Investigating strategies for lexical complexity prediction in a multilingual setting using generative language models and supervised approaches

Abdelhak Kelious[1], **Mathieu Constant**[1], **Christophe Coeur**[2]
[1]University of Lorraine and CNRS/ATILF, [2]Consultant
abdelhak.kelious@univ-lorraine.fr
mathieu.constant@univ-lorraine.fr
christophe.coeur@gmail.com

Abstract

This paper explores methods to automatically predict lexical complexity in a multilingual setting using advanced natural language processing models. More precisely, it investigates the use of transfer learning and data augmentation techniques in the context of supervised learning, showing the great interest of multilingual approaches. We also assess the potential of generative large language models for predicting lexical complexity. Through different prompting strategies (zero-shot, one-shot, and chain-of-thought prompts), we analyze model performance in diverse languages. Our findings reveal that while generative models achieve promising performances, their predictive quality varies and optimized task-specific models still outperform them when they benefit from sufficient training data.

1 Introduction

Lexical complexity prediction consists in assessing the difficulty of a target word in a given context, either as a binary classification (is the word difficult or not?) or as a continuous numerical value prediction indicating the degree of complexity. Such a task is potentially useful for computer-assisted language learning: e.g. for selecting relevant textual materials for learners or for identifying complex words in texts and then providing enriched information to help the reader's understanding.

Our study explores deep learning methodologies for multilingual lexical complexity prediction (LCP). We leverage recent advances in natural language processing models, such as transformers and generative models, to assess lexical complexity across various languages. More precisely, we first investigate various multilingual methods like transfer learning and data augmentation using a supervised approach. We then explore the capabilities of generative pre-trained large language models (LLMs) to perform LCP applying various prompt engineering and ensemble techniques. The experiments are carried out on multilingual datasets from two shared tasks: the 2018 Complex Word Identification task (Yimam et al., 2018a) for English, French, German and Spanish, and the Multilingual Lexical Simplification Pipeline (MLSP) shared task (Shardlow et al., 2024a) for a subset of languages (English, French, Japanese and Spanish).

2 Related work

Lexical complexity prediction has been a growing area of research, with several works contributing to the development of graded lexical resources and methodologies aimed at understanding word complexity from both native and non-native language learners' perspectives. For example, Gala et al. (2013) laid the groundwork for French lexical complexity by proposing a lexicon with difficulty measures. Building on this, François et al. (2014) introduced FLELex, a graded lexical resource specifically designed for French foreign learners. Tack et al. (2018) extended this research to Dutch with NT2Lex, a graded lexical resource linked to the Dutch WordNet. Meanwhile, Alfter and Volodina (2018) focused on predicting single-word lexical complexity, a task later expanded by Alfter (2021) to include multi-word expressions, highlighting the evolving nature of complexity prediction tasks. For more details on this task, North et al. (2023) provided a comprehensive overview of the computational approaches used.

2.1 Shared tasks

Lexical complexity prediction has also been the focus of multiple shared tasks over the last decade that strongly contributed to the advances of the field through the development of new dedicated

Abdelhak Kelious, Mathieu Constant and Christophe Coeur. Investigating strategies for lexical complexity prediction in a multilingual setting using generative language models and supervised approaches. *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning (NLP4CALL 2024)*. Linköping Electronic Conference Proceedings 211: 96–114.

datasets as well as novel technical methods to perform the task.

The 2016 Complex Word Identification (CWI) task at SemEval highlighted key findings in identifying complex words, especially for non-native English speakers. The dataset, dedicated to the English language, created with input from 400 non-native speakers, showed that complex words are generally rarer, less ambiguous, and shorter. Decision trees, ensemble methods, and particularly word frequency were found to be reliable predictors of word complexity (Paetzold and Specia, 2016). Top systems, such as those by UWB and LTG, utilized features like document frequency and contextual language models, achieving high rankings (Konkol, 2016; Malmasi et al., 2016). Despite various feature explorations and innovative methods like sequence labeling (Gooding and Kochmar, 2019), the fundamental effectiveness of word frequency remained central to CWI success (Zampieri et al., 2017).

The 2018 Complex Word Identification task, thereafter CWI 2018, part of the BEA workshop at NAACL 2018, focused on identifying difficult words in texts across multiple languages, including English, German, Spanish, and French. The task was divided into binary and probabilistic classification tracks, attracting 12 teams with various approaches. Notably, ensemble-based methods and feature engineering demonstrated strong performance (Yimam et al., 2018a). Systems such as those by the NLP-CIC team compared deep learning with feature engineering, showing comparable results (Aroyehun et al., 2018). Simple models based on character n-grams also performed competitively, sometimes matching more complex systems (Alfter and Pilán, 2018). The challenge highlighted the effectiveness of both traditional feature engineering and modern deep learning approaches in CWI.

The 2021 Lexical Complexity Prediction (LCP 2021) task (Shardlow et al., 2021) at SemEval involved predicting, for the English language, the complexity of single words and multi-word expressions in context using a five-point Likert scale. The competition attracted 198 teams, with top-performing systems leveraging advanced NLP techniques such as transformers and ensemble methods. The winning system used fine-tuned pre-trained language models with stacking mechanisms, achieving high Pearson correlation scores (Pan et al., 2021). Approaches varied widely, from logistic regression with linguistic features (Desai et al., 2021) to ensemble-based models combining different feature types (Vettigli and Sorgente, 2021). The task highlighted the effectiveness of combining traditional linguistic features with modern deep learning models to predict lexical complexity accurately.

Recently, a dataset was developed for the MLSP 2024 shared task (Shardlow et al., 2024a). It includes 5,624 instances across 10 target languages. Each instance features a sentence from an educational text with a specific target word highlighted. For each target word, there are two types of annotations: an aggregate complexity score (rated on a scale from 1 to 5 by 10 annotators) indicating the difficulty level of the word, and a list of possible substitutions that simplify the sentence while preserving its original meaning.

2.2 Multilingual approaches

Although many studies concentrate on English due to a relative shortage of resources in other languages, promising approaches such as transfer learning and data augmentation have been proposed to address this gap. Cross-lingual transfer learning significantly enhances Complex Word Identification (CWI) by leveraging models trained in high-resource languages for use in low-resource languages. Zaharia et al. (2020) demonstrated the effectiveness of zero-shot, one-shot, and few-shot learning techniques with state-of-the-art NLP models, achieving high F1-scores across multiple languages. Bingel and Bjerva (2018) used cross-lingual multitask learning, showing that language-agnostic models could generalize well across different languages. Additionally, Yimam et al. (2017) employed language-independent features to train multilingual and cross-lingual models, achieving comparable performance to monolingual systems.

2.3 Large language models' capabilities

Large Language Models (LLMs) like ChatGPT, Mistral, and Llama3 have significantly advanced natural language processing across various domains. Given that we are currently in the era of LLMs, it is crucial to compare and assess their role in our study to understand their impact on various tasks. They excel in industrial engineering tasks, such as automation and programming, though they have limitations with complex physics

equations (Ogundare et al., 2023). In mathematical problem-solving, LLMs effectively handle arithmetic tasks using chain-of-thought reasoning (Yuan et al., 2023). Their ability to use multimodal tools is enhanced by frameworks like GPT4Tools, which improve performance in visual tasks (Yang et al., 2023). Instruction-following datasets and fine-tuning, as seen with FLACUNA, enhance their problem-solving skills (Ghosal et al., 2023). Comprehensive evaluations reveal strengths in diverse tasks like question-answering and code generation, although challenges remain (Laskar et al., 2023). Techniques like role-play prompting further improve their reasoning capabilities, making LLMs versatile tools for a wide range of applications (Kong et al., 2023). The ANU team, participating in the MLSP 2024 task to predict word complexity based on context, relied on a prompting strategy with GPT-3.5 (i.e. GPT-3.5-turbo-instruct) for the tasks using zero, one, and few-shot strategies. The zero-shot strategy included the context and target word while the non-zero strategies relied on instructing the model with one or three random samples from the trial data according to the prompting template. Overall, the authors indicate under-performance for the LCP task, while demonstrating strong performance for English in lexical simplification (Seneviratne and Suominen, 2024).

3 Multilingual lexical complexity prediction based on supervised learning

In this section, we investigate two main strategies for the task of lexical complexity prediction (LCP) in multiple languages using a supervised approach:

1. **Monolingual training**: the model is trained on a dataset in the target language; the training data may be composed of native data in the target language, data translated to the target language from a resource-richer language (English in our case), or a combination of both where the native data is augmented with translated data;

2. **Multilingual training**: the model is trained on a multilingual dataset including or not data in the target language; the model is based on multilingual word embeddings to deal with transfer learning.

The actual implementation of these approaches will depend on the dataset on which they will be experimented, given their different nature and composition (cf. section 3.1 and section 3.3).

3.1 Datasets

Experiments to evaluate these strategies are performed on two multilingual datasets: CWI 2018 (Yimam et al., 2018b) and MLSP 2024 (Shardlow et al., 2024a), cf. section 2. The CWI 2018 dataset provided by (Yimam et al., 2018b) includes data in English, Spanish, and German for training and testing, and French solely for testing purposes, cf. table 1. Our focus is on Spanish, German, and French. We selected this dataset because it offers large possibilities of multilingual experiments using supervised learning. Two types of labels are available: binary and probabilistic. Our evaluation is conducted using the binary labels.

Language	Train	Dev	Test
English	27,299	3,328	4,252
German	6,151	795	959
Spanish	13,750	1,622	2,233
French	-	-	2,251

Table 1: The number of instances for each training, development and test set (Yimam et al., 2018b)

Additionally, we performed evaluation on the MLSP 2024 dataset (Shardlow et al., 2024a), which includes 5,624 instances across 10 target languages. The MLSP dataset provides probabilistic labels, where annotations are continuous values between 0 and 1. This dataset contains only testing and development data, the latter being limited to around 30 instances per language, i.e. 300 instances in total. We only focus on four languages (French, English, Japanese, and Spanish) in order to limit the energetic impact of our experiments and to focus on the languages studied in our working environment. Due to the lack of training data, we have decided to leverage the LCP 2021 dataset (Shardlow et al., 2021), which provides annotations highly similar to those in the MLSP task, for the English language.

3.2 The model

In our research, we adopt a recent system that has proven effective in predicting lexical complexity for English (Kelious et al., 2024). We replicate this model in a multilingual version. The model

combines a pre-trained language model with frequency characteristics based on Zipf's law. Such a system is in line with the literature showing that hybrid models using transformers (encoders) enhanced with additional linguistic features deliver more robust and effective results (Wilkens et al., 2024).

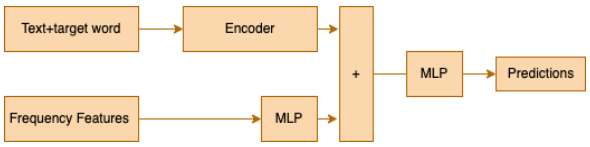

Figure 1: The overall architecture for predicting lexical complexity (Kelious et al., 2024).

Figure 1 illustrates the model described with more details in Kelious et al. (2024). This model is divided into two main parts. The first part relies on lexical embeddings: the encoder receives the target word and its context as input, formatted as follows: *[CLS]Context[SEP]Target Word*, where [CLS] and [SEP] are special tokens used in the Transformers model for processing texts. The second part incorporates five characteristics based on Zipf's frequency, processed by a multilayer perceptron (MLP). The whole, i.e. the concatenation of the two parts, is then processed by an additional MLP layer. The model's output is a continuous value between 0 and 1. To classify this output into binary classes, we add a sigmoid layer and apply a decision threshold set at 0.5 to convert the probabilities into binary classes for the experiments on CWI 2018.

The conversion of this model from monolingual to multilingual is relatively straightforward: for the frequency features, it suffices to extract frequency data in the target language from available corpora. As for the transformer (encoder) part, it is necessary to implement a multilingual model or a monolingual model suited to the specific language we wish to evaluate.

3.3 Experimental settings

The LCP model is based on various language models for encoding the input context. For multilingual training strategies, we selected the multilingual language model mdeberta-v3-base[1]. For monolingual training strategies, we selected Spanish BERT for Spanish (Cañete et al., 2020), German BERT for German (Chan et al., 2020), DeBerta (He et al., 2023) for English and mdeberta-v3-base for Japanese. The Zipf frequencies were computed using the python library Wordfreq[2]. For translating data from English to target languages such as French, German, Japanese and Spanish, we used the M2M100 model (Fan et al., 2021).

3.4 Experiments on CWI 2018

This section presents and evaluates the multilingual and monolingual training strategies developed on the CWI 2018 dataset using a supervised approach.

3.4.1 Evaluated methods

For the multilingual training approaches, the experiments were the following:

- **Multilingual (en, de, es)**: the LCP model is trained on the training data of all languages having training data, namely English (en), German (de) and Spanish (es);
- **Multilingual (zero shot)**: the model is trained on the training data of all languages having training data except the target language, resulting in a zero-shot scenario.

We also experimented the following monolingual training approaches:

- **Monolingual (native data)**: the LCP model is trained on the native train dataset of the target language;
- **Monolingual (native + translated data)**: the model is trained on the native train dataset of the target language, augmented with a portion of the English training dataset translated to the target language;
- **Monolingual (translated data)**: the model is trained on a portion of the English training dataset translated to the target language.

The experiments Monolingual (native data) and Monolingual (native + translated data) were not performed for French as it has no training data. The experiment Monolingual (translated data) was only performed for French.

[1] https://huggingface.co/microsoft/mdeberta-v3-base

[2] https://pypi.org/project/wordfreq/

3.4.2 Results

Tables 2 and 3 shows F1 scores for Spanish, German, and French. For the sake of comparison, we also provide the results of the CWI 2018 official baseline, of the best systems of the shared task, and of a random baseline randomly selecting the output from {0,1}. We can derive several insights and make observations regarding the performance and trends across different types of training strategies:

Multilingual learning. Generally, multilingual models trained on all languages (but French) have strong performance across all languages (Spanish : 0.800, German : 0.7911, French : 0.799).The zero-shot configuration, which involves using a model in scenarios where it hasn't been explicitly trained on the target language's data, performed reasonably well but not as well as multilingual models trained on all languages (Spanish: 0.746, German: 0.744), cf. Table 2. The high score for French 0.799 in Table 3 indicates that the model benefits significantly from being part of a multilingual setup where the knowledge from other languages can be effectively transferred to French even without direct training. It suggests that the underlying representations learned by the model are robust and applicable across languages.

Model	Spanish	German
Multilingual (es, en, de)	0.800	0.791
Multilingual (zero shot)	0.746	0.744
Monolingual (native data)	0.775	0.761
Monolingual (native data + 4k translated instances))	0.789	0.781
The highest score in (Yimam et al., 2018b)	0.769	0.745
Baseline, from (Yimam et al., 2018b)	0.723	0.754
Random	0.43	0.44

Table 2: F1 Scores for Spanish and German Language Models

Model	F1 Score
Multilingual (zero shot)	0.799
Monolingual (translated data - 2k)	0.770
Monolingual (translated data - 4k)	0.713
Monolingual (translated data - 10k)	0.751
Monolingual (translated data - 27k)	0.717
The highest score in (Yimam et al., 2018b)	0.759
Baseline, from (Yimam et al., 2018b)	0.634
Random	0.38

Table 3: F1 Scores for French

Monolingual learning. Focused training on a single language shows competitive results but still lags slightly behind the multilingual approach: Spanish: 0.775, German: 0.761, cf. Table 2. Augmenting the data with translations from the English data tends to be useful, as shown in Table 2, especially with an augmentation of 4k training instances translated from English to the target language. Other tested sizes tend to reach lower performance.

Regarding French, the LCP model does not use native training data but instead relies on data created by translating the English training dataset to French. This method shows varying performances as the data size increases (F1 scores: 0.770 with 2k instances, 0.713 with 4k, 0.751 with 10k, 0.717 with 27k, the full training set). The fluctuating performance with different dataset sizes indicates that the quality and consistency of translated data might vary significantly, impacting the model's learning and performance. Simply increasing the dataset size does not consistently improve performance. This approach highlights the challenges and limitations of relying on translated data for training language models, where nuances and context-specific elements of the original language might be lost or misrepresented in translation.

Baseline and Random. The baseline and random models provide a clear floor for performance, with baselines substantially outperforming random guessing across all languages (Baseline vs. Random: Spanish 0.7237 vs. 0.43, German 0.754 vs. 0.44, French 0.634 vs. 0.38). This reflects the effectiveness of even basic modeling techniques over uninformed strategies.

The analysis highlights that while multilingual training on all languages offers robustness and generalization across languages, targeted strategies such as monolingual training still hold importance, especially when resources are limited. The fluctuation in performance with different data sizes and types of augmentation indicates the need for careful data management and model tuning specific to each language's characteristics.

3.5 Experiments on MLSP 2024

In this section, we present the multilingual and monolingual experiments developed for the MLSP 2024 dataset using a supervised approach.

	English			French			Spanish			Japanese		
	Pearson	Spearman	R2	Pearson	Spearman	R2	Pearson	Spearman	R2	Pearson	Spearman	R2
Multilingual (LCP 2021)	0.80	0.76	0.64	0.52	0.49	0.27	0.67	0.64	0.46	0.63	0.65	0.40
Multilingual (LCP 2021+ Dev)	0.83	0.78	0.69	0.56	0.52	**0.31**	0.67	0.63	0.45	0.66	0.67	**0.43**
Multilingual (LCP 2021 + Dev + 2k translated data)	/	/	/	0.49	0.47	0.24	0.65	0.64	0.43	0.61	0.61	0.37
Multilingual (LCP 2021 + Dev + 4k translated data)	/	/	/	0.51	0.48	0.26	0.63	0.58	0.39	0.63	0.63	0.40
Monolingual (native data)	**0.87**	**0.80**	**0.72**	/	/	/	/	/	/	/	/	/
Monolingual (translated data)	/	/	/	0.44	0.42	0.19	0.67	0.58	0.39	0.57	0.57	0.32
Baseline MLSP 2024 (Shardlow et al., 2024b)	0.74	0.74	0.54	0.51	0.52	0.14	0.55	0.52	0.25	0.64	0.66	0.33
Highest mlsp score for English : (Goswami et al., 2024)	0.84	0.79	0.52	0.31	0.32	0.04	0.24	0.19	0.07	0.17	0.18	0.02
Highest mlsp score for French, Spanish and Japanese : (Enomoto et al., 2024)	0.81	0.75	0.51	**0.62**	**0.63**	0.27	**0.76**	**0.74**	**0.49**	**0.73**	**0.73**	0.41

Table 4: Scores for different languages and methods (Pearson, Spearman, R2)

3.5.1 Evaluated methods

Since we only have test and development data for the MLSP 2024 dataset, we will use for training the LCP 2021 dataset (Shardlow et al., 2021) containing 7,662 single-word instances exclusively in English. The evaluated methods using a multilingual training approach are the following:

- **Multilingual (LCP 2021)**: the LCP model is based on multilingual word embeddings and is trained exclusively on English data from LCP 2021 task;
- **Multilingual (LCP 2021 + Dev)**: the model based on multilingual word embeddings is trained on LCP 2021 (English data) augmented with the development data in the 10 languages of the MSLP 2024 task (around 30 instances per language) to improve adaptation to the target languages;
- **Multilingual (LCP 2021 + Dev + translated data)**: the model based on multilingual word embeddings is trained on the training data of Multilingual (LCP 2021 + Dev), augmented with 2k or 4k instances from LCP 2021 translated to the target language.

For the monolingual training setting, we evaluated the following approaches for which the LCP model is specific to each target language:

- **Monolingual (native data)**: the LCP model is trained on native data in the target language; this experiment is only performed for English using the LCP 2021 as training data.
- **Monolingual (translated data)**: the model is trained on the translation of LCP 2021 training data (English) to the target language; this experiment is performed on all languages but English.

3.5.2 Results

Table 4 presents the evaluation for predicting word complexity in English, French, Spanish, and Japanese using the learning methods presented in section 3.5.1. The evaluation metrics include the Pearson, Spearman, and R^2 scores, as is usually done for this task (cf. Shardlow et al. (2021)). The results of the best MSLP 2024 systems and of the official baseline are also provided for the sake of comparison:

- **Baseline Model**: The baseline is based on linear regression and is trained using log-frequency on the trial set for each language;
- **GMU Team** (Goswami et al., 2024): Employed a weighted ensemble of mBERT, XLM-R, and language-specific BERT models. All trial data was used for cross-lingual training and evaluation. For English, they augmented the data with the CompLex dataset (Shardlow et al., 2020).
- **TMU-HIT Team** (Enomoto et al., 2024): Used a chain-of-thought based prompting method employing GPT-4 to generate an instruction in English, and subsequently assigned complexity scores to target words across all languages based on the English instruction.

In English, the Monolingual method, specific to the target language, achieved the best scores (Pearson 0.87, Spearman 0.80, R^2 0.72), thanks to the use of specific annotated data. For French

and Japanese, Multilingual methods trained exclusively on English outperformed the monolingual method based on translation, indicating that multilingual training can be beneficial when annotated data is limited. Adding small amounts of multilingual development data (Multilingual (LCP 2021 + dev)) slightly improved performance in French and Japanese. However, increasing the data through translation (Multilingual(LCP 2021 + Dev + 2k or 4k translated data) did not yield significant improvements. The best scores for French, Spanish, and Japanese were achieved by Enomoto et al. (2024), suggesting that their approach is more effective for these languages.

4 Prompting Large Language Models for multilingual lexical complexity prediction

In this section, we focus on assessing the capability of generative large language models (LLMs) to predict the complexity of a word based on its context. To do this, we use three types of prompt strategies:

- **Zero-shot prompt (base):** The model receives instructions without any specific examples on how to perform the task, relying solely on the knowledge acquired during its training. (See Appendix A)
- **One-shot prompt (instruct):** This type of prompt includes some guidelines used during data annotation, along with an example, thus providing a frame of reference for the model. (See Appendix A)
- **Chain-of-thought prompt (Advanced COT):** This prompt includes detailed annotation instructions, methodological steps to follow and analysis before delivering an evaluation, illustrated by an example (See Appendix A).

4.1 Experimental settings

For this evaluation, we use five different language models: gpt-4o (June 10, 2024) [3], Llama3 (Dubey et al., 2024), Mistral (Jiang et al., 2023), Phi3 (Abdin et al., 2024), and Gemma (Team et al., 2024). The last four models are used in their 4-bit quantized versions. It's important to note that comparing these models might seem unfair if gpt-4o is included, however, our main goal remains to analyze the effectiveness of each type of prompt according to the model. Yet, the comparison in terms of performance remains relatively fair if gpt-4o is excluded, considering all other models share the same type of quantization. Nonetheless, the number of parameters of each model must be considered, for example, Phi3 with 3.8 billion parameters is significantly less than Gemma, which has 9 billion, while Mistral and Llama are approximately similar in size. We use Ollama [4], an open-source tool, to test these different LLMs, keeping the default settings provided. All the prompts are written in English, but they explicitly indicate the target language.

Detailed evaluation of these strategies is first undertaken using the MLSP 2024 dataset (Shardlow et al., 2024a). For this task, the generative models are asked through the prompts to predict a score on a scale (0, 0.25, 0.5, 0.75, 1) for the target word in a given context in the target language, in order to mimic the human annotators of the dataset. The evaluation metrics include the Pearson, Spearman, and R^2 scores, as is usually done for this task (cf. Shardlow et al. (2021)). We used a subset of the available languages (English, French, Japanese, and Spanish). In addition, we also evaluate on the binary classification data from CWI 2018 in French, German, and Spanish, adapting the prompts to each task and using the F1 score for evaluation (See Appendix A).

For the sake of comparison between the supervised approach and this one, we also provide the performance of a model specifically trained on this task using a multilingual supervised approach.

4.2 Results

In this part, we will evaluate the various prompt strategies for various LLMs for two different datasets: LCP 2018 and MLSP 2024.

4.2.1 CWI 2018

Table 5 presents the F1 scores for predicting word complexity based on context in French, German, and Spanish. The supervised method achieves the best results across all three languages. Among the language models, gpt-4o and Llama3 display the highest performance. For gpt-4o, the **Instruct** prompt yields the best scores in German and Spanish, while the **Base** prompt performs better in French. The Mistral model shows weak

[3] gpt-4o : https://openai.com

[4] https://ollama.com

Model	Version	French	German	Spanish
gpt-4o	Adv COT	0.637	0.694	0.676
	Base	0.672	0.628	0.447
	Instruct	0.602	0.699	0.683
llama3	Adv COT	0.597	0.654	0.654
	Base	0.550	0.630	0.637
	Instruct	0.600	0.671	0.603
mistral	Adv COT	0.198	0.183	0.131
	Base	0.516	0.646	0.673
	Instruct	0.410	0.371	0.281
phi3	Adv COT	0.578	0.667	0.609
	Base	0.551	0.642	0.653
	Instruct	0.493	0.516	0.395
gemma	Adv COT	0.462	0.577	0.594
	Base	0.452	0.563	0.578
	Instruct	0.468	0.587	0.608
Supervised (our approach)	-	**0.799**	**0.791**	**0.800**

Table 5: F1 score comparison across different languages, models and prompting strategy for CWI 2018

Model	Version	English			French			Spanish			Japanese		
		P	**S**	**R2**	**P**	**S**	**R2**	**P**	**S**	**R2**	**P**	**S**	**R2**
gpt-4o	Base	0.736	0.735	0.153	0.505	0.509	0.207	0.659	0.643	0.149	0.595	0.621	0.241
	Instruct	0.759	0.665	0.142	0.545	0.555	0.205	0.667	0.645	0.194	0.421	0.404	0.381
	Adv COT	0.781	0.670	0.144	0.542	**0.554**	0.192	**0.680**	**0.654**	0.165	0.574	0.594	0.315
Phi3 3.8B	Base	0.230	0.207	0.229	-0.022	-0.036	0.299	0.233	0.214	0.221	0.110	0.210	0.259
	Instruct	0.414	0.444	0.166	0.093	0.090	0.250	0.276	0.288	0.171	0.244	0.290	0.219
	Adv COT	0.412	0.484	0.151	0.107	0.194	0.284	0.208	0.290	0.244	0.137	0.249	0.259
LLama3 8.0B	Base	0.374	0.418	0.379	0.136	0.146	0.363	0.265	0.278	0.317	0.129	0.158	0.403
	Instruct	0.555	0.519	0.147	0.180	0.170	0.229	0.382	0.376	0.152	0.252	0.253	0.184
	Adv COT	0.657	0.614	0.134	0.276	0.284	0.225	0.384	0.364	0.165	0.346	0.344	0.283
Mistral 7.2B	Base	0.461	0.489	0.394	0.166	0.149	0.309	0.400	0.397	0.355	0.125	0.122	0.388
	Instruct	0.612	0.579	0.139	0.212	0.188	0.220	0.540	0.529	0.152	0.259	0.256	0.153
	Adv COT	0.675	0.594	0.160	0.315	0.283	0.213	0.532	0.528	0.191	0.364	0.368	0.163
Gemma 9b	Base	0.123	0.169	0.482	0.038	0.063	0.433	0.175	0.180	0.384	0.137	0.135	**0.455**
	Instruct	0.322	0.360	0.320	0.185	0.189	0.311	0.395	0.407	0.227	0.260	0.270	0.279
	Adv COT	0.401	0.440	0.323	0.230	0.253	0.370	0.376	0.394	0.267	0.222	0.227	0.434
Supervised (our approach)	-	**0.87**	**0.80**	**0.72**	**0.56**	0.52	**0.31**	0.67	0.63	**0.45**	**0.66**	**0.67**	0.43

Table 6: Model performance comparison across different Languages and prompting strategies for MLSP 2024 (P:Pearson, S:Spearman, R2: R^2)

performance with the **Advanced COT** prompt but significantly improves with the **Base** prompt. These findings suggest that the effectiveness of the prompt type depends on both the model and the language, highlighting the need to adapt prompt strategies according to the language and the model in use.

We then tried to replicate the annotation process using LLMs for the CWI 2018 dataset where an instance is labeled as complex if any annotator finds the word complex, assigning a value of 1, otherwise 0. For this, given a prompt strategy, each LLM play the role of a single annotator. We will simulate the annotation process using LLMs, where 5 LLMs and 3 different prompt strategies generate a total of 15 annotations. If any of the annotations equals 1, the final annotation is set to 1, otherwise, it is set to 0. Thereafter, this method is called AT_LEAST_1. For comparison purposes, we also implemented a majority vote annotation method (thereafter VOTING_MAX), where the final label for a given instance corresponds to the most frequent label among the 15 LLM annotations.

Method	Fr	De	Es
AT_LEAST_1	0.45	0.56	0.57
VOTING_MAX	0.62	0.69	0.70

Table 7: The F1 scores for French, German, and Spanish using two voting strategies.

Table 7 shows that the score obtained using the single annotation method is significantly lower than that achieved by majority voting and is also lower than using a single LLM, gpt-4o (Base). However, the results from majority voting are relatively close to those of gpt-4o (Base) as seen in Table 5. It is also believed that VOTING_MAX performs better than AT_LEAST_1, as a single vote out of 15 can lead to errors if an underperform-

ing LLM votes 1, causing the instance to be annotated as 1. Majority voting helps mitigate this issue by considering the decision of the majority of the LLMs.

4.2.2 MLSP 2024

Figure 2 displays the Pearson correlation scores for each prompt type used for each LLM. It shows certain trends across different languages.

English: There is a progressive improvement from "base" to "advanced COT". This suggests better predictions in more complex configurations. gpt-4o notably performs better than other models with a score of 0.78. There is also a significant difference between the "base" and "instruct" prompts, while the gap between "instruct" and "advanced COT" is closer.

French and Spanish: gpt-4o shows continuous improvement, similarly to the trends observed in English, although the scores are more moderate. Nearly all models demonstrate improvement when going to more complex prompts.

Japanese: There are noticeable drops for complex prompts, which may indicate a sensitivity to the types of prompts used for Japanese.

Supervised Model (cf. table 6): The supervised multilingual approach described in section 3 outperforms in most cases our LLM prompting strategy, despite the lack of training data for French, Spanish, and Japanese. This has to be further investigated given the results of the best MLSP 2024 system based on a different prompting strategy with a different LLM.

The analysis of Pearson correlation scores for predicting lexical complexity (in Figure 2 and table 6) reveals a clear trend where the "advanced COT" (Chain of Thought) configurations generally achieve the best performance across various languages (French, English, and Spanish). This approach, which incorporates more detailed instructions or chain-of-thought reasoning, appears to better capture the nuances of lexical complexity compared to simpler "zero shot" and "one shot with instruction" approaches. This superiority is reflected in higher Pearson scores, indicating a stronger linear correlation between the predictions and actual values.

Observations made in English, French, and Spanish do not parallel those in Japanese, which presents a unique structure that includes mixed-script writing, the absence of clear word delimitation, and grammatical specificity. This underscores the necessity of using specially designed prompts for this language when predicting lexical complexity. The distinctive features of Japanese, such as kanji and grammatical particles, require a more targeted approach to effectively capture lexical complexity. By adapting prompts to the particularities of Japanese, it may be possible to enhance the accuracy of predictions by accounting for these variations.

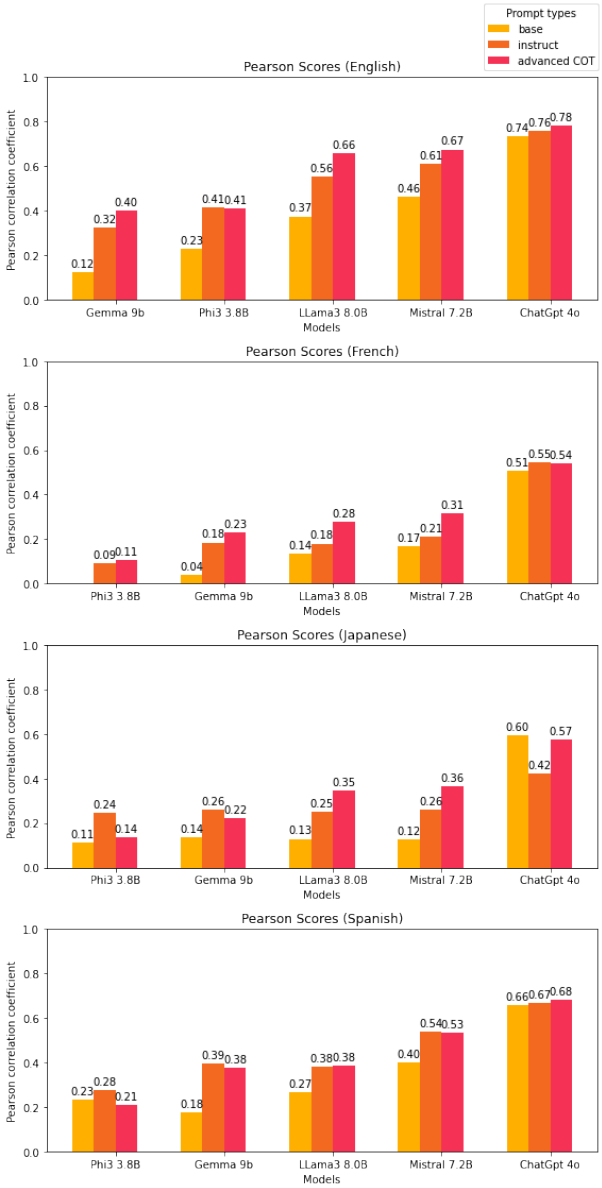

Figure 2: Correlation score for each llm based on the prompt type.

4.3 Are large language models (LLMs) a good alternative for multilingual lexical complexity prediction ?

While correlation scores are quite good for the MLSP 2024 dataset, R2 scores, which indicate the quality of prediction, suggest otherwise, cf. Table 6. Zero-shot generative models are not optimized for the specifics of a particular task. Although they can capture a linear relationship, they are less accurate in explaining the total variance of task-specific data, resulting in a lower R2 score. More specifically, asking an LLM to predict a score on a scale of five discrete values (0, 0.25, 0.5, 0.75, 1) penalizes it with respect to the way the dataset is annotated where each instance is annotated with a continuous value between 0 and 1 being the average of multiple human annotations. An intuitive method to address this issue with an LLM is to have it generate multiple outputs and then calculate the average, which might better disperse the data. Table 8 displays the average scores of gpt-4o with varying generation counts n (1, 10, 20, 30) for English. We have also included a model specifically trained for this task to facilitate comparison.

Models	P	S	R2
gpt-4o (n=1)	0.781	0.67	0.14
gpt-4o (n=10)	0.789	0.677	0.174
gpt-4o (n=20)	0.796	0.677	0.174
gpt-4o (n=30)	0.792	0.687	0.183
Supervised (ours)	**0.87**	**0.80**	**0.72**

Table 8: Performance metrics of gpt-4o vs Trained model for English (P:Pearson, S:Spearman, n:number of generations)

Table 8 indicates that the Pearson correlation scores do not increase significantly, with only slight improvements in the R2 score, which remains quite low compared to the 0.72 achieved by the model trained with a supervised approach.

What are the consequences of a low R2 score in this task? Let's take the example of the multilingual supervised model and gpt-4o (n=30) and analyze the scatter plot of each one's predictions. Graphs 3 and 4 illustrate the relationship between actual labels and the values predicted by two different models.

Graph 3 for gpt-4o shows a general trend that is well captured by the regression line, but with dispersion concentrated around the values (0, 0.25, 0.5, 0.75, 1), indicating larger prediction errors. On the other hand, Graph 4 displays a better fit between the predictions and the labels, with points more densely clustered around the regression line, suggesting increased accuracy and superior overall performance of the model.

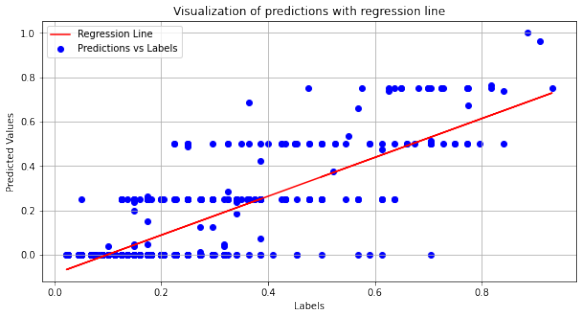

Figure 3: Scatter plot of gpt-4o's predictions (R=0.792,R2=0.183)

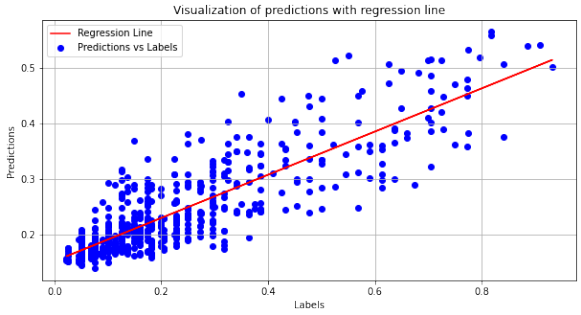

Figure 4: Scatter plot of trained model predictions (R=0.87,R2=0.72)

Graphs 5 and 6 display the dispersion of residuals e_i around the zero line.

$$e_i = y_i - \hat{y}_i$$

Each residual plot exhibits distinct characteristics reflecting the performance of two different prediction models. In Figure 6, the residuals are primarily concentrated around the mean prediction values (0.2 to 0.4), with a high density near the zero line, suggesting enhanced accuracy of the model within this range. A slight tendency to underestimate higher values is also observed, indicating a potential bias in the model. In contrast, Figure 5 shows a broader dispersion of residuals across all prediction values, with significant variations and distinct peaks at specific points (0.0, 0.2, 0.5, 0.8), suggesting a poorer fit of the model and reduced reliability, especially at the extremes.

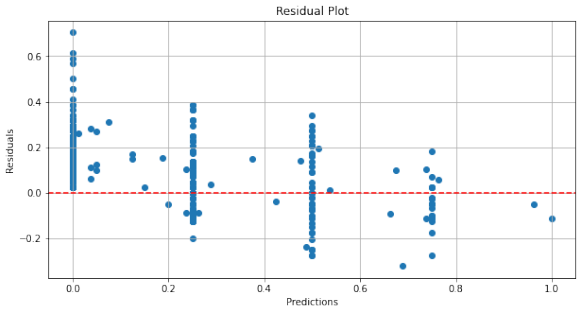

Figure 5: Residual plot for gpt-4o (R=0.792,R2=0.183)

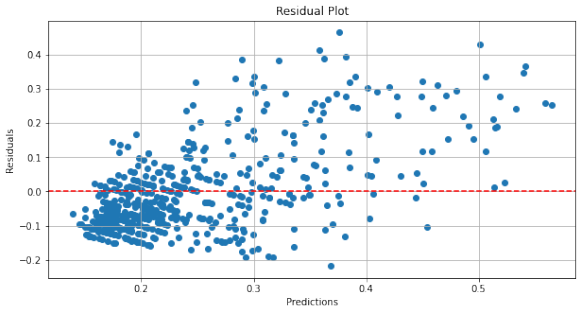

Figure 6: Residual plot for trained model (R=0.87,R2=0.72)

4.4 Can the R2 score be improved with large language models?

A good R2 score indicates better predictive quality of the model, with predicted values being closer to the actual values. In the dataset used, each instance is annotated by several evaluators who assess the complexity of a word on a five-point scale, with the final score being the average of these assessments. It is known that each evaluator may differ from each others in terms of level, and the score they assign also depends on their understanding of the instructions and their thought process before giving a score. Additionally, they can make errors. This process is very similar to that of LLMs: for example, we have seen in previous experiment that gpt-4o provides better results compared to others. Thus, we can imagine that the group of evaluators is analogous to a set of LLMs.

To test this hypothesis, we asked the five LLMs used in this experiment (gpt-4o , Llama3, Mistral, Phi3, and Gemma) to predict the score on a five-point scale (0, 0.25, 0.5, 0.75, 1) using the best prompt for English (advanced COT). We then calculated the average of these scores.

Table 9 presents the average and weighted average of LLM models compared to a single LLM and a model specifically trained for this task. The

Model	P	S	R2
One llm (gpt-4o)	0.781	0.670	0.144
Average All llm	0.710	0.673	0.450
Weighted average	0.792	0.717	0.610
Supervised (ours)	**0.870**	**0.800**	**0.720**

Table 9: Average and weighted average of large language models (LLMs) versus one LLM and a trained model.

weighted average is calculated by arbitrarily assigning weights to each LLM based on previously observed performances, as shown in Figure 2. The assigned weights are as follows: gpt-4o at 0.5, Mistral at 0.2, Llama3 at 0.1, Phi3 at 0.1, and Gemma at 0.1. These weights are used to determine if performance can be improved. Ideally and fairly, these weights should be derived from the training set and applied to the test set. As demonstrated in Table 9, the average score for all LLMs significantly improves the R2 score to 0.45, which is a substantial improvement compared to using a single LLM that scores 0.14. Performance further enhances with the use of a weighted average of 0.61, approaching the score of the model specifically trained for this task. These results strongly support our initial hypothesis. In conclusion, the use of multiple LLMs somewhat simulates the way data is annotated, providing better results in terms of R2 score.

5 Conclusion

In this study, we explored new methods aiming at enhancing the prediction of lexical complexity in a multilingual context using two distinct types of models: models trained specifically for the task in a supervised way and generative models not specifically trained for the task.

Regarding the supervised approach, our findings indicate that models trained on multiple languages outperform monolingual ones. Zero-shot models trained on multiple languages but the target one displayed variable performance compared with monolingual models. We also observed that data augmentation through automatic translation from English to the target language is feasible, although the required amount of augmentation instances may vary depending on the use case. Additionally, training a model directly from translated data is possible reasonable alternative, as we did

for French 3.

We further investigated the capabilities of generative models to predict lexical complexity on the MLSP 2024 dataset by varying the prompt strategy used. The results underscore the importance of prompt selection, with the "chain of thought" prompt proving particularly effective in English, French, and Spanish 2. However, this approach was not as effective for Japanese, a language that significantly differs from the others and might require a specially adapted prompt due to its unique complexity evaluation rules. Additionally, the findings for CWI 2018 reveal that the supervised approach outperforms our LLM prompting approaches. Majority voting further improved annotation quality.

Although generative models show good Pearson correlation scores, the quality of their predictions remains questionable, often due to very low R2 scores. To address this, we proposed an ensemble method using several generative models, which is akin to the human annotation process (cf. table 9). This opens new research perspectives.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

David Alfter. 2021. *Exploring natural language processing for single-word and multi-word lexical complexity from a second language learner perspective*.

David Alfter and I. Pilán. 2018. Sb@gu at the complex word identification 2018 shared task. pages 315–321.

David Alfter and Elena Volodina. 2018. Towards single word lexical complexity prediction. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 79–88.

S. Aroyehun, Jason Angel, D. Alvarez, and Alexander Gelbukh. 2018. Complex word identification: Convolutional neural network vs. feature engineering. pages 322–327.

Joachim Bingel and Johannes Bjerva. 2018. Cross-lingual complex word identification with multitask learning. pages 166–174.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. 2020. Spanish pre-trained bert model and evaluation data. In *PML4DC at ICLR 2020*.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Abhinandan Desai, Kai North, Marcos Zampieri, and C. Homan. 2021. Lcp-rit at semeval-2021 task 1: Exploring linguistic features for lexical complexity prediction. *ArXiv*, abs/2105.08780.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Taisei Enomoto, Hwichan Kim, Tosho Hirasawa, Yoshinari Nagai, Ayako Sato, Kyotaro Nakajima, and Mamoru Komachi. 2024. Tmu-hit at mlsp 2024: How well can gpt-4 tackle multilingual lexical simplification? In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 590–598.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Thomas François, Núria Gala, Patrick Watrin, and Cédrick Fairon. 2014. Flelex: a graded lexical resource for french foreign learners. In *International conference on Language Resources and Evaluation (LREC 2014)*.

Núria Gala, Thomas François, and Cédrick Fairon. 2013. Towards a french lexicon with difficulty measures: Nlp helping to bridge the gap between traditional dictionaries and specialized lexicons. In *eLex-Electronic Lexicography*.

Deepanway Ghosal, Yew Ken Chia, Navonil Majumder, and Soujanya Poria. 2023. Flacuna: Unleashing the problem solving power of vicuna using flan fine-tuning. *ArXiv*, abs/2307.02053.

Sian Gooding and E. Kochmar. 2019. Complex word identification as a sequence labelling task. pages 1148–1153.

Dhiman Goswami, Kai North, and Marcos Zampieri. 2024. Gmu at mlsp 2024: Multilingual lexical simplification with transformer models. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 627–634.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Abdelhak Kelious, Mathieu Constant, and Christophe Coeur. 2024. Complex word identification: A comparative study between ChatGPT and a dedicated model for this task. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3645–3653, Torino, Italia. ELRA and ICCL.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, and Xiaoxia Zhou. 2023. Better zero-shot reasoning with role-play prompting. *ArXiv*, abs/2308.07702.

Michal Konkol. 2016. Uwb at semeval-2016 task 11: Exploring features for complex word identification. pages 1038–1041.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq R. Joty, and J. Huang. 2023. A systematic study and comprehensive evaluation of chatgpt on benchmark datasets. pages 431–469.

S. Malmasi, M. Dras, and Marcos Zampieri. 2016. Ltg at semeval-2016 task 11: Complex word identification with classifier ensembles. pages 996–1000.

Kai North, Marcos Zampieri, and Matthew Shardlow. 2023. Lexical complexity prediction: An overview. *ACM Computing Surveys*, 55(9):1–42.

O. Ogundare, S. Madasu, and N. Wiggins. 2023. Industrial engineering with large language models: A case study of chatgpt's performance on oil gas problems. *ArXiv*, abs/2304.14354.

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.

Chunguang Pan, Bingyan Song, Shengguang Wang, and Zhipeng Luo. 2021. Deepblueai at semeval-2021 task 1: Lexical complexity prediction with a deep ensemble approach. pages 578–584.

Sandaru Seneviratne and Hanna Suominen. 2024. Anu at mlsp-2024: Prompt-based lexical simplification for english and sinhala. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 599–604.

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. 2024a. 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, Fernando Alva-Manchego, Riza Theresa Batista-Navarro, Stefan Bott, Saul Calderon-Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, and Anna Huelsing. 2024b. 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 2024)*, pages 571–589.

Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020. Complex: A new corpus for lexical complexity prediction from likert scale data. *arXiv preprint arXiv:2003.07008*.

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.

Anaïs Tack, Thomas François, Piet Desmet, and Cédrick Fairon. 2018. Nt2lex: A cefr-graded lexical resource for dutch as a foreign language linked to open dutch wordnet. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 137–146.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Giuseppe Vettigli and A. Sorgente. 2021. Compna at semeval-2021 task 1: Prediction of lexical complexity analyzing heterogeneous features. pages 560–564.

Rodrigo Wilkens, Patrick Watrin, Rémi Cardon, Alice Pintard, Isabelle Gribomont, and Thomas François. 2024. Exploring hybrid approaches to readability: experiments on the complementarity between linguistic features and transformers. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 2316–2331, St. Julian's, Malta. Association for Computational Linguistics.

Rui Yang, Lin Song, Yanwei Li, Sijie Zhao, Yixiao Ge, Xiu Li, and Ying Shan. 2023. Gpt4tools: Teaching large language model to use tools via self-instruction. *ArXiv*, abs/2305.18752.

Seid Muhie Yimam, Chris Biemann, S. Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Štajner, Anaïs

Tack, and Marcos Zampieri. 2018a. A report on the complex word identification shared task 2018. *ArXiv*, abs/1804.09132.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018b. 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.

Seid Muhie Yimam, Sanja Štajner, Martin Riedl, and Chris Biemann. 2017. Multilingual and cross-lingual complex word identification. pages 813–822.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, and Songfang Huang. 2023. How well do large language models perform in arithmetic tasks? *ArXiv*, abs/2304.02015.

George-Eduard Zaharia, Dumitru-Clementin Cercel, and M. Dascalu. 2020. Cross-lingual transfer learning for complex word identification. *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 384–390.

Marcos Zampieri, S. Malmasi, Gustavo Paetzold, and Lucia Specia. 2017. Complex word identification: Challenges in data annotation and system performance. *ArXiv*, abs/1710.04989.

A Appendix

1- Zero-shot prompt (base)

”””

You will be given a sentence and a word included in the sentence. Evaluate the complexity of the word in the context of the sentence, and provide a rating in scale of 0.0, 0.25, 0.5, 0.75, 1.0.

Sentence: ’{sentence}’

Word: ’{token}’

Complexity:

return only the number (0.0, 0.25, 0.5, 0.75, 1.0) that corresponds to the complexity of the word in context.

”””

2- One-shot prompt (instruct)

”””

You are a person without specialized knowledge or expertise in any specific field.You will receive a sentence containing a word, your task is to evaluate the word based on one metric.

Evaluation Criteria:

Complexity [0.0, 0.25, 0.5, 0.75, 1.0]: This measures how difficult it is to understand the word.

1. Carefully examine the sentence and the specified word to grasp the context in which it is used.

2. Assess the complexity of the word using the criteria provided

- 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.

Your personal knowledge of a word should not influence your rating. Instead, rate the word based on the understanding an average person might have

Example:

Sentence: ’The professor’s discourse was filled with intricate terminology that baffled the students.’ Word: ’discourse’.

For this example, 'discourse' might be rated as 0.25.

Please provide a complexity rating for the '{language}'word '{token}'.

Sentence: '{sentence}'

Word: '{token}'

return only the number (0.0, 0.25, 0.5, 0.75, 1.0) that corresponds to the complexity of the word.

""""

3- Chain-of-thought prompt (Advanced Cot)

""""

You are a person without specialized knowledge or expertise in any specific field.You will receive a sentence containing a word, your task is to evaluate the word based on one metric.

Evaluation Criteria:

Complexity [0.0, 0.25, 0.5, 0.75, 1.0]: This measures how difficult it is to understand the word.

Evaluation steps:

- **1.Understand the Context:** - Read the sentence and the word carefully to understand the context in which the word is used.
- **2. Analyze the Word's Frequency and Familiarity:** - Determine how commonly the word is used in everyday language. - Consider if the word is generally known by the average person or if it is specialized.
- **3. Evaluate the Morphological Complexity:** - Examine the structure of the word, including its length, composition, and any prefixes or suffixes.
- **4. Define the Word:** - Provide a definition of the word in its common usage. - Explain the specific meaning of the word in the given context.
- **5. Assess the Overall Complexity:** - Based on the analyses above, determine the complexity of the word using 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.
- **6. Assign a Complexity Rating:** - Based on your evaluation, assign a complexity rating to the word.

Your personal knowledge of a word should not influence your rating. Instead, rate the word based on the understanding an average person might have

Example:

Sentence: 'The professor's discourse was filled with intricate terminology that baffled the students.' Word: 'discourse'

1. Understand the Context: The word 'discourse' is used in a sentence discussing a professor's speech.

2. Analyze the Word's Frequency and Familiarity: 'Discourse' is somewhat specialized but can be understood by most people with some effort.

3. Evaluate the Morphological Complexity: 'Discourse' is a relatively long word but does not have complex prefixes or suffixes.

4. Define the Word: - Common usage: 'Discourse' means written or spoken communication. - Context-specific: In the sentence, 'discourse' refers to the professor's lecture.

5. Assess the Overall Complexity: Considering its moderate frequency, moderate morphological complexity, and clear context-specific meaning, 'discourse' might be rated as 0.25.

6. Assign a Complexity Rating: For this example, 'discourse' might be rated as 0.25.

Now, Please provide a complexity rating for the '{language}'word '{token}'.

Sentence: '{sentence}'

Word: '{token}'

return only the number (0.0, 0.25, 0.5, 0.75, 1.0) that corresponds to the complexity of the word.
"""

4- Zero-shot prompt (base-binary)

You will receive a sentence and a specific word from that sentence. Evaluate the complexity of the word within the context of the sentence and return 1 if the word is complex, or 0 if it is easy.

Sentence: 'sentence'

Word: 'token'

Complexity:

return only the complexity score: 1 or 0.

5- One-shot prompt (instruct-binary)

You are an individual without specialized knowledge or expertise in a specific area.

You will be given a sentence and a word included in the sentence.

Your task is to evaluate the complexity of the word in a binary format (0 or 1).

Please read and understand these instructions carefully. Keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Complexity (0, 1): Evaluate how difficult the word is to understand for an average person.

- 0: The word is simple and easily understandable by most people. - 1: The word is complex and may be difficult for an average person to understand.

Evaluation steps: 1. Read the sentence and word carefully to understand the context.

2. Determine the complexity of the word based on the criteria above.

3. Assign a complexity rating to the word.

Note: Your own familiarity with the word should not impact your rating. Base your judgment on an average person's understanding of the word.

Example:

Sentence: 'The professor's discourse was filled with intricate terminology that baffled the students.' Word: 'discourse'.

For this example, 'discourse' might be rated as 1.

Please assign a complexity rating to the 'lang' word.

Sentence: 'sentence'

Word: 'token'

Complexity:

return only the number (0 or 1) that corresponds to the complexity of the word.

6- Chain-of-thought prompt (Advanced COT-binary)

You are an individual without specialized knowledge or expertise in a specific area.

You will be given a sentence and a word included in the sentence.

Your task is to rate the word on one metric: complexity.

Please read and understand these instructions carefully. Keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Complexity (0 or 1): the complexity of a word in terms of how difficult the word is to understand.

Evaluation steps:

1. **Understand the Context:** - Read the sentence and the word carefully to understand the context in which the word is used.

2. **Analyze the Word's Frequency and Familiarity:** - Determine how commonly the word is used in everyday language. - Consider if the word is generally known by the average person or if it is specialized.

3. **Evaluate the Morphological Complexity:** - Examine the structure of the word, including its length, composition, and any prefixes or suffixes.

4. **Define the Word:** - Provide a definition of the word in its common usage. - Explain the specific meaning of the word in the given context.

5. **Assess the Overall Complexity:** - Based on the analyses above, determine the complexity of the word using the following criteria: - 0: The word is simple and easily understandable to most people. - 1: The word is complex and may be difficult to understand for the average person.

6. **Assign a Complexity Rating:** - Based on your evaluation, assign a complexity rating to the word.

Note: Your own familiarity with the word should not impact your rating. This should be based on an average person's understanding of the word.

Example:

Sentence: 'The professor's discourse was filled with intricate terminology that baffled the students.' Word: 'discourse'

1. Understand the Context: The word 'discourse' is used in a sentence discussing a professor's speech. 2. Analyze the Word's Frequency and Familiarity: 'Discourse' is somewhat specialized but can be understood by most people with some effort. 3. Evaluate the Morphological Complexity: 'Discourse' is a relatively long word but does not have complex prefixes or suffixes. 4. Define the Word: - Common usage: 'Discourse' means written or spoken communication. - Context-specific: In the sentence, 'discourse' refers to the professor's lecture. 5. Assess the Overall Complexity: Considering its moderate frequency, moderate morphological complexity, and clear context-specific meaning, 'discourse' might be rated as 0.

Now, apply this method to the given word and sentence.

Please assign a complexity rating to the 'lang' word.

Sentence: 'sentence'

Word: 'token'

Complexity:

Please return only the number (0 or 1) that corresponds to the complexity of the word. Do not include any additional information or explanations.