

FrenLyS: A Tool for the Automatic Simplification of French General Language Texts

Eva Rolin
UCLouvain
CENTAL

Quentin Langlois
UCLouvain

Patrick Watrin
UCLouvain
CENTAL

Thomas François
UCLouvain
CENTAL

Abstract

Lexical simplification (LS) aims at replacing words considered complex in a sentence by simpler equivalents. In this paper, we present the first automatic LS service for French, *FrenLyS*, which offers different techniques to generate, select and rank substitutes. The paper describes the different methods proposed by our tool, which includes both classical approaches (e.g. generation of candidates from lexical resources, frequency filter, etc.) and more innovative approaches such as the exploitation of CamemBERT, a model for French based on the RoBERTa architecture. To evaluate the different methods, a new evaluation dataset for French is introduced.

1 Introduction

It is widely acknowledged that reading difficulties, either due to insufficient education or to mental deficiencies for example, can hinder access to information, which is likely to result in a loss of autonomy and freedom (Mutabazi and Wallenhorst, 2020). Faced with this challenge, researchers imagined applying natural language processing (NLP) to automatically transform sentences in a text in order to make it more readable, thus facilitating access to information. This is the objective pursued in the field of *Automatic Text Simplification* (ATS), in which the main goal is to preserve grammaticality and meaning while carrying out effective transformations to make the text simpler.

ATS is generally investigated at the sentence level (Alva-Manchego et al., 2020) and has therefore mostly focused on two subtasks: on the one hand, lexical simplification, described by Saggion (2017) as the task of “replacing difficult words with easy-to-read (or understand) expressions while preserving the meaning of the original text segments”, on the other hand, syntactic simplification, that

consists in simplifying syntactic structures in a sentence by carrying out various transformations (splitting, clause deletion, etc.). Both tasks have been the subject of a great deal of research, as synthesized in Shardlow (2014); Siddharthan (2014); Saggion (2017); Paetzold and Specia (2017b); Alva-Manchego et al. (2020); Al-Thanyyan and Azmi (2021), but most of it has been carried out on English or, to a lesser extent, on Spanish. Other languages are hardly represented, which is also the case for French (Gala et al., 2018; Elguendouze, 2020). This why we have chosen to focus on this one.

In French, ATS was addressed at first at the syntactic level (Seretan, 2012; Brouwers et al., 2014) using rule-based systems. In parallel, lexical simplification was also investigated, based on lexicons or resources (Billami et al., 2018; Cardon, 2018; Hmida et al., 2018). Due to the lack of training data, machine translation approaches - which are standard for English - were applied to French (Rauf et al., 2020) only very recently, with mixed results. As a result, the situation of ATS for French is clearly lagging behind that of English. The only simplification package freely available for the research community has been published recently (Wilkens and Todirascu, 2020) and it remains preliminary and focused exclusively on syntax. *AMesure*, a web platform designed to help writers of administrative texts to write in plain language (François et al., 2020) is more encompassing. However, it is limited to detecting complex phenomena and suggesting simplifications.

In this paper, we aim to fill the gap in lexical simplification (LS) tools and resources for French by developing a tool in which several standard approaches of LS are available and by building a reference dataset to evaluate our results. This library, called *FrenLyS* for French Library for Simplification, follows the LS process first described

by [Shardlow \(2014\)](#) as a sequence of four steps: identifying complex terms, generating candidates for substitution, selecting the best candidates, and ranking them according to their degree of readability. Our work draws from similar packages in other languages, such as *LEXenstein* ([Paetzold and Specia, 2015](#)) for English, the *EASIER tool* ([Alarcon et al., 2019](#)) and *LexSis* ([Bott et al., 2012](#)) for Spanish or the work by [Qiang et al. \(2021\)](#).

The article is structured as follows. Section 2 presents the state of the art of lexical simplification. In Section 3, we describe the different approaches we implemented for each of the LS steps. Section 4 describes the methodology used to evaluate our approaches, which includes a new reference dataset for French. In section 5 we report and discuss the performance of each of these approaches on our evaluation dataset.

2 Related Work

The task of lexical simplification was first investigated by [Carroll et al. \(1998\)](#) who exploited a rather simple solution: they obtained candidates for substitution using WordNet synonyms ([Miller, 1995](#)) and ranked them according to their frequency. As a result of this work, researchers tried to improve different aspects of this process, either by collecting synonyms ([De Belder and Moens, 2010](#)), or by ranking the candidates ([Biran et al., 2011a](#)), etc. In his survey of the field, [Shardlow \(2014\)](#) provided a clear view of the different challenges within LS, identifying four steps in which recent work can be classified.

Complex Word Identification The first step in lexical simplification is the complex word identification (CWI). This step has been the object of several shared tasks ([Paetzold and Specia, 2016c](#); [Yimam et al., 2018](#)) and aims at identifying in a text the words or expressions likely to be problematic for a target audience of readers and on which the LS system should be applied. As [Gooding and Kochmar \(2019\)](#) pointed out, early works on the complex word identification operated by simplifying all words ([Thomas and Anderson, 2012](#); [Bott et al., 2012](#)) or were based on a threshold t over a given metric of simplicity (e.g. word frequency) that separates simple from complex words ([Biran et al., 2011b](#)). Another approach consists in finding complex words with the help of a lexicon : if the word appears in the resource it is considered as complex, otherwise as simple. This method has

been mostly used for lexical simplification of medical texts ([Chen et al., 2016](#); [Deléger and Zweigenbaum, 2009](#)). Other more recent attempts either used machine learning to classify words as either complex or simple based on some features such as word length, word frequency, number of senses, etc. ([Shardlow, 2013](#); [Alarcon et al., 2019](#)).

Substitution Generation Once complex terms have been identified, the next step is to produce candidates that can replace for the target complex word. This step, called substitution generation (SG), is most often carried out querying linguistic lexical resources, as evidenced by the work of [Carroll et al. \(1998\)](#), [Bott et al. \(2012\)](#), or [Hmida et al. \(2018\)](#). They generate synonyms by querying lexical databases such as *WordNet* or synonym resources such as *ReSyf* ([Billami et al., 2018](#)). As it is not always easy to find lexical databases and as those might have limited coverage, [Horn et al. \(2014a\)](#) proposed to use parallel corpora – Wikipedia and Simple Wikipedia – to automatically extract lexical simplification rules. [Deléger and Zweigenbaum \(2009\)](#) resorted to paraphrases to replace target complex words, a strategy that is more relevant for specialized languages. A currently popular approach was first suggested by [Glavaš and Štajner \(2015\)](#). It consists in obtaining synonyms in an unsupervised way relying on semantic representations such as embeddings. The complex word to be substituted is projected in the semantic space in order to generate the N closest semantic neighbors. More recently, [Qiang et al. \(2019\)](#) used BERT in a similar fashion.

Substitution Selection In order to obtain semantically correct sentences, each candidate has to go through a disambiguation step. The substitution selection (SS) step aims to decide which of the candidates collected at generation step best fits the context of the sentence to be simplified. [De Belder and Moens \(2012\)](#) proposed to carry out the task of disambiguation using a Latent Words Language Model (LWLM): they use Bayesian networks to represent words and their contextual meaning. Other studies took advantage of word sense disambiguation systems to explicitly label the senses of both the target and the candidates, in order to select a candidate having the same sense as the target ([Thomas and Anderson, 2012](#); [Nunes et al., 2013](#)). A third line of research leveraged semantic models to compare the semantic similar-

ity of each candidate with the sentence to simplify. [Bott et al. \(2012\)](#) exploited a vector space model, whereas [Paetzold and Specia \(2015\)](#) rather used a word embedding model. It is also possible to perform this step in a simpler way, by removing all candidates who do not share the same part of speech as the word to be replaced ([Paetzold and Specia, 2013](#)).

Substitution Ranking After having identified the complex words, generated synonyms, and selected the most coherent ones, the final step of LS consists in ranking the remaining candidates according to their reading ease. The first LS systems generally resorted on frequency ([Carroll et al., 1998](#); [De Belder and Moens, 2010](#); [Specia et al., 2012](#)) where it is considered that more frequent words are easier to understand. [Specia et al. \(2012\)](#) showed that this simple rule actually represents a very strong baseline, as it outperformed 9 out of the 11 ranking systems engaged in this task of SemEval 2012 ([Specia et al., 2012](#)). Other studies proposed simplicity metrics that can be combined word characteristics: [Biran et al. \(2011a\)](#) and [Bott et al. \(2012\)](#) combine frequency with word length. Finally, it is also possible to use statistical ranking algorithms ([Horn et al., 2014a](#); [François et al., 2016](#)) that we can combine with neural networks ([Paetzold and Specia, 2017a](#)).

Available datasets In parallel to the design of new LS methods, the development of reference datasets to train and evaluate those methods is key. Several datasets for lexical simplification are available for English, such as SemEval 2012 ([Specia et al., 2012](#)), LSeval ([De Belder and Moens, 2012](#)), LexMTurk ([Horn et al., 2014b](#)), NNSeval ([Paetzold and Specia, 2016b](#)), and BenchLS ([Paetzold and Specia, 2016d](#)). Other languages are not so well resourced: there are only 2 datasets for Japanese – SNOW E4 ([Kajiwara and Yamamoto, 2015](#)) and BCCWJ ([Kodaira et al., 2016](#)) –, but, to our knowledge, none for French. In French, the only available resource for text simplification is the ALECTOR corpus ([Gala et al., 2020](#)). It consists in 79 parallel texts with information about complex words, but there are no validated simpler synonyms, which are required to assess LS approaches.

3 Proposed Approach

Our system is the first to offer several methods for generating candidates for substitution, selecting

them based on semantic similarity with the target word and ranking them by difficulty in French. Most of these methods have been previously applied to English, and we adapted them to the case of French. A few are new. All of them are described hereafter.

It should be noted that *FrenLyS* does not implement any complex word identification algorithm. We believe this is a very complex task, which should be addressed as a whole and actually is ([Yimam et al., 2018](#)), especially because CWI requires to take into account the readers’ characteristics. Methods based on lexical characteristics or word lists overlook the reader’s characteristics and [Lee and Yeung \(2019\)](#) have rightly stressed that current approaches offer the same substitutions regardless of users. This tool is therefore based on the prerequisite that complex words already have been identified. For the sake of the evaluation of our tool, we relied on a manual annotation of complex words in our test set (see Section 4).

3.1 Substitution Generation

The task of substitution generation aims to generate candidate synonyms to replace complex words. To carry out this step, *FrenLyS* proposes three methods: synonyms are directly obtained from a resource of synonyms (see [ReSyf generator](#)), or are generated by embeddings, either produced by *FastText* (see [FastText generator](#)) or by the BERT architecture (see [CamemBERT generator](#)). We also propose a 4th approach (see [CamemBERT union](#)) that combines *Camembert* with *ResyF* or with *FastText* to take advantage of the contextual information that the model can provide.

ReSyf generator This module generates candidates from the graduated and disambiguated synonyms resource *ReSyF* ([Billami et al., 2018](#)). It is built from the semantic network *JeuxDeMots* and contains 57,589 entries that are connected to 148,648 synonyms (both in their lemma form). Its main asset is that synonyms corresponding to different meanings of a word have been manually and automatically clustered into different synsets. Another interesting feature of *ReSyF* is that the synonyms gathered in the same synset are ranked according to reading ease. Based on those characteristics, our method simply consults *ReSyF* using the lemma of the complex word and returns the lemmas of the top three simplest synonyms in each corresponding ‘synset’. At this step, we do not try

to disambiguate the meaning of the word to substitute, as this is the role of the substitution selection step.

FastText generator *FastText* (Bojanowski et al., 2017) is a library for efficient learning of word representations. Its advantage for our task is that it proposes character n-gram embeddings: we can thus obtain a vector representation even for a word that does not exist in the training corpus. Thanks to this technique, we return, for any given complex word, its k-nearest semantic neighbors (the inflected forms) based on cosine similarity.

CamemBERT generator In the same way that Qiang et al. (2019) generated synonyms with *BERT*, we rely on a pre-trained version of *CamemBERT* (Martin et al., 2020), based on the RoBERTa architecture, on the French subcorpus of the multilingual corpus OSCAR. Our method employs the masked language model (MLM) that masks some percentage of input tokens and predicts the masked words from their right and left contexts (Qiang et al., 2019). The idea is to mask the complex word and use the top predicted words (inflected forms) as candidate for substitution. As *FastText*, *CamemBERT* proposes a solution to deal with out-of-vocabulary tokens through their decomposition in wordpieces.

CamemBERT union This method is based on the 2 following observations : *ReSyf* and *FastText* generators only care about the complex word and not its context while *CamemBERT* only cares about the context but does not know the complex word. The solution is to combine the advantages of both approaches by computing the *CamemBERT*-score for each substitute generated by *ReSyf* or *FastText*. For this purpose, we predicted the n best candidates with the *CamemBERT* generator and the k best candidates for the other method. Then we retain only the words that are in the intersection of the two generators and sort them by their new score.

3.2 Substitution Selection

This step takes the list of candidate synonyms and selects only those that are acceptable in the context of the complex word to replace.

We have decided to implement two of the four approaches covered in section 2, as they rely on very different strategies: either by eliminating candidates that do not have the same part of speech (see **POS selector**), or by leveraging language models

such as *FastText* to verify the semantic compatibility between the candidate and the context (**FastTextWord selector**, **FastTextSentence selector**).

POS selector Following Paetzold and Specia (2013), we decided to include a function within our generation methods that checks whether the generated candidates and the word to be replaced share the same parts of the speech. To do this, we used the possible tags for this word as given in the *Delaf* dictionary¹. If the intersection of POS-tags for the 2 words (complex word and candidate) is empty, the candidate is rejected.

FastTextWord selector For each synonym, this selector first retrieves the *FastText* embeddings of the complex word and this synonym and compute the cosine similarity between both vectors. The more similar the meanings of both words are, the closer their vectors are. We therefore select candidates for which the cosine similarity with the complex word is greater than the heuristic threshold of 0.5.

FastTextSentence selector Instead of directly comparing the vectors of the complex words and their synonyms, as in the previous approach, we use the context of the complex word for vectorization: we compute the cosine distance between the vectors of the synonyms and the vector of the complex sentence. We select the candidates with a similarity score greater than the heuristic threshold of 0.35.

3.3 Substitution Ranking

Finally, the last part of our system classifies the synonyms according to their degree of reading ease.

For this step, we referred to common ranking methods in the literature (cf. section 2) as we relied on frequency (see **Lexique3 ranker**) and in a slightly more original way, we provide a method that ranks words according to the number of meanings they have and frequency (see **FreNetic ranker**). We also propose a method that combines various linguistic prescriptors through an SVMRank algorithm (Herbrich et al., 2000) (see **SVM ranker**).

Lexique3 ranker For this method, we use the commonly acknowledged fact that the more frequent a word is, the simpler it is. To obtain the

¹The *Delaf* (Courtois, 2004) contains about 792,260 entries (inflected forms). For each entry, the dictionary provides the following information: lemma, pos and inflectional information (e.g. *dictionaries,dictionary.N*).

frequency of the candidates for substitution, we use the French lexical database *Lexique3* (New, 2006) which provides, for 140,000 words of the French language, their frequencies of occurrences in literary texts and movie subtitles. We have chosen to use the frequencies estimated on the corpus of film subtitles because it contains a more up-to-date vocabulary.

FreNetic ranker In the same way as Elhadad (2006), this ranker exploits polysemy as a measure of familiarity and therefore of difficulty. Words from the general lexicon are more polysemous while technical terms tend to be monosemic. To collect the number of senses, we relied on *FreNet*², a python API for *WOLF* (Sagot and Fiser, 2008), a free French Wordnet. Synonyms are therefore ranked according to their number of meanings (more is easier). When several words get the same number of senses, we decide to further rank them based on their frequencies.

SVM ranker We also propose to perform the ranking task using a SVMRank algorithm described in François et al. (2016). It is able to rank any set of words using 21 word characteristics such as word frequency, presence of the word in a list of simple words, number of phonemes, number of letters, number of senses, number of orthographical neighbors, etc. To train it, we used the Manulex vocabulary list (Lété et al., 2004) that includes 19,038 lemmas annotated with their level of complexity. Based on that information, we prepared training pairs of two words, one of which is known to be more complex than the other, which were fed to the SVMRank algorithm. In their paper, François et al. (2016) report an accuracy of 77% with 10-fold cross-validation and a mean reciprocal rank of 0.84, obtained on a reference dataset of 40 synsets including a total of 150 synonyms that were ranked by 40 human annotators.

4 Evaluation Process

It is usual to create evaluation corpora with the help of human annotators but this requires time and lots of annotators, which may also overlook some valid synonyms. Therefore, we opted for a hybrid approach, i.e. we chose to use our synonym generation methods to propose an exhaustive list of synonyms and then we called upon annotators to select them in context and rank them according

to their difficulty. The advantage of this approach is that it combines several methods from very different generations, including a synonym dictionary that was created from propositions submitted by humans (Lafourcade, 2007). In this way, we still collect data made by humans but *a priori*. We explain the corpus creation process in the next section before discussing the evaluation measures we used.

4.1 A Tailor-Made Evaluation Corpus

We decided to fill the lack of resources evaluation in French LS by proposing a dataset of annotated sentences, collected from two sources. The first set of sentences was sampled from the French reference dataset ALECTOR (Gala et al., 2020): it includes sentences with complex words and candidates for substitution³. Complex words were detected through a reading experiment with dyslexic children. The second set of sentences comes from texts sampled from various French textbooks, ranging from grade 4 to grade 12 and covering three subjects: French literature, history, and sciences. Complex words have not been directly annotated in these sentences. However, they were read by various profiles of readers through an interface and reading times have been collected. Based on this information, we have manually identified seemingly complex words. Once the two sets of sentences were collected and the complex words were identified, we had to generate substitutes, and manually select them and classify them, as was described in the following subsections.

Generate and select candidates For each complex word, we produced synonyms that we annotated, using all our generation methods. The relevance of these synonyms in the context of the original sentence was then assessed by 3 expert linguists. They had to assign a score of 1 if the word is considered correct, otherwise 0. In this process, we applied the following guidelines : a word is considered synonymous as long as its replacement does not change the meaning of the sentence. To obtain a wide range of synonyms, we decided to accept hyperonyms and hyponyms – provided they fit the context – and to accept synonyms even if their register was different from that of the original word. This task is very complicated since there is no perfect synonymity and the validity of a candidate can therefore be perceived differently from

²<https://github.com/hardik-vala/FreNetic>

³The same sentence can be found several times with different complex words

one annotators to another one. However, thanks to the annotation guidelines and a discussion session between the annotators to discuss the criteria, the inter-rater agreement (Fleiss’ K) between the three annotators, computed on a sample of 500 candidates, is 0.638, which corresponds to a substantial agreement (Artstein and Poesio, 2008).

Ranking of substitutes Finally, for this last annotation, we resorted to on 20 non expert annotators aged from 20 to 57 years, whose native language is French. They had to rank the synonyms validated at the previous step by reading ease. To that end, we used the online *LimeSurvey* tool⁴ to deliver 2 different questionnaires of 25 items. The survey is presented as follows: each question includes a target sentence and the complex word in bold, as well as a list of synonyms in the left column. The task of the participants is to drag all the synonyms into the right-hand column to rank them, the top word being considered the most difficult.

Once all annotations have been collected, we proceed to average all annotators: in the same way as Specia et al. (2012), we assign each substitution a score based on the average of the scores assigned to it.

4.2 Evaluation Metrics

To compare our different methods of generation and selection, we used the following metrics as described in Paetzold and Specia (2016a): potential, precision, recall and F1. For the evaluation of ranking methods, we also employ the metrics trunk-at-i and recall-at-i as mentioned in Paetzold and Specia (2016a).

5 Results

This section presents the results for each step of simplification and compares the different methods proposed in *FrenLyS*. In view of the absence of comparable work for French, we put our results into perspective with those of Paetzold and Specia (2017c) for English and Qiang et al. (2021) for Chinese.

5.1 Substitution Generation

As we can see in Table 1, *FastText* generator, a method based on non contextual embeddings, slightly outperforms *Resyf* generator, based on a dictionary (F1 is 0.25 vs. 0.23). *Resyf* has a higher

potential and recall, but suffers from a lack in precision, which is due to the fact that no sense disambiguation is carried out in synonym selection. In contrast, *FastText* reject candidates that do not correspond to the most frequent meaning of a form (as *FastText* computes only one vector per form, the most frequent sense has the largest influence on it). *CamemBERT* is clearly the less efficient technique. It is not a complete surprise, as it can generate words that fits the context, but are not valid synonyms of the complex word.

FrenLys				
	Pot.	Prec.	Rec.	F1
Resyf	0.63	0.20	0.28	0.23
FastText	0.59	0.23	0.27	0.25
CamemBERT	0.45	0.13	0.18	0.15
CamemBERT + Resyf	0.55	0.29	0.23	0.26
CamemBERT + FastText	0.43	0.18	0.16	0.17
Paetzold (2017)				
	Pot.	Prec.	Rec.	F1
Paetzold-NE	0.88	0.31	0.14	0.19
Qiang et al. (2021)				
	Pot.	Prec.	Rec.	F1
Hybrid	0.90	0.43	0.26	0.33

Table 1: SG FrenLys results

We therefore tried to combine the advantages of *CamemBERT* (suitability to the context) with those of *Resyf* and *FastText* (better synonym generation). Considering the union between *CamemBERT* and *FastText* seems to hurt the performance (0.17 in F1 instead of 0.25), whereas combining *CamemBERT* and *ReSyF* produces our best results (0.26 in F1). It seems that *ReSyF* selects valid, but not necessarily context-appropriate synonyms, which are filtered by BERT based on the context. Although not directly comparable, the F1 of our best method is in line with those of Paetzold and Specia (2017c) and Qiang et al. (2021). At the potential level, the difference observed could be explained by a variation in the number of synonyms produced by the generators: potential is correlated with the number of generated synonyms.

5.2 Substitution Selection

We apply each of our selection methods on the union of all generated synonyms. The results obtained are presented in Table 2.

Results clearly reveal the importance of selecting synonyms that share the same POS as the word to substitute. This allows our system to reach a F1 of 0.31 for the generation of synonyms. It is however surprising that the POS approach outperforms both

⁴<https://www.limesurvey.org/fr/>

FrenLys (all generators)				
	Pot.	Prec.	Rec.	F1
POS	0.66	0.29	0.34	0.31
FastText Sentence	0.59	0.27	0.26	0.27
FastText Word	0.58	0.27	0.26	0.27
Paetzold (2017)				
	Pot.	Prec.	Rec.	F1
Paetzold-BR	0.97	0.23	0.26	0.24

Table 2: Benchmarking results for SS with substitutions generated by all generators

FastText methods by 0.04.

Once more, our results appears to be comparable with those of Paetzold and Specia (2017c), as our F1 is clearly higher, but we were not able to obtain such a high potential. It is interesting to notice that our selectors mostly improve precision.

5.3 Substitution Ranking

Finally, we tested our different ranking methods on the part of corpus that has been also annotated for the reading ease of synonyms. Results are displayed in Table 3.

FrenLys			
	TR-1	Rec-2	Rec-3
Lexique3	0.42	0.62	0.73
Frenetic	0.44	0.65	0.71
SVM	0.50	0.71	0.79
Paetzold (2017)			
	TR-1		
Neural	0.66		

Table 3: Benchmarking results for SR

Ranking candidates based on frequency remains a strong baseline, as the *Lexique3* ranker has a TRank-at-1 of 0.42 and a Recall-at-3 of 0.73. This means that, in 42% of our test sentences, using only the frequency allows to correctly predict the synonym defined to as the simplest by human judges (gold). Using the number of senses per words in addition to frequency does not bring much improvement, only 2% for TRank-at-1. In contrast, a much more sophisticated ranker using 21 word features, clearly improves performance, as it is able to select the easier synonyms in 50% of the cases. In this step, our results remain lower than those of Paetzold and Specia (2017c) in terms of TRank-at-1.

6 Conclusion

To conclude, we described the first tool for French lexical simplification that carries out three of the four classic LS steps. Our tool, *FrenLys*, will be

made available to the scientific community via a freely accessible web service⁵.

FrenLys includes five synonym generators, based on the two principal approaches in the field: using a resource and querying embeddings. Whereas Hmida et al. (2018) had concluded that using *ReSyF* as a resource was able to outperform the approach of Glavaš and Štajner (2015), we found that relying on *FastText* was more efficient. However, our best method combines a synonym database with *CamemBERT* as a way to filter inappropriate synonyms in context. These two sources bring information about the complex word semantic (*ReSyf*) and its context (*CamemBERT*), which comes close to the twofold strategy of Qiang et al. (2019). They indeed generate synonyms based on one sentence in which the complex is masked (contextual information) and the same sentence in the complex word is present in order to keep the semantic information conveyed by the complex word.

FrenLys offers three of them and the results showed that using a simple POS filter is sufficient to improve the F1 of our generators. Ranking synonyms can be done through three techniques, the best of which integrates 21 word characteristics into a SVM ranker. The results obtained for ranking seem lower than those of Paetzold and Specia (2017c). This could be due to variations in the test data, but maybe also to the use of a neural classifier. We plan to improve our ranking algorithm using a neural ranker in the future to investigate this issue.

Finally, in addition to the implementation of the first complete LS tool for French, this paper also proposes the first evaluation dataset for French LS. This dataset will be distributed through the same web site as the API⁶. We hope that the availability of both resources could help boosting current LS research in French, which lacks behind similar research for other European languages.

Acknowledgments

This project was partially funded by the "Direction de la langue française" from the Federation Wallonia-Brussels (AMesure project). We also want to thank Dr. Rémi Cardon for his in-depth reading and comments.

⁵<https://cental.uclouvain.be/frenlysAPI/>

⁶<https://cental.uclouvain.be/frenlysAPI/>

References

- Suha S Al-Thanyyan and Aqil M Azmi. 2021. Automated text simplification: A survey. *ACM Computing Surveys (CSUR)*, 54(2):1–36.
- Rodrigo Alarcon, Lourdes Moreno, Isabel Segura-Bedmar, and Paloma Martinez. 2019. Lexical simplification approach using easy-to-read resources. *Procesamiento de Lenguaje Natural*, 63:95–102.
- Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020. Data-driven sentence simplification: Survey and benchmark. *Computational Linguistics*, 46(1):135–187.
- Ron Artstein and Massimo Poesio. 2008. [Survey article: Inter-coder agreement for computational linguistics](#). *Computational Linguistics*, 34(4):555–596.
- Mokhtar Boumedyen Billami, Thomas François, and Núria Gala. 2018. [ReSyf: a French lexicon with ranked synonyms](#). In *Proceedings of 27th International Conference on Computational Linguistics (COLING 2018)*, pages 2570–2581.
- Or Biran, Samuel Brody, and Noémie Elhadad. 2011a. Putting it simply: a context-aware approach to lexical simplification. In *Proceedings of the 49th ACL*, pages 496–501.
- Or Biran, Samuel Brody, and Noémie Elhadad. 2011b. [Putting it simply: a context-aware approach to lexical simplification](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 496–501, Portland, Oregon, USA. Association for Computational Linguistics.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Stefan Bott, Luz Rello, Biljana Drndarevic, and Horacio Saggion. 2012. Can spanish be simpler? lexis: Lexical simplification for spanish. In *Proceedings of COLING 2012*, pages 357–374.
- Laetitia Brouwers, Delphine Bernhard, Anne-Laure Ligozat, and T. François. 2014. Syntactic sentence simplification for french. In *Proceedings of the 3rd International Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR 2014)*.
- Rémi Cardon. 2018. Approche lexicale de la simplification automatique de textes médicaux. In *Actes de la conférence Traitement Automatique de la Langue Naturelle, TALN 2018*, page 159.
- John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of english newspaper text to assist aphasic readers. In *Proceedings of AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 7–10.
- Jinying Chen, Jiaping Zheng, and Hong Yu. 2016. [Finding important terms for patients in their electronic health records: A learning-to-rank approach using expert annotations](#). *JMIR Medical Informatics*, 4:e40.
- Blandine Courtois. 2004. [Dictionnaires électroniques DELAF anglais et français](#). In Leclère Christian, Laporte Éric, Piot Mireille, and Silberstein Max, editors, *Lexique, Syntaxe et Lexique-Grammaire. Papers in Honour of Maurice Gross*, *Linguisticae Investigationes Supplementa 24*, pages 113–123. Amsterdam/Philadelphia : Benjamins. Incollection.
- Jan De Belder and Marie-Francine Moens. 2010. Text simplification for children. In *Proceedings of the SIGIR workshop on accessible search systems*, pages 19–26. ACM; New York.
- Jan De Belder and Marie-Francine Moens. 2012. A dataset for the evaluation of lexical simplification. In *Proceedings of the 13th CILing*.
- Louise Deléger and Pierre Zweigenbaum. 2009. Extracting lay paraphrases of specialized expressions from monolingual comparable medical corpora. In *Proceedings of the 2nd Workshop on Building and Using Comparable Corpora: from Parallel to Non-parallel Corpora (BUCC)*, pages 2–10.
- Sofiane Elguendouze. 2020. Simplification de textes: un état de l’art. In *Actes de TALN2020. Volume 3: Rencontre des Étudiants Chercheurs en Informatique pour le TAL*, pages 96–109. ATALA.
- Noemie Elhadad. 2006. Comprehending technical texts: Predicting and defining unfamiliar terms. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 2006:239–43.
- Thomas François, Mokhtar Billami, Núria Gala, and Delphine Bernhard. 2016. [Bleu, contusion, ecchymose : tri automatique de synonymes en fonction de leur difficulté de lecture et compréhension](#). In *JEP-TALN-RECITAL 2016*, volume 2 of *TALN*, pages 15–28, Paris, France.
- Thomas François, Adeline Müller, Eva Rolin, and Magali Norré. 2020. Amesure: A web platform to assist the clear writing of administrative texts. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 1–7.
- Nuria Gala, Thomas François, Ludivine Javourey-Drevet, and Johannes Christoph Ziegler. 2018. La simplification de textes, une aide à l’apprentissage de la lecture. *Langue française*, 3:123–131.

- Núria Gala, Anaïs Tack, Ludivine Javourey-Drevet, Thomas François, and Johannes Ziegler. 2020. Alecator: A parallel corpus of simplified french texts with alignments of misreadings by poor and dyslexic readers. In *Language Resources and Evaluation for Language Technologies (LREC)*, page 1353–1361.
- Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of ACL 2015: Volume 2: Short Papers*, pages 63–68.
- Sian Gooding and Ekaterina Kochmar. 2019. Complex word identification as a sequence labelling task. In *ACL*.
- Ralf Herbrich, Thore Graepel, and Klaus Obermayer. 2000. Large margin rank boundaries for ordinal regression. *Advances in Large Margin Classifiers*, 88.
- Firas Hmida, Mokhtar Boumedyen Billami, Thomas François, and Nuria Gala. 2018. [Assisted Lexical Simplification for French Native Children with Reading Difficulties](#). In *The Workshop of Automatic Text Adaptation, 11th International Conference on Natural Language Generation*, Tilbourg, Netherlands.
- Colby Horn, Cathryn Manduca, and David Kauchak. 2014a. [Learning a lexical simplifier using wikipedia](#). In *Proceedings of ACL 2014*, volume 2, pages 458–463.
- Colby Horn, Cathryn Manduca, and David Kauchak. 2014b. [Learning a lexical simplifier using wikipedia](#). In *ACL (2)*, pages 458–463.
- Tomoyuki Kajiwara and Kazuhide Yamamoto. 2015. [Evaluation dataset and system for Japanese lexical simplification](#). In *Proceedings of the ACL-IJCNLP 2015 Student Research Workshop*, pages 35–40, Beijing, China. Association for Computational Linguistics.
- Tomonori Kodaira, Tomoyuki Kajiwara, and Mamoru Komachi. 2016. [Controlled and balanced dataset for japanese lexical simplification](#). pages 1–7.
- Mathieu Lafourcade. 2007. Making people play for lexical acquisition with the jeuxdemots prototype. In *SNLP'07: 7th international symposium on natural language processing*, page 7.
- John Lee and Chak Yan Yeung. 2019. [Personalized substitution ranking for lexical simplification](#). In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 258–267, Tokyo, Japan. Association for Computational Linguistics.
- Bernard Lété, Liliane Sprenger-Charolles, and Pascale Colé. 2004. Manulex : A grade-level lexical database from French elementary-school readers. *Behavior Research Methods, Instruments and Computers*, 36:156–166.
- Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villenote de la Clergerie, Djamel Seddah, and Benoît Sagot. 2020. Camembert: a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.
- George A. Miller. 1995. [Wordnet: A lexical database for english](#). *Commun. ACM*, 38(11):39–41.
- Eric Mutabazi and Nathanaël Wallenhorst. 2020. Une citoyenneté de seconde classe ? n’ayons pas peur des mots ! *Zeitschrift für Bildungsforschung*.
- Boris New. 2006. Lexique 3: Une nouvelle base de données lexicales. *TALN Conference*, pages 892–900.
- Bernardo Pereira Nunes, Ricardo Kawase, Patrick Siehdnel, Marco A Casanova, and Stefan Dietze. 2013. As simple as it gets—a sentence simplifier for different learning levels and contexts. In *2013 IEEE 13th international conference on advanced learning technologies*, pages 128–132. IEEE.
- Gustavo H. Paetzold and Lucia Specia. 2013. Text simplification as tree transduction. In *STIL*.
- Gustavo H. Paetzold and Lucia Specia. 2015. Lexenstein: A framework for lexical simplification. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 85–90.
- Gustavo H. Paetzold and Lucia Specia. 2016a. Benchmarking lexical simplification systems. In *LREC*.
- Gustavo H. Paetzold and Lucia Specia. 2016b. [NNSeval: Evaluating Lexical Simplification for Non-Natives](#). [Http://ghpaetzold.github.io/data/NNSeval.zip](http://ghpaetzold.github.io/data/NNSeval.zip).
- Gustavo H. Paetzold and Lucia Specia. 2016c. [Se-meval 2016 task 11: Complex word identification](#). pages 560–569.
- Gustavo H. Paetzold and Lucia Specia. 2016d. Unsupervised lexical simplification for non-native speakers. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI’16*, page 3761–3767. AAAI Press.
- Gustavo H. Paetzold and Lucia Specia. 2017a. Lexical simplification with neural ranking. In *Proceedings of EACL 2017: Volume 2, Short Papers*, pages 34–40.
- Gustavo H. Paetzold and Lucia Specia. 2017b. A survey on lexical simplification.
- Gustavo H. Paetzold and Lucia Specia. 2017c. A survey on lexical simplification. *Journal of Artificial Intelligence Research*, 60:549–593.
- Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2019. [A simple bert-based approach for lexical simplification](#).

- Jipeng Qiang, Xinyu Lu, Yun Li, Yun-Hao Yuan, and Xindong Wu. 2021. Chinese lexical simplification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.
- Sadaf Abdul Rauf, Anne-Laure Ligozat, François Yvon, Gabriel Illouz, and Thierry Hamon. 2020. Simplification automatique de texte dans un contexte de faibles ressources. In *Actes de TALN2020. Volume 2*, pages 332–341. ATALA.
- Horacio Saggion. 2017. Automatic text simplification. *Synthesis Lectures on Human Language Technologies*, 10(1):1–137.
- Benoît Sagot and Darja Fiser. 2008. Construction d’un wordnet libre du français à partir de ressources multilingues.
- Violeta Seretan. 2012. Acquisition of syntactic simplification rules for french. In *Proceedings of the eight international conference on language resources and evaluation (LREC’12)*, pages 4019–4026.
- Matthew Shardlow. 2013. A comparison of techniques to automatically identify complex words. In *Proceedings of the ACL Student Research Workshop*, pages 103–109.
- Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70.
- Advait Siddharthan. 2014. A survey of research on text simplification. *ITL-International Journal of Applied Linguistics*, 165(2):259–298.
- Lucia Specia, Sujay K. Jauhar, and Rada Mihalcea. 2012. Semeval-2012 task 1: English lexical simplification. In *Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 347–355.
- Rebecca Thomas and Sven Anderson. 2012. Wordnet-based lexical simplification of a document. In *KONVENS*, pages 80–88.
- Rodrigo Wilkens and Amalia Todirascu. 2020. Un corpus d’évaluation pour un système de simplification discursive. In *Actes de TALN 2020. Volume 2*, pages 361–369. ATALA.
- Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo H Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A report on the complex word identification shared task 2018. *arXiv preprint arXiv:1804.09132*.