ISEP_Presidency_University at MLSP 2024 Shared Task: Using GPT-3.5 to Generate Substitutes for Lexical Simplification

Benjamin Dutilleul[1], **Mathis Debaillon**[1], and **Sandeep Mathias**[2,3]

[1]Institut Supérieur d'électronique de Paris,
[2]Information Retrieval Lab, Department of Computer Science & Engineering
[3]Presidency University, Bangalore

Correspondence: sandeepalbert@presidencyuniversity.in

Abstract

Lexical substitute generation is a task where we generate substitutes for a given word to fit in the required context. It is one of the main steps for automatic lexical simplification. In this paper, we introduce an automatic lexical simplification system using the GPT-3 large language model. The system generates simplified candidate substitutions for complex words to aid readability and comprehension for the reader. The paper describes the system that we submitted for the Multilingual Lexical Simplification Pipeline Shared Task at the 2024 BEA Workshop. During the shared task, we experimented with Catalan, English, French, Italian, Portuguese, and German for the Lexical Simplification Shared Task. We achieved the best results in Catalan and Portuguese, and were runners-up in English, French and Italian. To further research in this domain, we also release our code upon acceptance of the paper[1].

1 Introduction

Test simplification is an important educational application. It aims to simplify text to make the generated simpler text easier for reading and comprehension by different readers who may be either young learners, people with language disabilities (Eg. aphasia), second-language learners, etc. A lot of the research done in the area of text simplification is split into mainly 2 parts, namely syntactic simplification and lexical simplification.

Syntactic simplification involves splitting the sentences into smaller sentences (Klerke et al., 2016). Lexical simplification, on the other hand, involves simplifying the text by replacing more complex words and phrases with simpler, and in context, synonyms (Shardlow, 2014).

The lexical simplification pipeline consists of multiple sub-tasks, (Shardlow, 2014) as shown in Figure 1. These subtasks are complex word identification (where we identify which word we have to consider for simplification), substitution generation (where we generate candidate synonyms for the given complex word), substitution selection (where we select the candidate synonyms which are contextually correct), and substitution ranking (where we rank the selected candidates from easiest to most complex).

With the advent of large language models (LLMs) like GPT-3, the potential for automating this task has increased significantly. These models, trained on vast amounts of text, have shown remarkable proficiency in understanding context and generating human-like text. Unlike pre-trained language models like BERT (Devlin et al., 2019), LLMs are significantly harder to fine-tune due to the massively larger number of parameters (BERT has about 110 million parameters, while GPT-3 has about 175 **billion** parameters). Because of this, we use GPT-3 using prompt-engineering, where we provide a prompt to the system to generate substitutes.

1.1 Organization of the Paper

The rest of the paper is organized as follows. We define the problem statement of our work in Section 2. Section 3 summarizes some of the recent related work in this domain. We discuss the different datasets used in Section 4. We describe our system in Section 5. Our results are reported and discussed in Section 6 and we conclude our paper and mention future work in Section 7.

2 Problem Statement

The Multilingual Lexical Simplification Pipeline (MLSP) Shared Task dealt with 2 problems. The first was Lexical Complexity Prediction (LCP). In this task, the participants had to develop a system where they were given a context and word in a

[1]The code for the paper is available at: https://github.com/lwsam/ISEP-LS

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 605–609

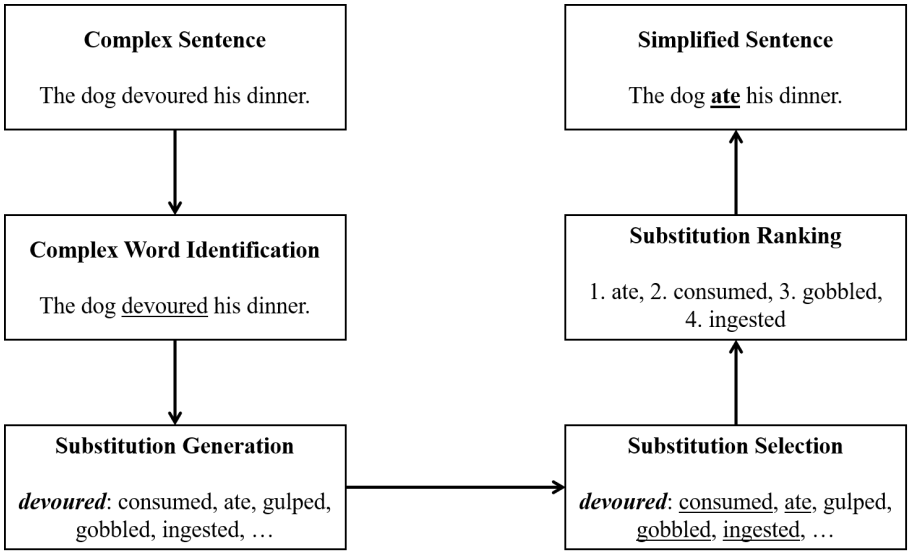

Figure 1: Lexical Simplification Pipeline showing the different tasks traditionally used in lexical simplification. In this example, we simplify a complex sentence ("The dog devoured his dinner.") to a simplified sentence ("The dog **ate** his dinner.").

given language, and they had to assess how easy / complex the word was[2]. This is similar to the SemEval 2021 Shared Task on Lexical Complexity Prediction (Shardlow et al., 2021).

The second problem was called Lexical Simplification (LS), where we are given an input context and complex word and we need to generate a **ranked list** of upto 10 simplifications in increasing order of complexity (i.e. from the simplest substitute to the most complex substitute). This is similar to the 2022 Text Simplification, Accessibility and Readability Shared Task on Lexical Simplification (Saggion et al., 2022).

Both these problem statements required participants to build systems in multiple languages, such as English, Catalan, French, German, Italian, Brazilian Portuguese, Spanish, Bengali, Sinhala, Filipino, and Japanese. In our paper, we mainly focus on the second task (lexical simplification) for the first 6 languages listed (English to Brazilian Portuguese). More details on the shared task are available inn the shared task report (Shardlow et al., 2024).

3 Related Work

There has been a number of shared tasks dealing with different aspects of the lexical simplification pipeline.

For complex word identification, one of the earliest shared tasks was held in 2016 (Paetzold and Specia, 2016a). The winners of that shared task used a system of soft voting with different "voters", where the voters are either lexicon based, threshold-based, or machine-learning assisted (Paetzold and Specia, 2016b). In 2018, another shared task on complex word identification was held as part of the BEA Workshop collocated with NAACL (Yimam et al., 2018). It had a monolingual track and multilingual tracks where the systems would be tested on German, French and Spanish. The winning team (Gooding and Kochmar, 2018), used a similar approach as Paetzold and Specia (2016b), but with a much wider range of features.

One of the challenges is trying to assess a score for how simple / complex a word is, given the context. This step is critical for complex word identification. In light of this, Shardlow et al. (2021) conducted a lexical complexity prediction task at SemEval 2021.

The advent of LLMs inspired a significant change in task specification. The 2022 TSAR Shared Task on Lexical Simplification (Saggion et al., 2022) had 3 languages - English, Spanish and Brazilian Portuguese. The participants in that shared task had to generate a set of substitutes for each language. While some systems such as Whistely et al. (2022) used a procedure of candidate generation (using pre-trained language models like BERT (Devlin et al., 2019)), cosine similar-

[2]We attempted to participate in this task as well, but due to some issues with the formatting in the output, we were unable to make a good submission by the shared task deadline.

ity and part-of-speech tagging as filters, the winning team (Aumiller and Gertz, 2022) used prompt-engineering on a large language model.

4 Datasets

Language	Test Set Size
English	570
Catalan	445
French	570
German	570
Italian	570
Portuguese	569

Table 1: Sizes of the testing dataset for each language.

For the shared task, participants were provided only with **trial** data. That is, a very few context-complex word pairs. Each language had a trial dataset of 30 context-complex word pairs[3]. Our systems were then evaluated on test sets of varying sizes. Table 1 shows the sizes of each language's testing dataset.

5 Experiment

5.1 System Used

For our system, we utilize Open AI's GPT 3.5 model[4]. We use a maximum of 256 tokens in the prompt with a frequency penalty of 0.5 and a presence penalty of 0.3.

The first step that we do is detect the language of the context. Based on the language chosen, we select a prompt for simplification. If no language is detected, then we default to the English prompt.

Once we detect the language, we next generate the prompt from a set of templates. We use 3 types of templates, similar to (Aumiller and Gertz, 2022).

5.2 Types of Prompts

Context-Free Prompt. This is a prompt that asks for synonyms of the complex word without providing any context. This tests the model's general knowledge of synonyms generation.

Context-Free Prompt. Template: "Give me ten simplified synonyms for the following word {complex word}". **Example:** "Give me ten simplified synonyms for the following word {**distraught**}"

Zero-Shot Prompt. This type of prompt provides the context and the complex word, and asks the LLM for simpler synonyms without any additional examples. This is used to gauge the model's ability to generate synonyms based solely on the given context and complex word.

Zero-shot Prompt. Template: "Context: {context} Question: Given the above context, list ten alternative words for {complex word} that are easier to understand. Answer:" **Example:** "Context: {*After Ron nearly dies drinking poisoned mead that was apparently intended for Professor Dumbledore, Hermione becomes so distraught that they end their feud for good.*} Question: Given the above context, list ten alternative words for {**distraught**} that are easier to understand. Answer:"

Single-shot Prompt. This is a prompt that includes one example of a complex word and its synonyms, followed by the target complex word. This aims to guide the model by showing an example of the desired output.

Single-shot Prompt. Template: "Question: Find ten easier words for **prerequisite**. Answer: 1. requirement 2. required 3. essential 4. need 5. precondition 6. prior 7. necessary 8. necessity 9. prior 10. prescribed. Question: Find ten easier words for {**complex word**}. Answer:"

Few-Shot Prompt. This is similar to the single-shot prompt, but with multiple examples provided to give the model a clearer understanding of the task.

5.3 Prompting the LLM

For each generated prompt, we send a request to the GPT-3.5 API. The predictions from GPT-3.5 are cleaned. Predictions from different prompts are aggregated and ranked and the top (at most) 10 synonyms are submitted as the output for our system.

5.4 Evaluation Metrics

We used the same evaluation metrics as given in the shared task. However, in Section 6, we report an **aggregate** of the evaluation metrics.

The different evaluation metrics used for automatic evaluation are:

- **MAP@K**. This metric uses an ordered list of gold-standard substitutes to compare the system output with. This metric takes into

[3]NOTE: The number of contexts for most of the languages are less than 30, as some contexts were repeated with different complex words.

[4]Model name - gpt-3.5-turbo-instruct-0914

English		Catalan		French	
System	**Performance**	**System**	**Performance**	**System**	
TMU-HIT	0.677	ISEP_PU	0.547	TMU-HIT	0.697
ISEP_PU	0.643	TMU-HIT	0.524	ISEP_PU	0.660
GMU	0.639	GMU	0.445	GMU	0.590
ANU	0.636	RETUYT-INCO	0.397	RETUYT-INCO	0.497
RETUYT-INCO	0.530	Archaeology	0.215	Archaeology	0.258
CocoNut	0.386	—	—	—	—
Archaeology	0.288	—	—	—	—
German		**Italian**		**Portuguese**	
System	**Performance**	**System**	**Performance**	**System**	**Performance**
TMU-HIT	0.626	TMU-HIT	0.673	ISEP_PU	0.571
GMU	0.548	ISEP_PU	0.635	TMU-HIT	0.551
RETUYT-INCO	0.413	GMU	0.607	RETUYT-INCO	0.379
ISEP_PU	0.257	RETUYT-INCO	0.225	Archaeology	0.230
Archaeology	0.142	Archaeology	0.225	—	—

Table 2: Results of our system compared with the best performances from all other systems based on the **mean** of all the evaluation metrics. Our system is highlighted in blue. Due to space constraints, we refer to it as "ISEP_PU".

account the ranking of each of the generated outputs. Here, $K = \{3, 5, 10\}$.

- **Accuracy@k@top1**. This is the percentage of instances, where, out of the top k outputs given by the system, at least one of them matches the top gold-standard substitute. Here, $k = \{1, 2, 3\}$.
- **Potential@k**. This is similar to the MAP@K metric, where we take $k = \{3, 5, 10\}$.

Based on the above metrics, we calculate our aggregate metric, **Performance**, which is the **arithmetic mean** of the other metrics.

6 Results and Analysis

We report the results of our experiments in Table 2. From the above table, we observe that we perform quite well compared to other systems, in almost all the languages except for German. We have achieved the best performances in Catalan and Brazilian Portuguese, as well as the second-best performances in English, French and Italian.

One of the challenges that we faced was in constructing the prompts for different languages. While the authors of the paper are L1 / fluent speakers of English and French, we needed the help of Google Translate to translate the prompts from English to other languages like German / Italian / Portuguese.

One of the challenges of using LLMs currently is that they are computationally intensive, requiring hundreds of GB of GPU power to fine-tune. Another challenge is that the current LLMs are focused on generating ranked substitutes irrespective of the target user. For example, young learners may have different requirements for simplification, as opposed to second-language learners, or people with reading disabilities. This can be tackled by modifying the prompts (especially the one-shot / few-shot prompts) to generate different simplifications based on the target user.

7 Conclusion and Future Work

Although we have performed reasonably well in the shared task for lexical simplification, we would like to extend our work for other languages which we were not able to participate in. Most of the other languages possess orthographic challenges because they do not use the Roman script, such as Bengali, Japanese, etc.

In the future, we would also like to focus on instruction tuning to improve the performance for personalizing the LLM for simplification. Currently, the predictions from the LLM are independent of the user. This means that a system built using this approach may generate the same output irresppective of the user the text should be simplified for. One method for resolving this is to utilize a user's cognitive information to try to perform complex word identification, as well as generate and rank candidate simplifications.

References

Dennis Aumiller and Michael Gertz. 2022. UniHD at TSAR-2022 shared task: Is compute all we need for lexical simplification? In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 251–258, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Sian Gooding and Ekaterina Kochmar. 2018. CAMB at CWI shared task 2018: Complex word identification with ensemble-based voting. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 184–194, New Orleans, Louisiana. Association for Computational Linguistics.

Sigrid Klerke, Yoav Goldberg, and Anders Søgaard. 2016. Improving sentence compression by learning to predict gaze. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1528–1533, San Diego, California. Association for Computational Linguistics.

Gustavo Paetzold and Lucia Specia. 2016a. SemEval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569, San Diego, California. Association for Computational Linguistics.

Gustavo Paetzold and Lucia Specia. 2016b. SV000gg at SemEval-2016 task 11: Heavy gauge complex word identification with system voting. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 969–974, San Diego, California. Association for Computational Linguistics.

Horacio Saggion, Sanja Štajner, Daniel Ferrés, Kim Cheng Sheang, Matthew Shardlow, Kai North, and Marcos Zampieri. 2022. Findings of the TSAR-2022 shared task on multilingual lexical simplification. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 271–283, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*.

Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Huelsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Peréz Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, Sanja Štajner, Marcos Zampieri, and Horacio Saggion. 2024. The BEA 2024 Shared Task on the Multilingual Lexical Simplification Pipeline. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*.

Matthew Shardlow, Richard Evans, Gustavo Henrique Paetzold, and Marcos Zampieri. 2021. SemEval-2021 task 1: Lexical complexity prediction. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1–16, Online. Association for Computational Linguistics.

Peniel Whistely, Sandeep Mathias, and Galiveeti Poornima. 2022. PresiUniv at TSAR-2022 shared task: Generation and ranking of simplification substitutes of complex words in multiple languages. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 213–217, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 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*, pages 66–78, New Orleans, Louisiana. Association for Computational Linguistics.