity of the speaker voices and recording conditions of the two speech corpora [18].

Because different languages have different phone inventories, whichever high-resource language we choose, some of the phones from the low-resource language will not be present in the high-resource language. For instance, when comparing Dutch and English, English has, e.g., the /æ/ (as in *fantastic*) and /θ/ (as in *three*) which are lacking from Dutch. So, in order to build an ASR system for a low-resource language, first the acoustic phone tokens of the low-resource language need to be discovered. We propose a three-step method: (1) Build a phone recognition system on a high-resource language, in our case Dutch. (2) The phone inventory of the ASR system trained on the high-resource language is remapped or 'transferred' to the phone inventory of the low-resource language. For those phones from the low-resource language that are not present in the high-resource language, acoustic units need to be created in the high-resource language ASR system. A 'baseline' or starting point for the missing acoustic unit of the low-resource language is then created by extrapolating between acoustic units that are present in the high-resource language. (3) The adapted model will iteratively be used to create a phone transcription of the low-resource language, after which the model will be retrained using the resulting self-labelled phone sequences in order to improve the acoustic phone units of the low-resource language.

The phone transcriptions created by the adapted models will contain errors which will have repercussions on the quality of the acoustic phone units. The main question for this paper is

therefore whether selecting only those self-labelled transcriptions that are ‘good enough’ to retrain the adaptation model will improve the quality of the acoustic phone units and subsequent phone transcriptions. These acoustic phone units should both capture the acoustic information of that phone and the semantic information correctly. Four criteria were investigated: ASR score, text-to-speech (TTS) synthesis score, translation text retrieval score, and a fusion of the three. In order to investigate the usefulness of these four criteria, a multi-modal database was needed, which in our case was the FlickrR_8K corpus [23],[6], so we chose to use English as a mock low-resource language.

2. Methodology

A baseline system was trained on Dutch, adapted to English, and then applied for the transcription of English utterances. The baseline was then compared to self-trained systems created using the four different criteria for determining which utterances to use in self-training. The different criteria capture different attributes of the speech, therefore we expect them to be complementary. ASR confidence scores measure the degree to which the transcription is a good match to the audio signal (relative to the model); in a sense, this is a measure of the phonetic quality of the transcription or the degree to which the transcription captures linguistically salient attributes of speech. TTS also measures phonetic quality, but with different models. TTS attempts to measure the adequacy of the transcription to capture all information that a human listener would hear. Translated text retrieval measures the degree to which the transcription is sufficient to communicate the meaning of the sentence. The experiments were run at the Pittsburgh Supercomputing Center (PSC; [20],[21]).

2.1. Speech materials

The Spoken Dutch Corpus (Corpus Gesproken Nederlands, CGN, [22]) is a corpus of almost 9M words of Dutch spoken in the Netherlands and in Flanders (Belgium), in over 14 different speech styles, ranging from formal to informal. For the experiments reported here, we only used the read speech material from the Netherlands, which amounts to 551,624 words for a total duration of approximately 64 hours of speech.

The English data came from the FlickrR_8K corpus [23],[6] which contains 5 different natural language text captions for each of 8000 images captured from the Flickr photo sharing website which were read aloud by crowdsource workers from Amazon Mechanical Turk. Additionally, within the context of the Frederick Jelinek Speech and Language Technology (JSALT) workshop 2017, tokenised translations into Japanese were obtained for each of the 40,000 captions. Moreover, forced alignments for FlickrR_8K were created using a DNN/HMM hybrid system using 8,000 CD states and logMELs as acoustic input features trained on data as described in [24].

To mimic a low-resource language we randomly selected 3660 utterances from the FlickrR_8K training set as a train-test set (training is done on a subset of this train-test set while testing is done on the train-test set, see also the Discussion Section), which corresponded to approximately 4 hours of speech (which corresponds to the number of hours of speech material for an actual low-resource language, Mboshi [1]).

2.2. Proposed systems: Baseline and self-trained

Figure 1 shows an overview of the proposed adaptation system. First, a *Baseline* DNN is trained on the Dutch CGN.

Next, the soft-max layer of the DNN is adapted from the Dutch to the English phone set (see Section 2.3): the *Adapted* model. Subsequently, the adapted soft-max layer is used to decode the English speech material using a free phone recognition pass. The projection and soft-max layers are then retrained with (1) all self-labelled utterances, or (2) only with those self-labelled utterances that have the best scores according to the four selection criteria. Two decoding and retraining iterations are carried out, yielding different *Self-trained* models.

All models are tested on our train-test set of 3660 utterances from FlickrR_8K. The accuracy of the output phone sequences of the different models is evaluated by comparing them to the gold standard as created by the forced alignment by calculating the edit-distance, and is reported as percentage Phone Error Rate (%PER).

2.2.1. Baseline model

The baseline model used for the experiments is trained using Connectionist Temporal Classification (CTC; [25]), implemented using Eesen [26]. The CTC paradigm uses a Recurrent Neural Network (RNN), trained using an error metric that compares the reference and hypothesis symbol sequences with no regard to the time alignment of symbols. The CTC-RNN models the mapping between the speech signal and the output labels without the need for an explicit segmentation of the speech signal into output labels (typically obtained using a forced-alignment), and models all aspects of the sequence within a single network architecture by interpreting the network outputs as a probability distribution over all possible label sequences, conditioned on a given input sequence.

The baseline RNN uses a six layer bidirectional LSTM Recurrent Neural Network. Each LSTM layer has 140 LSTM cells, and LSTM layers are connected using 80-dimensional projection layers. There is also an 80-dimensional projection layer at the input of the LSTM, which reduces the dimensionality of the input features, which consists of 3 stacked frames (at 10ms distance) of 40-dimensional FBank features. The network step size is 30 ms. The final LSTM outputs are connected to another 80-dimensional projection layer which is connected to the phone soft-max layer. The size of the soft-max layer depends on the phone set of the language; see Section 2.3. The network has been trained with Stochastic Gradient Descent for 20 epochs, using phones as targets.

We apply greedy decoding and thus take the output of the CTC network (consisting of a probability distribution over the phone set) and at every frame select the phone with the highest probability. Sequences of adjacent outputs with the same value are clustered into the same phone, and blanks (used by CTC to fill the distance between phonetic detections) are discarded. Note that all phones have an equal prior probability.

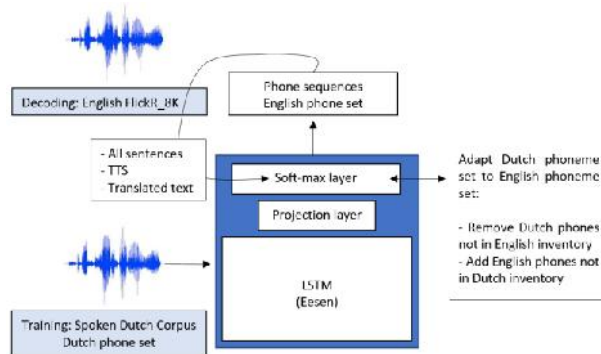


Figure 1. Overview of the proposed adaptation system.

2.2.2. Translated text retrieval

The translated text retrieval system is trained to retrieve the ID of the Japanese translated text from a database of Japanese translated texts which corresponds to the English transcription of the spoken utterance that is presented at its input. The Japanese translated texts database consists of all 3660 train-test utterances. The retrieval system is based on the image retrieval system by [6] but instead of matching images, we are matching phone sequences to translated texts. The retrieval system is implemented in the xnmt sequence-to-sequence neural machine translation architecture [31] using the DyNet neural network library [32]. The source and target encoders both consist of an input layer, LSTM hidden layer, and an output layer, each containing 512 nodes. The embeddings output by the encoders are fed into the retriever which calculates the dot-product of the two encoders and takes the smallest as the best match. The source encoder takes as input the phone sequences output by the soft-max layer of the CTC-DNN. The target encoder takes as input the IDs of the translated text utterances. Adam-training with a learning rate of 0.001 is used to map the phone sequences to the IDs of the Japanese translations. Training runs for 100 epochs. After training, the 3660 training utterances are run through the retrieval system again in order to retrieve the phone sequences that score best on the retrieval task. These are those utterances for which the correct ID of the translated text appears in the top N=10 of answers.

2.2.3. ASR score

The best scoring ASR utterances are those phone sequences that have the lowest PER on the train-test set. The number of selected phone sequences was identical to the number of phone sequences obtained from the translation retrieval measure.

2.2.4. TTS score

The TTS system used is ClusterGen [27]. TTS typically consists of four stages. First, text is converted to a graph of symbolic phonetic descriptors. This step is omitted in our case, as the output of the *Adaptation* model already consists of a sequence of phones. Second, the duration of each unit in the phonetic graph is predicted. Third, every frame in the training database is viewed as an independent exemplar of a mapping from discrete inputs to continuous outputs, and a machine learning algorithm (e.g., regression trees [27] and random forests [14]) is applied to learn the mapping. Discrete inputs include standard speech synthesis predictors such as the phone sequence and prosodic context, as well as variables uniquely available to ClusterGen such as the timing of the predicted frame with respect to segment boundaries at every prosodic level. Continuous outputs include the excitation and pitch, the mel-cepstrum [29], and a representation of the dynamic trajectory of the mel-cepstrum (its local slope and curvature); mel-cepstrum of the continuous signal is then synthesized using trajectory overlap-and-add. ClusterGen works well with small corpora because it treats each frame of the training corpus as a training example, rather than each segment, and it can be used to train a TTS system for a low-resource language (see [30] for an example). This makes it suitable for our low-resource scenario.

Synthesized speech can be compared to a reference speech signal using mean cepstral distortion (MCD, [27]). MCD measures the average distance between the log-spectra of the synthetic and natural utterances. MCD has been demonstrated to be an extremely sensitive measure of the perceived naturalness of speech utterances, e.g., an MCD difference between two synthesis algorithms of 0.3 (on the same test

corpus) is usually perceptible by human listeners as a significant difference in perceived naturalness [27].

In the experiments reported here, MCD measured the difference between synthetic and reference speech signals. Low MCD suggests that the ASR generated a pretty reasonable transcription of the utterance. MCD of the re-synthesis was therefore used as the third of our selection criteria.

2.2.5. Fusion of scores

System combination, of systems with complementary error patterns, often yields a combination system whose PER is lower than the PER of any component system [33]. Since translation, TTS, and ASR all capture different aspects of the speech signal, we expect that the PER of the combination system should be lower than the PERs of any component system. We therefore retrain the *Adaptation* model with those phone sequences that capture the acoustic and the semantic information best, i.e., we select those N utterances that appear in at least two of the three best utterances lists (giving preference to the combination of translation retrieval + TTS or ASR), where N is equal to the number of utterances for the other measures.

Table 1. Mapping of the English (L2) phone not present in the Dutch phoneme inventory, with an example of the sound (indicated with bold) in an English word.

Missing L2 phone	Example	Mapping		
		L1:1	L1:2	L1:3
æ	map	ε	a	ε
ʌ	cut	ε	ɑ	a
ð	they	v	z	v
ɜ	bird	ø	o	ø
θ	three	f	s	f
ʊ	book	i	u	i

2.3. Adaptation of the soft-max layer

The number of different Dutch phones in CGN is 42, while the English FlickR_8K has 45 different phones. There are three reasons for the difference between the phone sets. (1) Nine English phones are diphthongs or affricates which do not exist in Dutch, but which can easily be constructed from a sequence of two Dutch phones. These nine English phones are not represented in the soft-max layer but dealt with in a post-processing step. (2) Eleven Dutch phones are not present in English and these are removed from the soft-max layer. (3) Six English phones do not exist in Dutch (referred to as missing L2 phones) and need to be added to the soft-max layer. Vectors in the soft-max layer are created for these missing L2 phones on the basis of the trained Dutch (L1) phones; the created soft-max nodes are then adapted using the speech data selected according to the selection criteria described in Section 2.2.

The desired English-language phones are initialized by linearly extrapolating the missing L2 (English) node in the soft-max layer from existing vectors for the Dutch L1 phones using:

$$\vec{V}_{|\varphi|,L2} = \vec{V}_{|\varphi|,L1:1} + 0.5 (\vec{V}_{|\varphi|,L1:2} - \vec{V}_{|\varphi|,L1:3}) \quad (1)$$

where $\vec{V}_{|\varphi|,L2}$ is the vector of the missing L2 phone $\varphi,L2$ that needs to be created, $\vec{V}_{|\varphi|,L1:x}$ are the vectors of the Dutch L1 phones $\varphi,L1:x$ in the soft-max layer that are used to create the vector for the missing English phone $\varphi,L2$. Among the three Dutch phones, L1:1 refers to the phone which is used as the starting point from which to extrapolate the missing L2 phone, and L1:2 and L1:3 refer to the L1 phones whose displacement

is used as an approximation of the displacement between the Dutch L1 vector and the L2 phone that should be created. Table 1 lists the six missing L2 phones, and the Dutch L1 phones that are used to create the vectors for the missing English L2 phones.

3. Results

The PER of the *Adaptation* model, i.e., the *Baseline* model for which the soft-max layer had been adapted to the English phone set but not yet retrained, is 72.59%. Table 2 shows the PER results for the *Self-trained* models, i.e., the models after retraining. Iteration 1 refers to the models for which the projection layer and soft-max layer have been retrained with the best scoring self-labelled phone sequences according to the ASR, TTS, translation retrieval system, or the combination of these. Iteration 2 refers to the models for which the projection layer and soft-max layer have been retrained with the best-scoring (according to the DNN, TTS, and retrieval task) self-labelled utterances of the corresponding models after Iteration 1. The number of phone sequences used for retraining was 2468 (=67.43% Recall@10 of the translation retrieval task) for iteration 1 and 2101 (=57.40% R@10) for iteration 2.

Formal statistical significance tests have not yet been performed for these data, but an overly conservative model can be defined: if we assume that phone errors within a speech file are 100% correlated, and follow a Bernoulli model [34], then two ASR systems are significantly different if their PERs differ by at least $50\%/\sqrt{3660}=0.83\%$. By this overly conservative standard, 3 of the 5 systems at Iteration 1 and the fusion system at Iteration 2 are significantly better than the baseline (see bold numbers in Table 2), and there is no significant difference among these different methods of selecting self-labelled utterances. Except for the fusion system, the systems in the second iteration, however all performed worse than the Iteration 1 models, occasionally even worse than the baseline model.

4. Discussion and conclusions

We proposed a three-step method to build an ASR system for a low-resource language through the adaptation of an ASR system of a high-resource language, using a combination of linguistic knowledge and semi-supervised learning. Crucially, acoustic tokens of the phones that are present in the low-resource language but not in the high-resource language are created through a linear extrapolation between existing acoustic units in the soft-max layer after which the acoustic units are iteratively retrained using all utterances or only those utterances that have the best score according to four different criteria: ASR score, TTS score, translation retrieval score, and their fusion.

The baseline PER is comparable to the PERs of cross-language ASR systems (e.g., [2] reports PERs between 59.83% and 87.81% for 6 test languages). Re-training the system, using a self-labelling approach with confidence scoring, can significantly improve PER after the first iteration (see Table 2). The differences between the different approaches are however small, a more sensitive statistical test might demonstrate significance of some of the differences in Table 2. Retraining the systems for a second pass decreased performance for all measures but the fusion system, even surpassing the *Baseline*'s performance for some measures. Note, we used the training set also as a test set: 1) because the amount of available training data was so low that creating an independent test set would even further reduce the size of the training set; 2) the training set was only used to retrain the soft-max layer, not the hidden layers. Future research will test this method on an independent test set.

Table 2. *Phone error rates (%PER) on the 3660 FlickrR_8K train-test utterances for the different self-trained models. Bold indicates significantly better performance than Baseline.*

Selection criterion	Iteration 1 (2468 utts)	Iteration 2 (2101 utts)
All sentences	71.80	72.56
ASR	71.67	72.42
TTS	71.71	72.52
Translation retrieval	71.83	72.88
Fusion	71.76	71.72

The projection and soft-max layers were retrained using only the best scoring phone sequences according to four different criteria. However, since neural networks are extremely data hungry, the (in principle) improved quality of the training utterances at Iteration 2 did not outweigh the substantial decrease in training data from Iteration 1 to Iteration 2. The system retrained on all sentences, on the other hand, might have suffered from the presence of a couple of bad transcriptions. Only the fusion model's performance did not decrease from Iteration 1 to Iteration 2. This is likely due to the training utterances of this system capturing *both* phonetic and semantic information well. Thus, in order to train acoustic units using self-labelled data, training utterances are needed that capture multiple aspects of the speech signal.

Instead of discarding the bad data, future work will investigate the use of data augmentation methods to increase the importance of the good data. [34] demonstrated that ASR could be improved by making "perturbed" copies of each of the input waveforms, thus increasing the size of the training dataset. Perturbations include pitch shifting, speeding up, slowing down, or adding certain types of noise at different SNRs. The best-scoring phone sequences would then receive a duplication factor that is larger than those phone sequences which have a lower score. Relatedly, this would allow us to refine the fusion method by not using the majority vote but rather use the intersection of the three measures. Moreover, the current retraining only updates the projection and soft-max layers, because of the fairly low amount of (re)training data that is available. However, future work will investigate the effect of retraining the whole LSTM, or also introduce projections in the temporal dimension, or update only specific LSTM layers.

Although the aim of the paper is to investigate the possibility to build an ASR system for a low-resource language through the adaptation of an ASR system build for a high-resource language, the low-resource language we used in the current work is not an actual low-resource language. We plan to test our method on Mboshi [1], a Bantu language which is an actual low-resource language.

5. Acknowledgements

The work reported here was started at JSALT 2017 in CMU, Pittsburgh, and was supported by JHU and CMU via grants from Google, Microsoft, Amazon, Facebook, and Apple. This work used the Blacklight system (NSF award number ACI-1041726) of XSEDE (NSF award number ACI-1053575) at the Pittsburgh Supercomputing Center (PSC). OS was partially supported by a Vidi-grant from NWO (276-89-003). The authors would like to thank Markus Müller for providing the forced alignments of FlickrR_8K, and Graham Neubig for providing the tokenised Japanese translations, and for implementing and providing the xnmt/dynet tools for the translated text retrieval task.

6. References

- [1] Adda, G., Stüker, S., Adda-Decker, M., Ambourou, O., Besacier, L., Blachon, D., Bonneau-Maynard, H., Godard, P., Hamlaoui, F., Idiatov, D., Kourata, G.-N., Lamel, L., Makasso, E.-M., Rialland, A., Van de Velde, M., Yvon, F., Zerbian, S., “Breaking the unwritten language barrier: The BULB project”, Proc. SLTU-2016 5th Workshop on Spoken Language Technologies for Under-resourced languages, 2016.
- [2] Hasegawa-Johnson, M., Jyothi, P., McCloy, D., Mirbagheri, M., di Liberto, G., Das, A., Ekin, B., Liu, C., Manohar, V., Tang, H., Lalor, E.C., Chen, N., Hager, P., Kekona, T., Sloan, R., Lee, A.K.C., “ASR for under-resourced languages from probabilistic transcription,” *IEEE/ACM Trans. Audio, Speech and Language* 25(1):46-59, 2017. doi:10.1109/TASLP.2016.2621659
- [3] Huang, P.-S., Hasegawa-Johnson, M., “Cross-dialectal data transferring for Gaussian Mixture Model training in Arabic speech recognition,” in *Internat. Conf. Arabic Language Processing (CITALA)* pp. 119-122, ISBN 978-9954-9135-0-5, Rabat, Morocco, 2012.
- [4] Jansen, A., Dupoux, E., Goldwater, S., Johnson, M., Khudanpur, S., Church, K., Feldman, N., Hermansky, H., Metze, F., Rose, R., “A summary of the 2012 JHU CLSP workshop on zero resource speech technologies and models of early language acquisition,” Proc. ICASSP, 2013.
- [5] Ondel, L., Burget, L., Cernocky, J., “Variational Inference for Acoustic Unit Discovery”, *Procedia Computer Science*, 81, Elsevier Science, http://www.fit.vutbr.cz/research/view_pub.php?id=11224, 2016.
- [6] Harwath, D., Glass, J., “Deep multimodal semantic embeddings for speech and images,” *IEEE Automatic Speech Recognition and Understanding Workshop*, Scottsdale, Arizona, USA, 237-244, 2015.
- [7] Badino, L., Canevari, C., Fadiga, L., & Metta, G., “An auto-encoder based approach to unsupervised learning of subword units”, Proc. ICASSP, 2014.
- [8] Huijbrechts, M., McLaren, M., van Leeuwen, D., “Unsupervised acoustic sub-word unit detection for query-by-example spoken term detection”, Proc. ICASSP, 4436-4439, 2011.
- [9] Lee, C., Glass, J., “A nonparametric Bayesian approach to acoustic model discovery”, Proc. 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, 40-49, 2012.
- [10] Varadarajan, B., Khudanpur, S., Dupoux, E., “Unsupervised learning of acoustic subword units”, Proc. ACL-08: HLT, 165-168, 2008.
- [11] Jansen, A., Van Durme, B., “Efficient spoken term discovery using randomized algorithms”, Proc. Automatic Speech Recognition and Understanding (ASRU), 401-406, 2011.
- [12] Park, A. S., Glass, J. R., “Unsupervised pattern discovery in speech”, Proc. ICASSP, 16(1), 186-197, 2008.
- [13] Zhang, Y., Glass, J. R., “Towards multi-speaker unsupervised speech pattern discovery”, Proc. ICASSP, 4366-4369, 2010.
- [14] Harwath, D., Torralba, A., Glass, J., “Unsupervised learning of spoken language with visual context”, *Advances in Neural Information Processing System*, 1858-1866, 2016.
- [15] Chrupała, G., Gelderloos, L., Alishahi, A., “Representations of language in a model of visually grounded speech signal”, arXiv:1702.01991v3 [cs.CL] 30 Jun 2017.
- [16] Schultz, T., Waibel, A., “Experiments on cross-language acoustic modelling,” Proc. Interspeech, 2001.
- [17] Löff, J., Gollan, C., Ney, H., “Cross-language bootstrapping for unsupervised acoustic model training: rapid development of a Polish speech recognition system,” Proc. Interspeech, 2009.
- [18] Vesely, K., Karafiát, M., Grezl, F., Janda, M., Egorova, E., “The language-independent bottleneck features,” in Proc. SLT, 2012.
- [19] Xu, H., Do, V.H., Xiao, X., Chng, E.S., “A comparative study of BNF and DNN multilingual training on cross-lingual low-resource speech recognition,” Proc. Interspeech, 2132-2136, 2015.
- [20] Towns, J., Cockerill, T., Dahan, M., Foster, I., Gaither, K., Grimshaw, A., Hazlewood, V., Lathrop, S., Lifka, D., Peterson, G.D., Roskies, R., Scott, J.R. and Wilkens-Diehr, N., “XSEDE: Accelerating scientific discovery”, *Computing in Science & Engineering*. 16(5):62-74, 2014.
- [21] Nystrom, N., Welling, J., Blood, P. and Goh, E.L., “Blacklight: Coherent shared memory for enabling science”, In *Contemporary High Performance Computing: From Petascale Toward Exascale*; Vetter J., Ed.; CRC Computational Science Series, Taylor & Francis:Boca Raton, 431-450, 2013.
- [22] Oostdijk, N.H.J., Goedertier, W., Van Eynde, F., Boves, L., Martens, J.-P., Moortgat, M., Baayen, H., “Experiences from the Spoken Dutch Corpus project”, Proc. LREC – Third International Conference on Language Resources and Evaluation, Las Palmas de Gran Canaria, 340-347, 2002.
- [23] Rashtchian, C., Young, P., Hodosh, M., Hockenmaier, J., “Collecting image annotations using Amazon’s Mechanical Turk”, Proc. NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk, 2010.
- [24] Nguyen, T.-S., Müller, M., Sperber, M., Zenkel, T., Kilgour, K., Stüker, S., Waibel, A., “The 2016 KIT IWSLT speech-to-text systems for English and German”, Proc. 13th International Workshop on Spoken Language Translation (IWSLT), Seattle, USA, December 8-9, 2016
- [25] Graves, A., Fernandez, S., Gomez, F., Schmidhuber, J., “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” Proc. 23rd international conference on Machine learning (ACM), 369-376, 2006.
- [26] Miao, Y., Gowayyed, M., Metze, F., “EESSEN: End-to-end speech recognition using deep RNN models and WFST-based decoding,” arXiv:1507.08240v3, 2015.
- [27] Black, A.W., “CLUSTERGEN: A statistical parametric speech synthesizer using trajectory modeling”, Proc. ICSLP, 1762-1765, 2006.
- [28] Black, A.W., Kumar Muthukumar, P., “Random forests for statistical speech synthesis”, *Proceeding of Interspeech*, 1211-1215, 2015.
- [29] Fukada, T., Tokuda, K., Kobayashi, T., Imai, S., “An adaptive algorithm for mel-cepstral analysis of speech”, Proc. ICASSP, 1992. doi="10.1109/ICASSP.1992.225953.
- [30] Hasegawa-Johnson, M., Black, A., Ondel, L., Scharenborg, O., Ciannella, F. (2017). Image2speech: Automatically generating audio descriptions of images. *Proceedings of the International Conference on Natural Language, Signal and Speech Processing, Casablanca, Morocco*.
- [31] <https://github.com/neulab/xnmt>
- [32] Neubig, G., Dyer, C., Goldberg, Y., Matthews, A., Ammar, W., Anastasopoulos, A., Ballesteros, M., Chiang, D., Clothiaux, D., Cohn, T., Duh, K., Faruqui, M., Gan, C., Garrette, D., Ji, Y., Kong, L., Kuncoro, A., Kumar, G., Malaviya, C., Michel, P., Oda, Y., Richardson, M., Saphra, N., Swayamdipta, S., Yin, P., “DyNet: The dynamic neural network toolkit”, arXiv preprint arXiv:1701.03980, 2017.
- [33] Fiscus, J., “A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER),” *IEEE Workshop ASRU* pp. 347-354, 1997.
- [34] Gillick, L., Cox, S.J., “Some statistical issues in the comparison of speech recognition algorithms,” Proc. ICASSP 532-535, 1989.
- [35] Jaitly, N., Hinton, G.E., “Vocal tract length perturbation (VTLF) improves speech recognition”, Proc. ICML Workshop on Deep Learning for Audio, Speech and Language, 2013.

Data Selection in the Framework of Automatic Speech Recognition

Ismael Bada, Juan Karsten, Dominique Fohr, Irina Illina

MultiSpeech team

Université de Lorraine, LORIA, UMR 7503, Vandoeuvre-lès-Nancy, F-54506, France

Inria, Villers-lès-Nancy, F-54600, France

CNRS, LORIA, UMR 7503, Vandoeuvre-lès-Nancy, F-54506, France

Abstract

Training a speech recognition system needs audio data and their corresponding exact transcriptions. However, manual transcribing is expensive, labor intensive and error-prone. Some sources, such as TV broadcast, have subtitles. Subtitles are closed to the exact transcription, but not exactly the same. Some sentences might be paraphrased, deleted, changed in word order, etc. Building automatic speech recognition from inexact subtitles may result in a poor models and low performance system. Therefore, selecting data is crucial to obtain a highly performance models. In this work, we explore the lightly supervised approach, which is a process to select a good acoustic data to train Deep Neural Network acoustic models. We study data selection methods based on phone matched error rate and average word duration. Furthermore, we propose a new data selection method combining three recognizers. Recognizing the development set produces word error rate that is the metric to measure how good the model is. Data selection methods are evaluated on the real TV broadcast dataset.

Index terms: speech recognition, neural networks, acoustic model, data selection

1. Introduction

Automatic speech recognition (ASR) is one of the sub field of natural language processing with many practical applications: automatic closed captioning for hearing-disabled persons, taking notes of conversations between doctors and patients, voice control and many more. Despite of the rapid development of speech recognition, there are still many challenges in the field. One of the challenges is the training of a speech recognizer, which requires a huge amount of transcribed training data. The transcribed training data consists of audio data and the corresponding text transcriptions. However, transcribing audio manually is labor intensive and also time consuming. There exist many unlimited supply of audio data from internet, TV broadcasts, radio, as well as video streaming websites, but there is no available exact transcription. However, some TV broadcasts, such as *CNN headline news*, *ABC world news tonight*, *BBC*, have subtitles that can be used for training a speech recognition system.

To train a speech recognition system, one possibility is to use TV broadcasts data that have subtitles. These subtitles are close, but not exactly the same as what people uttered. Some sentences might be paraphrased, deleted, changed in word order, etc. There are some examples of approximate subtitles:

- Real transcription:
Russia started badly with the dropping at the hands of Spain. But, they got better and better. Spain looked

unstoppable to start with but since then they have looked a little.

- Corresponding subtitle:
Russia started badly with at beating at the hands of Spain. Spain looked then they have looked a little.

Furthermore, subtitles are often badly aligned with the audio. Some segments in training audio can contain unconstrained conversational speech, use of foreign words, high out-of-vocabulary rates, channel noise and simultaneous speech from more than one speaker. Even, thus audio data is sometimes difficult to be recognized by humans. These facts make hard to use subtitles for ASR.

The idea of using untranscribed audio data (or unsupervised training, no subtitles) has been proposed firstly in [15] and [5]. Authors of [15] proposed an iterative training procedure: decode untranscribed data and keep only the segments with high confidence score for the next training iteration. Even an 80% error rate system can improve itself automatically, but the system performance is limited. [6] were the first to propose *lightly supervised training* with a large amount of training data. Instead of using untranscribed training data, they trained speech recognition system using audio data with subtitles. Lightly supervised approach allows selecting "good" training data. First, an acoustic model from another task (or another corpus) is used to recognize audio data. The decoding results are compared with the subtitles and removed if they disagree. These selected data are used to train a new acoustic model. [3] proposed the confidence measure metric to remove the bad audio segments. When decoding acoustic inputs, an ASR produces word hypothesis and their corresponding confidence measure. The confidence measure value is used to remove potentially bad segments where the confident value is lower than a threshold. [10] applied lightly supervised approach on medical conversation data.

Very recently, a new point of view on the data selection has been proposed. [8] suggest an original two-stage crowdsourcing alternative. First, iteratively collects transcription hypotheses from the web and, then, asks different crowds to pick the best of them. [9] proposed an approach to domain adaptation that does not require transcriptions but instead uses a corpus of unlabeled parallel data, consisting of pairs of samples from the source domain of the well-trained model and the desired target domain.

In the present paper, the same problem of data selection for acoustic modeling training using a huge data corpus is considered. We want to select a good acoustic data to train Deep Neural Network acoustic models. The scientific contributions of this paper are:

- We study the impact of data selection on the word error rate.
- We explore different variations of slightly supervised training of acoustic models.

- We present a comparison of different data selection approaches in the context of TV broadcast news speech transcription.

2. Methodology

2.1. Lightly supervised data selection

To generate an accurate speech recognition, a very large training audio corpus with its exact corresponding transcription is required. This is particularly true for Deep Neural Network (DNN) based systems, having millions of parameters to train. However, transcribing audio is labor intensive and time consuming. There are unlimited supply of audio data in the internet, television, radio and other sources. But very few have available transcription. However, some TV broadcasts have subtitles. By utilizing these audio data with the corresponding subtitles, we hope to produce a high performance speech recognizer with less supervision. Nevertheless, some problems exist when using the data with subtitles as training dataset. Training using the subtitles faces several disadvantages compared to the manual transcriptions: indication of non-speech events (coughing, speaker turn) and acoustic conditions (background noise, music, etc.) are missing.

The main idea of lightly supervised approach is to use the automatic speech recognizer to transcribe training audio data. After this, only well transcribed segments (segments where automatic transcription corresponds to subtitles) will be used as training data [6].

We assume that we have a massive amount of training audio data and corresponding subtitles. In general, the lightly supervised approach operates as follow:

1. Randomly select a subset of the training set.
2. Train an acoustic model on a small amount of manually annotated data or use an acoustic model from another task.
3. Using ASR, recognize all training audio data.
4. Align the automatic transcriptions with the subtitles of the training data. Some transcriptions and subtitles might disagree. We can remove or correct these segments.
5. Retrain a new acoustic model using the data we selected in the previous step.
6. Optionally reiterate from step 3.

These steps can be iterated several times as long as the error rate is decreasing. This method uses the idea of training acoustic models in less supervised manner because the training dataset (subtitles) is not the actual transcription. Using subtitles as training data greatly reduces the manual transcription effort (20-40 time less).

2.2. Revisited lightly supervised data selection

In the lightly supervised approach presented previously, a very important step is the step 3. In the case of a disagreement between automatic transcriptions and subtitles, which part of subtitles to keep and which part to remove or correct? Can we use additional criteria to better choose the training data? How many training data to keep? In this section, we propose to study some of these questions.

2.2.1. Using AWD and PMER

According to [7], using of *Average Word Duration* (AWD) and *Phone Matched Error Rate* (PMER) during the data selection step (step 3) allows increasing greatly the quality of the selected training data. AWD is used as metric to detect if errors occur in aligning the start and end time of a segment or if something went wrong in the recognition process.

$$AWD = \frac{\text{utterance duration in second}}{\text{number of words in the recognized utterance}}$$

Usually, duration of a word cannot exceed an upper limit threshold and the duration cannot be lower than a bottom limit threshold. If it is the case, this means that the corresponding transcription or subtitle is wrong.

Phone Error Rate (PER) and *Word Error Rate* (WER) are usually used to measure the performance of a speech recognition system:

$$WER = 100 * \frac{\text{substitutions} + \text{deletions} + \text{insertions}}{\text{number_of_words}}$$

Error rate is obtained by comparing exact transcriptions and decoding transcriptions produced by the speech recognition system. Word error rate is obtained by the comparison at the word level, phone error rate at the phone level. Our training set has only subtitles. So we can only compare subtitles and recognized transcriptions. To avoid the confusion, we will use *Phone Matched Error Rate* (PMER). High PMER shows that at phone level the corresponding subtitle is very different compared to recognized transcription. This means possible problems in audio signal (noise, music) or in subtitle. In this case, it is better to discard this segment from the training set.

We chose to use PMER and not WMER because we use phone acoustic models. During acoustic training we interested by the phone sequence and not by the word sequence. For example, the words “too” and “two” have the same sequence of phones: /t u w /. If we misrecognized “two” instead of “too”, it will be sad to reject corresponding subtitle since these two words have a same phone sequence.

In our work, we propose to use these measures to increase the quality of the data selection. The proposed iterative methodology is as follow:

1. Randomly select a subset of the training set. This set is used to train an initial acoustic model.
2. Train an acoustic model using the audio and the subtitles of this training set.
3. Decode the full training set with the obtained acoustic model. This will produce new decoding results and new values of PMER and AWD. The new values of PMER are obtained from comparing the subtitles and the decoding results.
4. Select the subtitles from the training set based on AWD as follow: $threshold_1 < AWD < threshold_2$
5. Sort the obtained segments according to PMER. Choose N hours of the top PMER segments to make a new training set to train the next acoustic model.
6. Continue the step 2-5 until the data selection does not improve anymore.

At each iteration, the number N of selected hours can be augmented. To measure the improvement of the approach at each iteration, a development set recognition could be performed.

2.2.2. System combination

Usually, different ASR systems (with different acoustic models and/or language models) will make different errors. Thus, if several systems provide the same transcription as the original subtitle for one segment, it is very likely that the subtitle corresponds exactly to what has been uttered. We can use it reliably for training acoustic models.

The general idea of system combination approach is to combine different ASR systems by varying the language models or acoustic models or both. We have chosen to vary the language model because an acoustic model variation is a very time consuming task when we use a huge training set. The language models can be built with different constraints. A constrained

language model is trained only with the sentences of the training corpus used for selection. A less constrained LM is trained also on data from different sources. The idea is as follows: if the recognition result of one ASR and the recognition result of another ASR are the same, we can trust this recognition result. The proposed combination approach works almost in the same way as the method of section 2.3: training acoustic model with the subset of training data (audio data and their subtitles), recognizing and selecting from the full training data and repeating these steps to do better data selection. However, the difference lies in the recognition and data selection steps.

We built three speech recognition systems and each recognition system is used to perform recognition of the same training set. Consequently, we had three transcriptions which have the same amount of segments. We average the value of PMER and AWD from three corresponding decoding transcriptions. After this, we select the training data (step 4 of the approach presented in section 2.2.1) with the proposed combination algorithm:

Select the subtitles from the training set based on AWD as follow: $threshold_1 < AWD < threshold_2$

If a segment has zero PMER with one ASR, select the segment and corresponding subtitles

Else

If a segment have the same phone sequence using two ASRs and $PMER < threshold_{PMER}$, select the segment and the corresponding subtitle.

Else sort by average PMER. Choose top N hours segments with the lowest PMER. These subtitles will be chosen to train the next acoustic model.

We hope that using the recognition results when two ASRs agree will help. If a development corpus is available, the thresholds can be chosen to minimize word error rate.

3. Experiments

3.1. Audio corpus

We used the data from the *Multi Genre Broadcast (MGB)* challenge [1], [16]. MGB is a challenge to automatically transcribe TV broadcasts. TV broadcast data are recorded in highly diverse environments, speech with background music, non-speech events and sounds, etc. The challenge organizers only provided TV broadcast audio data and their corresponding subtitles. As presented previously, subtitles may be different compared to the actual transcription due to deletion, insertion, substitution and paraphrasing. Thus, MGB data recognition is a very difficult task.

MGB challenge data consists of:

1. A training set contains audio data with their corresponding subtitles. This training set is used for training speech recognition systems.
2. A development set contains around 8 hours of audio data and their corresponding *manual* transcriptions (exact transcriptions). This dataset can be used to evaluate studied approaches.
3. A text corpus: 640 million words of TV subtitles are provided. These data can be used to train ASR language model.

Datasets	# of shows	# hours of shows	# hours of speech
Training	751	470	349
Development	16	8.8	6.8

Table 1: MGB challenge datasets.

Table 1 shows the statistics of the training and the development sets. We can see that, in average, each show contains about 2/3 of speech and 1/3 of non-speech events. These non-speech events are difficult to recognize.

3.2. Transcription system

KATS (*Kaldi-based Automatic Transcription System*) speech recognition system is based on Context Dependent HMM-TDNN phone models [11] [2] [4]. We used Kaldi toolkit for training and for recognition [13]. The TDNN architecture has 6 hidden layers, each hidden unit utilizes Rectified Linear Unit (RELU) activation function. The TDNN has around 9100 output nodes (senones) with *softmax* activation function. The feature vectors are MFCC with 40 feature values. The baseline phonetic lexicon contains 118k pronunciations for 112k words. Using the *pocollm* [12], 3-gram language model is estimated on text corpora of about 640 million words.

4. Experimental results

4.1. Study of MGB training data

<i>Number of segments (subtitles)</i>	253 K
<i>Average segment duration</i>	4.96 sec
<i>Average number of words per segment</i>	14.4
<i>Vocabulary size</i>	52 K
<i>Total number of words</i>	3 650 K

Table 2: MGB challenge train set statistics.

Table 2 and figure 1 present some statistics of the training set of MGB. From the figure we observe that the training set has a high number of segments (subtitles) of average duration of 4.96 seconds and about 14.4 words per segment. Figure 1 shows that there is a large number of subtitles with only few words, so they correspond to a very short speech duration. These short segments can be not easy to recognize by an ASR.

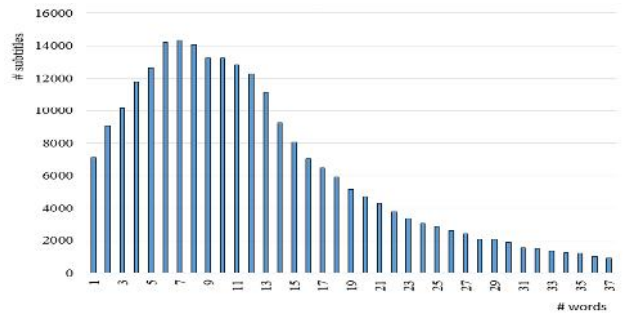


Figure 1: Histogram of number of words in function of number of subtitles. MGB training set.

Figure 2 displays the histogram of number of hours of speech in function of AWD for the training set. AWD values were given by the MGB organizers and obtained after ASR recognition. We can see that the majority of speech segments have an AWD between 0.25 seconds and 0.6 seconds. If one segment has a very small AWD or a very high one, it means that something went wrong and this segment corresponds rather to non-speech events. For safety reasons, for data selection we have extended this interval and we chose AWD between 0.16 and 0.6 for the following experiments.

Table 3 presents the number of speech hours in function of PMER on the training set. PMER values for each segment were given by the MGB organizers. We can observe that one third of the training data have PMER greater than 30%. This means that if we want to keep only a very good training data with very low PMER (so, with a very good quality subtitles), we will have a small training set. In contrast, if we keep all data, a large number

of subtitles do not correspond to what was uttered and a negative impact on training is observed.

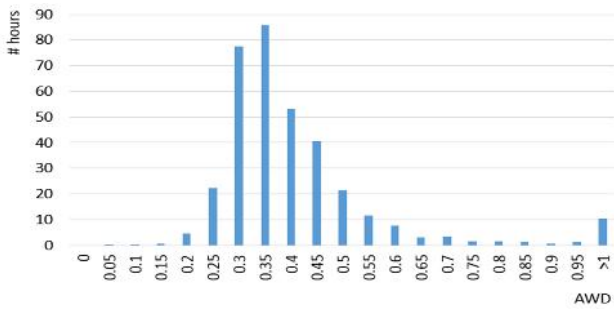


Figure 2: Distribution of number of hours of speech in function of AWD (in seconds) for the training set

PMER	<3	<15	<30	<50	<80	All
# hours	112	210	260	304	311	349
Duration %	32	60	74	87	89	100

Table 3: Number of hours of training speech according to PMER. Duration (%) as percentage of the total train set.

4.2. Impact of the data selection

In order to assess the influence of the data selection, we trained different ASR systems with different amount of training data. The amount of training hours is selected according to the PMER provided by the MGB organizers. For these experiments, we kept only training data with the PMER below some threshold.

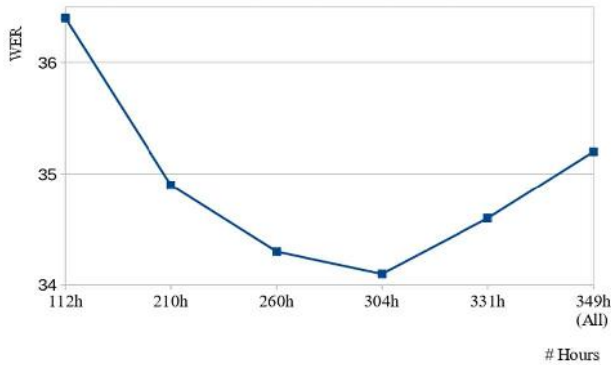


Figure 3: WER on the development set according to the number of hours selected for training the ASR system.

Figure 3 presents the WER on the development set according to the number of hours selected for training the ASR system (see Table 3). For example, for PMER below 50 we kept only 305 hours of corresponding training data. The trained system is used to recognize the development data and the obtained WER is presented in Figure 3. From this figure we can observe that, at first, WER decreases when the amount of training data increases. But selecting too much data (i.e. using subtitles with high PMER) the WER begins to increase. Therefore, it is important to find a compromise between the quantity of training data and the quality of training data. In conclusion, data selection is important to train an efficient ASR.

4.3. Results of data selection methods

We studied and evaluated the presented data selection approaches on the development corpus.

The execution time of one iteration of data selection takes about 43 hours. This is very time consuming. To speed up the experiments, at each iteration of data selection, we decided to select a different number of hours of data (parameter N in the selection algorithm). We hope that a strong selection at the first iteration and less constrained selection at the next iterations will improve the recognition results.

To build an initial acoustic model (called *ASR-AM0*), we selected randomly 100 hours because we do not have any information about the quality of available subtitles (in real life, only subtitles are available with no additional information, neither PMER nor AWD).

According to the algorithm of section 2.2.1, during the first iteration of data selection, we decode the whole training corpus with *ASR-AM0* with KATS system. We kept only segments whose AWD is inside $[0.16, 0.6]$. We excluded all other segments. We sorted the remaining segments according to PMER and we select $N=150$ hours. With these 150 hours, we trained *ASR-AM1*. For second iteration $N=200h$ (*ASR-AM2*) and for the last iteration $N=300h$ (*ASR-AM3*).

Table 4 presents the recognition results on the development set for each iteration of the data selection algorithm. Results are presented in terms of percent of correct recognition and in term of WER. The best results are highlighted in bold. Table shows that each data selection iteration improves the ASR system. The best result of 35% WER is obtained at the last iteration. Performing one more iteration gives the same result and is not presented in the table. Results presented in table 4 are not comparable with those in figure 3 because in figure 3, we used PMER given by the organizers.

ASR	#hours selected	PMER	Corr (%)	WER (%)
<i>ASR-AM0</i>	100	10	65.2	40.2
<i>ASR-AM1</i>	150	15	69.0	36.1
<i>ASR-AM2</i>	200	21	69.3	35.7
<i>ASR-AM3</i>	300	49	69.7	35.0

Table 4: WER on development set for different data selection iterations. Language model is estimated on text corpora of about 640 million words.

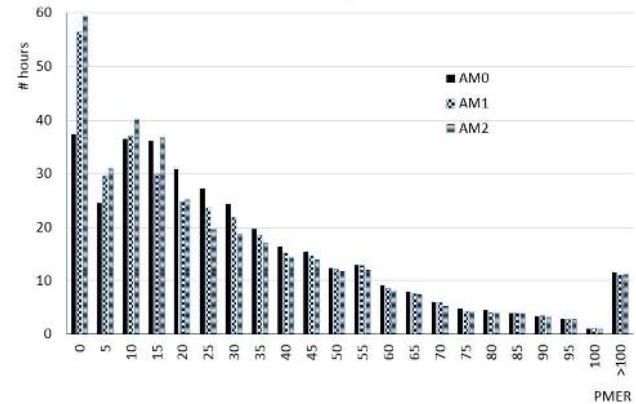


Figure 4: Number of hours of speech in function of PMER for the train set. For example, to PMER between 5 and 10 corresponds about 36 hours of speech.

Figure 4 gives more details about the training data distribution in function of PMER values and for different acoustic models. Firstly, this figure shows that a large amount of training data have a PMER below 15 and, so, have good quality subtitles. Secondly, PMER of 0 corresponds about 38 hours of speech at initial iteration, 57 hours for *ASR-AM1* and about 60 hours for *ASR-AM2*. This shows that from one iteration to another the acoustic model performance increases and the acoustic model choose better training data.

System combination

For system combination, we designed three different recognition systems. They share the same acoustic models (*ASR-AM3*), but the language models are different. For the first one only the subtitles of the training corpus are used to train the LM. This model is the most constrained. For the second one, the LM is trained using 640 million words of TV subtitles provided by the

organizers of the MGB challenge (least constrained model). The last one is a combination of the two previous language models. The $threshold_{PMER}$ was chosen experimentally and its value is 30.

Using these three ASRs for system combination, a relative improvement of 2% on WER was observed compared to *ASR-AM3* (cf. table 5). This improvement is significant. It could be interesting to combine systems using different acoustic models, for instance different acoustic features or different neural networks architecture (Long Short Term Memory, Highway networks).

ASR	#hours selected	Corr (%)	(Sub, Del, Ins) (%)	WER (%)
<i>ASR-AM3</i>	300	69.7	(14.3, 16.0, 4.7)	35.0
<i>System combination</i>	300	70.2	(13.8, 16.0, 4.5)	34.3

Table 5: WER on development set.

5. Conclusion

In this article, we explored different methods of data selection for building an automatic speech recognition system. The methods are inspired by lightly supervised technique. We studied data selection methods based on phone matched error rate and average word duration. Furthermore, we proposed a new data selection method combining three recognizers. The experiments are conducted on a TV broadcast corpus with subtitles. We have shown that selecting data is crucial for obtaining accurate acoustic models. We have studied the influence of PMER on data selection. The proposed system combination is beneficial to select better data and to obtain an efficient acoustic model.

6. Acknowledgements

This work is funded by the CPER LCNH project supported by the Great East region and by AMIS (Access Multilingual Information opinionS) Chist-Era project. Experiments presented in this paper were carried out using the Grid’5000 testbed, supported by a scientific interest group hosted by Inria, CNRS, RENATER and other Universities and organizations (<https://www.grid5000.fr>).

7. References

[1] Bell, P., Gales, MJF, Hain, T., Kilgour, J, Lanchantin, P., Liu, X., McParland, A., Renals, S., Saz, O., Wester, M., Woodland, PC. (2015). The MGB Challenge: Evaluating Multi-Genre Broadcast Media Recognition. In Proceedings of IEEE ASRU.

[2] Bengio, Y., Goodfellow, I., Courville, A. (2015). Deep Learning. Book in preparation for MIT Press.

[3] Chan, H. and Woodland, P. (2004). Improving broadcast news transcription by lightly supervised discriminative training. In 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing. ICASSP.

[4] Deng, L., Li, J., Huang, J.-T., Yao, K., Yu, D., Seide, F., Seltzer, M., Zweig, G., He, X., Williams, J., Gong, Y. and Acero A. (2013). Recent Advances in Deep Learning for Speech Research at Microsoft. Proceedings of ICASSP.

[5] Kemp, T. and Waibel, A. (1999). Unsupervised Training of a Speech Recognizer: Recent Experiments. In Proceedings of Eurospeech.

[6] Lamel, L., Gauvain, J.-L., and Adda, G. (2002). Lightly Super-vised and Unsupervised Acoustic Model Training. Journal Computer Speech and Language. 16:115-129.

[7] Lanchantin, P., Gales, M. J., Karanasou, P., Liu, X., Qian, Y., Wang, L., Woodland, P., and Zhang, C. (2016).

Selection of multi-genre broadcast data for the training of automatic speech recognition systems. In Proceedings of Interspeech.

[8] Levit, M., Huang, Y., Chang, S. and Gong Y. (2017). Don’t Count on ASR to Transcribe for You: Breaking Bias with Two Crowds. In Proceedings of Interspeech.

[9] Li, J., Seltzer, M., Wang, X., Zhao, R., Gong Y. (2017). Large-Scale Domain Adaptation via Teacher-Student Learning. . In Proceedings of Interspeech.

[10] Mathias, L., Yegnanarayanan, G., and Fritsch, J. (2005). Discriminative training of acoustic models applied to domains with unreliable transcripts. In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP.

[11] Peddinti, V., Povey, D., Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts Proceedings of Interspeech.

[12] Povey, Pocolm <https://github.com/danpovey/pocolm>

[13] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The kaldi speech recognition toolkit. In Proceedings of IEEE 2011 Workshop on Automatic Speech Recognition and Understanding (ASRU).

[14] Stolcke, A. (2002). SRILM - An Extensible Language Modeling Toolkit. Proceedings of ICSLP.

[15] Zavaliagos, G. and Colthurst, T. (1998). Utilizing Untranscribed Training Data to Improve Performance. DARPA Broadcast News Transcription and Understanding workshop.

[16] Woodland, P. C., Liu, X., Qian, Y., Zhang, C., Gales, M. J. F., Karanasou, P., Lanchantin, P., and Wang, L. (2015). Cambridge University Transcription Systems for the Multi-genre Broadcast Challenge. In IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU).

An analysis of psychoacoustically-inspired matching pursuit decompositions of speech signals

Khalid Daoudi¹, Nicolas Vinuesa²

¹INRIA Bordeaux-Sud Ouest, GeoStat team. France

²GIN UMR5296, CNRS-Université de Bordeaux. France

khalid.daoudi@inria.fr, vinuesa.nico@gmail.com

Abstract

Matching pursuit (MP), particularly using the Gammatones dictionary, has become a popular tool in sparse representations of speech/audio signals. The classical MP algorithm does not however take into account psychoacoustical aspects of the auditory system. For this reason two algorithms, called PAMP and PMP have been introduced in order to select only perceptually relevant atoms during MP decomposition. In this paper we compare this two algorithms on few speech sentences. The results suggest that PMP, which also has the strong advantage of including an implicit stop criterion, always outperforms PAMP as well as classical MP. We then raise the question of whether the Gammatones dictionary is the best choice when using PMP. We thus compare it to the popular Gabor and damped-Sinusoids dictionaries. The results suggest that Gammatones always outperform damped-Sinusoids, and that Gabor yield better reconstruction quality but with higher atoms rate.

Index Terms: Matching pursuit, Time-frequency decomposition, Sparse representation, Gammatones, Perceptual models.

1. Introduction

During the last two decades, the Matching pursuit (MP) algorithm [1] has been widely used as a powerful tool for sparse representation of signals using redundant dictionaries of time-frequency functions (atoms). MP is a greedy algorithm which iteratively approximates a signal $x(t)$ by a projecting it onto an overcomplete dictionary D of atoms ϕ_θ :

$$R_x^m(t) = \langle R_x^m(t), \phi_\theta \rangle \phi_\theta + R_x^{m+1}(t), \quad (1)$$

with $R_x^0(t) = x(t)$ at the first iteration $m = 0$. At each iteration m , a single atom ϕ_m is selected such that:

$$\phi_m = \arg \max_{\phi_\theta \in D} |\langle R_x^m(t), \phi_\theta \rangle| \quad (2)$$

where $\langle \cdot, \cdot \rangle$ is (generally) the Hermitian inner product. The signal $x(t)$ can be thus decomposed as:

$$x(t) = \sum_{m=1}^M \sum_{i=1}^{n_m} s_i^m \phi_m(t - \tau_i^m) + \epsilon(t), \quad (3)$$

where τ_i^m and s_i^m are the temporal position and weight of the i th instance of the atom ϕ_m , respectively. The notation n_m indicates the number of instances of ϕ_m , which need not to be the same across atoms, and M indicates the number of different atoms.

Recently a toolkit which efficiently implements the classical matching pursuit algorithm has been released: the Matching Pursuit ToolKit (MPTK) which is based on the work in [2]

and can be downloaded from <http://mptk.irisa.fr>. It can be installed on various platforms (Windows, Linux and Mac OSX) and is now massively used as it is the best available toolkit for (classical) MP analysis. MPTK provides fast implementation of different kind of dictionaries, including the Gabor dictionary.

In the field of speech/audio coding, it has been argued in [3, 4] that a relatively small dictionary of Gammatone atoms allow efficient coding of natural sounds using MP. The motivation behind this work is that early psychoacoustic experiments used Gammatone functions as a model of basilar membrane displacement [5] and where found to approximate cochlear responses of the cat [6]. Later it was stated in [7] that Gammatone functions also delineate the impulse response of human auditory filters. Real-valued Gammatone filters can be seen as gamma-modulated sinusoids and are defined as:

$$\gamma(t) = t^{n-1} e^{-2\pi b ERB(f_c) t} \cos(2\pi f_c t), \quad (4)$$

where f_c is the central frequency distributed on ERB (equivalent rectangular bandwidth) scales, n is the filter order, and b is a parameter that controls the bandwidth of the filter. $ERB(f_c) = 24.7 + 0.108 f_c$, $n = 4$ and $b = 1.019$ are commonly used as parameters.

On the other hand, classical MP (used in [3, 4]) focus on minimizing the energy of residuals at each iteration. Thus, it does not take into account psychoacoustical aspects of the auditory system which are crucial in any codec development. To address this issue, a psychoacoustically-adaptive inner product, considering frequency masking effects in sinusoidal decompositions, was presented in [8] and later refined with a perceptual model in [9]. The resulting algorithm is called PAMP. Later a new perceptual model, called PMP, was introduced in [10], taking into account both temporal and frequency masking effects. Similar to the perceptual model embedded in MPEG coders, the goal of psychoacoustically-based MP algorithms is to discard the perceptually irrelevant structures of the input signal and therefore increase the coding efficiency.

In this article we first experimentally compare these two algorithms when using a dictionary of Gammatone atoms. Our experiments suggest that PMP, which also has the strong advantage of including an implicit stop criterion, always outperforms PAMP as well as classical MP. We then raise the question of whether the Gammatones dictionary is the best choice when using PMP. We thus compare it to the popular Gabor and damped-sinusoids dictionaries. The results suggest that Gammatones always outperform damped-sinusoids, and that Gabor yield better reconstruction quality but with higher atoms rate.

The paper is organized as the following. In section 2, the two psychoacoustically-based MP algorithms are briefly described. Section 3 presents and analyzes the results of com-

parison between classical MP, PAMP and PMP. In section 4 we present an evaluation of PMP when using different dictionaries. Finally, we draw our conclusion and perspectives in section 5.

2. Psychoacoustically-inspired matching pursuits

In this section we give a short description of PAMP and PMP. Details about these algorithms can be found in [8, 9] and [11], respectively.

2.1. PAMP

PAMP [8, 9] relies on a perceptual model which predicts masked thresholds for sinusoidal distortions in audio coding. This model exploits the simultaneous masking effect (frequency masking) in order to determine what distortion level can be allowed such that it is perceptually not detectable. The model is based on a perceptual distortion measure [12] which estimates the probability that subjects can detect a distortion signal in the presence of a masking signal. This distortion measure defines a norm:

$$\|x\|^2 = \sum_f \hat{\alpha}(f) |\hat{x}(f)|^2 \quad (5)$$

where $\hat{x}(f)$ is the Fourier transform of the signal x and $\hat{\alpha}(f)$ is a real and positive weighting function representing the inverse of the masking curve for sinusoidal distortions. The norm is induced by the inner product:

$$\langle x, y \rangle = \sum_f \hat{\alpha}(f) \hat{x}(f) \hat{y}^*(f), \quad (6)$$

This inner product is then used in Eq. 2 instead of the classical Hermitian inner product in order to select the sinusoidal components of the signal in a MP decomposition with a dictionary of sinusoids. At the first iteration, i.e., when the residual equals the signal, $\hat{\alpha}(f)$ is set as the inverse of the threshold-in-quiet. Then, at each iteration $\hat{\alpha}(f)$ takes into account the sinusoidal distortion caused by the atom selected at the previous iteration.

2.2. PMP

PMP [11] relies on a perceptual model which take into account both temporal and frequency masking effects (as opposed to PAMP which considers frequency masking only). In PMP, a dictionary of Gammatone atoms is used and a masking pattern is created (and progressively updated) to determine a masking threshold at all time indexes and atom central frequencies.

At the first iteration, the masking pattern is set to the threshold-in-quiet, as in PAMP, for all time indexes. Then, all the inner products in Eq. 2 which are below the masking pattern are set to zero, meaning that projections that are below the threshold-in-quiet are ignored. Once the first atom has been selected, which will act as a masker, the masking pattern is elevated in a time interval around the atom temporal position and in the two adjacent critical bands. The updated masking pattern is then again used as a threshold, setting to zero all the inner products below it, thereby avoiding the search of atoms that would be masked by previously selected ones, i.e. perceptually irrelevant. This process is repeated until no inner product is above the masking threshold, meaning that there is no audible part left in the residual, and the algorithm stops. This implicit and perceptually-motivated stop criterion is a strong advantage over classical MP and PAMP.

3. Comparison between MP, PMP and PAMP

In this section, we experimentally compare the performance of the two psychoacoustically-based and the the classical matching pursuit algorithms. Since Gammatones have become popular waveforms in sparse speech/audio representations, we perform this comparison in the setting of MP using Gammatones dictionaries. The main idea is to analyse the behavior of MP when (Gammatone) atom selection is performed by the two perceptual models. We recall however that, while PMP has been introduced in this setting, PAMP have been developed in the framework of sinusoidal decompositions. By doing such a comparison, we are thus evaluating the behavior of PAMP when distortions are generated by Gammatone components.

We use four sentences from the TIMIT database [13] for the experiments. We selected these excerpts such that both speaker and phonetic variability is achieved: two male (sx54 and sx221) and two female (sx23 and sx136) speakers from different geographic regions are used in this study. The following speakers were used: mbma1 (sx54), fdw0 (sx23), fgcs0 (sx136) and mre0 (sx221). All files are sampled at 16 kHz with 16-bit quantization.

While signal to noise ratio (SNR) is a valid measure for *waveform* reconstructability, for audio coding problems this does not necessary reflect the perceived quality of the reconstruction. Therefore we use the well-known perceptual quality assessment measure PESQ [14] to estimate mean opinion scores (MOS). PESQ gives a continuous grading scale from 1 (very annoying) to 5 (no perceptual difference between original and reconstruction).

Since PMP is the only algorithm which implicitly has a perceptual stop criterion, we use the latter as an operating point to compare the 3 algorithms. We first run PMP and then compute the atoms rate per sample when it stops. Then, MP and PAMP are stopped when they reach this atoms rate. The experimental results of this process are shown in Table 1, for Gammatones dictionaries with 32, 64 and 128 atoms. A first observation is that PMP always yield a PESQ value above 3.5. Moreover, our listening tests confirm that the reconstruction quality is good without being perceptually transparent. This shows that the perceptual model of PMP achieves the desired goal. A second observation is that PMP always slightly outperforms MP and significantly outperforms PAMP. Finally, these results suggest that a good choice for the number of Gammatones in the dictionary is 64.

Figure 1 displays the evolution of the 3 algorithms, iteration after iteration, until they they reach the atoms rate given by PMP. The figure corresponds to only one sentence, but the behavior is very similar for the 4 sentences. Because the masking pattern is updated with the masking effect caused by each new atom, PMP behaves exactly like MP until most of the masker atoms have been extracted; only then (around atoms rate of 0.05) the newly selected atoms are perceptually relevant and the difference can be appreciated. From this rate, PMP starts selecting atoms which are perceptually relevant and thus yields higher PESQ values, while MP selects atoms which minimize the residual energy and thus yields higher SNR values. The weak performances of PAMP are most probably due to the fact that distortions generated by Gammatone decompositions do not satisfy the hypothesis made on distortions obtained in sinusoidal modeling.

In Table 2, we provide the atoms rate required by MP and PAMP to achieve the same PESQ value as PMP (at stop-

File	Dictionary	PESQ-PMP	PESQ-MP	PESQ-PAMP	Atoms Rate
sx54	32 Gammatones	3.66	3.56	3.17	0.09
	64 Gammatones	3.70	3.61	3.22	0.08
	128 Gammatones	3.70	3.61	3.21	0.08
sx23	32 Gammatones	3.77	3.45	3.07	0.15
	64 Gammatones	3.84	3.62	3.08	0.14
	128 Gammatones	3.85	3.67	3.15	0.14
sx136	32 Gammatones	3.60	3.43	2.98	0.09
	64 Gammatones	3.64	3.42	3.05	0.08
	128 Gammatones	3.62	3.47	3.08	0.08
sx221	32 Gammatones	3.55	3.41	3.19	0.09
	64 Gammatones	3.58	3.47	3.33	0.08
	128 Gammatones	3.59	3.49	3.35	0.08

Table 1: PESQ and atoms rate using Gammatone dictionary.

ping atoms rate), for 64 Gammatones. It is clear that PMP exhibits the highest efficiency among the three algorithms, as MP requires up to 40% and PAMP up to 80% more atoms to achieve the same perceived quality. All these experimental results suggest that PMP is a very good algorithm for efficient and perceptually-consistent sparse speech representations and coding.

File	Atoms rate PMP	Atoms rate MP	Atoms rate PAMP
sx54	0.08	0.11	0.13
sx23	0.14	0.17	0.23
sx136	0.08	0.10	0.15
sx221	0.08	0.10	0.11

Table 2: Atoms rate required by MP and PAMP to reach the same PESQ value as PMP.

4. Comparison of different dictionaries using PMP

Given the results of the previous section which are in favor of PMP, we now focus on the latter and raise the following question: is the Gammatones dictionary the best choice when using PMP? This question has been indeed central in classical matching pursuit. The original MP algorithm [1] used the Gabor dictionary defined as:

$$g_{\theta}(t) = K_{\theta} e^{-\pi \left(\frac{t-\tau}{s}\right)^2} e^{i\omega(t-\tau)}, \quad (7)$$

for index $\theta = \{s, \tau, \omega\}$ where s is the scale, τ the time translation, ω the frequency modulation, and K_{θ} such that $\|g_{\theta}\| = 1$.

Probably the most known work on this matter is [15], where the authors argued that a dictionary which consists only of atoms that exhibit symmetric time-domain behavior are not well suited for modeling asymmetric events such as transients in audio signals. They proposed the use of structured overcomplete dictionaries of damped sinusoids (DS) defined as:

$$d_{\theta}(t) = K_{\theta} \lambda^{(t-\tau)} e^{i\omega(t-\tau)} u(t-\tau), \quad (8)$$

File	Dictionary	PESQ-PMP	Atoms rate
sx54	64 Gammatones	3.69	0.08
	64 Gabor	3.85	0.12
	64 D-Sinusoids	3.55	0.08
sx23	64 Gammatones	3.84	0.14
	64 Gabor	4.04	0.20
	64 D-Sinusoids	3.54	0.15
sx136	64 Gammatones	3.64	0.08
	64 Gabor	3.87	0.12
	64 D-Sinusoids	3.39	0.08
sx221	64 Gammatones	3.58	0.08
	64 Gabor	3.85	0.12
	64 D-Sinusoids	3.34	0.09

Table 3: PESQ and atoms rate using different dictionaries.

for index $\theta = \{\lambda, \tau, \omega\}$ where λ is the damping factor, τ the time translation, ω the frequency modulation, K_{θ} such that $\|d_{\theta}\| = 1$ and $u(t-\tau)$ being the step function.

They showed, in the context of classical MP, that DS are more suited for modeling signals with transient behavior than symmetric Gabor atoms. The work in [16] proposed a comparison of Gabor atoms, complex exponentials and "Fonction d'onde Formantique". The authors argued that the Gabor dictionary performs sufficiently well.

This motivates us to analyse the behavior of PMP when using different dictionaries than Gammatones, within the ERB scale. We thus propose in this section a comparison between Gammatones, damped sinusoids and Gabor atoms.

Table 3 shows the results obtained using 64 atoms per dictionary, within the ERB scale. A first observation is that Gammatones and DS stop at almost the same atoms rate, but Gammatones dictionary outperforms DS. The most important observation is that the Gabor dictionary achieves higher PESQ values than the other dictionaries, but the atoms rate is also considerably higher. If we rely on atom rates as a measure of coding

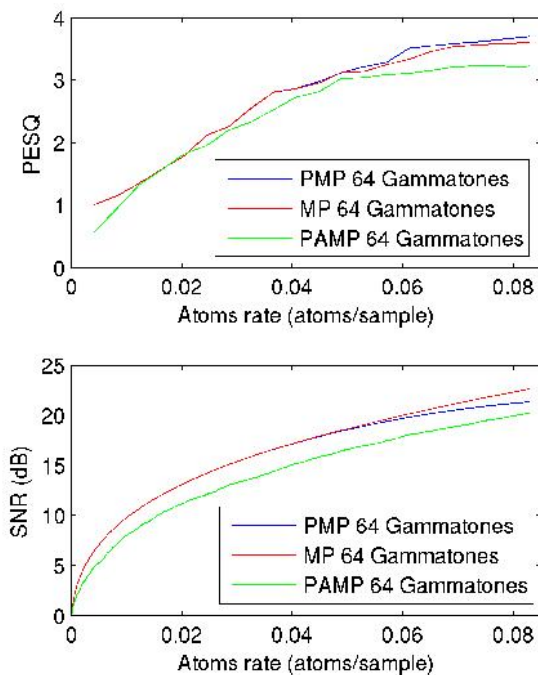


Figure 1: Comparison of PESQ and SNR vs. atoms rate for sentence sx54 and 64 Gammatones.

efficiency, we may consider that the Gammatones dictionary is the best choice, given that PMP requires about 50% more Gabor atoms to achieve a relatively smaller gain in PESQ. However, the best way to assess coding efficiency is to use bits per second, and the best way to assess perceptual quality is MOS. Moreover, the ERB scale may not be the optimal choice for Gabor and DS atoms. Finally, we used only 4 sentences in our experiments. A much larger and richer database should be used in order to have a good evaluation. Thus, all these factors (at least) should be taken into account before drawing final conclusions. The question we raised is then still open, but we may still argue that PMP with Gammatones presents a promising potential.

5. Conclusion

In this paper, we first presented an experimental comparison of two psychoacoustically-based matching pursuit algorithms (PMP and PAMP) as well as the classical MP algorithm. The results suggest that PMP always outperforms both MP as PAMP in term of sparsity and perceived reconstruction quality. In a second experiment, we compared different dictionaries (Gammatones, Gabor and damped-sinusoids) using PMP. The results suggest that Gammatones is the best choice if atoms rate is considered as a measure for coding efficiency. All these results suggest that PMP is a very good algorithm for efficient and perceptually-consistent sparse speech representations and coding. However, further work is required in refining the perceptual model in PAMP in order to take into account the distortions generated by Gammatone decompositions more accurately. A more in-depth study of the coding efficiency, using bits per second instead of atoms rate, is also necessary. We believe indeed that latter would allow to mitigate the results of [3, 4]. This

will be the purpose of a future work (also using the full TIMIT database in the experiments).

6. Acknowledgments

The authors would like to thank Ramin Pichevar and Hossein Najaf-Zadeh for the help provided in the implementation and fruitful discussions over PMP.

7. References

- [1] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3397–3415, 1993.
- [2] S. Krstulovic and R. Gribonval, "MPTK: Matching Pursuit made tractable," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP'06)*, vol. 3, Toulouse, France, May 2006, pp. III-496 – III-499.
- [3] E. Smith and M. S. Lewicki, "Efficient coding of time-relative structure using spikes," *Neural Computation*, vol. 17, no. 1, pp. 19–45, 2005.
- [4] E. C. Smith and M. S. Lewicki, "Efficient auditory coding," *Nature*, vol. 439, no. 7079, pp. 978–982, 2006.
- [5] J. L. Flanagan, "Models for approximating basilar membrane displacement," *The Journal of the Acoustical Society of America*, vol. 32, no. 7, pp. 937–937, 1960.
- [6] P. I. Johannesma, "The pre-response stimulus ensemble of neurons in the cochlear nucleus," in *Proceedings of the Symposium of Hearing Theory*, 1972.
- [7] R. Patterson, I. Nimmo-Smith, J. Holdsworth, and P. Rice, "An efficient auditory filterbank based on the gammatone function," *APU report*, vol. 2341, 1988.
- [8] R. Heusdens, R. Vafin, and W. B. Kleijn, "Sinusoidal modeling using psychoacoustic-adaptive matching pursuits," *Signal Processing Letters, IEEE*, vol. 9, no. 8, pp. 262–265, 2002.
- [9] S. van de Par, A. Kohlrausch, R. Heusdens, J. Jensen, and S. H. Jensen, "A perceptual model for sinusoidal audio coding based on spectral integration," *EURASIP Journal on Applied Signal Processing*, vol. 2005, pp. 1292–1304, 2005.
- [10] H. Najaf-Zadeh, R. Pichevar, L. Thibault, and H. Lahdili, "Perceptual matching pursuit for audio coding," *Audio Engineering Society Convention, Amsterdam, The Netherlands*, 2008.
- [11] R. Pichevar, H. Najaf-Zadeh, L. Thibault, and H. Lahdili, "Auditory-inspired sparse representation of audio signals," *Speech Communication*, vol. 53, no. 5, pp. 643–657, 2011.
- [12] S. Van De Par, A. Kohlrausch, G. Charestan, and R. Heusdens, "A new psychoacoustical masking model for audio coding applications," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, vol. 2. IEEE, 2002, pp. II-1805.
- [13] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, "DARPA TIMIT acoustic-phonetic continuous speech corpus," U.S. Dept. of Commerce, NIST, Gaithersburg, MD, Tech. Rep., 1993.
- [14] "Perceptual evaluation of speech quality (pesq), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs ITU-T Rec. P.862," 2001.
- [15] M. M. Goodwin and M. Vetterli, "Matching pursuit and atomic signal models based on recursive filter banks," *Signal Processing, IEEE Transactions on*, vol. 47, no. 7, pp. 1890–1902, 1999.
- [16] B. L. Sturm and J. D. Gibson, "Matching pursuit decompositions of non-noisy speech signals using several dictionaries," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 3. IEEE, 2006, pp. III-III.

Enthusiasm and disillusionment with voice assistants.

Franck Poirier

Lab-STICC, Université Bretagne Sud, France

Franck.Poirier@univ-ubs.fr

Abstract

An important choice in mobile interactive system design is to decide the best means for inputting information. Recently, with the advent of connected objects, virtual keyboards have been given up, and, for communicating, the user is encouraged to use a personal voice assistant. With the keyboard, the communication mode is textual, but with the voice assistant, communication mode becomes oral. This article attempts to study the consequences, disadvantages and limitations of oral interaction with mobile and wearable devices. The main conclusion is that if text input with keyboards is not suitable for tiny wearable devices, voice personal assistants can correctly handle only simple and well formulated requests and don't allow a natural oral communication between the user and the system. Basically, voice interaction on mobile devices changes the nature of the communication and gives the illusion that the system has abilities of comprehension that it does not have, which causes the disillusionment of the user.

Index Terms: Intelligent personal assistant, voice activated personal assistant, interaction modes, oral communication, dialog.

1. Introduction

In 2017, voice input is developing rapidly, especially in North America. Forbes said that the year 2017 was the year of vocal search. Two billion voice queries are handling by Siri per week. In North America, in 2017, 50 percent of mobile queries are voice searches [10]. Every month new voice assistants are announced. Now there are lots of voice assistants on the market (figure 1): Alexa from Amazon, Siri from Apple, Little Fish from Baidu, Google Assistant, LingLong DingDong from iFlytek, Cortana from Microsoft, Bixby from Samsung....

The user is strongly encouraged to use voice interaction on all their devices: smartphone, tablet and even PC. On wearable devices, sometimes the user has no choice, he/she can only use the voice assistant. That is the case, for instance, for smartwatches; Apple, Samsung, LG and the other producers gave up trying to integrate a keyboard on their watches.

Faced with the development of voice interaction, it is important to study the consequences, disadvantages and limitations of this new form of interaction with our mobile devices.

2. Vocabulary issue

There are many expressions which refer to voice assistants: smart voice assistant, smart speaker, smart home speaker, smart-speaker personal assistants, personal voice assistant, voice-controlled virtual assistant, voice-based digital assistant, voice activated personal assistant (VAPA). The last expression is more precise, but for reason of simplicity, "voice assistant" will be used in this article.



Figure 1: *Voice assistants at home: Amazon Echo (above), Google Assistant (below).*

3. Are words still useful for communicating with mobile devices?

Text communication has always been a basic service on personal assistants (PDA) first and then on smartphones. In more than 20 years, hundreds of reduced keyboards have been designed for text input on these devices. At the same time, the size of the screens has increased from less than 3 inches to more than 6 inches, which allowed to integrate a full qwerty keyboard in the current devices. These relatively large keyboards provide good quality lexical prediction and spell checking that make text entry relatively easy, so keyboards designed specifically for mobile devices are now rarely used [13].

Text input on mobile devices is indispensable for lots of tasks [1] like searching on the internet or on the device, dialing a phone number, sending a message (SMS, instant message, e-mail...), setting an alarm, setting a reminder, adding an item to a list, playing a music track, launching a video... Thus, text input by means of keyboarding is unavoidable on the pocket assistants.

The question that arises is which means of communication are most appropriate with the new wearable devices, in

particular with smartwatches and all devices that have a very tiny screen (less than 3 inches) or no screen at all.

What can be said is that in order to offer a real communication service, especially between humans, the future devices will have to be able to enter sentences, either by means of a keyboard or orally. In other words, the future devices that will replace or will be partners of the smartphone like the smartwatch, will have to manage more or less the language.

Words will still be needed in mediated communication between humans as well as in communication with the digital system... if only to do a search on the internet.



Figure 2: Voice assistants on smartphone: Microsoft Cortana (above), Apple Siri (middle), Samsung Bixby (below).

4. Keyboard or voice assistant?

To choose between keyboard and voice assistant, the form factor is decisive because the difficulty of entering information depends directly on the size of the interactive object. The smaller the object, the more difficult or impossible it is.

Connected objects for which text entry is truly difficult are wearable devices that mainly correspond to smartwatches, connected wristbands for physical training and accessories for health or well-being. For these devices, the screen size varies from less than one inch (2.5 cm) for activity trackers to 1.6 inches (4 cm diagonal or 4.5 cm in diameter) for biggest smartwatches. It should be noted that the largest watch screens are still 8 to 10 times smaller than those of an iPhone 7 Plus or Samsung Galaxy S8 + and include respectively 20 to 40 times less pixels.

With these tiny devices, a keyboard is virtually unusable, which is why manufacturers (Amazon, Apple, Huawei, LG, Samsung...) are currently choosing to integrate voice assistants into all these devices.

5. Speech interface is coming back

Speech interface is not new. Speech recognition research began in the 1950s with the early speech recognizers of the Bell Labs, and then developed strongly in the 1970s in the dynamics of DARPA projects.

Beginning in the 1990s, several large vocabulary continuous speech recognition systems (LVCSR systems) have been put on the market. More recently, the availability of very large learning data, improved learning algorithms (deep learning based on Convolutional Neural Networks) and artificial intelligence techniques have led to the development of Intelligent Personal Assistant (IPA), also known as VAPAs, actually usable and quite fascinating.

The most used VAPAs are Alexa from Amazon, Siri from Apple, Google Assistant, Cortana from Microsoft, Bixby from Samsung.

The real beginning of the use of these voice assistants dates from 2017 while Google Voice Search dates back to 2002 and Siri was bought by Apple in 2010. Voice assistants are mostly used in North America. They still only work in a limited number of languages like English, Chinese, French, German..., but Siri is now in 21 languages. In a nutshell, voice input is experiencing a new youth with the rise of connected devices.

ComScore forecasts that half of all searches will be voice searches by 2020 [10]. By 2021, Ovum [14] predicts that there will be more voice assistants than humans on the planet.

6. What are the users' current preferences?

Currently, the users prefer the keyboard rather than the assistant to enter information on his/her smartphone [1].

The voice assistant is rather used in private places, in the absence of others and for non-sensitive information [3]. The user wants to be sure that nobody heard what he/she said and also that the information is not analyzed or stored on the servers used by the voice assistant.

A survey conducted on 2,000 US and British respondents [14] showed that 50% of consumers currently consider voice

assistants "not useful". Many of them (around 20% in the UK) are also unaware of what these new devices can do.

In 2017, the most frequent user request for Google Assistant asked to Google is to be able to communicate with the personal assistant by means of the keyboard and not by voice. At the Google I/O 2017 conference, this new functionality was announced and this functionality is now available (figure 3).

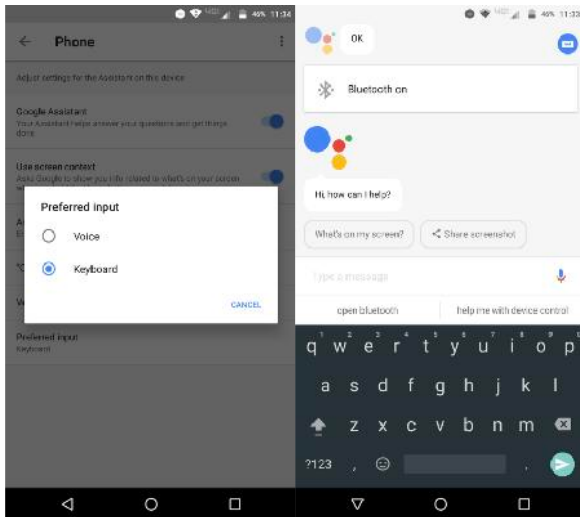


Figure 3: *Input preferences for Google Assistant.*

7. What are the difficulties for the user of voice interaction with a personal assistant?

At first glance, voice communication seems to be an ideal solution that simplifies the interaction between the user and the digital device. Manufacturers and marketers claim that with voice interaction the use of devices becomes intuitive. The reality is quite different. It is not at all intuitive because the user does not usually speak to a machine.

Over time, all the speech recognizer systems have been greatly improved, they recognize all users, in continuous speech, without learning and on broad vocabularies. On correctly pronounced and syntactically correct sentences, the level of recognition is excellent. Unfortunately, very often, that is not sufficient to understand what the user wants. Speech recognition and language processing researchers are well aware that oral communication are always problematic.

The user encounters difficulties that are specific to personal assistants. There are two main problems: the first is to know what to tell the assistant and the second one is to know how to tell it.

7.1. I don't know what to tell ya?

As Heidegger wrote [5] "Man speaks... We are continually speaking in one way or another. We speak because speaking is natural to us... Man is said to have language by nature... It is as one who speaks that man is-man". So, speaking is natural for human, but it is unnatural to speak to machine, more precisely, for the user, it is not common to talk to a machine. This is why users do not know what to say in front of a personal assistant.

In other words, users cannot imagine the mental model associated with the voice assistant.

In order to solve this problem, most assistants who have a screen display a message of encouragement and sometimes some examples of possible sentences. On figure 2, Cortana and Siri display the sentence "What can I help you with?", Bixby displays "Okay, I'm listening". Cortana suggests "Is it cold in Moscow right now?" and Bixby proposes a set of possible sentences like "Call mom", "Open YouTube my channel", "Wake me up tomorrow"...

7.2. I don't know how to tell it?

Human speaks well because he has learned how to speak for years and human has learned to speak only to other humans. They understand each other, often with difficulty, because they share a common linguistic code. Humans understand each other by exchanging phrases that are poorly formulated, stammering, incomplete or even semantically incorrect. These are the characteristics of spontaneous speech, of ordinary language.

That is not the same when talking to the machine, with your assistant, you must not stammer or interrupt yourself and you have to produce a well-formulated sentence.

As it is very different from a spontaneous communication between humans, it is not natural. So that is neither easy, intuitive nor effective.

A further complication is that the syntax is different for each assistant. For example, if you want to know the schedules of the planes from Nantes to Marrakech, you must say "flight Nantes Marrakech" with Google assistant, but "are there flights from Nantes to Marrakech" with Siri. However, if you ask "are there flights tomorrow for Marrakech from Nantes airport", neither Siri nor Google assistant will understand the request.

In fact, the voice assistant offers the user a conversational interface. It's not new at all, it was the first type of interaction with computer before the graphical user interface (GUI). It is a kind of vocal command line interaction (CLI). The difference is that with CLI, you type the request and with VAPAs, you tell what to do. With voice assistant you have no longer needs to go through a window, a menu or an icon... but you must know exactly what to tell and how to tell it.

Knowing what to tell and how to tell it is cognitively difficult because it involves a recall process. Face to the voice assistant, the user has no cues to help him/her. Conversely, face to a GUI, the user recognizes the available commands in a menu and can easily select the one he/she wants. It is well known [15] that recognition is preferable to recall in user interfaces. That is why, voice assistant is not so easy to use.

8. What does it change to talk to the machine?

The question is whether the form and content of exchanges between the machine and the user are linked to the means of input. Research has shown that the answer to this question is positive [6], [11], [12].

Inevitably when a user uses a voice assistant, she/he finds it magical. The system seems to have the super-power to understand the thoughts and satisfy all the desires of the user [2]. For the user, talking to a machine makes this machine more understanding, even more smart. Obviously, this is not

the case. The anthropomorphization of the system is misleading [4], [8]. The consequence is that the user initially seduced by the voice interaction is quickly disappointed. The disillusion comes quickly enough when the user realizes that the most common answer is " Sorry, I don't have the answer to that question" (known as the most frequent answer of Alexa !).

Voice interaction necessarily generates uncertainty, and for Cooper [2], whatever the progress of language processing, this uncertainty will persist forever.

9. A pronounced lack of subtlety?

It would be wrong to say that the assistants understand nothing. In fact, they understand very well the simple requests like "call my wife", "play some gnaoua music", "how are Alphabet's stocks doing?", "how's the weather today?", "add coke to my shopping list", "find me flights with Royal Air Maroc", "turn the TV to 2M", "what's the meaning of life"... Usually, as long as the request is simple, that is to say that only one thing is asked at a time, the assistant gives a satisfactory answer.

The problem is that oral communication between humans is totally different. Usually, human communication is not limited to one speaking-turn, voice assistant are able to answer to a simple question but not to manage a dialog with alternating turns. Additionally, oral sentences are not limited to a single subject, they are composed of several semantic units. A good example is "are there flights not too expensive to Marrakech without passing by Casa which leave not too early in the week". The assistant will certainly propose lots of flights to Marrakech but they will not satisfy the different criteria of the request.

In the same way, the voice assistant can't differentiate between these two requests "what is the number of calories in a pizza Margherita?" and "What is the number of calories in a slice of pizza Margherita?". It will erroneously give the same answer. Voice assistant are not able to manage the "subtle distinctions" that are specific to the ordinary language.

To conclude, let us say that it is not enough to understand all the words of the sentence to give a correct answer. According to Oswald Ducrot, "understand an utterance is to understand the reasons for its enunciation" [7]. To use the words of Moeschler and Reboul, the voice assistant understands the sentence but not the utterance which should be regarded as the sentence in its context. In other words, the main problem with voice assistants is that nowadays they are not able to make a pragmatic analysis of the statements. According to Norman [9] and Cooper [2], they are not ready to achieve this task, they should not and can not do.

10. Conclusion

What comes as a conclusion is that voice interaction on mobile devices changes the nature of the communication and gives the illusion that the system has abilities of comprehension that it does not have.

With voice assistants, the user encounters two problems: first, knowing what to say because it is unfamiliar to talk to a machine and secondly knowing how to say it because she/he is constrained to follow a relatively rigid syntax which is unnatural.

Finally, whatever the progress in automatic language processing (speech recognition performance is already excellent), that is a chimera to believe that the so-called "intelligent" voice assistants will allow a natural oral communication, a real dialog between the user and the systems.

If you want to know more about voice assistants, ask them directly how they imagine their future!

11. References

- [1] Cherian, E. and Pounder, J., "A global trends and insight report on voice technology and its impact on brands", J. Walter Thompson Group, report, 2017.
- [2] Cooper, A., "Alexa, Please Kill Me Now - Thoughts on Conversational UI", Plateforme en ligne de publication Medium, 12 juin 2017.
- [3] Easwara Moorthy, A. and Kim-Phuong, L. Vu, "Privacy Concerns for Use of Voice Activated Personal Assistant in the Public Space", Intl. Journal of Human-Computer Interaction, 31: 307-335, 2015.
- [4] Guthrie, S. E., "Anthropomorphism: A definition and a theory", in Mitchell R., Thompson N. et Miles, H. (eds) Anthropomorphism, anecdotes, and animals. State University of New York Press, 50-58, 1997.
- [5] Heidegger, M., "Poetry, Language, Thought", New York: Harper & Row, p. 187, 1971.
- [6] Luzzati, D., "Le dialogue verbal homme-machine , étude de cas", Masson, 1995.
- [7] Moeschler, J. and Reboul, A., "Dictionnaire encyclopédique de pragmatique", Seuil. 1994.
- [8] Ngadi Essame. V., "La personnalité du design : concept, structure et diffusion. Enquête menée dans un contexte italien". Thèse de l'université Paris 13, 2014.
- [9] Norman, D. A., "Cautious cars and cantankerous kitchens: How machines take control", The Design of Future Things, Basic Books, 2009.
- [10] Olson, C., "Just Say It: The Future of Search is Voice and Personal Digital Assistants", Campaign, 25 April 2016.
- [11] Perea, F., "Nature et technologie langagière dans les dialogues oraux homme-machine". Communication, vol.34/1, 2016.
- [12] Poirier, F., "Vers une communication naturelle homme-machine". Habilitation à diriger les recherches, université Paris Sud 11, 1994.
- [13] Poirier, F., "Quelle modalité pour l'interaction avec les petits appareils mobiles et vestimentaires : texte ou vocal ? Comment choisir entre clavier et assistant personnel ?", ACM IHM 2017, 2017.
- [14] de Renesse, R., "Digital Assistant and Voice AI-Capable Device Forecast: 2016-21", Ovum survey, 2017.
- [15] B. Shneiderman. Designing the User Interface: Strategies for Effective Human-Computer Interaction. Pearson, 2009.

Is statistical machine translation approach dead?

M.A. Menacer, D. Langlois, O. Mella, D. Fohr, D. Jouvét and K. Smaili

LORIA, Campus Scientifique, BP 239, 54506 Vandoeuvre-lès-Nancy, France

{mohamed-amine.menacer, odile.mella, dominique.fohr, denis.jouvet
david.langlois, kamel.smaili}@loria.fr

Abstract

Statistical phrase-based approach was dominating researches in the field of machine translation for these last twenty years. Recently, a new paradigm based on neural networks has been proposed: Neural Machine Translation (NMT). Even there is still challenges to deal with, NMT shows up promising results better than the Statistical Machine Translation (SMT) on some language pairs. The baseline architecture used in NMT systems is based on a large and a single neural network to translate a whole source sentence to a target one. Several powerful and advanced techniques have been proposed to improve this baseline system and achieve a performance comparable to the state-of-the-art approach. This article aims to describe some of these techniques and to compare them with the conventional SMT approach on the task of Arabic-English machine translation. The result obtained by the NMT system is close to the one obtained by the SMT system on our data set.

Index Terms: Machine translation, Neural network, Phrase-based machine translation, Neural machine translation.

1. Introduction

In machine translation several approaches were proposed. Some of them are based on dictionaries [1], others on examples [2], rules [3] or statistical approaches [4]. The widely used technique for these last twenty years is phrase-based statistical machine translation [5]. The main principle of this approach is the use of two components: translation and language models to maximize the likelihood of translating a source sentence f to a target sentence e . The translation model, which is estimated from a parallel corpus, expresses how well the sentence e is an appropriate translation for the source sentence f . The language model is learned by using a monolingual corpus in order to measure how likely the proposed target sentence e is.

Recently deep learning became a powerful technique that is widely used to achieve good performance on difficult problems such as automatic speech recognition, visual object recognition, sentiment analysis, etc. These methods have been known in Natural Language Processing (NLP) topics and others for several decades, but the bottleneck was the lack of data and the power limitation of computers.

Having regard to these factors, it's not surprising that deep learning recently kicked up a storm in translation to create a promising approach so-called Neural Machine Translation (NMT) [6, 7, 8].

Unlike the old paradigm of SMT where one should explicitly model latent structures, namely: word alignment, phrase segmentation, phrase reordering and language modeling, the new paradigm NMT is end-to-end model [9]. It aims to directly transform a source sentence into a target one by training a single and a large neural network.

Current NMT models are essentially based on the *encoder-decoder* framework [7], where the source language sentence is encoded into a fixed length vector, from which the target language sentence is decoded (generated). This basic idea has been improved to achieve results comparable to those obtained by the state-of-the-art approach. To do so, different techniques called *attention* [10] have been proposed.

Recently, [11, 12, 13] perform a detailed analysis of NMT vs. SMT in order to explore the challenges with the new paradigm and understand what linguistic phenomena are best modeled by neural models. In this article, we focus on some advanced techniques that improve neural machine translation systems. These techniques are described and compared with the conventional SMT approach. Several experiments are carried out, in this article, on the task of Arabic-English translation by using small data from UN and they are compared, on the same data set, with SMT.

In the next section, an overview about the conventional SMT and the NMT approaches is reported. Section 3 summarizes the data set used in the different experiments and discusses the translation quality achieved by using several advanced techniques.

2. NMT vs. SMT

From a probabilistic standpoint, the task of machine translation consists of finding a target sentence $\hat{E} = e_1 e_2 \dots e_{|E|}$ that maximizes the probability $P(E|F, \theta)$ of producing E given the source sentence $F = f_1 f_2 \dots f_{|F|}$ and the model parameters θ , as in Equation 1.

$$\hat{E} = \arg \max_E P(E|F, \theta) \quad (1)$$

The parameters θ are learned from parallel corpora, a set of aligned sentences in the source and the target languages.

In the conventional SMT approach, the probability $P(E|F)$ is decomposed into two knowledge sources by applying Bayes theorem [5]:

$$\hat{E} = \arg \max_E P(E)P(F|E) \quad (2)$$

Where the probability $P(E)$ represents the language model and $P(F|E)$ is the translation model. In practice, other models are combined with these two knowledge sources in order to calculate the cost assigned to a translation, namely the reordering model and the word penalty. This combination is performed by using the log-linear approach [5, 14] as it is shown in Equation 3.

$$\log P(E|F) = \sum_{i=1}^{|w|} w_i \times \log(p_i(E, F)) \quad (3)$$

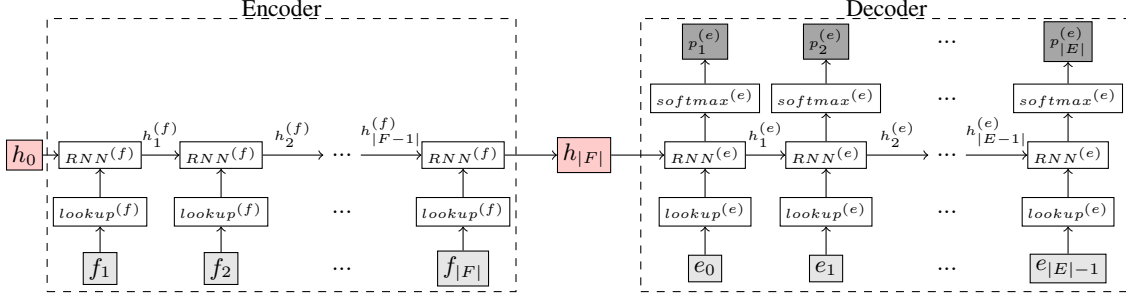


Figure 1: The architecture of the *encoder-decoder* framework.

The probability $P(E|F)$ is broken up into multiple models (language model, translation model, reordering model...); the score of each model $p_i(E, F)$ is weighted by a weight w_i and $|w|$ is the number of models.

From this standpoint, the SMT approach requires the integration of multiple components and processing steps in order to find the best translation. This leads to a complex architecture compared to the new paradigm of translation based on neural networks.

NMT aims to train one single and large neural network in order to translate a whole source sentence into a target one, which means that all models used in the conventional SMT approach are implicitly modeled by the neural network. This idea is carried out by using a sequence to sequence mapping models [7]. These models are based on the *encoder-decoder* framework where the *encoder* network takes as input a word sequence and maps it to an encoded representation, which is then used by the *decoder* network to generate an output word sequence. Each component is based on the use of one or multiple hidden layers of Recurrent Neural Networks (RNNs) [15] or Long Short-Term Memory (LSTM) networks [16].

To have a good understanding, Figure 1 illustrates the *encoder-decoder* architecture used in NMT with simple RNN. The *encoder* looks up the word representation/embedding for each word in F and calculates the output of the hidden state h_t according to Equation 4.

$$h_t = \begin{cases} \tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h), & \text{if } t \geq 1 \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Where W_{xh} , W_{hh} and b_h are the weights of the recurrent neural network to learn during the training stage. x_t is the vector representation of the word f_t and h_{t-1} is the output of the RNN.

Then, the *decoder* performs the treatment, in the same manner, as for the *encoder*, but with the target sentence E . The difference between the two procedures is the initial vector h_0 . This vector is initialized to zeros in the *encoder* ($h_0^{(f)} = 0$), however, in the *decoder*, it is initialized to the output of the final state of the *encoder* ($h_0^{(e)} = h_{|F|}^{(f)}$), which makes E depending on F . This vector is called the context vector (c) [10]. Furthermore, the first word in the target sequence refers to the symbol $\langle s \rangle$ indicating the sentence start. The final output of the *decoder* is the probability of translating F to E . More precisely, the likelihood of $E = (e_1, e_2, \dots, e_{|E|})$ is get by multiplying the probability of each word giving the context vector c and all the previous words $\{e_1, \dots, e_{t-1}\}$ as it is shown in Equation 5.

$$P(E) = \prod_{t=1}^{|E|} p(e_t|\{e_1, \dots, e_{t-1}\}, c). \quad (5)$$

Each conditional probability is modeled as:

$$p(e_t|\{e_1, \dots, e_{t-1}\}, c) = NN_output(e_{t-1}, h_t^{(e)}, c). \quad (6)$$

where NN_output is a function that generates the probability of e_t , while $h_t^{(e)}$ is the output of the hidden state of the RNN.

3. Experimental setup

In the first instance, we decided to train a NMT baseline system [17, 7] and compare the results with a conventional SMT system. The architecture used in this baseline system is quite simple, there are several techniques that are proposed in order to improve the quality of translation. Some of these techniques are described and tested to improve the results and achieve performance close to the SMT system.

3.1. Data set

All experiments are carried out on the task of Arabic-English translation. For training, we used a small corpus of 100K parallel sentences extracted from Multiun [18]. This corpus is a part of the official documents of the United Nations between 2000 and 2010. In order to prevent the over-fitting issue, i.e. the neural network models the training data too well, we used a data set of 1000 parallel sentences (Dev). Likewise, for testing the performance of our systems, another corpus of 1000 parallel sentences is used (Test). These two corpora are also extracted from Multiun [18].

All data sets are used after applying the normalization process proposed in [19]. Also, in order to handle unknown words that do not exist in the training data, all words that appear only once in the training corpus are replaced by a special token $\langle unk \rangle$. This process leads to a vocabulary of size 47K words for the source language and 20K words for the target language.

3.2. Experiment results

3.2.1. Baseline system

As it was mentioned before, the baseline system is a basic *encoder-decoder* model without any improving mechanism as it is proposed in [17, 7]. For this system, the architecture used is as follows: one hidden layer, RNN blocs for both the *encoder* and the *decoder* and each bloc has 100 hidden units. Furthermore, several optimization algorithms were tested in order to update the model parameters, namely:

- Stochastic Gradient Descent (SGD): this technique is based on the calculation of the derivative of the loss with respect to each parameter. Afterwards, this derivative is used to take a step in the direction that will reduce the loss according to the objective function. During this step, the weights are updated according a *learning rate*.
- Momentum [20]: SGD with momentum is used to help the optimizer to explore more efficiently the parameter space. It does that by keeping a fraction of the past gradient and add it to the current gradient. It ensures a faster convergence and reduces the oscillation when exploring the parameter space [21].
- AdaGrad [22]: this technique is very useful in the case of sparse data. Indeed, it adjusts dynamically the learning rate for each parameter separately in such a way that larger updates are performed for infrequent updated parameters and smaller updates are performed for frequent updated parameters.
- Adaptive moment estimation (Adam) [23]: this is another technique that computes individual adaptive learning rates for different parameters. It combines the advantages of momentum and AdaGrad by keeping a fraction of the first and second moments (mean and variance) of the past gradients. This is a popular technique for optimization as its convergence is greatly speed but it is highly recommended to compare it with the standard SGD method [24].

The performance of the translation system is evaluated in terms of BLEU [25]. The results according to the optimization algorithms are reported in Table 1. On our data set, the standard

Table 1: *Baseline NMT systems according to the optimization algorithms: BLEU(%)*.

Methods	Dev	Test
SGD	7.83	5.35
Momentum	4.19	2.89
AdaGrad	6.27	4.61
Adam	6.33	4.49

SGD optimization algorithm achieves better performance. It is shown that standard SGD optimization without momentum tends to find the optimal parameters, but the issue is that it is time consuming and there is a risk of getting stuck in saddle points [21]. We can also note that AdaGrad and Adam, which are based on the same idea, gives similar results.

All these results are bad because they produce translations that are not reliable. Also, it should be noted that, the conventional SMT system achieved a BLEU of 33.72 on the Dev and 24.54 on the Test, which is much better than the NMT baseline system. The translation model for the SMT system is trained on the same parallel data set. The language model is a 3-gram language model trained on the corpus of the target language. In order to boost the baseline neural model, we tested an advanced technique that theoretically improves the baseline system.

3.2.2. Attention technique

There are powerful techniques called *attention*, which are used to fix some problems of the baseline *encoder-decoder* model. In [10], the authors proposed to encode the source sentence into a context vector that is dynamically produced according to the

target word being generated. For this, they change the architecture of the *encoder* and the *decoder* as follows:

- **Encoder** Instead of encoding the input sequence F by starting from the first word f_1 up to the last one $f_{|F|}$ (i.e. usual RNN), a bidirectional RNN is used. In this kind of networks, the input sentence is, firstly, encoded as it is ordered to generate a sequence of hidden states $(\overrightarrow{h_1^{(f)}}, \dots, \overrightarrow{h_{|F|}^{(f)}})$. Afterwards, it is encoded in the reverse order resulting a sequence of hidden states $(\overleftarrow{h_1^{(f)}}, \dots, \overleftarrow{h_{|F|}^{(f)}})$. Finally, to obtain the annotation h_t for each word f_t , the two output hidden states $\overrightarrow{h_t^{(f)}}$ and $\overleftarrow{h_t^{(f)}}$ are concatenated as follow $h_t^{(f)} = [\overrightarrow{h_t^{(f)}}; \overleftarrow{h_t^{(f)}}]'$. With this approach, each word f_t will depend on both the preceding words and the following words, which will reduce the distance between words of the source sentence and those of the target sentence. This is very useful in the case of pairs of languages that share the same *Subject-Verb-Object (SVO)* structure (French-English for example).
- **Decoder** From the Equation 6, it is clear that the conditional probability is modeled by using the previous predicted words $\{e_1, \dots, e_{i-1}\}^1$ and a fixed-length context vector c generated from the source sentence. In the model with attention, this probability is conditioned by $\{e_1, \dots, e_{i-1}\}$ and a distinct context vector c_i for each target word e_i as it is shown in the Equation 7

$$p(e_i | \{e_1, \dots, e_{i-1}\}, c) = NN_output(e_{i-1}, h_i^{(e)}, c_i). \quad (7)$$

This context vector c_i depends on the sequence of annotations $h_1^{(f)}, \dots, h_{|F|}^{(f)}$ generated by the *encoder* as it was explained in the previous point. Hence, c_i is computed as a weighted sum of these annotations $h_j^{(f)}$

$$c_i = \sum_{j=1}^{|F|} \alpha_{ij} h_j^{(f)}. \quad (8)$$

The weight α_{ij} for each annotation $h_j^{(f)}$ represents the probability of aligning a source word f_j with a target word e_i . It is modeled by a feed-forward neural network, which depends on the annotation $h_j^{(f)}$ of the input sentence and the output of the previous RNN hidden state $h_{i-1}^{(e)}$.

By applying attention technique on our data, the BLEU score has been significantly improved: 28.10 on the Dev 20.63 on the Test. This improvement is essentially due to the explicit modeling of the alignment between words of the source sentence and those of the target sentence. Note that, contrary to the baseline model, in this one, two neural networks have been used, one for encoding and decoding and the second one for the alignment.

3.2.3. Handling unknown and rare words

NMT operates on constrained vocabulary and in addition it replaces a huge amount of rare words by the <unk> token. This

¹Henceforth, we used i to index the target sentence words and j to index those of the source sentence.

solution is the most used even in the statistical approach, however it suffers from some shortcomings. In fact, the produced target sentence can contain $\langle \text{unk} \rangle$ that breaks the structure of the sentence and consequently it changes its meaning.

One way to handle this issue is to take benefit from an external dictionary containing a list of words with their translations and a score for each translation.

The first approach, proposed by [26], consists of using the alignment function to map unknown words in the target sentence with their corresponding words in the source sentence. Afterwards, an external dictionary is used to translate each unknown word. This solution is known as the $\langle \text{unk} \rangle$ replacement technique, referred in the following as **RepUnk**.

Another approach [27] handles the issue of the translation of infrequent words, in the training data. To do so, the probability of each target word of the vocabulary is recalculated by taking into account the probability of words in an external lexicon. With this, the probability of an infrequent word in the vocabulary is adjusted by the one calculated from the external lexicon. The probability of a word e_i of the external lexicon is estimated as follows:

$$p_{lex}(e_i | \{e_1, \dots, e_{i-1}\}, F) = \sum_{j=1}^{|F|} \alpha_{ij} P_{lex}(e_i | f_j) \quad (9)$$

Afterwards, this probability is added as a bias to the *softmax* probabilities calculated by the neural network. This technique of adjustment will be named in the following **ProbAdjust**.

In order to test these two approaches, we built automatically a dictionary of 10M entries from a corpus of 9M parallel sentences. By incorporating our lexicon in the NMT system with attention, we achieved the results reported in Table 2. Using an

Table 2: BLEU(%) after incorporating an external lexicon in NMT system with attention.

Methods	Dev	Test
Attention	28.1	20.63
RepUnk	28.09	21.03
ProbAdjust	25.88	19.79

external lexicon to replace unknown words improves a little bit the translation on the Test. However, by adjusting translation probabilities, the system performance decreases on our data set. One reason of this could be that the external probabilities may improve the likelihood of infrequent events, but they can also collapse the probabilities of those that are frequent.

3.2.4. NMT architecture

For the purpose of better modeling and learning long-term dependencies, we used LSTM blocs rather than recurrent neural networks. The advantage of LSTM, among others, is to prevent the Vanishing gradient issue [24] for which RNNs suffer from.

Testing this architecture by varying the number of hidden layers from 2 to 6 and by using LSTM has not shown an improvement on our data set.

3.2.5. Beam search

Beam search is an heuristic search technique that is used by the decoder to generate the best translation. The idea is to explore in each step a subset of possible translations of size m (the beam size). This size has a strong impact on the translation quality; by

increasing the beam size, the decoder explores a larger subset of possible translations and consequently it ensures a better translation. In the previous tests, the beam size was fixed to 1, which we consider as too restrictive. The experiments show also that the word penalty score may have a positive impact on the quality of translation. The translation performance in accordance to the beam size and the word penalty is presented in Figure 2.

A word penalty of 1 means that no alteration is done on the probability of the generated sentence, since it is multiplied by 1. By decreasing this parameter, the decoder tends to produce longer sentences and specially when the beam size gets longer. In this case the results could be improved. The curves of Figure 2 show that the best size of the beam is 20 and the best word penalty is 0.5, which leads to a BLEU of 32.05 on the Dev corpus. With these parameters, the Recurrent Neural Network method with the following features: one hidden layer, Attention technique and Replacement $\langle \text{unk} \rangle$ approach, achieves a BLEU score of 24.37 on the test corpus. Even this sophisticated method, it does not outperform the statistical approach that achieves a BLEU score of 24.54. This is due to the size of training corpus which is rather limited [12].

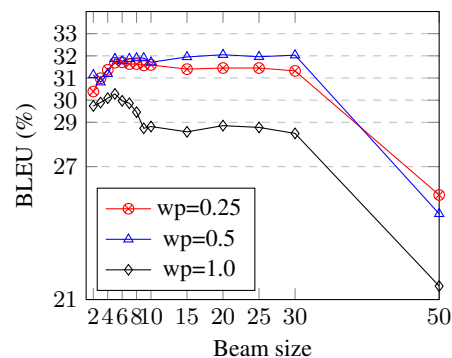


Figure 2: The translation quality on the Dev with respect to the beam size and the word penalty.

4. Conclusions

Neural machine translation is a new paradigm that uses deep learning to build a single large artificial neural network translating a whole source sentence to a target one. In this paper, we described the baseline architecture proposed for this approach, namely the *encoder-decoder* framework. This baseline system was compared with the conventional SMT approach on the task of Arabic-English machine translation; the results showed that SMT performs much better than NMT (an absolute difference of 19% in the BLEU score). Afterwards, in the aim of improving the performance of the baseline NMT system, several advanced techniques were detailed and tested. Although these techniques reduced significantly the gap between SMT and NMT (an absolute difference of 0.17% in BLEU), there is still several issues to deal with. The advantage of NMT is to use a component that encodes and decodes, but through the experiments we did in this paper, we show that this architecture needs external components and some alteration of probabilities to come closer the performance of SMT.

5. Acknowledgements

We would like to acknowledge the support of Chist-Era for funding this work through the AMIS (Access Multilingual Information opinionS) project.

6. References

- [1] W. J. Hutchins, "The georgetown-ibm experiment demonstrated in january 1954," in *Conference of the Association for Machine Translation in the Americas*. Springer, 2004, pp. 102–114.
- [2] H. Somers, "Example-based machine translation," *Machine Translation*, vol. 14, no. 2, pp. 113–157, 1999.
- [3] L. Dugast, J. Senellart, and P. Koehn, "Statistical post-editing on systran's rule-based translation system," in *Proceedings of the Second Workshop on Statistical Machine Translation*, ser. StatMT '07. Stroudsburg, PA, USA: Association for Computational Linguistics, 2007, pp. 220–223. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1626355.1626387>
- [4] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Comput. Linguist.*, vol. 19, no. 2, pp. 263–311, Jun. 1993. [Online]. Available: <http://dl.acm.org/citation.cfm?id=972470.972474>
- [5] R. Zens, F. J. Och, and H. Ney, *Phrase-Based Statistical Machine Translation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 18–32.
- [6] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models." Seattle: Association for Computational Linguistics, October 2013.
- [7] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *CoRR*, vol. abs/1409.3215, 2014. [Online]. Available: <http://arxiv.org/abs/1409.3215>
- [8] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *CoRR*, vol. abs/1409.1259, 2014. [Online]. Available: <http://arxiv.org/abs/1409.1259>
- [9] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, ukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016. [Online]. Available: <http://arxiv.org/abs/1609.08144>
- [10] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.
- [11] P. Isabelle, C. Cherry, and G. F. Foster, "A challenge set approach to evaluating machine translation," *CoRR*, vol. abs/1704.07431, 2017. [Online]. Available: <http://arxiv.org/abs/1704.07431>
- [12] P. Koehn and R. Knowles, "Six challenges for neural machine translation," *CoRR*, vol. abs/1706.03872, 2017. [Online]. Available: <http://arxiv.org/abs/1706.03872>
- [13] L. Bentivogli, A. Bisazza, M. Cettolo, and M. Federico, "Neural versus phrase-based machine translation quality: a case study," *CoRR*, vol. abs/1608.04631, 2016. [Online]. Available: <http://arxiv.org/abs/1608.04631>
- [14] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, ser. NAACL '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 48–54. [Online]. Available: <https://doi.org/10.3115/1073445.1073462>
- [15] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179 – 211, 1990. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/036402139090002E>
- [16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [17] K. Cho, B. van Merriënboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *CoRR*, vol. abs/1406.1078, 2014. [Online]. Available: <http://arxiv.org/abs/1406.1078>
- [18] A. Eisele and Y. Chen, "Multiun: A multilingual corpus from united nation documents," in *Proceedings of the Seventh conference on International Language Resources and Evaluation*, D. Tapias, M. Rosner, S. Piperidis, J. Odjik, J. Mariani, B. Maegaard, K. Choukri, and N. C. C. Chair, Eds. European Language Resources Association (ELRA), 5 2010, pp. 2868–2872.
- [19] M. A. Menacer, O. Mella, D. Fohr, D. Jouvett, D. Langlois, and K. Smaili, "An enhanced automatic speech recognition system for Arabic," in *The third Arabic Natural Language Processing Workshop - EACL 2017*, ser. Arabic Natural Language Processing Workshop - EACL 2017, Valencia, Spain, Apr. 2017. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01531588>
- [20] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Networks*, vol. 12, no. 1, pp. 145 – 151, 1999. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0893608098001166>
- [21] S. Ruder, "An overview of gradient descent optimization algorithms," *CoRR*, vol. abs/1609.04747, 2016. [Online]. Available: <http://arxiv.org/abs/1609.04747>
- [22] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.
- [23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [24] G. Neubig, "Neural machine translation and sequence-to-sequence models: A tutorial," *CoRR*, vol. abs/1703.01619, 2017. [Online]. Available: <http://arxiv.org/abs/1703.01619>
- [25] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.
- [26] T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba, "Addressing the rare word problem in neural machine translation," *CoRR*, vol. abs/1410.8206, 2014. [Online]. Available: <http://arxiv.org/abs/1410.8206>
- [27] P. Arthur, G. Neubig, and S. Nakamura, "Incorporating discrete translation lexicons into neural machine translation," *CoRR*, vol. abs/1606.02006, 2016. [Online]. Available: <http://arxiv.org/abs/1606.02006>

Assessing the Usability of Modern Standard Arabic Data in Enhancing the Language Model of Limited Size Dialect Conversations

Tiba Zaki Abulhameed^{1, 3}, Imed Zitouni², Ikhlas Abdel-Qader¹, Mohamed Abusharkh⁴

¹Western Michigan University MI, USA

²Microsoft Research WA, USA

³Al-Nahrain University, Baghdad, Iraq

⁴Ferris State University, MI, USA

tibazaki.alhabba@wmich.edu

Abstract

Conversations are mostly spoken through a variety of dialects and the need for accurate speech recognition systems is growing exponentially in a wide range of applications. This paper presents an evaluation of the feasibility of using Modern Standard Arabic (MSA) data to enhance the perplexity (pp) of the Language Model (LM) of limited size Iraqi dialect conversations. We are interested in adapting the MSA's LM to Iraqi dialect by exploiting the capabilities of word2vec to deliver clusters of words' vector representations classifying dialect specific words along with relevant MSA words. The vocabulary set, of size 21kword, is extracted from Iraqi conversational speech from Appen LDC, while the MSA was from GALE phase2 part1 and 2 from LDC. Two approaches were evaluated; 1) combining both corpora as one training set, and 2) generating separate LM for each corpus, and producing interpolated LMs. Results show that the second approach produced the best improvement of around 21% over the baseline Iraqi tri-gram alone and 15% over interpolated Iraqi tri-gram with Iraqi word2vec class n-gram while due to the different structure of Iraqi dialect and MSA, no improvement was gained from mixing them.

Index Terms: word2vec, Arabic dialect, Language modeling, Speech recognition

1. Introduction

Speech recognition tools have made major leaps in recent years and the accuracy of such technology grew from around 70% in 2010 to around 95% currently for market leaders like Google Now and Siri[1]. This can be attributed to many reasons but certainly robust LM is critical to predict unclear sentences or sentence parts. However, producing efficient LM with low perplexity remains a challenge that is needs addressing. This is especially true in the case of Arabic language in which a complex lexicon and multiple living dialects make LM-based prediction a more challenging task.

A closer look at LM for Arabic dialects highlights the challenges that cause higher ambiguity than the standard language commonly termed MSA. Dialects pronunciation and rules differ from those of MSA. This can occur to the extent that these dialects can become mostly unrecognizable for MSA speakers and significant effort is required to facilitate communication between speakers of different dialects. Looking at the Iraqi dialect as an example, Iraqi dialect has diversity in pronunciation based on region, spanning from north to the south of Iraq, and many

words originated from other regional languages such as Turkish, Farsi, and English. Also, some words come from MSA but have a slightly changed pronunciation. This causes an unbounded vocabulary set to develop. Moreover, speakers may use unrestricted grammar [2]. An additional challenge comes from the tendency of dialects to adjust the vocabulary and language usage through relatively short time intervals to reflect the cultural and generational changes. As an illustration, a new expression for "a friend" was introduced by one popular Iraqi T.V. comedy series. This made many young people to use the new term and thus a new vocabulary synonym of close friend is introduced in the daily conversations. Such word usage varies over time and LM should be enabled to recognize such evolution and adapt.

Within our assessment of the usability of MSA in enhancing the Iraqi dialect LM, we are addressing three main challenges for the Iraqi dialect conversations LM; Data sparsity, adapting different speech domains (conversations vs. broadcast news), and the data size limitation. Resolving these issues would result in a more accurate the LM.

First issue is the data sparsity, which is a feature inherited from the mother language MSA is our first challenge and the most difficult one. We resolved the data sparsity issue by using class-gram LM and employing word2vec [3] in a comprehensive scheme that produces class n-gram. word2vec was introduced as a machine learning tool that constructs feature vectors for words in the input data set [3]. It was successfully used in sentiment analysis and document classification work [4], [5]. Since we are using both MSA and Iraqi, words will appear in many different contexts. We hypothesize that employing word clustering capabilities based on k-means classifier can produce effective language modeling when words are used in various ways. Thus, clustering words and using the class probability, where the cluster number is word's class, would reduce the sparse words probability. More specifically, we propose using word2vec to cluster corpus words into classes that, in turn, would be used to build a language model using class n-gram technique [6].

The second challenge is due to the fact we are targeting conversational speech transcription language modeling and aiming to produce improvements by adapting MSA broadcast news and reports LM to Iraqi dialect phone conversations. We resolved this issue by creating the training data set through interpolation of different LMs.

The third challenge is due to the small size Iraqi training data, which we resolved by expanding our training data with data taken from MSA corpus. This work is focused on exploring

the feasibility of mixing the training data to enlarge the limited size dialect data.

We define this problem as Adaptation problem and consider Iraqi dialect as domain specific, while MSA as the general big background data set. Our scheme has the objective of minimizing LM perplexity by adapting the MSA LM to Iraqi dialect, which can lead to significant improvements of the speech recognition systems for dialect-based conversations.

The paper is organized as follows. Section 2 discusses the most pertinent research efforts. Section 3 presents the language model used in this work while section 4 experimental proposed scheme 5 relays the setup, results, and analysis. Finally, Section 6 concludes the paper.

2. Related Work

Class-based LM was shown to be effective to solve data sparsity problems of datasets. Instead of depending on independent word prediction, words are clustered into classes and via modeling, prediction can be achieved. This was shown in [7] where Part Of Speech (Part-of-speech-tag (POS)) was used to classify the data and produce neural network (NN) n-gram LM. Previous efforts considering Egyptian dialect LM for ASR reported that the best prediction results are achieved when n-gram is combined with class n-gram [8]

The challenge in this scenario rises from the assumption that the dataset is fully classified. Yet, most of available dialect data sets are not annotated for the purpose of word classification. A major data classification method that was proven to be effective for other languages is the count base statistical clustering of the corpus such as the hierarchical Brown clustering. NN-based word embedding was also used to replace manual tagging of POS. In [9], where it was shown that word embedding can be used for unsupervised POS tagging.

In the context of Arabic language, word2vec was used as a word embedding tool for Arabic sentiment analysis in [10]. They considered MSA and dialectal Arabic sentiment opinion (specifically Egyptian dialect). The features were extracted from word2vec as an alternative of hand-crafted methods. In addition, word2vec proved its applicability in Arabic information retrieval and short answer grading in [11]. This was tested using twitter and book reviews that are considered as the combination of more than one dialect, while new articles were considered for MSA. Also, word2vec was used to produce a comparable corpus (CALLYOU) from Youtube for Algerean, MSA, and French comments [12]. Linear interpolation has been proposed in many scenarios as an effective way of LMs adaptation. In the literature, the adaptation of domain specific LM of same language was mainly explored [13] [14]. In a similar way, we are considering the dialect as a separate language domain that has intersection with MSA domain. For mixing heterogeneous text data of domain-specific LM, [15] produced term weighting that is used to decide in-domain and out-of domain text segments for the purpose of document classification. This inspired us to produce a new weighting function to filter the MSA from text segments of less common words with the dialect.

Our approach aims to make use of the semantic representation of the word embedding in word2vec in generating class-gram. It is noted in the literature that most of the efforts are focused on MSA with a few efforts considering Egyptian and Levantine dialects. Despite being rich and commonly used, Mesopotamian dialect group is spoken by approximately 30 million people but yet there is a lack of research catering to this prominent dialect. Thus, there is a need of a comprehensive LM

that improves speech recognition performance and potentially highlights dependencies between Iraqi dialect and MSA.

3. Language model

One of the main factors in efficient ASR is the language model that will be fed to the system to decide the hypothesized spoken word using context-based probabilities. The pp is the metric that is used for computing the Language model efficiency [6]. A lower value of pp means high value of the expected word probability in a certain context and naturally a lower number of bits needed to encode the words. This enables easier decision making (i.e. reduced sparsity)[8].

The simplest possible regular LM is the unigram. Each unit of the language has a probability of its count divided by total words count in the corpus. An n-gram model is a model that counts the word probability taking into consideration word history (i.e. context) of length n-1. So, the probability of a certain word x to appear in the corpus given the previous word was word y is

$$p(x|y) = \frac{\text{count}(yx)}{\text{count}(y)} \quad (1)$$

Certain algorithms can be applied to smooth the probabilities and achieve better pp such as Kneser-Ney [8]. Smoothing possibly includes three operations to improve accuracy, namely, discounting, back-off and interpolation. For example, class based bigram computes the probability of word x, given the previous word, y as

$$p(x|y) = p(x|class(x))p(class(x)|class(y)) \quad (2)$$

where the conditional probability of the word x, given that it appears in a unique class, class(x), is defined as the ratio of the number of occurrences of word x, to the total number of occurrences of it's class within the corpus. Classes are mutually exclusive where a word belongs to one class only. In addition, classes lengths are not necessarily equals.

$$p(x|class(x)) = \frac{\text{count}(x)}{\text{count}(class(x))} \quad (3)$$

$$p(class(x)|class(y)) = \frac{\text{count}(class(y)class(x))}{\text{count}(class(y))} \quad (4)$$

where

$$\text{count}(class(y)) = \sum_{i \in I} \text{count}(w_i \in class(y)) \quad (5)$$

4. Methodology

4.1. Running word2vec

Data preprocessing includes removal of unneeded tags and considering words, that are not in the Iraqi 21k word vocabulary set, as unk words (unknown). As shown in Fig.1, the preprocessed training dataset is then used as an input into the word2vec tool. word2vec uses a single hidden layer-fully connected NN. The input and output layers are both of the same size of vocabulary words. To reduce computation complexity, the hidden layer was replaced by simple projection. This makes word2vec capable of large data training. The feature vector is extracted from the weights matrix between the input layer and the hidden layer where the end target would be the word's neighbors in context [3]. Continuous Bag Of Words (CBOW) was tested to predict a word from the input context window with 700 class, vector size 350, and window width 5 was set as word2vec parameter.

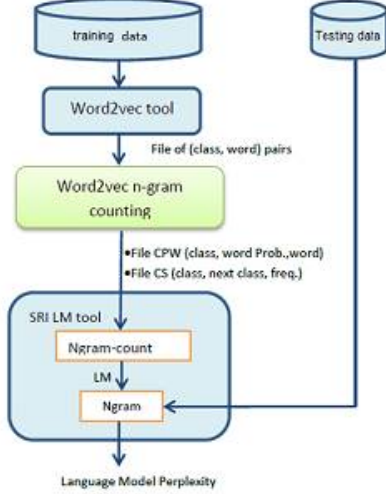


Figure 1: Counting and LM Generation

Table 1: examples from classes.txt of words semantic clustering output of word2vec using CBOW gram of 10 cluster

class9	class3	class6
السلام greeting peace	سوية together	مشغولة busy
أهلاً greeting Hi	تجي come	ملتهدبي busy
أهلاً greeting Hi	ع تروح go	مرضة sick
اهلا greeting Hi	ترجع return	مشاكل problems

Multiple output are produced by word2vec to be then used in describing the language model. In our work, we are more interested in the output of word2vec (classes.txt) file which contains class-word pairs for all of the words in the dataset. To show how word2vec semantically classifies the data, we present few examples of word classes from the corpus in table 1. For example, class 3 shows some words of the same root clustered together which clearly is reflecting the semantic and syntactic clustering dependency. In class 0, we also notice the semantically related words clustered together. In class 9, three different orthography formats of same word أهلا, which means "Hi or welcome", appeared in the same class. This implies that word2vec was -to a certain extent- successfully capturing semantic and syntactic relations of Iraqi dialect.

Next, we built a supporting tool in Python that takes one file (classes.txt) from word2vec output and generates two files, CPW and CS, with the Iraqi unclassified words separated in one class of its own. Also, each line in CPW contains the tuple class C , probability P of a word within class, and the word W . P was computed per equation 3. In CS file, each line contains the tuple: (class C_i , next class C_j , the number of occurrence of any word of class C_i followed by any word of class C_j for all classes i and j). The "start" of the sentence is represented using the tag $\langle s \rangle$ while the sentence end is indicated by the tag $\langle /s \rangle$. We used two approaches to handle the out of vocabulary(OOV) words. In the first, we tagged those words as $\langle unk \rangle$ and were treated as words in their own unified class while in the second, they were just left as it is and classified with the vocabulary

words, but were not considered in computing the probabilities for the language model.

CS is then used with SRILM to count class n-gram LM, while CPW is used in computing the pp as follows:- See the following command :

```
$ ngram-count -order 2 \
  -read CS -write f1.ngrams
```

```
$ ngram-count -order 2 \
  -read f1.ngrams -lm w2vClassBased.lm
```

```
To calculate the \ac{pp},
$ ngram -lm w2vClassBased.lm \
  -classes CPW -ppl test-data
```

4.2. Combining Iraqi and GALE Corpora

We propose to combine Iraqi with the GALE Corpora to compensate for the small data size that is available for this project. The combination was performed via two methods as follows:

- The whole Iraqi corpus with nearly equivalent part of GALE corpus.
- Knowing that Iraqi corpus is about 0.10 of the GALE MSA, we duplicated the Iraqi corpus ten times to be equivalent to GALE, so that Iraqi vocabulary words would not be excluded from classification due to its frequency (data set size).

4.3. Interpolating with LMs

The interpolation is presented using the following equation:

$$p(w) = 0.5p_{IraqiTri-gram}(w) + 0.3p_{IraqiCB}(w) + 0.2p_{GALECB}(word) \quad (6)$$

where CB refers to word2vec class n-gram. The lambda weights were estimated using the Iraqi data tuning set. The lambda weights combination that produced best results for the tuning set were chosen for the final interpolated versions. The flow diagram of the interpolation that produced the best results is shown in fig.2. In addition, refining the GALE data was also considered for which we computed, for each sentence, the probability of $\langle unk \rangle$ word. If the probability is more than 0.3, then the sentence will be discarded and considered as noise. Also, the same interpolation technique was tested using the refined version of the data.

4.4. Calculating perplexity using SRILM tool

SRI tool for Language Modeling[16] was introduced as a solution to generate LM and compute Perplexity of the test data. All tests were unified on the same Iraqi devtest and the same vocabulary set. For class based n-gram, experiments were repeated 10 times and the average of the results was considered.

5. Experimental Results

5.1. Work Setup

The corpus used in the experiments is the Iraqi Arabic Conversational Telephone Speech (LDC2006S45) [17]. This corpus is taken from Linguistic Data Consortium (LDC) and it contains

Table 2: A comparison of the results of different LM mixing for multiple clustering methods.

Mixing corpora	pp	Separate corpora	pp
-	-	Iraqi alone tri-gram	373.053
-	-	Iraqi alone CB	387.146
Iraqi + 0.10 of GALE tri-gram	390	GALE CB MSA alone	745.976
Refined Iraq + 0.10 of GALE tri-gram	377.74	Refined GALE CB MSA alone	351.99
Iraqi + 0.10 of GALE CB	424.269	Interpolated Iraq CB with Iraqi tri-gram	345.7
10 times duplicated Iraq+ GALE tri-gram	656.98	Interpolated Iraq CB with (OOV as <unk>) GALE CB and Iraqi tri-gram	294.6
Refined 10 times duplicated Iraq+ GALE tri-gram	595.2	Interpolated Iraq CB with Refined GALE CB and Iraqi tri-gram	298.65
10 times duplicated Iraq+ GALE CB	645.9	Interpolated Iraq CB with (classified OOV) GALE CB and Iraqi tri-gram	293.127

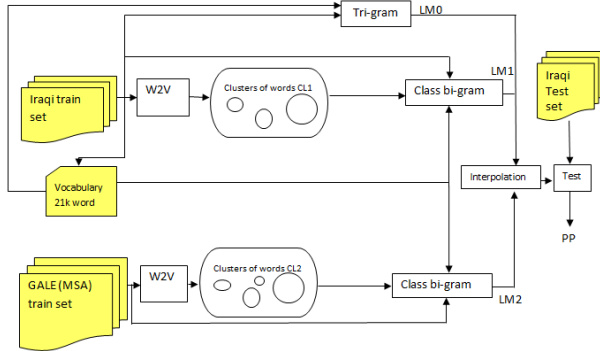


Figure 2: Counting and LM Generation

276 Iraqi Arabic speakers in the form of Iraqi dialect telephone conversations. The data set is subdivided as train-c1, train-c2, and devtest. Train-c1 represents one side of recorded phone conversation and train-c2 is the second side. the combined training set transcriptions contains 199kword and 1.8MB. The devtest is a balanced 6% of the data and it is a certified standard test set according to the test process applied by the National Institute of Standards and Technology (NIST). Another 4% of the data were used as tuning set. For our experiment, we used 90% of the corpus for training , that is 199k word, while devtest is used for testing and it is 102KB of about 12kword. GALE dataset contains about 1387kword of MSA broadcast news and reports [18] [19].

5.2. Results and Discussion

Results from combining the two corpora are shown on the left side of table 2 while on the right we present interpolated versions of the separated corpora. In our attempt to expand the Iraqi data by mixing it with GALE, one might not expect that only less than 50% of Iraqi words will actually appear in the MSA. This was clear when we extracted the intersecting words between them. The Iraqi vocabulary is 21kword and the GALE vocabulary is about 111kword. Surprisingly, we found that the intersection set is only 9kword, yet only about 5kword is considered for clustering using word2vec. This is because in our experiment, we ignored words of frequency less than five from classification. This is similar to the default setup in word2vec. This may justify the lack of improvements under the method of mixing of two corpora. Indeed, Iraqi dialect and MSA are both Arabic but they have almost different vocabularies which prevented this method from performing. Also, the data nature, that

is news and Broadcast, is different in context and vocabulary choices from those in phone conversations.

As an illustration, one arbitrary segment of each corpus is further explored in figure 3(a,b). One can notice that <unk> words appeared in GALE due to the use of Iraqi vocabulary alone. These <unk> words refer to word appear in GALE but not in Iraqi. This causes high ambiguity when testing on the Iraqi devtest data. In fact the only common words in this arbitrary segments are أنف على ما ها هو , that is most of it are

connecting words except أنف. These connecting words does have high frequency in both corpora, which means that the gain of having them in GALE was not that significant in enhancing the LM. We need to support words that appear in an average frequency in the Iraqi data. Though, refining GALE data before calculating word2vec class n-gram achieved 6% improvement over Iraqi alone tri-gram and 10% improvement over Iraqi alone word2vec class n-gram. Interpolating the refined version did not give better results that unrefined one. Best results for this interpolation was recorded when Iraqi alone tri-gram, Iraqi CB, and GALE CB λ weights were 0.4, 0.5, and 0.1 respectively. This give us the clue that maintaining OOV words can be used further in better way. At the same time, one can also observe that the lowest perplexity was produced by the interpolated Iraqi word2vec class n-gram with GALE word2vec class n-gram and Iraqi tri-gram. Iraqi and MSA appears to produce the same structure (syntax / n-gram). This actually indicates that interpolation is useful.

Interpolating the language models and enforcing the MSA's LM contributions to be small via the small Lambda weight, leads to sharing common words probability and reducing the sparsity.

Also, we would like to note on the classification of words that are in Iraqi but not in GALE. Theoretically, the class probability would enhance the conditional (word|class) probability for these words, however, discarding all GALE MSA vocabulary did not allow for this enhancement to occur. Nevertheless, if we would to consider all GALE vocabulary, the words sparsity will be higher causing higher pp.

We have actually considered the MSA words in the vocabulary and the results show that not all Iraqi words clustered within similar MSA words but there are some that did, such as words with different spelling but same actual use and syntax. This justified for us to discard all OOV words before passing them to word2vec. This procedure resulted in 1% (with error range ± 0.06) less improvement over passing them to be clustered. That is the improvement over interpolated Iraqi tri-gram and class gram when replacing oov by <umk> is 14%, and 15% when leaving them. Consistently, we have improvement over Iraqi tri-gram when replacing oov by <umk> is 20%, and 21%

GAWO: Genetic-based optimization algorithm for SMT

Ameur Douib¹, David Langlois¹, Kamel Smaili¹

¹University of Lorraine, LORIA, France

ameur.douib@inria.fr, david.langlois@loria.fr, kamel.smaili@loria.fr

Abstract

In this work, we propose GAWO, a new method for SMT parameters optimization based on the genetic algorithms. Like other existing methods, GAWO performs the optimization task through two nested loops, one for the translation and the other for the optimization. However, our proposition is especially designed to optimize the feature weights of the fitness function of GAMaT, a new genetic-based decoder for SMT. We tested GAWO to optimize GAMaT for French-English and Turkish-English translation tasks, and the results showed that we outperform the previous performance by +4.0 points according to the BLEU for French-English and by +2.2 points for Turkish-English.

Index Terms: Statistical Machine Translation, log-linear approach, optimization of feature weights, genetic algorithms

1. Introduction

Statistical machine translation (SMT) systems combine a set of features in order to evaluate the quality of a translation hypothesis at the decoding time. Each feature estimates the quality of the hypothesis in a particular aspect. The best translation in the output of the system is the one that maximizes this evaluation score. The feature values are mainly probabilities, i.e. language model probability, translation model probabilities, but other kinds of features are calculated and not interpreted as probabilities, i.e. phrase and word penalties.

In the SMT community, the log-linear approach [1, 2] is largely used to combine these features as follows:

$$Score(e) = \sum_{i=1}^{|\lambda|} \lambda_i \times \log(h_i(e, f)) \quad (1)$$

Where e and f are, respectively, a translation hypothesis and the source sentence, and h_i is the i^{th} feature function. To define the influence of each feature in the final score, a weight λ_i is associated to the feature h_i . The weight values have an effect on the scoring of the translation hypotheses at the decoding time, so, they have an effect on the choice of the best one in the output of the decoder. Therefore, within SMT systems the optimization of the weights of the features is a crucial step to insure a satisfactory translation quality.

To optimize these weights for a SMT system, we assume that we have a *development* set consisting of source sentences and their reference translations. Then, using the decoder of the system, we translate the source sentences. The goal of the optimization is to find the set of weights λ that minimizes or maximizes the error function (Err) defined to estimate the difference between the output translations and the references. Ideally, to perform this setting, we apply a classical optimization algorithm by running the decoder many times and adjusting the set of weights, until a convergence, and as error function we use an evaluation machine translation metric. But, the run of the decoder several times is very expensive.

This is why, in practice, the weight optimization algorithms for SMT minimize the run of the decoder. They perform in two nested loops; in the outer loop, the decoder uses a set of weights to translate the *development* set and produces the *k-best* translations for each source sentence. In the inner loop, an optimization algorithm is applied to perform the weight tuning. In the inner loop, the main goal is to find the optimal weights that select the best translation for each source sentence from the *k-best*. The selected translations must minimize or maximize the error function. The optimal obtained weights are re-injected into the outer loop to run the decoder again. The process is stopped when the weights can no longer be improved, or no new translations can be produced.

MERT (Minimum Error-Rate Training) [3] is the default algorithm used to optimize the feature weights in the SMT community, and especially applied on MOSES [4], the reference translation system. As explained before the algorithm performs in two nested loops, where MOSES is used as a decoder and in the inner loop, a line-search optimization algorithm is applied to perform the tuning. Other algorithms have been proposed to optimize the weights of the log-linear approach. The main difference in all these algorithms lies in the inner loop, where different approaches are proposed to solve the problem of weight optimization. For instance, in [5], Hasler et al. used Margin Infused Relaxed Algorithm (MIRA) as an optimization algorithm. Hopkins and May proposed PRO [6], which works by learning a weight set that ranks translation hypotheses in the same order as the translation metric. More recently, Kocur and Bojar proposed to use a Particle Swarm Optimization algorithm (PSO) [7] to solve the problem in the inner loop.

However, the common point, and the weakness, of all these algorithms is the fact that they are specially designed to optimize the feature weights for the MOSES decoder. This makes their application difficult to another decoder.

In this paper, a Genetic Algorithm for weight optimization (GAWO) for SMT decoders is proposed. GAWO has the same principle as the others algorithms, with an outer loop for the translation and an inner loop for the optimization. However, our proposition is designed to optimize the parameters of GAMaT [8], a new genetic-based SMT decoder, which differs from MOSES (see Sections 2.1 and 2.2). In a previous work [8], GAMaT was tested and compared to MOSES, but by using a set of weights optimized by MERT for MOSES. In this work, GAWO will be used to optimize these weights for GAMaT to obtain an evaluation function more robust and better adapted to GAMaT which will improve the translation performance.

Genetic algorithms are known for their capacity of exploration and exploitation of the search space [9], particularly in the case of optimization problems [10]. So, a genetic algorithm is an interesting candidate to implement, in the inner loop, for the optimization of feature weights.

The paper is structured as follows. In Section 2, we present

the MOSES and GAMaT decoders. In Section 3 we present the tuning process for optimization of feature weights. In Section 4, we present GAWO the genetic-based algorithm for the optimization of feature weights. In Section 5, we present some experimental and comparative results. We conclude in Section 6.

2. SMT Decoder

In SMT systems the translation problem at the decoder level is considered as an optimization problem. The goal of the decoder is to find the best possible translation \hat{e} that maximizes the translation evaluation score:

$$\hat{e} = \underset{e}{\operatorname{argmax}} \left[\sum_{i=1}^{|\lambda|} \lambda_i \times \log(h_i(e, f)) \right] \quad (2)$$

As presented in the introduction, the weights λ_i must be optimized to obtain a robust evaluation function.

2.1. MOSES

MOSES is a SMT decoder, which uses a Beam-search algorithm [4] in order to retrieve the best possible translation. Starting with an empty set, the solution building process consists in producing incrementally a set of complete solutions from partial ones provided by a translation table. Because the translation is built incrementally, it is then difficult to challenge a previous decision of translation that can eliminate a partial hypothesis, even if it could lead to a good final solution.

2.2. GAMaT

An alternative to MOSES is to start with a complete translation hypothesis and try to refine it to retrieve the best solution. With complete translation hypotheses, it is possible to revisit each part of the search space and modify it, if necessary.

GAMaT is a new decoder for SMT based on a genetic algorithm. It has the advantage that it can refine several complete solutions in an iterative process and produce acceptable solutions. In fact, a possible solution is encoded as a chromosome, where the chromosome encloses several pieces of information (the source sentence segmented into phrases, a translation hypothesis also segmented into phrases, and the alignment between source and target segments) [8]. Then, from an initial population of chromosomes, we produce new ones by applying crossover and mutation functions [8]. The crossover function takes two chromosomes from the population as parents, and crosses them to produce two new chromosomes considered as children of the parents. The produced chromosomes share the chromosomal information of the parents. On the other hand, the mutation functions are applied to diversify the existing population. A mutation function takes one chromosome and modifies it at the phrase level to produce new one. At the end of each iteration, we estimate the fitness (score) of chromosomes to select which of them will be kept, for next generations. This is called the selection process. The same process is repeated from one generation to another, until convergence. So, the final translation is the one that is encoded in the best chromosome from the final population.

To evaluate the chromosomes, the fitness is calculated using the log-linear approach to combine a set of feature values, as presented in the Equation 1. In this work, eight basic features [8, 4] are combined:

- Language model probability

- Direct and inverse translation model probabilities
- Direct and inverse lexical weighting translation model probabilities
- Word penalty
- Phrase penalty
- Reordering score

3. Tuning

The tuning of a SMT system consists in optimizing the weights λ of the evaluation function (Equation 1). To this end, let's assume a *development* (*Dev*) set of source sentences $\{f_1, \dots, f_n\}$ with their reference translations $\{r_1, \dots, r_n\}$. The decoder produces a set of k translation hypotheses $\{e_i^1, \dots, e_i^k\}$ for each source sentence f_i .

The optimal set of weights is the one which minimizes the sum of errors between the translations $\{\hat{e}_1, \dots, \hat{e}_n\}$ and the references:

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmin}} \left[\sum_{i=1}^n \operatorname{Err}(\hat{e}_i, r_i) \right] \quad (3)$$

Where Err is the loss function to optimize at tuning, and \hat{e}_i is the highest scored hypothesis from the set of k -translations of f_i using the set of weights λ . In other words, \hat{e}_i is the best translation of f_i according to the weights λ .

The loss function estimates the quality of a set of translation hypotheses compared to the references. In the state-of-the-art, there are several choices [11] to define Err , ex: error function, soft-max loss, ranking loss. But the studies showed [11, 3] that the use of the translation evaluation metric BLEU (Bilingual Evaluation Understudy) [12] gives the best optimization performance. Therefore, in this work we use the BLEU metric to perform the weight optimization.

4. GAWO

As explained in the introduction, GAWO is a genetic algorithm for weight optimization. Similarly to the existing algorithms, GAWO works through two nested loops (see Figure 1): the outer loop, in which we use GAMaT as a decoder to translate the *Dev* set and the inner loop, in which the genetic algorithm is applied to optimize the weights. The process of these two loops is shown in the Figure 1 and detailed in what follows.

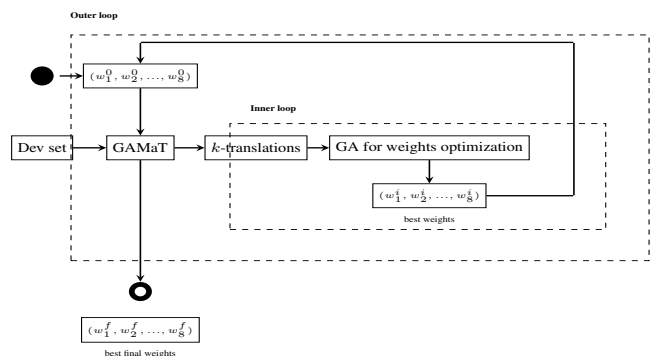


Figure 1: The GAWO process through the outer and the inner loop.

4.1. GAWO: outer loop

The *Dev* set is composed of n source sentences $\{f_1, \dots, f_n\}$ with their reference translations $\{r_1, \dots, r_n\}$. In the outer loop we use GAMaT to translate the source sentences, and produce for each one (f_i) a set of k -translations $E_i = \{e_i^1, \dots, e_i^k\}$. Therefore, the result of this loop is sets of k -translations $\{E_1, \dots, E_n\}$. These sets are injected into the inner loop for the optimization process. For the first iteration of the outer loop, we use a randomly generated set of weights. For the following iterations, the set of weights obtained by the inner loop is used to run GAMaT to produce novel sets of k -translations. The outer loop is stopped when the set of weights does not lead to new improvement.

4.2. GAWO: inner loop

In the inner loop, a classical implementation of the genetic algorithm is applied to find the best set of weights that selects a set of translations $\{\hat{e}_1, \dots, \hat{e}_n\}$ maximizing the BLEU score (see Equation 3). The main idea is to start with a population of chromosomes (solutions) i.e. a population of vectors of feature weights, and iteratively improve the quality of the population by producing new chromosomes.

To ensure the good evolution of the population in a genetic algorithm, a fitness function must be defined according to the task to solve, in order to evaluate the chromosomes and apply the selection process at the end of each iteration. As we deal with the task of optimizing weights, the fitness function is the BLEU score (see Equation 3). The process continues iteratively, until it reaches the best possible weights for the current sets of k -translations. In the next sections, we explain with more details the genetic process used in GAWO.

4.2.1. Chromosome encoding

In the genetic algorithm, a possible solution is encoded in a chromosome. Therefore, in our case a chromosome represents a vector of eight elements ($c = \lambda$), where each position in c represents the weight value λ_i of a feature function h_i used in the evaluation function in GAMaT (see Section 2.2).

4.2.2. Population Initialization

The chromosomes (solutions) of the first population are randomly generated. Where, each feature weight takes its value in the range $[-1, 1]$. To this randomly generated population, we add the set of weights used in the outer loop to run GAMaT.

4.2.3. Crossover function

The crossover function is applied to couple the chromosomes of the existing population to enhance the quality of this population. The function takes randomly two chromosomes, c_a and c_b , from the population and selects a random position s in the chromosome, to cross them. The crossover function produces two new chromosomes c_c and c_d , where $c_c = \{c_a\}^{left(s)} \cap \{c_b\}^{right(s)}$ and $c_d = \{c_b\}^{left(s)} \cap \{c_a\}^{right(s)}$.

In practice the crossover function is applied to couple all the possible pairs of chromosomes in the population.

4.2.4. Mutation function

As presented previously, the crossover function couples the existing chromosomes in the population, which limits the search space to the values of the weights generated at initialization. Consequently, the algorithm has a high probability to converge

towards a local optimum. This is why the mutation function is applied, in order to diversify the population. So, the mutation function selects a chromosome c_a from the population and modifies randomly one of its weight values to produce a new one c_b . For a better diversification, the mutation function is applied on all the chromosomes of the population.

4.3. Chromosome evaluation function

As presented before, the fitness function is the BLEU score. Therefore, to evaluate a chromosome c_a from the population, we process as follows. First, using the set of weights λ encoded in c_a , we recalculate the log-linear scores (see Equation 1) of every translation produced by GAMaT in the outer loop. After, for each source sentence f_i , we select from E_i the translation \hat{e}_i that maximizes the log-linear score. The result of this step is a set of n translations $(\{\hat{e}_1^\lambda, \dots, \hat{e}_n^\lambda\})$. Finally, we calculate the BLEU score between the selected translations and the references. The obtained BLEU score represents the fitness of the chromosome c_a .

In this way, the optimal set of weights is the one that is encoded in the chromosome maximizing the BLEU score.

5. Experiments & Results

5.1. Corpora

For the experiments, we use the translation data task of the workshop on Statistical Machine Translation. We take two pairs of languages to test GAWO in order to optimize the weights for GAMaT. The first pair is the French-English (FR-EN) a classical translation task. The second pair is the Turkish-English (TR-EN) which is a new task and poor in data. In addition, there is a stronger syntactic difference between Turkish and English. The training corpus of the language pairs FR-EN and TR-EN is composed of 1,3M and 280K parallel sentences respectively. Concerning the tuning and the test corpus, for both pairs of languages, both of them are composed of 1,000 parallel sentences. We use GIZA++ [13] to generate the translation model, and SRILM [14] to produce a 4-gram language model.

As presented in Section 4.3, the BLEU metric is used to perform the optimization. to evaluate the translation quality of the system on the test, we use BLEU, TER [15] and METEOR [16] metrics.

5.2. Results

In the Figures 2 and 3, we analyze the evolution of the translation quality of the *Dev* set throughout the optimization process. To this end we draw two curves, the first one (inner-loop) represents the evolution of the BLEU score through the inner loop, where each point represents the BLEU value of the best translations $\{\hat{e}_1, \dots, \hat{e}_n\}$ selected from the sets of k -translations by using the best weights produced by the inner loop. In other words, each point represents the fitness score of the best solution produced by GAWO in the inner loop. The second curve (outer-loop) represents the evolution of the BLEU value of the k -translations produced by GAMaT in the outer loop. This last curve allows us to analyze the evolution of the translation population quality produced by GAMaT.

After some experiments and like what is made in MERT [3] we fixed the number of the translations for each source sentence to 100 ($k=100$), these translations are injected in GAWO to perform the optimization process. To have a large variety of translations for each source, we add to the translations produced

by GAMaT at the iteration i those of the iteration $i-1$.

Knowing that the genetic algorithms that we use have a random behaviour, we analyzed the robustness and the stability of our algorithm to verify if GAWO achieves the convergence or not in any case. For this we did multiple runs (5) starting with the same weight values at each run (see Figures 2 and 3).

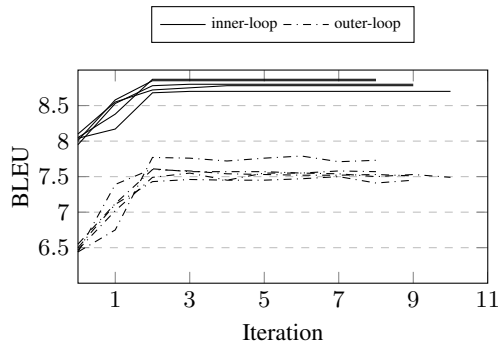


Figure 2: The evolution of the BLEU on the Dev set for TR-EN. The 1-best output from the inner loop and the 100-best output from the outer loop (GAMaT).

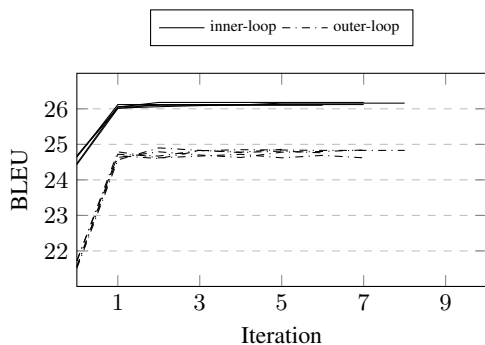


Figure 3: The evolution of the BLEU on the Dev set for FR-EN. The 1-best output from the inner loop and the 100-best output from the outer loop (GAMaT).

The first remark that can be made through these curves, is that the process of the optimization converges for all the runs and this for the both language pairs. Thanks to this convergence, we can conclude that GAWO allows GAMaT to produce better translations, by generating weight values more adapted for GAMaT.

We can also see that on average, three iterations are sufficient for the process to reach the maximum scores. This can be explained by the fact that we add the translations produced in the previous iteration for the current one. Therefore, after three iterations, GAWO deals with a huge number of translations for each source sentence ($k=300$). This allows to make better decisions and find the best weight values that helps to select the best translation for each source from 300 possible translations.

In terms of performance in the inner loop (inner-loop curves in Figures 2 and 3), for the TR-EN pair, the process achieves an average BLEU score of 8.8 for the five runs. for the FR-EN pair, it achieves an average BLEU score of 26.11. On the other hand, for the performance of the outer loop (outer-loop curves), for the TR-EN pair, GAMaT produces a population of translations with an average BLEU quality

around 7.6 and an average BLEU quality around 24.83 for the FR-TR pair.

In Table 1, we present the translation performance on the *Test* set, according to the three evaluation metrics previously cited. To estimate the improvement provided by the optimization of the weights by GAWO on the translation performance of GAMaT, we run GAMaT with the feature weights optimized by MERT for MOSES (*GAMaT-MERT*), and we run it also with the weights obtained from the different runs of GAWO. We show in Table 1 the average score of the five runs (*GAMaT-GAWO*). The confidence intervals are calculated with 95% of confidence.

Table 1: Translation performance on the test set according BLEU, TER and METEOR.

Language	Decoder	BLEU \uparrow	TER \downarrow	METEOR \uparrow
TR-EN	MOSES+MERT	10.29	83.03	20.2
	GAMaT-MERT	6.46	82.05	18.39
	GAMaT-GAWO	8,73 (± 0.1)	80,27 (± 0.41)	18,93 (± 0.04)
FR-EN	MOSES+MERT	31.29	52.14	29.97
	GAMaT-MERT	24.84	57.18	28.55
	GAMaT-GAWO	28.91 (± 0.14)	53,68 (± 0.07)	28,87 (± 0.07)

The obtained results show that the optimization of the weights using GAWO improve considerably the translation performance of GAMaT. Indeed, according to the BLEU score, the metric used at the tuning time, we outperform *GAMaT - MERT* by more than 2.2 points for the TR-EN pair and more than 4.0 points for the FR-EN pair. But we do not exceed those of the reference system decoder MOSES. The improvement is visible also according to the TER, where we outperform *GAMaT - MERT* by more than 1.7 point and those of MOSES by 2.76 points for the TR-EN pair. For the FR-EN we outperform also *GAMaT - MERT* by more than 3.5 points. The improvement is less visible according to the METEOR, because this metric is less sensitive than BLEU, when they compare the translations with the references.

On the other hand, the small intervals of the confidence intervals prove that GAWO is stable and allows to GAMaT to achieve the same translation performance each time we run it.

6. Conclusions

We presented GAWO, a new method for SMT parameter optimization based on the genetic algorithms. GAWO was tested to optimize the feature weights of the fitness function of GAMaT for two different pairs of languages. The obtained results of the different experiments demonstrated the feasibility of our proposition, where the algorithm allows to converge towards an optimum set of weights. Moreover, the translation performance, according to three evaluation metrics, showed that the optimization of the weights allows GAMaT to outperform the previous configuration.

For future work, we will make more experiments by testing other metrics to perform the optimization and use GAWO to optimize the feature weights for MOSES, and compare our proposition with the different optimization algorithms.

7. References

- [1] R. Zens, F. J. Och, and H. Ney, "Phrase-based statistical machine translation," in *Annual Conference on Artificial Intelligence*. Springer, 2002, pp. 18–32.
- [2] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, 2003, pp. 48–54.
- [3] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2003, pp. 160–167.
- [4] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*. Association for Computational Linguistics, 2007, pp. 177–180.
- [5] E. Hasler, B. Haddow, and P. Koehn, "Margin infused relaxed algorithm for mooses," *The Prague Bulletin of Mathematical Linguistics*, vol. 96, pp. 69–78, 2011.
- [6] M. Hopkins and J. May, "Tuning as ranking," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 1352–1362.
- [7] V. Kocur and O. Bojar, "Particle swarm optimization submission for wmt16 tuning task," in *Proceedings of the First Conference on Machine Translation*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 518–524. [Online]. Available: <http://www.aclweb.org/anthology/W/W16/W16-2344>
- [8] A. Douib, D. Langlois, and K. Smaili, "Genetic-based decoder for statistical machine translation," in *Springer LNCS series, Lecture Notes in Computer Science*, Dec. 2016. [Online]. Available: <https://hal.inria.fr/hal-01336546>
- [9] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM Computing Surveys (CSUR)*, vol. 45, no. 3, p. 35, 2013.
- [10] S. Binitha, S. S. Sathya *et al.*, "A survey of bio inspired optimization algorithms."
- [11] G. Neubig and T. Watanabe, "Optimization for statistical machine translation : survey," *Computational Linguistics*, pp. 1–54, 2016.
- [12] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.
- [13] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational linguistics*, vol. 29, no. 1, pp. 19–51, 2003.
- [14] A. Stolcke, J. Zheng, W. Wang, and V. Abrash, "SRILM at sixteen: Update and outlook," in *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop*, vol. 5, 2011.
- [15] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proceedings of Association for Machine Translation in the Americas*, vol. 200, no. 6, 2006.
- [16] S. Banerjee and A. Lavie, "METEOR: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, vol. 29, 2005, pp. 65–72.

Statistical Machine Translation from Arab Vocal Improvisation to Instrumental Melodic Accompaniment

Fadi Al-Ghawanmeh¹, Kamel Smaili²

¹Music Department, University of Jordan, Jordan

²SMarT Group, LORIA, F-54600, France

¹f_ghawanmeh@ju.edu.jo, ²kamel.smaili@loria.fr

Abstract

Vocal improvisation is an essential practice in Arab music. The interactivity between the singer and the instrumentalist(s) is a main feature of this deep-rooted musical form. As part of the interactivity, the instrumentalist recapitulates, or translates, each vocal sentence upon its completion. In this paper, we present our own parallel corpus of instrumentally accompanied Arab vocal improvisation. The initial size of the corpus is 2779 parallel sentences. We discuss the process of building this corpus as well as the choice of data representation. We also present some statistics about the corpus. Then we present initial experiments on applying statistical machine translation to propose an automatic instrumental accompaniment to Arab vocal improvisation. The results with this small corpus, in comparison to classical machine translation of natural languages, are very promising: a BLEU of 24.62 from Vocal to instrumental and 24.07 from instrumental to vocal.

Index Terms: Arab music, Statistical machine translation, Automatic accompaniment, Maqam, Mawwal.

1. Introduction

Vocal improvisation is a primary musical form in Arab music. It is called Mawwal in the eastern part of the Arab world and istikhbar in the Maghreb. It is a non-metric musical practice that shows the vocalist's virtuosity when singing narrative poetry. It is tightly connected to the sense of *saltanah*, or what can be referred to as modal ecstasy. In performance, an instrumentalist sets the stage for the singer by performing an improvisation on the given Maqam of the Mawwal. Then, the aesthetic feedback loop between the singer and the accompanying instrumentalists goes on. The Instrumentalists interact with the singer by playing along with him or her throughout every vocal sentence, then by recapitulating that sentence instrumentally upon its completion [1][2]. The audience takes part of this loop of aesthetic feedback especially by reacting to performers' expressiveness and virtuosity. This can be expressed by clapping or other means of showing excitement. Early contributions toward automating the instrumental musical accompaniment started in the mid-eighties [3][4]. However, researching automatic accompaniment in the context of Arab music started just recently [5], and has not yet been introduced to the capabilities and complexities of machine learning. Toward proposing an improved automatic accompaniment to Arab vocal improvisation, we stud-

ied the part of the accompaniment in which the instrumentalist recapitulates, or translates, the singer's musical sentence upon completion. To handle this challenge, we imagined it as a statistical machine translation problem, and then make use of techniques previously used in computational linguistics. Accordingly, our experiments require a parallel corpus consisting of vocal sentences and corresponding instrumental responses. Building our own corpus has been a necessity due to the lack of available transcriptions of accompanied Arab improvisations, also because selecting accompanied improvisations from the web and transcribing them automatically can be challenging for a variety of reasons. In our work we applied automatic transcription indeed, but on our own recordings performed by our singers and instrumentalists in equipped recording rooms. This was to ensure decent machine transcription. The remaining of this paper is organized as follows: we present related work in section two. In section three we discuss the idea of looking at the challenge of automating the melodic accompaniment from the perspective of statistical machine translation. In section four we present our corpus, and we apply machine translation experiments on it in section five, then results are presented in section six.

2. Related Works

Several harmonic accompaniment models have been proposed for different musical styles, such as jazz [6] and chorale style [7]. More generic models were also proposed, such as [8], which considered rock and R&B, among others. Several techniques were applied in the context of harmonic accompaniment, such as musical knowledge, genetic algorithms, neural networks and finite-state methods [9]. Fewer contributions considered non-harmonic accompaniment, including [5] and [10], who proposed Arab and Indian style melodic accompaniment, respectively. These last models used musical knowledge rather than machine learning methods. The model in [5] suggested a knowledge-based accompaniment to Arab vocal improvisation, Mawwal. The melodic instrumental accompaniment lines were very simple and performed slightly modified, or simplified, versions of vocal figures, all in heterophony with the vocal improvisation. Then and upon completion of each vocal figure, there was an instrumental imitation that repeated a full or partial parts of the vocal figure at a speed that could vary slightly from the speed of the vocal. In [11] and when analyzing scores of vocal improvisations along with correspond-

ing oud accompaniment, it was illustrated that, although at times the melodic lines of the instrumental accompaniment might follow the progression of the vocal lines, the particular melodic contour might twist in a way that is challenging to model. Indeed, such results encourage experimenting with corpus-based approaches to improve the automatic accompaniment. In [12] a corpus for arab-Andalusian music was built for computational musicology. The corpus consisted of audio materials, metadata, lyrics and scores. The contribution accented on the importance of the task of determining the design criteria according to which corpora are built. In [13] a research corpus for computational musicology was presented and consisted of audio and metadata for flamenco music. The contribution stressed on the idea that the distinctiveness of melodic and rhythmic elements, as well as its improvised interpretations and diversity in styles, are all reasons making flamenco music still largely undocumented. In [14] a parallel corpus of music and corresponding lyrics was presented. Crowdsourcing was used to enhance the corpus with notes on six basic emotions, annotated at line level. Early experiments showed promising results on the use of such corpus for song processing, particularly on emotion classification.

3. Melodic accompaniment and language models

Statistical Language Models work toward estimating the distribution of a variety of phenomena when processing natural languages automatically. These models seek regularities as a mean to improve the performance of applications [15]. In this contribution we investigated applying techniques common in statistical machine translation to handle the problem of automating the accompaniment to Arab vocal improvisation. In other words, we investigated translating the vocal improvisation into an instrumental accompaniment. We handled this translation problem sentence by sentence. Each vocal idea – whether as short as a motive or as long as a sentence – was considered a distinct musical sentence. The same was applied to instrumental responses; each response to the singer’s previous sentence was considered one instrumental sentence. Indeed, In the Mawwal practice, the singer separates vocal sentences with relatively long rests, and accompanying instrumentalists fill these rests by recapitulating the singer’s previous sentence. This type of instrumental response is referred to as “tarjama,” literally meaning “translation” [2].

In general, each musical sentence consists of several musical notes, and each note has two main features: pitch and duration. In our proposed approach we represented them as scale degree and quantized duration, respectively. Section 5 justifies this choice of representation with further clarification. For each sentence, whether vocal or instrumental, we considered the degree as an element and the quantization step as another element. Elements might also be called words, as in natural languages. Figure 1 shows a score of a musical idea (or sentence, as in natural languages). It is a descending four-note motive in the maqam bayati that has its tonic on the note *D*. So the scale degrees of this sentence, respectively, are: 3rd degree, 2nd



Figure 1: Example of a short musical idea in maqam bayati

degree, 1st degree and 1st degree (one octave lower). In our approach we neither document nor process musical sentences in their traditional graphical music transcription. We rather use textual representations so we can apply statistical techniques common in natural languages processing directly to text files. Now for the textual representation of the musical idea, or sentence, in figure one; the first two elements are (dg_3) and (dr_6), both belong to the first note and tell its scale degree and quantized duration, respectively. This means that the scale degree of this note is 3, and the duration is of rank 6. The full textual sentence for this musical sentence is: (dg_3)(dr_6)(dg_2)(dr_3)(dg_1)(dr_5)(dg_1)(dr_8).

4. The corpus

We built our own corpus with initial size of 2779 parallel sentences (vocal and instrumental). The goal is to use it to construct a statistical language model and apply a statistical machine translation paradigm. In this section we justify the need for building our own corpus and explain the procedure of building it. We also present some statistics about our corpus.

4.1. Why build it ourselves?

There are two main reasons that led us to build the corpus ourselves. Firstly, there is a lack of available transcriptions of Arab vocal improvisation, and it is much more difficult to find instrumentally accompanied improvisations. This is while taking into consideration that machine learning usually needs thousands of musical figures, not tens nor hundreds even. Secondly, although there are plenty of recordings of accompanied Mawawel (plural of Mawwal) available on several audio- and video-sharing websites, transcribing such Mawawel automatically is very challenging for a variety of reasons, including:

- The challenge of automatically transcribing the vocal improvisation with several instrumental melodic lines that are improvising accompaniment in a non-metric context.
- This musical form is highly interactive; so clapping and shouting from the audience can make the process more challenging.
- Arab music has many different Maqamat, and the same Maqam can have differences in microtonal tuning across different regions, especially for neutral tones. It is also common for the Mawwal to include modulations from a particular Maqam to others. Transcribing unknown audio files would

require a robust Maqam-finding algorithm. This is a different research problem that is tackled, yet not completely solved, by other researchers [16]. Indeed, automatically selecting and transcribing quality Mawaweel performances with instrumental accompaniment from YouTube and other online sources is a research challenge that needs further research. Unfortunately this was not within the scope of this project. For the reasons above, neither relying on available transcriptions nor transcribing Mawaweel from the Internet could have been a viable solution for building our parallel corpus at this time. We therefore decided to build our own corpus with our own singers, MIDI keyboard instrumentalists, and equipped recording rooms. Standardizing the recording process allowed us to avoid the issue of transcription quality in this research.

4.2. Procedure of building the corpus

To build the parallel corpus, we decided to use live vocal improvisation and Arab keyboard accompaniment. Indeed, the keyboard can emulate Arab instruments to a sufficient degree, and many singers today are accompanied by keyboardists rather than acoustic instruments. Moreover, transcribing keyboard accompaniment has perfect accuracy. This is because we only export the MIDI file that includes the transcription details, such as pitch and duration, as opposed to applying signal processing tasks to convert audio to transcription. In the latter approach, accuracy is decent, yet not perfect. In other words, when we sequence a MIDI score derived from a keyboard instrument, we hear the exact transcribed performance, but when we sequence a score of automatically transcribed audio, we are more likely to hear a deformed version of the original performance. Our choice reproduced the real-life scenario of the desired Mawwal automatic accompaniment, where the input is a vocal signal transcribed with a decent, yet not perfect, accuracy, and the output is an instrumental accompaniment that recapitulates the vocal input and its score is generated and reproduced audibly with perfect accuracy. Accordingly, building instrumental corpora using MIDI instruments would allow for incorporating instrumental accompaniment signals without deformity caused by transcription inaccuracy.

4.3. Corpus statistics

Statistics on the parallel corpus as a whole are presented in Table 1. As shown in the table, the vocal improvisation is in general longer than the instrumental accompaniment. This is although the number of instrumental notes is bigger. This is normal because the keyboard instrument imitated a plucked string instrument, the oud. Thus, the sound does not sustain for a long time, and this requires the instrumentalist to keep plucking in order to keep the instrument sounding. For both vocal and oud, the ranges of durations of notes are very wide. The table also shows that the overwhelming majority of vocal sentences lay within one octave; also half of the instrumental sentences lay in this pitch range. Table 2 presents corpus statistics at sentence level. For both vocal and instrumental sentences, it is clear that the sentence length

	Vocal	Instrumental
Total duration	17907 s	13787 s
Note count	35745	55667
Total number of sentences	2779	2779
Percentage of sentences with tone range within octave	83.62	49.44
Maximum note duration	7.7 s	4 s
Minimum duration	0.14 s	0.002 s
Mean of durations	0.5 s	0.24 s
STD of durations	0.45	0.21

Table 1: Statistics on the parallel corpus as a whole

may vary extremely. The sentence can be as short as one note or as long to have tens of notes.

	Vocal corpus	Instrumental
Maximum note count	82	140
Minimum note count	1	1
Averages note count	12.86	20.03
STD of note count	10.70	17.96

Table 2: Statistics on the parallel corpus within one sentence

5. Data representation

The development of quality NLP models requires very large corpora. Our corpus, however, is both small and diverse. It is important, then, to represent this musical data with minimal letters and words from our two proposed languages, vocal improvisation and instrumental response. Yet it is also crucial that such minimization not deform the essence of the musical data. We analyze two main musical elements in this corpus, pitch and duration, and represent them as scale degree and quantized duration. The following two sub-sections discuss this process in detail.

5.1. Scale degree

Our corpus draws from a wide variety of Maqamat (musical modes), including Maqamat with neutral tones (tones with $\frac{3}{4}$ interval), and transpositions of Maqamat to less keys. Furthermore, the pitch range of both the vocal improvisation and the instrumental accompaniment can exceed two octaves. When using pitches as letters in our proposed language, the total count of letters can exceed 48 (24 pitches per octave with a minimum interval of $\frac{1}{4}$). When using pitch-class representation, which equates octaves, the total count of letters does not exceed 24 pitches. This number remains high relative to the small size of the corpus. Given this issue, and the complication of incorporating different Maqamat in varying keys, we decided to use scale degree representation. Arab Maqamat are often based on seven scale degrees, allowing us to have the total number of letters as low as seven. One drawback to this method, however, is the inability to distinguish accidentals, the pitches that deviate from the given Maqam. Applying this configuration to the

automatic transcriber of vocal improvisation, however, allows for a significantly improved transcription quality [10] that outweighs the necessity to track accidentals.

5.2. Quantized duration

Here we present two histograms of note durations, one for vocal improvisation and the other for oud accompaniment. Analyzing the histograms helps determine the best total number of quantization steps, and also the duration range of each step. We need to have as few steps as possible in order to have better translation results, but it is crucial to retain the quality of the translation. Figure 2 shows the histogram of note durations of the vocal improvisation. We adjusted the value of the pin size to 0.139 seconds, and this is the minimum note duration (MND) in our adopted solution for the automatic transcription of vocal improvisation. Figure 3 depicts the percentage of notes located within or below each pin in the vocal improvisation. As shown in this figure, 89.3% of the note durations are within or below the first 7 pins. The remaining durations, which are relatively very long, are concentrated along other upper pins. It therefore follows to group these long (upper) durations into two bigger pins, each of which holds about half of these long durations. While taking into consideration that the first pin is empty because no note can be below the MND of the transcriber, the total count of used pins, or language letters, for the vocal corpus is 8.

Figure 4 shows the histogram of note durations of in-

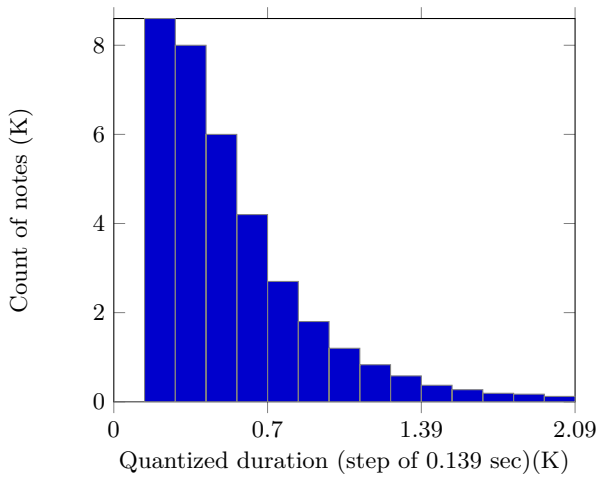


Figure 2: Note durations of the vocal improvisation

strumental accompaniment. We adjusted the value of the pin size to 0.07 second. This is half of the vocal pin size, because in our corpus, the average duration of oud notes is half of the average duration of vocal notes. Figure 5 illustrates the percentage of notes located within or below each pin in the instrumental accompaniment. As can be noticed from the figure, about 89.9% of the note durations are within or below the first 6 pins. The remaining durations, the relatively very long ones, are concentrated along other upper pins. We group these long durations into two bigger pins, each of which incorporates about half of these long durations. Accordingly, the total count of used pins, or language letters, for the oud corpus is 8.

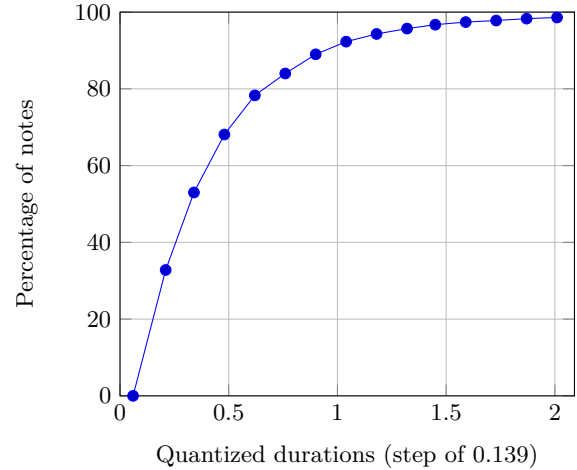


Figure 3: Percentage of vocal notes with durations below or equal each quantization step

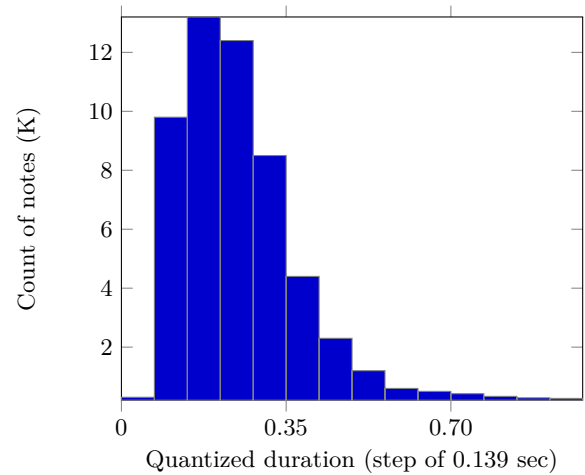


Figure 4: Note durations of instrumental accompaniment

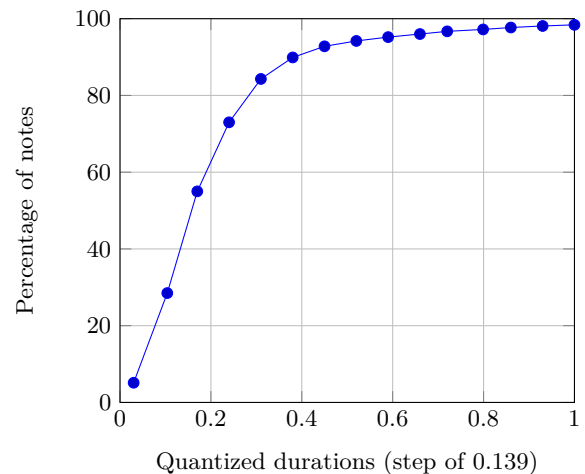


Figure 5: Percentage of instrumental notes with durations below or equal each quantization step

6. Machine Translation Experiments

Machine translation has been used to translate improvisation in both sides Vocal to Instrumental and Instrumental to vocal. In order to find the best model, we tested several representations of the music format. The MT system is a classical one with default settings: bidirectional phrase and lexical translation probabilities, distortion model, a word and a phrase penalty and a trigram language model. For the development and the test we used corpora of 100 parallel sentences for each of them. We used the bilingual evaluation understudy measure (BLEU)[17] to evaluate the quality of the translation. The formats and the BLEU scores are given In Table 3. Each format has different settings of three types of choice:

- Score reduction: this means that the music score was simplified using the formula in reference [5] in order to make the musical sentences shorter (with less notes). We used two representations for the reduced score:
 - Reduced Sustain: means that each unessential note was removed and its duration was added to its previous essential note, i.e., sustaining the essential note.
 - Reduced Silence: means that adjacent unessential notes were replaced by anew silent note that incorporates the durations of these unessential notes.
 - Unreduced: means no score reduction was applied. Apparently score reduction did not give good results, possibly because the reduction oversimplifies the patterns of melodic sentences and makes regularity ambiguous.
- Merging adjacent similar notes:
 - Merged: replace each two similar adjacent notes by one longer note to minimize the size of the musical sentences.
 - Unmerged: do not apply merging adjacent similar notes.
- Note representations:
 - Scale degree
 - Quantized duration
 - Scale degree and quantized duration

The best results have been achieved by merging adjacent similar notes, but without applying score reduction. Results are promising as the BLEU is 19.03. We also listened to the automatic accompaniment, and we believe it does have potential. Better BLEU score for this format was achieved when considering only one part of the musical information: either the duration or the scale degree, results were 21.27 and 24.62, respectively. The results of translating features separately (degrees alone and durations alone) could not be used to create accompaniment sentences, or translations, because creating a music notation need durations and degrees to have equal

count. However, when separating the vocal sentence before translation into two parts, the number of resulting instrumental durations after translation does not necessarily equal the number of scale degrees. For example, when applying separated translation on a vocal sentence of 20 notes, i.e. 20 scale degree and 20 durations, the count of resulting instrumental translation can be 28 scale degrees and 32 durations. We cannot make a meaningful music notation in this case. Nevertheless, the results of translating musical features separately give an idea on where to apply more improvement in future research.

7. Conclusions

As part of efforts to improve the automated accompaniment to Arab vocal improvisation (Mawwal), in this contribution we considered the type of melodic accompaniment in which the instrumentalist(s) responses to, or translates, each vocal sentence after its completion. We built a relatively small parallel corpus; vocal and instrumental. We explained why we needed to construct this corpus ourselves. Then, we discussed data representation, also some statistics gathered from the corpus. After that we experimented with statistical machine translation. Results were positively surprising with a BLEU score reaching up to 24.62 from Vocal to instrumental, also 24.07 from instrumental to vocal. In addition, listening to translated music assured that this approach of automatic accompaniment is promising. Future work will include expanding the parallel corpus and introducing subjective evaluation side by side with the objective BLEU.

8. Acknowledgements

The authors acknowledge financial support of this work, part of TRAM (Translating Arabic Music) project, by the Agence universitaire de la Francophonie and the Arab Fund for Arts and Culture (AFAC).

Format of data	Vocal → Oud	Oud → Vocal
Unreduced Unmerged Scale Degree and quantized Duration	7.87	14.01
Reduced Merged Sustain Scale Degree	7.95	11.25
Reduced Merged Silence Scale Degree and quantized Duration	9.21	7.30
Reduced Merged Silence Scale Degree	9.94	15.92
Reduced Merged Sustain Scale Degree and quantized Duration	11.58	9.07
Reduced Merged Silence quantized Duration	14.10	8.40
Unreduced Unmerged quantized Duration	15.66	18.76
Unreduced Unmerged Scale Degree	15.66	24.41
Reduced Merged sustain quantized Duration	16.93	11.16
Unreduced Merged Scale Degree and quantized Duration	19.03	9.66
Unreduced Merged quantized Duration	21.27	22.38
Unreduced Merged Scale Degree	24.62	24.07

Table 3: BLEU score for each format data

9. References

- [1] A. J. Racy, “Improvisation, ecstasy, and performance dynamics in arabic music,” *In the course of performance: Studies in the world of musical improvisation*, pp. 95–112, 1998.
- [2] “Arabic musical forms (genres),” 2007. [Online]. Available: <http://www.maqamworld.com/forms.html>
- [3] R. B. Dannenberg, “An on-line algorithm for real-time accompaniment,” in *ICMC*, vol. 84, 1984, pp. 193–198.
- [4] B. Vercoe, “The synthetic performer in the context of live performance,” in *Proc. ICMC*, 1984, pp. 199–200.
- [5] F. Al-Ghawanmeh, “Automatic accompaniment to arab vocal improvisation “mawwāl”,” Master’s thesis, New York University, 2012.
- [6] D. Martín, “Automatic accompaniment for improvised music,” Ph.D. dissertation, Master’s thesis, Département de technologies de l’information et de la communication, Universitat Pompeu Fabra, Barcelone, 2009.
- [7] J. Buys and B. v. d. Merwe, “Chorale harmonisation with weighted finite-state transducers,” in *23rd Annual Symposium of the Pattern Recognition Association of South Africa*, 2012.
- [8] I. Simon, D. Morris, and S. Basu, “Mysong: automatic accompaniment generation for vocal melodies,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2008, pp. 725–734.
- [9] J. P. Forsyth and J. P. Bello, “Generating musical accompaniment using finite state transducers,” in *16th International Conference on Digital Audio Effects (DAFx-13)*, 2013.
- [10] P. Verma and P. Rao, “Real-time melodic accompaniment system for indian music using tms320c6713,” in *VLSI Design (VLSID), 2012 25th International Conference on*. IEEE, 2012, pp. 119–124.
- [11] F. Al-Ghawanmeh, M. Al-Ghawanmeh, and N. Obeidat, “Toward an improved automatic melodic accompaniment to arab vocal improvisation, mawwāl,” in *Proceedings of the 9th Conference on Interdisciplinary Musicology-CIM14*, 2014, pp. 397–400.
- [12] M. Sordo, A. Chaachoo, and X. Serra, “Creating corpora for computational research in arab-andalusian music,” in *Proceedings of the 1st International Workshop on Digital Libraries for Musicology*. ACM, 2014, pp. 1–3.
- [13] N. Kroher, J.-M. Díaz-Báñez, J. Mora, and E. Gómez, “Corpus cofla: a research corpus for the computational study of flamenco music,” *Journal on Computing and Cultural Heritage (JOCCH)*, vol. 9, no. 2, p. 10, 2016.
- [14] C. Strapparava, R. Mihalcea, and A. Battocchi, “A parallel corpus of music and lyrics annotated with emotions,” in *LREC*, 2012, pp. 2343–2346.
- [15] R. Rosenfeld, “Two decades of statistical language modeling: Where do we go from here?” *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1270–1278, 2000.
- [16] M. A. K. Sağun and B. Bolat, “Classification of classic turkish music makams by using deep belief networks,” in *INnovations in Intelligent SysTems and Applications (INISTA), 2016 International Symposium on*. IEEE, 2016, pp. 1–6.
- [17] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

Image2speech: Automatically generating audio descriptions of images

Mark Hasegawa-Johnson¹, Alan Black², Lucas Ondel³, Odette Scharenborg⁴, Francesco Ciannella²

1. University of Illinois, Urbana, IL USA
2. Carnegie-Mellon University, Pittsburgh, PA USA
3. Brno University of Technology, Brno, Czech Republic
4. Centre for Language Studies, Radboud University, Nijmegen, Netherlands

Abstract

This paper proposes a new task for artificial intelligence. The image2speech task generates a spoken description of an image. We present baseline experiments in which the neural net used is a sequence-to-sequence model with attention, and the speech synthesizer is clustergen. Speech is generated from four different types of segmentations: two that require a language with known orthography (words and first-language phones), and two that do not (pseudo-phones and second-language phones). BLEU scores and token error rates indicate that the task can be performed with better than chance accuracy. Informal perusal of the output (phone strings, word strings, and synthesized audio) suggests that the audio contains complete, intelligible words organized into intelligible sentences, and that the most salient errors are caused by mis-recognition of objects and actions in the image.¹

1. Introduction

This paper proposes a new task for artificial intelligence: the generation of a spoken description of an image. The automatic generation of text is the topic of natural language processing (NLP), whereas the analysis of images is the topic of the field of computer vision. In both fields, great advances have been made on these separate topics, and recently they have been combined into a new research field: img2txt [1]. However, many of the world’s languages do not have a written form [2], therefore many people do not have access to these and other speech and NLP technologies. In this work, we propose a new research task: image2speech, which is similar to img2txt, but can reach people whose language does not have a natural or easily used written form. An image2speech system should generate a spoken description of an image directly, without first generating text.

Experiments reported in this paper convert image feature vectors into speech unit sequences. In order to implement this pipeline, four types of standard open-source software toolkits are used. First, the VGG16 [3, 4] visual object recognizer converts each image into a sequence of feature vectors. Second, the XNMT [5] machine translation toolkit accepts image feature vectors as inputs, and generates speech units as output. Third, the ClusterGen [6] speech synthesis toolkit generates audio from each speech unit sequence. Fourth, in order to train a synthetic speech voice, Clustergen needs a corpus of audio files,

each of which is transcribed using some type of discrete symbolic units; automatic speech recognition (ASR) systems based on Kaldi [7] and Eesen [8] perform this transcription.

The complete image2speech system is trained using a corpus of (image,description) pairs, where each description is an audio file. Four different types of speech units are tested, distinguished by the type of technology used to segment the audio training data. Two types of unit sequences, Words and L1-Phones (first-language phones), are generated using a same-language ASR, and would therefore never be applicable to a language without orthography, but they provide us with an upperbound performance on the image2speech task. Two other unit sequences, L2-Phones and Pseudo-Phones, are generated without transcribed same-language speech, and would therefore be applicable even in a language lacking orthography. L2-phones (second-language phones) are generated by an ASR that has been trained in some other language. Pseudo-phones are generated by an unsupervised acoustic unit discovery system.

This paper describes preliminary experiments in the image2speech task. Section 2 describes toolkits and baseline methods. Section 3.1 describes datasets. Section 3 describes methods. Section 4 presents numerical results for two img2txt baselines, and four image2speech experimental systems. Section 5 gives example image2speech outputs. Section 6 concludes.

2. Background

Imagenet [9] is an image database organized according to the WordNet [10] noun hierarchy. ImageNet currently has 14m images, provided as examples of 22k nouns. The ILSVRC (Imagenet Large Scale Visual Recognition Challenge) has been held annually since 2010. The best single-network solution in ILSVRC 2014 Sub-task 2a, “Classification+localization with provided training data,” was a 13-layer convolutional neural network (CNN) [3]; Implementations in TensorFlow ([4], used in this paper) and Keras [11] are now redistributed as the VGG16 network. VGG16 is a 13-layer CNN, followed by a two-layer fully-connected network (FCN). The last convolutional layer is composed of 512 channels, each of which is a 14×14 image; it is useful to interpret this layer as a set of $14 \times 14 = 196$ feature vectors of dimension 512. Each feature vector is the nonlinear transformation of a 40×40 -pixel sub-image, which is to say, about 3% of the original 224×224 input-image.

XNMT (the eXtensible Machine Translation Toolkit [5]) was used to implement/train the image2speech system. XNMT is specialized in the training of sequence-to-sequence neural networks, which means it reads in a sequence of inputs, and then generates a different sequence of outputs.

XNMT is based on DyNet [12], a library for the training of neural networks with variable-length inputs. Prior to DyNet, most neural network modeling toolkits assumed that every train-

¹This work was started at the Jelinek Speech and Language Technology Workshop 2017, in Pittsburgh, supported by JHU and CMU via grants from Google, Microsoft, Amazon, Facebook, and Apple. The Extreme Science and Engineering Discovery Environment (XSEDE) is supported by National Science Foundation grant number ACI-1548562. O.S. was partially supported by a Vidi-grant from NWO (276-89-003).

ing and test input is exactly the same size. DyNet introduced a new type of graph compilation: dynamic compilation, in which each layer of the neural net is represented as a compiled function, rather than a compiled data structure.

XNMT [5] is a DyNet-based library of standard components frequently re-used in neural machine translation. The library is designed so that existing components can be easily rearranged to run new experiments, and new components can be easily added. Available components are categorized as embedders (e.g., one-hot, linear, and continuous vector embedders), encoders (e.g., CNN, LSTM and pyramidal LSTM encoders), attention models (e.g., dot product, bilinear, and MLP attention models), decoders (e.g., an MLP decoder applied to the state vector of the encoder), and error metrics (e.g., BLEU, cross-entropy, word error rate). Among other applications, the flexibility of XNMT has been demonstrated in the use of attention models to select between neural and phrase-based translation probability vectors, a method that has particular utility in the translation of low-frequency content words [13].

Text-to-speech synthesis is typically a four-stage process. First, the text is converted to a graph of symbolic phonetic descriptors. Second, the duration of each unit in the phonetic graph is predicted. Third, the mel-cepstrum [14], pitch, and multi-band excitation [15] are predicted using a dynamic model such as an HMM (hidden Markov model, [16]) or RNN (recurrent neural network, [17]), or by applying separate discrete-to-continuous mapping algorithms to each frame of the synthetic utterance [6]. Fourth, the speech signal is generated by inverting the mel-cepstral transform [14], and exciting it with the specified excitation.

The Clustergen speech synthesis algorithm [6] differs from most other speech synthesis algorithms in that there is no pre-determined set of speech units, and there is no explicit dynamic model. Instead, every frame in the training database is viewed as an independent exemplar of a mapping from discrete inputs to continuous outputs, and a machine learning algorithm (e.g., regression tree [6] or random forest [18]) is applied to learn the mapping. Clustergen is particularly applicable to the problems considered in this paper because it is able to generate intelligible and pleasant synthetic voices from very small training corpora, and using an arbitrary discrete labeling of the corpus that need not include any traditional type of phoneme [19].

3. Experimental Methods

Fig. 1 gives an overview of experimental methods used in this paper. image2speech models were trained using (image, audio) pairs drawn from the Flickr8k, MSCOCO, Flickr-Audio, and SPEECH-COCO corpora. Each image is represented as a sequence of 196 vectors, each of dimension 512, created from the last convolutional layer of the VGG16 network. Audio files are converted to units via Kaldi forced alignment (Words and L1-phones) or via Eesen or AMDTK phone recognition (L2-phones and Pseudo-phones). XNMT then learns to convert a sequence of image feature vectors into a sequence of speech units, while Clustergen learns to convert speech units into audio.

The image2speech model learned by XNMT is a sequence-to-sequence model, composed of an encoder, an attender, and a decoder. The encoder is a one-layer bidirectional LSTM (implemented using XNMT’s PyramidalLSTM model), with a 128-dimensional state vector. The attender is a three-layer perceptron, implemented using XNMT’s StandardAttender model. For each combination of an input LSTM state vector and an output LSTM state vector (128 dimensions each), the attender uses a

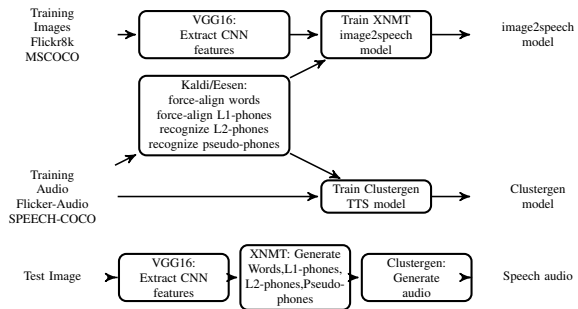


Figure 1: Experimental methods. XNMT and Clustergen models are first trained using (image, audio) pairs. A test image is then passed through VGG16, XNMT, and Clustergen to generate its audio description.

three-layer perceptron (two hidden layers of 128 nodes each) to compute a similarity score. The decoder is another three-layer perceptron (1024 nodes per hidden layer), which views an input created as the attention-weighted summation of all input LSTM state vectors, concatenated to the state vector of the output LSTM. The output of the decoder is a softmax with a number of output nodes equal to the size of the speech unit vocabulary.

3.1. Data

Experiments in this paper used two databases: the Flickr8k image captioning corpus with its associated Flickr-Audio speech corpus, and the MSCOCO image captioning corpus with its associated SPEECH-COCO speech corpus.

The Flickr8k Corpus [20] includes five text captions for each of 8000 images, as well as links to the images. Text captions were written by crowd workers, hired on Amazon Mechanical Turk. There is considerable variability among the captions provided for each image. For example, the five different captions available for the first image in the corpus (image 1000268201.693b08cb0e) are:

- A child in a pink dress is climbing up a set of stairs in an entry way.
- A girl going into a wooden building.
- A little girl climbing into a wooden playhouse.
- A little girl climbing the stairs to her playhouse.
- A little girl in a pink dress going into a wooden cabin.

In 2015, Harwath and Glass [21] proposed a data retrieval system in which speech files are used to retrieve corresponding images from a large image database, and vice versa. In order to make their proposal possible, they hired crowd workers on Mechanical Turk to read aloud the 40,000 captions from the Flickr8k corpus. The resulting set of 40,000 spoken captions is distributed as the Flickr-Audio corpus.

The Microsoft COCO (Common Objects in COntext) corpus was initially developed as an object detection corpus [22]. After initial release of the corpus, text captions of 150,000 of the images (four captions each) were distributed [23], making MSCOCO the largest database available for training img2txt systems. The SPEECH-COCO spoken transcriptions [24] were created using eight different synthetic voices, reading the MSCOCO text transcriptions. All eight synthetic voices were created from low-noise recordings of professional broadcast announcers; most

listeners can't tell that the speech is synthetic. Because the speech is synthetic, exact time alignment of the phones and words is available, and is distributed with the corpus.

Experiments in this paper did not use the entire SPEECH-COCO corpus, because we did not have enough compute time to train a neural network using the whole corpus. Instead, experiments reported in this corpus used a subset of MSCOCO with training, validation, and test corpora sized to match those of Flickr8k: 6000 training images, 1000 validation images, and 1000 test images. When an image is part of the training or validation corpus, all of its captions are used, thus experiments using the MSCOCO corpus had a training corpus containing 24,000 image-audio pairs (6000 distinct images), while the Flickr8k training corpus included 30,000 image-audio pairs (6000 distinct images).

3.2. Speech Units

Systems were trained and tested using four different types of speech units: Words, L1-Phones, L2-Phones, and Pseudo-Phones.

Words and L1-Phones are aligned to the speech2image training audio files using ASR forced alignment trained in the target language, therefore speech segmentations of this kind can only be performed in a language that has a writing system. The two databases used in this paper were transcribed in two different ways. The larger corpus, SPEECH-COCO [24], is distributed with phone transcriptions (phones in this database are transcribed using a phonetic alphabet based on X-SAMPA). The Flickr-Audio corpus [21] is not distributed with phonetic transcriptions, but text transcriptions are available [20]; from these, aligned L1-Phone transcriptions were generated using the KIT English transcription system [25].

L2-Phone transcription does not use any information about the writing system of the target language, and could therefore be used in a language that lacks any writing system. In this method, an ASR is first trained in a different language (in our case, Dutch). The L2 ASR is then used to generate a phone transcription of the target audio. In the experiments reported in this paper, an Eesen speech recognizer [8] was first used to train a Dutch ASR. Dutch phones were then mapped to English phones using linguistic knowledge only, without the use of any English writing or transcriptions, and the English-adapted Dutch ASR was used to transcribe English audio. Thus the ASR has some prior exposure to English audio, but has no knowledge about English text [26].

Pseudo-Phones were generated from the Acoustic Unit Discovery (AUD) system of [27] with two major modifications. First, the truncated Dirichlet process of [27] was replaced by a symmetric Dirichlet distribution, since, as pointed out in [28], the symmetric Dirichlet distribution provides a good and yet simple approximation of the Dirichlet Process. Second, to cope with the relatively large database, the Variational Bayes Inference algorithm originally used in [27] was replaced with the faster Stochastic Variational Bayes Inference algorithm. It was found experimentally that these modifications, while considerably speeding up the training, yield negligible drop in accuracy. The source code of the AUD model is available at <https://github.com/amdtkdev/amdtk>.

4. Results

The baseline models, with Word-sequence outputs, are standard img2txt networks, e.g., comparable to the result reported in [20]. In these networks, the output vocabulary of the network

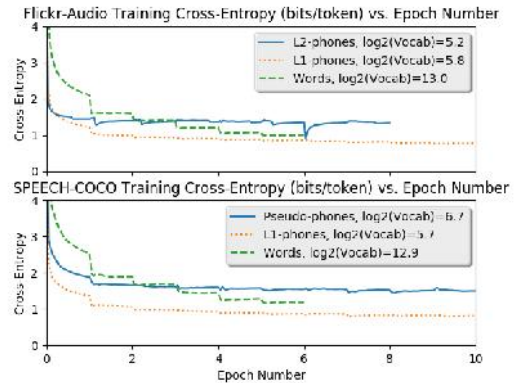


Figure 2: Training loss (cross-entropy, bits/symbol) of sequence-to-sequence networks trained to generate speech unit sequences from image features.

Table 1: BLEU scores (%) and unit error rates (UER, %) achieved in two baseline img2txt experiments (Word outputs) and four experimental image2speech experiments (L1-Phone, L2-Phone, and Pseudo-Phone), on both validation and test sets. ** means UER < chance (Student's T, Chebyshev standard error, $p < 0.001$; chance=90.2% Flickr8k, 88.7% MSCOCO).

Dataset, Targets	Validation		Test	
	BLEU	UER	BLEU	UER
Flickr8k, Words	4.7%	91.3%	3.7%	130%
Flickr8k, L1-Phones	13.7	87.9	13.7	84.9**
Flickr8k, L2-Phones	5.4	115	6.1	101
MSCOCO, Words	4.8		5.5	88.5
MSCOCO, L1-Phones	15.1		16.3	78.8**
MSCOCO, Pseudo-Ph.	2.2		1.4	123

is the set of all distinct words in the training corpus: 7993 words in Flickr8k, 7476 in MSCOCO8k.

The experimental systems generate phone outputs: L1-Phone, L2-Phone, and Pseudo-Phone. Flickr8k and MSCOCO L1-phone systems both use English phone sets, but with slightly different sizes: 54 for Flickr8k, 52 for MSCOCO. The L2-Phone system (only tested for Flickr8k) contains 38 phones. The Pseudo-Phone system (only tested for MSCOCO) was adjusted to produce 103 phones, as pseudo-phone sets of about this size have been useful in previous experiments [27].

Fig. 2 shows the training loss (cross-entropy) of sequence-to-sequence networks trained (using XNMT) to generate speech unit sequences from image features. Training loss is measured in bits per output symbol. Word-generating models start with training loss much higher than that of any phone-generating model, apparently because the number of distinct words is larger than the number of distinct phones. Training loss of the Word networks falls below those of the L2-phone and Pseudo-phone networks after some training, apparently because Words are more predictable than Pseudo-Phones or L2-Phones. The Word models never achieve training losses below those of the L1-Phone models. In fact, the Word and L1-Phone models converge to very similar endpoints, suggesting that the L1-Phone network may be learning the same thing as the Word model: it might be learning to generate a sequence of phones that always corresponds to complete Words.



Flickr8k Example #1

Ref #1: The boy +um+ laying face down on a skateboard is being pushed along the ground by +laugh+ another boy.

Ref# 2: Two girls +um+ play on a skateboard +breath+ in a court +laugh+ yard.

Network: SIL +BREATH+ SIL T UW M EH N AA R R AY D IX NG AX R EH D AE N W AY T SIL R EY S SIL.

Flickr8k Example #2

Ref #1: A boy +laugh+ in a blue top +laugh+ is jumping off some rocks in the woods.

Ref #2: A boy +um+ jumps off a tan rock.

Network: SIL +BREATH+ SIL EY M AE N IH Z JH AH M P IX NG IH N DH AX F AO R EH S T SIL.

Figure 3: Image examples from the flickr8k corpus. The table lists, for each image, two of its reference transcriptions, and the output of the L1-Phone image2speech system.

Table 1 shows BLEU scores (higher is better) and unit error rates (UER; lower is better) of four experimental systems and two baselines, measured on the validation and test sets of the Flicker-Audio and MSCOCO8k corpora. For Word-generating systems, UER=word error rate; for Phone-generating systems, UER=phone error rate. Rank-ordering of the experimental systems is roughly the same in Table 1 as in Fig. 2, though the Word-based system achieves a very poor unit error rate on the Flickr8k test corpus. Both the L2-Phone and Pseudo-Phone systems suffer UER > 100%. The L1-Phone systems, however, demonstrate unit error rates that are significantly better than chance (where “chance” is the error rate of a system that always generates the majority phone label).

Synthetic speech examples have been generated by the Clustergen algorithm for some of the L1-phone network outputs. Quality of the audio examples has not yet been quantified, but informal listening confirms the impression given by Fig. 2: generated audio is not perfectly natural, but is composed of intelligible words arranged into intelligible sentences.

5. Examples

Fig. 3 shows examples of two images from the validation subset of the Flickr8k corpus. For each image, three transcriptions are shown: two of the five available reference transcriptions (to give the reader a feeling for the difference among reference transcriptions), and one transcription generated by the L1-Phone image2speech network. The L1-Phones for Flickr8k are the ARPABET phones of [25]. As shown, the network is able to generate a phone string that is composed entirely of intelligible words, sequenced in an intelligible and semantically reasonable sentence. In these two examples, the phone strings shown can be read as English sentences that mislabel boys as men, but are otherwise almost plausible descriptions of the images: “Two men are riding a red and white race,” and “A man is jumping in the forest.”



MSCOCO Example #1

Ref #1: A group of men enjoying the beach, standing in the waves or surfing.

Ref# 2: A group of people standing on a beach next to the ocean.

Network: # @ g r u u p @ v p i i p l= s t a n d i n g o n @ b i i c h #

MSCOCO Example #2

Ref #1: A, a black and white photo of a fire hydrant near a building.

Ref #2: Aa, a fire hydrant that is out next to a house.

Network: # @ p @ @ s n= w o o k i n g @ t @ m e d i= d a u n @ n d @ r e d f a i r h a i d r @ n t #.

Figure 4: Image examples from the MSCOCO corpus. The table lists, for each image, two of its reference transcriptions, and the output of the L1-Phone image2speech system.

Fig. 4 shows similar examples from the SPEECH-COCO corpus. In the first example, the network has generated the sentence “A group of people standing on a beach,” which is perfectly correct. In the second example, the network generated “A person working at a metal down, and a red fire hydrant.” It is interesting that the neural net has noticed something about the image (the person working in the background) that was not noticed by either of the human transcribers.

6. Conclusions

This paper proposes a new task for artificial intelligence: image2speech, the task of generating spoken descriptions of input images, with no intermediate text. image2speech is trained using a database of paired images and audio descriptions. Experimental results are presented using the Flicker-Audio and SPEECH-COCO corpora. Measured UER scores are better than chance, but less than perfect. Informal perusal of results shows that image2speech is able to generate intelligible words, and to sequence them into intelligible sentences.

7. References

- [1] J-Y Pan, H-J Yang, P Duygulu, and C Faloutsos, “Automatic image captioning,” in *Proc. IEEE Internat. Conf. Multimedia and Expo (ICME)*, 2004.
- [2] G Adda, S Stüker, M Adda-Decker, O Ambourou, L Besacier, D Blachon, M Bonneau-Maynard, P Godard, F Hamlaoui, D Idiatov, G-N Kouarata, L Lamel, E-M Makasso, A Rialland, M Van de Velde, F Yvon, and S Zerbian, “Breaking the unwritten language barrier: The BULB project,” in *Proceedings of the SLTU-2016 5th Workshop on Spoken Language Technologies for Under-resourced languages*, 2016.
- [3] K Simonyan and A Zisserman, “Very deep convolutional networks for large-scale image classification,”

- <https://arxiv.org/abs/1409.1556>, 2014, Accessed: 2017-09-14.
- [4] D Frossard, “Vgg16 in tensorflow,” <https://www.cs.toronto.edu/~frossard/post/vgg16/>, 2016, Accessed: 2017-09-14.
- [5] G Neubig, “eXtensible Neural Machine Translation,” <https://github.com/neulab/xnmt>, 2017, Accessed: 2017-09-14.
- [6] AW Black, “CLUSTERGEN: A statistical parametric speech synthesizer using trajectory modeling,” in *Proc. Internat. Conf. Spoken Language Process. (ICSLP)*, 2006, pp. 1762–1765.
- [7] D Povey, A Ghoshal, G Boulianne, L Burget, O Glembek, N Goel, M Hannemann, P Motlicek, Y Qian, P Schwarz, J Silovsky, G Stemmer, and K Vesely, “The Kaldi speech recognition toolkit,” in *Proc. of ASRU*, 2011.
- [8] Y Miao, M Gowayyed, and F Metze, “EESSEN: End-to-end speech recognition using deep RNN models and WFST-based decoding,” in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015.
- [9] J. Deng, K. Li, M. Do, H. Su, and L. Fei-Fei, “Construction and Analysis of a Large Scale Image Ontology,” in *Vision Sciences Society*, 2009.
- [10] C Fellbaum, *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, MA, 1998.
- [11] F Chollet, “VGG16 model for keras,” <https://github.com/fchollet/keras/>, 2017, Accessed: 2017-09-14.
- [12] G Neubig, C Dyer, Y Goldberg, A Matthews, W Ammar, A Anastasopoulos, M Ballesteros, D Chiang, D Clothiaux, T Cohn, K Duh, M Faruqui, C Gan, D Garrette, Y Ji, L Kong, A Kuncoro, G Kumar, C Malaviya, P Michel, Y Oda, M Richardson, N Saphra, S Swayamdipta, and P Yin, “DyNet: The dynamic neural network toolkit,” <https://arxiv.org/pdf/1701.03980.pdf>, 2017, Accessed: 2017-09-14.
- [13] P Arthur, G Neubig, and S Nakamura, “Incorporating discrete translation lexicons into neural machine translation,” in *Empirical Methods in Natural Language Processing (EMNLP)*, 2016.
- [14] T Fukada, K Tokuda, T Kobayashi, and S Imai, “An adaptive algorithm for mel-cepstral analysis of speech,” in *Internat. Conf. Acoust. Speech and Sign. Process. (ICASSP)*, 1992.
- [15] T Yoshimura, K Tokuda, T Masuko, T Kobayashi, and T Kitamura, “Mixed excitation for HMM-based speech synthesis,” in *Proc. Eurospeech*, 2001, pp. 2263–2266.
- [16] K Tokuda, T Yoshimura, T Masuko, T Kobayashi, and T Kitamura, “Speech parameter generation algorithms for HMM-based speech synthesis,” in *Proc. ICASSP*, 2000.
- [17] S-H Chen, S-H Hwang, and Y-R Wang, “An RNN-based prosodic information synthesizer for Mandarin text-to-speech,” *IEEE Trans. Speech and Audio Processing*, vol. 6, no. 3, pp. 226–239, 1998.
- [18] AW Black and PK Muthukumar, “Random forests for statistical speech synthesis,” in *Proc. Interspeech*, 2015, pp. 1211–1215.
- [19] PK Muthukumar and AW Black, “Automatic discovery of a phonetic inventory for unwritten languages for statistical speech synthesis,” in *Proc. ICASSP*, 2014.
- [20] C Rashtchian, P Young, M Hodosh, and J Hockenmaier, “Collecting image annotations using amazon’s mechanical turk,” in *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, 2010.
- [21] D Harwath and J Glass, “Deep multimodal semantic embeddings for speech and images,” in *2015 IEEE Automatic Speech Recognition and Understanding Workshop*, Scottsdale, Arizona, USA, 2015, pp. 237–244.
- [22] T-Y Lin, M Maire, S Belongie, J Hays, P Perona, D Ramanan, P Dollár, and CL Zitnick, “Microsoft COCO: Common objects in context,” *Lecture Notes in Computer Science*, vol. 8693, pp. 740–755, 2014.
- [23] X Chen, H Fang, T-Y Lin, R Vedantam, S Gupta, P Dollar, and C: Zitnick, “Microsoft COCO captions: Data collection and evaluation server,” <https://arxiv.org/abs/1504.00325>, 2015, Downloaded 2017-09-14.
- [24] W Havard, L Besacier, and O Rosec, “SPEECH-COCO: 600k visually grounded spoken captions aligned to MSCOCO data set,” in *ISCA Workshop on Grounding Language Understanding (GLU2017)*, 2017.
- [25] K Kilgour, M Heck, M Müller, M Sperber, S Stüker, and A Waibel, “The 2014 KIT IWSLT speech-to-text systems for english, german and italian,” in *Internat. Worksh. Spoken Language Translation (IWSLT)*, Lake Tahoe, 2014, pp. 73–79.
- [26] O Scharenborg, F Ciannella, S Palaskar, A Black, F Metze, L Ondel, and M Hasegawa-Johnson, “Building an asr system for a low-research language through the adaptation of a high-resource language asr system: Preliminary results,” in review, 2017.
- [27] L Ondel, L Burget, and J Černocký, “Variational inference for acoustic unit discovery,” *Procedia Computer Science*, vol. 2016, no. 81, pp. 80–86, 2016.
- [28] K Kurihara, M Welling, and YW Teh, “Collapsed variational dirichlet process mixture models,” in *Proc. IJCAI*, 2007.

Speaker-dependent Selection of Cohort-utterances for Score Normalization in Speaker Recognition Systems

Ayoub BOUZIANE, Jamal KHARROUBI, Aرسالane ZARGHILI

Intelligent Systems and Applications Laboratory, Sidi Mohamed Ben Abdellah University

{ayoub.bouziane, jamal.kharroubi, arsalane.zarghili}@usmba.ac.ma

Abstract

This paper proposes a novel speaker-dependent cohort-utterances selection approach for score normalization in speaker recognition systems. The main idea of the proposed approach consists in selecting the cohort-utterances based on their scores against the target speaker.

The proposed approach was assessed for channel-dependent score normalization (C-norm) in both speaker verification and speaker identification tasks. The experiments were carried out using a speaker recognition system based on the Mel-frequency cepstral coefficients as features and the hybrid GMM-SVM approach for speaker modeling and pattern matching. The obtained results show that the proposed approach improves significantly the recognition performance of the system in the both tasks compared to the standard normalization technique (C-norm) using the overall available impostor utterances.

Index Terms: *Automatic Speaker Recognition, Speaker Verification, Speaker Identification, Score Normalization Techniques, Handset-dependent Score Normalization (H-norm), Cohort utterances Selection.*

1. Introduction

The output scores of speaker recognition systems are mainly affected by the enrollment and test conditions variabilities (e.g. the speaker model quality, the speaking style, the duration of speech utterances, the language and the channel conditions). By way of illustration, a client speaker model may tend to produce higher scores than other models, as well as, a speech utterance may tend to produce higher scores than other utterances. Accordingly, the output scores will not be as important for decision-making as their relative values compared to the output scores of impostor models or trials.

Being aware of this problem, several score normalization methods have been proposed in the literature, either for speaker- and test-dependent variability compensation [1]. The main aim of these methods consists of scaling the output scores to a global distribution where the scores vary in the same range. The commonly used score normalization approaches are the zero normalization (Z-norm) and the test-normalization (T-norm). Both techniques attempt to normalize the impostor scores into a standard normal distribution (zero mean and unitary standard deviation):

$$\hat{s} = \frac{s - \mu}{\sigma}$$

where \hat{s} and s are the normalized and original score respectively, and μ and σ are the mean and standard deviation of the impostor scores distribution. The T-norm attempts to compensate the test dependent variability by estimating the impostor statistics μ and σ from the scores of the test

utterances against a set of impostor models. On the other hand, the Z-Norm attempts to compensate the speaker dependent variability by deriving the impostor statistics μ and σ from the scores of a set of impostor speaker utterances against the target speaker model. Compared to T-Norm, the advantage of the Z-Norm lies in that the normalization parameters are estimated offline as a part of the training process, which is not the case for the T-norm where the normalization parameters should be estimated online as a part of the testing process.

The success of Z-norm and T-norm has further motivated researchers to propose several variants of them to alleviate the session variability effects [2]. Among these variants, there is the handset (H-norm, HT-norm) [3], [4] and the channel (C-norm, CT-norm) normalization methods[5]. The normalization parameters (μ and σ) of H-norm or C-Norm are estimated from the scores of a set of handset-dependent or channel-dependent impostor utterances against the target speaker model. On the other hand, the normalization parameters (μ and σ) of HT-norm or CT-norm are computed from the scores of the test utterances against a set of handset-dependent or channel-dependent impostor models. During the testing phase, the type of the handset or the channel related to the test utterance is first determined and then the corresponding normalization parameters (μ and σ) are used for score normalization.

The impostors speakers are generally selected from the development set of the system. However, it has been found in the literature that the score normalization techniques perform better when the impostor speakers are selected in a speaker-dependent way [6]–[9]. More specifically, the more the impostor speakers are similar to the target speaker, the better the recognition performance. The impostor speakers the most closely to the target speaker are often referred as cohort impostors, as well as, the impostor utterances the most closely to the voice of the target speaker are referred as cohort utterances.

Several methods have been proposed for selecting the cohort models/utterances. In [8], the cohort impostors were selected for T-norm based on some broad speaker specific information. Moreover, a speaker adaptive cohort selection method has been proposed based on a city-block distance. In the same context, Ramos-Castro et al [6], [7] proposed an approximation of Kullback–Leibler (KL) divergence as a distance measure between the target model and the impostor models. Another method based on speaker model clusters (SMCs) was used select the cohort utterances for Z-norm and the cohort models for T-norm [10].

In essence, the overall of these methods consists of selecting the cohort impostors/utterances based on the degree of similarity or the distance of their models to the target speaker's model. In this paper, we propose a novel approach for speaker-dependent cohort-utterances selection where the

cohort impostors/utterances are estimated based on the scores of their utterances against the target speaker.

The remainder of this paper is structured as follows. The first section gives a brief overview of general structure contemporary speaker recognition systems. In the second section, the proposed approach is outlined. Next, the experimental protocol, results and discussions are presented in the third section. Finally, the conclusions of the study are drawn in the final section.

2. Automatic Speaker Recognition Systems

Contemporary speaker recognition systems are generally composed of two main building blocks: The feature extraction block, the speaker modeling & scoring (pattern matching) block [11]. The feature extraction component involves the processing of speech signal and the extraction of speaker's characteristics. The modeling & scoring block aims to train a reference model for each client speaker on the basis of its extracted characteristics, as well as, to score the test utterances.

The general process of speaker recognition systems, as shown in Fig 2, involves three stages: the training phase, the enrolment phase and the testing phase. During the training phase, a large collection of speech utterances is collected from a background population of speakers, their corresponding features vectors are extracted and used to train the system hyper-parameters that reflect the general characteristics of the human speech (Universal background model, Total variability models ...). During the enrolment phase, speech samples are collected from client speakers and passed to the feature extraction component, which subsequently, generates the corresponding features vectors that summarize the characteristics of their vocal tracks. Next, the extracted feature vectors are then used together with the previously trained hyper-parameters to build or train a reference model for each

speaker. In the testing phase, the test utterance of the unknown or the claimant speaker is acquired, and the corresponding feature vectors are extracted. The extracted feature vectors are compared with the previously enrolled models (speaker identification task) or the claimed speaker's model (speaker verification task) already trained and stored during the training phase. Posteriorly, a similarity or matching score(s) is computed on the basis of this comparison. Finally, the computed score is normalized and used to make a decision about the speaker identity (in speaker identification) or acceptance rejection of the claimant identity (in speaker verification).

3. The proposed approach for Speaker-dependent Selection of Cohort-utterances

The main idea of the proposed approach consists of selecting the cohort-utterances on the basis of their scores against the target speaker. Firstly, as shown in Fig 2, the all available impostor utterances are scored against the target speaker. Next, the obtained scores are then partitioned using the K-means algorithm into two groups: (1) a group of the wolves' scores and (2) a group of sheep's scores [12]. The first group, whose center is the greater, represents the scores of the impostors who are able to imitate the target speaker. On the other hand, the second group represents the scores of the impostors who are 'normal' speakers and tend to match poorly against the target speaker. Once the impostor scores are partitioned, the sheep's scores are neglected and the wolves' scores are used to estimate the normalization parameters of the target speaker.

Compared to the other approaches where the number of cohort utterances should be defined, our proposed approach can automatically estimate the suitable number of cohort utterances for each target speaker.

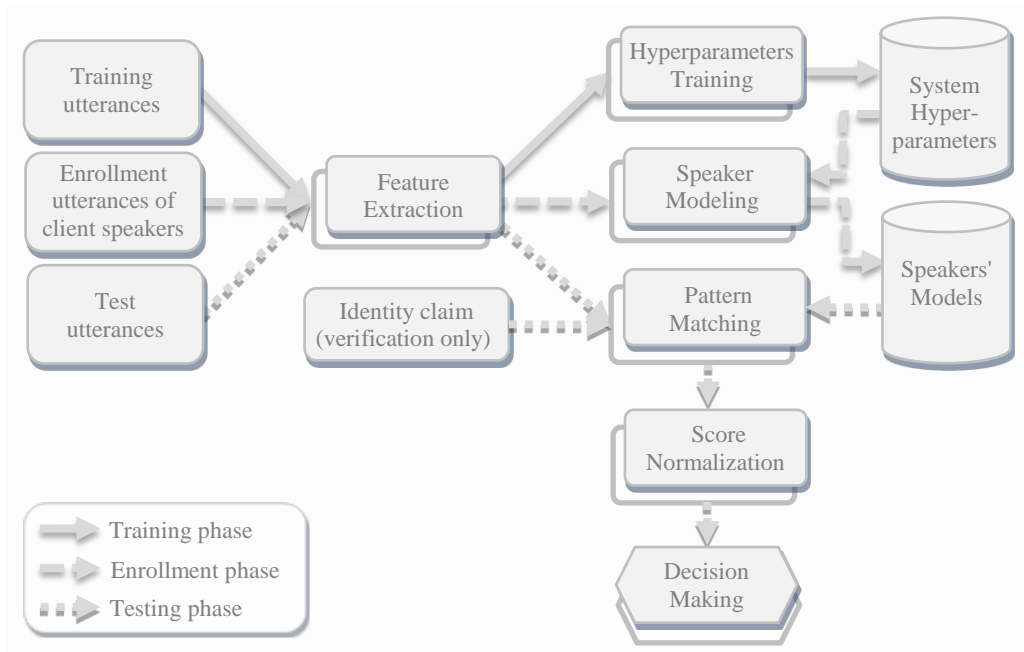


Figure 1: The basic framework and components of contemporary speaker recognition systems

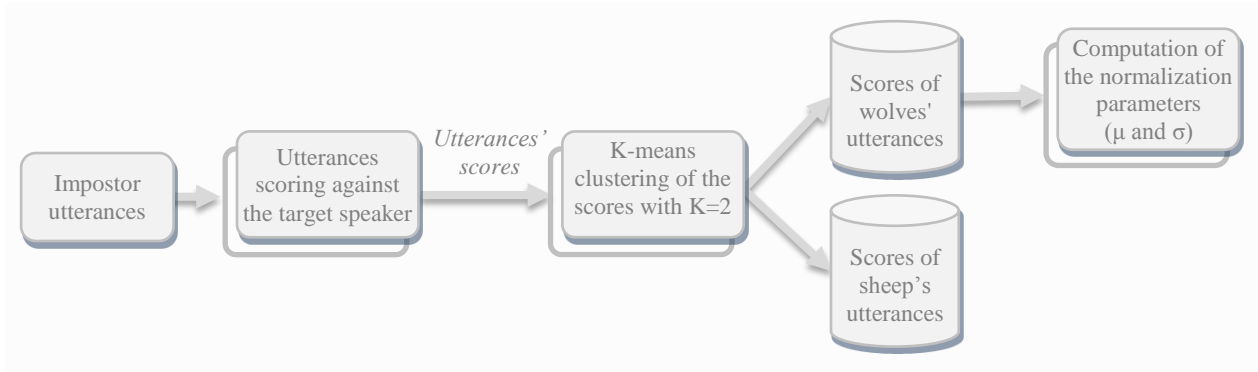


Figure 2: The proposed approach for speaker-dependent selection of cohort-utterances.

4. Experiments, Results and Discussion

4.1. Experimental Protocol

The performed experiments in this study were conducted on the THUYG-20 SRE database [13]. This database is composed of 353 speakers, collected in a controlled environment (silent office by the same carbon Microphone). The entire speech corpus was divided into three data sets: the first dataset consists of 200 gender balanced speakers (100 Male and 100 Female) and devoted to train the Universal Background Model (UBM), the second and the third data sets are composed of the same set of 153 client speakers. The first dataset comprises the enrollment speech data, whereas the second comprises the testing speech data of the 153 speakers.

The feature vectors of the overall speech utterances were extracted using the MFCC approach: the digitized speech is firstly emphasized using a simple first order digital filter with transfer function $H(z) = 1 - 0.95z$. Next, the emphasized speech signal is blocked into Hamming-windowed frames of

25 ms (400 samples) in length with 10 ms (160 samples) overlap between any two adjacent frames. Finally, 19 Mel-Frequency Cepstral Coefficients were extracted from each frame [11].

In the training phase, a universal background model (UBM) of 1024 Gaussian components was trained on the overall training data (7.5 hours of speech) using the EM algorithm. During the enrolment phase, 15 client utterances and 3768 background utterances were used to derive the GMM supervectors that feed the speaker's SVM classifiers [14], [15]. During the testing phase of the speaker verification task, the client speakers were tested against each other, resulting in total of 896,886 trials of 4 seconds and 441,558 trials of 8 seconds. As regards the identification task, the evaluation data consisted of 5862 identification tests of 4 seconds and 2886 identification tests of 8 seconds.

In order to assess the proposed approach for channel-dependent score normalization, the test utterances were passed through the opus codec (i.e. encoded-decoded) at various bitrates (8, 12, 16, 24 and 32 kbps).

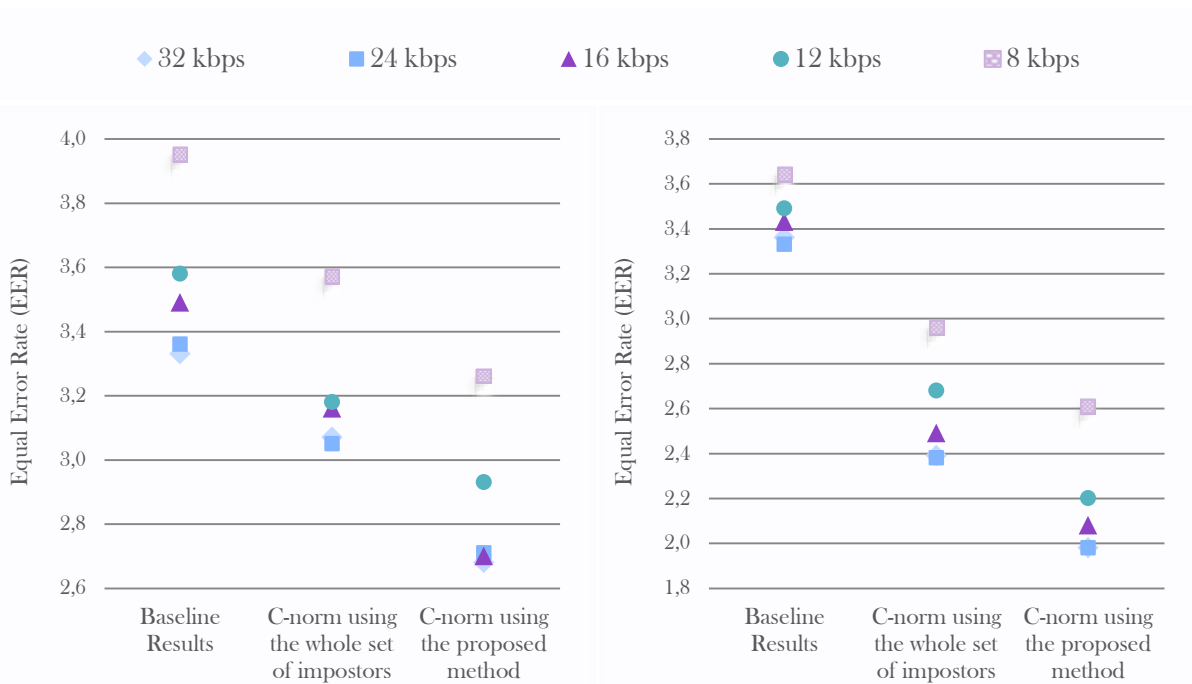


Figure 3: The obtained Equal Error Rates using test utterances of 4 seconds (left figure) and 8 seconds (right figure)

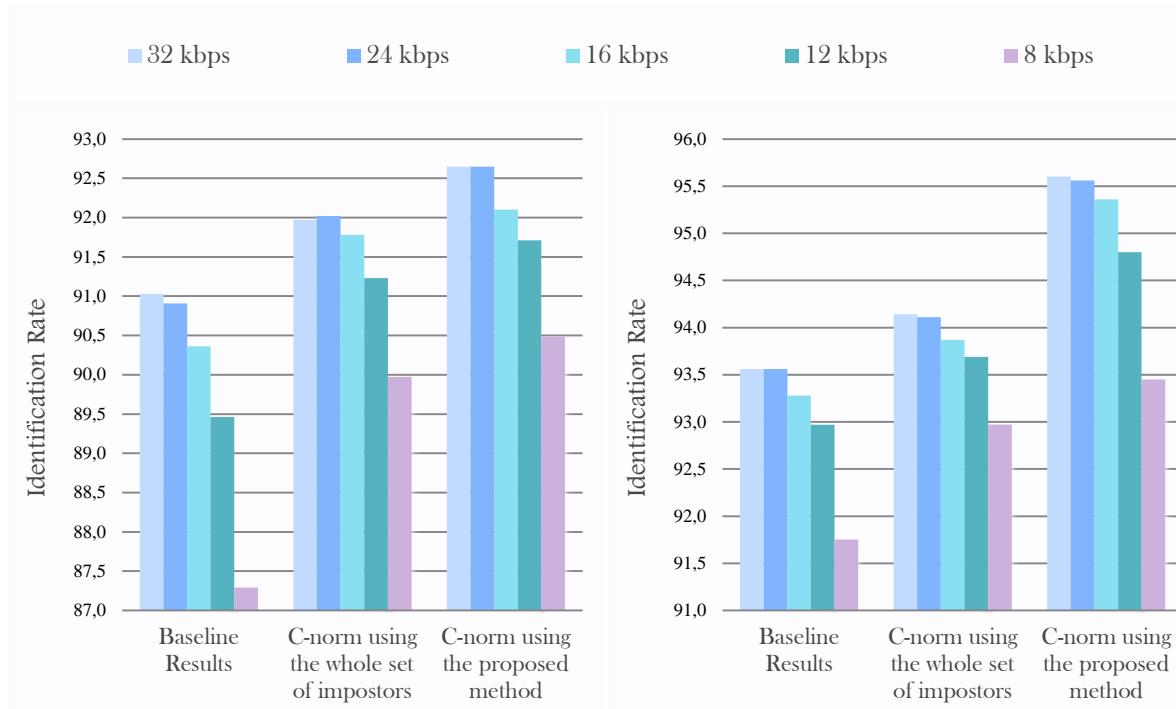


Figure 4: The obtained Identification Rates using test utterances of 4 seconds (left figure) and 8 seconds (right figure)

4.2. Results and Discussion

The obtained results in the verification and identification tasks are shown in Figs. 3 and 4, respectively. The figures illustrate the system performance in three cases: (1) no normalization has been performed (Baseline results), (2) a channel dependent normalization has been performed using the overall available impostor utterances, and (3) a channel dependent normalization has been performed using our proposed method for selecting the cohort utterances. Furthermore, the system performance in these cases was reported across the various bitrates used for transcoding the test utterances.

First and foremost, as may be seen from Fig 3 and 4, the system performance in the both tasks is decreased as the bite rate used for transcoding the test utterances decrease. Additionally, it can be noticed that the system performance has been improved by using the standard channel dependent normalization where the normalization parameters are from the overall available impostor utterances. As well as, it can be seen that our proposed approach boosts the system performance more than the standard normalization.

For instance, as it can be derived from Fig 3, the system verification performance using transcoded test utterances at 8kbps shows a relative error reduction of about 9% for 4s-test utterances and 18% for 8s-test utterances when using the standard normalization method, and a more significant of about 17% for 4s-test utterances and 28% for 8s-test utterances when using our proposed approach. As well as, the verification performance using transcoded test utterances at 32kbps demonstrates a relative error reduction of about 7% for 4s-test utterances and 28% for 8s-test utterances when using the standard normalization method, and a more

significant of about 19% for 4s-test utterances and 41% for 8s-test utterances when using our proposed approach.

In the same way, as shown in Fig 4, the system identification performance using transcoded test utterances at 8kbps shows a relative error reduction of about 21% for 4s-test utterances and 14% for 8s-test utterances when using the standard normalization method, and a more significant of about 25% for 4s-test utterances and 21% for 8s-test utterances when using our proposed approach. As well as, the identification performance using transcoded test utterances at 32kbps shows a relative error reduction of about 10% for 4s-test utterances and 9% for 8s-test utterances when using the standard normalization method, and a more significant of about 18% for 4s-test utterances and 31% for 8s-test utterances when using our proposed approach.

5. Conclusion

In this paper, a novel speaker-dependent cohort-utterances selection approach has been proposed for score normalization in speaker recognition systems. The basic idea of the proposed approach consists of partitioning the overall available impostor utterances on the basis of their scores against the target speaker into two groups. The resulted group of scores, whose center is the greater, is then used for estimating the normalization parameters.

The experimental results show that the proposed approach improves significantly the recognition performance of the system compared to the standard normalization technique using the overall available impostor utterances.

6. References

- [1] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score Normalization for Text-Independent Speaker Verification Systems," *Digit Signal Process*, vol. 10, no. 1, pp. 42–54, Jan. 2000.
- [2] F. Bimbot *et al.*, "A Tutorial on Text-independent Speaker Verification," *EURASIP J Appl Signal Process*, vol. 2004, pp. 430–451, Jan. 2004.
- [3] D. A. Reynolds, "The effects of handset variability on speaker recognition performance: experiments on the Switchboard corpus," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, 1996, vol. 1, pp. 113–116 vol. 1.
- [4] L. P. Heck and M. Weintraub, "Handset-dependent background models for robust text-independent speaker recognition," in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, vol. 2, pp. 1071–1074 vol.2.
- [5] R. B. Dunn, T. F. Quatieri, D. A. Reynolds, and J. P. Campbell, "Speaker recognition from coded speech and the effects of score normalization," in *Conference Record of Thirty-Fifth Asilomar Conference on Signals, Systems and Computers (Cat.No.01CH37256)*, 2001, vol. 2, pp. 1562–1567 vol.2.
- [6] D. Ramos-Castro, D. Garcia-Romero, I. Lopez-Moreno, and J. Gonzalez-Rodriguez, "Speaker verification using fast adaptive Tnorm based on Kullback-Leibler divergence," *Biom. Internet*, p. 49, 2005.
- [7] D. Ramos-Castro, J. Fierrez-Aguilar, J. Gonzalez-Rodriguez, and J. Ortega-Garcia, "Speaker verification using speaker- and test-dependent fast score normalization," *Pattern Recognit. Lett.*, vol. 28, no. 1, pp. 90–98, Jan. 2007.
- [8] D. E. Sturim and D. A. Reynolds, "Speaker adaptive cohort selection for Tnorm in text-independent speaker verification," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, 2005, vol. 1, p. I–741.
- [9] R. A. Finan, A. T. Sapeluk, and R. I. Dampier, "Impostor cohort selection for score normalisation in speaker verification," *Pattern Recognit. Lett.*, vol. 18, no. 9, pp. 881–888, Sep. 1997.
- [10] V. R. Apsingekar and P. L. De Leon, "Speaker verification score normalization using speaker model clusters," *Speech Commun.*, vol. 53, no. 1, pp. 110–118, Jan. 2011.
- [11] B. Ayoub, K. Jamal, and Z. Arsalane, "An analysis and comparative evaluation of MFCC variants for speaker identification over VoIP networks," in *2015 World Congress on Information Technology and Computer Applications Congress (WCITCA)*, 2015, pp. 1–6.
- [12] G. Doddington, W. Liggett, A. Martin, M. Przybocki, and D. Reynolds, "Sheep, goats, lambs and wolves: A statistical analysis of speaker performance in the NIST 1998 speaker recognition evaluation," NATIONAL INST OF STANDARDS AND TECHNOLOGY GAITHERSBURG MD, 1998.
- [13] A. Rozi, D. Wang, Z. Zhang, and T. F. Zheng, "An open/free database and Benchmark for Uyghur speaker recognition," in *Oriental COCOSDA held jointly with 2015 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE), 2015 International Conference*, 2015, pp. 81–85.
- [14] N. Dehak and G. Chollet, "Support vector GMMs for speaker verification," in *2006 IEEE Odyssey-The Speaker and Language Recognition Workshop*, 2006, pp. 1–4.
- [15] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 308–311, May 2006.

Enhancement of esophageal speech using voice conversion techniques

Imen Ben Othmane^{1,2}, Joseph Di Martino², Kais Ouni¹

¹Research Unit Signals and Mechatronic Systems, SMS, UR13ES49,
National Engineering School of Carthage, ENICarthage University of Carthage, Tunisia

²LORIA - Laboratoire Lorrain de Recherche en Informatique et ses Applications,
B.P. 239 54506 Vandœuvre-lès-Nancy, France

imen.benothmen@hotmail.fr, joseph.di-martino@loria.fr, kais.ouni@enicarthage.rnu.tn

Abstract

This paper presents a novel approach for enhancing esophageal speech using voice conversion techniques. Esophageal speech (ES) is an alternative voice that allows a patient with no vocal cords to produce sounds after total laryngectomy: this voice has a poor degree of intelligibility and a poor quality. To address this issue, we propose a speaking-aid system enhancing ES in order to clarify and make it more natural. Given the specificity of ES, in this study we propose to apply a new voice conversion technique taking into account the particularity of the pathological vocal apparatus. We trained deep neural networks (DNNs) and Gaussian mixture models (GMMs) to predict "laryngeal" vocal tract features from esophageal speech. The converted vectors are then used to estimate the excitation cepstral coefficients and phase by a search in the target training space previously encoded as a binary tree. The voice resynthesized sounds like a laryngeal voice i.e., is more natural than the original ES, with an effective reconstruction of the prosodic information while retaining, and this is the highlight of our study, the characteristics of the vocal tract inherent to the source speaker. The results of voice conversion evaluated using objective and subjective experiments, validate the proposed approach.

Index Terms: Esophageal speech, deep neural network, Gaussian mixture model, excitation, phase, KD-Tree.

1. Introduction

Speech is a spontaneous communication tool. Unfortunately, many people are unable to speak correctly. For example people who have undergone a total removal of their larynx (laryngectomees) due to laryngeal cancer or accident, cannot produce laryngeal speech sounds anymore. They need another process to produce speech sounds without a larynx. There process are for example the Artificial Larynx Transducer (ALT), the tracheo-esophageal (TE) prosthesis, or the esophageal speech.

Among them, ES is more natural when compared to the voice generated by the other process. However, the degradation of naturalness and the low intelligibility of esophageal speech is caused by some factors such as chaotic fundamental frequency, and specific noises.

Consequently, laryngeal speech is not comparable to esophageal speech.

In order to be able to integrate again laryngectomees to their individual, social and work activities, some different approaches have been proposed to enhance speech after laryngectomy surgery. In [1] and [2] a static approach has been implemented for enhancing ES in order to increase its quality. Other approaches based on the transformation of the acoustic features,

such as smoothing [3] or comb filtering [4] have been proposed. But it is difficult to improve ES by using those simple modification methods because the properties of acoustic features of esophageal speech are totally different from those of normal speech. In this paper, we propose to use voice conversion techniques to enhance ES. The goal of voice conversion (VC) is to transform the audio signal from a source speaker as if a target speaker had spoken it. We train DNNs and GMMs [20], [29], [5], [6] with the acoustic features of esophageal speech and those of normal speech. For realizing this training procedure two parallel corpora consisting of utterance-pairs of the source esophageal speech and the target normal speech are used.

This paper is structured as follows: section 2 presents the principle of the proposed technique; the obtained results from the realized tests and experiments are exhibited in section 3; section 4 presents conclusions with prospects.

2. Problem definition and related work

2.1. Esophageal speech

ES is characterized by low intelligibility, high noise perturbation and unstable fundamental frequency. When compared with laryngeal (normal) voice, ES is hoarse, has a low and chaotic F0 and therefore is difficult to understand.

Figure 1 shows an example of speech waveforms and spectrograms for normal and esophageal speech for the same sentence. We can observe that the acoustic features of normal speech are very different from those of esophageal speech.

Esophageal speech often includes some specific noisy perturbations produced through a process of generating excitation signals by releasing the air from the stomach and the esophagus afterwards pumping it into them. The intensity was determined by measuring the spectral power of the speech signal. Figure 2 shows the spectral power and the pitch variation of normal and esophageal speech. It is clear that, esophageal speech allows larger variations of intensity. However, fundamental frequency is chaotic and difficult to detect. These unstable variations of intensity and F0 are responsible of the poor audio quality of ES.

For this reason, several approaches have been proposed to improve the quality and intelligibility of the alaryngeal speech. To enhance the quality of esophageal speech, Qi attempted replacing the voicing source of esophageal speech using a LPC method [7], [8], [9]. In [10] the authors proposed to use a simulated glottal waveform and a smoothed F0 to enhance tracheo-esophageal speech (TE). In order to reduce breath and hardness of the original speech, [11] used a synthetic glottic waveform and a model for jitter and reflection reduction. For synthesizing a laryngeal voice from the whispered voice, [12] proposed a

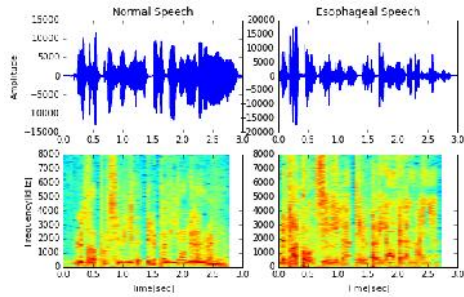


Figure 1: Example of waveforms and spectrograms of both normal and esophageal speech

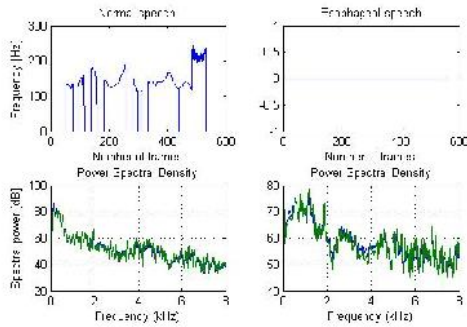


Figure 2: Example of F0 contours and spectral power of both normal and esophageal speech

Mixed-Excitation Linear Prediction (MELP) which consists in using formant structure modification and pitch estimation. This technique does not work in real time and furthermore the unvoiced phonemes are not converted. In order to produce more natural speech, [13] used a linear code excitation prediction (CELP) for estimating the pitch contours from whispered voice. Other approaches have been tempted to enhance pathological speech, based on transformations of their acoustic characteristics. Such as: combined root cepstral subtraction and spectral subtraction procedure for denoising electrolarynx speech [14]; the use of formant synthesis [3]; comb filtering [4]; the use of LPC for approximating the vocal tract filter [15].

To increase the quality of TE speech, del Pozo and Young [16] proposed to estimate the new durations of TE phones with a prediction by regression trees constructed from laryngeal data.

To enhance alaryngeal speech, Tanaka et al. [17] proposed a spectral subtraction to reduce noise and a statistical voice conversion method to predict excitation parameters. Doi et al. [18] proposed to convert alaryngeal speech to be perceived as pronounced by a speaker with laryngeal voice.

However, all the conversion methods proposed are often quite complex and in addition they can generate errors in the parameters estimation. As a result these methods produce artificial synthetic speech because of the absence of realistic excitation signals estimation.

That is why, we propose a new algorithm for enhancing ES using a new voice conversion technique, based on estimating cepstral excitation (and phase) by a search of realistic examples in the target training space.

2.2. Voice conversion algorithm based on cepstral excitation prediction

We describe in the sequel a conversion method based on cepstral excitation prediction. In the proposed algorithm, vocal tract and excitation coefficients are separately estimated. The proposed method consists of a training and conversion phase. The training phase consists of three stages: speech analysis or feature extraction, alignment and computation of a mapping function. The conversion procedure consists of conversion of cepstral parameters, excitation and phase prediction and synthesis.

2.3. Feature extraction

In order to convert esophageal speech into normal speech, we use two parallel corpora, the first one from the source speaker (esophageal speech) and the second one from the target speaker (laryngeal speech). These corpora undergo a parameterization process, which consists in extracting cepstral feature vectors. To estimate spectral/cepstral features, different features have been considered by several researchers: Log spectrum was used in [19]; Mel-cepstrum (MCEP) was used in [20], [21], [35]; Mel-frequency cepstral coefficients MFCC were used in [22]. In this work, we use real Fourier cepstrum [23]. Figure 3 details the different steps involved in transforming the speech signal into its cepstral domain representation.

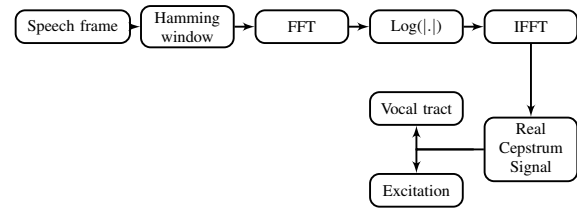


Figure 3: Bloc diagram of cepstrum based feature extraction

The linguistic contents of the speech signal are encoded into a set of coefficients:

- The first cepstral coefficient c_0
- The vocal tract cepstral vector $[c_1 \dots c_{32}]$
- The cepstral excitation $[c_{33} \dots c_{256}]$
- The phase coefficients $[p_0 \dots p_{256}]$

So we must find transformation prediction methods for the c_0 , the vocal tract vector, the excitation vector and the phase coefficients.

2.4. Alignment

Dynamic time warping DTW [24], [33] is used to find an optimal alignment between the source sequences of vectors $X=[X_1, X_2, \dots, X_s]$ and target sequences of vectors $Y=[Y_1, Y_2, \dots, Y_t]$. To align these two sequences of vectors, we must find an alignment path (A,B) where $A=[a_1, a_2, \dots, a_U]$ and $B=[b_1, b_2, \dots, b_U]$ are sequences of indices used to align X and Y. X and Y aligned are given by $X=[X_{a_1}, X_{a_2}, \dots, X_{a_U}]$ and $Y=[Y_{b_1}, Y_{b_2}, \dots, Y_{b_U}]$.

2.5. Training of the conversion function

To train the mapping function F, the aligned sequence of vocal tract vectors are used. The mapping function is estimated using a Gaussian Mixture Model (GMM) [25] (our baseline system), or a Deep Neural Network (DNN) [26].

2.5.1. Gaussian Mixture Model

The joint probability of vector z , which is the concatenation of a source vector x and its mapped target vector y , is represented as the sum of G multivariate Gaussian densities, given by:

$$p(z) = \sum_{i=1}^G \alpha_i N_i(z, \mu_i, \Sigma_i) \quad (1)$$

where $N_i(z, \mu_i, \Sigma_i)$ denotes the i -th Gaussian distribution with a mean vector μ_i and a covariance matrix Σ_i . G represents the total number of mixture components and α_i is the mixture weight:

$$\alpha_i = \frac{N_{s,i}}{N_s} \quad (2)$$

where $N_{s,i}$ and N_s are respectively the number of vectors in class i and the total number of source vectors classified. the mean vector μ_i is calculated as follow:

$$\mu_i^z = \frac{\sum_{k=1}^{N_{s,i}} z_i^k}{N_{s,i}} \quad (3)$$

The conversion function is then defined as a regression $E[y/x]$ given by formula 4:

$$F(x) = E[y/x] = \sum_{i=1}^Q p(i/x) (\mu_i^y + \Sigma_i^{yx} (\Sigma_i^{xx})^{-1} (x - \mu_i^x)) \quad (4)$$

where

$$\Sigma_i^z = \begin{bmatrix} \Sigma_i^{xx} & \Sigma_i^{xy} \\ \Sigma_i^{yx} & \Sigma_i^{yy} \end{bmatrix} \text{ and } \mu_i^z = \begin{bmatrix} \mu_i^x \\ \mu_i^y \end{bmatrix}$$

Formula 4 represents the mapping function that transforms a source cepstium vector x into its corresponding target cepstrum vector $F(x)$.

2.5.2. Deep neural network

We propose to use a Multi Layer Perceptron (MLP) [20], [29]. A DNN is a feed-forward neural network with many hidden layers. Most studies have applied a single network on spectral envelopes [21], [23]. We further apply a deep neural network in order to transform a source voice into a target voice. The goal of a DNN is to approximate some function $f(\cdot)$ defined as:

$$y = f(x, \theta) \quad (5)$$

by learning the value of the parameters θ giving the best function approximation mapping all inputs x to outputs y . For each hidden unit i , an activation function $g(\cdot)$ is used to map the inputs from the layer x_i to an outputs y_i .

$$y_i = f(x_i) \quad (6)$$

where

$$x_i = b_i + \sum_k y_k w_{ki} \quad (7)$$

and b_i is the bias of unit; k is the unit index of layer; w_{ki} is the weight of the connexion.

Recently Rectified Linear Unit ReLU activation function has become more and more popular [30], [31], [32] for its simplicity and good performance. In this work, we choose a deep feedforward network with ReLU activation function. To reduce the problem of DNN over-fitting we use the Dropout technique [36]. Dropout allows to give up units (hidden and visible) in a deep neural network. Consequently, it prevents deep neural network from over-fitting by providing a way of approximately combining exponentially many different deep neural network architectures efficiently.

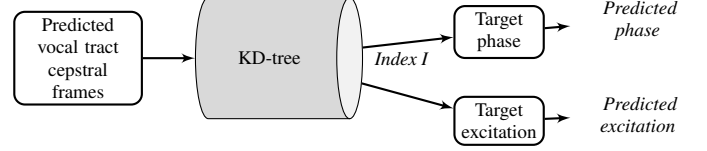


Figure 4: KD-Tree query for excitation and phase prediction

2.6. Conversion

At this stage, cepstral coefficients are extracted from the esophageal speech signal. Then only the vocal tract vectors $X_i = [X_i^1 \dots X_i^{32}]$ are converted by the previously described methods.

2.7. Synthesis

The main contribution of our approach consists in the prediction of the cepstral excitation and phase using a KD-Tree [27], [34]. As shown in Figure 4, converted vocal tract cepstral vectors are used to estimate cepstral excitation and phase coefficients.

The binary KD-Tree is constructed with concatenated target vocal tract cepstral vectors of length $N B_{frame}$. The KD-Tree is queried by the predicted vocal tract cepstral vectors previously concatenated as a frame of length $N B_{frame}$. This query provides an index I which corresponds to the nearest target vocal tract cepstral frame.

The target cepstral excitation vectors are concatenated to form a frame of length $N B_{frame}$. In the same manner the target phase vectors are concatenated.

Thereafter, index I is used as the index of a desired cepstral excitation and phase frame. Since excitation and phase are estimated, a complex spectrum is formed using magnitude and phase spectra. The spectral synthesizer we use is based on overlapping and adding the inverse Fourier transform temporal signals using OLA-FFT. To preserve the characteristics of the vocal tract of the source speaker the first cepstral packet (vocal tract packet), in one of our experiments, has not been modified at the resynthesis stage.

3. Experimental evaluations

3.1. Datasets

The Datasets have been created by 3 French speakers: two laryngectomees (PC and MH), and a speaker AL with normal voice. For each speaker 289 phonetically balanced utterances have been recorded. The speech of each corpus is sampled at 16 kHz.

3.2. Experimental conditions

All conditions of the experiments are summarized in Table 1 :

3.3. Objective evaluations

To evaluate our system, we have chosen to calculate the Log Spectral Distortion and Signal to Error Ratio (SER). The Log Spectral Distortion (LSD), via cepstrum representation becomes the cepstral distance CD [28].

$$CD(x, y) [dB] = \frac{10}{\log 10} \sqrt{\sum_k (c_k(x) - c_k(y))^2} \quad (8)$$

Table 1: *EXPERIMENTAL CONDITIONS.*

Number of GMMs	64
Window length	32 ms
Shift length	4 ms
Number of training utterances	200
Number of test utterances	20
FFT size	512
NB_{frame}	20
DNN structure	512*5
Number of epochs	100

where $c_k(x)$ and $c_k(y)$ are respectively the k-th cepstral coefficient of converted and target cepstrum vectors.

SER is represented by the following formula

$$SER [dB] = -10 * \log_{10} \frac{\sum_k \|y_k - \hat{y}_k\|^2}{\sum_k \|y_k\|^2} \quad (9)$$

where y_k and \hat{y}_k are respectively the target and converted cepstral vectors. Tables 2 and Table 3 show the different values of Signal to Error Ratio (SER) and cepstral Distance (CD) between the source and target cepstrum, then between predicted and target cepstrum.

Table 2: *SER [dB]*

Methods	PC	MH
Extracted	2.7	2.97
GMM-based method	12.33	11.39
DNN-based approach	12.99	11.80

Table 3: *CD [dB]*

Methods	PC	MH
Extracted	9.28	9.03
GMM-based method	5.37	5.53
DNN-based approach	5.29	5.28

We can observe that the cepstral vectors of esophageal speech are very different from those of normal speech. We can see also that the proposed conversion methods can take into account these large differences. More significantly, it is clear that the proposed DNN-based approach performs much better than the GMM-based method for the voice conversion task.

3.4. Perceptual evaluations

We conducted two opinion tests: one for naturalness and the other one for intelligibility. The following four types of speech samples were evaluated by ten listeners.

- ES (esophageal speech)
- GMM_CVT: The vocal tract cepstral vectors are converted using the GMM model, then the converted vocal tract cepstral vectors are used to estimate excitation and phase. The converted vocal tract vectors are used in the synthesis process.
- DNN_CVT: The vocal tract cepstral vectors are converted using the DNN model, and the converted vocal tract cepstral vectors are used in the synthesis process.

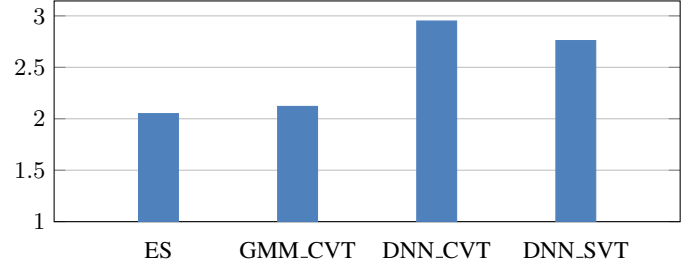


Figure 5: Mean opinion scores on naturalness

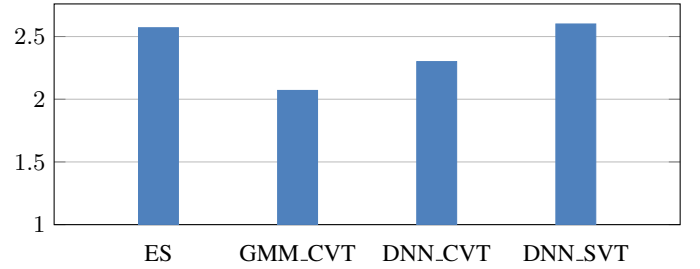


Figure 6: Mean opinion scores on intelligibility

- DNN_SVT: We use the source vocal tract cepstral vectors in the synthesis process.

Each listener evaluated 32 samples in each of the two tests. Figures 5 and 6 show the results for naturalness and intelligibility tests. The rating scale was a standard MOS (Mean Opinion Score) scale (1=bad 2=poor 3=fair 4=good 5=excellent). The results show that DNN methods perform better than GMM method and these methods are expected to have a fairly good acceptance among laryngectomees. Finally, about 70 % of the listeners participating in the subjective evaluation preferred the use of the proposed system based on preserving the source vocal tract cepstral vectors. The other listeners preferred DNN_CVT: the results provided by this method seem to them more natural than those provided by DNN_SVT. Some samples obtained by this work are presented in the following web link: [demo](#). These evaluation results show that our proposed system is very effective for improving the naturalness of esophageal speech while preserving its intelligibility.

4. Conclusions and prospects

In this article, we propose a voice conversion algorithm for esophageal speech enhancement. The originality of our approach lies in the prediction of cepstral excitation and phase using a KD-Tree. The vocal tract cepstral vectors are converted using two methods, one based on DNN and the other one on GMM, and these converted vectors are used for predicting excitation and phase. But in the synthesis stage (in experiment DNN_SVT) those converted vectors are not used in order to retain the vocal tract characteristics of the source speaker. Objective and subjective evaluations have demonstrated that our method provides significant improvements in the quality of the converted esophageal speech while preserving its intelligibility.

In a near future we intend to further increase the naturalness of the transformed esophageal speech and consequently its intelligibility.

5. References

- [1] H. Doi, K. Nakamura, T. Toda, H. Saruwatari, and K. Shikano, "Statistical approach to enhancing esophageal speech based on Gaussian mixture models", *Acoustics Speech and Signal Processing (ICASSP)*, pp. 4250-4253, 2010.
- [2] H. Doi, K. Nakamura, T. Toda, H. Saruwatari, K. Shikano, "Enhancement of Esophageal Speech Using Statistical Voice Conversion", in *APSIPA 2009*, Sapporo, Japan, pp. 805-808, Oct. 2009.
- [3] K. Matsui, N. Hara, N. Kobayashi, H. Hirose, "Enhancement of esophageal speech using formant synthesis", *Proc. ICASSP*, pp. 1831-1834, 1999-May.
- [4] A. Hisada and H. Sawada, "Real-time clarification of esophageal speech using a comb filter", *Proc. ICDVRAT*. 2002.
- [5] T. Toda, A. Black and K. Tokuda, "Voice Conversion Based on Maximum-Likelihood Estimation of Spectral Parameter Trajectory", *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 8, pp. 2222-2235, 2007.
- [6] Y. Stylianou, O. Cappé and E. Moulines, "Continuous probabilistic transform for voice conversion", *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 131-142, 1998.
- [7] Ning Bi and Yingyong Qi, "Application of speech conversion to alaryngeal speech enhancement", *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 2, pp. 97-105, 1997.
- [8] Y. Qi and B. Weinberg, "Characteristics of Voicing Source Waveforms Produced by Esophageal and Tracheoesophageal Speakers", *Journal of Speech Language and Hearing Research*, vol. 38, no. 3, p. 536, 1995.
- [9] Y. Qi, "Replacing tracheoesophageal voicing sources using LPC synthesis", *The Journal of the Acoustical Society of America* 88.3, pp 1228-1235, 1990.
- [10] Y. Qi, B. Weinberg and N. Bi, "Enhancement of female esophageal and tracheoesophageal speech", *The Journal of the Acoustical Society of America*, vol. 98, no. 5, pp. 2461-2465, 1995.
- [11] A. del Pozo and S. Young, "Continuous Tracheoesophageal Speech Repair", In *Proc. EUSIPCO*, 2006.
- [12] H. I. Trkmen and M. E. Karsligil, "Reconstruction of dysphonic speech by melp", *Iberoamerican Congress on Pattern Recognition*. Springer, Berlin, Heidelberg, 2008.
- [13] H. Sharifzadeh, I. McLoughlin and F. Ahmadi, "Reconstruction of Normal Sounding Speech for Laryngectomy Patients Through a Modified CELP Codec", *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 10, pp. 2448-2458, 2010.
- [14] D. Cole, S. Sridharan, M. Moody, S. Geva, "Application of Noise Reduction Techniques for Alaryngeal Speech Enhancement", *Proc. of The IEEE TENCON*, pp. 491-494, 1997.
- [15] B. García, J. Vicente, and E. Aramendi, "Time-spectral technique for esophageal speech regeneration", *11th EUSIPCO (European Signal Processing Conference)*. IEEE, Toulouse, France. 2002.
- [16] A. del Pozo and S. Young, "Repairing Tracheoesophageal Speech Duration", In *Proc. Speech Prosody*, 2008.
- [17] K. Tanaka, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "A hybrid approach to electrolaryngeal speech enhancement based on noise reduction and statistical excitation generation", *IEICE Trans. on Inf. and Syst.*, vol. E97-D, no. 6, pp. 14291437, Jun. 2014.
- [18] H. Doi and al. "Alaryngeal speech enhancement based on one-to-many eigenvoice conversion", *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22.1, pp 172-183, 2014.
- [19] F.-L. Xie, Y. Qian, Y. Fan, F. K. Soong, and H. Li, "Sequence error (SE) minimization training of neural network for voice conversion", in *Proc. Interspeech*, 2014.
- [20] D. Srinivas, E. V. Raghavendra, B. Yegnanarayana, A. W. Black, K. Prahallad, "Voice conversion using artificial neural networks", *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 3893-3896, Apr. 2009.
- [21] Ling-Hui Chen, Zhen-Hua Ling, Li-Juan Liu and Li-Rong Dai, "Voice Conversion Using Deep Neural Networks With Layer-Wise Generative Training", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1859-1872, 2014.
- [22] T. Nakashika, T. Takiguchi and Y. Ariki, "Voice Conversion Using RNN Pre-Trained by Recurrent Temporal Restricted Boltzmann Machines", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 580-587, 2015.
- [23] T. Nakashika, R. Takashima, T. Takiguchi, Y. Ariki, "Voice conversion in high-order eigen space using deep belief nets", pp. 369-372, 2013.
- [24] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43-49, 1978.
- [25] H. Valbret, "Système de conversion de voix pour la synthèse de parole", *Diss.* 1993.
- [26] Z.-H. Ling, S.-Y. Kang, H. Zen, A. Senior, M. Schuster, X.-J. Qian, H. Meng, L. Deng, "Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends", *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 35-52, May 2015.
- [27] S. Arya, "Nearest neighbor searching and applications", Ph.D. thesis, Univ. of Maryland at College Park, 1995.
- [28] M. M. Deza, E. Deza, "Encyclopedia of Distances", Berlin, Springer, 2009.
- [29] S. Desai, A. Black, B. Yegnanarayana, K. Prahallad, "Spectral mapping using artificial neural networks for voice conversion", *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 5, pp. 954-964, Jul. 2010.
- [30] V. Nair and G.E. Hinton, "Rectified linear units improve restricted boltzmann machines", *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010.
- [31] A. L. Maas, A. Y. Hannun, A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models", *Proc. ICML*, vol. 30, 2013.
- [32] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification", *arXiv preprint arXiv:1502.01852*, 2015.
- [33] M. Muller, "Information retrieval for music and motion", Vol. 2. Heidelberg, Springer, 2007.
- [34] K. Zhou, Q. Hou, R. Wang and B. Guo, "Real-time KD-tree construction on graphics hardware", *ACM Transactions on Graphics*, vol. 27, no. 5, p. 1, 2008.
- [35] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks", *Acoustics, Speech and Signal Processing (ICASSP)*, 2015 IEEE International Conference on. IEEE, 2015.
- [36] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting." *Journal of machine learning research* 15.1 (2014): 1929-1958.

Sentence Boundary Detection for French with Subword-Level Information Vectors and Convolutional Neural Networks

Carlos-Emiliano González-Gallardo¹, Juan-Manuel Torres-Moreno^{1 2}

¹LIA, Université d'Avignon et des Pays de Vaucluse

²École Polytechnique de Montréal

carlos-emiliano.gonzalez-gallardo@alumni.univ-avignon.fr, juan-manuel.torres@univ-avignon.fr

Abstract

In this work we tackle the problem of sentence boundary detection applied to French as a binary classification task ("sentence boundary" or "not sentence boundary"). We combine convolutional neural networks with subword-level information vectors, which are word embedding representations learned from Wikipedia that take advantage of the words morphology; so each word is represented as a bag of their character n-grams.

We decide to use a big written dataset (French Gigaword) instead of standard size transcriptions to train and evaluate the proposed architectures with the intention of using the trained models in posterior real life ASR transcriptions.

Three different architectures are tested showing similar results; general accuracy for all models overpasses 0.96. All three models have good F1 scores reaching values over 0.97 regarding the "not sentence boundary" class. However, the "sentence boundary" class reflects lower scores decreasing the F1 metric to 0.778 for one of the models.

Using subword-level information vectors seem to be very effective leading to conclude that the morphology of words encoded in the embeddings representations behave like pixels in an image making feasible the use of convolutional neural network architectures.

Index Terms: Convolutional Neural Networks, Automatic Speech Recognition, Machine Learning, Sentence Boundary Detection

1. Introduction

Multimedia resources provide nowadays a big amount of information that automatic speech recognition (ASR) systems are capable to transcribe in a very feasible manner. Modern ASR systems like the ones described in [1] and [2] obtain very low Word Error Rate (WER) for different French sources (17.10% and 12.50% respectively), leading to very accurate transcriptions that could be used in further natural language processing (NLP) tasks.

Some NLP tasks like part-of-speech tagging, automatic text summarization, machine translation, question answering and semantic parsing are useful to process, analyze and extract important information from ASR transcriptions in an automatic way [3, 4]. For this to be accomplished a minimal syntactic structure is required but ASR transcriptions don't carry syntactic structure and sentences boundaries in ASR transcriptions are inexistent.

Sentence Boundary Detection (SBD), also called punctuation prediction, aims to restore or predict the punctuation in transcripts. State of the art show that research has been done for different languages like Arabic, German, Estonian, Portuguese

and French [5, 6, 7, 8, 9]; nevertheless English is the most common one [3, 4, 5, 10, 11]. In this paper we focus on French, nevertheless, the proposed architectures and the concepts behind can be used to other languages.

There exist two different types of features in SBD and the use of each type depends of their availability and the methods that will be used. Acoustic features rely on the audio signal and the possible information that could be extracted like pauses, word duration, pitch and energy information [8, 12, 13]. Lexical features by contrast, depend on transcriptions made manually or by ASR systems, dealing to textual features like bag of words, word n-grams and word embeddings [3, 4, 6, 7].

Conditional random fields classifiers have been used in [4, 10] to predict different punctuation marks like comma, period, question and exclamation marks. In [8], adaptive boosting was used to combine many weak learning algorithms to produce an accurate classifier also for period, comma and question marks. Hand-made contextual rules and partial decision tree algorithms where considered in [7] to find sentence boundaries in Tunisian Arabic. In [6], a hierarchical phrase-based translation approach was implemented to treat the sentence boundary detection task as a translation one.

Deep neural networks were used with word embeddings in [3] to predict commas, periods and questions marks. Three different models were presented: the first one considered a standard fully connected deep neural network while the other two implemented convolutional neural network architectures. Concerning the word embeddings, 50-dimensional pre-trained GloVe word vectors were chosen to perform experiments; this embeddings use a distinct vector representation for each word ignoring the morphology of words. Che *et al.* recovered the standard fully connected deep neural network architecture presented in [3], then an acoustic model was introduced in a 2-stage joint decision scheme to predict the sentence boundary positions.

Following the scheme described in [3], we aboard the SBD as a binary classification task. The objective is to predict the associated label of a word w_i inside a context window of m words using only lexical features. Audio sources are normally used to train and test SBD models which are not reused for later applications. We want to create models that can be reutilized in further SBD work, so we approach the topic in a different manner using a big written dataset.

2. Model Description

2.1. Subword-Level Information Vectors

Subword-Level Information (SLI) vectors [14] are word embedding representations based on the continuous skip-gram model

proposed in [15] and created using the fastText library¹.

Compared to other word embedding representations that assign a distinct vector to each word ignoring their morphology [15, 16, 17, 18], SLI vectors learn representation for character n-grams and represent words as the sum of those vectors. This provides a major advantage because it makes possible to build vectors for unknown words. Nevertheless for our research we found useful the intrinsic relation between vector’s components.

For the present research we used the French pre-trained SLI vectors in dimension 300 trained on Wikipedia using fastText².

2.2. Convolutional Neural Network Models

Convolutional Neural Networks (CNN) are a type of Deep Neural Network (DNN) in which certain hidden layers behave like filters that share their parameters across space.

The most straightforward application for CNN is image processing, showing outstanding results [19, 20]. However they are useful for a variety of NLP tasks like sentiment analysis and question classification [21]; part-of-speech and named entity tagging, semantic similarity and chunking [22]; sentence boundary detection [3] and word recognition [23] between others.

The input layer of a CNN is represented by a $m \times n$ matrix where each cell c_{ij} may correspond to an image’s pixel in image processing. For our purpose this matrix represents the relation between a window of m words and their corresponding n dimensional SLI vectors. The hidden layers inside CNN consist of an arrange of convolutional, pooling and fully connected layers blocks.

2.2.1. Text matrix representation

Given the intrinsic relation between the components of SLI vectors, we think it is feasible to make an extrapolation to the existing relation between adjacent pixels of an image. This way the $m \times n$ matrix of the input layer will be formed by the context window in (1) where w_i is the word for which we want to get the prediction. The columns of the matrix will be represented by each one of n components of their corresponding SLI vectors.

$$\{w_{i-(m-1)/2}, \dots, w_{i-1}, w_i, w_{i+1}, \dots, w_{i+(m-1)/2}\} \quad (1)$$

2.2.2. CNN-A

The hidden architecture of the first model (Figure 1 (CNN-A)) is based on a model presented in [3]. It is composed of three convolutional layers (A_conv-1, A_conv-2 and A_conv-3), all three with valid padding and stride value of one. A_conv-1 has a 2x4-shape kernel and 64 output filters, A_conv-2 has a 2-shape kernel and 128 output filters and A_conv-3 has a 1x49-shape kernel and 128 output filters. After A_conv-1, a max pooling layer (A_max_pool) with 2x3-shape kernel and stride of 2x3 is staked. After the convolution phase, two fully connected layers (A_f_c-1 and A_f_c-2) with 4096 and 2048 neurons each and a final dropout layer (A_dropout) are added. The output of all convolutional, max pooling and fully connected layers are in function of RELU activations.

2.2.3. CNN-B

In our second model (Figure 1 (CNN-B)) we tried to reduce the complexity generated by the three convolution layers of

CNN-A. For this model there are only two convolutional layers (B_conv-1 and B_conv-2), both with valid padding and stride value of one. B_conv-1 has a 3-shape kernel and 32 output filters while B_conv-2 has a 2-shape kernel and 64 output filters. To downsample and centralize the attention of the CNN in the middle word of the window, a max pooling layer (B_max_pool) with 2x3-shape kernel and stride of 1x3 is implemented after B_conv-2. The final part of the CNN is formed by 3 fully connected layers (B_f_c-1, B_f_c-2 and B_f_c-3) with 2048, 4096 and 2048 neurons each and a dropout layer (B_dropout) attached to B_f_c-3. The output of all convolutional max pooling and fully connected layers are in function of RELU activations.

2.2.4. CNN-C

Finally, in our third model (Figure 1 (CNN-C)) we simplified the fully connected layers of CNN-B. The convolutional and max pool layers (C_conv-1, C_conv-2 and C_max_pool) are the same than in CNN-B. For this model, only one fully connected layer of 2048 neurons is present (C_f_c-1) which is attached to a dropout layer (C_dropout). The output of all convolutional max pooling and fully connected layers are in function of RELU activations.

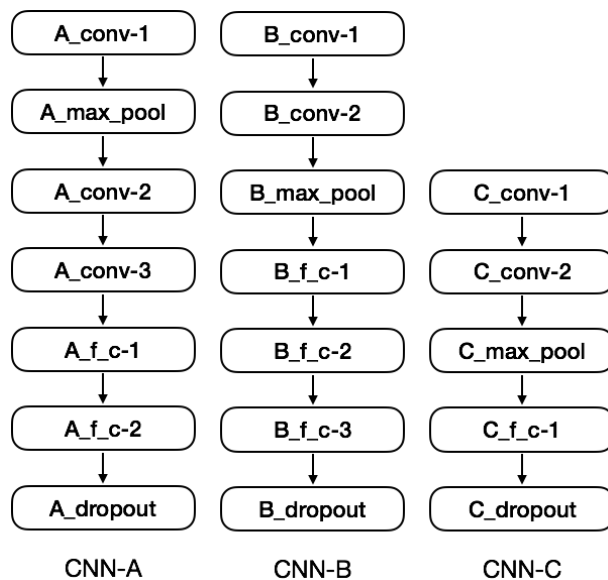


Figure 1: CNN hidden architectures

3. Experimental Evaluation

3.1. Dataset

SBD experimentation datasets normally rely on automatic or manual transcriptions to train and test the proposed systems [3, 6, 11]. As shown in Table 1, the amount of tokens is, in average 21.25k, which only 2.5k (12.48%) correspond to any punctuation mark.

In order to reuse the proposed architectures and trained models for real life ASR transcriptions and further NLP applications we opted for a big written dataset. It consists of one section of the French Gigaword First Edition³ (GW_afp) created

¹<https://github.com/facebookresearch/fastText>

²<https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md>

³<https://catalog.ldc.upenn.edu/LDC2006T17>

Table 1: *Oral datasets.*

Dataset	tokens	punctuation	percentage
[10] WSJ	51k	5k	9.8%
[10] TED Ref	17k	2k	11.8%
[10] TED ASR	17k	2k	11.8%
[10] Dict	25k	3k	12%
[3] Ref	13k	2k	15.4%
[3] ASR	13k	2k	15.4%
Average	21.25k	2.5k	12.48%

by the Linguistic Data Consortium. Before any experimentation, the following normalization rules were applied during a preprocessing cleaning process over the *GW_afp* dataset:

- XML tags and hyphens elimination
- Lowercase conversion
- Doubled punctuation marks elimination
- Apostrophes isolation
- Substitution of (?, !, ;, :, .) into "< SEG >"

The amount of tokens after the cleaning process for the *GW_afp* dataset is 477M, where 9% correspond to any punctuation mark (Table 2). This proportion is very similar to the *Nicola et al. (2013) WSJ*'s dataset presented in Table 1, which consists of newspaper text. 80% of the tokens were used during training and validation while 20% was used exclusively for testing.

Table 2: *GW_afp dataset statistics.*

Dataset	tokens	punctuation	percentage
<i>GW_afp</i>	477M	43M	9%

3.2. Metrics

To evaluate our models we considered necessary two types of metrics. At a first glance we opted for Accuracy (2), a general metric that could measure the performance of the models regardless the class. Nevertheless, given the disparity of samples between the two classes, Accuracy is very likely to be biased; so Precision (3), Recall (4) and F1 (5) metrics were calculated for each one of the two classes.

$$Accuracy = \frac{\#correctly_predicted_samples}{\#samples} \quad (2)$$

$$Precision_{ci} = \frac{\#correctly_predicted_samples_{ci}}{\#total_predicted_samples_{ci}} \quad (3)$$

$$Recall_{ci} = \frac{\#correctly_predicted_samples_{ci}}{\#total_samples_{ci}} \quad (4)$$

$$F1_{ci} = 2 * \frac{Precision_{ci} * Recall_{ci}}{Precision_{ci} + Recall_{ci}} \quad (5)$$

3.3. Results

Three different baselines are shown in Table 3. In their experiments, Authors of [3] compute only Precision, Recall and F1 for the "sentence boundary" class. CNN-2 and CNN-2A refer to the same convolutional neural network model but in CNN-2A is only taken into account half the value of softmax output for the "no sentence boundary" class. This variation equilibrates Precision and Recall of CNN-2 reaching a F1 score value of 0.788.

CNN-A_u refers to the untrained CNN-A model. We wanted to have this as a baseline to visualize how the unbalanced nature of the samples impacts all measures and may mislead general metrics like Accuracy.

Accuracy over all the proposed models is higher than in CNN-A_u, reaching the higher score for CNN-B. Concerning Precision, CNN-B and CNN-A overperform for different classes. CNN-2A reflects a higher Recall than the rest of the baselines and models. Finally, F1 score for both classes is higher in CNN-B.

Given the similar results of the models we wanted to see the behavior of the models during training process. Cross entropy during training process is plotted in Figures 2 to 4. The three curves show a similar behavior and converge in a value below 0.09. CNN-B slightly overperforms the rest of the models (Table 4).

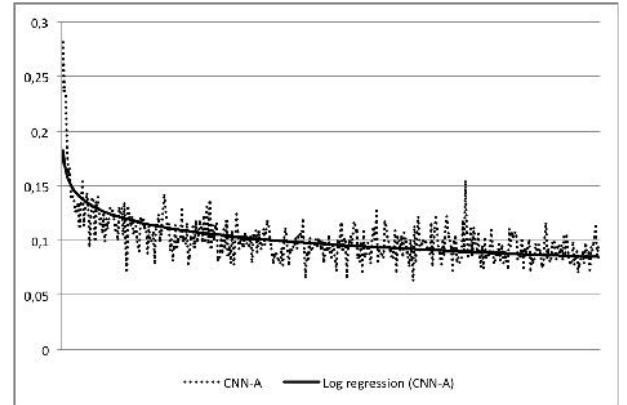


Figure 2: Cross entropy (CNN-A)

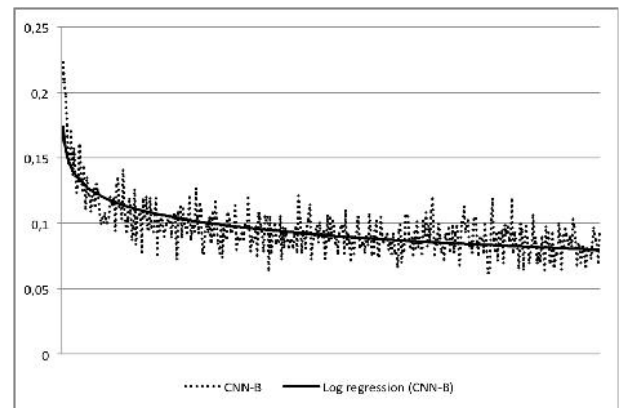


Figure 3: Cross entropy (CNN-B)

Table 3: Results for CNN models.

Model	Accuracy	Precision		Recall		F1	
		NO_SEG	SEG	NO_SEG	SEG	NO_SEG	SEG
CNN-2 [3]	-	-	0.836	-	0.723	-	0.775
CNN-2A [3]	-	-	0.776	-	0.799	-	0.788
CNN-A_u	0.909	0.909	0	1	0	0.952	0
CNN-A	0.963	0.972	0.853	0.988	0.718	0.980	0.778
CNN-B	0.965	0.975	0.845	0.986	0.754	0.981	0.795
CNN-C	0.963	0.974	0.832	0.985	0.75	0.980	0.787

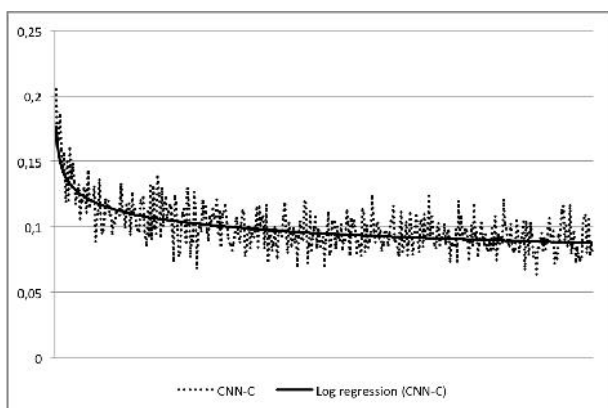


Figure 4: Cross entropy (CNN-C)

Table 4: Cross entropy during training

Model	Cross entropy
CNN-A	0.082
CNN-B	0.080
CNN-C	0.089

4. Conclusions

In this paper we combined CNN networks with SLI vectors to tackle the problem of sentences boundary detection as a binary classification task for French. We used a big written dataset instead of standard size transcriptions to reuse the trained models in further transcriptions. SLI vectors, that represent words as the sum of their characters vectors taking advantage of their morphology, showed to be very effective working with our three CNN models. In a future, we will include other languages like Arabic and English. Also we will reuse the trained models in a variety of ASR transcriptions of newscasts and reports domain.

5. Acknowledgements

We would like to acknowledge the support of Chist-Era for funding this work through the *Access Multilingual Information opinionS (AMIS)*, (France - Europe) project.

6. References

- [1] D. Fohr, O. Mella, and I. Illina, "New paradigm in speech recognition: Deep neural networks," in *IEEE International Conference on Information Systems and Economic Intelligence*, 2017.
- [2] N.-T. Le, B. Lecouteux, and L. Besacier, "Disentangling ASR and MT Errors in Speech Translation," in *MT Summit 2017*, Nagoya, Japan, Sep. 2017. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01580877>
- [3] X. Che, C. Wang, H. Yang, and C. Meinel, "Punctuation prediction for unsegmented transcript based on word vector," in *The 10th International Conference on Language Resources and Evaluation (LREC)*, 2016.
- [4] W. Lu and H. T. Ng, "Better punctuation prediction with dynamic conditional random fields," in *Proceedings of the 2010 conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2010, pp. 177–186.
- [5] O. Tilk and T. Alumäe, "Bidirectional recurrent neural network with attention mechanism for punctuation restoration," *Interspeech 2016*, pp. 3047–3051, 2016.
- [6] S. Peitz, M. Freitag, and H. Ney, "Better punctuation prediction with hierarchical phrase-based translation," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, South Lake Tahoe, CA, USA, 2014.
- [7] I. Zribi, I. Kammoun, M. Ellouze, L. Belguith, and P. Blache, "Sentence boundary detection for transcribed tunisian arabic," *Bochumer Linguistische Arbeitsberichte*, p. 323, 2016.
- [8] J. Kolář and L. Lamel, "Development and evaluation of automatic punctuation for french and english speech-to-text," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.
- [9] F. Batista, H. Moniz, I. Trancoso, and N. Mamede, "Bilingual experiments on automatic recovery of capitalization and punctuation of automatic speech transcripts," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 474–485, 2012.
- [10] U. Nicola, B. Maximilian, and V. Paul, "Improved models for automatic punctuation prediction for spoken and written text," in *Proceedings of INTERSPEECH*, 2013.
- [11] N. Ueffing, M. Bisani, and P. Vozila, "Improved models for automatic punctuation prediction for spoken and written text," in *INTERSPEECH*, 2013, pp. 3097–3101.
- [12] M. Igras and B. Ziólko, "Detection of sentence boundaries in polish based on acoustic cues," *Archives of Acoustics*, vol. 41, no. 2, pp. 233–243, 2016.
- [13] X. Che, S. Luo, H. Yang, and C. Meinel, "Sentence boundary detection based on parallel lexical and acoustic models," in *Interspeech*, 2016, pp. 2528–2532.
- [14] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv preprint arXiv:1607.04606*, 2016.
- [15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

- [16] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: <http://www.aclweb.org/anthology/D14-1162>
- [17] A. Mnih and K. Kavukcuoglu, "Learning word embeddings efficiently with noise-contrastive estimation," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 2265–2273. [Online]. Available: <http://papers.nips.cc/paper/5165-learning-word-embeddings-efficiently-with-noise-contrastive-estimation.pdf>
- [18] O. Levy and Y. Goldberg, "Linguistic regularities in sparse and explicit word representations," in *Proceedings of the eighteenth conference on computational natural language learning*, 2014, pp. 171–180.
- [19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [20] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional neural networks for large-scale remote-sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 645–657, 2017.
- [21] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.
- [22] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 160–167.
- [23] D. Palaz, G. Synnaeve, and R. Collobert, "Jointly learning to locate and classify words using convolutional networks," in *INTER-SPEECH*, 2016, pp. 2741–2745.

An empirical study of the Algerian dialect of Social network

Karima Abidi, Kamel Smaili

SMarT Group, LORIA, F-54600, France
{karima.abidi, kamel.smaili}@loria.fr

Abstract

In this paper, we present analysis on the use of Algerian dialect in Youtube. To do so, we harvested a corpus of 17M of words. This latter was exploited to extract a comparable Algerian corpus, named CALYOU by aligning pairs of sentences written in Latin and Arabic. This one was built by using a multilingual word embeddings approach. Several experiments have been conducted to fix the parameters of the Continuous Bag of Words approach that will be discussed in this article. The method we proposed achieved a performance of 41% in terms of Recall. In the following, we present several figures on the collected data that led to several unexpected results. In fact, 51% of the vocabulary words are written in Latin script and 82% of the total comments are subject to the phenomenon of code-switching.

Index Terms: Algerian dialect, Code-switching, comparable corpora, Word embedding.

1. Introduction

Modern Standard Arabic (MSA) is the official language shared by the entire Arab world that is a simplified form of the old Arabic (Classical Arabic), this last one is found only in the religious texts. Beside of MSA, there is another form of Arabic widely used, but it is generally dedicated to the daily communications, named Arabic dialect or *Darija* in Maghrebi countries. Nowadays, with the advent of social networks, the Arabic dialect is widely used, because the Maghrebi people prefer posting their messages in their local colloquial instead of MSA. Arabic Dialect is henceforth written, it arises several new NLP issues. In fact, this form of Arabic undergone a great morpho-syntactic modifications by relaxing several grammatical constraints of MSA. Furthermore, each Arabic region has its own dialect, which leads to different linguistic variations. There is a large number of Arabic dialects: Arabian Peninsula, Levantine, Mesopotamian, Egyptian, and Maghrebi.

In this work, we are interested by the Algerian dialect that has several specificities, among them, the use of a lot of French words and since recently, English words due to the new immigration of Algerian people to English-speaking countries.

In a previous work [1], we addressed the difficult issue of creating not only a dialectal corpus, but a comparable corpus. In fact, parallel or comparable corpora are an essential material for several NLP applications, such as machine translation, building bilingual dictionaries, and so on. In [1] we proposed to build automatically a Comparable spoken ALgerian corpus extracted from YOUTube (CALYOU). The method proposed is based

on the concept of learning multilingual word embeddings (Word2Vec).

In this paper, the objective is to make a deep analysis concerning the collected corpus. This will help to understand how the Algerian people write in social networks and to analyze the important phenomenon of code-switching. In fact, people switch from the Algerian dialect to MSA, French and sometimes to English.

The rest of the paper is organized as follows: Section 2 concerns the related work, while Section 3 we describes the collected corpus and we conduct a deep analysis on these data. Section 4 discusses the automatic multilingual word embedding method used to develop CALYOU, we present also in this section some experimentations to set up Word2Vec parameters. In Section 5 we present some figures concerning CALYOU and then we conclude.

2. Related Works

The NLP community started few years ago to pay attention to Arabic dialects processing. However, at the beginning the majority of these works has been limited only to some dialects and mainly to develop NLP tools rather than resources. In this section, we will present a global overview of research related to Arabic dialect processing with a focus on Algerian dialect corpora. Building resources such as corpora or lexicon is a time consuming process, especially for complex or vernacular languages. For Arabic dialect several researches handle the issue of developing resources. In [2], authors created parallel corpora by using crowdsourcing approach to translate sentences from Egyptian and Levantine into English. A multi-dialect Arabic (MSA and dialects from Egypt, Arabic Peninsula and Levantine) speech parallel corpus has been proposed in [3]. This kind of resources is very rare, since it is expensive and time consuming. In fact, 32 speech hours have been recorded that corresponds to 1291 recordings for MSA and 1069 for dialects. Mubarak and Darwish in [4] used Twitter to collect an Arabic multi-dialect corpus for Saudi Arabian, Egyptian, Algerian, Iraqi, Lebanese and Syrian dialect. In [5], the authors presented a multi-dialect Arabic parallel corpus (2000 sentences): Egyptian, Tunisian, Jordanian, Palestinian, Syrian, MSA and English. At the best of our knowledge, there is no work consisting of aligning parallel corpora for Algerian dialect except PADIC (Parallel Arabic DIAlect Corpus) [6]¹, which covers beside MSA, six Arabic dialects : Annaba (Algerian), Algiers (Algerian), Tunisian, Moroccan, Syrian and Palestinian. This interesting resource has been used to launch the first ma-

¹PADIC can be downloaded from <http://smart.loria.fr/pmwiki/pmwiki.php/PmWiki/Corpora>

chine translation for Algerian dialect [7],[8] and could be used also in several other applications.

In a previous work [1], we developed automatically a comparable spoken Algerian corpus. The interest of this resource is that, it is trained automatically on data extracted from social networks, in the opposite to what has been done in PADIC or in other resources. In this work, CALYOU has been updated, it includes: Algerian dialect, MSA, French and English.

3. Collected corpus

To build an Algerian dialect corpus, we harvested comments posted by Algerians corresponding to Youtube videos, by using the Google’s API ². To ensure that the comments collected mainly concern topics posted by Algerians, we chose few keywords to form queries to retrieve videos concerning national news, Algerian celebrity, local football, etc. Table 3 shows some figures before and after preprocessing the collected data, where $|C|$ is the number of comments, $|W|$ is the number of words and $|V|$ is the vocabulary size. We can mention that after the cleaning

	Raw corpus	Cleaned Corpus
$ C $	1.3M	1.1M
$ W $	20M	17.7M
$ V $	1.3M	0.99M

Table 1: The collected YouTube Algerian Dialect Corpus.

process, the corpus has been reduced by around 15% and the vocabulary by around 24%.

3.1. Investigating Algerian Youtube Corpus

In the following, we will present a study concerning the collected corpus. To our knowledge, there is not a recent study about the Algerian dialect used in social networks. The objective of this study is to have an idea about the characteristics of the Algerian dialect. In the first and second lines of Table 2, we present

	Youtube	Percentage
$ LS $	557K	47%
$ AS $	623K	53%
$ FR $	15457	1%
$ AR $	88982	8%

Table 2: Figures on Youtube Algerian comments

the number of comments written respectively in Latin Script (LS) and Arabic Script (AS). The table shows that 47% of the comments are written in Latin Script. This high rate could be explained by the fact that people are influenced by the French culture and also, in Algeria the mobile phone keyboards are by default in French, which makes writing in Latin script easier, even if it is possible to configure the mobile phone in order to have an Arabic keyboard. We remind that, comments in LS correspond to either Arabizi, French or

²Available at: <https://developers.google.com/YouTube>

English. Similarly, comments in AS correspond to MSA or dialect. The proportion of French comments in the total corpus and in the subset of the corpus where the comments are written in Latin script are 1% and 2.7% respectively. A comment in LS is considered as French, if each of its words is French found in a dictionary of 6 millions of words [9]. Similarly, 14.2% (which corresponds to 8% of the total number of comments) of the Arabic posts correspond to MSA comments. Such as for French, we used a MSA dictionary of 9 millions of entries [10]. Table 3 gives the length of comments,

	AS	LS
<i>Min</i>	2	2
<i>Max</i>	5211	4046
<i>Mean</i>	15	13

Table 3: Some statistics on the extracted comments

for both Arabic and Latin script. We can remark that comments written in AS or LS are in average almost similar in terms of length.

The pie chart of Figure 1 gives details about the distribution of the vocabulary’s words. The Modern Standard Arabic represents 21% of the vocabulary, but this result is biased, since a dialectal word may exist in MSA, but it may have a different meaning. For instance, the MSA word شابة means *young*, but in Algerian dialect it means *beautiful*. That is why this rate is not accurate, it is, in fact, difficult to estimate it since we do not have an Algerian dialect lexicon.

Words which are not in a MSA dictionary are considered such as dialectal words, they represent 74% of the whole dictionary. Among them, 46% are written in Latin script. The harvested corpus is composed by 0.99M of distinct words, 51% of them are written in Latin script. This illustrates the important use of Latin script in the Algerian dialect used in Youtube. One can remark through this experiment that people prefer writing in Latin script and in addition they do not pay special attention to grammar. Furthermore, for uneducated people, they write a word as they want, or at the best, such as it is pronounced. This probably explains the high number of words of the vocabulary written in Latin script. This diversity of words is illustrated by the

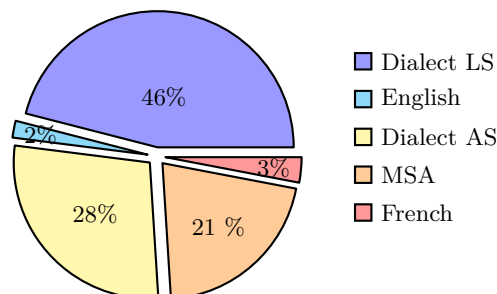


Figure 1: Vocabulary

example (يرحمك) that has been written in 66 different ways in our corpus (see Table 4).

يرحمك → yr7mk yr7mak yrhamak yarhamek yarhemak yarhamk yr7mek yere7mek yarhamak yarhemek yrhmk yar7mak yarhmak yarhmek yar7mik arahmak yar7mek arahmk yerhemek arahmek yerehmek yerhamek yer7mak yare7mek yerhamak yer7mek yerehemek yarhmeke rahimaka yrahmek yrahmak irahmak irhmak irahmek yra7mk irhmk yrehmak yera7mak yerehmek yera7mek yrehmek yara7mak yarehmek yara7mek yarahmeke yarehmak yarhmk yerhmk yarhmeek yra7mak ir7mak yra7mek yarhamoka yrehmk yar7mk yrahmk ira7mak irehmek yerhmek yarahmk yrhmek yarahmek yerhmak yarahmak yrhmak yarahemek

Table 4: Different ways to write in Latin Script the word **يرحمك**

The pie chart of Figure 2 illustrates an important result concerning the influence of code-switching in the Algerian dialect. In fact, 82% of the comments are a mixture of several languages (MSA, dialect, French and sometimes English). The posts, which are entirely in dialect constitutes only 9% of the total corpus. This proves that the processing of the Algerian dialect necessitates particular NLP tools. In comparison to Hindi, for which the issue of code-switching is also important, the rate of Hindi language, in a corpus of 30 minutes [11] is high (67.7%), while in our corpus of Algerian dialect is only 39%. The phenomenon of code-switching is crucial, since if we add up all what it has been written in Arabic (Dialect (Arabizi or not) and MSA) in this corpus, only 15.5% of the comments are entirely written in Arabic. All the others are mixture of several codes.

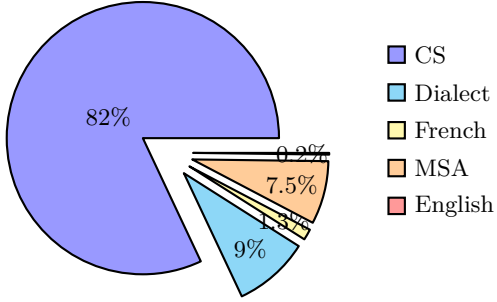


Figure 2: Code-switching distribution

4. Multilingual word embeddings to build CALYOU

In a previous work [1], we addressed the difficult issue of matching comments from YouTube for a vernacular language (Algerian dialect) for which no writing rules do exist. This leads to more difficulties in the processing of the corresponding texts. We recall in this section the main idea of the method proposed in [1]. The comparability of comments cannot be addressed, only by looking for a word into two different comments. In fact, a word, as explained in this paper, may have several ways

of writing. That is why, in this method we decided to find for each word, the corresponding ones. That means all the entries, which are correlated to this word and those which are similar but are written differently (see the example of Table 4) constitutes a set, in which the algorithm of comparability looks for the matching. The proposed approach is based on the concept of learning multilingual word embeddings (Word2Vec). The objective is to build a lexicon that contains, for each word its correlated words. To learn a Correlated Words Lexicon (CWL), for each word (w_s) of the corpus, where s is the Arabic or the Latin script, we learned its correlated words ($w_{\bar{s}}$), where \bar{s} is a script different from s . We opted for a continuous bag of words (CBOW) method [12]. For each w_s , we keep its n best correlated words $w_{\bar{s}}$. Then CWL has been exploited in the matching process of documents to produce comparable comments. This process has been iterated to improve, at each step, the quality of the supposed comparable documents (Figure 3). This method achieved good results and allowed us to build a comparable Algerian dialect corpus named CALYOU.

In Figure 4, we plot the result of the comparability in

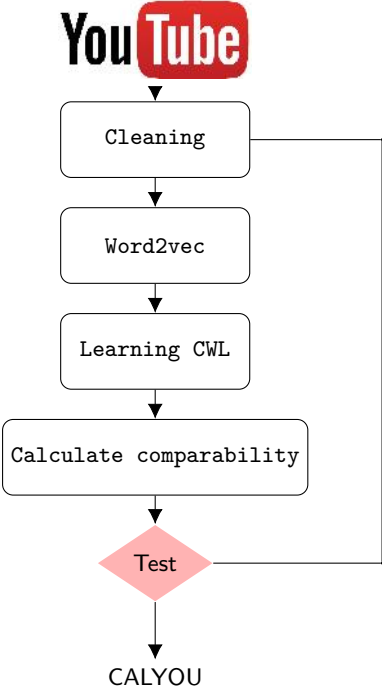


Figure 3: Iterative word embeddings training algorithm

terms of Recall in order to retrieve the best value of the hidden layer (N) of the CBOW algorithm. The curves show that the best value of N is 200 obtained at iteration 2 of the Word2Vec process. Another important parameter is the window-size (ws), which has been determined by making several experiments and by fixing N to 200. The best parameter, in accordance to Figure 5 is equal to 100. These parameters have been tested on a tuning corpus of 310 comparable comments.

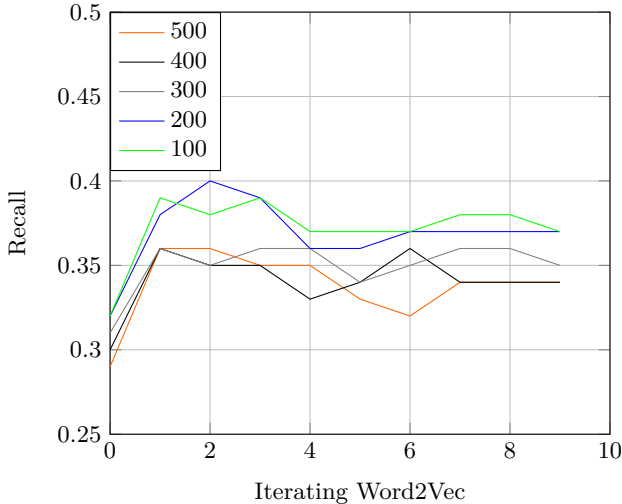


Figure 4: Evolution of the Recall in accordance to the number (N) of neurons in the hidden layer and in terms of Word2Vec iterating process

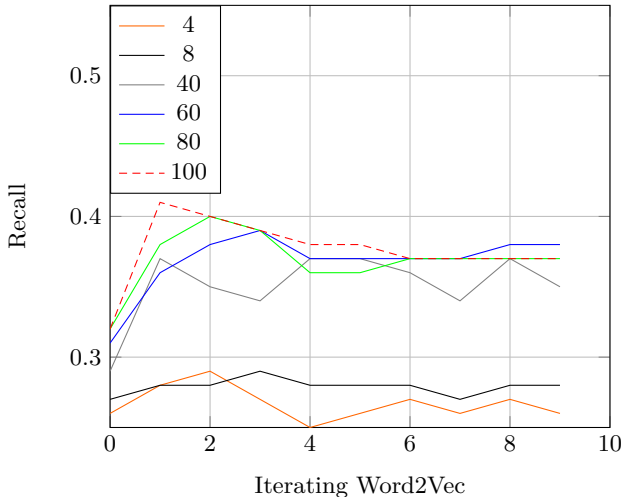


Figure 5: Evolution of the Recall in accordance to the window-size parameter and in terms of Word2Vec iterating process

5. Investigating CALYOU

Such as in Section 3.1, we present in the following some figures concerning the learned comparable corpora CALYOU. The result of CALYOU consists in a list of pairs of comments, for which the source is written in Latin script (Arabizi, French or English) and the target is written in Arabic (MSA or dialect). An example, extracted from CALYOU is given in Table 5.

Table 6 shows that the repartition of Arabizi in the source side of CALYOU is very high (97.3%), while the percentage of comments in French constitutes 2.59% of comments. Concerning the target side, 81% of comments are in Arabic dialect and 19% are in MSA. In the pie chart of Figure 6, such as in the Youtube corpus (Figure 1), the highest frequency distribution concerns the Alge-

Source j'ai trop aimé la tenu c mon style
Translation I like too much your outfit, this is my style
Target عجبوني بزاف نحب هاد ستيل
Translation I like them too much, I like this style
Source Deradji reste avec le sport
Translation Deradji continue taking care of sport
Target: يا سي دراجي انت مختص في الرياضة ابقا في الرياضة بنقو نحبوك
Translation Mr derradji you are expert in Sport, please continue in sport and we will continue appreciating you
Source radja meziane vraiment cette chanson djat thebel be la voix dialek w rabi yerhem kamel messaoudi
Translation Raja Meziane this song is wonderful with your voice, may God bless the soul of Kamel Messaoudi
Target كلمات روعة وجات مع صوتك ربي يرحمو كمال مسعودي
Translation Beautiful Lyrics especially with your voice, may God bless the soul of Kamel Messaoudi

Table 5: Example of comparable comments extracted from CALYOU

	Arabizi	MSA	AD	FR	EN
Source (%)	97.3	-	-	2.59	0.04
Target (%)	-	19	81	-	-

Table 6: Figures on comparable comments of CALYOU

rian dialect written in LS. The second highest frequency distribution of words in CALYOU concerns entries written in MSA (21%).

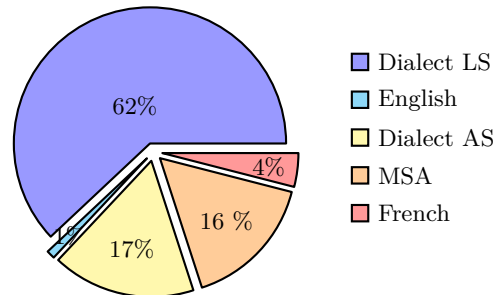


Figure 6: CALYOU Vocabulary

6. Conclusion

In this paper, we presented an analysis of the corpus collected from Youtube (more than 17M of words). The study of this corpus shows that Algerian people, in this social network, use the Latin script intensively. In fact, 47% of the comments are written with this script that was a real surprise for us. To reinforce this observation, we noticed that the percentage of distinct dialect words written in Arabizi is also high (46%) excluding those in French and English. Another crucial issue is the strong presence of the code-switching in the corpus. In fact 82% of the comments are a mixture of several varieties of languages, while only 9% of the comments are entirely written in dialect.

We proposed a multilingual word embedding approach to extract from the latter corpus a comparable one (CALYOU). In other words, each comment in Latin script is aligned with the best corresponding one in Arabic script. We trained, on a tuning corpus the parameters of the

CBOW method, then with the best parameters, we got a performance of 41% in terms of Recall by using iteratively the Word2Vec approach.

Finally, CALYOU is composed by 325K pairs of comparable comments, 62% of its vocabulary is in Arabizi and only 17% is written in Arabic script.

7. References

- [1] K. Abidi, M. A. Menacer, and K. S. and, “Calyou: A comparable spoken algerian corpus harvested from youtube,” in *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, Sweden August 20-24 2017, 2016*, 2017.
- [2] R. Zbib, E. Malchiodi, J. Devlin, D. Stallard, S. Matsoukas, R. M. Schwartz, J. Makhoul, O. Zaidan, and C. Callison-Burch, “Machine translation of arabic dialects,” in *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 3-8, 2012, Montréal, Canada, 2012*, pp. 49–59.
- [3] K. Almeman, M. Lee, and A. A. Almiman, “Multi dialect arabic speech parallel corpora,” in *2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSIPA)*, 2013.
- [4] H. Mubarak and K. Darwish, “Using twitter to collect a multi-dialectal corpus of arabic,” in *The EMNLP 2014 Workshop on Arabic Natural Language Processing 1?7*, 2014.
- [5] H. Bouamor, N. Habash, and K. Ofazer, “A multidialectal parallel corpus of arabic,” in *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014.*, 2014, pp. 1240–1245.
- [6] K. Meftouh, S. Harrat, S. Jamoussi, M. Abbas, and K. Smaïli, “Machine translation experiments on PADIC: A parallel arabic dialect corpus,” in *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation, PACLIC 29, Shanghai, China, October 30 - November 1, 2015*, 2015.
- [7] S. Harrat, K. Meftouh, M. Abbas, and K. Smaïli, “Building Resources for Algerian Arabic Dialects,” in *15th Annual Conference of the International Communication Association Interspeech*. Singapour, Singapore: ISCA, Sep. 2014.
- [8] S. Harrat, K. Meftouh, M. Abbas, S. Jamoussi, M. Saad, and K. Smaili, “Cross-Dialectal Arabic Processing,” in *International Conference on Intelligent Text Processing and Computational Linguistics*, ser. Lecture Notes in Computer Science, cairo, Egypt, Apr. 2015.
- [9] A. Douib, D. Langlois, and K. Smaili, “Genetic-based decoder for statistical machine translation,” in *Springer LNCS series, Lecture Notes in Computer Science*, Dec. 2016.
- [10] M. Menacer, O. Mella, D. Fohr, D. Jovet, D. Langlois, and K. Smaili, “Development of the arabic loria automatic speech recognition system (alacr) and its evaluation for algerian dialect,” in *Third International Conference On Arabic Computational Linguistics, Dubai, November 2017*, 2017.
- [11] A. Dey and P. Fung, “A hindi-english code-switching corpus,” in *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014.*, 2014, pp. 2410–2413.
- [12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *CoRR*, vol. abs/1301.3781, 2013.

Maghrebi Arabic dialect processing: an overview

S. Harrat¹, K. Meftouh², K. Smaili³

¹Ecole Supérieure d'Informatique (ESI),
Ecole Normale Supérieure de Bouzareah (ENSB), Algiers, Algeria

²Badji Mokhtar University-Annaba, Algeria

³Campus scientifique LORIA, Nancy, France

slmhrrt@gmail.com, Karima.meftouh@univ-annaba.org, smaili@loria.fr

Abstract

Natural Language Processing for Arabic dialects has grown widely these last years. Indeed, several works were proposed dealing with all aspects of Natural Language Processing. However, some AD varieties have received more attention and have a growing collection of resources. Others varieties, such as Maghrebi, still lag behind in that respect. Maghrebi Arabic is the family of Arabic dialects spoken in the Maghreb region (principally Algeria, Tunisia and Morocco). In this work we are interested in these three languages. This paper presents a review of natural language processing for Maghrebi Arabic dialects.

Index Terms: Arabic dialect, Maghrebi Arabic dialects, Tunisian Arabic, Algerian Arabic, Moroccan Arabic

1. Introduction

The Arabic language is characterized by its plurality. It consists of a wide variety of languages, which includes the Modern Standard Arabic (MSA), and a set of various dialects differing according to regions and countries. The varieties of Arabic dialects (AD) are distributed over the 22 countries in the Arab World. Geographically, Arabic dialects are classified in two main blocs, namely Middle East (Mashriq) and North Africa (Maghreb) dialects. Maghrebi dialects are the languages that are spoken in this geographical area (Maghreb). They are characterized by the coexistence of several languages: MSA, dialectal Arabic, Berber and French. The Berber dialects constitute the oldest linguistic substratum of this region and are, therefore, the mother tongue of a part of the population. Since the Islamic conquest of the Maghreb, several Arab tribes have intermingled, especially in pastoral areas, because of the similarity of their way of life. This coexistence reinforced the Arabization of the Berber tribes. The influence of the Arabic language on the Berber world spread fairly rapidly, and this practically all over the Maghreb [1]. The French language was introduced by the colonial occupation. First, as the language of the colonial administration, this language has spread to a large part of the population through education and administration. This language spread in its written and oral uses, it influenced the spoken languages (Berber and AD) by the borrowings that these made to it [2].

The Maghreb is composed, in its central part, of Algeria, Tunisia and Morocco. In this paper, we are interested in the Arabic spoken in these three countries. This interest is justified by the fact that these countries have in common a lot of socio-historical similarities and an identical linguistic situation. We therefore present in this work an overview of these dialects, first on several levels of linguistic representation (section 2) and then

in terms of research work dealing with these languages (section 3). We believe that such a study is very useful for the scientific community working in the field of Natural Language Processing (NLP) in general and more specifically those working on NLP of Maghrebi Arabic dialects.

2. Linguistic overview

Maghrebi Arabic dialects include principally Algerian Arabic, Moroccan Arabic and Tunisian Arabic. In this section, we give an overview of these three languages regarding phonological, lexical, morphological and syntactic level.

2.1. At phonological level

The three Maghrebi Arabic dialects share the most features of standard Arabic. Besides the 28 Arabic consonants phonemes, the three dialects of the Maghreb use non Arabic phonemes /g/, /p/ and /v/ which are mainly used in words borrowed from foreign languages as French. Also, the (ظ) is uttered as /dʰ/ (ض), whereas (ذ) and (ث) are mostly pronounced as /d/ (ذ) and /t/ (ث) for both Algerian¹ and Moroccan dialects and not for Tunisian where the utterance of these two consonants is the same as in MSA. Furthermore, the letter (ق) is particular in the way that it has different pronunciations. For the three dialects it is uttered as /q/ and /g/. It should be noted that the use of /g/ is observed not only in rural places but also in urban cities. In addition, the (ق) is uttered as the glottal stop /ʔ/ as in Tlemcen (west of Algeria) and Fes (Morocco), just like in Egyptian dialect. In some eastern cities of Algeria a particular pronunciation of the (ق) is /k/ (this phenomenon does not exist in Tunisian and Moroccan). Also, The consonant (ج) has different pronunciations /dj/, /j/ or /z/ (for Tunisian dialect and the dialect of Tlemcen and other cities in the east of Algeria). Other notable features of Maghrebi dialects are the collapse of short vowels both in nouns and verbs and the glottal stop (Hamza) omission particularly in the middle and the end of words.

2.2. At Lexical level

Maghrebi dialects' vocabulary is mostly inspired from Arabic but it is phonologically altered, with significant Berber substrates, and many loanwords from French, Italian, Turkish and Spanish. Like for Arabic vocabulary, these dialects' vocabularies include verbs, nouns, pronouns and particles.

¹In some rural dialects they are pronounced as in MSA.

2.3. At Morphological level

The morphology of Arabic dialectal words shares a lot of features with MSA morphology. Furthermore, dialect inflection system is simpler in some aspects than MSA, whereas affixation system seems to be more complicated than MSA. Indeed inflection system is simplified by the elimination of a wide range of rules. In fact, as in all Arabic dialects, Algerian, Moroccan and Tunisian do not accept the singular word declension which corresponds to the nominative, the genitive, and the accusative cases which take the short vowels ُ , ِ and َ respectively in the end of the word. Similarly, the three doubled case endings expressing nominal indefiniteness are also dropped. It should be noted that for the three dialects, the singular nouns declension to the plural (feminine/masculine regular plural and broken plural) follows MSA rules but with the difference that the three cases enumerated above are not distinguished.² In addition, the three dialects do not have the nominal dual which is a distinctive feature of standard Arabic. The verb conjugation of the three dialects uses a set of affixes slightly different with MSA ones besides a variation in vocalization. We mention that the dual and feminine plural of MSA are lost in the dialects. Moreover, the negation in the three dialects seems to be more complex than in MSA, the circumfix negation (ش + ما) surrounds the verbs with all its affixed direct and indirect object pronouns.

2.4. At Syntactic level

The words order of a declarative sentence in the three dialects is relatively flexible but the most commonly used order is the SVO order (Subject-Verb-Object)[3],[4],[5]. The Other orders are also allowed, the speaker generally begins his sentence with the item that he wants to highlight.

3. NLP of the three dialects

In this section, we are interested in the research work developed for these dialects in various NLP issues.

3.1. Corpora and lexicons

A dictionary containing 18K MSA and Moroccan dialects entries was built in [6]. The authors used manual translation from MSA dictionary to Moroccan dialect and vice-versa.

In [7] authors created an annotated corpus of 223K that they collected from Moroccan social media sources. The corpus has been annotated on token-level by three native speakers of Moroccan dialect.

In [8] a focus was made on Tunisian dialect processing. The authors extracted textual user-generated contents from social networks that they filtered and classified automatically. From the built corpora they drew a picture of the main features related to Tunisian dialect.

The authors in [9] presented a bilingual lexicon of deverbal nouns between MSA and Tunisian dialect that has been created automatically. They extended an existing Tunisian verbal lexicon by using a table of deverbal patterns in order to generate pairs of Tunisian and MSA deverbal nouns.

²Example: Depending on its function in the sentence, the masculine regular plural of MSA word مُسْلِم (Muslim) could be مسلمون (nominative case) or مسلمين (accusative or genitive). In contrast, for the dialect word مَسْلَم (Muslim) always takes مسلمين for the regular plural whatever its grammatical category.

The work presented in [10] is related to the construction of a railway domain ontology from a Tunisian speech corpus created for this purpose within this study. The authors used a statistical method for term and concept extraction whereas for semantic relation extraction they choose a linguistic approach.

In [11], authors generated automatically phonetic dictionaries for Tunisian dialect by using a rule approach. The work is part of an automatic speech recognition framework of the Tunisian Arabic in the particular field of railway transport.

In [12] is presented STAC (Spoken Tunisian Arabic Corpus), 5 transcribed hours of spontaneous Tunisian Arabic speech enriched with morpho-syntactic and disfluencies annotations.

For Algerian dialect, in [13], the authors crawled an Algerian newspaper to extract comments that they used to build a romanized code-switched Algerian Arabic-French corpus. In this study, the authors highlighted the particular Algerian linguistic situation by discussing its main features. It should be noted that the corpus is annotated by language identification at word-level.

KALAM'DZ, An Arabic Spoken corpus dedicated to Algerian dialectal varieties was built in [14] by exploiting Web resources such as Youtube and other Social Media, Online Radio and TV. The dataset covers a large number of Algerian dialects with 4881 native speakers and more than 104 hours.

An other Speech corpus dedicated to Algerian dialect, AM-CASC (Algerian Modern Colloquial Arabic Speech Corpus) was presented in [15]. Authors used this corpus for the purpose of evaluating their automatic regional accent recognition approaches based on GMM-UBM and i-vectors frameworks.

In the same vein, authors of [16] presented their methodology to build an Arabic Speech Corpus for Algerian dialects. The authors proceeded by recording speeches uttered by 109 native speakers from 17 different regions in Algeria.

In [17], CALYOU, a Comparable Corpus of the spoken Algerian was built from Youtube comments. It consists of 853K comments including a total of 12.7M words. This work deals with the issue of comparability of comments extracted from Youtube. It presents a Word2Vec based method of alignment which achieves the best comparability results among the other methods that the authors experimented.

3.2. Identification

Several efforts dealing with Maghrebi Arabic dialects are those dedicated to the identification and recognition. In fact, Arabic dialects differ from one country to another and even in the same Arab country there is a lot of dialect varieties. In this context, authors of [18] addressed the problem of spoken Algerian dialect identification by using prosodic speech information (intonation and rhythm). They performed an experiment of their approach on six dialects from different Algerian departments. An other study [19] showed that Algiers and Oran dialects can be identified by prosodic cues.

In [20], for the classification of Tunisian and Moroccan dialects, two methods were used namely the feed forward back propagation neural network (FFBPNN) and the support vector machine (SVM). The former (FFBPNN) performs better than the later in terms of recognition rates.

In the context of dialect identification within social media (Facebook comments), authors of [21] used an Algiers dialect lexicon and perform different ways of identification: total (word matching), partial (prefix and suffix matching) and by applying improved Levenshtein distance.

The work cited in [22] presents DATOOL a graphical tool for annotating tweets. A native speaker of Moroccan dialect annotated an average of 250 (mixed-language and mixed-script) tweets per hour. The obtained corpus has been used for the purpose of dialect identification.

3.3. Orthography

A particular attention is devoted to dialect orthography because of their spoken nature and thus a total absence of standard writing rules. Some efforts were made to resolve this issue. The authors of [23] presented orthography guidelines for transcribing Tunisian speech corpora based on the standard Arabic transcription conventions. Later, the CODA map (Conventional Orthography for Dialectal Arabic) described in [24] was adapted to Tunisian dialect [25], Algerian dialect [26] and finally in general for Maghrebi dialects [25].

3.4. Morphological analysis

In [27], a morphological analyzer for the Tunisian dialect based on a MSA analyzer was proposed. Furthermore, as an expansion of a MSA lexicon, a lexicon for the Tunisian dialect was built. This last lexicon has been used in [28] to convert a standard Arabic corpus for creating a large Tunisian dialect corpus, in order to train a POS tagger. A similar approach was adopted in [29] where the authors exploited also the closeness between standard Arabic and Tunisian dialect. They developed a POS tagger by converting a Tunisian sentence to MSA lattice, after a disambiguation step, a MSA target sentence is then produced and tagged simply with a MSA tagger.

For Moroccan dialect, in [30] a morphological analyzer has been developed in addition of an annotated corpus that has been created within this work. It should be noted that specific CODA guidelines for Moroccan dialect has been also created (inspired from [24] cited above).

For Algerian dialect, a morphological analyzer was developed in [31]. Authors adapted the well-known morphological analyzer BAMA dedicated for MSA.

3.5. Sentiment analysis

Sentiment analysis is a promising and challenging direction research in the area of dialect NLP. Indeed, Arab people use their dialects on social media and discussion forums to express their opinions. Sentiment analysis for Maghrebi dialects is still in an earlier stage. Most of the work are recent compared to contributions related to MSA or a relatively more-resourced dialect such as Egyptian dialect.

In [32], the authors proposed a lexicon-based approach for sentiment analysis of Algerian dialect. They used a manually annotated dataset and three Algerian Arabic lexicons.

Authors of [33] presented an approach for emotion analysis of Tunisian Facebook pages. They introduced a new method to create emotion dictionaries by using emotion symbols as sentiment polarity indicators. Recently, in [34] the focus was also made on Tunisian dialect sentiment analysis. Their approach is based on machine learning techniques for determining comments polarity. Within this research, a corpus of 17K Facebook comments has been created and annotated.

3.6. Machine translation

Machine translation is an other issue related to Arabic dialects and Maghrebi ones particularly. In fact, Machine translation requires specific resources like parallel corpora in the context

of data-based approach and strong linguistic studies in the case of rule-based approach, while this dialects suffer from a lack of resources especially parallel corpora. Few efforts have been deployed to deal with machine translation of Maghrebi dialects, most issues are not yet solved. There is still much work to be done in this area.

In [35] is proposed a machine translation system between MSA and Tunisian dialect verbal forms (in both directions). It is based on deep morphological representations of roots and patterns (a specific feature of Arabic). Another work dedicated to Tunisian dialect is described in [36]. The authors attempted to translate Tunisian dialect text of social media into MSA by using a bilingual lexicon and a set of grammatical mapping rules and a disambiguation step.

In [37], a machine translation system from Moroccan dialect to MSA is presented. The work used a rule-based approach in addition to a language model. The system used transfer rules based on a morphological analysis (with Alkhalil morphological analyzer [38] which the authors adapted to Moroccan dialect).

In [39] a hybrid machine translation system combining statistical and rule-based approaches is presented. It translated from Arabic dialects to English. Dialects concerned by this study were those of the middle-east in addition to Tunisian, Moroccan and Libyan. MSA was as a pivot language. This system showed that the hybridization of statistical and rule-based approaches performs better than using each approach separately.

Authors of [40] presented PADIC a multi-dialect Arabic corpus that includes MSA, Maghrebi dialects (Algerian and Tunisian and in the last version Moroccan) and Levantine dialects (Palestinian and Syrian). They conducted several experiments on different Statistical Machine Translation (SMT) systems between all pairs of languages (MSA and dialects). They studied the impact of the language model on machine translation by varying the smoothing techniques and by language model interpolation.

3.7. Other resources

In [41] authors dealt with the detection of sentence boundary in transcribed spoken Tunisian Arabic. They proposed a rule-based method and a statistical method, in addition to a third method which combines these two last. Their detection system has been used to improve the accuracy of a POS tagger of transcribed Tunisian dialect.

In [42] an automatic diacritics restoration system was built for Algiers dialect. The system was based on a statistical approach and allowed to vocalize the Algerian part of PADIC [40]. This vocalized corpus has been used in [43] for the purpose of grapheme to phoneme conversion. This last combined a rule-based and a statistical approaches.

In [44], the authors proposed a method to disambiguate the output of a morphological analyzer of the Tunisian dialect (cited in [27]) by using machine-learning techniques.

4. Conclusion

We focused in this paper on Maghrebi Arabic dialects particularly Algerian, Moroccan and Tunisian Arabic. After a linguistic overview, we provided a survey of the research work dealing with these languages. Several comments can be made based on this work. In view of the various published works, we can see that the research efforts dealing with Maghrebi Arabic dialects are at an early stage. Most of the research work dealing with

these dialects has been devoted to the construction of corpora and lexicon. This is mainly due to the fact that these languages are under-resourced. The identification task has also been researched. While the morphology of the Maghrebi Arabic dialects has been addressed in few papers, the syntactic analysis remains totally ignored. It is also worth noting the small number of works devoted to machine translation of these dialects. In addition, these few existing contributions are dedicated to the translation between dialects and MSA, no work has considered the French language.

5. References

- [1] F. Khelef and R. Kebieche, "Evolution ethnique et dialectes du maghreb," *Synergies Monde Arabe* no 8, 2011.
- [2] G. Grandguillaume, "L'arabisation du maghreb," *Revue d'Aménagement linguistique, Aménagement linguistique au Maghreb, Office Québécois de la langue française*, no. 107, pp. 15–40, 2004.
- [3] A. Mahfoudhi, "Agreement lost, agreement regained: A minimalist account of word order and agreement variation in Arabic," *California Linguistic Notes*, vol. 27, no. 2, pp. 1–28, 2002.
- [4] M. L. Souag, "Explorations in the syntactic cartography of Algerian Arabic," Ph.D. dissertation, School of Oriental and African Studies (University of London, 2006.
- [5] M. Ennaji, *Multilingualism, cultural identity, and education in Morocco*. Springer Science & Business Media, 2005.
- [6] R. Tachicart, K. Bouzoubaa, and H. Jaafar, "Building a Moroccan dialect electronic dictionary (MDED)," in *5th International Conference on Arabic Language Processing*, 2014, pp. 216–221.
- [7] Y. Samih and W. Maier, "An Arabic–Moroccan darija code-switched corpus," in *Proceedings of 10th Language Resources and Evaluation Conference (LREC 2016)*, 2016.
- [8] J. Younes, H. Achour, and E. Souissi, *Constructing Linguistic Resources for the Tunisian Dialect Using Textual User-Generated Contents on the Social Web*. Cham: Springer International Publishing, 2015, pp. 3–14. [Online]. Available: <https://doi.org/10.1007/978-3-319-24800-4-1>
- [9] A. Hamdi, N. Gala, and A. Nasr, "Automatically building a Tunisian lexicon for deverbal nouns," in *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, 2014, pp. 95–102.
- [10] J. Karoui, M. Graja, M. Boudabous, and L. H. Belguith, "Domain ontology construction from a Tunisian spoken dialogue corpus," in *International Conference on Web and Information Technologies, ICWIT'2013*, 2013.
- [11] A. Masmoudi, Y. Estève, M. E. Khmekhem, F. Bougares, and L. H. Belguith, "Phonetic tool for the Tunisian Arabic," in *Spoken Language Technologies for Under-Resourced Languages*, 2014.
- [12] I. Zribi, M. Ellouze, L. H. Belguith, and P. Blache, "Spoken Tunisian Arabic corpus" STAC": Transcription and annotation," *Research in computing science*, vol. 90, pp. 123–135, 2015.
- [13] R. Cotterell, A. Renduchintala, N. Saphra, and C. Callison-Burch, "An Algerian Arabic-French code-switched corpus," in *Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools Workshop Programme*, 2014, p. 34.
- [14] S. Bougrine, A. Chorana, A. Lakhdari, and H. Cherroun, "Toward a web-based speech corpus for Algerian Arabic dialectal varieties," *WANLP 2017 (co-located with EACL 2017)*, p. 138, 2017.
- [15] M. Djellab, A. Amrouche, A. Bouridane, and N. Mehallegue, "Algerian modern colloquial Arabic speech corpus (AMCASC): regional accents recognition within complex socio-linguistic environments," *Language Resources and Evaluation*, vol. 51, no. 3, pp. 613–641, Sep 2017.
- [16] S. Bougrine, H. Cherroun, D. Ziadi, A. Lakhdari, and A. Chorana, "Toward a rich Arabic speech parallel corpus for Algerian sub-dialects," in *LREC16 Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools (OSACT)*, 2016, pp. 2–10.
- [17] K. Abidi, M. Menacer, and K. Smaïli, "CALYOU: A comparable spoken Algerian corpus harvested from youtube," *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, pp. 3742–3746, 2017.
- [18] S. Bougrine, H. Cherroun, and D. Ziadi, "Prosody-based spoken Algerian Arabic dialect identification," in *International Conference on Natural Language and Speech Processing, ICNLSP'2015*, 2015.
- [19] I. Benali, "The identification of two Algerian Arabic dialects by prosodic focus," in *7th Tutorial and Research Workshop on Experimental Linguistics, ExLing 2016*, 2016, p. 37.
- [20] M. Hassine, L. Boussaid, and H. Messaoud, "Maghrebian dialect recognition based on support vector machines and neural network classifiers," *International Journal of Speech Technology*, vol. 19, no. 4, pp. 687–695, 2016.
- [21] I. Guellil and F. Azouaou, "Arabic dialect identification with an unsupervised learning (based on a lexicon). application case: Algerian dialect," in *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), 2016 IEEE Intl Conference on*. IEEE, 2016, pp. 724–731.
- [22] S. Tratz, D. M. Briesch, J. Laoudi, and C. R. Voss, "Tweet conversation annotation tool with a focus on an Arabic dialect, Moroccan darija," in *LAW@ ACL*, 2013, pp. 135–139.
- [23] I. Zribi, M. Graja, M. E. Khmekhem, M. Jaoua, and L. H. Belguith, "Orthographic transcription for spoken Tunisian Arabic," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2013, pp. 153–163.
- [24] N. Habash, M. T. Diab, and O. Rambow, "Conventional orthography for dialectal Arabic," in *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2012, pp. 711–718.
- [25] I. Zribi, R. Boujelbane, A. Masmoudi, M. Ellouze, L. H. Belguith, and N. Habash, "A conventional orthography for Tunisian Arabic," in *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2014, pp. 2355–2361.
- [26] H. Saadane and N. Habash, "A conventional orthography for Algerian Arabic," in *ANLP Workshop 2015*, 2015, p. 69.
- [27] I. Zribi, M. E. Khmekhem, and L. H. Belguith, "Morphological analysis of Tunisian dialect," in *International Joint Conference on Natural Language Processing*, 2013, pp. 992–996.
- [28] R. Boujelbane, M. Mallek, M. Ellouze, and L. H. Belguith, "Fine-grained pos tagging of spoken Tunisian dialect corpora," in *International Conference on Applications of Natural Language to Data Bases/Information Systems*. Springer, 2014, pp. 59–62.
- [29] A. Hamdi, A. Nasr, N. Habash, and N. Gala, "POS-tagging of Tunisian Dialect Using Standard Arabic Resources and Tools," in *Workshop on Arabic Natural Language Processing*, Beijing, China, 2015, pp. 59 – 68.

- [30] F. Al-Shargi, A. Kaplan, R. Eskander, N. Habash, and O. Rambow, "Morphologically annotated corpora and morphological analyzers for Moroccan and sanaani yemeni Arabic," in *10th Language Resources and Evaluation Conference (LREC 2016)*, 2016.
- [31] S. Harrat, K. Meftouh, M. Abbas, and K. Smaïli, "Building resources for Algerian Arabic dialects," in *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2014, pp. 2123–2127.
- [32] M. Mataoui, O. Zelmati, and M. Boumechache, "A proposed lexicon-based sentiment analysis approach for the vernacular Algerian Arabic," *Research in Computing Science*, vol. 110, pp. 55–70, 2016.
- [33] H. Ameer, S. Jamoussi, and A. B. Hamadou, "Exploiting emoticons to generate emotional dictionaries from facebook pages," in *Intelligent Decision Technologies 2016*. Springer, 2016, pp. 39–49.
- [34] S. Medhaffar, F. Bougares, Y. Estève, and L. Hadrich-Belguith, "Sentiment analysis of Tunisian dialects: Linguistic resources and experiments," pp. 55–61, 2017.
- [35] A. Hamdi, R. Boujelbane, N. Habash, and A. Nasr, "The effects of factorizing root and pattern mapping in bidirectional Tunisian-standard Arabic machine translation," in *MT Summit*, 2013.
- [36] F. Sadat, F. Mallek, M. Boudabous, R. Sellami, and A. Farzindar, "Collaboratively constructed linguistic resources for language variants and their exploitation in nlp application, the case of Tunisian Arabic and the social media," in *Proceedings of the Workshop on Lexical and Grammatical Resources for Language Processing*. Association for Computational Linguistics and Dublin City University, 2014, pp. 102–110.
- [37] R. Tachicart and K. Bouzoubaa, "A hybrid approach to translate Moroccan Arabic dialect," in *Proceedings of the 9th International Conference on Intelligent Systems: Theories and Applications (SITA-14)*, 2014, pp. 1–5.
- [38] A. Boudlal, A. Lakhouaja, A. Mazroui, A. Meziane, M. O. A. o. Bebah, and M. Shoul, "Alkhalil morpho sys1: A morphosyntactic analysis system for Arabic texts," in *Proceedings of the International Arab Conference on Information Technology, ACIT*, 2010.
- [39] H. Sawaf, "Arabic dialect handling in hybrid machine translation," in *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*, Denver, Colorado, 2010.
- [40] K. Meftouh, S. Harrat, S. Jamoussi, M. Abbas, and K. Smaïli, "Machine translation experiments on PADIC: A Parallel Arabic Dialect Corpus," in *Proceedings of the 29th Asia Conference on Language, Information and Computation (PACLIC)*, 2015, pp. 26–34.
- [41] I. Zribi, I. Kammoun, M. Ellouze, L. Belguith, and P. Blache, "Sentence boundary detection for transcribed Tunisian Arabic," *Bochumer Linguistische Arbeitsberichte*, p. 323, 2016.
- [42] S. Harrat, M. Abbas, K. Meftouh, and K. Smaïli, "Diacritics restoration for Arabic dialect texts," in *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2013, pp. 125–132.
- [43] S. Harrat, K. Meftouh, M. Abbas, and K. Smaïli, "Grapheme to phoneme conversion: An Arabic dialect case," in *Proceedings of 4th International Workshop On Spoken Language Technologies For Under-resourced Languages SLTU*, 2014, pp. 257–262.
- [44] I. Zribi, M. Ellouze, L. H. Belguith, and P. Blache, "Morphological disambiguation of Tunisian dialect," *Journal of King Saud University - Computer and Information Sciences*, vol. 29, no. 2, pp. 147 – 155, 2017.

AUTOMATIC DIALECTAL RECOGNITION IN ARABIC BROADCAST MEDIA

Bilal Belainine, Fatma Mallek, and Fatiha Sadat

Université du Québec à Montréal (UQAM)
201 Président Kennedy, Montréal,
H3X 2Y7, QC, Canada
{belainibe.bilal, mallek.fatma}@courrier.uqam.ca, sadat.fatiha@uqam.ca

ABSTRACT

This paper describes the first participation of UQAM's NLP team at the Arabic Multi-Genre Broadcast (MGB) Challenge for ASRU 2017.

The data used in this shared task was collected from Egyptian multi-genres YouTube videos and includes seven genres as follows: comedy, cooking, family/kids, fashion, drama, sports, and science.

In this shared task, we used supervised learning methods that emphasize only on labelled data, to discriminate and distinguish between four major Arabic dialects: Egyptian, Levantine, North African, Gulf and Modern Standard Arabic (MSA).

Our experiments rely on several machine learning algorithms and their combinations involving Voting Ensemble, Multi-Layer perceptron and Logistic classifiers.

Our best results were obtained during this shared task on closed submission using the Voting Ensemble with an overall accuracy of 56.10, followed by the simple logistic and Multi-Layer perceptron with an overall accuracy of 54.56 and 53.55, respectively.

Index Terms— Arabic dialects; broadcast speech; multi-layer perceptron; logistic; ensemble combination

1. INTRODUCTION

Dialects Identification (DID) is very crucial and hard task for many NLP applications, especially when dealing with noisy and unstructured data such as social media and speech.

The task of DID is a special case of the more general problem of Language Identification (LID). LID refers to the process of automatically identifying the language class for given speech segment or text document. DID is arguably a more challenging problem than LID, since

it consists of identifying the different dialects within the same language class [13].

Arabic is a morphologically rich and complex language, which presents significant challenges for Natural Language Processing (NLP) and its applications [14, 15]. It is the official language in 22 countries spoken by more than 350 million people around the world. Moreover, the Arabic language exists in the state of diglossia where the standard form of the language, Modern Standard Arabic (MSA) and the regional dialects (AD) live side-by-side and are closely related [16]. Arabic has more than 22 dialects; some countries share the same dialect, while many dialects may exist alongside MSA within the same Arab country.

Modern Standard Arabic (MSA) is the lingua franca of the so-called Arab world, which includes northern Africa, the Arabian Peninsula, and Mesopotamia. However, Arabic speakers generally use dramatically different languages (or dialects) in daily interactions and in social media. Both NLP and Speech community are interested in the problem of DID, which has potential to improve Speech and Language applications [13].

The Arabic MGB3 challenge targets dialectal Arabic speech and uses data collected from YouTube videos across different genres [12].

This paper deals with DID using supervised learning methods that emphasize only on labelled data, to discriminate and distinguish between four major Arabic dialects: Egyptian, Levantine, North African, Gulf and Modern Standard Arabic (MSA). We conducted several experiments using machine learning algorithms and their combinations and involving Voting Ensemble, Multi-Layer perceptron and Logistic classifiers.

Our best results were obtained during this shared task on closed submission using the Voting Ensemble with an overall accuracy of 56.10, followed by the simple logistic and Multi-Layer perceptron with an overall accuracy of 54.56 and 53.55, respectively.

This paper is organized as follows: Section 2 presents related

work. Sections 3 and 4 describe the methodology and conducted experiments and their results. Conclusions and work are presented in Section 5.

2. STATE OF THE ART

There have been several works on Arabic Natural Language Processing (NLP). However, most traditional techniques have focused on MSA, since it is understood across a wide spectrum of audience in the Arab world and is widely used in the spoken and written media. Few works relate the processing of dialectal Arabic that is different from processing MSA. First, dialects leverage different subsets of MSA vocabulary, introduce different new vocabulary that are more based on the geographical location and culture, exhibit distinct grammatical rules, and adds new morphologies to the words. The gap between MSA and Arabic dialects has affected morphology, word order, and vocabulary [17]. Almeman and Lee [18] have shown in their work that only 10% of words (uni-gram) share between MSA and dialects.

Second, one of the challenges for Arabic NLP applications is the mixture usage of both AD and MSA within the same text or speech.

In relation to multimodal or speech data, an effective and well-studied method in language and dialect recognition is the i-vector approach [7, 16, 17]. The i-vector involves modeling speech using a universal background model (UBM) – typically a large GMM – trained on a large amount of data to represent general feature characteristics, which plays a role of a prior on how all dialects look like. The i-vector approach is a powerful technique that summarizes all the updates happening during the adaptation of the UBM mean components to a given utterance. Malmasi et al. [6] find that character p-grams are “in most scenarios the best single feature for this task”, even in a cross-corpus setting. Their findings are consistent with the results of Ionescu and Popescu [23] in the ADI Shared Task of the DSL 2016 Challenge [7], as they ranked on the second place using solely character p-grams from Automatic Speech Recognition (ASR) transcripts.

This year (2017), ADI shared tasks, such as VARDIAL¹ was the first shared task to provide participants with the opportunity to carry out Arabic dialect identification using a dataset containing both audio and text (transcriptions). The first edition of the ADI shared task, organized in 2016 as a sub-task of the

DSL shared task [7] used a similar dataset to the ADI 2017 dataset, but included only transcriptions.

The 2017 ADI Shared Task data set [7] contained the original audio files and some low-level audio features, called i-vectors, along with the ASR transcripts of Arabic speech collected from the Broadcast News domain. Some experiments have indicated that the audio features produced a much better performance, probably because there are many ASR errors (perhaps more in the dialectal speech segments) that make Arabic dialect identification from ASR transcripts much more difficult.

3. METHODOLOGY

Our proposed approach for Arabic DID focuses on the concatenation of i-vector audio representation vectors with bi-gram character-based vectors.

Character n-gram model is well suited for language identification and dialect identification tasks that have many languages and/or dialects, little training data and short test samples.

One of the main reasons to use a character-based model is that most of the variation between dialects, is based on affixation, which can be extracted easily by the language model, though also there are word-based features which can be detected by lexicons.

The i-vector system was initially developed by Dehak et al. [19], with an improvement made by Burget et al. [20]. The system involves training a matrix T to model the total variability of a set of statistics for each audio track. The statistics primarily involve the first-order Baum-Welch statistics of the low-level acoustic feature frames (i.e., MFCCs) of each audio track [22].

Our methodology relies on the combination of bi-grams vectors with the i-vectors. Latent Dirichlet Allocation (LDA) was used for dimensionality reduction. LDA is based on a hypothetical generative process for a corpus and has many advantages for topic modeling, including its relative simplicity to implement and the useful topics that it unearths [21].

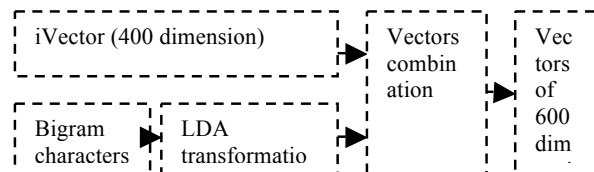


Figure 1. The full process of our methodology on generating and combining the vectors

¹ <http://ttg.uni-saarland.de/vardial2017/sharedtask2017.html>

We fixed the number of topics to generate the dimensions and the matrix to 200 topics for each paragraph. Moreover, we have made a combination of the vectors generated using LDA with vectors generated by the i-vector compression.

The complete combination process is explained in Figure 1.

In relation to the classification algorithms, we used the Voting Ensemble, Multi-Layer perceptron and Logistic classifiers.

Figures 2, 3 and 4 show how these algorithms were used in the Arabic DID.

The Arabic Dialect Identification (ADI) task requires the discrimination of the speech at the utterance level between five different five Arabic dialects.

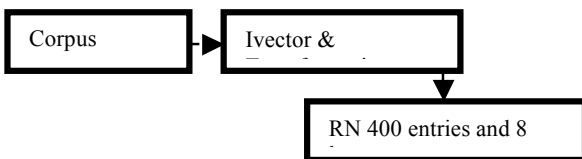


Figure 2. Multi-Layer Perceptron for Arabic DID

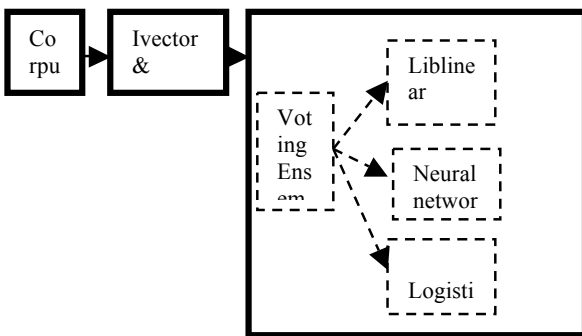


Figure 3. Voting ensemble for Arabic DID



Figure 4. Logistic Classifier for Arabic DID

4. EVALUATIONS

In this challenge, both acoustic and lexical features were used as explained in [12].

Our experiments were based only on labeled data. The results, we obtained using the test file that was provided by the shared task organizers are shown in Table 1, in terms of the accuracy measure. While, using

cross-validation and in relation to VARDIAL 2017 shared task² on Arabic DID, we obtained the results of Table 2, in terms of Precision, Recall and accuracy measures. Detailed experiments in terms of precision, recall and F-measures, using the three classification algorithms and the cross-validation are shown in Table 3, 4 and 5 for the three classification algorithms.

Based on these results, we can notice that combining different classification algorithms (Voting ensemble), gives the best results in term of accuracy. Moreover, using the cross-validation, we notice that our obtained F-measures for the three algorithms and for each Arabic dialect are situated between 72 % and 89%. The average F-measure on the five dialects is situated between 81% and 84%, which is very promising.

Classification algorithms	Accuracy
Voting Ensemble	56.10
Logistic	54.56
Multi-layer perceptron	53.6

Table 1. Results on the test file of ASRU/ MGB challenge

Accuracy	53.6
Precision	54.3
Recall	59.5

Table 2. Results on the test file of VARDIAL 2017 Arabic DID challenge using the Multi-layer perceptron classifier

Dialect	Precision	Recall	F-Measure
GLF	0,841	0,840	0,841
LAV	0,771	0,772	0,772
NOR	0,895	0,900	0,897
EGY	0,840	0,857	0,848
MSA	0,849	0,821	0,835
Avg.	0,840	0,840	0,840

Table 3. Results of the Arabic DID using Cross-validation and the multi-layer perceptron classifiers

Dialect	Precision	Recall	F-Measure
GLF	0,819	0,863	0,840
LAV	0,768	0,741	0,754
NOR	0,893	0,894	0,893
EGY	0,843	0,856	0,849
MSA	0,849	0,808	0,828
Avg.	0,834	0,834	0,834

Table 4. Results of the Arabic DID using Cross-validation and the voting ensemble classifiers

² <http://ttg.uni-saarland.de/varDial2017/sharedtask2017.html>

Dialect	Precision	Recall	F-Measure
GLF	0,820	0,818	0,819
LAV	0,737	0,716	0,726
NOR	0,878	0,885	0,881
EGY	0,812	0,821	0,816
MSA	0,790	0,803	0,797
Avg.	0,809	0,810	0,810

Table 5. Results of the Arabic DID using Cross-validation and the Logistic classification algorithm

5. CONCLUSION

In this paper, we presented the methodology and results related to our participation at ASRU MGB Arabic DID shared task; which is considered as a very hard and challenging task. We studied the impact of the character n-gram model and its combination with i-vectors representation.

In this shared task, we used supervised learning methods that emphasize only on labelled data, to discriminate and distinguish between four major Arabic dialects: Egyptian, Levantine, North African, Gulf and Modern Standard Arabic (MSA).

Our proposed approach for Arabic DID focuses on the concatenation of i-vector audio representation vectors with bi-gram character-based vectors.

We used several machine learning algorithms and their combinations involving Voting Ensemble, Multi-Layer perceptron and Logistic classifiers.

Our best results were obtained during this shared task on closed submission using the Voting Ensemble with an overall accuracy of 56.10, followed by the simple logistic and Multi-Layer perceptron with an overall accuracy of 54.56 and 53.55, respectively.

As for future work, it would be interesting to explore semi-supervised learning algorithms using more unlabelled data. Another interesting extension to this work is to study a hybrid model for dialect identification involving character-based and word-based models.

6. REFERENCES

[1] Liu, G., Lei, Y., and Hansen, J. H. 2010. Dialect identification: Impact of differences between read versus spontaneous speech. EUSIPCO- 2010: European Signal Processing Conference, Aalborg, Denmark, 2010.

[2] Ali, A., Dehak, N., Cardinal, P., Khurana, S., Yella, S. H., Glass, J., ... & Renals, S. 2015. Automatic dialect detection in arabic broadcast speech. *arXiv preprint arXiv:1509.06928*.

[3] Zampieri, M., Malmasi, S., Ljubešić, N., Nakov, P., Ali, A., Tiedemann, J., ... & Aepli, N. 2017. Findings of the VarDial Evaluation Campaign 2017.

[4] Dehak, N., Torres-Carrasquillo, P. A., Reynolds, D. A., & Dehak, R. 2011. Language Recognition via i-vectors and Dimensionality Reduction. In *Interspeech* (pp. 857-860).

[5] Bahari, M. H., Dehak, N., Burget, L., Ali, A. M., & Glass, J. 2014. Non-negative factor analysis of gaussian mixture model weight adaptation for language and dialect recognition. *IEEE/ACM transactions on audio, speech, and language processing*, 22(7), 1117-1129.

[6] Malmasi, S., & Dras, M. 2015. Language identification using classifier ensembles. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)* (pp. 35-43).

[7] Malmasi, S., Dras, M., & Zampieri, M. 2016. LTG at SemEval-2016 Task 11: Complex Word Identification with Classifier Ensembles. *Proceedings of SemEval*, 996-1000.

[8] Xiang, Y., Wang, X., Han, W., & Hong, Q. 2015. Chinese grammatical error diagnosis using ensemble learning. *ACL-IJCNLP 2015*, 99.

[9] Malmasi, S., & Dras, M. 2015. Language identification using classifier ensembles. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)* (pp. 35-43).

[10] Le Cessie, S., & Van Houwelingen, J. C. 1992. Ridge estimators in logistic regression. *Applied statistics*, 191-201.

[11] Arora, R. 2012. Comparative analysis of classification algorithms on different datasets using WEKA. *International Journal of Computer Applications*, 54(13).

[12] Ahmed Ali, Stephan Vogell, Steve Renals. 2017. SPEECH RECOGNITION CHALLENGE IN THE WILD: ARABIC MGB-3 (draft). In *Proceedings of MGB3-ASRU 2017 conference*. Okinawa, Japan. December 2017.

[13] Ali, A., Dehak, N., Cardinal, P., Khurana, S., Yella, S.H., Glass, J., Bell, P., Renals, S. 2016. Automatic Dialect Detection in Arabic Broadcast Speech. *Proc. Interspeech 2016*, 2934-2938.

[14] Sadat, F., F. Kazemi, and A. Farzindar. 2014. Automatic Identification of Arabic Language Varieties and Dialects in Social Media. In *proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP): Association for Computational Linguistics and Dublin City University*, pp. 22-27, 08/2014.

[15] Sadat, F., F. Kazemi, and A. Farzindar. 2014. Automatic Identification of Arabic Dialects in Social Media. *SoMeRA 2014: International Workshop on Social Media Retrieval and Analysis: SIGIR 2014*, pp. 6 pages, 07/2014.

- [16] Elfardy H. and Diab M. 2013. Sentence-Level Dialect Identification in Arabic, In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, Sofia, Bulgaria. 2013.
- [17] Kirchhoff K. and Vergyri D. 2004. Cross-dialectal acoustic data sharing for arabic speech recognition. In Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on, volume 1, pages 1-765. IEEE, 2004.
- [18] Almeman K. and Lee M. 2013. Automatic building of arabic multi dialect text corpora by bootstrapping dialect words. In Communications, Signal Processing, and their Applications (ICCSPA), 2013.
- [19] Dehak, N., Dehak, R., Kenny, P., Brummer, N., Ouellet, P., and Dumouchel, P. 2009. Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification,” in *Proceedings of Interspeech*, Brighton, UK, 2009.
- [20] Burget, L., Oldřich, P., Sandro, C., Glembek, O., Matejka, P. and Brummer, N. 2011. Discriminantly trained probabilistic linear discriminant analysis for speaker verification,” in *Proceedings of ICASSP*, Prague, Czech Republic, 2011.
- [21] Crain, S. P., Zhou, K., Yang, S-H., Zha, H. 2012. Dimensionality Reduction and Topic Modeling: From Latent Semantic Indexing to Latent Dirichlet Allocation and Beyond. In *Mining Text Data*. Pp. 129-161.
- [22] Elizalde B., Lei H., Friedland G. 2013. An I-Vector based Approach for Audio Scene Detection. In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*.
- [23] Ionescu R.T., Popescu M., and Cahill, A. 2016. String kernels for native language identification: Insights from behind the curtains. *Computational Linguistics*, 42(3):491–525.

Building the Moroccan Darija Wordnet (MDW) using Bilingual Resources

Khalil Mrini¹, Francis Bond²

¹Ecole Polytechnique Fédérale de Lausanne, Switzerland

²Nanyang Technological University, Singapore

khalil.mrini@epfl.ch, bond@ieee.org

Abstract

Moroccan Darija is one of the Arabic dialects, a continuum of under-resourced vernaculars. We develop a Moroccan Darija Wordnet (MDW) using a bilingual Moroccan-English dictionary, from which we collect nearly 13,000 definitions and over 15,000 lemmas. A Moroccan alphabet is set to make the MDW user-friendly. We link the Moroccan-English definitions to the Princeton WordNet using a method that found matches for about 77% of these, and estimated accuracy using confidence scores. Over 2,300 Moroccan synsets were verified as a first step of manual validation and are now included in the MDW, which is released as part of the Open Multilingual WordNet.

Index Terms: Moroccan Darija, WordNet, Arabic Dialect, Language Resource, Under-resourced Language

1. Introduction

Moroccan Darija is one of many variants of the Arabic language that can be defined as "informal spoken dialects that are the media of communication for daily life" [1]. Ethnologue lists it as *Arabic, Moroccan Spoken* (ISO 639-3 **ary**) [2]. In the 2014 census by Morocco's Higher Planning Commission [3], it is reported that Morocco has around 33.6 million inhabitants, and that 90.9% of them speak Moroccan Darija. It is therefore spoken throughout the country, albeit with small regional differences. It is called a dialect in Morocco, and as such has no standard orthography or official alphabet.

This paper is about the development of the Moroccan Darija Wordnet (MDW). It is released as part of the freely available Open Multilingual WordNet (OMW) [4, 5], and is the first dialect to be included in it. It is not however the first wordnet for a dialect, as a wordnet for the Iraqi dialect (IAWN) [6] was developed.

In this paper, we first describe the WordNet and previous work in Languages Resources for Arabic Dialects and in WordNet linking. Then, a dictionary of Moroccan Darija to English [7] is used to get the vocabulary for the MDW, for which an alphabet is set. Finally, the Moroccan vocabulary is linked to the English-language Princeton WordNet [8].

2. Related Work

2.1. Language Resources and Work on Arabic Dialects

Arabic dialects remain considered as under-resourced languages [9] and limited work has been done around them in Natural Language Processing.

Cavalli-Sforza et al. [6] develop an Iraqi Arabic WordNet (IAWN) and a method to link Arabic dialects to the PWN, as well as to the Arabic WordNet (AWN) [10]. The assumption used to make the IAWN is that a dialect is close enough to its

base language so that their respective wordnets would have similar structures. This assumption is not used for the development of the MDW.

Habash and Rambow [11] present a morphological analyser for Modern Standard Arabic (MSA), and then adapt it for Levantine Arabic with linguistic data from Jordan. Chiang et al. [12] also exploit the similarities between MSA and Levantine Arabic to study the parsing of the latter through experiments in sentence and grammar transduction. Zbib et al. [9] crowdsource large English-Egyptian and English-Levantine parallel corpora to create a Machine Translation model.

Belgacem et al. [13] build a vocal corpus for nine dialects (Morocco, Algeria, Tunisia, Egypt, Lebanon, Syria, Iraq, the GCC countries, Yemen) and then propose a model for automatic recognition of Arabic dialects using Gaussian Mixture Models. Their results show how complex it is to distinguish Arabic dialects as they form a continuum.

2.2. Moroccan Darija Language Resources

Tachicart et al. [14] develop the Moroccan Dialect Electronic Dictionary (MDED). MDED is a MSA-Moroccan bilingual dictionary of 15,000 entries, obtained by translating an MSA dictionary to Moroccan Darija, and by translating a Moroccan Darija dictionary to MSA. The Moroccan Darija dictionary used was the French-Moroccan bilingual "*Dictionnaire Colin d'arabe dialectal marocain*" [15]. The alphabet used in MDED for Moroccan Darija is the Arabic one.

Samih and Maier [16] introduce an annotated Moroccan Darija code-switched corpus influenced by MSA. They demonstrate its possible uses by using it to detect code-switching at token and text level in Moroccan social media [17].

2.3. The Princeton WordNet

The WordNet is a lexical database created in the Cognitive Science Laboratory of Princeton University. It was first a database for the English language known as the Princeton WordNet [18, 19]. It regroups words by meaning in synsets, which are unordered sets of synonyms. Its latest release (WordNet 3.0) contains more than 150,000 unique words and around 120,000 synsets and is available for use via a web browser¹.

The WordNet divides synsets into four main parts of speech: nouns, verbs, adjectives and adverbs. However, it does not reference function words like prepositions and determiners. Therefore, these are excluded from the Moroccan Darija Wordnet. The noun synsets are also connected through relationships such as hyponymy, hypernymy, meronymy and homonymy. Likewise, verb synsets can have relationships such as hypernymy, troponymy and entailment. The advantage of the Word-

¹Accessible on <http://wordnetweb.princeton.edu/perl/webwn>

Net is that correctly connecting a new language's synsets to the existing synsets keeps most of these relationships relevant.

2.4. The Open Multilingual WordNet

Wordnets were created for other languages and linked to the Princeton WordNet to form the Open Multilingual WordNet (OMW) [5]. It is akin to a programming-ready multilingual dictionary mainly used in applications of Machine Learning and Natural Language Processing, as it can be accessed through the Python-based NLTK package [20].

At the start of the OMW, it contained 26 wordnets. As of the time of writing, it lists 34 open wordnets merged. Linking the MDW to the OMW will therefore link it to 34 other wordnets and to 33 other languages, including Afro-Asiatic languages such as Arabic [10], Indo-European languages like French [21] and East Asian languages like Chinese [22, 23].

2.5. WordNet-linking approaches

Most of the freely available wordnets use the *expand* approach [24], which is mapping lemmas of the new language to the existing PWN English synsets. An English-Moroccan dictionary therefore enables the use of this approach for the MDW. Wordnets such as the Thai [25] and Indonesian [26] ones use it, although it is recognised as an imperfect method [27]. The approach in this paper is similar to the one presented in [28]. It uses two WordNet-linking attempts, and confidence scores to gauge the accuracy of the links they establish.

There exists also the *merge* approach for wordnet construction, which consists of building an independent monolingual wordnet and then mapping it to the PWN using bilingual resources. The EuroWordNet [29] is a multilingual wordnet project that uses this approach and has introduced an interlingual index (ILI) [30] to merge ontologies between different languages. Bond et al. [31] propose a collaborative ILI to expand it to other languages. The merge approach is used by wordnets such as the Urdu one [32], as well as along with the expand approach for the Russian one [33].

3. Creating a Moroccan Darija Wordnet

The OMW is based on the English-language Princeton WordNet, and it is the wordnet with the biggest number of synsets. Therefore, it makes sense to look for English definitions of Moroccan words and then link them to the Princeton WordNet. Moreover, even though there are French-Moroccan dictionaries, using them could cause inaccuracies as the French WordNet is sometimes deemed not as semantically reliable as the English one. Therefore an English-Moroccan bilingual dictionary would be most useful.

3.1. The Bilingual Dictionary

The dictionary used as basis for the MDW is the 1963 *A Dictionary of Moroccan Arabic: Moroccan-English*, edited by Richard S. Harrell. The entries were compiled by Thomas Fox and Mohammed Abu-Talib. Their goal was to collect the words that make up the core everyday vocabulary used by Moroccans. The dictionary does not cover dialect variations or terms deemed too technical. The entries were collected by interviewing educated Moroccans in three cities: Rabat, the capital; Casablanca, the most populated city; and Fez, the second-most populated city, according to the 2014 census. The choice of cities therefore ensures that the Moroccan Darija vocabulary

banefsež same as *bellefžuž*
banka pl. -t bank *mša xerrež le-flus men l-banka*. He went and got the money from the bank.
ħanyu pl. -yat, -wat bathtub
baqa pl. -t bouquet (flowers)
baqa used in the expr. *baqa^o li-llah* said interjectionally upon hearing of the death of s.o. (implying a "that's-the-way-it-goes" idea)
baqi a.p. of *bqa*
baqiya pl. -t rest, remainder
bar ibur v.i. 1. to be left over *had s-selħa bar-et-lna*. We have this merchandise left over. 2. to be or become an old maid *dak l-bent ġadi tbur tul ħayatha*. That girl is going to be an old maid all her life.
 ¶ *ħateq bayra* pl. *ħwateq bayrin* spinster

Figure 1: Example of entries in Richard S. Harrell's *A Dictionary of Moroccan Arabic: Moroccan-English*

collected is typical Moroccan urban speech, understood in most parts of the country.

The entries are Moroccan lemmas followed by their English definitions or translations. Examples of entries can be seen in Figure 1. These could be either their direct translations, or a definition for words that are more specific. Another type of entry has one or more Moroccan lemmas followed by "same as" and one or more Moroccan lemmas, which have each an English definition entry somewhere else in the dictionary. Verbal nouns are denoted as "[verbal noun] v.n. of [corresponding verb]" in a separate entry, or as "[verb] v.n. [corresponding verbal noun]" in the entry of the corresponding verb. The plural forms, feminine forms, active and past principles are denoted likewise. Only the Moroccan-English definitions and "[...] same as [...]" entries are considered for simplification purposes, as other forms are inflected ones and cannot be linked to the WordNet.

In total, the dictionary contains 12,923 Moroccan-English definitions, or Moroccan synsets, with 14,409 lemmas, to which 720 are added from the "same as" entries, making a total of 15,129 lemmas.

3.2. Alphabet

There are existing Arabic transliterations such as the one proposed in [34] and the romanized Arabic transliteration engine proposed in [35]. However, the Moroccan Darija Wordnet is meant to be user-friendly and therefore it must be largely inspired by the unofficial alphabet that is used in daily life by Moroccans. Moreover, there are Arabic letters not used in Moroccan Darija, such as ذ, ث, and Moroccan sounds that cannot be denoted by the Arabic alphabet, such as *l* (*l-lur*), *r* and *z* (*zərbiya*).

The alphabet used by the Moroccan-English dictionary is based on the Latin alphabet and differentiates between short and long vowels. It denotes sounds not present in the Latin alphabet by Arabic letters, unlike the unofficial alphabet, or by using dots below letters for emphatic letters. Emphatic letters in the Arabic alphabet are ض (*ḍ*), ط (*ṭ*), ظ (*ḏ* or *ṭ*) and ص (*ṣ*). Their non-

Dictionary Alphabet	ġ	ح	ž	?	ء
MDW Alphabet	8	7	j	2	3

Table 1: Differences between the dictionary’s alphabet and the one used in the Moroccan Darija Wordnet (MDW)

emphatic equivalents are respectively *د* (*d*), *ت* (*t*), *ذ* (*ð*) and *س* (*s*). The dictionary’s alphabet is based on phonology and has a one-letter-one-sound rule.

The alphabet used in this paper is as close as possible to the unofficial Moroccan alphabet to make it readable for the Moroccan public. The phonology rule was kept, and the alphabet includes emphatic letters and distinguishes between short and long vowels. In keeping with the writing that Moroccans use in daily communications, the Arabic letters were replaced by their respective numerals. The letters that were changed are referenced in Table 1. All other letters (*a, ă, b, d, đ, e, f, g, h, i, ĩ, k, l, ĩ, m, n, o, q, r, s, š, t, ũ, v, w, x, y, z, ž*) remain the same. The correspondance between both alphabets is reversible. In future work, we would like to explore how to facilitate looking up a lemma in the MDW, as errors may be frequent given that the orthography is not set and depends on pronunciation.

KibDarija [36] is an example of a project that tries to define a specific alphabet for Moroccan Darija. It defines an alphabet in Arabic and an equivalent one in Latin letters with accents, but with no numbers. It follows Moroccan phonology as close as possible in both versions. To do so, it also associates one letter with one sound.

4. Linking to the WordNet

The Moroccan-English definitions are connected to the WordNet with a linking method consisting of two attempts. Then, we validate the results manually.

4.1. WordNet-linking Method

To connect Moroccan-English definitions to WordNet synsets, those definitions are first separated and given a Moroccan Darija Wordnet ID. Then there are two attempts, with the pseudocode of the first one provided in Algorithm 1.

In both attempts, everything that is between parentheses is discarded, as it could be noise or explanations that did not contain the bulk of the definition. With regards to the format of the dictionary, in which verb definitions start with "to", the part-of-speech tag is assumed to be *verb* when that occurs and therefore the definition would only be connected to synsets referring to verbs (lines 3 to 5 in Algorithm 1). Moreover, Part-of-Speech (PoS) tags in the beginning of the definition are taken into account to search for the correct PoS tag in the WordNet (lines 23 and 27 in Algorithm 1).

Each definition is split in one or more sub-definitions by commas or semicolons (line 6 in Algorithm 1). Each word in a sub-definition is filtered and thrown out if it belongs to the English "stop words" list of NLTK [20], with the exception of words such as "up, down, out, in, on, off" which accompany a verb and change its sense. A stop word is kept in the exceptional case where there is only one word in the sub-definition (lines 10 to 22 in Algorithm 1). The first attempt is stricter than the second. The second attempt was necessary to pick up matches with the WordNet that the first attempt could not establish.

4.1.1. First Attempt

In the first attempt, when searching for matching WordNet synsets, the lemma search uses underscore ("_") to connect the words in each filtered sub-definition (line 23 in Algorithm 1). For instance, the verb "2amen b-", with English definition "to believe in", is recognised as a verb because the definition starts with *to*. The WordNet is queried for "believe_in" with the part-of-speech tag being verb. The query yields one synset.

If searching with words connected with an underscore does not give results, the search takes each word in a sub-definition and matches it with synsets to form one set of synsets per word (line 27 in Algorithm 1). The final set of synsets for the sub-definition is the intersection of each word’s set of synsets (line 29 in Algorithm 1). The sub-definition is considered only if its set is non-empty. Likewise, the final set of synsets for the definition is the intersection each of its sub-definitions’ set of synsets (line 33 in Algorithm 1). If the definition’s final set is non-empty, the synsets are associated to it. Otherwise, the definition cannot be connected to WordNet in the first attempt. The confidence score given to each match is 1.0 divided by the number of synsets.

As an example, the Moroccan noun "2asel" has the English definition "origin, lineage". We query the WordNet for "origin" and "lineage", and we get respectively 6 and 5 synsets. These sets of synsets have as intersection 1 synset, which becomes the synset corresponding to this Moroccan noun. The confidence score given is 1.0.

4.1.2. Second Attempt

While the first attempt only considers one possible set of WordNet synsets per Moroccan synset, the second attempt considers more than one. The second attempt repeats the same WordNet-matching method as the first attempt, starting by joining words with an underscore, and then splitting them if no results are obtained. However, it considers each sub-definition as an independent definition. Therefore, this attempt returns the final sets of synsets of individual sub-definitions rather than their intersection.

For example, if a definition has two sub-definitions, each with a non-empty set of synsets, that definition is split into two Moroccan synsets, respectively associated with the two sets of synsets. It is the case of the Moroccan noun "amir" with English definition "emir, prince". The WordNet queries for the words "emir" and "prince" give 1 distinct synset each, and therefore do not overlap. Therefore this definition’s two sub-definitions "emir" and "prince" are considered as two separate definitions, each with 1 link to the WordNet.

The confidence score given to each match in the second attempt is 0.7 divided by the number of synsets. Therefore the maximum confidence score here is 0.7, hereby penalising the flexibility of this second run. In the above example, the two sub-definitions, which are linked to 1 synset each, both have 0.7 as confidence score.

4.2. Results and Validation

The matches resulting from the WordNet-linking method are given in Table 2.

The two attempts have resulted in a total of 12,224 Moroccan synsets connected to the WordNet. The 2,936 unconnected Moroccan synsets are mostly specific words embedded in the Moroccan culture and thus are not present in the WordNet. We have also left a list of 1,877 verbal nouns which are associated

Attempt	Synsets linked out of the total of 12, 923	Links with 1 synset out of attempt's links	Links with 2 synsets out of attempt's links
First	59.6% (7, 704)	33.0% (2, 540)	17.6% (1, 355)
Second	17.7% (2, 283), split into 4, 520 sub-definitions	22.0% (998)	18.6% (825)

Table 2: Results of the two attempts of the WordNet-linking method for the Moroccan-English dictionary

to verbs in the dictionary. In future work, they could be associated to the Moroccan Darija Wordnet through an additional relation.

The confidence scores enable us to quantify the accuracy of a WordNet link for manual validation. The validation here was conducted by one of the authors, who is a native speaker of Moroccan Darija. We first selected the 2,540 Moroccan synsets that are linked to the WordNet with confidence score 1.0. This confidence score means that they were each matched to exactly 1 synset.

Each link between a Moroccan synset and a WordNet synset is validated or rejected using the lemmas and definitions in the Moroccan-English dictionary and in the WordNet. During the validation, 8.7% (221) of the synsets were rejected. A main source of the errors is that there are adjectives and adverbs that are not tagged as such in the dictionary. Another error in linking are concepts which definitions contain light verbs, such as *get* and *make*, or started with *kind of*.

The 2,319 (91.3%) correctly linked Moroccan synsets are added to the version of the Moroccan Darija Wordnet that is now part of the OMW. They correspond to 2,571 Moroccan lemmas. In the future, we aim at completing the validation for all WordNet links to the MDW and add the verified data consequently.

5. Conclusions

We propose a Moroccan Darija Wordnet (MDW), with an alphabet close to the informal one popularly used in Morocco while complying with a one-letter-one-sound rule. The MDW is released as an extension to the Open Multilingual WordNet and is linked to the Princeton WordNet with a lexicon from a bilingual Moroccan-English dictionary. The dictionary has 12,923 Moroccan-English definitions, totalling 15,129 lemmas.

The dictionary is connected to the WordNet using two synset-connecting algorithms. Both attempts ignore English stop words with the exception of one-word definitions and separate verbs from nouns, adjectives and adverbs. The first one queries the WordNet for synsets and tries to find the common ones between words, cutting definitions into sub-definitions by commas or semicolons. The second one allows for more than one Moroccan synset per Moroccan-English definition. Both attempts attribute confidence scores to the matches they establish.

The first attempt has connected 59.6% of the Moroccan definitions, and the second one has linked an additional 17.7%. They both resulted in 12,224 Moroccan-English definitions or sub-definitions connected to the WordNet. The 2,936 definitions remaining with no WordNet link are mostly words embedded in Moroccan culture and have no equivalent in the WordNet.

At the time of writing, all 2,540 Moroccan synsets with a 1.0 confidence score have been verified manually and 91.3% (2,319) passed the test. These are now included in the Moroccan Darija Wordnet. The latter will be enlarged as more synsets are validated.

Algorithm 1 First Attempt in WordNet Matching

```

1: function FIRST-MATCH(entry) ▷ Where entry is an entry
   in the dictionary
2:   Let def be the definition in entry, id its ID, and pos
   its WordNet Part-of-Speech Tag
3:   if pos starts with "to" then
4:     pos = wordnet.VERB
5:   end if
6:   subDefs = def.split(',',';')
7:   Let finalSynsetSet and defSenses be empty arrays
8:   for subDef in subDefs do
9:     words = subDef.split('')
10:    if pos == wordnet.VERB then
11:      for word in words do
12:        if size(subDef) > 1 and word in
   stopwords except (up, down, out, in, on, off) then
13:          Remove word from words
14:        end if
15:      end for
16:    else
17:      for word in words do
18:        if size(subDef) > 1 and word in
   stopwords then
19:          Remove word from words
20:        end if
21:      end for
22:    end if
23:    union = wordnet.synsets('_.join(words), pos =
   pos)
24:    if union is empty then
25:      Let subDefSenses be an empty array
26:    for word in words do
27:      Append wordnet.synsets(word, pos =
   pos) to subDefSenses
28:    end for
29:    union = intersection of non-empty sets in
   subDefSenses
30:    end if
31:    Append union to defSenses
32:  end for
33:  finalSynsetSet = intersection of non-empty sets in
   defSenses
34:  return finalSynsetSet
35: end function

```

6. References

- [1] N. Y. Habash, "Introduction to arabic natural language processing," *Synthesis Lectures on Human Language Technologies*, vol. 3, no. 1, pp. 1–187, 2010.
- [2] M. P. Lewis, G. F. Simons, and C. D. Fennig, Eds., *Ethnologue: Languages of the World*, 18th ed. SIL International, 2015, <http://www.ethnologue.com/18/>.
- [3] H. C. P. Haut Commissariat au Plan du Maroc, "Recensement de la population," 2014.
- [4] F. Bond and K. Paik, "A survey of wordnets and their licenses," in *Proceedings of the 6th Global WordNet Conference, Matsue, Japan*, 2012, pp. 64–71.
- [5] F. Bond and R. Foster, "Linking and extending an open multilingual wordnet," in *51st Annual Meeting of the Association for Computational Linguistics: ACL-2013, Sofia*, 2013, pp. 1352–1362.
- [6] V. Cavalli-Sforza, H. Saddiki, K. Bouzoubaa, L. Abouenour, M. Maamouri, and E. Goshey, "Bootstrapping a wordnet for an arabic dialect from other wordnets and dictionary resources," in *Computer systems and applications (aiccsa), 2013 acs international conference on*. IEEE, 2013, pp. 1–8.
- [7] R. S. Harrell, "A dictionary of moroccan arabic: Moroccan-english," *Georgetown University Press*, 1963.
- [8] C. Fellbaum, "Wordnet: An electronic lexical database," 1998, mIT Press.
- [9] R. Zbib, E. Malchiodi, J. Devlin, D. Stallard, S. Matsoukas, R. Schwartz, J. Makhoul, O. F. Zaidan, and C. Callison-Burch, "Machine translation of arabic dialects," in *Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies*. Association for Computational Linguistics, 2012, pp. 49–59.
- [10] W. Black, S. Elkateb, H. Rodriguez, M. Alkhalifa, P. Vossen, A. Pease, M. Bertran, and C. Fellbaum, "The arabic wordnet project," in *Proceedings of International Conference on Language Resources and Evaluation*, 2006.
- [11] N. Habash and O. Rambow, "Magead: a morphological analyzer and generator for the arabic dialects," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2006, pp. 681–688.
- [12] D. Chiang, M. Diab, N. Habash, O. Rambow, and S. Shareef, "Parsing arabic dialects," in *11th Conference of the European Chapter of the Association for Computational Linguistics*, 2006.
- [13] M. Belgacem, G. Antoniadis, and L. Besacier, "Automatic identification of arabic dialects," in *Proceedings of International Conference on Language Resources and Evaluation*, 2010.
- [14] R. Tachicart, K. Bouzoubaa, and H. Jaafar, "Building a moroccan dialect electronic dictionary (mded)," in *Proceedings of the 5th International Conference on Arabic Language Processing*, 2014.
- [15] Z. Iraqui-Sinaceur, *Le dictionnaire Colin d'arabe dialectal marocain*. Al Manahil, 1994.
- [16] Y. Samih and W. Maier, "An arabic-moroccan darija code-switched corpus," in *In Proceedings of the International Conference on Language Resources and Evaluation*, 2016.
- [17] —, "Detecting code-switching in moroccan arabic social media," *Proceedings of SocialNLP@IJCAI-2016, New York*, 2016.
- [18] C. Fellbaum, Ed., *WordNet: An Electronic Lexical Database*. MIT Press, 1998.
- [19] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [20] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly, 2009, (www.nltk.org/book).
- [21] B. Sagot and D. Fišer, "Building a free french wordnet from multilingual resources," in *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08), Marrakech, Morocco*, 2008.
- [22] S. Wang and F. Bond, "Building the chinese open wordnet (cow): Starting from core synsets," in *Sixth International Joint Conference on Natural Language Processing*, 2013, pp. 10–18.
- [23] C.-R. Huang, S.-K. Hsieh, J.-F. Hong, Y.-Z. Chen, I.-L. Su, Y.-X. Chen, and S.-W. Huang, "Chinese wordnet: Design and implementation of a cross-lingual knowledge processing infrastructure," *Journal of Chinese Information Processing*, vol. 24, no. 2, pp. 14–23, 2010, (in Chinese).
- [24] P. Vossen, "Building wordnets," 2005, accessed: 2017-08-07. [Online]. Available: <http://www.globalwordnet.org/gwa/BuildingWordnets.ppt>
- [25] S. Thoongsup, K. Robkop, C. Mokratat, T. Sinthurahat, T. Charoenporn, V. Sornlertlamvanich, and H. Isahara, "Thai wordnet construction," in *Proceedings of the 7th workshop on Asian language resources*. Association for Computational Linguistics, 2009, pp. 139–144.
- [26] D. D. Putra, A. Arfan, and R. Manurung, "Building an indonesian wordnet," in *Proceedings of the 2nd International MALINDO Workshop*, 2008, pp. 12–13.
- [27] C. Fellbaum and P. Vossen, "Challenges for a multilingual wordnet," *Language Resources and Evaluation*, vol. 46, no. 2, pp. 313–326, 2012.
- [28] K. Mrini and M. Benjamin, "Towards producing human-validated translation resources for the fula language through wordnet linking," in *Proceedings of the Workshop on Human-informed Translation and Interpreting Technology, held in conjunction with RANLP 2017, Varna, Bulgaria*, 2017, pp. 58–64.
- [29] P. Vossen, "Introduction to eurowordnet," *Computers and the Humanities*, vol. 32, no. 2-3, pp. 73–89, 1998.
- [30] C. Fellbaum and P. Vossen, "Challenges for a global wordnet," in *Online Proceedings of the First International Workshop on Global Interoperability for Language Resources*, 2008, pp. 75–82.
- [31] F. Bond, P. Vossen, J. P. McCrae, and C. Fellbaum, "Cili: the collaborative interlingual index," in *Proceedings of the Global WordNet Conference*, vol. 2016, 2016.
- [32] A. Zafar, A. Mahmood, F. Abdullah, S. Zahid, S. Hussain, and A. Mustafa, "Developing urdu wordnet using the merge approach," in *Proceedings of the Conference on Language and Technology*, 2012, pp. 55–59.
- [33] V. Balkova, A. Sukhonogov, and S. Yablonsky, "Russian wordnet," in *Proceedings of the Second Global Wordnet Conference*, 2004.
- [34] N. Habash, A. Soudi, and T. Buckwalter, "On arabic transliteration," in *Arabic computational morphology*. Springer, 2007, pp. 15–22.
- [35] A. Chalabi and H. Gerges, "Romanized arabic transliteration," 2012.
- [36] T. Daouda and N. Regragui, "Qyas kibdarija : Projet pour un double standard pour l'écriture de l'arabe marocain ou darija," 2012, (in French). [Online]. Available: <http://www.ktbdarija.com/>

A Word Embedding based Method for Question Retrieval in Community Question Answering

Nouha Othman¹, Rim Faiz², Kamel Smaili³,

¹ LARODEC, Institut Supérieur de Gestion de Tunis, Université de Tunis, Tunisia

² LARODEC, IHEC Carthage, Université de Carthage, Tunisia

³SMarT, LORIA Campus Scientifique BP 139, 54500 Vandoeuvre Lès-Nancy Cedex, France

othmannouha@gmail.com, rim.faiz@ihec.rnu.tn, smaili@loria.fr

Abstract

Community Question Answering (cQA) continues to gain momentum owing to the unceasing rise of user-generated content that dominates the web. CQA are platforms that enable people with different backgrounds to share knowledge by freely asking and answering each other. In this paper, we focus on question retrieval which is deemed to be a key task in cQA. It aims at finding similar archived questions given a new query, assuming that the answers to the similar questions should also answer the new one. This is known to be a challenging task due to the verbosity in natural language and the word mismatch between the questions. Most traditional methods measure the similarity between questions based on the bag-of-words (BOWs) representation capturing no semantics between words. In this paper, we rely on word representation to capture the words semantic information in language vector space. Questions are then ranked using cosine similarity based on the vector-based word representation for each question. Experiments conducted on large-scale cQA data show that our method gives promising results.

Index Terms: Community question answering, Question retrieval, Word embeddings, Cosine similarity

1. Introduction

Over the last decade, with the boom of Web 2.0, the world has witnessed a huge spread of user-generated content, which became a crucial source of information on internet. This brings great attention to the emerging concept of community Question Answering (cQA) that refers to platforms that enable users to interact and answer to other users' questions [10]. Nowadays, there exists a full panoply of cQA services such as Yahoo! Answers¹, Stackoverflow², MathOverflow³ and LinuxQuestions⁴. Such community services have built up massive archives of question-answer pairs that are considered as valuable resources for different tasks like question-answering [21]. The cQA archives are continuously increasing accumulating duplicated questions. As a matter of fact, users cannot easily find the good answers and consequently post new questions that already exist in the archives. In order to avoid wasting time waiting for a new answer, cQA should automatically search the community archive to verify if similar questions have previously been posted. If a similar question is found, its associated answer can

be directly returned. Owing to its importance, significant research efforts have been recently put to retrieve similar questions in cQA [21, 3, 2, 16, 22, 12]. Indeed, question retrieval is a non trivial task presenting several challenges, mainly the data sparseness, as questions in cQA are usually very short. Another great challenge is the lexical gap between the queried questions and the existing ones in the archives [21], which constitutes a real obstacle to traditional Information Retrieval (IR) models since users can formulate the same question employing different wording. For instance, the questions: *How to lose weight within a few weeks?* and *What is the best way to get slim fast?*, have the same meaning but they are lexically different. The word mismatching is a critical issue in cQA since questions are relatively short and similar ones usually have sparse representations with little word overlap. From this, it is clear that effective retrieval models for question retrieval are strongly needed to take full advantage of the sizeable community archives.

In order to bridge the lexical gap problem in cQA, most state-of-the-art studies attempt to improve the similarity measure between questions while it is hard to set a compelling similarity function for sparse and discrete representations of words. More importantly, most existing approaches neither take into account the contextual information nor capture enough semantic relations between words. Recently, novel methods for learning distributed word representations, also called word embeddings, have shown significant performance in several IR and Natural Language Processing (NLP) tasks, amongst other questions retrieval in cQA [28]. Word embeddings are low-dimensional vector representations of vocabulary words that capture semantic relationships between them. It is worth noting that to date, the investigation of word embeddings in question retrieval is still in its infancy but the studies in this line are encouraging.

Motivated by the recent success of these emerging methods, in this paper, we propose a word embedding-based method for question retrieval in cQA, *WECOSim*. Instead of representing questions as a bag of words (BoW), we suggest representing them as Bag of-Embedded-Words (BoEW) in a continuous space using word2vec, the most popular word embedding model. Questions are therefore ranked using cosine similarity based on the vector-based word representation for each question. A previous posted question is considered to be semantically similar to a queried question if their corresponding vector representations lie close to each other according to the cosine similarity measure. The previous question with the highest cosine similarity score will be returned as the most similar question to the new posted one. We test the proposed method on a large-scale real data from Yahoo! Answers. Experimental

¹<http://answers.yahoo.com/>

²<http://stackoverflow.com/>

³<http://www.mathoverflow.net>

⁴<http://www.linuxquestions.org/>

results show that our method is promising and can outperform certain state-of-the-art methods for question retrieval in cQA.

The remainder of this paper is organized as follows: In Section (2), we give an overview of the main related work on question retrieval in cQA. Then, we present in Section (3) our proposed word embedding based-method for question retrieval. Section (4) presents our experimental evaluation and Section (5) concludes our paper and outlines some perspectives.

2. Related Work

In cQA, the precision of the returned questions is crucial to ensure high quality answers. The question retrieval task is highly complex due to the lexical gap problem since the queried question and the archived ones often share very few common words or phrases.

Over the recent years, a whole host of methods have been proposed to improve question retrieval in cQA. Several works were based on the vector space model referred to as VSM to calculate the cosine similarity between a query and archived questions [7, 3]. However, the major limitation of VSM is that it favors short questions, while cQA services can handle a wide range of questions not limited to concise or factoid questions. In order to overcome the shortcoming of VSM, BM25 have been employed for question retrieval to take into consideration the question length [3]. Okapi BM25 is the most widely applied model among a family of Okapi retrieval models proposed by Robertson et al. in [15] and has proven significant performance in several IR tasks. Besides, Language Models (LM)s [4] have been also used to explicitly model queries as sequences of query terms instead of sets of terms. LMs estimate the relative likelihood for each possible successor term taking into consideration relative positions of terms. Nonetheless, such models might not be effective when there are few common words between the user’s query and the archived questions.

To overcome the vocabulary mismatch problem faced by LMs, the translation model was used to learn correlation between words based on parallel corpora and it has obtained significant performance for question retrieval. The basic intuition behind translation models is to consider question-answer pairs as parallel texts, then relationship of words can be constructed by learning word-to-word translation probabilities such as in [21, 2]. Within the same context, [1] presented a parallel dataset for training statistical word translation models, composed of the definitions and glosses provided for the same term by different lexical semantic resources. In [24], the authors tried to improve the word-based translation model by adding some contextual information when building the translation of phrases as a whole, instead of translating separate words. In [16], the word-based translation model was extended by incorporating semantic information (entities) and explored strategies to learn the translation probabilities between words and concepts using the cQA archives and an entity catalog. Although, the aforementioned basic models have yielded good results, questions and answers are not really parallel, rather they are different from the information they contain [22].

Advanced semantic based approaches were required to further tackle the lexical gap problem and to push the question retrieval task in cQA to the next level. For instance, there were few attempts that have exploited the available category information for question retrieval like in [4, 3, 27]. Despite the fact that these attempts have proven to significantly improve the performance of the language model for question retrieval, the use of category information was restricted to the language model.

Wang et al [20] used a parser to build syntactic trees of questions, and rank them based on the similarity between their syntactic trees and that of the query question. Nevertheless, such an approach is very complex since it requires a lot of training data. As observed by [20], existing parsers are still not well-trained to parse informally written questions.

Other works model the semantic relationship between the searched questions and the candidate ones with deep question analysis such as [7] who proposed to identify the question topic and focus for question retrieval. Within this context, some studies relied on a learning-to-ranking strategy like [17] who presented an approach to rank the retrieved questions with multiple features, while [19] rank the candidate answers with a single word information instead of the combination of various features. Latent Semantic Indexing (LSI) [6] was also employed to address the given task like in [14]. While being effective to address the synonymy and polysemy by mapping words about the same concept next to each other, the efficiency of LSI highly depends on the data structure.

Otherwise, other works focused on the representation learning for questions, relying on an emerging model for learning distributed representations of words in a low-dimensional vector space namely Word Embedding. This latter has recently been subject of a wide interest and has shown promise in numerous NLP tasks [18, 5], in particular for question retrieval [28]. The main virtue of this unsupervised learning model is that it doesn’t need expensive annotation; it only requires a huge amount of raw textual data in its training phase. As we believe that the representation of words is vital for the question retrieval task and inspired by the success of the latter model, we rely on word embeddings to improve the question retrieval task in cQA.

3. Description of WECOSim

The intuition behind the method we propose for question retrieval, called *WECOSim*, is to transform words in each question in the community collection into continuous vectors. Unlike traditional methods which represent each question as Bag Of Words (BOWs), we propose to represent a question as a Bag-of-Embedded-Words (BoEW). The continuous word representations are learned in advance using the continuous bag-of-words (CBOW) model [11]. Each question is, therefore, be defined as a set of words embedded in a continuous space. Besides, the cosine similarity is used to calculate the similarity between the average of the word vectors corresponding to the queried question and that of each existing question in the archive. The historical questions are then ranked according to their cosine similarity scores in order to return the top ranking question having the maximum score, as the most relevant one to the new queried question. The proposed method for question retrieval in cQA consists of three steps namely, question preprocessing, word embedding learning and question ranking.

3.1. Question Preprocessing

The question preprocessing module intends to process the natural language questions and extract the useful terms in order to generate formal queries. These latter are obtained by applying text cleaning, tokenization, stopwords removal and stemming. Thus, at the end of the question preprocessing module, we obtain a set of filtered queries, each of which is formally defined as follows: $Q = \{t_1, t_2, \dots, t_q\}$ where t represents a separate term of the query Q and q denotes the number of query terms.

3.2. Word Embedding Learning

Word embedding techniques, also known as distributed semantic representations play a significant role in building continuous word vectors based on their contexts in a large corpus. They learn a low-dimensional vector for each vocabulary term in which the similarity between the word vectors can show the syntactic and semantic similarities between the corresponding words. Basically, there exist two main types of word embeddings namely Continuous Bag-of-Words model (CBOW) and Skip-gram model. The former one consists in predicting a current word given its context, while the second does the inverse predicting the contextual words given a target word in a sliding window. It is worthwhile to note that, in this work, we consider the CBOW model [11] to learn word embeddings, since it is more efficient and performs better with sizeable data than Skip-gram. As shown in Figure 1, the CBOW model predicts the center word given the representation of its surrounding words using continuous distributed bag-of-words representation of the context, hence the name CBOW. The context vector is got by

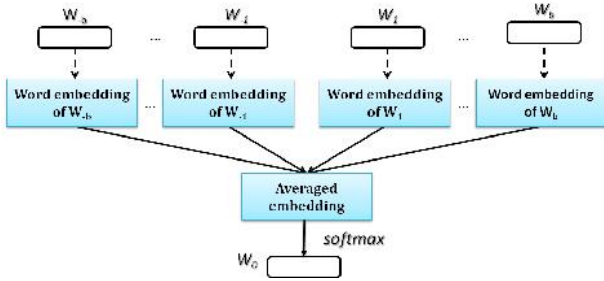


Figure 1: Overview of the Continuous Bag-of-Words model.

averaging the embeddings of each contextual word while the prediction of the center word w_0 is obtained by applying a softmax over the vocabulary V . Formally, let d be the word embedding dimension, the output matrix $O \in \mathbb{R}^{|V| \times d}$ maps the context vector c into a $|V|$ -dimensional vector representing the center word, and maximizes the following probability:

$$p(w_0 | w_{[-b,b]-\{0\}}) = \frac{\exp v_0^T O_c}{\sum_{v \in V} \exp v^T O_c} \quad (1)$$

where b is a hyperparameter defining the window of context words, O_c represents the projection of the context vector c into the vocabulary V and v is a one-hot representation. The strength of CBOW is that it does not rise substantially when we increase the window b .

3.3. Question Ranking

Once the questions are presented as Bag of-Embedded-Words (BoEW), we compute the average vector v_q of the queried question. Similarly, for each historical question, we calculate its average vector v_d . The similarity between a queried question and a historical one in the vector space is calculated as the cosine similarity between v_q and v_d .

4. Experiments

4.1. Dataset

In our experiments, we used the dataset released by [23] for evaluation. In order to construct the dataset, the authors crawled

questions from all categories in Yahoo! Answers, the most popular cQA platform, and then randomly splitted the questions into two sets while maintaining their distributions in all categories. The first set contains 1,123,034 questions as a question repository for question search, while the second is used as the test set and contains 252 queries and 1624 manually labeled relevant questions. The number of relevant questions related to each original query varies from 2 to 30. The questions are of different lengths varying from two to 15 words, in different structures and belonging to various categories e.g. Computers and Internet, Yahoo! Products, Entertainment and Music, Education and Reference, Business and Finance, Pets, Health, Sports, Travel, Diet and Fitness. Table 1 shows an example of a query and its corresponding related questions from the test set. To train the word embeddings, we resorted to another large-

Table 1: Example of questions from the test set.

Query:	How can I get skinnier without getting in a diet?
Category:	Diet and Fitness
Topic:	Weight loss
Related questions	<ul style="list-style-type: none"> - How do I get fit without changing my diet? - How can i get slim but neither diet nor exercise? - How do you get skinny fast without diet pills? - I need a solution for getting fit (loosing weight) and I must say I cant take tough diets ?

scale data set from cQA sites, namely the Yahoo! Webscope dataset⁵, including 1,256,173 questions with 2,512,345 distinct words. Some preprocessing was performed before the experiments; all questions were lower cased, tokenized, stemmed by Porter Stemmer⁶ and all stop words were removed.

4.2. Learning of Word Embedding

We trained the word embeddings on the whole Yahoo! Webscope dataset using word2vec in order to represent the words of the training data as continuous vectors which capture the contexts of the words. The training parameters of word2vec were set after several tests: the dimensionality of the feature vectors was fixed at 300 (size=300), the size of the context window was set to 10 (window=10) and the number of negative samples was set to 25 (negative=25).

4.3. Evaluation Metrics

In order to evaluate the performance of our method, we used Mean Average Precision (MAP) and Precision@n (P@n) as they are extensively used for evaluating the performance of question retrieval for cQA. Particularly, MAP is the most commonly used metric in the literature assuming that the user is interested in finding many relevant questions for each query. MAP rewards methods that not only return relevant questions early, but also get good ranking of the results. Given a set of queried questions Q , MAP represents the mean of the average precision for each queried question q and it is set as follows: $MAP = \frac{\sum_{q \in Q} AvgP(q)}{|Q|}$ where $AvgP(q)$ is the mean of the precision scores after each relevant question q is retrieved.

Precision@n returns the proportion of the top-n retrieved questions that are relevant. Given a set of queried questions

⁵The Yahoo! Webscope dataset Yahoo answers comprehensive questions and answers version 1.0.2, available at "http://research.yahoo.com/Academic_Relations"

⁶http://tartarus.org/martin/PorterStemmer/

Q , $P@n$ is the proportion of the top n retrieved questions that are relevant to the queries, and it is defined as follows: $P@n = \frac{1}{|Q|} \sum_{q \in Q} \frac{Nr}{N}$ where Nr is the number of relevant questions among the top N ranked list returned for a query q . In our experiments, we calculated $P@10$ and $P@5$.

4.4. Main Results

We compare the performance of WECOSim with the following competitive state-of-the-art question retrieval models tested by Zhang et al. in [23] on the same dataset:

- **TLM** [21]: A translation based language model which combines the translation model estimated using the question and the language model estimated using the answer part. It integrates word-to-word translation probabilities learned by exploiting various sources of information.
- **PBTM** [24]: A phrase based translation model which employs machine translation probabilities and assumes that question retrieval should be performed at the phrase level. TLM learns the probability of translating a sequence of words in a historical question into another sequence of words in a queried question.
- **ETLM** [16]: An entity based translation language model, which is an extension of TLM by replacing the word translation with entity translation in order to incorporate semantic information within the entities.
- **WKM** [29]: A world knowledge based model which used Wikipedia as an external resource to add the estimation of the term weights to the ranking function. A concept thesaurus was built based on the semantic relations extracted from the world knowledge of Wikipedia.
- **M-NET** [28]: A continuous word embedding based model, which integrates the category information of the questions to get the updated word embedding, assuming that the representations of words that belong to the same category should be close to each other.
- **ParaKCM** [23]: A key concept paraphrasing based approach which explores the translations of pivot languages and expands queries with the paraphrases. It assumes that paraphrases contributes additional semantic connection between the key concepts in the queried question and those of the historical questions.

From Table 2, we can see that PBTM outperforms TLM which demonstrates that capturing contextual information in modeling the translation of phrases as a whole or consecutive sequence of words is more effective than translating single words in isolation. This is because, by and large, there is a dependency between adjacent words in a phrase. The fact that ETLM (an

Table 2: Comparison of the question retrieval performance of different models.

	TLM	PBTM	ETLM	WKM	M-NET	ParaKCM	WECOSim
P@5	0.3238	0.3318	0.3314	0.3413	0.3686	0.3722	0.3432
P@10	0.2548	0.2603	0.2603	0.2715	0.2848	0.2889	0.2738
MAP	0.3957	0.4095	0.4073	0.4116	0.4507	0.4578	0.4125

extension of TLM) performs as good as PBTM proves that replacing the word translation by entity translation for ranking improves the performance of the translation language model.

Although, ETLM and WKM are both based on external knowledge resource e.g. Wikipedia, WKM uses wider information from the knowledge source. Specifically, WKM builds a Wikipedia thesaurus, which derives the concept relationships (e.g. synonymy, hypernymy, polysemy and associative relations) based on the structural knowledge in Wikipedia. The different relations in the thesaurus are treated according to their importance to expand the query and then enhance the traditional similarity measure for question retrieval. Nevertheless, the performance of WKM and ETLM are limited by the low coverage of the concepts of Wikipedia on the various users' questions. The results show that our method WECOSim slightly outperforms the aforementioned methods by returning a good number of relevant questions among the retrieved ones early. A possible reason behind this is that context-vector representations learned by word2vec can effectively address the word lexical gap problem by capturing semantic relations between words, while the other methods do not capture enough information about semantic equivalence. We can say that questions represented by bag-of-embedded words can be captured more accurately than traditional bag-of-words models which cannot capture neither semantics nor positions in text. This good performance indicates that the use of word embeddings along with cosine similarity is effective in the question retrieval task. However, we find that sometimes, our method fails to retrieve similar questions when questions contain misspelled query terms. For instance, questions containing *sofwar* by mistake cannot be retrieved for a query containing the term *software*. Such cases show that our approach fails to address some lexical disagreement problems. Furthermore, there are few cases where WECOSim fails to detect semantic equivalence. Some of these cases include questions having one single similar question and most words of this latter do not appear in a similar context with those of the queried question. M-NET, also based on continuous word embeddings performs better than our method owing to the use of metadata of category information to encode the properties of words, from which similar words can be grouped according to their categories. The best performance is achieved by ParaKCM, a key concept paraphrasing based approach which explores the translations of pivot languages and expands queries with the generated paraphrases for question retrieval.

5. Conclusion

In this paper, we lay out a word embedding based method to tackle the lexical gap problem in question retrieval from cQA archives. In order to find semantically similar questions to a new query, previous posted questions are ranked using cosine similarity based on their vector-based word representations in a continuous space. Experimental results conducted on large-scale cQA data show the effectiveness of the proposed method. However, word embedding models assume that each word preserves only a single vector. It is the reason why it faces lexical ambiguity due to polysemy and homonymy, and it is therefore an important problem to address. On the other hand, while the cosine similarity is shown to be effective in identifying semantically closest words, this measure becomes insufficient when the order of words is not needed. In future work, we look forward to improving our method by investigating the performance of certain powerful techniques such as Latent Semantic Indexing (LSI) along with word embeddings. We also consider incorporating various types of metadata information into the learning process in order to enrich word representations.

6. References

- [1] Bernhard, D. and Gurevych, I., “Combining lexical semantic resources with question and answer archives for translation-based answer finding”, In Proceedings of ACL, pages 728–736, 2009.
- [2] Cai, L., Zhou, G., Liu, K., and Zhao, J., “Learning the latent topics for question retrieval in community qa”, In Proceedings of IJCNLP, pages 273–281, 2011.
- [3] Cao, X., Cong, G., Cui, B., and Jensen, C. S., “A generalized framework of exploring category information for question retrieval in community question answer archives”, In Proceedings of WWW, pages 201–210, 2010.
- [4] Cao, X., Cong, G., Cui, B., Jensen, C. S., and Zhang, C., “The use of categorization information in language models for question retrieval”, In Proceedings of the 18th ACM conference on Information and knowledge management, pages 265–274, 2009.
- [5] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P., “Natural language processing (almost) from scratch”, *Journal of Machine Learning Research*, pages 2493–2537, 2011.
- [6] Deerwester, S., Dumais, S. T., Furnas, G.W., Landauer, T. K., and Harshman, R., “Indexing by latent semantic analysis”, *Journal of the American society for information science*, 41(6):391, 1990.
- [7] Duan, H., Cao, Y., Lin, C.-Y., and Yu, Y., “Searching questions by identifying question topic and question focus”, s. In Proceedings of 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL:HLT), volume 8, pages 156–164, 2008.
- [8] Kenter, T. and De Rijke, M., “Short text similarity with word embeddings”, In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pages 1411–1420, 2015.
- [9] Levy, O., Goldberg, Y., and Dagan, I., “Improving distributional similarity with lessons learned from word embeddings”, *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015.
- [10] Liu, Y., Bian, J., and Agichtein, E., “Predicting information seeker satisfaction in community question answering”, In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pages 483–490, 2008.
- [11] Mikolov, T., Chen, K., Corrado, G., and Dean, J., “Efficient estimation of word representations in vector space”, arXiv preprint arXiv:1301–3781, 2013.
- [12] Nakov, P., Hoogeveen, D., M’arquez, L., Moschitti, A., Mubarak, H., Baldwin, T., and Verspoor, K., “Semeval-2017 task 3: Community question answering”, In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval–2017), pages 27–48, 2017.
- [13] Othman, N. and Faiz, R., “A multilingual approach to improve passage retrieval for automatic question answering”, In International Conference on Applications of Natural Language to Information Systems, Springer, pages 127–139, 2016.
- [14] Qiu, X., Tian, L., and Huang, X., “Latent semantic tensor indexing for community-based question answering”, In In Proc. of the 51st Annual Meeting of the Association for Computational Linguistics, pages 434–439, 2013.
- [15] Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., Gafford, M., et al., *Okapi at TREC-3*, Nist Special Publication Sp, 109:109, 1995.
- [16] Singh, A., “Entity based q&a retrieval”, In Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pages 1266–1277, 2012.
- [17] Surdeanu, M., Ciaramita, M., and Zaragoza, H., “Learning to rank answers on large online qa collections”, . In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL:HLT), volume 8, pages 719–727, 2008.
- [18] Turian, J., Ratnoff, L., and Bengio, Y., “Word representations: a simple and general method for semisupervised learning”, In Proceedings of the 48th annual meeting of the association for computational linguistics, pages 384–394, 2010.
- [19] Wang, B., Wang, X., Sun, C., Liu, B., and Sun, L., “Modeling semantic relevance for question-answer pairs in web social communities”, In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 1230–1238, 2010.
- [20] Wang, K., Ming, Z., and Chua, T.-S., “A syntactic tree matching approach to finding similar questions in community-based qa services”, In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, pages 187–194, 2009.
- [21] Xue, X., Jeon, J., and Croft, W. B., “Retrieval models for question and answer archives”, In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pages 475–482, 2008.
- [22] Zhang, K., Wu, W., Wu, H., Li, Z., and Zhou, M., “Question retrieval with high quality answers in community question answering”. In Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, pages 371–380, 2014.
- [23] Zhang, W.-N., Ming, Z.-Y., Zhang, Y., Liu, T., and Chua, T.-S., “Capturing the semantics of key phrases using multiple languages for question retrieval”, *IEEE Transactions on Knowledge and Data Engineering*, 28(4):888–900, 2016
- [24] Zhou, G., Cai, L., Zhao, J., and Liu, K., “Phrase-based translation model for question retrieval in community question answer archives”, In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pages 653–662, 2011.
- [25] Zhou, G., Chen, Y., Zeng, D., and Zhao, J., “Towards faster and better retrieval models for question search”, In Proceedings of the 22nd ACM international conference on Conference on information and knowledge management, pages 2139–2148, 2013.
- [26] Zhou, G., He, T., Zhao, J., and Hu, P., “Learning continuous word embedding with metadata for question retrieval in community question answering”, In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, pages 250–259, 2015.
- [27] Zhou, G., Liu, Y., Liu, F., Zeng, D., and Zhao, J. (2013b). “Improving question retrieval in community question answering using world knowledge”, In IJCAI, volume 13, pages 2239–245, 2013.

Active Learning for Classifying Political Tweets

Erik Tjong Kim Sang¹, Marc Esteve del Valle², Herbert Kruitbosch², Marcel Broersma²

¹Netherlands eScience Center

²University of Groningen

e.tjongkimsang@esciencecenter.nl, {m.esteve.del.valle,h.t.kruitbosch,m.j.broersma}@rug.nl

Abstract

We examine methods for improving models for automatically labeling social media data. In particular we evaluate active learning: a method for selecting candidate training data whose labeling the classification model would benefit most of. We show that this approach requires careful experiment design, when it is combined with language modeling.

Index Terms: machine learning, active learning, social media data, political science, fastText

1. Introduction

Social media, and in particular Twitter, are important platforms for politicians to communicate with media and citizens [1]. In order to study the behavior of politicians on Twitter, we have labeled tens of thousands political tweets written in four languages (Dutch, English, Swedish and Italian) with respect to several categories, like function and topic. Labeling tweets is a time-consuming manual process which requires training of the human annotators. We would like to minimize the effort put in labeling future data and therefore we are looking for automatic methods for classifying tweets based on our annotated data sets.

The task of automatically assigning class labels to tweets is a variant of document classification. This is a well-known task for which several algorithmic solutions are known [2]. A recently developed tool for document classification is fastText [3]. It consists of a linear classifier trained on bags of character n-grams. This is a useful feature for our task: in a compound language like Dutch, useful information can be present at the character n-gram level. For example, if a word like *bittersweet* appears in the data only once, an n-gram-sensitive system could still pickup similarities between this word and the words *bitter* and *sweet*. fastText also includes learning language models from unlabeled text [4], an excellent feature for our task, where labeled data is scarce and unlabeled data is abundant.

In a typical time line of our work, we would study the tweets of politicians in the weeks preceding an election and then again in the weeks preceding the next election, some years later. Given the long time between the periods of interest, we expect that the classification model will benefit from having manually labeled data of each period. However, we would like to limit the human labeling effort because of constraints on time and resources. We will apply active learning [5] for selecting the best of the new tweets for the classification model, and label only a small selection of these tweets. Active learning has previously been used for reducing the size of candidate training data with more than 99%, without any performance loss [6].

The contribution of this paper is two-fold. Firstly, we will show that fastText can predict a non-trivial class of our political data with reasonable accuracy. Secondly, we will outline how active learning can be used together with fastText. We found that this required careful experiment design.

After this introduction, we will present some related work in Section 2. Section 3 describes our data and the machine learning methods applied in this study. The results of the experiments are presented in Section 4. In Section 5, we conclude.

2. Related work

Social media have amplified the trend towards personalization in political communication. Attention has shifted from political parties and their ideological stances to party leaders and individual politicians [7]. One way of studying personalization, is by examining the behavior of politicians on social media, in particular during campaigns leading to an election. Studies have focused on various social media like Twitter [1], Facebook [8] and Instagram [9]. Because of its open nature, Twitter is especially popular for studying online political communication [10].

Document classification is a well-known task which originates from library science. Automatic methods for performing this task, have been available for more than twenty years, for example for spam filtering [11] and topic detection in USENET newsgroups [12]. While the restricted length of social media text poses a challenge to automatic classification methods, there are still several studies that deal with this medium [13, 14]. Popular techniques for automatic document classification are Naive Bayes [15] and Support Vector Machines [16]. Despite its relatively young age, fastText [3] has also become a frequently used tool for automatic document classification and topic modeling [17, 18]. The word vector-based language models used by fastText, were originally proposed by Mikolov et al. [19].

The term of active learning was introduced in the context of machine learning in 1994 [20], referring to a form of learning where the machine can actively select its training data. Since then active learning has been applied in many contexts [5]. A well-known application in natural language processing was the study by Banko and Brill [6], which showed that with active learning, more than 99% of the candidate training data could be discarded without any performance loss.

In the study described in this paper, we employ labeled tweets developed by the Centre for Media and Journalism Studies of the University of Groningen [21]. Broersma, Graham et al. have performed several studies based on these data sets [22, 1, 23]. Most importantly for this paper, Tjong Kim Sang et al. [24] applied fastText to the Dutch 2012 part of the data set. They also evaluated active learning but observed only decreasing performance effects.

3. Data and methods

Our data consist of tweets from Dutch politicians written in the two weeks leading up to the parliament elections in The Netherlands of 12 September 2012. The tweets have been annotated by

the Groningen Centre for Media and Journalism Studies [21]. Human annotators assigned nine classes to the tweets, among which tweet topic and tweet function. In this paper we exclusively deal with the tweet function class. This class contains information about the goal of a tweet, for example campaign promotion, mobilization, spreading news or sharing personal events. A complete overview of the class labels can be found in Table 3. A tweet can only be linked to a single class label.

The data annotation process is described in Graham et al. [1]. The tweets were processed by six human annotators. Each tweet was annotated by only one annotator, except for a small set of 300 randomly chosen tweets. The small tweet subset was used for computing inter-annotator agreement for four classes with average pairwise Cohen kappa scores [25]. The kappa scores were in the range 0.66–0.97. The function class proved to be the hardest to agree on: its kappa score was 0.66. This corresponds with an pairwise inter-annotator agreement of 71%.

Twitter assigns a unique number to each tweet: the tweet id. We found that the data set contained some duplicate tweet ids. We removed all duplicates from the data set. This left 55,029 tweets. They were tokenized with the Python’s NLTK toolkit [26] and converted to lower case. Next we removed tokens which we deemed useless for our classification model over long time frames: reference to other Twitter users (also known as tweet handles), email addresses and web addresses. These were replaced by the tokens USER, MAIL and HTTP. Finally the tweets were sorted by time and divided in three parts: test (oldest 10%), development (next 10%) and train (most recent 80%). We chose to have test and development data from one end of the data set because there are strong time dependencies in the data. Random test data selection would have increased the test data scores and would have made the scores less comparable with the scores that could be attained on other data sets.

We selected the machine learning system fastText [3] for our study because it is easy to use, performs well and allows for incorporation of language models. We only changed one of the default parameter settings of fastText: the size of the numeric vectors used for representing words in the text (dim): from 100 to 300. The reason for this change was that pretrained language models often use this dimension, for example models derived from Wikipedia [4]. By using the same dimension, it becomes easier to use such external language models and compare them with our own¹. We explicitly set the minimal number of word occurrences to be included in the model (minCount) to 5. This should be the default value for this parameter but we have observed that fastText behaves differently if the parameter value is not set explicitly.

Because of the random initialization of weights in fastText, experiment results may vary. In order to be able to report reliable results, we have repeated each of our experiments at least ten times. We will present average scores of these repeated results. We found that the test evaluation of fastText (version May 2017) was unreliable, possibly because some test data items are skipped during evaluation. For this reason we did not use the test mode of the tool but rather made it predict class labels which were then compared to the gold standard by external software [27].

In active learning, different strategies can be used for selecting candidate training data. In this study, we compare four informed strategies with three baselines. Three of the informed strategies are variants of uncertainty sampling [5]. The machine

learner labeled the unlabeled tweets and the probabilities it assigned to the labels were used to determine the choices in uncertainty sampling. As an alternative, we have also experimented with query-by-committee [5]. We found that its performance for our data was similar to uncertainty sampling.

The data selection strategies used in this study are:

Sequential (baseline) choose candidate training data in chronological order, starting with the oldest data. Because there are strong time-dependent relations in our data, we also evaluate the variant **Reversed sequential** (baseline) which selects the most recent data first.

Random selection (baseline) randomly select data.

Longest text choose the longest data items first, based on the number of characters.

Least confident first select the data items with an automatically assigned label with the lowest probability.

Margin choose the data with of which the probability of the second-most likely label is closest to the probability of the most likely label

Entropy first select data items of which the entropy of the automatic candidate labels is highest.

The methods Entropy, Margin and Least confident select the data the machine learner is least confident of while Longest text selects the data that are most informative. The entropy is computed with the standard formula $-\sum_i p_i * \log_2(p_i)$ [28] where p_i is the probability assigned by the machine learner to a candidate training data item in association with one of the twelve class labels.

In their landmark paper, Banko and Brill [6] observed that having active learning select all the new training data, resulted in the new data being biased toward difficult instances. They solved this by having active learning select only half of the new training data, while selecting the other half randomly. We will adopt the same approach. Dasgupta [29] provides another motivation for this strategy: the bias of an initial model might prevent active learning from looking for solutions in certain parts of the data space. Incorporating randomly chosen training items can help the model to overcome the effect of this bias.

4. Experiments and analysis

We started our experiments with reproducing the results reported by previous work on this data set. Tjong Kim Sang [24] reported a baseline accuracy of $51.7 \pm 0.2\%$ when training fastText on the most recent 90% of the data and testing on the oldest 10% (averaged over 25 runs). We repeated this experiment and derived a model from the train and development section of the data set and evaluated this model on the test section. We obtained an accuracy of $51.6 \pm 0.7\%$, averaged over 10 runs, which is similar to the earlier reported score. This baseline score is not very high but as the low pairwise interannotator agreement (71%) showed, this is a difficult task.

In this study, we will compare several techniques and select the best. In order to avoid overfitting, we will leave this data set alone. Unless mentioned otherwise, scores reported in this paper will have been derived from testing on the development data section after training on the training data section, or a part of this section. We repeat the initial experiment, this time training fastText on the train section and evaluating on the development section. We obtained an average accuracy over ten runs of $54.2 \pm 0.4\%$, which shows that the labels of the development section are easier to predict than those of the test section.

¹See Tjong Kim Sang et al. [24] for a comparison between models build from tweets and models build from Wikipedia articles.

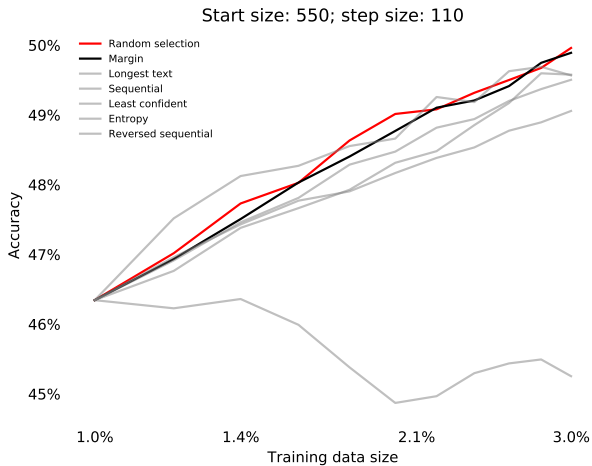


Figure 1: Performance of seven data selection methods, averaged over thirty runs. The Random selection baseline (red line) outperforms all active learning methods at 3.0% of the training data. Margin sampling (black line) is second best. There is no significant difference between the accuracies of the best six methods at 3.0% (see Table 1). Note that the horizontal axis is logarithmic.

Next, we evaluated active learning. Earlier, Tjong Kim Sang et al. [24] performed two active learning experiments. Both resulted in a decrease of performance when the newly annotated tweets were added to the training data. We do not believe that data quantity is the cause of this problem: their extra 1,000 tweets (2%) of the original training data size should be enough to boost performance (see for example Banko and Brill [6]’s excellent results with 0.7% of the training data). However, the quality of the data could be a problem. The data from the active data set and the original data set were annotated by different annotators several years apart. While there was an annotation guideline [21], it is possible that the annotators interpreted it differently. It would have been better if both training data and the active learning data had been annotated by the same annotators in the same time frame.

In order to make sure that our data was consistently annotated, we only use the available labeled data sets. We pretend that the training data is unannotated and only use the available class labels for tweets that are selected by the active learning process. The process was split in ten successive steps. It started with an initial data set of 1.0% of all labeled data, selected with the Sequential strategy. FastText learned a classification model from this set and next 0.2% of the data was selected as additional training data: 0.1% with active learning and 0.1% randomly, as described in Section 3. These steps were repeated ten times. The final training data set contained 3.0% of all labeled data. In order to obtain reliable results, the active learning process was repeated 30 times.

The random initialization of fastText pose a challenge to a successful combination with active learning. During the training process, fastText creates numeric vectors which represent the words in the data. However, when we expand the training data set and retrain the learner on the new set, these word vectors might change. This could invalidate the data selection process: the newly selected training data might work fine with

Train size	Accuracy	Method
80.0%	55.6±0.3%	Ceiling (all training data)
3.0%	50.0±0.9%	Random selection
3.0%	49.9±0.9%	Margin
3.0%	49.6±0.7%	Longest text
3.0%	49.6±0.9%	Sequential
3.0%	49.5±1.0%	Least confident
3.0%	49.1±0.8%	Entropy
3.0%	45.3±1.3%	Reversed sequential
1.0%	46.3±0.8%	Baseline

Table 1: Results of active learning experiments after training on 3.0% of the available labeled data in comparison with training on 80.0%. The Random selection baseline outperforms all evaluated active learning methods on this data set, although most of the measured differences are insignificant. Margin sampling is second-best. Numbers after the scores indicate estimated error margins ($p < 0.05$).

the old word vectors but not with the new word vectors. In order to avoid this problem, we need to use the same word vectors during an entire active learning experiment. This means that the word vectors needed to be derived for all of the current and future training data before each experiment, without using the data labels. We used the skipgram model for this, with the fastText parameter setting described in Section 3. A set of such word vectors is called a language model. Providing the machine learner with word vectors from these language models improved the accuracy score: from 54.2±0.4% to 55.6±0.3%.

The results of the active learning experiments can be found in Figure 1 and Table 1. All the data selection strategies improve performance with extra data, except for the Reversed sequential method. The initial 1.0% of training data selected with the Sequential method was a good model of the development set, since it originated from the same time frame as the development data. The data from the Reversed sequential process came from the other end of the data set and was clearly less similar to the development set.

The differences between the other six evaluated methods proved to be insignificant (see Table 1). It is unclear why neither Margin, nor Entropy, nor Least confident could outperform the Random selection baseline. Perhaps the method for estimating label probabilities (fastText-assigned confidence scores) was inadequate. However, we also evaluated bagging for estimating label probabilities and this resulted in similar performances. The Longest text method did not have access to as much information as the other three informed methods. It would be interesting to test a smarter version of this method, for instance one that preferred words unseen in the training data.

It is tempting to presume that if Margin, Longest text, Least confident and Entropy perform worse than Random selection, then their reversed versions must do better than this baseline. We have tested this and found that this was not the case. Shortest text (49.1%), Smallest entropy (48.9%), Largest margin (48.9%) and Most confident (48.8%) all perform worse than Random selection and also worse than their original variant.

Since no active learning method outperformed the random baseline, we used Random selection for our final evaluation: selecting the best additional training data while evaluating on the data sets of Tjong Kim Sang et al [24]: train (49,526 tweets), test (5,503) and unlabeled (251,279). A single human annotator labeled the selected tweets. At each iteration 110 tweets were selected randomly. After labeling, the tweets were added to

Method	Train size	Accuracy
Baseline	90.0%	51.6±0.7%
+ language model	90.0%	55.5±0.4%
+ active learning data 1	90.2%	55.4±0.4%
+ active learning data 2	90.4%	55.6±0.5%
+ active learning data 3	90.6%	55.6±0.3%

Table 2: Results of active learning (with Random selection) applied to the test set. Additional pretrained word vectors improve the classification model but active learning does not.

the training data and the process was repeated. Three iterations were performed. Each of them used the same set of skipgram word vectors, obtained from all 300,805 non-test tweets.

The result of this experiment can be found in Table 2. The extra training data only marginally improved the performance of the classifier: from 55.5% to 55.6%. The improvement was not significant. This is surprising since we work with the same amount of additional data as reported in Banko and Brill [6]: 0.6%. They report an error reduction of more than 50%, while we find no effect.

However, the percentages of added data do not tell a complete story. A close inspection of Figure 4 of the Bank and Brill paper shows that the authors added 0.6% of training data to 0.1% of initial training data. This amounts to increasing the initial training data with 600%, which must have an effect on performance, regardless of the method used for selecting the new data. Instead, we add 0.6% to 90% of initial training data, an increase of only 0.7%. Unfortunately, we don't have the resources for increasing the data volume by a factor of seven. The goal of our study was to improve classifier performance with a small amount of additional training data, not with a massive amount of extra data.

If relative data volumes are not enough to explain the differences between Table 1 and Table 2, there could be two other causes. First, the distribution of the labels of the active learning data is different from that of the original data. The latter were collected in the two weeks before the 2012 Dutch parliament elections while the first were from a larger time frame: 2009-2017. We found that the original data contained more campaign-related tweets, while the active learning data had more critical, news-related and non-political tweets (Table 3).

The second reason for the differences between Tables 1 and 2 could be low inter-annotator agreement. We have included 110 tweets from the training data in each iteration, to enable a comparison of the new annotator with the ones from 2012. While Graham et al. [1] reported an inter-annotator agreement of 71% for the 2012 labels, we found that the agreement was of the new annotator with the previous ones was only 65%, despite the fact that the annotator had access to the guesses of the prediction system. A challenge for the annotator was that some of the contexts of tweets that earlier annotators had access to, was not available on Twitter anymore and therefore could not be used for choosing the most appropriate label. The resulting lower quality of the new labels might have prevented the machine learner from achieving better performances.

5. Concluding remarks

We have evaluated a linear classifier in combination with language models and active learning on predicting the function of Dutch political tweets. In the process, we have improved the best accuracy achieved for our data set, from 54.8% [24] to

Class	Frequency		Frequency	
Campaign Promotion	12,017	(22%)	53	(16%)
Campaign Trail	10,681	(19%)	61	(18%)
Own / Party Stance	9,240	(17%)	50	(15%)
Critique	8,575	(16%)	71	(21%)
Acknowledgement	6,639	(12%)	32	(10%)
Personal	4,208	(8%)	19	(6%)
News/Report	1,662	(3%)	32	(10%)
Advice/Helping	1,292	(2%)	0	(0%)
Requesting Input	307	(1%)	0	(0%)
Campaign Action	216	(0%)	0	(0%)
Other	116	(0%)	12	(4%)
Call to Vote	76	(0%)	0	(0%)
All data	55,029	(100%)	330	(100%)

Table 3: Distribution of the function labels in the annotated data set of 55,029 Dutch political tweets from the parliament elections of 2012 (left) and the 330 tweets selected with active learning (right). The 2012 data contain more campaign-related tweets while the active learning data contain more critical, news-related and non-political tweets (class Other).

55.6%. We found that combining the classifier fastText with active learning was not trivial and required careful experiment design, with pretrained word vectors, parameter adjustments and external evaluation procedures. In a development setting, none of the evaluated four informed active learning performed better than the random baseline, although the performance differences were insignificant. In a test setting with the best data selection method (random sampling), we measured no performance improvement. The causes for this could be the small volume of the added data, label distribution differences between the new and the original training data and the fact that it was hard for annotators to label the data consistently.

We remain interested in improving the classifier so that we can base future data analysis on accurate machine-derived labels. One way to achieve this, would be re-examine the set of function labels chosen for our data set. We could make the task of the classifier easier by collapsing labels but this would make them less informative and less interesting for follow-up work. Alternatively, we could split labels, for example by creating a separate binary label for each current label value. This would make possible assigning multiple labels to one tweet, freeing the current burden of annotators of having to choose a single label even in cases where three or four different labels might be plausible. Making the task of the annotators easier would improve the inter-annotator agreement and may even improve the success of applying active learning to this data set.

How to best split the labels while still being able to use the current labels in the data, remains a topic for future work.

6. Acknowledgments

The study described in this paper was made possible by a grant received from the Netherlands eScience Center. We would like to thank three anonymous reviewers for valuable feedback on an earlier version of this paper.

7. References

- [1] Todd Graham, Dan Jackson, and Marcel Broersma. New platform, old habits? Candidates use of Twitter during the 2010 British and Dutch general election cam-

- paigns. *New Media & Society*, 18:765–783, 2016. doi:10.1177/1461444814546728.
- [2] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys (CSUR)*, 34, 2002.
- [3] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 427–431. ACL, Valencia, Spain, 2017.
- [4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association of Computational Linguistics*, 5:135–146, 2017.
- [5] Burr Settles. *Active Learning Literature Survey*. Computer Sciences Technical Report 1648, University of Wisconsin-Madison, 2010.
- [6] Michele Banko and Eric Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th annual meeting on Association for Computational Linguistics*, pages 26–33. Association for Computational Linguistics, 2001.
- [7] Peter Van Aelst, Tamir Sheafer, and James Stanyer. The personalization of mediated political communication: A review of concepts, operationalizations and key findings. *Journalism*, 13:203–220, 2012.
- [8] Gunn Sara Enli and Eli Skogerbø. Personalized campaigns in party-centered politics. *Information, Communication & Society*, 16(5):757–774, 2013. doi:10.1080/1369118X.2013.782330.
- [9] Younbo Jung, Ashley Tay, Terence Hong, Judith Ho, and Yan Hui Goh. Politicians strategic impression management on instagram. In *Proceedings of the 50th Hawaii International Conference on System Sciences (HICSS)*. IEEE, Waikoloa Village, HI, USA, 2017. doi:10.24251/HICSS.2017.265.
- [10] Andreas Jungherr. Twitter use in election campaigns: A systematic literature review. *Journal of Information Technology & Politics*, 13(1):72–91, 2016. doi:10.1080/19331681.2015.1132401.
- [11] Ion Androutsopoulos, John Koutsias, Konstantinos V. Chandrinos, George Paliouras, and Constantine D. Spyropoulos. An evaluation of naive bayesian anti-spam filtering. In G. Potamias, V. Moustakis, and M. van Someren, editors, *Proceedings of the workshop on Machine Learning in the New Information Age*, pages 9–17. Barcelona, Spain, 2000.
- [12] Scott A. Weiss, Simon Kasif, and Eric Brill. *Text Classification in USENET Newsgroups: A Progress Report*. AAAI Technical Report SS-96-05, 1996.
- [13] Liangjie Hong and Brian D. Davidson. Empirical study of topic modeling in Twitter. In *Proceedings of the First Workshop on Social Media Analytics (SOMA'10)*. ACM, Washington DC, USA, 2010.
- [14] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing Twitter and Traditional Media Using Topic Models. In *ECIR 2011: Advances in Information Retrieval*, pages 338–349. Springer, LNCS 6611, 2011.
- [15] Andrew McCallum and Kamal Nigam. A Comparison of Event Models for Naive Bayes Text Classification. In *AAAI-98 workshop on learning for text categorization*, pages 41–48, 1998.
- [16] Larry M. Manevitz and Malik Yousef. One-Class SVMs for Document Classification. *Journal of Machine Learning Research*, 2:139–154, 2001.
- [17] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. Deep Learning for Hate Speech Detection in Tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 2017. doi:10.1145/3041021.3054223.
- [18] Francesco Barbieri. Shared Task on Stance and Gender Detection in Tweets on Catalan Independence - LaSTUS System Description. In *Proceedings of the Second Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2017)*. Murcia, Spain, 2017.
- [19] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv*, 1301.3781, 2013.
- [20] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 15:201–221, 1994.
- [21] The Groningen Center for Journalism and Media Studies. *The Tweeting Candidate: The 2020 Dutch General Election Campaign: Content Analysis Manual*. University of Groningen, 2013.
- [22] Todd Graham, Marcel Broersma, Karin Hazelhoff, and Guido van t Haar. Between broadcasting political messages and interacting with voters: The use of twitter during the 2010 uk general election campaign. *Information, Communication and Society*, 16:692–716, 2013.
- [23] Marcel Broersma and Marc Esteve Del Valle. Automated analysis of online behavior on social media. In *Proceedings of the European Data and Computational Journalism Conference*. University College Dublin, 2017.
- [24] Erik Tjong Kim Sang, Herbert Kruitbosch, Marcel Broersma, and Marc Esteve del Valle. Determining the function of political tweets. In *Proceedings of the 13th IEEE International Conference on eScience (eScience 2017)*, pages 438–439. IEEE, Auckland, New Zealand, 2017. doi:10.1109/eScience.2017.60.
- [25] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), 1960.
- [26] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. O’Reilly Media, Inc., 2009.
- [27] Erik Tjong Kim Sang. *Machine Learning in project Online Behaviour*. Software repository available at <https://github.com/online-behaviour/machine-learning>, 2017.
- [28] C.E. Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27(3), 1948.
- [29] Sanjoy Dasgupta. Two faces of active learning. *Theoretical Computer Science*, 412:1756–1766, 2011.

In-the-wild chatbot corpus: from opinion analysis to interaction problem detection

Irina Maslowski^{1,2}, Delphine Lagarde¹, Chloé Clavel²

¹EDF Lab Paris-Saclay, Palaiseau, France

²Télécom ParisTech, Paris, France

irina.maslowski@edf.fr, delphine.lagarde@edf.fr, chloe.clavel@telecom-paristech.fr

Abstract

The past few years have seen growing interests in the development of online virtual assistants. In this paper, we present a system built on chatbot data corresponding to conversations between customers and a virtual assistant provided by a French energy supplier company. We aim at detecting in this data the expressions of user's opinions that are linked to interaction problems. The collected data contain a lot of "in-the-wild" features such as ungrammatical constructions and misspelling. The detection system relies on a hybrid approach mixing hand-crafted linguistic rules and unsupervised representation learning approaches. It takes advantage of the dialogue history and tackles the challenging issue of the opinion detection in "in-the-wild" conversational data. We show that the use of unsupervised representation learning approaches allows us to noticeably improve the performance (F-score = 74.3%) compared to the sole use of hand-crafted linguistic rules (F-score = 67.7%).

Index Terms: Chatbot dialog, Interaction problem, Opinion mining, Human-computer interaction, Written interactions

1. Introduction

Virtual agents and chatbots taking the role of on-line advisers have recently gained in popularity in the websites of the companies. The challenge remains the same as for human advisers: to improve customer satisfaction. In this paper, we propose to contribute to the detection of problematic interactions in a written chat with a virtual adviser with a system named DAPI¹. The present study takes place in the concrete application context of a French energy supplier EDF, using EDF chatbot corpus, gathering "in-the-wild" and rich spontaneous expressions.

We rely on the definition of [1] which defines a problematic situation as a reflection of the user's dissatisfaction with the conversational system answer. We call such kind of situations *interaction problems* (IP). We propose a hybrid approach to detect IP: hand-crafted linguistic rules based on finite state transduction over annotations and unsupervised representation learning to determine the word semantics.

Hitherto, the majority of studies that have tried to predict or to detect problems in human-machine interactions, were carried out for spoken dialog systems (SDS). Various types of cues are thus used to detect IP: prosodic cues [2, 3], speech-based system logs [4] – such as the low confidence of the outputs of the speech recognition, the direct feedback of the users about their satisfaction towards the interaction [5], semantic [3] and linguistic cues [6, 3, 7]. The studies carried out on chat-oriented

dialog systems are still less numerous, even though the use of chatbot systems by companies is increasing.

Linguistic features for the detection of IP are classically used as an input of supervised machine learning techniques. They range from basic linguistic features such as bag of words, n-grams [6, 3] and basic linguistic distances [3, 8] to Parts of Speech (POS) and statistic term frequency-inverse document frequency features [9]. The linguistic cues integrate various context of the dialogue history ranging from one to six user-agent turns [6, 3].

Some approaches integrate scores of semantic similarity between utterances in order to detect IP. For example, [3] use the inner product to calculate the score of semantic similarity between sentence vectors. The sentence vectors are build using neural network approaches. [1] use a knowledge-base for the same task for a general domain chat.

Other studies choose to also use opinion or affect cues in order to detect IP : [3], (in a SDS) and [1] (in a general domain chatbot in Chinese) use a lexicon-based approach for the detection of the affect or a sentiment in order to detect a problematic communication. [1] enhance a lexicon-based approach by regular expressions to model sentiment patterns.

In line with [1]'s approach on a general domain online chatbot in Chinese language, we propose a pioneering study that considers the user's opinions and emotions for the detection of IP in a domain-specific chatbot (customer relationship for electricity company) in French. Our main contributions are as follows: i) to take advantage of the entire dialogue history: the rules integrate linguistic cues contained in all preceding user's utterances; ii) to model the IP as the expressions of user spontaneous opinion or emotion towards the interaction; iii) to integrate web-chat and in-the-wild language specificities as linguistic cues for our rules; iv) to take advantage of word embeddings representations learned on our big unlabelled chatbot corpus in order to model semantic similarities.

In the following, we present our corpus of human-virtual agent written dialogues (Section 2). We introduce our system architecture (Section 3) and discuss our system evaluation results (Section 4). Finally we conclude and speak about our future work directions (Section 5).

2. Human - Virtual Agent Chat Corpus

The corpus (described in detail in [10]) contains all the interactions between users and the virtual agent (VA) collected from EDF company web-site from January to November 2014 totaling 1,813,934 dialogues. The role of the VA is to answer the users' questions about the EDF website navigation or the services and products of the company. A dialogue is composed at

¹"Détection Automatique de Problèmes d'Interaction" (automatic detection of interaction problems)

least of one adjacent pair (AP) that contains a user’s utterance and a VA utterance.

The dialogues contain “Failed” metadata given by the chatbot system but we are not using those as to remain as generic as possible in our corpus usage. The corpus of the EDF company has been anonymized and is private.

The main feature of the corpus is that it carries characteristics of French chat as described by [11]: emoticons (though rare), abbreviations, a phonetic spelling, “echo characters”, multiple punctuation and Anglicisms. The corpus contains typing and misspelling errors: 12% of words are tagged as <unknown> by TreeTagger [12]². This specificity of a gathered “in-the-wild” corpus renders the data difficult to process. However, such linguistic specific features are important because they carry information on the user opinion or emotions [10], e.g. “Parfait merci ;)” or “pfff”.

A subset of the big corpus was annotated in IP using GATE interface [14]. Following the strategy presented in [15] in order to simplify the annotation task, we define an annotation process guided by questions and information summaries. An IP taxonomy was thus proposed (see Figure 1) and integrated within a simplified decision tree. The taxonomy allows distinguishing *explicit interaction problems (EIP)* (an expression of the user negative emotion or opinion towards the interaction) and *implicit interaction problems (IIP)* (other linguistic clues: user’s repetitions, user’s contact request or “how does it work?” inquiries). The *EIP* are represented by a relation consisting of a triplet: source - opinion - target. The representation of a user’s opinion or a user’s emotion is based on the relation model from the appraisal theory of [16]. We have chosen this model according to the analysis of existing approaches exposed in [17]. We will use *OPEM* acronym which stands for *OPinion* and *EMotion* in order to gather all the opinion-related phenomena. Only the *OPEM* that have the interaction as a target were annotated. The interaction as the target, can be mentioned by the user *explicitly* (e.g. “tu es virtuelle, tu ne peux pas m’aider” [you are virtual, you can not help me]) or *implicitly* (e.g. Agent: “Veuillez m’excuser, je n’ai pas compris ce que vous venez de dire.” [I beg your pardon, I have not got what you said.] User: “pfff”).

We have held two manual annotation campaigns to create: i) the “DevCorpus”, for the development of the current system; ii) the “T-Corpus” for the evaluation of the current system. We choose to call upon a specialist in semiology – familiar with the analysis of the corpora of the EDF company – for the annotation. Even though this choice does not allow us to obtain a quantitative measure of the annotation reliability, it corresponds to a good compromise between reliability and annotation cost. The corpora statistics are presented in Table 1 and shows that both corpora contain a similar proportion of IP. In both cor-

Table 1: Statistics of manually annotated corpora.

Main Statistics	DevCorpus	T-Corpus
Dialogues	3,000	3,000
Adjacent Pairs (AP)	8,576	8,630
Dialogues with at least one IP	741	845
AP with IP	15%	17%
Problematic AP in a problematic dialogue (mean)	2	1.5

pora, the ratio of dialogues with at least one IP is relatively low:

²It’s worth noting that, a similar assessment was done in the customer - human agent chat corpus presented in [13]

25% and 28% respectively. Only 15,5% of all user’s utterances contain an IP. Only 11% of IP in the development corpus and 6% of IP in the reference corpus are explicit. IP are annotated at the utterance level. Despite the fact that our system does not need to detect the fine classes of IP, they are a good support for the linguistic analysis of the system annotation results.

3. Hybrid Approach

The DAPI system aims to detect utterances containing IP in written conversations between a user and a VA by analyzing in real-time the user’s utterances. The overall architecture of our system based on the GATE framework [14] is presented in Fig. 2. It relies on a hybrid approach combining hand-crafted linguistic rules and unsupervised representation learning to determine the word semantics. After a preprocessing step, the linguistic rules are used in order to extract expressions of user’s negative opinion towards the interaction and other linguistic cues of IP. They rely on the GATE JAPE (Java Annotation Patterns Engine) that provides finite state transduction over annotations [18] based on regular expressions. The linguistic rules take advantage of dialogue history and integrate Internet French chat features. The learned word semantics is used to improve the detection of user’s repetitions and problem reformulations that are featuring IP (Section 3.3.3).

The preprocessing is composed of the data anonymization, the elimination of hyper-links, the text tokenization, and the POS and chunks annotation by TreeTagger [12]. According to the used version of DAPI (see Section 4), it is possible to include a spell checking step using PyEnchant³ library of Python. In order to avoid cleaning valuable clues of IP, before applying the spell checker, we verify that words are not in the dictionaries of Internet slang⁴ (e.g. *lol*), emotions (lists of emotions and insults from LIWC for French [19]) and business terms (lexicon of business terms grouped into concepts and consisting of 400 entries provided by the EDF company and constructed on the basis of different business corpora including our *DevCorpus*). This preliminary check is carried out using “difflib” library⁵ of Python, which is an extension of the Ratcliff and Obershelp algorithm [20]. We describe the following processing steps according to the type of context which is taken into consideration.

3.1. At the level of the user’s utterance

The annotation rules are designed to detect relations between a source, an *OPEM*, and a target. They combine lexical clues based on Internet slang, LIWC and several small hand-made lexicons of: basic emoticons, potential sources of opinion (first personal pronoun variants, as we focus on the user’s opinion), opinion verbs and expressions (20 entries), expressions of different concepts (e.g. gratitude, greetings, demand) (30 entries). *Relations* are modelled by seven relation patterns [*Source OPEM Target*] depending on the presence of a target, a source and an *OPEM* in the same sentence. First, each element (source, *OPEM* or target) that can potentially be a part of a relation is detected. Then, if matching a relation pattern, the user’s utterance is annotated as containing an IP. The *potential OPEM* is modelled by thirteen rules of three and four levels of complexity. They include the negation processing which is

³<http://pythonhosted.org/pyenchant/>

⁴http://fr.wiktionary.org/wiki/Annexe:Liste_de_termes_d%28%99argot.Internet

⁵<https://docs.python.org/3/library/difflib.html#difflib.SequenceMatcher>

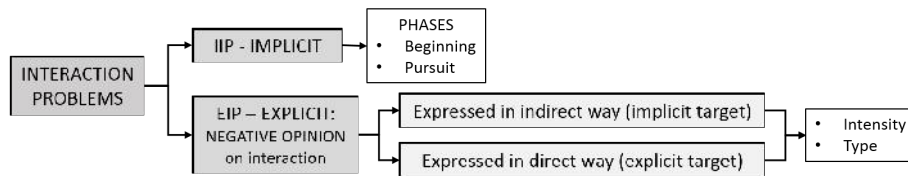


Figure 1: Taxonomy of Interaction Problems

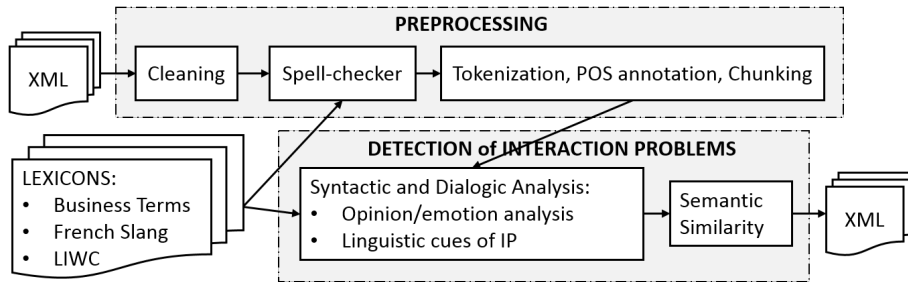


Figure 2: DAPI System Scheme

based on [15] chunk approach. Typographical clues such as the multiple punctuation and the smileys, and expressions of dissatisfaction (ex. "Laisse moi tranquille") [leave me alone] are used to detect relations with an implicit target.

The *explicit interaction target* is modeled by eleven sub-concepts linked to: the mutual comprehension during the interaction, the efficiency of the VA's work or the adequacy of the VA's answer. The sub-concept "réponse", for example, contains the following list of synonyms: "réponse, résultat, réaction, solution, explication, réplique, retour"⁶.

3.2. Using the context of agent's utterance

Spontaneous Contact Requests. We define a spontaneous user contact request as a set of lemmas of the following groups of words: 1) "contacter" [to contact], "téléphoner" [to phone], "téléphone" [a phone]; 2) conseiller [advisor], EDF [company name], where at least one word of each group should be present in the user's utterance, otherwise a string "appel" [a call] should be present. The rules based on the detection of the *user contact requests* are the following: 1) contact requests (like in [9]) and inquiries on the chatbot functioning are treated as problematic if they are not in the first user's utterance; 2) if the Agent's contact suggestion comes before the user's request, the user's utterance is not considered as problematic.

Expressions of dissatisfaction towards agent's answer. We use expressions of user dissatisfaction to detect IP according to the following rule:

IF the agent expresses its inability to help the user **AND** the user's utterance contains echo characters and/or onomatopoeia ("pfff") or Internet slang expressions as means to close the dialogue (ex. English word "bye"), **THEN** the user's utterance is labeled <interaction problem>.

⁶answer, result, reaction, solution, explanation, reply, feedback

3.3. Modelling over several user's turns : user repetition and reformulation

We use the following approaches to detect the *user repetitions or reformulations*: linguistic distances, the detection of the repetition of business concepts or terms and semantic similarity.

3.3.1. Detecting user's repetitions by using linguistic distances.

We calculate linguistic distances between the current user's utterance and all the preceding user's utterances by applying the Jaccard distance improved by [21] and the Levenshtein distance [22]. The Jaccard distance measures the common part of the vocabularies for two user's utterances. The Levenshtein distance measures the differences between the character sequences in order to manage typing errors. It is worth noting that, though these distances are the most commonly used metrics for user repetition detection [3, 8], the Jaccard distance we use allows a better performance in long phrases. The final distance for an utterance is the minimal distance between the current utterance and each previous utterance. The rules are based on the comparison of the distances to thresholds (≤ 4 for Levenshtein and ≤ 0.85 for Jaccard) that have been optimized on the *DevCorpus* in order to detect IP. The Levenshtein distance is complementary to the Jaccard distance as it detects user repetitions containing misspelled words.

3.3.2. Detecting user's repetitions by retrieving business concepts.

Two different rules are based on business concepts⁷ or terms. The retrieving of business concepts is carried out with the lexicon of business concepts and with patterns dedicated to retrieve multi-word expressions of business terms such as 'customer space'. The first rule is based on multiple punctuation and constant presence of business terms in user's utterances. It also takes into account the dialogue history. The presence of

⁷a business concept is a synset of business terms

business terms in the previous user’s utterances disambiguate multiple punctuation concerning the interaction from that concerning products or services. The rule is as follows:

IF a user’s utterance contains a business term followed by a multiple punctuation (ex. !!, ??),

AND a business term was already contained in the previous user’s utterances,

THEN the user’s utterance is labeled *<interaction problem>*.

In the second rule, we are looking for the presence of the same business concept in the previous user’s utterances. If a business concept on the current user’s utterance was already mentioned in one of the previous user’s utterances, the current utterance is annotated as containing an IP. This is the case of the third user’s utterance (U3) in Example 1. In this example, the business terms “carte bleue” [credit card] and “carte bancaire” [bank card] belong to the same concept “Carte Bancaire”.

Example 1 *Detection of an interaction problem based on the repetition of business concepts*

User [U1]: je régler par carte bleue je ne le trouve plus⁸

Agent: EDF met plusieurs [URL] modes de paiement à votre disposition. (...) ⁹

User [U2]: [URL]

Agent: Je viens de vous rediriger vers la page demandée.¹⁰

User [U3]: je veut régler par carte bancaire¹¹

Agent: (...)

3.3.3. Detecting user’s reformulation by using semantic similarity measures and word embeddings.

We use the *semantic similarity* in order to detect more user reformulations (DAPI-3 and DAPI-4). The computation of the semantic similarity between two user’s utterances is based on the representation of words in a vector space. We have allocated the larger part of our corpus (named the “ChatBot Embedded” corpus) for training the word/utterance embeddings models. The “ChatBot Embedded” raw corpus contains 2,112,860 user’s utterances (11,087,419 words). The corpus went through the following transformations: the separation of articles from words, the letter case homogenization, deletion of numbers, nonce words and stop words. The final number of words is 8,888,049. We have chosen the **word2vec model** [23] to transform the words of our corpus into vectors. We use the standard word2vec library for python ¹² with the following training parameters: size = 100, cbow = 0, verbose = False, iter = 5. The word vectors are summed to obtain the vector of the utterance. The cosine distance between the vectors of two utterances with a threshold of 0.85 (optimized on the *DevCorpus*) determines whether two utterances are similar. If so, the second user’s utterance is annotated as an IP. The following section presents the results of the evaluation of DAPI system.

4. Evaluation and Discussion

To our knowledge, there is no other system that can serve us as a baseline for the detection of IP in a French written chat with a virtual adviser. Hereafter, we describe the steps we follow to

⁸I pay with a credit card I can not find it any more

⁹EDF puts several payment methods at your disposal(...)

¹⁰I have just redirected you to the requested page.

¹¹I want to pay by bank card

¹²<https://pypi.python.org/pypi/word2vec>

establish a baseline.

As the major clues of IP are the users’ repetitions/reformulations and the users’ opinions/emotions on the interaction, we apply two methods separately: the classique Jaccard distance [24] to detect repetitions and the Naïve Bayes classification which is commonly used for sentiment analysis [25]. The 0.15 threshold for the Jaccard distance is determined on the basis of the best trade-off between recall and precision on the DevCorpus. The 10-fold cross-validation is applied to the Naïve Bayes classification on the T-Corpus. Considering the low results of the both approaches (see Table 2), we choose the basic configuration of our system (DAPI-1) as a baseline. In order to evaluate the contributions of the spell checker and of the word embeddings representation, we compare four versions of our system: **DAPI-1** system with only linguistic rules; **DAPI-2** integrating the spell-checker, **DAPI-3** integrating the computation of the score of semantic similarity but not the spell-checker and **DAPI-4** combining both the spell-checker and the computation of the score of semantic similarity. The systems are evaluated on the *T-corpus* by computing Precision, Recall and F-score [26] for the detection at the utterance level. The IP detection scores are shown in Table 2.

Table 2: Results in % for the detection of IP in the T-corpus.

System	Precision	Recall	F-score	Accuracy
Naïve Bayes	25.9	14.6	18.6	90.1
Jaccard	55.5	38.6	45.6	79.2
DAPI-1	72.4	63.6	67.7	90.1
DAPI-2	72.0	65.4	68.5	90.2
DAPI-3	72.0	77.0	74.4	91.4
DAPI-4	71.1	77.8	74.3	91.3

The use of word embeddings (DAPI-3 and DAPI-4) provides a noticeable improvement of the system performance. DAPI-3 obtains the best F-score. However, DAPI-4 allows a higher recall, which is important in our context (it is important to detect the maximum of existing IP). It is worth noting that we have also experimented to train word2vec on the corpus processed with the spell-checker but the results of the calculation of the score of semantic similarity dropped. The utterances detected as similar using the semantic similarity can be characterized as: user repetitions with a highly misspelled context (rules using linguistic distances detect simpler cases of repetitions); reformulations containing words with the same word-root (e.g. the word “payer”¹³ in the user’s utterance “ je ne trouve pas ma facture pour la payer en ligne”¹⁴ and “paiement”¹⁵ in “je ne veux pas le télépaiement je veux le paiement par carte bleue”¹⁶, similarity score 0.877) and reformulations containing at least one expression in common (e.g. the expression “je souhaite”¹⁷ in the following user’s utterances “bonjour, je souhaite voir le récapitulatif de mes prélèvements”¹⁸/ “je souhaite savoir combien je suis relevé par mois”¹⁹, similarity score 0.869).

The linguistic rules based on the tracking of the repetition of business concepts detect reformulations as well. These are reformulations containing business terms with a common root

¹³to pay

¹⁴I can not find my bill to pay it online

¹⁵payment

¹⁶I don’t want the telebanking, I want the credit card payment

¹⁷I would like

¹⁸Goodday, I would like to see the summary of my withdrawals

¹⁹I would like to know how much is my bank withdrawal per month

(e.g. "pourquoi paie t on d'avance l'abonnement"²⁰ / "paiement abonnement d'avance"²¹, where the words with the common root are "payer"²² and "paiement"²³). The joint use of both approaches to the detection of the user reformulation as a mark of IP contributes to the robustness of the system to cope with the challenges of the "in-the-wild" corpus. However, both our approaches to the user reformulation detection (business concept repetition and semantic similarity) still create a lot of false positives (e.g. in the cases when the user clarifies his/her previous utterance or carries on with the same topic) that are difficult to handle.

The joint model of the specificities of the chat language and the dialogue history contributes, for example, to detecting a user irritation towards the interaction with the chatbot (the rule combining multiple punctuation and business terms). In particular, multiple punctuation clues take an important role in the detection (78,5% of correct matches done with the rules exploiting the specificities of the chat language, are done considering the multiple punctuation clue).

5. Conclusion and Future Work

In this paper, we present the DAPI system based on a hybrid approach for the detection of interaction problems in dialogues between a human and a virtual adviser. The system focuses on the expressions of user spontaneous opinion or emotion that feature interaction problems. DAPI combines an approach based on hand-crafted rules for finite state transduction over annotations and semantic similarity measures computed on word embeddings learnt from a big unsupervised corpus. We have tried different configurations of DAPI system. The best performance from the application point of view (higher recall) is obtained by the version of the system combining the semantic similarity and the linguistic rules with the spell-checker. The semantic similarity based on word embeddings detects complex user reformulations and misspelled repetitions. In future work, we would like to investigate other types of hybridization between unsupervised representation learning and rule-based approaches, allowing to take advantage of our *big* unlabeled chatbot corpus.

6. References

- [1] Y. Xiang, Y. Zhang, X. Zhou, X. Wang, and Y. Qin, "Problematic situation analysis and automatic recognition for chinese online conversational system," *Proc. CLP*, pp. 43–51, 2014.
- [2] A. Batliner, C. Hacker, S. Steidl, E. Nöth, and J. Haas, "User states, user strategies, and system performance: how to match the one with the other," in *ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems*, 2003.
- [3] S. Georgiladakis, G. Athanasopoulou, R. Meena, J. Lopes, A. Chorianopoulou, E. Palogiannidi, E. Iosif, G. Skantze, and A. Potamianos, "Root cause analysis of miscommunication hotspots in spoken dialogue systems," in *INTERSPEECH*, 2016, pp. 1156–1160.
- [4] M. A. Walker, I. Langkilde-Geary, H. Wright Hastie, J. Wright, and A. Gorin, "Automatically training a problematic dialogue predictor for a spoken dialogue system," *Journal of Artificial Intelligence Research*, vol. 16, pp. 293–319, 2002.
- [5] H. W. Hastie, R. Prasad, and M. Walker, "What's the trouble: automatically identifying problematic dialogues in darpa communicator dialogue systems," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2002, pp. 384–391.
- [6] A. van den Bosch, E. Krahmer, and M. Swerts, "Detecting problematic turns in human-machine interactions: Rule-induction versus memory-based learning approaches," in *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2001, pp. 82–89.
- [7] F. Cailliau and A. Cavet, "Mining automatic speech transcripts for the retrieval of problematic calls," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2013, pp. 83–95.
- [8] J. Liscombe, G. Riccardi, and D. Hakkani-Tür, "Using context to improve emotion detection in spoken dialog systems," in *Ninth European Conference on Speech Communication and Technology*, 2005.
- [9] I. Beaver and C. Freeman, "Detection of user escalation in human-computer interactions," in *INTERSPEECH*, 2016, pp. 2075–2079.
- [10] I. Maslowski, "Quelles sont les caractéristiques des interactions problématiques entre des utilisateurs et un conseiller virtuel?" *PARIS Inalco du 4 au 8 juillet 2016*, p. 94, 2016.
- [11] F. Achille, "Constitution d'un corpus de français tchaté," in *RECITAL*, Dourdan, France, 2005.
- [12] H. Schmid, "Probabilistic part-of-speech tagging using decision trees," in *Proceedings of International Conference on New Methods in Language Processing*. Manchester, UK: Association for Computational Linguistics, 1994.
- [13] A. Nasr, G. Damnati, A. Guerraz, and F. Bechet, "Syntactic parsing of chat language in contact center conversation corpus," in *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2016, p. 175.
- [14] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, N. Aswani, I. Roberts, G. Gorrell, A. Funk, A. Roberts, D. Damljanovic, T. Heitz, M. A. Greenwood, H. Saggion, J. Petrak, Y. Li, and W. Peters, *Text Processing with GATE (Version 6)*, 2011. [Online]. Available: <http://tinyurl.com/gatebook>
- [15] C. Langlet and C. Clavel, "Improving social relationships in face-to-face human-agent interactions: when the agent wants to know user's likes and dislikes," in *ACL 2015*, 2015.
- [16] J. R. Martin and P. R. White, "The language of evaluation," *Appraisal in English*. Basingstoke & New York: Palgrave Macmillan, 2005.
- [17] C. Clavel and Z. Callejas, "Sentiment analysis: from opinion mining to human-agent interaction," *IEEE Transactions on affective computing*, vol. 7, no. 1, pp. 74–93, 2016.
- [18] H. Cunningham, D. Maynard, and V. Tablan, "Jape-a java annotation patterns engine, department of computer science, university of sheffield," 2000.
- [19] A. Piolat, R. Booth, C. Chung, M. Davids, and J. Pennebaker, "The french dictionary for liwc: Modalities of construction and examples of use— la version française du dictionnaire pour le liwc: modalités de construction et exemples d'utilisation," 2011.
- [20] J. W. Ratcliff and D. E. Metzener, "Pattern-matching-the gestalt approach," *Dr Dobbs Journal*, vol. 13, no. 7, p. 46, 1988.
- [21] E. Brunet, "Peut-on mesurer la distance entre deux textes?" *Corpus*, no. 2, 2003.
- [22] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8, 1966, pp. 707–710.
- [23] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
- [24] P. Jaccard, "Distribution de la flore alpine dans le bassin des drouces et dans quelques regions voisines." vol. 37(140), pp. 241–272, 1901.

²⁰ why do we pay in advance the subscription

²¹ the in advance payment of subscription

²² to pay

²³ the payment

- [25] F. Saad, "Baseline evaluation: an empirical study of the performance of machine learning algorithms in short snippet sentiment analysis," in *Proceedings of the 14th International Conference on Knowledge Technologies and Data-driven Business*. ACM, 2014, p. 6.
- [26] C. Van Rijsbergen, "Information retrieval. dept. of computer science, university of glasgow," *URL: citeseer.ist.psu.edu/vanrijsbergen79information.html*, vol. 14, 1979.

SMRE: Semi-supervised Medical Relation Extraction

Sana Trigui¹, Ines Boujelben¹, Salma Jamoussi¹

¹Miracl, University of Sfax, Tunisia

truquisana0@gmail.com, Boujelben_ines@yahoo.fr,
jamoussi@gmail.com

Abstract

Relation extraction between named entities presents a useful task for several applications of Natural Languages Processing (NLP) such as a question-answering system [1], automatic summarization [2] and ontology construction [3]. This paper reports our semi-supervised system SMRE to extract relations between medical named entities. Given that semi-supervised learning approach relies on a few labeled data, it requires less human effort and time consuming. Our system is applied to a well-known medical corpus [4] that is built from a Medline medical bibliographic database. The evaluation of our system showed promising results of 83.20% in terms of accuracy without knowing the medical named entities and 100% otherwise.

Index Terms: Named entity, Relation extraction, Semi-supervised learning, Medical domain.

1. Introduction

The task of medical relation extraction serves to discover useful relationships between two medical named entities (NEs). A medical NE is a NE¹ related to the medical field such as diseases (Asthma, Diabetes mellitus), treatments (Antibiotic, analgesic), symptoms (Vomiting, dry mouth), medications (Metformin, Omeprazole) and examinations (Computed tomography, pulmonary x-ray). The medical field is characterized by the complexity and the instability of its vocabulary. This problem, in a sensitive field such as medicine, forces us to develop base of knowledge (medical NE and relations between medical NEs). This base helps us to better understand the text, discover new information and improve the quality of patient care. Given the usefulness of the task of relation extraction between NEs, many researchers have been interested to this task where each of them has tried to find the most optimal solution for solving this problem. Indeed, some works have used linguistic methods (rules, grammars) [5] to solve this problem. Others have chosen to work with automatic classification techniques based on supervised learning [6]. However, these supervised methods are based on large annotated corpora. The annotation of these corpora requires a lot of effort, expertise and time consumption. These limitations have encouraged the introduction of the semi-supervised learning paradigm as a reliable classification tool. Therefore, we have resorted to working with semi-supervised learning methods. The paper is organized as the following: first, we will survey prior semi-supervised

studies on relation extraction between medical NEs. The third section is going to illustrate the architecture of our semi-supervised system, in which we detail its main steps. Afterward, we will present the different experiments from which we are going to discuss the reported results. Finally, some conclusions are drawn in order to structure future works.

2. Semi-supervised relation extraction

Semi-supervised learning use both labeled and unlabeled data [7]. It falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data). From the labeled data, we can predict labels from the unlabeled data. Several kinds of methods have been developed to carry out the task of semi-supervised learning, among which we can mention Self-training, Co-training, Tri-training. Self-training [8] consists in training a classifier on labeled data (DL). This classifier is then used to label the unlabeled data (DU). Labeled data with a high degree of confidence are then added to the training data (DL). The classifier is re-trained on the DL data and the procedure is repeated until the unlabeled data disappears. Self-training has been applied to several NLP tasks. [9] are based on self-training for the morpho-syntactic categorization of sentences using a Markov model classifier. They obtained a rate of accuracy² of 85% when applied to the North American News Corpus NANC. The authors in [10] proposed a method for the classification of sentiments. They showed that the classification rate obtained by their method (84%) is better than the rate obtained by a supervised method (73%). Self-training is a wrapper algorithm. However, this method encounters the problem of the lack of external information; the classifier is supposed to provide additional information from the non-annotated examples based on its own output, including his confidence score. In order to avoid these problems of divergence, the idea here is to use several different classifiers to improve the classification task. This strategy was recognized as a Co-training method. Co-training [11] can be perceived as an extension of the Self-training method. Instead of using a single classifier, Co-training uses two classifiers, and the set of attributes is divided into two independent sets. Its main idea is to train a first classifier with a first set of attributes in order to label the unlabeled data (DU). Then, the labeled data with a high degree of confidence are added to the training data (DL) to learn the second classifier. Subsequently, the same procedure is repeated

¹NE includes the proper names of persons (e.g. Louis), locations (e.g. New York), and organizations (UNESCO).

²Calculated by the percentage of correct classified data

with the second classifier to learn the first classifier. The procedure is repeated until the unlabeled data disappears. An extension of Co-training is recognized as a Tri-training method [12]. This algorithm uses three classifiers. These three classifiers are first trained in labeled data and then used to label unlabeled data. The labeling phase is done by combining the classifiers using the majority vote. Authors in [13] have relied on Co-training for lexical disambiguation of medical words. They have defined two sets of attributes where the first set contains the context of the words around the ambiguous word, and the second set is used to find the different meanings related to that word using the semantic network UMLS³. They obtained an accuracy of 85%. The system proposed by [14] is based on a Co-training method for the extraction of two types of relations, one between a disease and symptom, and the other between a symptom and a treatment from Medline. For this purpose, the authors constructed two corpora: the first contains disease-symptom relations and the second contains symptom-treatment relations. They obtained an f-measure of 80% for the second type of relations. To use the Co-training method, the authors [11] defined two conditions: one must have two different views (set of attributes) of the data to be classified, and these two views must be compatible and independent. A last hypothesis for the proper functioning of Co-training is that each view must be sufficiently consistent to learn a classifier independently. However, the use of this method with the constraints cited above is sometimes impossible because in some cases we cannot have two independent views.

3. Proposed method

Since we seek to rely on a few labeled data, we propose our semi-supervised system "SMRE" for relation extraction between medical NEs. Its general architecture is composed of three general modules of processing, as illustrated in Figure 1: (1) building training data, (2) selection of relevant subsets of attributes, and (3) Semi-supervised training process.

3.1. Building training data

In this module, we construct our training corpus. We used the same corpus of [4] which was also used by [15], [16] and [17]. This corpus which is called the Berkeley⁴ corpus, is constructed from the titles and abstracts of Medline. It has been annotated with seven types of semantic relationships between the medical entities disease (DIS) and treatment (TREAT). As a first step, we apply a morphological and syntactical analysis using the Stanford⁵ tool, which allows segmentation, lemmatization, and morph syntactic categorization. This corpus is used subsequently to extract the different features (attributes) of training in order to build our training base. The different features used in our work are listed in Table 1.

³<https://www.nlm.nih.gov/research/umls/>

⁴biotext.berkeley.edu

⁵<http://nlp.stanford.edu/software/tagger.shtml>

3.2. Selection of relevant subsets of attributes

The process of selecting attributes consists on reducing the number of used attributes by choosing from a large set of attributes the more interesting subset. This reduction can improve the performance of a relation extraction system. We distinguish three approaches to attribute selection: the wrapper approach [18] which uses the classification algorithm to evaluate the subset of generated attributes, the filter-type approach [19] that is completely independent of the used algorithm and uses an evaluation function based on statistical, entropy, consistency and distance to evaluate the subset of attributes. Finally, the embedded approach that combines the two previous approaches [20]. We can notice that the filtering methods are the best in terms of execution time and generality⁶. This is probably the main reason why these methods are the most popular. In order to have the best compromise between the time constraints and the quality of the results, we are interested to filtering methods. However, it is important to note that these filters require choosing the appropriate number of attributes to be selected. For this reason, we try to find an efficient solution to find the right number of attributes to select. The application of a filtering method generates a list of all the attributes sorted in decreasing order according to an evaluation function. In our case, we will use the following gain functions: dependence measure [21], information gain measure and Principal Component Analysis (PCA) [22].

The idea of selecting the most relevant d attributes from this list is to calculate the accuracy of the first attribute that has the highest score using a training corpus and a test corpus. Then, with every iteration, we add the attribute that succeeds it in the list until to arrive to evaluate all attributes. The selection of d attributes is done by selecting a subset of attributes, which presents the maximum accuracy. In order to have n subsets of relevant attributes, we follow the same approach while using n evaluation criteria (information gain, gain ratio, etc.).

3.3. Semi-supervised training process

The third module presents the core of our method. It takes place in five phases, which are: training phase, selection phase of K unlabeled data, labeling phase, combining phase of the classifiers and finally the phase of updating the corpus.

- **First phase : training phase**

This phase consists in inducing n classifiers (same classifier with n different subsets of relevant attributes, the n subsets were obtained in the second module) on the training kernel which contains only a small number of labeled data, evaluate these n classifiers using a test corpus. The purpose of using a test corpus is to calculate a confidence score for each classifier. This score is the accuracy measure calculated using the following formula:

⁶https://en.wikipedia.org/wiki/Feature_selection

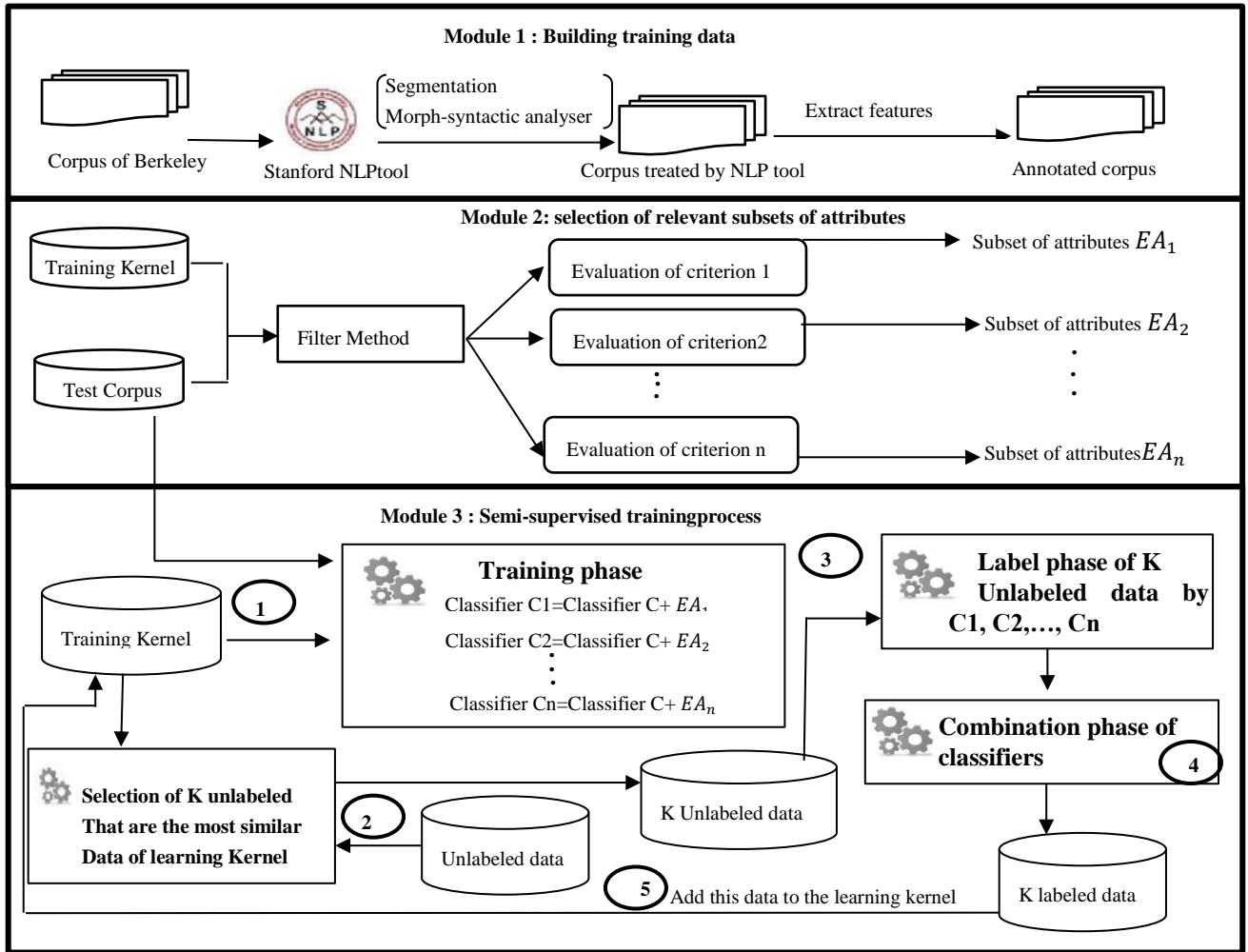


Figure 1: The architecture of our semi-supervised method

Type	Feature	Description
Lexical	catM1E1	The part of speech of the first word before NE1
	catM2E1	The part of speech of the second word before NE1
	catM1E2	The part of speech of the first word before NE2
	catM2E2	The part of speech of the second word before NE2
	catM1E1E2	The part of speech of the first word between the two NEs
	catM2E1E2	The part of speech of the second word between the two NEs
Semantic	typeEN1	The first NE tag (label)
	typeEN2	The second NE tag
	PaireE1E2	The appearance order of NEs
Syntactic	Type-phrase	Type of sentence(nominal, verbal)
Numeric	NbrM	Number of words in the sentence
	posE1	Position of the first NE
	posE2	Position of the second NE
	nbrMavE1	Number of words before the first NE
	nbrMapE1	Number of words after the first NE
	nbrMavE2	Number of words before the second NE
	nbrMapE2	Number of words after the second NE
	nbrME1E2	Number of words between the two NEs

Table 1. The used features.

$$\text{Score}(C) = 1 - \text{error rate} \quad (1)$$

$$\text{error rate} = \frac{\text{number of misclassified data}}{\text{number total of data}} \quad (2)$$

- **Second phase : selecting K unlabeled data**

In this phase, we select the K unlabeled most similar data to the training kernel. The similarity calculation is done by calculating the distance between the unlabeled data and the labeled data. If we have m labeled data and m1 unlabeled data, we will perform m * m1 distance calculations. This requires a lot of time, which will decrease the execution time of our method. To solve this problem, we present the following solution:

- 1) We start by decomposing the training kernel into M groups with M being the number of classes.
- 2) For each group, we calculate its barycenter (b) of coordinates x_1, x_2, \dots, x_l using the following formula:

$$x_i = \frac{\sum_{j=1}^l x_{ij}}{l} \quad (3)$$

Where l is the number of attributes

- 3) We calculate the distance between the M groups and the unlabeled data using the Euclidean distance:

$$D(b, \text{data}) = \sum_{i=1}^l |x_i - y_i| \quad (4)$$

- 4) For each data, it is allocated the smallest distance among all the distances of M groups.
- 5) The data is sorted increasingly according to distance
- 6) We select the K data that have the smallest distances where n is the number of attributes, x and y are coordinates of barycenter and the data respectively and $i \in [1 \dots \text{number of attributes}]$.

- **Third phase : labeling phase**

The K similar data that is obtained from the second phase will be labeled by the n classifiers, where each classifier assigns a probability for a data, by knowing each class. This probability is of the form $P(\text{data}/\text{class}_h, C)$:

$$\sum_{h=1}^{\text{number of class}} P(\text{data}/\text{class}_h, C) = 1 \quad (5)$$

Where $h \in [1, \text{number of class}]$.

- **Fourth phase: combination of classifiers**

Our method is based on weighted vote. This is a vote based on weights associated with basic classifiers. All classifiers must label each data. Since each classifier assigns for this data a probability of belonging to each class, this data takes a class h if the probability of belonging to this class is maximum, with respect of the other classes. Then, using this formula (formula 6), we can easily know the class attributed by a classifier c to the instance:

$$P'(\text{data}/\text{class}_h, C_j) = \arg\text{-max} P(\text{data}/\text{class}_h, C_j) \quad (6)$$

For each classifier, we multiply its maximum probability of belonging to a class h for a data by its confidence score (according to formula 1) to obtain a calculated weight of a data for its belonging to a class:

$$\text{Weight}(\text{data}/\text{class}_h, C_j) = \text{Score}(C_j) * \arg\text{-max} P'(\text{data}/\text{class}_h, C_j) \quad (7)$$

For each class, we sum up the weights:

$$S(\text{class}_h) = \sum \text{Weight}(\text{data}/\text{class}_h, C_j) \quad (8)$$

The final labeling of an instance is done by assigning the class whose $S'(\text{class}_h)$ is maximal.

$$S'(\text{class}_h) = \text{Arg-max}(S(\text{class}_h)) \quad (9)$$

- **Fifth phase: update of all corpus**

As a last step of our method, we add the K labeled data by the n classifiers to the training kernel and remove them from the corpus of unlabeled data.

The third module (Semi supervised training process) is repeated until the unlabeled data disappears. After having presented the architecture of our system, we present the different results obtained when applying our system on our medical corpus.

4. Experimentations and results

The Berkeley corpus has been used by several researchers. Some research studies treated only three relations (Cure, Prevent, SideEffect), others are based on seven relations. For this, we propose to treat these two cases. The statistical study of our corpus reveals the following characteristics⁷ as represented in Table 2.

Relations	Number of instances
Cure	783
Only DIS	600
Only TREAT	163
Prevent	58
Vague	45
SideEffect	41
No Cure	4

Table 2. *The statistical characteristics of our medical base*

Our corpus is divided into three sub-corpora: the first sub-corpus is a training kernel, the second is a test sub-corpus, and the third is a sub-corpus of unlabeled data. Table 3 shows the amount of instances allocated for each sub-corpus. We choose to assign 347 instances for the test; it presents the same number of instances used by [4].

Number of instances in the training kernel	229
Number of instances in the test corpus	347
Number of unlabeled data	1080

Table 3. *The distribution of data for each sub-corpus*

We treat the relation extraction task in two cases: the first where the two NEs are not recognized, and the second when the NEs are recognized. Indeed, the recognition of the types of medical NE facilitates certainly the identification of the type of interacting relationship between these NEs, as illustrated in example 1 and 2:

⁷The number of sentences available for downloads is not the same as the ones from the original data set published in [Rosario and Hearst, 2004]

<DIS_PREV> Measles </DIS_PREV><TREAT_PREV> vaccination </TREAT_PREV> and inflammatory bowel disease (1).

Over half thought <DISONLY> HIV</DISONLY> transmission occurred most times or always (2).

As illustrated in example 1, the recognition of the two types of NEs (DIS-PREV and TREAT_PREV) makes it easy to identify the "Prevent" relationship without the need to apply a training algorithm. The same for example 2, when knowing that the NE is annotated by DISONLY, it will be easy to know that the relation is disonly.

We evaluate our method using many classifiers and the best result is obtained by the classifier PART. We use this classifier from WEKA project which is implemented in Java [23]. The parameters used in our system are $n=3$ (number of relevant subsets of attributes), $K=10\%$ (number of unlabeled data selected in each iteration).

Afterwards, we evaluate three semi-supervised algorithms Self-training, Co-training and YATSI [24] on the used medical corpus. YATSI contains two steps. It can use any training algorithm (Wrapper algorithm) with the nearest neighbor algorithm. In the first step, an algorithm is trained on the labeled data to construct a training model. Then, this model is used to label unlabeled data. In the second step, the nearest neighbor algorithm is applied using the initial labeled data and the newly labeled data. In addition to the initial classification algorithm, the nearest neighbor algorithm is used to adjust or correct labels assigned to unlabeled data. For the YATSI implementation, we use collective classification package available from MARSDEN project⁸ Programs which are written and tested in Java programming language in Eclipse environment. While for Self-training and Co-training, we implement them. To the best of our knowledge, there is no study that has adopted the semi-supervised method using this corpus. Thus, our proposed method represents the first semi-supervised work using the corpus of [4].

As reported in Table 6, we note that our method is more efficient than the three semi-supervised methods considered. Indeed, with the base of 7 relations without recognizing the NEs, we get an accuracy of 83.20%, while with YATSI we get 54%. When applying Self-training method, we get 31.38%. The application of Co-training on the same corpus obtains 57.06% of accuracy. By knowing the types of NEs, the value of accuracy is high for the four algorithms. Here, it is not logical to use NE types as attributes because we have seen previously that the recognition of NEs types facilitates the task of relation extraction and we do not even need to apply a training algorithm.

The originality of our semi-supervised method can be assumed in the contributions acquired, of each phase. For the selection phase of the relevant subset of attributes, we

used filtering methods to evaluate the attributes of the data rather than their interactions with a particular classifier. The evaluation of these methods shows more generality (they do not depend on the classifier). Each subset of attributes is the best according to a criteria to be optimized (information gain, ACP,...). The subsets obtained according to the various evaluation criterions may be overlapping or independent. At this level, there are no assumptions or conditions on the relationship between the subsets of attributes found, unlike other works like the work of [11] where they defined a condition that concerns the independence between the subsets of attributes. For the labeling phase of the unlabeled data, we intend to improve it somewhat by selecting in each iteration the most similar data to the labeled data. In addition, our method combines several classifiers. The combination of these classifiers is based on the weighted sum vote. As a conclusion, we can mention that the combination of several classifiers improves the classification results better than the Self-training method that uses only one classifier.

5. Conclusion

In this paper, we described our semi-supervised system SMRE to extract relations between medical NEs. The obtained results are encouraging and promising since the used techniques allowed us to reach a classification rate equal to 100% when we recognize the NEs, and 83.20% without recognizing this information. Using our semi-supervised method, we were able to resolve the labeling problem using semi-labeled corpus. Our proposed method treats only binary relations (between two NEs). As a future works, we intend to treat relations holding between more than two NEs. In addition, we are planning to evaluate our system with other NEs types and different corpora languages and domains.

⁸http://www.cs.waikato.ac.nz/fracpete/projects/collective_classification/

Medical corpus	Self-training	Co-training	YATSI	SMRE
Base with 3 relations (NEs are un known)	57.22%	86.67%	82.87%	90%
Base with 3 relations (NEs are known)	70%	100%	92.72%	100%
Base with 7 relations (NEs are unknown)	31.38%	57.06%	54%	83.20%
Base with 7 relations (NEs are known)	65.68%	93.95%	100%	100%

Table 6. Comparison of our system with other semi-supervised systems in term of accuracy

References

- Iftene A. and Balahur-Dobrescu A., "NamedEntity Relation Mining Using Wikipedia", In Proceedings of the Sixth International Language Resources and Evaluation (LREC'08). 28-30 May, Marrakech, Morocco, 2008.
- Yu X. and Lam W., "Jointly Identifying Entities and Extracting Relations in Encyclopedia Text via A Graphical Model Approach", In Proceedings COLING (Posters), pages 1399–1407, 2010.
- Nakamura-Delloye Y. and Stern R., "Extraction de relations et de patrons de relations entre entités nommées en vue de l'enrichissement d'une ontologie", Actes de TOTh 2011 (Terminologie & Ontologie : Théories et applications), Annecy, France, 2011.
- Rosario B. and Hearst M., "Classifying semantic relations in bioscience text". In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004), Barcelona. July 2004.
- Embarek M. and Ferret O., "Une expérience d'extraction de relations sémantiques à partir de textes dans le domaine médical", In TALN 2007, pages 37–46, Toulouse, France, 2007.
- Kim J., Choe Y. and Mueller K., "Extracting Clinical Relations in Electronic Health Records Using Enriched Parse Trees", INNS Conference on Big Data 2015 Program San Francisco, CA, Pages 274-283, USA 8-10, 2015.
- X. Zhu, "Semi-supervised training tutorial", Technical report, Department of Computer Sciences University of Wisconsin, Madison, USA, 2007.
- Agrawala A., "Training with a probabilistic teacher", IEEE Transactions on Information Theory, Vol. 16, pp.373-379, 1970.
- Clark S., Curran J. R. and Osborne M., "Bootstrapping postaggers using unlabelled data". In CoNLL, 2003
- Drury B. and Delopes A., "the identification of indicators of sentiment using a multi-view self-training algorithm", Oslo Studies in Language 7(1), 379–395. (ISSN 1890-9639 / ISBN 978-82-91398-12-9), 2015.
- Blum A. and Mitchell T., "Combining labeled and unlabeled data with co-training", COLT: Proceedings of the Workshop on Computational Training Theory, 1999.
- Zhou Z. and Li M., "Tri-training: exploiting unlabeled data using three classifiers", IEEE Transactions on Knowledge and Data Engineering, 17:15291541, 2005
- Jimeno Y. A and Aronson A., "Self-training and Co-training in biomedical word sense disambiguation", Proceedings of BioNLP 2011 Workshop. 2011, Portland, Oregon, USA: Association for Computational Linguistics, 182-183, 2011.
- Feng Q., Gui Y., Yang Z. and Wang L., Li Y., "Semisupervised Training Based Disease-Symptom and Symptom-Therapeutic Substance Relation Extraction from Biomedical Literature", Hindawi Publishing Corporation BioMed Research International Volume 2016, Article ID 3594937, 13 pages, 2016.
- Dejori M., M. Bundschuh, S. Martin, T. Volker and Hans-Peter K., "Extraction of semantic biomedical relations from text using conditional random fields", BMC Bioinformatics. 9: 207-10.1186/1471-2105-9-207, 2008.
- Frunza O. and Inkpen D., "Extraction of disease-treatment semantic relations from biomedical sentences". In : Proceedings of the 2010 Workshop on Biomedical Natural Language Processing, Uppsala, Sweden. Association for Computational Linguistics; pp. 91–8, 2010.
- Ben Abacha A. and Zweigenbaum P., "A Hybrid Approach for the Extraction of Semantic Relations from MEDLINE Abstracts", In Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing'11), volume 6608 of Lecture Notes in Computer Science, pages 139-150, Tokyo, Japan., 2011
- John G.H., Kohavi R. and Pfleger K., "Irrelevant features and the subset selection problem". In Proceedings of the Eleventh International Conference on Machine Training, pages 121–129, 1994.
- Liu H., Yu L., "Toward Integrating Feature Selection Algorithms for Classification and Clustering", Department of Computer Science and Engineering, Arizona State University, 2005.
- Navin T., Chappelle O., Weston J. and Elisseeff A., "Feature extraction, foundations and applications, chapter Embedded Methods", Series Studies in Fuzziness and Soft Computing, PhysicaVerlag, Springer, pages 139-167, 2006.
- Dash M. and Liu H., "Feature selection for classification", Intelligent Data Analysis, 1 :131-156, 1997.
- Jolliffe T., "Principal Component Analysis", Springer-Verlag, 1986.
- Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., and Witten IH., "The weka data mining software: an update". ACM SIGKDD Explorations Newsletter;11(1):10–18, 2009.
- Driessens K., Reutemann P., Pfahringer B. and Leschi C., "Using weighted nearest neighbor to benefit from unlabeled data", PAKDD. Mar;60–69, 2006

Contents

Employing Context-Independent GMMs to Flat Start Context-Dependent CTC Acoustic Models Mohamed Elfeky, Parisa Haghani, Seungji Lee, Eugene Weinstein and Pedro Moreno	16
About lexicon adaptation for automatic speech recognition of video data Denis Jouvét, David Langlois, Mohamed Amine Menacer, Dominique Fohr, Odile Mella and Kamel Smaili	21
Building an ASR System for a Low-resource Language Through the Adaptation of a High-resource Language ASR System: Preliminary Results Odette Scharenborg, Francesco Ciannella, Shruti Palaskar, Alan Black, Florian Metze, Lucas Ondel and Mark Hasegawa-Johnson	26
Data Selection in the Framework of Automatic Speech Recognition Ismael Bada, Juan Karsten, Dominique Fohr and Irina Illina	31
An analysis of psychoacoustically-inspired matching pursuit decompositions of speech signals Khalid Daoudi and Nicolas Vinuesa	36
Enthusiasm and disillusionment with voice assistants Franck Poirier	40
Is statistical machine translation approach dead? Mohamed Amine Menacer, David Langlois, Odile Mella, Dominique Fohr, Denis Jouvét and Kamel Smaili	44
Assessing the Usability of Modern Standard Arabic Data in Enhancing Language Model of Limited Size Dialect Conversations Tiba Abdulhameed, Imed Zitouni, Ikhlas Abdel-Qader and Mohamed Abusharkh	49
GAWO: Genetic-based optimization algorithm for SMT Ameur Douib, David Langlois and Kamel Smaili	54
Statistical Machine Translation from Arab Vocal Improvisation to Instrumental Melodic Accompaniment Fadi Al-Ghawanmeh and Kamel Smaili	59
Image2speech: Automatically generating audio descriptions of images Mark Hasegawa-Johnson, Alan Black, Lucas Ondel, Odette Scharenborg and Francesco Ciannella	65
Speaker-dependent Selection of Cohort-utterances for Score Normalization in Speaker Recognition Systems Ayoub Bouziane, Jamal Kharroubi and Aرسالane Zarghili	70
Enhancement of esophageal speech using voice conversion techniques Imen Ben Othmane, Joseph Di Martino and Kais Ouni	75
Sentence Boundary Detection for French with Subword-Level Information Vectors and Convolutional Neural Networks Carlos-Emiliano González-Gallardo and Juan-Manuel Torres-Moreno	80
An empirical study of the Algerian dialect of Social network Abidi Karima and Smaili Kamel	85
Maghrebi Arabic dialect processing: an overview Salima Harrat, Karima Meftouh and Kamel Smaili	90
Automatic Dialectal Recognition in Arabic Broadcast Media Bilal Belainine, Fatma Mallek and Fatiha Sadat	95
Building the Moroccan DarijaWordnet (MDW) using Bilingual Resources Khalil Mrini and Francis Bond	100

A Word Embedding based Method for Question Retrieval in Community Question Answering	
Nouha Othman, Rim Faiz and Kamel Smaili	105
Active Learning for Classifying Political Tweets	
Erik Tjong Kim Sang, Marc Esteve del Valle, Herbert Kruitbosch and Marcel Broersma	110
In-the-wild chatbot corpus: from opinion analysis to interaction problem detection	
Irina Maslowski, Delphine Lagarde and Chloé Clavel	115
SMRE: Semi-supervised Medical Relation Extraction	
Triki Sana, Boujelben Ines and Jamoussi Salma	121