# SQUREL at TSAR 2025 Shared Task: CEFR-Controlled Text Simplification with Prompting and Reinforcement Fine-Tuning

**Daria Sokova**[*] University of Surrey d.sokova@surrey.ac.uk
**Anastasiia Bezobrazova**[*] University of Surrey a.bezobrazova@surrey.ac.uk
**Constantin Orăsan** University of Surrey c.orasan@surrey.ac.uk

## Abstract

This paper summarises the submissions of our team to the TSAR 2025 Shared Task on Readability-Controlled Text Simplification, which aims to create text simplifications that balance reduced linguistic complexity, meaning preservation, and fluency while meeting a predefined target readability level. In this work, we proposed two different methods for CEFR-controlled text simplification: a setup which employed reinforcement fine-tuning of large language models (LLMs) and a conservative lexical pipeline which relied on prompting LLMs to simplify sentences.

## 1 Introduction

Readability-controlled text simplification (RCTS) aims to generate simplifications within specified readability levels while preserving the original meaning (Barayan et al., 2024). While instruction-tuned LLMs have been shown to be useful in zero-shot RCTS, balancing readability control and meaning preservation remains challenging (Farajidizaji et al., 2023). This paper presents our participation in the TSAR 2025 Shared Task on Readability-Controlled Text Simplification (Alva-Manchego et al., 2025). The Shared Task invited participants to simplify sentences at the B1 and A2 CEFR levels. Due to the scarcity of labelled data for supervised training, we explore approaches that do not require high-quality parallel data labelled with CEFR levels. We propose two methods for producing readability-controlled simplifications:

1. **Reinforcement Fine-Tuning with Group Relative Policy Optimization**: this method proposes a fine-tuning strategy aimed at conditioning an open-weight LLM to produce simplifications that balance CEFR-level accuracy and meaning preservation.
2. **Lexical Simplification**: which aims to produce simplifications corresponding to the specified CEFR level through careful lexical substitution with the help of LLM prompting.

The rest of the paper is structured as follows. Section 2 describes the two methods we employed to obtain the outputs submitted to the Shared Task, followed by the discussion of the results in Section 3. We also describe limitations of our work and provide concluding observations. The prompts and examples from the outputs can be found in the Appendix A.

## 2 Methods Description

### 2.1 Group Relative Policy Optimization Fine-Tuning

Data annotated with CEFR labels at document level is scarce, which makes it difficult to use a supervised fine-tuning approach. For this reason, we experiment with reinforcement fine-tuning that does not require labelled data, specifically, with Group Relative Policy Optimization (Shao et al., 2024). In an attempt to balance CEFR-level accuracy and meaning preservation, we use two reward functions to score candidate completions, compute rewards and update the model's weights. The first function computes rewards for compliance with the target CEFR level, whilst the second one scores candidate completions for meaning preservation. We submitted two similar systems developed using this method. They are described in Sections 2.1.2 and 2.1.4.

#### 2.1.1 Data

To obtain texts for generating predictions during fine-tuning, we use the CEFR Levelled English Texts dataset available on Kaggle.[1] Originally, the dataset contained around 1,500 texts labeled with

[*]These authors contributed equally to this work

[1]https://www.kaggle.com/datasets/amontgomerie/cefr-levelled-english-texts



*Proceedings of the Fourth Workshop on Text Simplification, Accessibility and Readability (TSAR 2025)*, pages 242–250

CEFR levels. As the shared task targets simplification of documents written at upper-intermediate or advanced levels, we split the dataset and keep only texts at levels B2, C1 and C2. After splitting texts longer than 150 words into separate examples, we obtained 1,350 unique training examples. We prompt (see Prompt 3 in the Appendix) the model to simplify each instance into each of the target levels: A1, A2 and B1, obtaining a total of 4,050 texts to generate predictions during fine-tuning.

#### 2.1.2 Setup for Run 1

We fine-tune the Llama 3.2 3B Instruct [2] model using the GPRO Trainer from the Transformer Reinforcement Learning [3] library. This model was chosen for its modest size and good instruction-following capabilities. Due to computational constraints, we set up the model to generate 3 candidate simplification for each instance in the dataset. Each of the candidate predictions is scored with the reward functions and ranked. Then, the model's weights are updated to increase the probability of generating high-reward completions and decrease the probability of generating low-reward ones.

#### 2.1.3 Reward Functions

The CEFR compliance reward utilises the CEFR labelling models proposed by the Shared Task organizers in the evaluation scripts. The models are used to produce a CEFR label for each of the 3 candidate completions. The reward formula calculates the absolute difference between the predicted CEFR level and target CEFR level, then converts this distance into a reward score. The reward starts at 1.0 for perfect matches and decreases by 0.5 for each level of deviation, with a minimum reward of 0.0 for texts that are 2+ levels away from the target.

The meaning preservation reward uses the SentenceTransformer model (all-MiniLM-L6-v2) to generate vector embeddings of the original text and a candidate completion. Then, cosine similarity between the embeddings is computed and the scores are cubed to create a more distinctive variance between positive and negative scores. This way, completions that deviate from the original meaning are penalized more strictly.

Due to resource limitations, we choose conservative settings for the GRPO fine-tuning configuration. We use a learning rate of $5 \times 10^{-6}$ and fine-tune for 1 training epoch. For each training step, the model generates 3 candidate completions with an effective batch size of 3. To reduce computational costs, we applied Low-Rank Adaptation (LoRA) fine-tuning.

#### 2.1.4 Setup for Run 2

Run 2 is a variant of the system described in the previous section with some alterations in the training configuration. We use a more lenient CEFR reward function, which reduces penalties for larger gaps between predicted and target CEFR levels as during the exploratory study it showed to lead to better scores for CEFR adjacency compliance. We also slightly upgraded the training parameters by setting a higher learning rate of $1 \times 10^{-5}$ and increased the number of generations in each step from 3 to 4 and the gradient accumulation steps from 3 to 16 for more stability in updating weights. We also train for 2 epochs instead of 1.

In addition, the system employs a revised version of the prompt (see Prompt 4 in the Appendix) for generating candidate predictions. Unlike the system in Run 1, this version of the prompt does not provide examples of texts readable at the corresponding CEFR level.

#### 2.1.5 Inference in the reinforcement fine-tuning pipeline

We prompt the fine-tuned model to simplify documents to the target CEFR level using a zero-shot prompt (see Prompt 5 in the Appendix). Initially, we experimented with several versions of the prompt to generate predictions for fine-tuning, including the prompt with examples of texts readable at the corresponding CEFR levels. The results obtained on the trial data indicated that the model performed best when prompted with a zero-shot prompt without examples of CEFR-level appropriate text.

### 2.2 Lexical simplification

This method employed implements a lexical simplification pipeline that combines a CEFR-annotated lexicon, WordNet synonyms (Miller, 1992), and controlled LLM rewrites. The system is designed to simplify sentences to CEFR A2 and B1 levels while preserving meaning as much as possible. Previous work has attempted CEFR-targeted simplification with LLMs, but the results were often inconsistent, particularly for lower levels such as A2 and B1 (Barayan et al., 2024). To avoid issues like the ones

[2] https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct

[3] https://huggingface.co/docs/trl/en/index

| run | weighted-f1 | cefr-adj | rmse | meaningbert-orig | bertscore-orig |
|---|---|---|---|---|---|
| Run 1 (grpo-ft-v1) | 0.590 | 0.995 | 0.633 | 0.779 | 0.928 |
| Run 2 (grpo-ft-v2) | 0.543 | 0.985 | 0.718 | 0.821 | 0.937 |
| Run 3 (lex-simpl) | 0.578 | 0.710 | 1.269 | 0.972 | 0.985 |

Table 1: Evaluation scores for the best runs of the proposed systems obtained on the Shared Task trial data.

| run | rmse | meaningbert-orig | meaningbert-ref | AvgScore |
|---|---|---|---|---|
| Run 1 (grpo-ft-v1) | 0.718 | 0.821 | 0.797 | -0.076 |
| Run 2 (grpo-ft-v2) | 0.632 | 0.779 | 0.778 | -0.153 |
| Run 3 (lex-simpl) | 1.153 | 0.979 | 0.819 | -0.022 |

Table 2: Final scores and ranking of the submitted systems in the Shared Task obtained on the Shared Task test data.

noticed with previous approaches, this method is deliberately conservative: it avoids uncontrolled rewriting and enforces strict vocabulary constraints. Each of the steps of our method is presented below.

#### 2.2.1 Complex word identification and candidate generation

Sentences are tokenised with `spaCy`, and each content word (noun, verb, adjective, adverb) is checked for possible substitutions. For a word to be included in the replacement table, WordNet must provide a POS-compatible synonym whose lemma also appears at the *target* CEFR level in our lexicon (strict A2/B1, same-level only)[4]. If no such synonyms exist, the word is ignored and may remain unchanged.

#### 2.2.2 Constrained replacement

We present the sentence to the LLM together with a table of *allowed* replacements and instruct it to select at most one option per listed token, leave all other tokens unchanged, and not introduce new vocabulary. For example:

`assist` ⇒ {`help`}, `purchase` ⇒ {`buy`}

An example of the full prompt 1 is provided in the Appendix A.

#### 2.2.3 Style polishing

A second prompt asks the LLM to polish the text to CEFR-specific style: sentence-length limits (A2: max 14 words; B1: max 22), preference for active voice, and restricted connectors (*and, but, or, because, so* for A2; plus *when, if, before, after* for B1). We generate $k$=4 candidates using varied sampling ($temperature \in \{0.2, 0.3, 0.35, 0.45\}$, $top-p \in \{0.95, 0.9, 0.85, 0.8\}$).
An example of the full prompt 2 is provided in the Appendix A.

[4]https://www.kaggle.com/datasets/nezahatkk/10-000-english-words-cerf-labelled

#### 2.2.4 Candidate selection

From the multiple polished outputs, the best candidate is selected using an automatic scoring function. Each candidate is evaluated along four dimensions: (1) CEFR compliance, measured with a ModernBERT classifier; (2) meaning preservation, estimated with MeaningBERT when available, with BERTScore as a fallback and lexical overlap as a final backup; (3) copy ratio, i.e., the proportion of words retained from the original, with penalties applied if this exceeds a level-specific threshold (0.60 for A2 and 0.75 for B1); and (4) sentence length, with penalties applied when the target CEFR limit is exceeded. A single selection score is then computed as a weighted combination of these factors: CEFR compliance (weight 1.0), meaning preservation (+0.15), penalties for excess copy ratio (–0.18×excess over 0.60 for A2 or 0.75 for B1), and sentence length (–0.10×relative excess over the level limit). These weights are predefined hyperparameters rather than learned parameters and were determined by experimenting with the development data. The candidate with the highest score is selected.

#### 2.2.5 Iteration

The system repeats the pipeline until the output is both simple enough and faithful enough to the original. Simplicity is measured by a CEFR compliance score that rewards predictions *at or below* the target level (A2 or B1) and incorporates classifier confidence; this score must reach at least 0.80. Faithfulness is measured by a meaning-preservation score, which must also reach at least 0.80. If both conditions are met, or if six rounds have already been run, the process stops. Additionally, we reject any candidate that lowers the meaning-preservation score by more than 0.05 compared to the previous round. An example of the iterative process is presented in Figure 1.

**Original:** The committee endeavoured to facilitate the distribution of resources in an equitable manner.

[1] endeavoured → tried ⇒ The committee tried to facilitate the distribution of resources in an equitable manner.

[2] facilitate → help ⇒ The committee tried to help the distribution of resources in an equitable manner.

[3] equitable → just ⇒ The committee tried to help the distribution of resources in a just manner.

**Final:** The committee tried to help the distribution of resources in a just manner.

Figure 1: Example of iterative lexical simplification for target level A2

#### 2.2.6 Implementation

The pipeline was implemented in Python with `spaCy` for tokenisation/POS, `NLTK/WordNet` for synonyms, a CEFR lexicon for strict vocabulary control[5]. We used the `transformers` text-classification pipeline with `AbdullahBarayan/ModernBERT-base-reference_AllLang2-Cefr2` for CEFR compliance, `evaluated` for MeaningBERT and BERTScore, and the OpenAI API (`gpt-4o-mini`) for constrained rewrite and style polishing. The performance of the method is presented in Section 3.

### 2.3 Evaluation

The methods presented above were run with a number of parameters on the trial data. We used the evaluation scripts provided by the organizers (Alva-Manchego et al., 2025) to inform the choice of runs to submit to the Shared Task.

For each system, we ran inference iteratively and chose the best-performing runs. We based our choice of the best runs on the scores achieved for weighted F1 and CEFR-level adjacency accuracy. The results for the chosen runs are shown in Table 1.

We observe a trade-off between the meaning preservation capabilities and adjacency accuracy of our systems. The reinforcement fine-tuning method demonstrates higher CEFR adjacency accuracy while having lower meaning preservation scores. The lexical simplification approach, on the other hand, produces outputs that preserve the original meaning due to careful lexical substitution. However, it does not attain high accuracy in CEFR level adjacency.

## 3 Results and Discussion

Table 2 presents the official evaluation results obtained by our systems on the test data. The methodology for the final Shared Task ranking released by the organizers excludes some of the computed metrics and produces a weighted score that relies on RMSE and meaning preservation scores against the original text and the references measured with MeaningBERT.

The Lexical Simplification system (Run 3) we submitted achieves the highest original meaning preservation scores across all systems submitted to the shared task. Due to careful meaning preservation, the Lexical Simplification System ranks higher than the models fine-tuned with GRPO despite lower accuracy in achieving target CEFR levels. The suboptimal RMSE of the reinforcement fine-tuning method might be due to our choice to optimize the reward functions for CEFR adjacency accuracy, which considers outputs successful if their CEFR level is within one level of the specified target, leading to lower accuracy.

Our lexical simplification pipeline, does not employ a word sense disambiguation module to prefilter candidates based on the sense of a word to be replaced. Instead we employ a large language model (gpt-4o-mini) to infer which word fits in the given context. This enables the model to select morphologically and syntactically well-formed substitutes without relying on a separate WSD component.

### 3.1 Error Analysis

We manually analysed the outputs for a more detailed insight into the trade-off between meaning preservation and CEFR compliance scores.

The analysis reveals that the fine-tuned systems overall make more transformations to arrive at a simplification. Some of these transformations, such as sentence splitting and changes to the syntactic structure of the original sentence, are in line with the general text accessibility guidelines. The output of the fine-tuned models appears generally easier to read in comparison with our best-performing system. However, aside from occasional awkward phrasing and slight information loss, it contains multiple semantic errors that sometimes cause sig-

[5]https://www.kaggle.com/datasets/nezahatkk/10-000-english-words-cerf-labelled

nificant distortion of meaning. To demonstrate these, we provide our systems' outputs for the texts from the Shared Task's test data with text-id 113-a2 and 113-b1 (see Table 3) in the Appendix.

Omission is a major cause of semantic errors in the output. For instance, the mention that spider venom is not mainly used to attack humans was not retained in the following examples: *run-1-113-a2*, *run-2-113-a2* and *run-2-113-b1*. This resulted in a significant deviation in meaning. Another example of a critical semantic error is *run-1-113-b1*. The original says 'Spider venom ... serves the purpose of stunning or killing their prey rather than attacking humans', but in the generated simplification the meaning is distorted to "it helps them catch their food by stunning or killing it, not by hurting humans". Not only is this change of preposition unnecessary, it makes the wording ambiguous and may cause confusion. Apart from that, unjustified additions are another cause of meaning distortion in the output of the fine-tuned systems, for example, in *run-1-113-a2*.

As for *run-3-113-a2*, it is an exact copy of the source, which fails the stated goal of A2 readability, for example, low-frequency items such as "fatalities," "urticating," and "embed" remain and there is no simplification of syntax or lexis. As a result, the text is unlikely to be accessible to A2 readers, even though it would score highly on meaning preservation. By contrast, *run-3-113-b1* applies principled reductions and largely preserves meaning. It breaks up long sentences and replaces the clause *"serves the purpose...their prey rather than attacking humans"* with a clearer two-step formulation: *"Most spider species have venom that helps them catch prey. They do not usually attack humans."*. Several lexical substitutions also improve accessibility: *"has not produced any fatalities"* → *"has not caused any deaths"*; *"ejecting a cloud of urticating hairs"* → *"releasing a cloud of irritating hairs"*; *"embed themselves"* → *"stick."*

In addition, several A2 outputs introduce risky lexical changes: *run-3-27-a2* shifts scope *"poor areas"* → *"poor countries"*; *run-3-22-a2* misrenders *"wild dogs"* as *"frank dogs"*; *run-3-38-a2* *"wild animals"* → *"violent creatures"*. Notably, these anomalies are confined to A2-level, B1 outputs generally retain key terms and avoid such errors.

Overall, manual error analysis indicates that the GRPO fine-tuned systems often produce outputs that deviate in meaning from the original, despite the implemented meaning-preservation rewards. While the generative pipeline inherently offers less control than the more conservative rule-based one, a more carefully tailored weighting of the reward functions and implementation of more advanced metrics, such as MeaningBERT (Beauchemin et al., 2023), for computing meaning preservation rewards might improve performance. As for the lexical simplification system, many sentences remain unchanged, particularly at A2-level, so meaning is preserved but CEFR aims are often unmet. Where A2 lexical edits are made, they are sometimes odd or misleading, whereas B1 simplifications tend to be more controlled and effective.

## 4 Conclusions

In this paper, we presented two approaches on readability-controlled text simplification for the TSAR 2025 Shared Task. The lexical pipeline based on strict CEFR-constrained substitutions with light post-editing generally preserves source meaning but quite often fails to replace the problematic words, and at times introduces critical errors, showing that control alone does not preclude serious failures. The models fine-tuned with GRPO are better at producing simplifications corresponding to the specified CEFR level but this is quite often at the expense of keeping the original meaning. These results underline the limitations of both approaches: the lexical pipeline achieves better meaning preservation while often failing at achieving the required readability level. At the same time, although the fine-tuned systems produce simpler texts, the output often deviates form the original meaning while still not matching the required readability level perfectly.

## Limitations

A key limitation of the reinforcement fine-tuning method is the lack of experimentation with reward function design, the weighting of the rewards and training parameters. In addition, the dataset used to generate predictions during reinforcement fine-tuning contains automatically produced labels, and the genres and content of the texts differ from those of the Shared Task test data, which may affect performance. The lexical pipeline used a CEFR word list from Kaggle rather than the official *CEFR-J Vocabulary* (Version 1.5) from TUFS[6]. This resource

[6]Yukio Tono, *The CEFR-J Wordlist Version 1.5*, retrieved from http://www.cefrj.org/download.html on 20 Jan-

deviates from the original CERF list, which may have influenced our results. Finally, the lexical simplifier relied on the commercial **GPT-4o-mini**, so results may be hard to replicate as future versions could behave differently.

## Lay Summary

Reading difficult text can be challenging for many people, including language learners, children, and those with reading difficulties. This research explored ways to automatically rewrite complex texts to make them simpler while keeping the original meaning. Our team developed automatic systems to simplify complex texts to target levels of reading difficulty.

For example, a text that requires advanced language skills needs to be simplified so that speakers with intermediate or elementary skills can understand it easily. When the task says that a text should be simplified for elementary skills, then the result is only successful if the simplified text can be easily understood by speakers with elementary skills. It is not considered successful if the text does not match the required skill level, even if it is generally simpler than the original. The simplified text also needs to keep the meaning of the original difficult text.

Our team developed two different approaches. The first method works like a careful editor. It swaps difficult words for simpler alternatives. This system used a dictionary that labelled words by difficulty level and only replaced complex words with simpler synonyms that meant the same thing. After making these swaps, the system polished the sentences by shortening them and using simpler grammar. The polishing was done using a large language model – a computer program that can generate text. This approach was good at keeping the original meaning but sometimes struggled to make texts simple enough.

The second method involved teaching a large language model to learn how to simplify text to a specific skill level. Rather than following strict rules, the system learned through practice. It performed the task repeatedly and received scores on how well it matched the needed difficulty level and how accurately it preserved meaning. This approach was better at simplifying texts to the target difficulty levels, but it often changed the meaning too much or left out important information.

When tested, the careful word-swapping method performed best overall because it preserved meaning more reliably, even though it did not always achieve the target difficulty level. The taught large language model was better at generating simpler text and achieving the target difficulty level but performed worse overall because it introduced errors or left out important details.

This research highlights that it is challenging to create automatic text simplification systems that match the required difficulty level and keep the original meaning at the same time.

## References


Fernando Alva-Manchego, Regina Stodden, Joseph Marvin Imperial, Abdullah Barayan, Kai North, and Harish Tayyar Madabushi. 2025. Findings of the TSAR 2025 shared task on readability-controlled text simplification. In *Proceedings of the Fourth Workshop on Text Simplification, Accessibility, and Readability (TSAR 2025)*, Suzhou, China. Association for Computational Linguistics.

Abdullah Barayan, Jose Camacho-Collados, and Fernando Alva-Manchego. 2024. Analysing zero-shot readability-controlled sentence simplification. *arXiv preprint arXiv:2409.20246*.

David Beauchemin, Horacio Saggion, and Richard Khoury. 2023. Meaningbert: assessing meaning preservation between sentences. *Frontiers in Artificial Intelligence*, 6:1223924.

Asma Farajidizaji, Vatsal Raina, and Mark Gales. 2023. Is it possible to modify text to a target readability level? an initial investigation using zero-shot large language models. *arXiv preprint arXiv:2309.12551*.

George A. Miller. 1992. WordNet: A lexical database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.


uary 2020

# A Appendix

## A.1 The Prompts Used During Simplification

**Prompt 1 (Method 1)**

```
You are a careful lexical simplifier.
Target CEFR level: {target}. Preserve meaning exactly. Keep names and numbers.
{style_extra}
{vocab_rule}

RULES:
1) You may replace a token ONLY if it appears in the list below.
2) For each listed token, choose at most one alternative from its line.
3) If none of the alternatives fit the meaning, KEEP the original token.
4) Do NOT invent alternatives or use words not in the list.
5) Keep punctuation and sentence order; light edits for grammar are OK.

ALLOWED REPLACEMENTS (source ⇒ options):
{token_1} ⇒ {option_1a, option_1b, ...}
{token_2} ⇒ {option_2a, option_2b, ...}
. . .

Original text:
{original_text}

Output ONLY the rewritten text with your chosen replacements.
```

**Prompt 2 (Method 1)**

```
You are a professional text editor.
Target CEFR level: {target}. Preserve meaning and chosen vocabulary.
{style_extra}
{vocab_rule}

Do NOT add definitions or extra info.
Split long sentences if needed. Prefer active voice.

Original text:
{rewritten_text}

Polished text:
```

**Prompt 3 (Used to generate candidates during fine-tuning for Method 2, Run 1)**

```
System Prompt:

You are an expert in text simplification. You simplify text to the CEFR level
that perfectly aligns with the target CEFR level. You only output simplified
texts. You do not include anything else in your answer.

User Prompt:

Please simplify the following Complex Text to make it easier to read and
understand by {target_cefr} CEFR English learners. To simplify, you may replace
difficult words with simpler ones, elaborate or remove them when possible. You
may also break down a lengthy sentence into shorter, clear sentences. Ensure
the revised sentence is grammatically correct, fluent, and maintains the core
message of the original, without changing its meaning.
Please simplify the following Complex Text to make it easier to read and
understand by {target_cefr} CEFR English learners. To simplify, you may replace
difficult words with simpler ones, elaborate or remove them when possible. You
may also break down a lengthy sentence into shorter, clear sentences. Ensure
the revised sentence is grammatically correct, fluent, and maintains the core
message of the original, without changing its meaning.
The following are the examples of sentences readable by {target_level} CEFR
level English learners:
```

```
Example 1: e1
Example 2: e2
Example 3: e3 Use these examples as reference, do not produce any examples.
Complex Text: {text}

Simplified Text:
```

**Prompt 4 (Used to generate candidates during fine-tuning for Method 2, Run 2)**

```
System Prompt:

You are an expert in text simplification. You simplify text to the CEFR level that
perfectly aligns with the target CEFR level.
You only output simplified texts. You do not include anything else in your answer.

User Prompt:

Please simplify the following Complex Text to make it easier to read and understand by
{target_cefr} CEFR English learners. To simplify, you may replace difficult words with
simpler ones, elaborate or remove them when possible. You may also break down a lengthy
sentence into shorter, clear sentences. Ensure the revised sentence is grammatically
correct, fluent, and maintains the core message of the original, without changing its
meaning.
Only produce the simplified text, do not include anything else.

Complex Text: {text}

Simplified Text:
```

**Prompt 5 (Used during inference for Method 2)**

```
You are an expert in text simplification.

You simplify text to the CEFR level that perfectly aligns with the target CEFR level.
You only output simplified texts. You do not include anything else in your answer.
Please simplify the following Complex Text to make it easier to read and understand by
{target_cefr} CEFR English learners. To simplify, you may replace difficult words with
simpler ones, elaborate or remove them when possible. You may also break down a lengthy
sentence into shorter, clear sentences. Ensure the revised sentence is grammatically
correct, fluent, and maintains the core message of the original, without changing its
meaning.
Only produce the simplified text, do not include anything else.

Complex Text: {text}

Simplified Text:
```

## A.2 Examples Discussed in Error Analysis

| example-id | example-text |
|---|---|
| original-113 | Spider venom is present in most species and serves the purpose of stunning or killing their prey rather than attacking humans. In fact, only 25 of the known spider species produce venom which can actually harm humans, and although spider bites can be painful, they are rarely deadly. Australia's notorious Sydney funnel-web spider has not produced any fatalities since anti-venom was developed in 1981. However, take care not to rub hairy spiders like tarantulas up the wrong way. When they feel in danger, they defend themselves by ejecting a cloud of urticating hairs. These irritating hairs then embed themselves in the skin or eyes of the attacker. |
| run-1-113-a2 | Most spiders have venom that helps them catch their food. [omission] This venom can hurt people, but it is not usually deadly. Only 25 out of many spider species can hurt humans. In Australia, there is a big spider called the Sydney funnel-web spider. It is not deadly since a medicine was made to stop its venom. However, it is still dangerous if you touch it. Tarantulas are also hairy and can hurt you if they feel scared. If they feel threatened, they will throw out irritating hairs to defend themselves. These hairs can get stuck in your skin or eyes. You should be careful around spiders. |
| run-1-113-b1 | Spider venom is present in most spiders. It helps them catch their food by stunning or killing it, not by hurting humans. Most spiders do not harm humans. Only 25 species can hurt humans. Their bites can be painful, but usually not deadly. Australia's Sydney funnel-web spider has not killed anyone since the development of anti-venom in 1981. However, be careful when you touch hairy spiders, like tarantulas. If they feel threatened, they can defend themselves by releasing irritating hairs into the air. These hairs can stick to your skin or eyes if you touch them. |
| run-2-113-a2 | Many spiders have venom. They use it to catch their food. [omission] Most spiders can't hurt people. Only a few species can. If you get bitten by a spider, it can hurt. But it's not usually very bad. In Australia, there's a very dangerous spider called the Sydney funnel-web. It won't kill you. But be careful. Some hairy spiders, like tarantulas, can hurt you if you touch them. When they feel threatened, they spit out hair. These hairs can hurt your skin or eyes. |
| run-2-113-b1 | Most spiders have venom. They use it to catch or kill their food. [omission] Only 25 out of 100 spider species can hurt humans. Spider bites can be painful. But they are not usually deadly. Australia's Sydney funnel-web spider is very dangerous. But it is not deadly since medicine was invented in 1981. If you touch hairy spiders like tarantulas, be careful. When they are scared, they can hurt you by throwing hairs. These hairs can hurt your skin or eyes. |
| run-3-113-a2 | Spider venom is present in most species and serves the purpose of stunning or killing their prey rather than attacking humans. In fact, only 25 of the known spider species produce venom which can actually harm humans, and although spider bites can be painful, they are rarely deadly. Australia's notorious Sydney funnel-web spider has not produced any fatalities since anti-venom was developed in 1981. However, take care not to rub hairy spiders like tarantulas up the wrong way. When they feel in danger, they defend themselves by ejecting a cloud of urticating hairs. These irritating hairs then embed themselves in the skin or eyes of the attacker. |
| run-3-113-b1 | Most spider species have venom that helps them catch prey. They do not usually attack humans. Only 25 spider species have venom that can harm humans. Spider bites can be painful, but they are rarely deadly. The Sydney funnel-web spider in Australia has not caused any deaths since anti-venom was created in 1981. However, be careful not to handle hairy spiders like tarantulas improperly. When they feel threatened, they protect themselves by releasing a cloud of irritating hairs. These hairs can stick to the skin or eyes of the attacker. |

Table 3: Examples discussed in the Error Analysis subsection with errors highlighted in red.