Identification of complex words and passages in medical documents in French

Kim Cheng Sheang, Anaïs Koptient, Natalia Grabar, Horacio Saggion

Identification of complex words and passages in medical documents in French

Kim Cheng Sheang[1] Anaïs Koptient[2] Natalia Grabar[2] Horacio Saggion[1]
(1) LaSTUS Lab/TALN Group, Universitat Pompeu Fabra, Spain
(2) CNRS, Univ Lille, UMR 8163 – STL, F-5900 Lille, France
kimcheng.sheang@upf.edu, anais.koptient.etu@univ-lille.fr,
natalia.grabar@univ-lille.fr, horacio.saggion@upf.edu

RÉSUMÉ

The purpose of automatic text simplification is to provide a new version of documents that are easier to understand by a given population or easier to process by other NLP applications. However, it is important to know what should be simplified exactly within the documents before the simplification is done. Indeed, even in technical and specialized documents, it is unnecessary to simplify everything but just those segments that present understanding difficulty. Typically, the purpose of complex word identification is to diagnose the difficulty of a given document to detect complex words or passages within it. We propose to address the issue of identifying complex words and passages within biomedical documents in French.

ABSTRACT

Identification de mots et passages difficiles dans les documents médicaux en français.

L'objectif de la simplification automatique des textes consiste à fournir une nouvelle version de documents qui devient plus facile à comprendre pour une population donnée ou plus facile à traiter par d'autres applications du TAL. Cependant, avant d'effectuer la simplification, il est important de savoir ce qu'il faut simplifier exactement dans les documents. En effet, même dans les documents techniques et spécialisés, il n'est pas nécessaire de tout simplifier mais juste les segments qui présentent des difficultés de compréhension. Il s'agit typiquement de la tâche d'identification de mots complexes : effectuer le diagnostic de difficulté d'un document donné pour y détecter les mots et passages complexes. Nous proposons de travail sur l'identification de mots et passages complexes dans les documents biomédicaux en français.

MOTS-CLÉS : Détection de mots difficiles, Simplification de texte.

KEYWORDS: Complex word identification, Text simplification.

1 Introduction

The purpose of automatic text simplification (Saggion, 2017) is to provide a new version of documents that are easier to understand by a given population (Son *et al.*, 2008; Paetzold & Specia, 2016b; Chen *et al.*, 2016; Arya *et al.*, 2011; Leroy *et al.*, 2013) or easier to process by NLP applications (Chandrasekar & Srinivas, 1997; Vickrey & Koller, 2008; Blake *et al.*, 2007; Stymne *et al.*, 2013; Wei *et al.*, 2014; Beigman Klebanov *et al.*, 2004). Yet, it is important to know what should be

simplified exactly within the documents before the simplification is done. Indeed, even in technical and specialized documents, it is unnecessary to simplify everything but just those segments that present understanding difficulty. For example, the following sentence from a biomedical document contains difficult words such as (*OPA (acute pulmonary edema), résolutif (resolvent), VNI (NIV), oxygénothérapie (oxygen therapy)*) which, if simplified or explained, could make the text more understandable.

> *Le patient est donc transféré en réanimation : l'OAP est résolutif sous VNI et oxygénothérapie.* (*Hence, the patient is transferred to intensive care : acute pulmonary edema is resolvent with NIV and oxygen therapy.*)

Generally, biomedical documents are often considered one of the most challenging texts to read and understand by a large population. The difficulty is mainly on a lexical level since the vocabulary of medical texts are very specific and because of the document's rich terminological status. Therefore, the purpose complex word identification model (CWI) is to make the diagnosis of the difficulty within a given document in order to detect words or passages within it.

In our work, we propose to address the issue of identifying complex words and passages within biomedical documents in French. We first present some related work (Section 2). We then introduce the reference data and our approach (Section 3) and present the results obtained (Section 4). Finally, we conclude in Section 5. [1]

2 Related Work

Complex word identification has attracted the attention of researchers recently, either as part of text simplification systems (Shardlow, 2013; Paetzold & Specia, 2015) or as an independent task, such as promoted by SemEval 2016 (Paetzold & Specia, 2016a), 2018 CWI shared task (Yimam *et al.*, 2018), or SemEval 2021 (Shardlow *et al.*, 2021).

The first complex word identification SemEval was introduced in 2016 to identify difficult words in an English corpus. The task was a binary classification task in which the model had to decide whether the target word was difficult or not for non-native speakers (Paetzold & Specia, 2016a). Participants exploited different classifiers such as Support Vector Machine (SVM), Decision Tree, Random Forest, Logistic Regression and Recurrent Neural Network (RNN) with features like word embeddings, lexical, morphological, and psycholinguistic properties of the target words and POS tags. The participating systems showed an accuracy between 0.465 and 0.922.

In 2018, the CWI shared task extended its purpose to the identification of difficult words in different languages (English, German, French, Spanish), as well as in a multilingual corpus (Yimam *et al.*, 2018). Two tasks were proposed : binary classification task (to decide whether the word is difficult or not) and probabilistic classification task (to assign a probability to a given word as being difficult). Participants used different classifiers (e.g., SVM, Naive Bayes, Random Forest) and different combinations of features such as word frequency, semantics, lexical, morphological, and psycholinguistic properties. In addition, some participants started to use the context of target words and word embeddings. The results of the binary classification task, F-measures were between 0.176 and 0.874 for the English corpus and between 0.577 and 0.745 for other languages.

1. The data and source code are available at : https://github.com/KimChengSHEANG/MedCWI

The purpose of the second SemEval 2021 task was also to identify difficult words in texts from different genres in English : European Parliament, Bible and biomedical texts (Shardlow *et al.*, 2021). Two main differences from previous tasks : words are annotated using a 5-point Likert scale (from *very easy* to *very difficult*) and consideration of polylexical units. Participants used language models, such as BERT (Devlin *et al.*, 2018) or RoBERTa (Liu *et al.*, 2019), and Gradient Boosted Regression. The evaluation metric was Pearson's Correlation and the participants obtained scores between 0.7886 and -0.0272 for the processing of single words and scores between 0.8612 and 0.1860 for the processing of polylexical units.

Besides challenge papers, (Gala *et al.*, 2014) proposed an approach to predict the lexical complexity of French words for non-native learners. The authors use SVM with different features (e.g., word length, number of phonemes, number of syllables, phoneme/spelling coherence). They obtain accuracy between 43% and 63%. In another work, three methods for the detection of difficult words in general language English corpora are evaluated (Shardlow, 2013) : (1) simplify everything, (2) exploit frequency using reference corpus, (3) train SVM model. The SVM-based method shows the highest performance with 0.771 precision, while the simplifying everything method has 0.738 precision and the frequency-based method has 0.709 precision. The frequency threshold is exploited in another work for the detection of familiarity with medical terms (Zeng *et al.*, 2005). The experience obtains 0.196 mean absolute error and 0.293 root mean square error. Yet, another way to determine word complexity is based on the rarity of words : the words that are not found in different lexica are considered as difficult (Borst *et al.*, 2008). This method shows 92% accuracy.

Sheang (2019) has introduced a Convolutional Neural Network (CNN) approach that combines different features to identify the complex word(s) in a sentence. The model uses GloVe word embedding (Pennington *et al.*, 2014) feature along with other features such as word frequency, word length, number of syllables, number of vowels, TFIDF, part-of-speech, dependency, and stop word. The model performs quite well on three datasets : English, German, and Spanish. However, there was no French language included in the experiments.

3 Experiments

In this section, we present the dataset, the preprocessing steps, and the models we propose in this paper. The goal of the experiments is to classify the target text in a sentence as complex or not complex and evaluate it based on the manual annotation of the dataset.

3.1 Dataset

We use 100 French clinical cases randomly selected from the CAS corpus (Grabar *et al.*, 2018). This corpus contains a total of 41,384 words. Clinical cases are medical documents similar to clinical reports. They describe the medical background of patients, the reason of their consultation, the healthcare process and treatments proposed and performed, and the outcome. Such clinical documents can be encountered by patients in their everyday lives, which motivates their use in the current work. Clinical cases deal with different topics and specialties. They are published and are freely accessible from different sources. They are already anonymous. The corpus with clinical cases is pre-processed. The documents are syntactically analyzed by the Cordial parser (Laurent *et al.*, 2009) to divide them

into syntactic groups (chunks). When a given word belongs to a chunk within another chunk, we keep the minimal chunk. The corpus contains in total 15,053 chunks.

[Ses antécédents médicaux] [montrent] [notamment] [un diabète gestationnel probable] [et une HG] [lors de sa première grossesse]. [La patiente] [avait alors été hospitalisée] et [avait recu] [un traitement intraveineux] [de métoclopramide associé] [à de la diphénhydramine suivi] [d'un relais] [par voie orale] [au métoclopramide et] [à l'hydroxyzine]. [Une réaction extrapyramidale] ([rigidité] [de la mâchoire et] [difficulté] [à parler]) [avait nécessité] [l'arrêt] [du métoclopramide]. [L'hydroxyzine] [avait] [ensuite été remplacée] [par l'association] [de doxylamine] [et de pyridoxine] (DiclectinMD).

[Her medical background] [shows] [a probable gestational diabetes] [and an HG] [during her first pregnancy]. [The patient] [had then been hospitalized] and [received] [an intravenous treatment] [of metoclopramide with] [diphenhydramine followed] [by oral treatment] [with metoclopramide and] [hydroxyzine]. [An extrapyramidal reaction] ([jaw] [stiffness and] [difficulty] [to talk]) [caused] [the cessation] [of metoclopramide]. [Hydroxyzine] [had] [then been replaced] [by the combination] [of doxylamine] [and pyridoxine] (Di-clectinMD).

FIGURE 1 – Example of an annotated clinical case

Documents are then annotated manually by nine annotators in order to mark up the chunks with understanding difficulty. The annotators are all native French speakers, and they have no medical knowledge or training. Few of them are chronically ill. During the annotation process, the annotators were advised not to use dictionaries or information available on the Internet. They had to do the annotations on the basis of their own knowledge. The annotators are presented with whole documents, where chunks are between brackets, such as shown in Figure 1. For each chunk, the annotators have to indicate whether they cannot understand it (in red) or whether they are not sure to understand it (in blue). In the case they understand a given chunk, they do not have to annotate it.

In the end, each document is annotated by at least four annotators, while some documents are annotated by up to six annotators. We computed the kappa of Fleiss (Fleiss, 1971) for four annotators who annotated all the documents, which gives a low 0.175 kappa. For some pairs of annotators, kappa shows slightly higher values (0.292 and 0.316). This means that this annotation task is very subjective and heavily depends on own knowledge and experience of each person.

The corpus is then segmented into sentences containing one target chunk per sentence. In total, we have 9,709 sentences with 3,482 complex and 6,227 non-complex chunks. Then for each training and evaluation, we shuffle the data with a different seed number and split it into three parts : 70% for training, 15% for validation, and 15% for testing.

3.2 Features

Our models rely on several features to perform the classification. Here are all the features incorporated :

1. *FastText Embedding* (Bojanowski *et al.*, 2016), a language model pre-trained on large unlabeled corpora, is used as word representation. We prefer FastText of other non-contextualized word embeddings, such as GloVe (Pennington *et al.*, 2014), because FastText is trained with

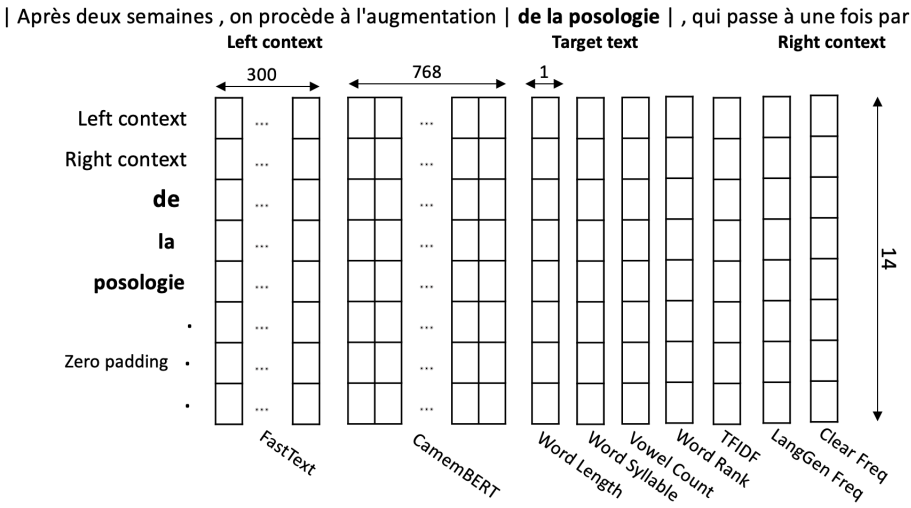

FIGURE 2 – Sentence representation of all features.

both word and subword information. It can deal better with rare or unknown words, which we believe suitable for medical texts ;

2. *CamemBERT Embedding* (Martin *et al.*, 2020) a French version of BERT (Devlin *et al.*, 2018), is pre-trained on a large amount of texts using a corrupt span masking approach. In this experiment, we use Flair (Akbik *et al.*, 2019) to extract word embeddings for the whole sentence from the 12 embedding layers, and then compute the average. After that, we extract the embedding of each word in the sentence with the dimension of 768 each ;
3. *Word Length* is the number of characters in a word ;
4. *Word Syllable* is the number of syllables in each word, extracted using PyHyphen [2] ;
5. *Vowel Count* is the number of vowels in each word ;
6. *Word Rank* is the frequency order taken from FastText pre-trained model ;
7. *TFIDF* (Salton, 1991) permits to measure how a sentence is relevant to a document ;
8. *LangGen Frequency* is a frequency computed from French Wikipedia ;
9. *Clear Frequency* is a frequency computed from a French medical corpus (Grabar *et al.*, 2018).

3.3 Preprocessing

Figure 2 shows vector representations of all features. First, each sentence is separated into three parts : target chunk, its left context, and its right context. Then, we compute the values of all features, as shown in Section 3.2, and cache them into files so that we can load them back quickly depending on the feature set. The feature values of the left context and right context are on average. They are used as context information for the target chunk. However, the feature values of each word in the target chunk are the actual values, which is one row per word, as shown in Figure 2.

2. https ://github.com/dr-leo/PyHyphen

3.4 Models

In this section, we describe each model of our experiments.

CNN or Convolutional Neural Network (LeCun *et al.*, 1995) is often used for image classification tasks, but it is also efficient for complex word identification (Aroyehun *et al.*, 2018; Sheang, 2019). Our model follows the approach of (Sheang, 2019) by adding more frequency lists, removing some linguistic features, and replacing GloVe with FastText and CamemBERT. To train the model, first, the vector representations of all features for a sentence (as shown in Figure 2) are appended as a two-dimensional vector depending on the number of features. Then, we pass it through CNN layer with the number of filters 128, the stride of 1, and the kernel size of 3, 4, and 5. After that, we apply Max Pooling and pass it through three Fully-Connected (FC) layers with the output of 256, 64, and 1. The last layer is the output, which gives 0 (non-complex) or 1 (complex). The model is trained with Adam optimizer, a learning rate of 0.003, dropout 15% between each layer, and batch size of 64. We use weighted cross-entropy as a loss function with the weight of 1.7 for the positive to counterbalance the negative, as the data in the negative class is 64% and positive 36%. We train the model for 200 epochs and validate it every 100 iterations, then save the model that achieves the highest macro F1-score and use it to evaluate the test set.

CatBoost (Dorogush *et al.*, 2018) is a gradient boosting on Decision Trees library is often used for ranking, regression, and classification tasks. CatBoost is an ensemble learning library that combines multiple machine learning algorithms (Decision Trees) to obtain a better model. To train the model, we follow the same data preparation step as in the CNN model and then flatten it into a long vector. The model is trained for 1200 iterations with the learning of 0.03.

XGBoost (Chen & Guestrin, 2016) stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree machine learning library similar to CatBoost. It is one of the leading machine learning libraries for regression, classification, and ranking problems. To train the model, the data is prepared the way as in CatBoost model, and then trained with the max depth of 10, the learning rate of 0.03, and the number of estimators of 500.

4 Results and Discussion

Table 1 shows the results of our three models (CNN, CatBoost, and XGBoost) trained with different feature sets. The results are indicated in a macro average of precision (P), recall (R), and F1-score (F1). The models are trained with each feature set five times : the average values are indicated. The column *Features* indicates the combinations of features used in each model. Each feature number corresponds to those in Section 3.2. The CNN model performs better in all cases, except when the CNN model is trained without word embeddings, it performs lower than CatBoost and XGBoost. We get up to 0.854 F1-score.

We started with the CNN model trained with only FastText word embedding, and then we kept adding more features one by one. We could see that the result was improved each time we added a new feature. Then, we tried removing the embedding feature, and the result dropped significantly. Next, we tried BERT embedding (CamemBERT), and the result was way better than all models with FastText. It can be due to the fact that BERT embedding has captured better information on words than FastText. Next, we combined FastText with BERT ; as a result, the CNN model performs worse

Features	CNN			CatBoost			XGBoost		
	P	R	F1	P	R	F1	P	R	F1
1	0.818	0.811	0.814	0.798	0.773	0.783	0.798	0.770	0.780
1, 3	0.817	0.813	0.815	0.799	0.771	0.781	0.796	0.770	0.779
1, 3, 4	0.819	0.815	0.816	0.804	0.781	0.790	0.800	0.772	0.782
1, 3, 4, 5	0.820	0.817	0.818	0.803	0.780	0.788	0.797	0.770	0.780
1, 3, 4, 5, 6	0.826	0.818	0.821	0.829	0.812	0.819	0.799	0.774	0.783
1, 3, 4, 5, 6, 7	0.823	0.820	0.821	0.830	0.813	0.820	0.826	0.811	0.817
1, 3, 4, 5, 6, 7, 8	0.824	0.824	0.824	0.826	0.809	0.816	0.825	0.812	0.818
1, 3, 4, 5, 6, 7, 8, 9	0.827	0.826	0.826	0.831	0.817	0.823	0.826	0.812	0.818
3, 4, 5, 6, 7, 9	0.787	0.778	0.781	0.803	0.786	0.793	0.802	0.789	0.795
1, 2	0.848	0.843	0.845	0.838	0.824	0.830	0.834	0.817	0.824
2	0.848	0.847	0.847	0.823	0.807	0.814	0.824	0.799	0.809
2, 3	0.848	0.847	0.847	0.825	0.811	0.817	0.825	0.802	0.811
2, 3, 4	0.852	0.851	0.851	0.826	0.812	0.818	0.825	0.802	0.811
2, 3, 4, 5	**0.854**	0.852	0.853	0.822	0.808	0.814	0.824	0.798	0.808
2, 3, 4, 5, 6	0.851	0.850	0.851	0.821	0.806	0.812	0.820	0.796	0.805
2, 3, 4, 5, 6, 7	0.851	0.847	0.849	0.829	0.817	0.822	0.834	0.816	0.823
2, 3, 4, 5, 6, 7, 8	0.851	0.846	0.848	0.822	0.810	0.815	0.827	0.815	0.820
2, 3, 4, 5, 6, 7, 8, 9	0.849	0.844	0.846	0.832	0.817	0.824	0.835	0.822	0.827
2, 3, 4, 5, 6, 8, 9	0.853	**0.856**	**0.854**	0.826	0.815	0.820	0.834	0.822	0.827

TABLE 1 – This table shows the results in a macro average of precision (P), recall (R), and F1-score (F1) of our three models trained with different combinations of features. All models are trained with each feature set five times and computed the average. Higher value means better. The column feature lists all combinations of features used in the training of each model, and each number represents the corresponding feature listed in Section 3.2.

than the model with BERT alone, whereas CatBoost and XGBoost models perform better; especially the CatBoost model performs the best among all of its models. Even though CatBoost and XGBoost models perform pretty well, it is still significantly lower than the CNN model.

In comparison with all the models, the results show that in most cases the CNN model performs better than CatBoost and XGBoost, except in the model without the word embedding, this could be an indication that CatBoost and XGBoost learn better than the CNN model when having less information.

5 Conclusion

In this paper, we have proposed three classifiers based on CNN, CatBoost, and XGBoost with different feature sets such as FastText, BERT, word length, word syllable, vowel count, word rank, TFIDF, LangGen frequency, and Clear frequency to detect complex words in French biomedical documents. We have also created the dataset and provided the set of experiments for the evaluation. The results have shown that our models perform quite well, and the CNN model overtakes the others.

Acknowledgments

Our work is partly supported by the project Context-aware Multilingual Text Simplification (ConMuTeS) PID2019-109066GB-I00/AEI/10.13039/501100011033 awarded by Ministerio de Ciencia, Innovación y Universidades (MCIU), by Agencia Estatal de Investigación (AEI) of Spain, and by the French National Agency for Research (ANR) as part of the CLEAR project (Communication, Literacy, Education, Accessibility, Readability), ANR-17-CE19-0016-01.

Références

AKBIK A., BERGMANN T., BLYTHE D., RASUL K., SCHWETER S. & VOLLGRAF R. (2019). Flair : An easy-to-use framework for state-of-the-art nlp. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, p. 54–59.

AROYEHUN S. T., ANGEL J., ALVAREZ D. A. P. & GELBUKH A. (2018). Complex word identification : Convolutional neural network vs. feature engineering. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, p. 322–327.

ARYA D. J., HIEBERT E. H. & PEARSON P. D. (2011). The effects of syntactic and lexical complexity on the comprehension of elementary science texts. *Int Electronic Journal of Elementary Education*, **4**(1), 107–125.

BEIGMAN KLEBANOV B., KNIGHT K. & MARCU D. (2004). Text simplification for information-seeking applications. In R. MEERSMAN & Z. TARI, Éds., *On the Move to Meaningful Internet Systems 2004 : CoopIS, DOA, and ODBASE*. Berlin, Heidelberg : Springer, LNCS vol 3290.

BLAKE C., KAMPOV J., ORPHANIDES A., WEST D. & LOWN C. (2007). Query expansion, lexical simplification, and sentence selection strategies for multi-document summarization. In *DUC*.

BOJANOWSKI P., GRAVE E., JOULIN A. & MIKOLOV T. (2016). Enriching word vectors with subword information. arxiv 2016. *arXiv preprint arXiv :1607.04606*.

BORST A., GAUDINAT A., BOYER C. & GRABAR N. (2008). Lexically based distinction of readability levels of health documents. In *MIE 2008*. Poster.

CHANDRASEKAR R. & SRINIVAS B. (1997). Automatic induction of rules for text simplification. *Knowledge Based Systems*, **10**(3), 183–190.

CHEN P., ROCHFORD J., KENNEDY D. N., DJAMASBI S., FAY P. & SCOTT W. (2016). Automatic text simplification for people with intellectual disabilities. In *AIST*, p. 1–9.

CHEN T. & GUESTRIN C. (2016). Xgboost : A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, p. 785–794.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2018). Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv :1810.04805*.

DOROGUSH A. V., ERSHOV V. & GULIN A. (2018). Catboost : gradient boosting with categorical features support.

FLEISS J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, **76**(5), 378–382.

GALA N., FRANÇOIS T., BERNHARD D. & FAIRON C. (2014). A model to predict lexical complexity and to grade words (un modèle pour prédire la complexité lexicale et graduer les mots) [in French]. In *Proceedings of TALN 2014*, p. 91–102, Marseille, France.

GRABAR N., CLAVEAU V. & DALLOUX C. (2018). CAS : French corpus with clinical cases. In *LOUHI 2018*, p. 1–12, Bruxelles, Belgique.

LAURENT D., NÈGRE S. & SÉGUÉLA P. (2009). L'analyseur syntaxique Cordial dans Passage. In *Traitement Automatique des Langues Naturelles (TALN)*.

LECUN Y., BENGIO Y. *et al.* (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, **3361**(10), 1995.

LEROY G., KAUCHAK D. & MOURADI O. (2013). A user-study measuring the effects of lexical simplification and coherence enhancement on perceived and actual text difficulty. *Int J Med Inform*, **82**(8), 717–730.

LIU Y., OTT M., GOYAL N., DU J., JOSHI M., CHEN D., LEVY O., LEWIS M., ZETTLEMOYER L. & STOYANOV V. (2019). Roberta : A robustly optimized bert pretraining approach. *arXiv preprint arXiv :1907.11692*.

MARTIN L., MULLER B., ORTIZ SUÁREZ P. J., DUPONT Y., ROMARY L., DE LA CLERGERIE E., SEDDAH D. & SAGOT B. (2020). Camembert : a tasty french language model. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. DOI : 10.18653/v1/2020.acl-main.645.

PAETZOLD G. & SPECIA L. (2015). LEXenstein : A framework for lexical simplification. In *ACL-IJCNLP*, p. 85–90.

PAETZOLD G. & SPECIA L. (2016a). SemEval 2016 task 11 : Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, p. 560–569, San Diego, California : Association for Computational Linguistics. DOI : 10.18653/v1/S16-1085.

PAETZOLD G. H. & SPECIA L. (2016b). Benchmarking lexical simplification systems. In *LREC*, p. 3074–3080.

PENNINGTON J., SOCHER R. & MANNING C. D. (2014). Glove : Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, p. 1532–1543.

SAGGION H. (2017). *Automatic text simplification*. Morgan & Claypool Publishers.

SALTON G. (1991). Developments in automatic text retrieval. *Science*, **253**, 974–979.

SHARDLOW M. (2013). A comparison of techniques to automatically identify complex words. In *ACL Student Research Workshop*, p. 103–109.

SHARDLOW M., EVANS R., PAETZOLD G. H. & ZAMPIERI M. (2021). SemEval-2021 task 1 : Lexical complexity prediction. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, p. 1–16, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.semeval-1.1.

SHEANG K. C. (2019). Multilingual complex word identification : Convolutional neural networks with morphological and linguistic features. In *Proceedings of the Student Research Workshop Associated with RANLP 2019*, p. 83–89, Varna, Bulgaria : INCOMA Ltd. DOI : 10.26615/issn.2603-2821.2019_013.

SON J. Y., SMITH L. B. & GOLDSTONE R. L. (2008). Simplicity and generalization : Short-cutting abstraction in children's object categorizations. *Cognition*, **108**, 626–638.

STYMNE S., TIEDEMANN J., HARDMEIER C. & NIVRE J. (2013). Statistical machine translation with readability constraints. In *NODALIDA*, p. 1–12.

VICKREY D. & KOLLER D. (2008). Sentence simplification for semantic role labeling. In *Annual Meeting of the Association for Computational Linguistics-HLT*, p. 344–352.

WEI C.-H., LEAMAN R. & LU Z. (2014). SimConcept : A hybrid approach for simplifying composite named entities in biomedicine. In *BCB '14*, p. 138–146.

YIMAM S. M., BIEMANN C., MALMASI S., PAETZOLD G., SPECIA L., ŠTAJNER S., TACK A. & ZAMPIERI M. (2018). A report on the complex word identification shared task 2018. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, p. 66–78, New Orleans, Louisiana : Association for Computational Linguistics. DOI : 10.18653/v1/W18-0507.

ZENG Q. T., KIM E., CROWELL J. & TSE T. (2005). A text corpora-based estimation of the familiarity of health terminology. In *ISBMDA 2006*, p. 184–92.