Effective and Efficient Word-type aware Chinese Lexical Simplification

Anonymous ACL submission

Abstract

In this paper, we address the task of Chinese lexical simplification (CLS), which aims to replace complex words in a given sentence with simpler alternatives of equivalent meaning. We propose an effective and efficient CLS method that combines small and large models in a complementary way based on the type of complex words. Specifically, we analyze the strengths and weaknesses of small models and ChatGPT. We find that ChatGPT performs well in simplifying in-dictionary common words and Chinese idioms, while small models struggle with them. Therefore, we propose an automatic knowledge distillation approach to fine-tune small models with in-dictionary words-oriented training data generated by ChatGPT. On the other hand, we find that both small models and ChatGPT have difficulties with out-of-dictionary (OOD) words. To address this issue, we use a retrieval-based interpretation augmentation strategy to enrich the input with relevant information obtained from external sources. With this strategy, both small models and ChatGPT can significantly improve their performance in simplifying OOD words. Finally, we introduce a simple controller that selects the best model or tool for each complex word according to its type. This hybrid approach can balance performance and cost and achieve better results than any single model.

1 Introduction

Lexical Simplification (LS) is the task of replacing complex words in a sentence with simpler alternatives while preserving their structure and original meaning. This task can improve the readability of the text to benefit a wide range of people, such as students (De Belder and Moens, 2010), non-native speakers (Paetzold and Specia, 2016), and individuals with cognitive impairments (Feng, 2009; Saggion, 2017). However, it is a challenging task that requires both linguistic knowledge and contextual awareness.

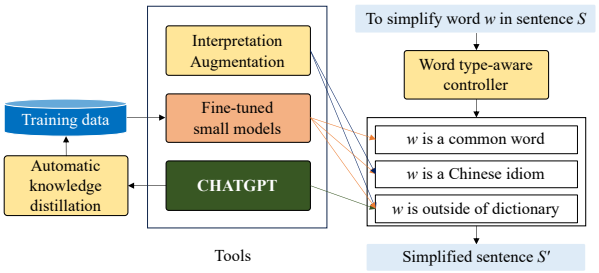

Figure 1: The general framework of the proposed word type-aware Chinese lexical simplification method.

This paper focuses on Chinese lexical simplification (CLS). One big barrier for CLS is the lack of enough training data. So recent work focuses on unsupervised methods based on pre-trained language models (PLMs), e.g., the state-of-the-art CLS system BERT-LS (Qiang et al., 2021) generates substitution candidates based on the pre-trained masked language model (MLM) BERT. Despite its simplicity, the model cannot fully understand the task, resulting in conservative word substitution and performance bottleneck.

We observe that the recent large generative pre-trained language models, such as ChatGPT (GPT-3.5 and GPT-4 (OpenAI, 2023)), can understand the task better through task instructions and a few demonstrations, while a medium model with 6B parameters still cannot get a satisfactory performance, reflecting the decisive role of model scale. However, the cost of training, maintaining, and invoking large language models is enormous. We face a trade-off between performance and cost when choosing between small or large models.

In this paper, we aim to improve a small model by learning from and collaborating with ChatGPT. We expect the final system to achieve competitive performance compared to ChatGPT while significantly reducing the inference cost. To accomplish this goal, we present the following contributions.

First, we conduct a thorough analysis of the un-

supervised CLS methods based on small, medium, and large language models, to gain a deeper understanding of the advantages and disadvantages of the current methods. We discover that ChatGPT has advantages in task understanding, reducing the loss of details and degree of information compared with small models. Linguistic resources can help small models obtain competitive performance for common words in the dictionary. All the models have a lot of room for improvement on out-of-dictionary (OOD) words.

Second, we propose a knowledge distillation framework called **PivotKD** for fine-tuning small models with the in-dictionary words-oriented training data, which are generated by ChatGPT. PivotKD samples pivot words from a dictionary, lets ChatGPT generate sentences containing pivot words, and replace them with alternatives belonging to different lexical difficulty levels in an automatic way. Evaluation shows that fine-tuning a model with 700m or 6B parameters can obtain superior performance compared with ChatGPT on simplifying common words and Chinese idioms.

Third, we propose a retrieval-based interpretation augmentation strategy for enhancing simplification on OOD words. We ask a search engine for the interpretation of a target complex word and use the interpretation for in-context learning. Based on this simple strategy, both ChatGPT and the fine-tuned small models gain large improvements in simplifying OOD words.

Finally, as shown in Figure 1, we propose a simple controller that selects the best model or tool for each complex word according to its type. The hybrid approach can balance performance and cost and get better results than any single model.

2 Related Work

Lexical Simplification is the process of replacing complex words in a given sentence with simpler alternatives of equivalent meaning (Paetzold and Specia, 2017b). This task makes contributions to performing text simplification focusing on lexical information and has wide applications in assisting readers with low language proficiency, cognitive impairments, or disabilities. Traditionally, lexical simplification mainly consists of a pipeline: the identification of complex words, the generation of substitution candidates, the selection of those candidates based on the context, and the ranking of the selected substitutes according to their simplicity.

Complex word identification aims to identify which word is considered complex in a sentence by a given target population (Shardlow, 2013; Yimam et al., 2018; Dehghan et al., 2022). In this paper, we do not pay attention to this aspect and assume that the target complex words are given by the users. Readers can refer to a recent survey for more information (North et al., 2023).

Knowledge-based methods Early lexical simplification research relied on lexical knowledge databases to generate substitutions (Carroll et al., 1998; Drndarevic and Saggion, 2012). However, the databases are expensive to construct and update, and have a limited word coverage.

Word embedding-based methods With the advent of deep learning, semantic similarity computation based on word embeddings has become a popular method for substitution generation and ranking (Paetzold and Specia, 2017a). But this method still suffers from the word coverage problem.

PLM-based methods Subsequently, pre-trained language models (PLMs) have been suggested for this task. For example, BERT-LS (Qiang et al., 2020) proposed an unsupervised method, employing BERT to generate substitutions for target complex words according to the encoding of the surrounding context. PromptLS (Vásquez-Rodríguez et al., 2022) found that fine-tuning PLMs can obtain better performance than the unsupervised setting. ConLS (Sheang et al., 2022) fine-tuned an encoder-decoder model T5 for substitution generation which naturally predicts simple words with multiple tokens. One challenge of fine-tuning is the scarcity of supervised training data for some languages, such as Chinese.

LLM-based methods Recently, large language models (LLMs) such as GPT-3 have been applied for lexical simplification through prompt learning-based methods. It shows that GPT-3 can understand the task and learn to predict based on task instructions and a few demonstrations and obtains good performance for the English language (Aumiller and Gertz, 2022). This indicates that LLMs already embed rich linguistic knowledge and have a strong in-context learning ability. The key is to find proper ways to guide LLMs to generate the required output. However, training, deploying, and applying LLMs is still very expensive.

This paper focuses on the Chinese language. Previous methods are mostly unsupervised (Qiang et al., 2021) due to the lack of enough training data, resulting in a performance bottleneck. Our

motivation is to build a system that can effectively combine small and large models to keep a balance between performance and cost. The following work is also relevant to our research.

Knowledge distillation Knowledge distillation (KD) aims to train a small student model to perform better by learning from a larger teacher model (Jianping et al., 2021). We expect to learn a better small model from powerful LLMs, such as ChatGPT. Since we can only access ChatGPT's predictions, we adopt a black-box KD method that fine-tunes the student model on the data generated by the teacher model (Kim and Rush, 2016). To get high-quality training data that are correct and diverse, we propose a pivot word-based approach for automatic data generation based on ChatGPT.

Retrieval-augmented LLMs OOD words are challenging to simplify because the model may have little knowledge about them. Motivated by recent work on retrieval-augmented LLMs (Lewis et al., 2020; Nakano et al., 2021), we propose a retrieval-based interpretation augmentation approach that dynamically brings in word interpretations from the web to enhance in-context learning.

3 Task, Data and In-Depth Analysis

In this section, we briefly introduce the task and the data, and analyze representative baselines which are based on BERT and LLMs.

3.1 Lexical Simplification Settings

An LS system first identifies complex words in a sentence and then generates candidate substitutions, which is known as substitute generation (SG). Considering complex word identification depends on a target population, we assume that a sentence and a target complex word are given following previous work (Qiang et al., 2021).

Formally, given a sentence $\mathbf{s}$ and a complex word w in $\mathbf{s}$, the task is to generate a simpler alternative v, a word or a group of words, to form a simpler sentence $\mathbf{s}'$, which is expected to be smooth, clear, and maintain the same meaning as $\mathbf{s}$.

3.2 Dataset and Metrics

3.2.1 Dataset

We use the publicly available Chinese lexical simplification dataset HanLS (Qiang et al., 2021). HanLS includes 524 sentences, each sentence contains a complex word from the advanced level of the Chinese Proficiency Test (Hanyu Shuiping Kaoshi, HSK), and each complex word has 8.51 annotated simple alternatives on average as reference answers.

3.2.2 Evaluation Metrics

Following previous work (Paetzold and Specia, 2016), we use precision and accuracy as metrics.

Precision (PRE): The proportion of predicted alternatives that are the original complex word itself or appear in the reference answers.

Accuracy (ACC): The proportion of predicted alternatives that are different from the target complex word and appear in the reference answers.

3.3 Baseline Systems

We adopt BERT-LS (Qiang et al., 2021) and three LLMs of various scales as the baselines. We try to analyze the behaviors and gain a deeper understanding of the advantages and disadvantages of these models.

3.3.1 BERT-LS

The input of BERT-LS is formed by concatenating the original sentence and its copy. The target complex word in the duplicate sentence is replaced with [MASK]. BERT takes the input and predicts the masked part as substitution candidates.

Notice that a Chinese word often involves more than one Chinese character, while BERT's tokenizer is based on characters so BERT-LS allows BERT to make predictions through different numbers of [MASK] tags, e.g., from 1 to 4. All predictions are added to a list of candidates. When the target word is in a Chinese synonymy thesaurus (Mei, 1983), its synonyms are used as substitution candidates. Finally, BERT-LS ranks these candidates with multiple sources of evidence including word embeddings, BERT scores, and word frequencies.

3.3.2 LLMs

We use ChatGPT (GPT-4-1106-preview API), ChatGLM2-6B (ChatGLM for short) (Du et al., 2022) and ChatYuan-large-v2 ($700m$ parameters, ChatYuan for short) (Xuanwei Zhang and Zhao, 2022). ChatGLM and ChatYuan are two open-source LLMs for dialogue, supporting both Chinese and English languages.

We explore the LLMs in a few-shot setting, where task instructions and demonstrations are included in the context. Figure 2 shows an example. We use three demonstrations in this paper. For ChatGPT, we extract predictions from its responses. For ChatGLM and ChatYuan, we first extract the

Instruction	任务是将句子中给定的难词(由#标记)替换为一个简单的词或短语，同时保持句子的结构和意思不变并尽量流畅。 (The task is to replace the complex word (marked by #) in the sentence with a simple word or phrase, while keeping the structure and meaning of the sentence unchanged and as smooth as possible.)
Input 1	练习与理解不是#截然#对立的，而是相辅相成的。 (Practice and understanding are not #thoroughly# opposed but complement each other.)
Response	练习与理解不是#完全#对立的，而是相辅相成的。 (Practice and understanding are not #completely# opposed but complement each other.)
Input 2	他#呕心沥血#写了这本书。
Response	[Let LLM generate the response]

Figure 2: An example of instruction and demonstration design for prompting LLMs for CLS.

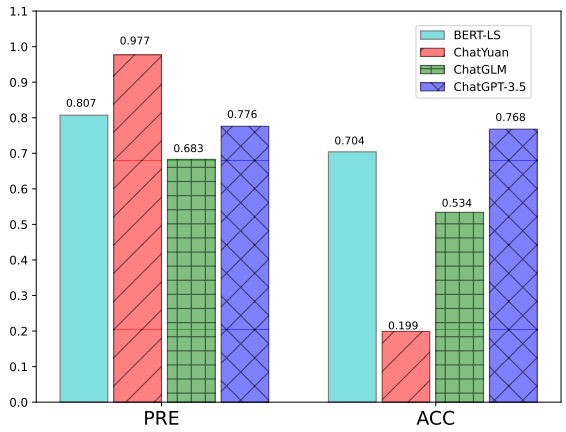

Figure 3: Overall results of BERT-LS and three LLMs in the few-shot setting.

top 10 candidates from model predictions and then re-rank them in the same way as BERT-LS.

3.4 Analysis and Discussion

3.4.1 Overall Results

Figure 3 shows the overall results of these systems. ChatYuan, BERT-LS, and ChatGLM are models that do not perform much simplification, as shown by their high PRE scores and low ACC scores. BERT-LS outperforms the other two models, possibly because for LS, MLM is a better fit for dialogue-oriented models in a few-shot setting. ChatGPT, on the other hand, simplifies almost all complex words and achieves high scores in both RPE and ACC, indicating its effectiveness.

Based on task instructions and demonstrations, ChatYuan and ChatGLM have difficulty in understanding the task. ChatGLM has a slight edge over ChatYuan, which may be attributed to its larger scale. The unsupervised BERT-LS also fails to grasp the LS task, despite using external resources to re-rank candidates and compensate for its poor task understanding. ChatGPT, however, demonstrates a strong task understanding and a good performance in lexical simplification.

Models	Common		Idioms		OOD	
	PRE	ACC	PRE	ACC	PRE	ACC
BERT-LS	86.8	76.9	70.8	**41.7**	34.0	28.3
ChatYuan	**98.4**	22.4	**91.7**	8.30	**94.3**	3.80
ChatGLM	72.0	58.3	62.5	29.2	39.6	22.6
ChatGPT	81.6	**80.7**	29.2	29.2	66.0	**66.0**

Table 1: Detailed results of BERT-LS and three LLMs on simplifying 3 types of complex words.

Original sentence	小猪#似懂非懂#，心想幸福怎么会是我的尾巴呢？ The little pig #seemed to understand but didn't really understand#, thinking how could happiness be my tail?
BERT-LS	小猪#不解#，心想幸福怎么会是我的尾巴呢？ The little pig #puzzled#, thinking how could happiness be my tail?
ChatGPT	小猪#有些糊涂#，心想幸福怎么会是我的尾巴呢？ The little pig #was a bit confused#, thinking how could happiness be my tail?

Figure 4: The outputs of BERT-LS and ChatGPT on simplifying a Chinese idiom.

3.4.2 Analysis

We analyze the relation between the models' performance and the types of complex words. Specifically, we divide the complex words into 3 types:

- **Common words**: Refer to non-idiomatic words included in the dictionary Xinhua Zidian, which covers more than $320k$ words.
- **Chinese idioms**: Idioms or Chengyu, are an essential part of the Chinese language. They are usually composed of four Chinese characters and often express a moral or a lesson in a concise and elegant way.
- **Out-of-dictionary (OOD) words**: Refer to the words excluded in Xinhua Zidian, many of which are new words or internet terms.

Table 1 shows that ChatGPT outperforms BERT-LS, ChatYuan, and ChatGLM in simplifying common words, but lags behind BERT-LS on Chinese idioms. ChatGPT also has a remarkable advantage in simplifying OOD words, although none of the models achieve satisfactory results.

We compare the predictions of ChatGPT and BERT-LS in simplifying Chinese idioms and discover that ChatGPT's performance in simplifying

Chinese idioms is underrated because it generates phrases instead of single words as the reference answers do. For example, in Figure 4, ChatGPT produces a more understandable and fluent substitution than BERT-LS. BERT-LS only replaces the idiom with a single word, which may lose some descriptive details and degree of information.

Original sentence	我最近网上冲浪的时候总能刷到好多#镁铝#哦！ I always see a lot of #magnesium aluminum# when I surf the internet recently!
BERT-LS	我最近网上冲浪的时候总能刷到好多#金属#哦！ I always see a lot of #metal# when I surf the internet recently!
ChatGPT	我最近网上冲浪的时候总能刷到好多#美食#哦！ I always see a lot of #gourmet food# when I surf the internet recently!

Figure 5: The outputs of BERT-LS and ChatGPT on simplifying an OOD word.

Simplifying OOD words is a challenge for all models. Figure 5 shows an example. The term "magnesium-aluminum" is a Chinese internet slang that sounds like "beauty" and refers to beautiful women. Neither BERT-LS nor ChatGPT can produce a suitable answer, probably because they have limited knowledge of these OOD words.

In summary, we discover the following observations through the in-depth analysis:

- **Task understanding**: Without enough supervision, the small and medium models, BERT-LS, ChatYuan, and ChatGLM, could not grasp the task well. ChatGPT shows a much better understanding of the task and performs well.
- **Sensitive performance to the type of complex words**: The difficulty of the simplification task depends on the types of complex words. BERT-LS and ChatGPT perform well in simplifying common complex words. For Chinese idiom simplification, ChatGPT has an advantage in preserving more descriptive details and degree of information. Simplifying OOD words is a challenge for all models.

ChatGPT performs the best but it is also expensive, which creates a trade-off between performance and cost. Moreover, it cannot deal with OOD words either. These factors lead us to the following research questions:

- **RQ 1**: How to use ChatGPT effectively as a teacher model to improve the task understanding and performance of smaller models on simplifying in-dictionary words?
- **RQ 2**: What are the effective strategies to enhance the performance of both large and small models in simplifying OOD words?
- **RQ 3**: What is the optimal way to integrate small and large models to achieve a trade-off between performance and cost?

4 The Proposed Method

We propose a framework as shown in Figure 1. It has 3 modules: automatic knowledge distillation from ChatGPT, retrieval-based interpretation augmentation, and a word type-aware controller.

4.1 Automatic Knowledge Distillation

We aim to create a high-quality CLS training dataset by distilling ChatGPT. We expect the generated sentences should be correct in spelling and grammar, cover diverse topics, and have accurate substitutions. However, since there are many factors to consider, it is unavoidable to bring in bias.

We propose an automatic knowledge distillation strategy named **PivotKD**, which only relies on ChatGPT and does not need any human intervention. Figure 6 illustrates its main workflow.

4.1.1 Pivot Word Sampling

Our analysis shows that ChatGPT works well on common words and idioms but poorly on OOD words. So we should avoid OOD words in data generation. Therefore we sample words from the dictionary Xinhua Zidian and call the sampled words *pivot words*. We limit the word to be a noun, verb, adjective, adverb or idiom. To enhance diversity, each word can be sampled at most once. The pivot words would be used for pivot sentence generation.

4.1.2 Pivot Sentence Generation

Given a pivot word, we ask ChatGPT to generate a sentence containing the word. The generated sentence would be used as the target sentence. This manner has the following benefits depending on the strengths of ChatGPT: 1) ChatGPT can generate correct and smooth sentences, avoiding spelling and grammar errors that often occur in data collected from the web or existing corpus; 2) ChatGPT can generate sentences covering diverse topics since we do not limit the topics in sampling pivot words and sentence generation, we can assume that the generated dataset is topic independent.

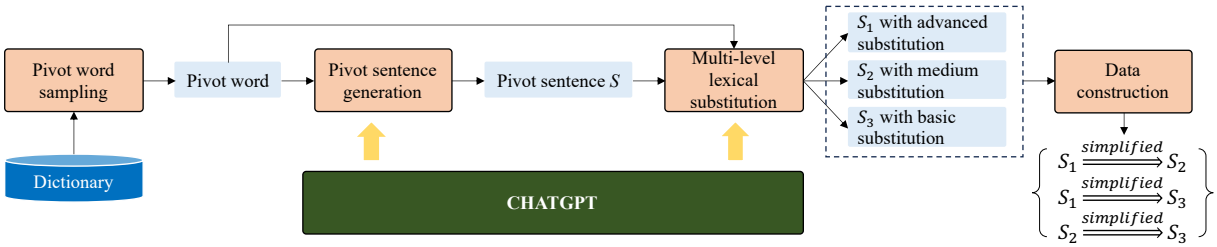

Figure 6: The main workflow of the PivotKD framework for generating CLS data based on ChatGPT.

任务是将句子中给定的难词(由#标记)替换为简单的词或短语，同时保持句子的结构和意思不变并尽量流畅。词语难度可以分为高级、中级和基本三个等级，请为每个等级分别生成n个替换词。
(We need to find a simpler word or phrase for the complex word, marked by #, in a sentence and keep the same structure and meaning of the original sentence while making the new sentence smooth. The word difficulty can be divided into three levels: advanced, intermediate, and basic. Please generate n replacement words for each level.)

Figure 7: An instruction for 3-level lexical substitution.

4.1.3 Multi-level Lexical Substitution

After the above two steps, we have a pivot sentence and a pivot word. Now, we let ChatGPT generate substitutions belonging to three difficulty levels to replace the pivot word in the generated sentence. We define the three lexical difficulty levels as *advanced, medium*, and *basic*. We convey the requirements to ChatGPT through instructions as shown in Figure 7, and assume that ChatGPT itself can understand lexical difficulty.

4.1.4 Data Construction

For one pivot word, we use ChatGPT to generate n simpler alternatives for each difficulty level. Given the $3n$ sentences, we can construct a set of sentence pairs as training data based on the lexical difficulty levels of the substitutions. Specifically, a complex-to-simple sentence pair $(\mathbf{s}, \mathbf{s}')$ can be constructed if the lexical difficulty level of the substitution in $\mathbf{s}$ is higher than the substitution in $\mathbf{s}'$.

Notice that the sentence pair does not necessarily contain the pivot word. The pivot word only plays the role of starting point for the automatic knowledge distillation and data construction process, which implies the meaning of *pivot*.

4.1.5 Instruction Fine-tuning

We conduct instruction fine-tuning with ChatYuan and ChatGLM. We fine-tune all parameters of ChatYuan and use LoRA (Hu et al., 2021) to fine-tune ChatGLM.

The training data is in the sequence-to-sequence style based on the constructed sentence pairs $\{(\mathbf{s}, \mathbf{s}')\}$. The input includes a task instruction, the same as the one shown in Figure 2, the target sentence $\mathbf{s}$, and the target complex word marked with a tag # in $\mathbf{s}$, while the output is the corresponding simplified sentence $\mathbf{s}'$ with the word substitution marked with # as well.

4.2 Retrieval-based Interpretation Augmentation

To ensure the quality of the generated data, we have focused on common words. However, simplifying OOD words is very challenging even for ChatGPT. To tackle this problem, we propose a retrieval-augmented strategy by looking for the interpretation of a target complex word from the web.

Retrieving Word Interpretation Many OOD words are new words or internet slang that the pre-trained models may have limited knowledge of them. But there are usually interpretations for these words on the web. Therefore we utilize the Baidu search engine to fetch search results for the query "*What does the word* [***complex word***] *mean?*" and extract the content of the top k snippets as the interpretation.

Injecting Interpretation for Inference We use retrieval-based interpretation augmentation to provide additional context after the task instructions for ChatGPT or the fine-tuned models. So it is flexible and can be easily integrated as a plug-in.

4.3 Word-type aware Controlled Inference

Now we have several models and tools to handle CLS: ChatGPT, fine-tuned small models, and retrieval-based interpretation augmentation. We aim to find an effective and efficient way to integrate these modules.

Our solution is based on the observation that the performance of CLS is sensitive to the type of complex words. We prefer to use small models as the basic model and ask for help from ChatGPT

Models	Common		Idioms		OOD		All	
	PRE	ACC	PRE	ACC	PRE	ACC	PRE	ACC
BERT-LS	86.8	76.9	70.8	41.7	34.0	28.3	80.7	70.4
ChatGPT	81.6	80.7	29.2	29.2	66.0	66.0	77.6	76.8
+RIA	-	-	-	-	**79.3**	**79.3**	-	-
ChatGLM (frozen)	72.0	58.3	62.5	29.2	39.6	22.6	68.3	53.4
ChatGLM (fine-tuning with LoRA)	83.0	82.1	**66.7**	**66.7**	60.4	58.5	80.0	79.0
+RIA	**83.6**	**82.5**	58.3	58.3	**64.2**	**62.3**	**80.5**	**79.4**
ChatYuan (frozen)	*98.4*	22.4	*91.7*	8.3	*94.3*	3.8	*97.7*	19.9
ChatYuan (fine-tuning all parameters)	85.2	80.5	**75.0**	**70.8**	56.6	45.3	81.8	76.5
+RIA	**86.3**	**82.5**	**75.0**	**70.8**	**68.0**	**62.3**	**84.0**	**80.0**
Hyb-CLS	**86.3**	**82.5**	**75.0**	**70.8**	**79.3**	**79.3**	**85.1**	**81.6**

Table 2: System comparisons on HanLS. ChatGLM and ChatYuan are experimented with frozen, fine-tuning, and hybrid settings. RIA indicates utilizing retrieval-based interpretation augmentation during inference.The results with the highest accuracy are bolded, and the best results obtained by small models are marked with underlines.

or retrieval-based interpretation augmentation to handle Chinese idioms or OOD words. We design a rule-based word-type aware controller for deciding a proper inference strategy. We will discuss the optimal strategies for different types of words in the evaluation section.

5 Evaluation

5.1 Experimental Settings

For PivotKD, we sampled 5,000 pivot words from Xinhua Zidian. We avoid using the complex words in HanLS as pivot words. Since the sentences are fully generated by ChatGPT, there is also no overlap with HanLS.

The multi-level lexical substitution module generates $n = 1$ substitution for each lexical difficulty level. Finally, we collect 8,962 sentence pairs, covering 4,269 distinct substitutions. More details about the data can be seen in Appendix A.

We conduct a human evaluation on 500 samples from the augmented dataset. For each complex word and simplified substitution pair, we let two persons judge whether the relative lexical difficulty between them are *clearly reasonable*, *hard to distinguish*, or *contradiction or irrelevant*. The proportions of the 3 options are 70%, 25%, and 5% respectively, indicating the quality of the dataset produced by PivotKD is acceptable.

We fine-tune ChatYuan for 1 epoch and ChatGLM with LoRA for 3 epochs. ChatYuan is based on T5, therefore the fine-tuned ChatYuan can be seen as the re-implementation of ConLS (Sheang et al., 2022). Detailed parameter settings are described in Appendix B. For retrieval-based interpretation augmentation, we use the top $k = 1$ snippet since it already obtains satisfactory performance.

5.2 Experimental Results

5.2.1 Auto-Evaluation

Table 2 shows the overall results and specific results on three types of complex words by BERT-LS, ChatGPT, ChatGLM, and ChatYuan and some variants. We can see some trends:

(1) **The effects of PivotKD** The fine-tuned ChatGLM and ChatYuan obtain competitive overall performance compared with ChatGPT and greatly outperform BERT-LS. For example, ChatGLM gets a 80.0 PRE score and a 79.0 ACC score, outperforming ChatGPT, while ChatYuan gets a 81.8 PRE score and a 76.5 ACC score.

The results demonstrate the effectiveness of PivotKD. The small models (e.g., ChatYuan) and medium models (e.g., ChatGLM) can benefit from supervised instruction fine-tuning based on the automatically generated data. The fine-tuned models understand the task much better and gain large improvements compared with the frozen models.

The effect of increasing the number of training samples on the performance of the fine-tuned ChatYuan and ChatGLM models is illustrated in Figure 8. Generally, the performance of the models can be enhanced by increasing the number of training samples. ChatGLM can reach a steady performance using relatively fewer samples, while the performance of ChatYuan has a consistent improvement with the increase of the training samples, indicating that smaller models may need more training data.

(2) **The effects of retrieval-based interpretation augmentation (RIA)** Table 2 shows that RIA can significantly enhance the ability of all three

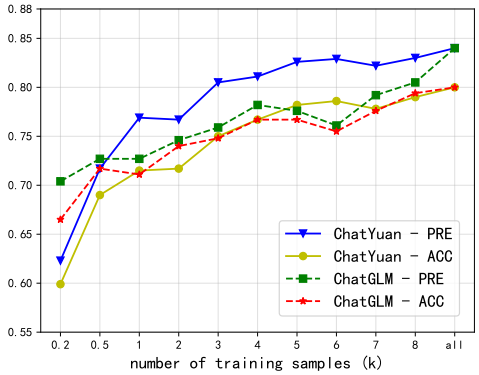

Figure 8: The effects of the number of training samples for fine-tuning ChatYuan and ChatGLM.

models to simplify OOD words, confirming that the retrieved word interpretation provides useful information for OOD words. ChatGLM and ChatYuan achieve similar performance to ChatGPT, while RIA boosts ChatGPT's PRE and ACC scores by a large margin of 16%.

(3) **Performance on different word types** The performance of the fine-tuned models on different types of complex words is still very volatile, better on common words than Chinese idioms and OOD words. RIA also has different impacts on ChatGLM and ChatYuan. For example, ChatYuan gains further improvements for common words with RIA, while ChatGLM almost remains the same. Perhaps ChatGLM already has enough knowledge of the common words.

(4) **A hybrid approach** Since the performance is word-type sensitive, we can apply different settings for different types of words. We call the strategy **Hyb-CLS** (a hybrid approach for CLS). Specifically, we use the fine-tuned ChatYuan (700m) with RIA for common words and Chinese idioms, while call for ChatGPT to handle OOD words. Table 2 shows that the hybrid approaches obtain further improvements compared with any single model.

5.2.2 Human Evaluation

The system outputs may be reasonable but outside the reference answers. So we conduct a human evaluation. We sample 20 common words, 20 Chinese idioms, and 20 OOD words from HanLS. Three raters rate the mixed outputs of different systems according to the following criteria:

- 4 points: The substitution is simpler and has the same meaning as the target complex word without any information loss, and the resulting sentence is smooth.
- 2 points: The substitution is simpler and has the same meaning as the target word, but there is a loss of information in terms of details and degree, or the output is not so smooth.
- 0 points: The substitution is not simpler or its meaning is different from the target word.

Models	Common	Idioms	OOD	All
BERT-LS	3.17	1.23	0.6	1.67
ChatGPT + RIA	**3.70**	**3.17**	**2.60**	**3.16**
ChatGLM	3.37	2.60	1.93	2.63
+RIA	3.50	2.83	2.53	2.95
ChatYuan	3.37	2.60	1.50	2.49
+RIA	3.43	2.83	2.2	2.82

Table 3: Human evaluation of three models in different settings. The rating ranges from 0 (worst) to 4 (best).

Table 3 shows the averaged human evaluation results. We can see that ChatGPT still has an advantage in simplifying idioms and OOD words, indicating the strong ability of very large language models. The fine-tuned small models achieve similar performance and the performance is also close to ChatGPT. RIA is verified to be effective as well. The human evaluation confirms that with proper manipulation of the fine-tuned small models and very large models, it is possible to keep a balance between performance and cost.

6 Conclusion

This paper presents a word-type aware approach for Chinese lexical simplification. The core idea is to consider the types of complex words to effectively and efficiently combine small and large language models. We find that ChatGPT performs well in simplifying in-dictionary complex words and idioms. So we propose an automatic knowledge distillation framework called PivotKD to generate training data with ChatGPT for fine-tuning small models. The results show that the fine-tuned small models can outperform ChatGPT in simplifying such common words. Besides, we observe that both small and large models face challenges in simplifying OOD words. We propose a retrieval-based interpretation augmentation strategy, which significantly improves the simplification of OOD words for all models. Therefore, we can control the inference strategy according to the type of complex words, which efficiently combines small and large models and helps to obtain the best performance.

7 Limitations

There are three possible limitations of this work. First, our evaluation is based on the HanLS dataset, which is limited in size and coverage. We plan to extend the dataset. Second, we assume that ChatGPT understand the lexical difficulty levels, but we verify this assumption by analyzing the relative lexical difficulty between a pair of words in the generated data. More detailed and specially designed probing analysis can be conducted. Third, this paper focuses on Chinese lexical simplification, but the proposed method can be potentially applied to other languages. We plan to address these limitations in the future work.

References

Dennis Aumiller and Michael Gertz. 2022. Unihd at tsar-2022 shared task: Is compute all we need for lexical simplification? In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 251–258.

John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of english newspaper text to assist aphasic readers. In *Proceedings of the AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 7–10. Citeseer.

Jan De Belder and Marie-Francine Moens. 2010. Text simplification for children. In *Prroceedings of the SIGIR workshop on accessible search systems*, pages 19–26. ACM; New York.

Mohammad Dehghan, Dhruv Kumar, and Lukasz Golab. 2022. GRS: Combining generation and revision in unsupervised sentence simplification. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 949–960, Dublin, Ireland. Association for Computational Linguistics.

Biljana Drndarevic and Horacio Saggion. 2012. Towards automatic lexical simplification in spanish: An empirical study. In *PITR@ NAACL-HLT*, pages 8–16.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Lijun Feng. 2009. Automatic readability assessment for people with intellectual disabilities. *ACM SIGACCESS accessibility and computing*, pages 84–91.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Gou Jianping, Yu Baosheng, Stephen J Maybank, and Tao Dacheng. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Jiaju Mei. 1983. *Synonymy Thesaurus of Chinese Words*. Shanghai Lexicographical Publishing House, Shanghai.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

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

OpenAI. 2023. Gpt-4 technical report.

Gustavo Paetzold and Lucia Specia. 2016. Unsupervised lexical simplification for non-native speakers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Gustavo Paetzold and Lucia Specia. 2017a. Lexical simplification with neural ranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 34–40.

Gustavo H Paetzold and Lucia Specia. 2017b. A survey on lexical simplification. *Journal of Artificial Intelligence Research*, 60:549–593.

Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2020. Lexical simplification with pretrained encoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8649–8656.

Jipeng Qiang, Xinyu Lu, Yun Li, Yunhao Yuan, and Xindong Wu. 2021. Chinese lexical simplification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1819–1828.

Horacio Saggion. 2017. Automatic text simplification. *Synthesis Lectures on Human Language Technologies*, 10(1):1–137.

Matthew Shardlow. 2013. A comparison of techniques to automatically identify complex words. In *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*, pages 103–109, Sofia, Bulgaria. Association for Computational Linguistics.

Kim Cheng Sheang, Daniel Ferrés, and Horacio Saggion. 2022. Controllable lexical simplification for english. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 199–206.

Laura Vásquez-Rodríguez, Nhung Nguyen, Matthew Shardlow, and Sophia Ananiadou. 2022. Uom&mmu at tsar-2022 shared task: Prompt learning for lexical simplification. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 218–224.

Liang Xu Xuanwei Zhang and Kangkang Zhao. 2022. Chatyuan: A large language model for dialogue in chinese and english. https://github.com/clue-ai/ChatYuan.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A report on the complex word identification shared task 2018. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 66–78, New Orleans, Louisiana. Association for Computational Linguistics.

Attribute	Value
Sentence pairs	8,962
Avg. length of sentences	22.38
Distinct substitutions	4,269
Common words	4,186
Avg. length of substitutions	2.04

Table 4: Basic statistics of the augmented dataset via PivotKD.

A Details in Dataset Construction

We sampled 5,000 pivot words for data generation. After constructing sentence pairs according to the difficulty levels of the substitutions, we use some rules to further reduce noise.

Firstly, we excluded substitutions that exist in the target complex word list of HanLS, thus there is no overlap between the augmented data and the test data. Secondly, we constrain that for the complex word in each constructed sentence pair should be in the Xinhua Zidian dictionary.

Some basic statistics of the final dataset for fine-tuning the small models are shown in Table 4. Table 5 shows a constructed sentence pairs and the corresponding training sample.

A sentence pair	
Complex	他#悄无声息#地走进房间，以免吵醒熟睡中的孩子。 (He slipped into the room #stealthily#, so as not to wake the sleeping child.)
Simple	他#偷偷#地走进房间，以免吵醒熟睡中的孩子。 (#quietly#.)
A training sample	
Prompt	他#悄无声息#地走进房间，以免吵醒熟睡中的孩子。 (Same as the sentence above) 你的任务是将句子中给定的难词#悄无声息#替换为一个简单的词或短语，同时保持句子结构和意思不变并尽量流畅。 (The task is to replace the complex word #stealthily# in the sentence with a simple word or phrase, while keeping the structure and meaning of the sentence unchanged and as smooth as possible.)
Response	他#偷偷#地走进房间，以免吵醒熟睡中的孩子。 (#quietly#.)

Table 5: An example of a constructed sentence pair and the corresponding training sample.

B Parameter Settings

The hyper-parameters used for fine-tuning ChatYuan-large-v2 and ChatGLM2-6B are listed in Table 7 and Table 8 respectively.

We set the value of the temperature parameter of ChatGPT API as 0 because we emphasize the generation quality and control the diversity through pivot words.

C Human Rating

We conducted human evaluation for ChatGLM, ChatYuan, ChatGPT and BERT-LS. We sampled 20 words for each type of complex word, merging the predictions of the models for human evaluation.

Settings	Value
GPU	Nvidia A6000
GPU memory	48 GB
CPU	AMD EPYC 7542
OS	Ubuntu 20.04.5 LTS
Pytorch version	1.31.1
CUDA version	11.6

Table 6: Infrastructure for conducting our experiments.

We set the ratings to be 0,2,or 4 according to the criteria introduced in the main content. We employed three raters who are students in a normal university. They are volunteers and unaware of the model information of these predictions. We reported the average rating for each prediction. The mean variance of the ratings between different raters is 0.55. Table 9 demonstrate two examples of predictions and human ratings.

Hyper-parameters	Value
max_seq_length (encoder)	512
max_seq_length (decoder)	512
num_epoch	1
learning_rate	5e-5
scheduler	cosine
batch_size	16
gradient_accumulation_steps	1

Table 7: Hyper-parameter settings used for fine-tuning ChatYuan-large-v2.

Hyper-Parameters	Value
max_seq_length (encoder)	512
max_seq_length (decoder)	512
num_epoch	3
learning_rate	5e-5
scheduler	cosine
batch_size	16
gradient_accumulation_steps	1
lora_rank	8
lora_alpha	32
lora_dropout	0.1

Table 8: Hyper-parameter settings used for fine-tuning ChatGLM2-6B.

Sentence	加强民族团结， #捍卫#国家完整。 (Enhance ethnic unity and #safeguard# national integrity.)
Prediction	加强民族团结， #维护(maintain)#国家完整。
Rater A	4
Rater B	4
Rater C	2
Sentence	那个年代，汤姆有一点叛逆， 有一个梦想就是去当 #绿林好汉#. (In that era, Tom was a bit rebellious, and he had a dream of becoming an #outlaw hero#.)
Prediction	那个年代，汤姆有一点叛逆， 有一个梦想就是去当 #土匪(bandit)#.
Rater A	2
Rater B	0
Rater C	2

Table 9: Two examples of predictions and human ratings.