TEXT SIMPLIFICATION

Tom Wysocki
IDMC
Université de Lorraine
Nancy
tom.wysocki8@etu.univ-lorraine.fr

Peter Pribil
IDMC
Université de Lorraine
Nancy
peter.pribil7@etu.univ-lorraine.fr

Morgan Ruiz-Huidobro
IDMC
Université de Lorraine
Nancy
morgan.ruiz-huidobro1@etu.univ-lorraine.fr

Margot Guettier
IDMC
Université de Lorraine
Nancy
margot.guettier3@etu.univ-lorraine.fr

February 7, 2022

ABSTRACT

Text simplification consists of the simplification in grammar, structure, lexicon ,and prose of an input text. Nowadays, most text simplification systems use machine learning or deep learning methods. These methods require a large amount of annotated data for training. Such data is available for English, but for other languages it can be limited. A rule-based method would allow a possibility of simplification even for low resources languages. In this document, we will present our rule-based system for English text simplification.

***K*eywords** Text Simplification · Rule-based method · Syntactic Simplification · Lexical Simplification

1 Introduction

In the language, there exist different ways to simplify a text, depending on its topic, the size of the text, the size of the sentences. Text simplification can benefit a large public, from people with learning disabilities, new language learners, to people with neuro-linguistic disorders. Therefore, to the general public, it is a useful process to make a text intelligible. Moreover, this process is helpful in natural language processing (NLP) for some domains of research such as information extraction, parsing, machine translation, and semantic role labelling. Text simplification can be applied as a pre-processing step for different tasks. It allows reducing the amount of data requested by these systems.

The recent systems of text simplification use machine learning or deep learning methods. These methods are extremely useful to identify features that complicate documents. The downside of these systems is the training, it relies heavily on an input of data. Therefore, adapting the system to low-resourced languages will produce inferior results. Moreover, most of the time the existing models consider only one aspect of simplification, such as passive to active or relative clauses, whereas we want to assemble several levels of simplification.

The goal of the project was to create a rule-based text simplification system for English. We chose this method for multiple reasons. The first one was by curiosity, to see if we could obtain competitive results compared to current systems. The second one was in order to control more precisely the syntactical modifications we wanted to implement. At last, we had the idea that if the system works it could be adapted to languages with fewer data available.
Our project can be explained in three different parts. Firstly we worked on syntactic simplification, then, we focused on lexical simplification. Afterward, we created an interface for the system.

2 Background and Related Work

The concept of text simplification is not new in the research domain. A lot of papers detail how sentences are built. Some others study which patterns are more complex than others. Articles also show the purpose of simplified text structures and describe automatic methods. In this section, we present the theoretical framework regarding text simplification and study how this thematic has been approached in the literature.

2.1 Neurolinguistics

We started out our search for simplification by asking the question : What is actually causing difficulties in sentence comprehension from a syntactic point of view ?
Among the most common reactions, to name a few, are sentence length, multiply embedded sentences, passive phrases. However, is this justified from a neurological perspective, or is just a guess that seems rational ?
Can the findings in neurolinguistics be applied to regular humans, or only to people with speech disorders ?
Also, how much will it ultimately affect the reader if a text is simplified?
We have found little, but fMRI studies have shown higher activation in certain brain regions for different syntactic structures. These had a positive influence on our motivation to go deeper into the simplification process.

Linguistics studies that research language comprehension for children, neuro-divergent people, and new language learners seem to agree on some points. Passive voice is more complex than the active voice to understand. Subject relative clauses (SR) are often easier to process than object relatives (OR). Longer sentences are more difficult to process than a shorter one because it rely on memory retrieval.

2.2 Lexical Simplification

The first system we found for lexical simplification used Bert's masked language model. The article[1] written by Armand Olivares, describes the process. Firstly they used a model that identifies a complex word. Then BERT's task masked language modeling (MLM) predicts the token needing replacement in a sentence. This system factor in the left and right context. After that the best candidate is selected based on Zipf values, meaning the most frequent word is chosen.
We decided to try a similar approach with our project but instead of Bert, we wanted to use the lexical database, WordNet[2, 3]. The amount of data in WordNet seemed quite interesting to use for lexical simplification, especially the lexical relation between words, such as synonyms and hypernym.

3 Syntactic simplification

In this section, syntactic structures that are known to have a high cognitive cost to process are introduced, and the simplification implemented are described through examples. The syntactic simplification is processed in three steps:

- Passive to active conversion
- Relative clause linearization
- Prepositional Phrase Removal

3.1 Passive to Active Conversion

To carry out the conversion from passive voice to active voice, we used an already implemented system[1]. However, we detected some issues in particular with tenses and person concordance during the conversion. The modifications made allowed us to improve the results (Table 1).

3.2 Relative Clause Linearization

The second syntactic modification we have carried out is the linearization of the relative sentences. It is a matter of removing the relative sentence when possible and reintroducing the complements directly before or after the referent of the relative sentence. First of all, we have to detect such structures. At first, we wanted to use the dependency relations

[1]https://github.com/DanManN/pass2act

	Output sentence	
Input Sentence	Original System	Modify System
The dog is kept by him	He **kept** the dog	He keeps the dog
The dog is being kept by me	I **is** keeping the dog	I am keeping the dog
The dog has been kept by her	She **has** kept the dog	She has kept the dog

Table 1: passive to active conversion

to find relations like *advcl* or *relcl* which indicate the presence of relative. However, it turned out that some errors of the parser used (Spacy) disturbed this detection. As you can see in 1 between 1aand 1b only the verb in the relative clause changes and the dependency relation is not the same.

ccomp

(1) a. She lives in New York, which she hates.

advcl

b. She lives in New York, which she hates.

Since dependency relations lead to many parser errors, we decided to first use the detection of relative markers such as *that* or *which*. To detect these markers we use Part of Speech tags. If none of the markers are present then we check the dependency relations. Once the relative clauses are detected we have to determine how to process them. Not all relative phrases are essential to the meaning of the sentence, so referring to [4] we first decided to remove the non-restrictive relatives that are in apposition 2.

(2) a. The play**, which debuted last week,** has been sold out every night.
b. The play has been sold out every night.

The second type of relative is the one that introduces only one or more adjectives. For this, we retrieve the adjective and reintroduced it before the noun it qualifies3.

(3) a. I see a man **that is tall**.
b. I see a **tall** man.

When the modifier introduces a verb we check the governor of the verb. If it is another verb we reintroduce it before its governor 4, if it modifies a noun we leave the sentence as it is 5.

(4) a. She lives in New York, **which she hates**.
b. She **hates to** live in New York.

(5) a. I love the apple **that I eat**.
b. I love the apple that I eat.

In some cases we have chosen to separate the sentence into several sentences instead of reintroducing the elements in the sentence. This is how we process for the cases of recursions 6.

(6) a. The cat **that eats the mouse** <u>that is white</u> is big
b. The cat is big. The cat **eats the <u>white</u> mouse**.

The last case of relative that we treat for the moment is when there is no relative marker. We have to first use the dependency relations and introduce the recurrence marker. Finally, this sentence is treated as described above. In 7 you can see the three steps : detection 7a, marker introduction 7b and linearization 7c.

(7) a. The treasure the pirates found **was empty**.
b. The pirates found the treasure **that** was empty.
c. The pirates found the **empty** treasure.

The rules implemented take into account the most common configurations. Some structures with ditransitive verbs are not yet handled by this system as well as conditional structures.

3.3 Prepositional Phrase Removal

The third simplification process dealt with the removal of the prepositional phrases (PP).
Nowhere in our research in neurolinguistics have we found indicators that these kinds of phrases specifically could potentially be difficult to comprehend. They only increase the overall length of the sentences, which by itself makes comprehension harder.

With this, and along with another hypothesis, which is prepositional phrases being completely optional, unless the argument of the verb specifically asks for a PP, we decided to get rid of them in an aggressive manner.
Soon after the first attempts, underestimation turned into fear.
The task turned out to be very complex, and not only from a syntactic point of view, which we will discuss in the following section.

This being said, let's discuss what prepositional phrases are.

Prepositional phrases usually modify verbs or nouns.

(8) a. Peter decided to hang himself **with a rope**.
b. He couldn't figure out which one to buy **at the store**.
c. A guy **with a moustache** helped him choose.

The bold text represents the prepositional phrases in the sentences.
In 8a it is a verb-modifying prepositional phrase, while in 8b and 8c they are noun-modifying prepositional phrases.

In hopes of finding these bits in our input text, we use Spacy's dependency parser and select those with PPs in them.
Once we have a match, we check the root: if it is a copula, we skip to the next sentence with PPs. This lead us to save sentences like "He was on time" from oversimplifying.
Once we have a PP with a ROOT that is not a copula, and the type of preposition is not "to", "of", or "out", we save the index of the words in each PP.
These three were frequently misclassified, and they were used with transitive verbs too often, and that's something we couldn't figure out how to deal with.
The next step is to set the threshold for simplification, and then take the PPs out of the original sentence based on their indices that we saved in the previous step. Why we're doing it like this? It's messy to deal with Spacy tokens.
What is the threshold? If The number of tokens in a prepositional phrase is above the threshold that is set to 5 (which can optionally be changed), keep it, otherwise delete it.

3.4 Limitations For PP

There are three major limitations to proper PP removal.
The first one is deciding if a verb is transitive or not. Without this information, we may oversimplify to the extent where the sentence becomes ungrammatical. There are certain ways to see if a verb has a transitive attribute (Verbnet), but even with that, we are given too many different contexts where the verb might be transitive, or not.
The second problem is parsing. Too many multi-word expressions are being misclassified (e.g.: "step-by-step"). This is solvable by providing a list of known MWEs, but that is not necessarily a delicate approach.
The third problem is a bit more complex, and it's about contextual information. We have seen a great number of examples where the PP was not a necessary argument of any of the verbs, yet it was needed:

- ungrammatical: "Shaun loses his balance and the barrel **in which** he has been floating careens over." –> "Shaun loses his balance and the barrel he has been floating careens over."
- loses topic: "If one of the two events represents the sending of a signal from one location and the second event represents the reception of the same signal **at another location**. . . " –> "If one of the two events represents the sending of a signal and the second event represents the reception of the same signal. . . "

- changes meaning: She came out **for the encore** and performed Rock Lobster with the band. –> She came out and performed Rock Lobster.

Syntactically optional PPs become a necessity in certain contexts, and this is an interesting and challenging problem.

4 Lexical Simplification

Lexical simplification was another important part of the project. It consists in replacing words judged complex in the sentences. In this section we present the steps followed to get a functional application for lexical simplification. Firstly we will describe how we select the word to replace. Secondly, we will explain how we select replacement words using the lexical database of semantic relations WordNet[2, 3].

4.1 Complex word identification (CWI)

The first step of the system was to identify the noun classified as complex. For that, we needed to tokenize the sentence into words and to recognize nouns. We take into account simple word and multi-word-expression during tokenization. The nltk multi-word-expression tokenizer needed an input of multi-word tokens. These tokens were created based on the list of synsets given by WordNet. We decided to consider word placed after an adjective and a comma as simple word. For that we used nltk word tokenizer.

Once the word was selected we needed to judge the complexity. To consider a word as complex we check its frequency in a corpus. It is computed based on Zipf frequency.

The Zipf scale is a logarithmic frequency scale proposed by Marc Brysbaert[5]. The python module wordfreq[2] compute this score based on the Exquisite Corpus. The Data come from various domains for the English version (Wikipedia, subtitle, news, book, web, Twitter, Reddit). A frequency is hard to interpret since it depends on the corpus size. Therefore, the goal of the Zipf scale is to map word frequencies to small positive numbers to make it understandable. A word rates as x on the Zipf scale when it occurs 10^x times per billion words. With this corpus, reasonable words have a score between 0 and 8.
For the system we decided to choose a threshold of 3.0, which mean that word with a Zipf inferior to that value are considered as difficult to understand. A value inferior to 3.0 can be interpreted as a word occurring less than once per million words.

4.2 Replace Words

After detecting complex words, we had to find simpler words with a close meaning and select the best one. This part of the system relies heavily on WordNet which is a lexical database for English. Words are grouped with cognitive synonyms into synsets. Each set is interconnected with other concepts through conceptual-semantic and lexical relations.
The first step of our system was to checks the WordNet synsets of a word[3] and store the possible hypernyms. The hypernyms are at a depth level of 4 to avoid selecting a word with a meaning far from the original context. We have chosen to go so high up in the tree because some words were entangled in many synsets before the general meaning. For example, the multi-word "golden retriever" passes by "retriever" and "sporting dog" before finding the word "dog". Then, once the possible replacement was selected, we computed the zipf frequency of each one and we kept the one with the biggest score. Finally, current words are replaced by the selected one and the sentence is entirely rebuilt.

We have also implemented some small adjustments to prevent any grammatical errors. The first one was to adapt the determiner if there was one, change “a" to “an" and vice versa. The second one was to consider the number agreement, and put the word in its plural form if it’s needed. We did this transformation thanks to the “inflect" python library. It’s important to notice that the module used is not perfect and creates some errors.

4.3 Limitations

The function runs well the way we have implemented it, but they are still some issues, and the methods we used for lexical simplification have some limitations. The most crucial one we encountered was the lack of consideration for the

[2]https://pypi.org/project/wordfreq/
[3]We put the word in its singular form to search in WordNet

context. Indeed, since we process word by word, we do not consider the context before or after, and such a feature could be really helpful when looking for a synonym to the current word. A great example of this is homonyms. Furthermore, in some context, some words don't need simplification even if the system judge them as complex. For example in the sentence "The owner has two dogs, a great dane and a golden retriever." our system will simplify it as "The owner has two dogs, a dog and a dog.". In this utterance we can clearly see an oversimplification.
To summarize, the only way to resolve the sort of issues encountered by our lexical simplification is to change the method. The new approach should take into account the context. We talked about it in the beginning but creating a system relying on Bert could be a solution.

5 Evaluation

To evaluate the performances of our simplifications we processed to an automatic evaluation based on metrics, and a manual evaluation based on the appreciation of readers. The aligned Wikipedia Data Sets [6] is used for both automatic and manual evaluation.

5.1 Metrics for Automatic Evaluation

The evaluation of the text simplification system consists in determining if the output of our system (the simplified sentence) is indeed simpler while keeping a structure that respects the syntax of the language and sufficient information so that the reader is able to have the important information related to the document. The evaluation is done in two steps : first, an automatic part using readability measures. Such measures take into account the difficulty of the words and the length of the sentences. More than one of these measures exist, we have decided to use three of them which work on a similar system but whose computation of word difficulty is different :

- Gunning FOG Formula : a difficult word is a word with 3 or more syllables.

$$0.4[(\frac{words}{sentences}) + 100 * (\frac{complexwords}{words})]$$

- Coleman–Liau index : difficulty according to number of character per word

$$CLI = 0.0588 * (\text{letter per 100 words}) - 0.296 * (\text{sentence per 100 words}) - 15.8$$

- Dale–Chall readability formula : use a list of 3000 common words to determine if a word is difficult.

$$0.1579 \left(\frac{\text{difficult words}}{\text{words}} \times 100\right) + 0.0496 \left(\frac{\text{words}}{\text{sentences}}\right)$$

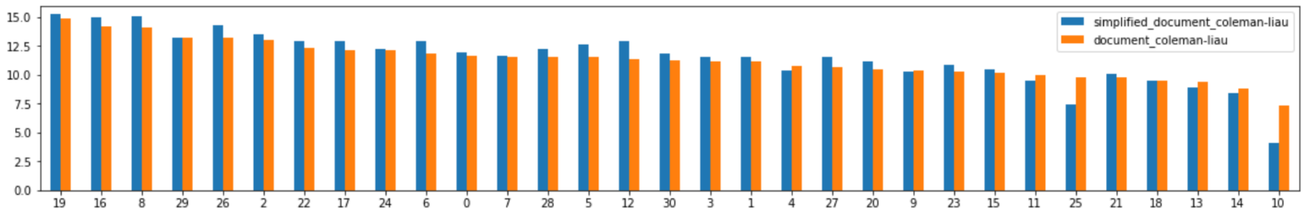
Among these three scores the Dale-Chall is a priori the most relevant one for qualify the lexical simplification. Indeed, as explain in section 4, in order to simplify the vocabulary the simplification method use frequency of the word and its hypernyms, and this score use a list of common/frequent words.

As the system modify the grammatical structure of sentences, the evaluation needs to qualify the new structure. And finally, the most important part is to determine if the simplify sentence make the reading and comprehension easier while keeping the most important information of the original text. To carry out this part of the evaluation the human validation is required.

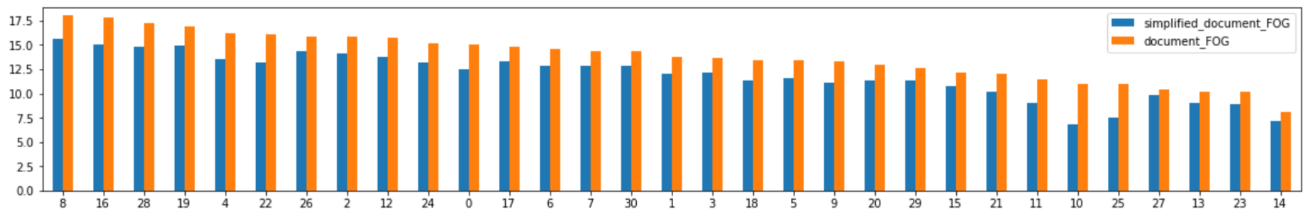
5.2 Results for Automatic Evaluation

For the automatic evaluation we use a sample of 31 documents. The sentences were simplified by the system and then compared the three scores for original and simplify sentences. Each sentences comes from different topics and thus the level of complexity can still be high for some of them.
The figure 1 shows that the results obtained by the readability metrics do not allow us to estimate how much the documents have been simplified. The parameters taken into account in these scores can explain these bad results. Indeed, the linearization of relative clauses causes in some cases sentence splitting and a duplication of words. If the duplicated words are considered as complicated, then the results will be worse, instead of having one occurrence of a complicated word we get two. On the other hand, the Coleman-Liau index takes into account the average length for 100 words, so when we delete propositional sentences, short words are deleted, increasing the average length. Concerning the gunning Fog the results show more clearly the simplification, because here only the words having a length of 3syllable or more are considered as difficult, this metric is thus less impacted by the suppression of the short words. The table 2 shows us

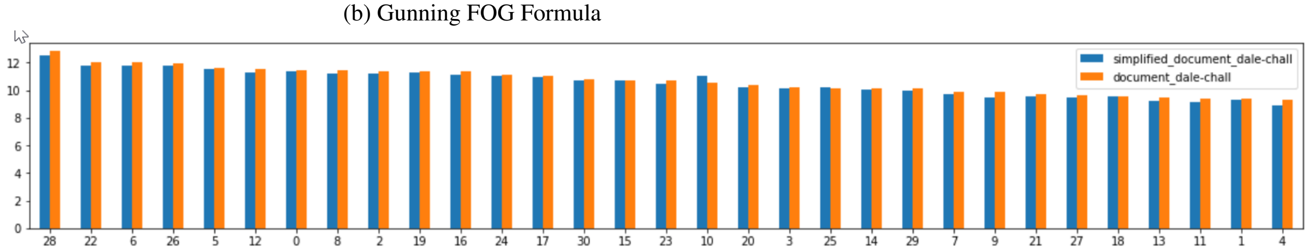
that on 31 documents the number of sentences tends to increase and the number of word to decrease what means shorter sentences that are potentially simpler to treat.



(a) Coleman–Liau index



(b) Gunning FOG Formula



(c) Dale–Chall readability formula

Figure 1: Result for 31 documents

	Original	Simplify
Number of sentences	2951	3296
Number of tokens	77598	68956
Average sentence length	26.29	20.92
Average token length	4.42	4.35

Table 2: Measure for original and simplify sentences

5.3 Manual evaluation

For manual evaluation, a sample of 100 sentences of the Wikipedia Datasets is used. The forms were divided in 5 with 20 pairs of original and simplified sentence each. People were asked to rate if the sentence is grammatical and to compare original and simplify version to qualify the quality of the simplification. Then they were asked if the simplified version is grammatically correct and tell what the simplification did : cause an ungrammatical sentence, simplify a bit, too much or the way they like.
We have obtained around 30 answers among the 5 forms. The results show that for 78% of those who completed the forms found grammatically correct the simplified sentence. This can be explained that most of the time the sentence is still readable after we remove some part of it, such as prepositional phrases or relative clauses. However, that means also that simplification caused grammatical error(s) in the process which is not desired. The following section illustrate some results we get from the forms.

5.4 Results for Manual Evaluation

Those results brought us different feedback concerning our simplification system, not only on the grammatical aspect but also on the appreciation of the simplification process. In fact, some sentences are sometimes considered as too much simplified by the audience, with too much elements removed. While others find the same sentence simplified as they want. This can't be considered as an issue since it's more a personal aspect.

The table 3 shows a sample of relevant results we reach to obtain with our system, the second table 4 shows some recurring errors that occurs sometimes on sentences.

Input sentence	Output sentence	Explanation
Despite being unhappy with the franchise tag, Suggs reported to offseason minicamps to practice with the team.	Despite being unhappy, Suggs reported to offseason minicamps to practice.	Remove the prepositional phrase in the first part and second part of the sentence
It's pretty cold out today, you should really wear a titfer.	It 's pretty cold out today, you should really wear a hat.	Apply lexical simplification on the word "titfer"
The TTC also runs Wheel-Trans, a paratransit service for the physically disabled with special low-floor buses designed to accommodate wheelchairs and to make boarding easier for ambulatory customers with limited mobility.	The TTC also runs Wheel-Trans, a paratransit service for the physically disabled with special low-floor buses designed to accommodate wheelchairs and to make boarding easier.	Reduce sentence length by removing a prepositional phrase
Despite the two claiming that the rift had been healed after Bellamy scored a last-minute winner in a 4-3 victory over Manchester City in late October, it was announced on 23 January 2005 that the player had been omitted from the team for a Premier League match with Arsenal at Highbury stadium, a game that ended 1-0 in the Gunners ' favor.	Despite the two claiming that the rift had been healed after Bellamy scored a last-minute winner in a 4-3 victory, it was announced that the player had been omitted for a Premier League match, a game that ended 1-0 in the Gunners ' favor.	Remove some prepositional phrases to reduce the sentence length

Table 3: Text simplification functional use cases

After a thorough evaluation of the feedback, we share some of the issues we considered to be interesting for further investigation :

- **Deceptive meaning** : Incorrect meaning while remaining grammatical
  - "from 15 cents for 39 pages total to 25 cents for 52 pages." →
  - "from 15 cents for 39 pages total to 25 cents."
- **Introduced ambiguity**: "Coming out" associated with LGBTQ by itself.
  - "She came out for the encore and performed Rock Lobster with the band." →
  - "She came out and performed Rock Lobster."
- **Topic underrepresentation** : Topic-wise crucial information is lost.
  - "Carter was not well received at Oyo, and the Egba chiefs advised him not to interfere with slavery. . . " →
  - "Carter was not well received, and the Egba chiefs advised him not to interfere. . . "

Plenty of mistakes came from incorrect parsing, but we won't be taking up space with them. That's a different difficult project on its own. Neither will we mention ungrammaticalities or nuances that are caused by lack of information on verb types. They are also extremely interesting, but we focus on what's working, and why it's not working the way we would like it to work.

Input sentence	Output sentence	Explanation
In 1962 Reynolds secured a guest appearance on Perry Mason in " The Case of the Counterfeit Crank ".	1962 Reynolds secured a guest appearance in " The Case of the Counterfeit Crank ".	The parse took the year “1962” as part of Reynolds and remove by mistake “in”
His only injury was from biting his tongue.	His only injury was.	In this case the part starting by “from” as been taken as a prepositional phrase and removed
To the very end of its existence, the Nazi Party claimed to respect the traditional government of Germany and, to that that end, local and state governments were allowed to exist side-by-side with regional Nazi leaders.	To the very end of its existence, the Nazi Party claimed to respect the traditional government of Germany and, to that that end, local and state governments were allowed to exist side-	The word “side-by-side” has been cut since it has been considered as a prepositional phrase instead of a multi word expression
Bradfield plays guitar on both songs.	Bradfield plays guitar.	This case is grammaticaly correct, but the loss of information makes irrelevant the sentence and far from its initial meaning

Table 4: Text simplification failures

6 Software Interface

The final goal in our project was to create an software interface. Imagine that people using it online on a smooth interface is important also. We designed a web interface for the system (Figure 2). That way even people without any coding experience can enjoy our project. We use the Flask framework of python in order to create the visual part and link it to the code. As you can see on 2 the use is really intuitive. The user types the text he want to simplify on the left part, and our system propose the simplification of it in the right-hand box.

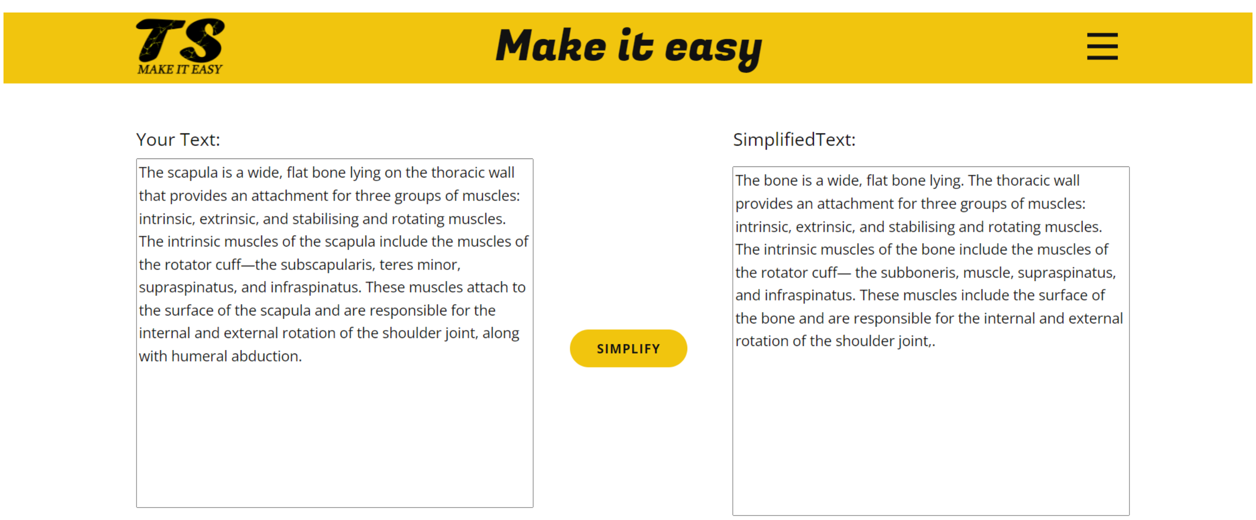


Figure 2: Application interface

7 Conclusion

In this paper, we have presented our work on text simplification using a rule-based method. We took a lot of time reading papers to know what are the expectations in terms of simplifications. The importance of each structure in a sentence, the different cases that occur but also to the target in the use of such a tool we want to build. After a long process and investigation, we decided to use a rule-based method.

We are overall satisfied with the outcome of our text simplification algorithm. What it lacks in accuracy, it makes up for explainability.
We have learned that dealing with sentence transformations that seem simple theoretically, is indeed quite challenging in practice. In addition, we learned the power of human evaluation and commenting, as it proved to be an invaluable source of knowledge for further investigation.

As a future work, we need to figure out a way to handle verb transitivity, and incorporate topic identification, so we get to keep relevant optional phrases.
We need to adjust the lexical simplification process to produce more realistic exchanges.
We also need to deal with sentence splitting in a way that is not necessarily complementizer or coordination splitting.

The results show that the rule-based methods have some limitations. The most recurrent one we have encountered during this project was its failure to consider the context.

Acknowledge

Thanks to all the fails and feedback we received, we have a clear sight of where to continue our work.

Thank you to our classmates for filling in the forms and give a valuable feedback on our algorithm.

References

[1] Armand Olivares. How to use bert for lexical simplification, 2020.

[2] George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[3] Christiane Fellbaum. Wordnet: An electronic lexical database. *Cambridge, MA: MIT Press*, 1998.

[4] Iustin Dornescu, Richard Evans, and Constantin Orasan. Relative clause extraction for syntactic simplification. 08 2014.

[5] W.J.B. Van Heuven, P. Mandera, E. Keuleers, and M. Brysbaert. Subtlex-uk: A new and improved word frequency database for british english. *Quarterly Journal of Experimental Psychology*, 67:1176–1190, 2014.

[6] David Kauchak. Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1537–1546, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[7] Shiwen Feng, Jennifer Legault, Long Yang, Junwei Zhu, Keqing Shao, and Yiming Yang. Differences in grammatical processing strategies for active and passive sentences: An fmri study. *Journal of Neurolinguistics*, 33:104–117, 2015. Language, Brain, and Gene: The Chinese Context.

[8] Ina Bornkessel, Stefan Zysset, Angela D. Friederici, D. Yves von Cramon, and Matthias Schlesewsky. Who did what to whom? the neural basis of argument hierarchies during language comprehension. *NeuroImage*, 26(1):221–233, 2005.

[9] Michal Ben-Shachar, Talma Hendler, Itamar Kahn, Dafna Ben Bashat, and Yosef Grodzinsky. The neural reality of syntactic transformations evidence from functional magnetic resonance imaging. *Psychological science*, 14:433–40, 10 2003.

[10] Marcel Just, Patricia Carpenter, Timothy Keller, William Eddy, and Keith Thulborn. Brain activation modulated by sentence comprehension. *Science (New York, N.Y.)*, 274:114–6, 11 1996.