TMU-HIT at MLSP 2024: How Well Can GPT-4 Tackle Multilingual Lexical Simplification?

Taisei Enomoto[†*], **Hwichan Kim**[†*], **Tosho Hirasawa**[†], **Yoshinari Nagai**[†]
Ayako Sato[†], **Kyotaro Nakajima**[†], **Mamoru Komachi**[‡]
[†]Tokyo Metropolitan University, [‡]Hitotsubashi University
{enomoto-taisei@ed., kim-hwichan@ed., toshosan@, nagai-yoshinari@ed.,
sato-ayako@ed., nakajima-kyotaro@ed.}tmu.ac.jp, mamoru.komachi@hit-u.ac.jp

Abstract

Lexical simplification (LS) is a process of replacing complex words with simpler alternatives to help readers understand sentences seamlessly. This process is divided into two primary subtasks: assessing word complexities and replacing high-complexity words with simpler alternatives. Employing task-specific supervised data to train models is a prevalent strategy for addressing these subtasks. However, such approach cannot be employed for low-resource languages. Therefore, this paper introduces a multilingual LS pipeline system that does not rely on supervised data. Specifically, we have developed systems based on GPT-4 for each subtask. Our systems demonstrated top-class performance on both tasks in many languages. The results indicate that GPT-4 can effectively assess lexical complexity and simplify complex words in a multilingual context with high quality. The code used in our experiments is available at the following URL [1].

1 Introduction

The presence of unfamiliar words within a sentence can significantly impede its comprehension for readers. Such complex words may cause misunderstandings of the sentence's content or result in wasted time as readers may find themselves compelled to consult definitions of unfamiliar words. The development of a system capable of automatically simplifying complex words would enable readers to proceed without interruption. To achieve this, it is essential to first identify complex words and then replace them with more comprehensible alternatives. Numerous researchers have been undertaken focusing on each challenge, engaging in specialized endeavors known as Lexical Complexity Prediction (LCP) (Paetzold and Specia, 2016; Shardlow et al., 2021) and Lexical Simplification (LS) (McCarthy and Navigli, 2007; Specia et al., 2012; Saggion et al., 2022).

LCP is a task that assesses the complexity of a target word, i.e. its level of difficulty for understanding. Various methodologies have been proposed to tackle this task. A classical strategy is the frequency-based approach (Kajiwara and Komachi, 2018), which attributes higher complexity scores to words of lower frequency. Given the availability of supervised data, one viable option is to train a regression model to evaluate the word's complexity (Bani Yaseen et al., 2021; Pan et al., 2021). However, such abundant linguistic resources for supervised learning are scarce for many languages (Joshi et al., 2020). Therefore, there exists a need for an approach capable of determining lexical complexity without reliance on supervised data.

LS is a task that replaces a complex word with easier synonyms while maintaining the original meaning or information of the sentence. The LS pipeline comprises three primary components (North et al., 2023): substitute generation (SG), substitute selection (SS), and substitute ranking (SR). SG returns several candidate alternative words for the target word in a given sentence. SS then shifts through these alternatives, eliminating those unsuitable for the target word, such as more complex words or words with different parts of speech. SR sorts the candidate alternatives to prioritize words more suitable as alternatives, ensuring they appear higher in the ranking. Recent LS studies (Qiang et al., 2019; Przybyła and Shardlow, 2020) have often employed pre-trained language models, and systems using GPT-3.5 (Aumiller and Gertz, 2022) have demonstrated particularly effective performance.

This paper outlines our approach to the MLSP-2024 Shared Task (Shardlow et al., 2024a). Our system was based on GPT-4 and exclusively utilized the trial data provided by the organizers (Bott et al., 2024; North et al., 2024; Shardlow et al.,

[*]These two authors contributed equally to this work.
[1]https://github.com/tmu-nlp/GPT4MLSP

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 590–598

2024b), without incorporating any supplementary datasets. In the LCP task, our system achieved the highest Pearson's correlations in nine out of the ten languages. Additionally, in the LS task, our system exhibited the highest performance in seven out of the ten languages. The findings indicate that GPT-4 has capabilities of assessing lexical complexity and replacing complex words with simpler alternatives in a multilingual context with a high level of quality.

2 LCP Subtask

In Lexical Complexity Prediction (LCP), we assessed the complexities using GPT-4[2] based system. Specifically, we employed G-EVAL (Liu et al., 2023), a scoring framework using large language models with a chain-of-thoughts, for our system.

2.1 G-EVAL for LCP

We constructed a base prompt following G-EVAL. Specifically, we first manually designed a prompt that contains the definition of the LCP task and the desired evaluation criteria as follows:

> *You will be given a sentence and a word included in the sentence. Your task is to rate the word on one metric. Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.*
>
> *Evaluation Criteria:*
>
> *Complexity (0.0, 0.25, 0.5, 0.75, 1.0): the complexity of a word in terms of how difficult the word is to understand.*

Then, we added a line of "*Evaluation Steps:*" to the prompt and let GPT-4 generate the following chain-of-thoughts (CoT) automatically:

> *Evaluation steps:*
>
> *1. Read the sentence and word carefully to understand the context.*
>
> *2. Determine the complexity of the word based on the following criteria:*
>
> *- 0.0: The word is simple and easily understandable to most people.*
>
> *- 0.25: The word may have some complexity or be specific to a certain field, but can still be understood with some effort.*
>
> *- 0.5: The word is moderately complex and may require some background knowledge or explanation to understand fully.*
>
> *- 0.75: The word is quite complex and may be difficult to understand without significant knowledge or explanation.*
>
> *- 1.0: The word is extremely complex and likely only understood by experts or individuals with specialized knowledge.*
>
> *3. Assign a complexity rating to the word.*

We denote this prompt as P_{base}. We added a test example (sentence and target word) to P_{base} and let GPT-4 generate the complexities for the example n times. We used the average of those as the final complexity.

We had multiple options regarding the type of language to use for a prompt. Although the language of the test example is expected to be the most intuitive and effective, previous studies (Lin et al., 2022; Ahuja et al., 2023) demonstrated that English prompt achieves the best performance for most test languages. Furthermore, we manually and automatically translated P_{base} to Japanese and French, respectively, and compared performances of P_{base} and the translated prompt in each language using trial data. The Pearson's correlation of P_{base} and the translated prompt were 0.821 and 0.600 in Japanese 0.416 and 0.205 in French, respectively. Therefore, we used P_{base} regardless of languages.

2.2 Prompts to Specify Language and Role

In addition to P_{base}, we defined and added a prompt to specify the language of the test example. Specifically, we added "*Please assign a complexity rating to the* `LANG_NAME` *word*" to the end of P_{base} where `LANG_NAME` is a language name of a test example, such as *English*, *Japanese*, and *French*. We denote the prompt with the language as P_{lang}.

In our preliminary observation, the complexities generated by P_{base} distributed nearly 0.0 to 0.1, which means that almost all words are easy to understand for GPT-4. Furthermore, this distribution differed from that of the gold complexities as shown in Figure 1. One of the potential reasons is that GPT-4 is familiar with the target words unlike human annotators because it was pre-trained by massively data. To fill the gap between GPT-4

[2]We used `gpt-4-0613` following Liu et al. (2023) for LCP.

	Ca	En	Fil	Fr	De	It	Ja	Pr	Si	Es
P_{base}	**0.646**	0.733	0.462	0.416	**0.793**	0.615	**0.821**	0.836	**0.347**	0.641
P_{lang}	0.493	0.734	0.516	0.516	0.783	0.666	0.674	0.802	-0.077	**0.659**
P_{role}	0.470	**0.783**	0.513	0.513	0.740	0.537	0.794	**0.849**	0.292	0.654
$P_{\text{lang+role}}$	0.484	0.729	**0.595**	**0.595**	0.771	**0.672**	0.598	0.803	0.056	0.631

Table 1: Pearson correlations on trial datasets for each language. The best scores are indicated in bold.

	Pearson	Spearman	MAE	MSE	R2
Zero-shot (Run 1)	0.5609	0.5697	0.1771	0.0487	-0.3111
Three-shot (Run 2)	**0.6241**	**0.6215**	**0.1327**	**0.0280**	**0.2456**

Table 2: LCP results on the all language's test dataset. MAE and MSE denote Mean Absolute Error and Mean Squared Error.

and human annotators, we gave the role to GPT-4. Specifically, we added "*You are an individual without specialized knowledge or expertise in a specific area.*" to the first of P_{base}. We denote the prompt with the role as P_{role}.

We compared performances of P_{base}, P_{lang}, P_{role}, and $P_{\text{lang+role}}$, the prompt to which both of the language and role are added, per each language using trial data. Table 1 shows Pearson's correlations of each prompt per each language. The table indicates that the best prompts differ for each language.

2.3 Experiments

Experimental settings. We used the test datasets provided by Shardlow et al. (2024a)[3] for our evaluations. The datasets encompass those for ten languages, and a composite test dataset that amalgamates the individual datasets for all languages. For details about the languages and the size of each dataset, please refer to the Appendix.

For evaluation metrics, we employed both Pearson's and Spearman's correlations, Mean Absolute Error (MAE), Mean Squared Error (MSE), and R2 following Shardlow et al. (2021). We reported the performance of the composite test dataset.

We chose prompts for each language that achieved the highest Pearson's correlation in Table 1. We scored the complexities in zero- and three-shot settings.[4] In the three-shot setting, we randomly sampled three examples from the trial data.

Experimental results. Table 2 shows the result on the test set of all languages and indicates that the three-shot settings consistently outperform the zero-shot one. The findings indicate the importance of providing demonstration examples in LCP and suggest the possibility that performance will be enhanced by increasing the number of shots.

3 LS Subtask

In TSAR-2022 Shared Task (Saggion et al., 2022) of LS, the system using GPT-3.5 (Aumiller and Gertz, 2022) demonstrated a significant lead over other neural approaches such as those using mask language models. Following these findings, we employed a GPT-based method using the latest available GPT-4[5] for LS.

3.1 Substitution Generation

The Base system. We manually designed a prompt [6] that instructs GPT-4 to generate ten alternative words for the target word as follows:

> *I will give you a* `LANG_NAME` *sentence and a word in the 'Sentence' and 'Word' format. List ten alternatives for the Word that are easier to understand, separated by ','.*
> *You must follow these four rules.*
> *1. Take into account the meaning of the Word in the Sentence.*
> *2. Alternatives must be easier to understand than the Word.*
> *3. Each alternative consists of one word.*
> *4. Do not generate an explanation.*

[3] https://github.com/MLSP2024/MLSP_Data/tree/main

[4] We indicated the hyperparameters, such as n, temperature, and frequency_penalty, in Table 4.

[5] We used `gpt-4-0125-preview` in LS experiments.

[6] We designed a specific prompt for the Japanese. Please refer to Appendix A for details.

	ACC@k@Top1			Potential@k				MAP@k		
	k=1	k=2	k=3	k=1	k=3	k=5	k=10	k=3	k=5	k=10
Base system (Run 1)	**0.3772**	**0.4919**	**0.5498**	**0.6739**	**0.8071**	**0.8407**	**0.8759**	**0.4652**	**0.3421**	**0.2026**
w/ Ranking$_{\mathrm{GPT-4}}$ (Run 2)	0.3573	0.4792	**0.5498**	0.6391	**0.8071**	**0.8407**	**0.8759**	0.4570	0.3371	0.2001
w/ Ranking$_{\mathrm{XGLM}}$ (Run 3)	0.2933	0.4554	**0.5498**	0.5918	**0.8071**	**0.8407**	**0.8759**	0.4461	0.3306	0.1969

Table 3: LS results on the test dataset for all languages.

The rules 3 and 4 are to ensure generating an alternative word consisting of a single word. We observed that GPT-4 generates "descriptions" rather than truly synonymous expressions without the rules. For instance, "neither positive nor negative" was generated as an alternative word for "neutrally." Since these "descriptions" were not appropriate as alternative words, we added the rules 3 and 4 to the prompt.

We let GPT-4 generate alternatives using the prompts for n times. Then, we ensemble the $n \times 10$ alternatives following Aumiller and Gertz (2022). We refer to this approach as "Base" (Run 1).

3.2 Substitution Ranking

We observed that the Base system exhibited high Potential@3 scores in the trial dataset [7], indicating that in numerous instances, at least one of the top three alternatives predicted by the system was present in the gold annotations. Therefore, we hypothesized that scores on metrics such as ACC@1 can be enhanced by re-ranking the top three words. In Run2 and Run3, we undertook the re-ranking of the top three alternatives for each instance from the Base system.

GPT-4-based re-ranking. Previous studies ranked alternative words based on their semantic similarity to the target word (Seneviratne et al., 2022; Whistely et al., 2022) or their familiarity to people (frequency of occurrence in a corpus) (Li et al., 2022; North et al., 2022). Following the studies, we designed two distinct prompts for re-ranking the generated alternatives in terms of semantic similarity to the target word and the alternatives' ease, respectively. We re-ranked the alternatives through each prompt and used a composite ranking as the final prediction. We refer to the approach as "Ranking$_{\mathrm{GPT-4}}$" (Run 2).

XGLM-based re-ranking. In addition, we hypothesized that words' preference varies between human annotators and GPT-4 due to disparities in the extent of knowledge accumulated. Therefore, we trained a re-ranking model to fill the gap and reflect annotators' preferences. Specifically, we performed an instruction-tuning of XGLM (Lin et al., 2022) using the trial data[8]. We re-ranked alternatives using the resulting model. We refer to this approach as "Ranking$_{\mathrm{XGLM}}$" (Run 3).

3.3 Experiments

Experimental settings. We employed the same datasets as described in Subsection 2.3 for evaluation. For evaluation metrics, we used ACC@k@Top1, Potential@k, and MAP@k following Saggion et al. (2022).

Experimental results and discussions. Table 3 shows results on the test set of all languages. The Base system outperformed the re-ranking systems, and this trend held in nine out of the ten languages except for Sinhala.

These results indicate that the ranking of alternatives generated by GPT-4 within the Base system is comparatively appropriate, whereas Ranking$_{\mathrm{GPT-4}}$ and Ranking$_{\mathrm{XGLM}}$ do not yield appropriate rankings. Notably, the scores of Ranking$_{\mathrm{XGLM}}$ are significantly degraded, suggesting that it is difficult to train a re-ranking model using only the trial data (i.e. 30 examples for each language). Developing a better re-ranking strategy is one of the challenges to further enhance the scores.

4 Conclusion

In this paper, we introduced GPT-4-based systems designed to assess word complexities and replace complex words with simpler ones. Our systems achieved superior performance in multiple languages for both LCP and LS tasks within MLSP-2024 Shared Task.

[7] Table 6 shows the scores of the Base system on the trial dataset for each language.

[8] The details about how to create instruction-tuning data are described in Appendix B.

To score complexities, we created a base prompt following G-EVAL (Liu et al., 2023) and added to the base prompt supplementary prompts to delineate the language of the test example and the role of the LLM. Our prompt, when applied within a three-shot setting, consistently achieved the highest Pearson's correlation across the majority of languages. Furthermore, our experiments suggest the potential for performance enhancement through the augmentation of few-shot examples. Therefore, we plan to explore the change in performance resulting from an increment in the number of few-shot examples.

For the task of replacing complex words with simpler alternatives, we manually crafted prompts. The experimental results indicate that these prompts yield alternatives of commendable quality. Additionally, we explored the possibility of enhancing the selection of generated alternatives by employing a re-ranking strategy using either GPT-4 or XGLM that were instruction-tuned by trial data. However, the re-raking approaches degraded the scores compared to the ones before re-ranking. For future work, we plan to devise an improved re-ranking methodology.

5 Limitations

Our approach leverages the OpenAI API, which can be costly. In order to make Lexical Simplification easily available to many users, it might be essential to devise an approach built on open-source models, achieves comparable performance to this study.

Acknowledgments

This work was supported by JST, PRESTO Grant Number JPMJPR2366, Japan, and the National Institute of Information and Communications Technology (NICT) under the Research and Development of externally controllable modeling of multimodal information to enhance the accuracy of automatic translation.

References

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. 2023. MEGA: Multilingual evaluation of generative AI. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267, Singapore. Association for Computational Linguistics.

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.

Tuqa Bani Yaseen, Qusai Ismail, Sarah Al-Omari, Eslam Al-Sobh, and Malak Abdullah. 2021. JUST-BLUE at SemEval-2021 task 1: Predicting lexical complexity using BERT and RoBERTa pre-trained language models. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 661–666, Online. Association for Computational Linguistics.

Stefan Bott, Horacio Saggion, Nelson Peréz Rojas, Martin Solis Salazar, and Saul Calderon Ramirez. 2024. Multils-sp/ca: Lexical complexity prediction and lexical simplification resources for catalan and spanish. *Preprint*, arXiv:2404.07814.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Tomoyuki Kajiwara and Mamoru Komachi. 2018. Complex word identification based on frequency in a learner corpus. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 195–199, New Orleans, Louisiana. Association for Computational Linguistics.

Xiaofei Li, Daniel Wiechmann, Yu Qiao, and Elma Kerz. 2022. MANTIS at TSAR-2022 shared task: Improved unsupervised lexical simplification with pretrained encoders. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 243–250, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval:

NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Diana McCarthy and Roberto Navigli. 2007. SemEval-2007 task 10: English lexical substitution task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 48–53, Prague, Czech Republic. Association for Computational Linguistics.

Kai North, Alphaeus Dmonte, Tharindu Ranasinghe, and Marcos Zampieri. 2022. GMU-WLV at TSAR-2022 shared task: Evaluating lexical simplification models. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 264–270, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Kai North, Tharindu Ranasinghe, Matthew Shardlow, and Marcos Zampieri. 2023. Deep learning approaches to lexical simplification: A survey. *Preprint*, arXiv:2305.12000.

Kai North, Tharindu Ranasinghe, Matthew Shardlow, and Marcos Zampieri. 2024. Multils: A multi-task lexical simplification framework. *arXiv preprint arXiv:2402.14972*.

Gustavo Paetzold and Lucia Specia. 2016. 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.

Chunguang Pan, Bingyan Song, Shengguang Wang, and Zhipeng Luo. 2021. DeepBlueAI at SemEval-2021 task 1: Lexical complexity prediction with a deep ensemble approach. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 578–584, Online. Association for Computational Linguistics.

Piotr Przybyła and Matthew Shardlow. 2020. Multiword lexical simplification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1435–1446, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2019. Lexical simplification with pretrained encoders. In *AAAI Conference on Artificial Intelligence*.

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.

Sandaru Seneviratne, Elena Daskalaki, and Hanna Suominen. 2022. CILS at TSAR-2022 shared task: Investigating the applicability of lexical substitution methods for lexical simplification. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 207–212, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

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. 2024a. 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, 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, Marcos Zampieri, and Horacio Saggion. 2024b. An Extensible Massively Multilingual Lexical Simplification Pipeline Dataset using the MultiLS Framework. In *Proceedings of the 3rd Workshop on Tools and Resources for People with REAding DIfficulties (READI)*.

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.

Lucia Specia, Sujay Kumar Jauhar, and Rada Mihalcea. 2012. SemEval-2012 task 1: English lexical simplification. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 347–355, Montréal, Canada. 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.

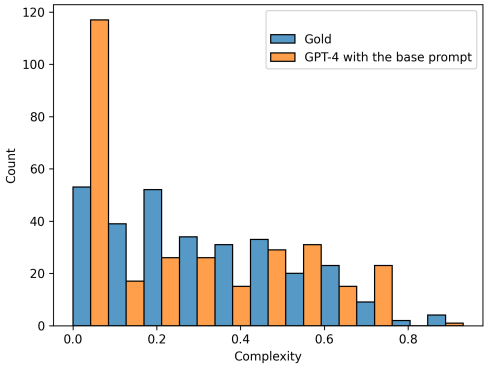

Figure 1: The histograms of the gold complexities and those derived from GPT-4 using the base prompt P_{base}. This figure shows that the complexities generated by GPT-4 are distributed predominantly within the range of 0.0 to 0.1.

	LCP	LS		
		SG	SG (ja)	SR
temperature	0.7	0.7	0.7	0.7
frequency_penalty	0.0	0.5	0.0	0.0
presence_penalty	0.0	0.3	0.0	0.0
n	20	10	10	10

Table 4: Hyperparameters

A Japanese Specific Prompt

In the case of Japanese, instead of generating only alternative words, we instructed GPT-4 to generate sentences in which the target word was replaced with each alternative word. Unlike the other nine languages, Japanese doesn't have spaces between words. Additionally, Japanese verbs, adjectives, adjectival verbs and auxiliary verbs undergo "Katsuyou" (inflection), wherein the ending of a word changes depending on the subsequent word. Some target words in the Japanese dataset are in Katsuyou-form; for instance, "募集し" is in the Katsuyou-form, while "募集する" is in the Basic-form. We observed that when we instructed GPT-4 to generate alternative words for a target word in Katsuyou-form, it often generated words in Katsuyou-form that did not suit the sentence or words in Basic-form. On the other hand, when we instructed GPT-4 to generate sentences in which the target word was replaced with each alternative word, GPT-4 could generate alternative words that have the correct Katsuyou-form to fit the sentence. Table 7 shows examples of GPT-4 outputs for each method. The details of the prompt are shown in Table 8.

Language	Number of Examples
English	570
Catalan	445
French	570
German	570
Spanish	593
Italian	570
Portuguese	569
Filipino	570
Japanese	570
Sinhala	600

Table 5: The size of test datasets.

Language	ACC@1	Potential@k		
		k=3	k=5	k=10
Catalan	0.600	0.866	0.866	0.900
English	0.766	0.833	0.866	0.866
Filipino	0.566	0.633	0.633	0.700
French	0.866	0.966	0.966	0.966
German	0.800	0.933	0.933	0.933
Italian	0.866	0.933	0.933	0.933
Japanese	0.800	0.966	0.966	0.966
Portuguese	0.666	0.766	0.800	0.900
Sinhala	0.600	0.733	0.766	0.800
Spanish	0.766	0.833	0.866	0.900

Table 6: LS results on the trial dataset for each language.

B Dataset Creation for Instruction-Tuning of XGLM

The alternative words listed as gold are ranked by frequency of suggestion by the annotators. We used this ranking to create data for instruction-tuning of XGLM from the trial data in eight languages except Filipino and Sinhala. [9] The query of the created data consisted of a contextual sentence, a target word, two alternative words in the trial data, and an instruction letting a model select a more suitable alternative word. The answer was the alternative with the highest ranking among the two alternatives. When Alternative 1 was ranked higher than Alternative 2 in the trial data, the template is as follows:

> *### Instruction : I will give you a* `{LANG_NAME}` *sentence, a word contained in the sentence and alternatives*

[9] Since Filipino and Sinhala are not included in the XGLM pre-training data, we exclude these languages from the fine-tuning data.

Sentence	ドラマに関する感想を募集し、週ごとにピックアップして回答も掲載した。
Target word	募集し
Gold	集め, 促し, 募り, 探し, 集めて, 呼びかけ, 広く集め, 呼びかけて, たくさん求め, 書いてもらい, ...
Only word	集めています, 求めています, 探しています, 募っています, 応募を受け付けています, 呼びかけています, 求めている, 探している, 求人しています, 集めている
With sent	集め, 求め, 探し, 求めて, 探して, 招待し, 募って, 要求し, 呼びかけ, 呼びかけて

Table 7: Examples of GPT-4 output in Japanese. "Gold" represents the correct answer in the trial data. "Only word" and "With sent" represent outputs when we instructed GPT-4 to generate ten alternative words and sentences where the target words are replaced with each alternative word, respectively.

for the word in the 'Sentence', 'Word' and 'Alternatives' format. Choose a more suitable alternative word to the Word in the Sentence.
Sentence：{SENTENCE}
Word：{TARGET_WORD}
Alternatives：{ALTERNATIVE 1, ALTERNATIVE 2}
Response：{ALTERNATIVE 1}

We conducted re-ranking by employing a XGLM instruction-tuned on this dataset to predict the portion following *"### Response:"*.

Setting	Prompt Template
SG (non ja)	*I will give you a* {LANG_NAME} *sentence and a word in the 'Sentence' and 'Word' format. List ten alternatives for the Word that are easier to understand, separated by ','.* *You must follow these four rules.* *1. Take into account the meaning of the Word in the Sentence.* *2. Alternatives must be easier to understand than the Word.* *3. Each alternative consists of one word.* *4. Do not generate an explanation.* *Sentence:* {SENTENCE} *Word:* {TARGET_WORD} *Alternatives:*
SG (ja)	*I will give you a Japanese sentence and a word in the 'Sentence' and 'Word' format. Think ten easier alternatives for the Word in the Sentence. Then, output sentences where you have replaced the Word with each alternative enclosed by '**'.* *You must follow these three rules.* *1. Take into account the meaning of the Word in the Sentence.* *2. Alternatives must be easier to understand than the Word.* *3. Do not generate an explanation.* *Sentence:* {SENTENCE} *Word:* {TARGET_WORD} *Alternative sentences:*
SR (ease)	*I will give you a* {LANG_NAME} *sentence, a word and alternatives for the word in the 'Sentence', 'Word' and 'Alternatives' format. Arrange the Alternatives in order of their ease. Do not generate an explanation.* *Sentence:* {SENTENCE} *Word:* {TARGET_WORD} *Alternatives:* {ALTERNATIVES} *Sorted Alternatives:*
SR (sim)	*I will give you a* {LANG_NAME} *sentence, a word and alternatives for the word in the 'Sentence', 'Word' and 'Alternatives' format. Arrange the Alternatives in order of their semantic similarity to the Word, taking into account the meaning of the Words in the Sentence. Do not generate an explanation.* *Sentence:* {SENTENCE} *Word:* {TARGET_WORD} *Alternatives:* {ALTERNATIVES} *Sorted Alternatives:*

Table 8: Prompt templates used for GPT-4 in LS experiments. "SG" and "SR" represent the Substitute Generation and Substitute Ranking, respectively. LANG_NAME is empty when the language is English. In SG (ja), SENTENCE is a sentence with the target word encloseed by '**'. In SR, "ease" represents ranking based on ease of each alternative word, and "sim" represents ranking based on semantic similarity of each alternative word to the target word.