From Complex Word Identification to Substitution: Instruction-Tuned Language Models for Lexical Simplification

Tonghui Han[1,3], **Xinru Zhang**[2*], **Yaxin Bi**[1], **Maurice Mulvenna**[1] **and Dongqiang Yang**[2†]
[1] Ulster University, Belfast BT15 1ED, UK
[2] Shandong Jianzhu University, Jinan 250101, China
[3] Binzhou Polytechnic, Binzhou 256603, China
{han-t, y.bi, md.mulvenna}@ulster.ac.uk
xinru.zhang198@gmail.com ydq@sdjzu.edu.cn

Abstract

Lexical-level sentence simplification is essential for improving text accessibility, yet traditional methods often struggle to dynamically identify complex terms and generate contextually appropriate substitutions, resulting in limited generalization. While prompt-based approaches with large language models (LLMs) have shown strong performance and adaptability, they often lack interpretability and are prone to hallucinating. This study proposes a fine-tuning approach for mid-sized LLMs to emulate the lexical simplification pipeline. We transform complex word identification datasets into an instruction–response format to support instruction tuning. Experimental results show that our method substantially enhances complex word identification accuracy with reduced hallucinations while achieving competitive performance on lexical simplification benchmarks. Furthermore, we find that integrating fine-tuning with prompt engineering reduces dependency on manual prompt optimization, leading to a more efficient simplification framework.

1 Introduction

Lexical simplification (LS) aims to enhance text readability and understandability by replacing complex words or phrases with simpler alternatives without shifting the original meaning or introducing grammatical errors. Traditional LS methodologies typically follow the pipeline, including identifying complex components, finding substitution candidates, and choosing the optimal candidate (Qiang et al., 2020; Lee and Yeung, 2019; Atharva et al., 2023; Paetzold and Specia, 2017; Glavaš and Štajner, 2015). Although this strategy is widely adopted, it suffers from fundamental limitations in both design and execution. In particular, the sequential architecture is prone to error propagation, where errors in early steps negatively impact the quality of the subsequent outcomes. Consequently, accurately identifying complex words and phrases within a given sentence plays a pivotal role in the entire LS system. However, existing LS systems identify complex words by evaluating each word separately. The word semantic complexity relies on its context in real-world linguistic environments. The isolated evaluation strategy tends to reduce the recall of the complex word identification (CWI) module, lowering overall system performance. Although n-gram-based tokenization strategies can partially alleviate this issue, this static segmentation scheme limits its effectiveness in dealing with dynamic language environments (Ferrés et al., 2017).

Furthermore, previous approaches face challenges in generating appropriate substitutions and effectively ranking them. Static approaches, such as vector space-based approaches (Glavaš and Štajner, 2015; Ferrés et al., 2017), rank the substitute candidates by measuring the lexical semantic distance, which may result in semantic drift (Paetzold and Specia, 2017) or part-of-speech (PoS) tag mismatching (Glavaš and Štajner, 2015). Although dynamic approaches mitigate these limitations by leveraging pre-trained language models that utilize contextual knowledge to generate substitutes (Qiang et al., 2020; Atharva et al., 2023), similar to static approaches, these models are constrained to replace complex terms using isolated lexical units, which results in limited effectiveness in handling complex multi-word terms. As shown later in Section 4.2, our fine-tuned system is competitive with traditional models in preserving both grammaticality and adequacy.

Large language models have demonstrated remarkable language understanding capabilities in text generation tasks. Using LLMs enables effective adaptation to diverse linguistic contexts, efficiently identifying complex words and phrases, and generating appropriate substitutions by analyzing

[*] Equal contribution.
[†] Corresponding author.

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 48–58

the entire context (Baez and Saggion, 2023; North et al., 2024). However, since LLMs suffer from hallucinations, inconsistency, and limited controllability, depending on prompting alone is unreliable for LS tasks. Moreover, LLMs are sensitive to the prompt format, leading to an extensive manual design and tuning, making the development process both time-consuming and resource-intensive.

This paper proposes an LS approach that fine-tunes LLMs on a modified corpus and instructs the model to identify and simplify complex terms within given sentences. The experimental results demonstrate that the proposed system can identify variable-length phrases or single words by leveraging its advanced language understanding capability. It simultaneously replaces all complex terms during simplification, while avoiding semantic drift arising from iterative substitution processes in traditional approaches. Moreover, this system addresses the limitations of purely prompt-based methods. Since our system only requires the user to write the instructions in the same format as the training corpora, it does not require users to engage in tuning prompt format. Besides, the fine-tuning process effectively enhances the accuracy rate with decreasing hallucinations. To enable LLMs to comprehend and execute instructions precisely, we construct an LS-instruction-answer (LS-I-A) dataset based on the CWIG3G2 English dataset (Yimam et al., 2017). Experimental results indicate that the proposed approach, integrating fine-tuning with a prompt mechanism, achieves superior simplification efficiency compared with traditional LS pipelines. Moreover, unlike few-shot and zero-shot methods, this approach eliminates the requirements for prompt tuning while effectively mitigating hallucination rates. The key contributions of this paper are as follows.

- We propose a two-step instruction-tuned framework for lexical-level sentence simplification, which explicitly separates complex word identification and lexical substitution, enabling more accurate, context-aware simplifications.

- We construct an instruction-based dataset by reformatting the CWIG3G2 (Yimam et al., 2017) corpus into structured CWI and LS instances, facilitating instruction tuning and downstream evaluation on lexical simplification.

- We conduct comprehensive experiments across multiple LLM backbones (Mistral (Jiang et al., 2023), Qwen (Yang et al., 2025), and LLaMA (Touvron et al., 2023)), comparing fine-tuned and non-fine-tuned settings using both standard automatic metrics (Section 4).

2 Related Works

Traditional LS approaches follow a three-step pipeline—complex word identification, substitution generation, and ranking—yet suffer from error propagation and context-insensitive evaluations (Paetzold and Specia, 2017; Glavaš and Štajner, 2015). To address these issues, edit-based models such as EditNTS (Dong et al., 2019) and EditTS (Kumar et al., 2020) perform explicit token-level operations (e.g., delete, replace, keep), achieving practical simplification with interpretable edits. Similarly, GRS (Dehghan et al., 2022) combines generation and revision in an unsupervised manner, incorporating paraphrasing and deletion at the lexical level. Dress-LS (Zhang and Lapata, 2017) introduces a sequence-to-sequence model trained with reinforcement learning to optimize simplification quality, but offers limited control over specific linguistic properties. ACCESS (Martin et al., 2019) extends this by enabling controllable simplification through attribute-specific constraints such as length and lexical complexity. Despite these advances, many of these systems rely on rigid heuristics or lack the scalability to handle diverse inputs dynamically.

Recent work also explores the use of LLMs such as LLaMA (Touvron et al., 2023) and Mistral (Jiang et al., 2023) for lexical simplification (Baez and Saggion, 2023; North et al., 2024). With strong instruction-following and contextual understanding abilities, these models can simplify complex terms in context via prompting or fine-tuning, offering greater flexibility and scalability. However, they often suffer from limited interpretability and occasional hallucinations, making them less reliable for controlled LS tasks.

In contrast, our work leverages LLMs fine-tuned on a structured instruction-based dataset to identify and directly simplify complex terms in context. Unlike edit-based systems or prompt-only LLM approaches, our method balances controllability and robustness while mitigating hallucinations and format sensitivity.

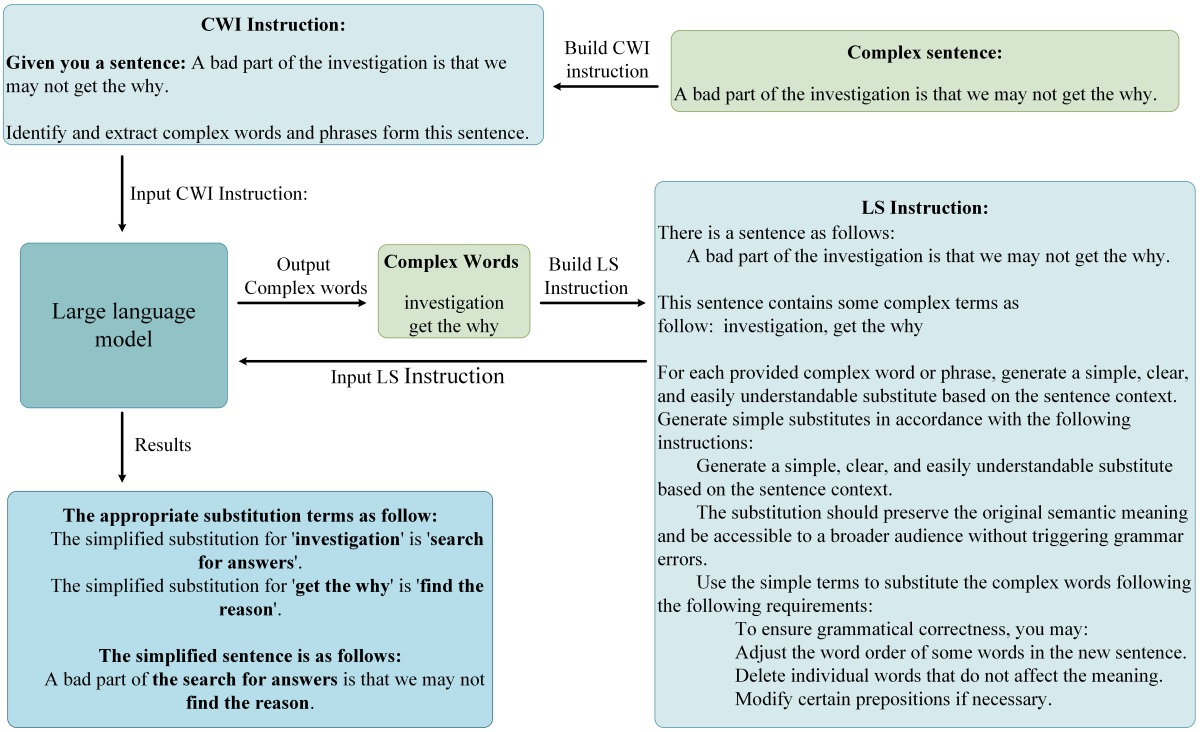

Figure 1: The workflow of the LS system: it first identifies complex words using CWI instructions. Then, it builds LS instructions by utilizing the extracted complex words. Finally, the system outputs the simplified versions under the guidance of the constructed LS instructions.

3 Methodologies

We design a two-stage workflow for lexical simplification using a large language model, as Figure 1 shows. The first stage identifies complex words and phrases within a given sentence. In the second stage, the LLM is prompted with the original sentence and the list of identified complex terms. It is instructed to generate simpler, contextually appropriate alternatives that preserve the original meaning and ensure grammatical correctness. The model may also perform minor syntactic adjustments such as reordering words, modifying prepositions, or removing redundant tokens to maintain fluency. This approach leverages the LLM's contextual understanding and generation capabilities to perform high-quality, semantic-preserving simplification without requiring manually crafted rules or external linguistic resources.

In the following sections, we detail methodologies used in designing this LS system, including constructing the LS-I-A corpus, fine-tuning strategy for LLMs, and evaluation metrics.

3.1 LS-I-A Dataset

3.1.1 Complex Word Identification

We build on the CWI instruction dataset based on the CWIG3G2 English dataset (Yimam et al., 2017) which contains human annotations of complex words across various genres such as News, WikiNews, and Wikipedia. In this study, we manually alter the dataset to transfer the original dataset into instruction-answer format. To convert the dataset into an instruction–answer format suitable for prompting LLMs, we manually revise each instance. This process involves the following steps:

- Sentence reformatting: we extract sentences from the original tabular format and integrate them into instructional prompts as is presented in Figure 1 (CWI Instruction).
- Annotation consolidation: All annotated complex terms within each sentence are aggregated and listed in the answer as follows.

```
The complex words and phrases:
   investigation, get the why.
```

3.1.2 Lexical Simplification

We further adapt the dataset for lexical simplification. While CWIG3G2 was originally designed

for CWI, we extend it via GPT-assisted simplification and expert validation, converting it into an instruction-tuning dataset for LS. First, we employ GPT-4o to generate appropriate simple substitute candidates for each complex term. Next, the complex terms are replaced with their corresponding substitutes. To ensure adequacy and grammatical correctness, we engage native English speakers to assess and correct the simplified sentences. Simultaneously, non-native English speakers with experience in English language teaching in China assess the readability of the simplified terms for non-native readers. Following the evaluation and revision processes, we retain 1,784 instances in the training set and 225 instances in the validation set. The final dataset is formatted into an instruction-answer format with the following structure.

- Instruction: It contains the original complex sentence, followed by identified complex terms. The instruction also specifies that each complex term should be simplified based on contextual knowledge. Furthermore, we outline certain restrictions to keep the original meaning and avoid grammatical errors. The instance is detailed in Figure 1 (LS Instruction)

- Answer: The answer should list simple substitution terms and present the simplified sentence.

Therefore, the LS-I-A dataset comprises four groups of attributes. The attributes `cwi_instruction` and `cwi_answer` provide instructions and references designed for fine-tuning LLMs to identify complex terms. Similarly, `ls_instruction` and `ls_answer` are constructed to guide the fine-tuning process for lexical simplification.

3.2 Fine-tuning

Notably, our fine-tuning is performed on a small-scale dataset, demonstrating the feasibility of low-resource instruction tuning for lexical simplification. In the training process, to enable the model to simulate the LS pipeline, for each instance, its `cwi_instruction` is first fed into the LLMs, immediately followed by its `ls_instruction`. All the instances are iteratively fed into the LLMs following these steps, enabling them to learn the simplification rules and strategies.

We use our newly constructed dataset to fine-tune LLMs for complex word identification and lexical simplification. In this study, we select three mid-sized open-source instruction-aligned LLMs: Llama-3.1-8B-Instruct[1], Qwen2.5-7B-Instruct-1M[2], and Mistral-8B-Instruct-2410[3].

All models are fine-tuned using **LoRA** (Low-Rank Adaptation) with 4-bit precision (NF4 quantization) to enhance memory efficiency. The fine-tuning process is executed on an **RTX 4090 GPU**, with gradient checkpointing enabled to reduce memory consumption. The models are trained for 10 epochs with a per-device batch size of 2 and a gradient accumulation of 4. To optimize performance, a cosine learning rate scheduler with an initial learning rate of 1e-4 is applied, complemented by mixed precision (fp16). The optimizer is configured as **AdamW**, with a weight decay of 0.05 and a maximum gradient norm of 0.3. Model checkpoints and evaluations are performed at the end of each epoch, and the best-performing model is restored after the completion of training.

3.3 Evaluation Metrics

For CWI tasks, information retrieval metrics (precision, recall, and F1-score) are employed to evaluate the systems, with the basic metrics illustrated by Manning (2009). In addition to these basic information retrieval metrics, we introduce a novel evaluation metric, hallucination rate (HR), to assess the reliability of the LLMs. To obtain the HR, we define a hallucination term as follows.

A **hallucination term** is defined as any term generated by the LLMs that either does not appear in the original sentence or is identified as a numerical value, special character, or stop word.

The equation for calculating HR is presented below.

$$HR = \frac{1}{n}\left[\sum_{i=1}^{n}\frac{h_i}{m_i}\right] \quad (1)$$

where, $\boldsymbol{n}$ represents the test size, h_i denotes the number of hallucination terms in instance $\boldsymbol{i}$, and m_i is the number of identified terms in instance $\boldsymbol{i}$.

We evaluate the effectiveness of our system on the lexical-level sentence simplification task using three established metrics: SARI (Xu et al., 2016) for simplicity, FKGL (Flesch–Kincaid Grade

[1]https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

[2]https://huggingface.co/Qwen/Qwen2.5-7B-Instruct-1M

[3]https://huggingface.co/mistralai/Ministral-8B-Instruct-2410

Level) (Kincaid et al., 1975) for fluency, and $BERT_{score}$ (Zhang et al., 2019) for adequacy. SARI and FKGL are obtained through the EASSE evaluation toolkit (Stodden, 2024), which provides a standardized framework for assessing simplification quality, while $BERT_{score}$ (Zhang et al., 2019) is calculated via the bert-score module[4].

4 Results

4.1 CWI results

The reformulated CWI task requires the model to extract all complex terms in a sentence rather than focusing on preselected target words, thereby expanding the instance space and altering the task definition. Consequently, direct comparison with traditional CWI methods is not methodologically valid. Instead, we investigate the effectiveness of instruction-based fine-tuning by contrasting it with non-fine-tuned counterparts under identical conditions.

We revise the CWIG3G2 test set following the methodology outlined in Section 3.1.1 to assess the performance of our system on CWI tasks. The resulting test set comprises 325 instances, each consisting of an instruction paired with a corresponding reference. The evaluation results are presented in Table 1. As demonstrated, the fine-tuned LLaMA model achieves the highest F1-score (0.8188), along with notable precision (0.7784) and the highest recall (0.8635), indicating a balanced and highly effective identification of complex words. The fine-tuned Mistral model also performs strongly, with an F1-score of 0.8064 and the lowest hallucination rate of 0.0055, demonstrating its reliability. Additionally, the fine-tuned Qwen model shows competitive performance, achieving an F1-score of 0.8079 and the highest precision (0.7864) among all fine-tuned models while maintaining a reasonable recall (0.8306).

In contrast, the non-fine-tuned versions of all three models yield substantially lower F1 scores and significantly higher hallucination rates. For example, non-fine-tuned Qwen, despite achieving the highest precision overall (0.8388), suffers from a low recall (0.4727), resulting in a significantly lower F1-score (0.6077). These results underscore the effectiveness of fine-tuning in enhancing the accuracy and robustness of LLMs for CWI tasks.

[4]https://github.com/Tiiiger/bert_score

4.2 Simplification Results

This section presents the simplification results of our system and provides a comparative analysis against state-of-the-art lexical simplification models as well as non-fine-tuned LLMs. All the LS systems are evaluated on the TurkCorpus (Xu et al., 2016) and ASSET (Alva-Manchego et al., 2020) datasets. This study selects NTS+SARI (Nisioi et al., 2017), ACCESS (Martin et al., 2019), EditNTS (Dong et al., 2019), Edit-Unsup-TS (Kumar et al., 2020), Dress-LS (Zhang and Lapata, 2017), GRS (Dehghan et al., 2022), non-fine-tuned LLaMA, non-fine-tuned Qwen, and non-fine-tuned Mistral as baseline models due to their relevance to lexical-level sentence simplification. The evaluation results are summarized in Table 2 for clarity and comparison.

As shown in Table 2, our fine-tuned models demonstrate competitive or superior performance compared with both traditional lexical simplification systems and non-fine-tuned LLMs. Notably, the fine-tuned Qwen model achieves the highest SARI score on the ASSET benchmark (41.41), along with a $BERT_{score}$ of 0.9492 and an FKGL of 7.568, indicating its strong capability in producing simplified text while preserving semantic content. Fine-tuned Mistral also performs well, with SARI scores of 39.55 on TurkCorpus and 41.24 on ASSET, $BERT_{score}$s of 0.942 and 0.953, and an FKGL of 7.32 on both datasets. Similarly, fine-tuned LLaMA attains SARI scores of 39.13 on TurkCorpus and 40.59 on ASSET, with corresponding $BERT_{score}$s of 0.941 and 0.950.

Among traditional systems, ACCESS achieves the best performance on TurkCorpus, with the highest SARI score (42.08) and the lowest FKGL (7.29). However, its $BERT_{score}$ (0.955) remains slightly lower than those of EditNTS (0.961) and Dress-LS (0.964). On ASSET, GRS yields the lowest FKGL (4.17), indicating enhanced readability, while EditNTS achieves the highest $BERT_{score}$ (0.970), reflecting strong semantic fidelity.

Overall, these results confirm the effectiveness of fine-tuning large language models for lexical simplification. All fine-tuned models consistently outperform their non-fine-tuned counterparts and match or exceed the performance of established simplification systems.

model	precision ↑	recall ↑	F1 ↑	HR ↓
ft Mistral	0.7667	0.8545	0.8064	**0.0055**
ft Qwen	0.7864	0.8306	0.8079	0.0145
ft LLaMA	0.7784	**0.8635**	**0.8188**	0.0065
non-ft Mistral	0.6207	0.7388	0.6778	0.0616
non-ft Qwen	**0.8388**	0.4727	0.6077	0.0335
non-ft LLaMA	0.7300	0.6712	0.6994	0.0532

Table 1: Evaluation results for the CWI task, comparing fine-tuned models with their original versions. Background color indicates performance level (darker green = better; lower is better for HR).

Model	TurkCorpus			ASSET		
	SARI ↑	FKGL ↓	BERTscore ↑	SARI ↑	FKGL ↓	BERTscore ↑
NTS+SARI (Nisioi et al., 2017)	36.93	8.18	0.959	34.02	8.18	0.967
ACCESS (Martin et al., 2019)	**42.08**	7.29	0.955	40.12	7.29	0.966
EditNTS (Dong et al., 2019)	38.51	8.37	0.961	34.94	8.37	**0.970**
Edit-Unsup-TS (Kumar et al., 2020)	38.09	6.44	–	38.94	6.39	–
Dress-LS (Zhang and Lapata, 2017)	36.89	7.58	**0.964**	36.90	7.58	0.951
GRS (Dehghan et al., 2022)	–	–	–	37.9	**4.17**	–
non-ft Mistral	33.43	8.99	0.924	38.69	8.99	0.936
non-ft Qwen	32.61	**5.24**	0.901	39.13	5.24	0.915
non-ft LLaMA	34.68	7.47	0.921	39.71	7.47	0.932
ft Mistral	39.55	7.32	0.942	41.24	7.32	0.953
ft Qwen	38.60	7.57	0.940	**41.41**	7.568	0.949
ft LLaMA	39.13	7.59	0.941	40.59	7.59	0.950

Table 2: Performance of different models on TurkCorpus and ASSET benchmarks. Cell background color indicates performance: darker green = better. For FKGL, lower values are better and mapped to deeper green.

5 Analysis

As shown in Table 1 and Table 2, the instruction-tuned models exhibit substantial improvements over their non-fine-tuned counterparts on the CWI and LS tasks. These results suggest that instruction-based fine-tuning enhances the ability of large language models to identify complex lexical items with greater accuracy. Furthermore, our system is competitive with existing systems, generating simplifications that are more fluent and contextually aligned. These findings underscore the effectiveness of our framework in steering LLMs toward more precise, context-aware, and semantically faithful simplification.

To gain deeper insight into the behavior of our instruction-tuned lexical simplification system, we analyze the SARI subcomponents alongside the structural and lexical metrics presented in Figures 2 and 3. These fine-grained evaluations go beyond aggregate performance scores, offering a more nuanced understanding of how the model balances adequacy, fluency, and simplicity in its simplification strategy.

5.1 SARI Subcomponent Analysis

As shown in Figure 2, our fine-tuned models exhibit competitive performance across the three SARI subcomponents—add, keep, and delete—demonstrating the effectiveness of our system in capturing the diverse operations involved in sentence simplification. Notably, the fine-tuned LLaMA model achieves the highest add score on TurkCorpus and maintains strong performance on ASSET, indicating that it is particularly effective at introducing simplified content that aligns well with the surrounding context. This capability is associated with our instruction tuning framework, which separates complex word identification from substitution generation. By decoupling these stages, the model is first guided to identify complex items by leveraging word morphology and contextual cues, and subsequently to generate replacements that are better informed by the identified term and its contextual knowledge, resulting in more appropriate

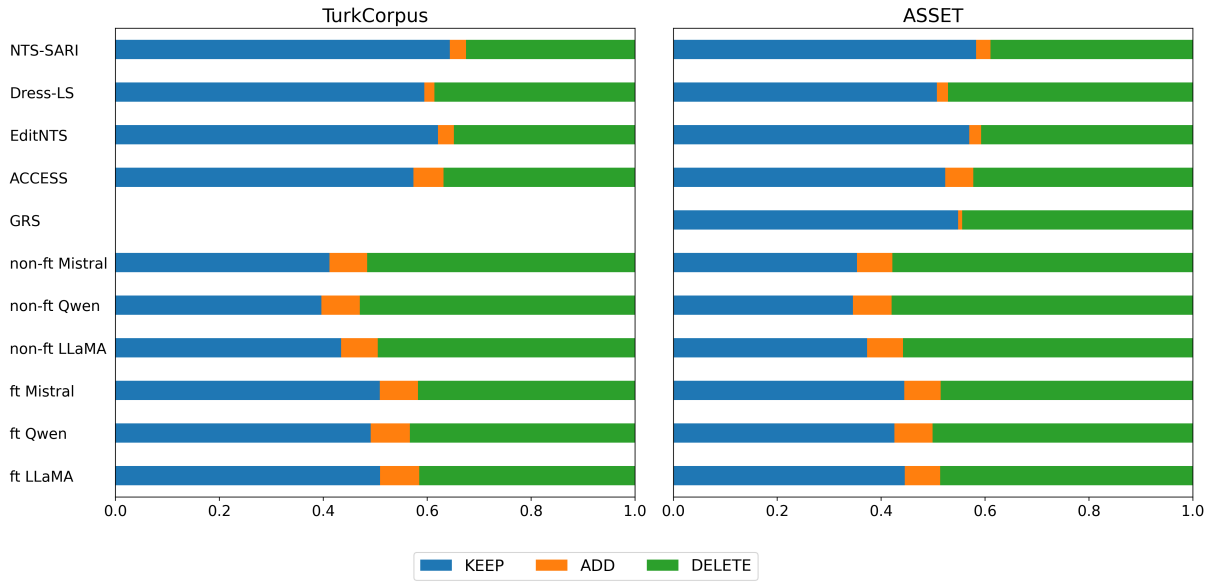

Figure 2: SARI subcomponent ratio (add, keep, delete) for TurkCorpus and ASSET datasets across different models.

and semantically coherent simplifications.

Although the keep scores of our fine-tuned models are lower than those of traditional systems such as EditNTS and ACCESS, they reflect a more transformative simplification strategy. While higher keep scores generally indicate stronger preservation of the original sentence structure, they may also reflect a more conservative editing approach with limited modification. In contrast, our models are more inclined to rephrase the complex items and partially restructure the input, yielding moderate keep scores that maintain core semantic content while enabling both lexical and syntactic simplification. This editing behavior is further supported by consistently strong delete scores, which suggest that our models are effective at removing unnecessary or overly complex content, thereby enhancing clarity and conciseness.

5.2 Structural and Lexical Evaluation

Figure 3 offers a visual demonstration of how instruction-based fine-tuning enhances the quality of text simplification along both structural and lexical dimensions. Compared to baseline models, our fine-tuned models exhibit a balanced distribution across key structural indicators. Notably, the compression ratio (blue segments) for the fine-tuned models remains close to 0.90, indicating that they reduce sentence length effectively without aggressive truncation. In contrast, non-fine-tuned models display more extreme behaviors—for example, non-fine-tuned Qwen heavily compresses (left-shifted blue segment), while non-fine-tuned Mistral barely shortens inputs at all.

Sentence split proportions (orange segments) are consistent across all models, with fine-tuned variants maintaining structural cohesion. The "Exact copies" segments (green) are minimal for fine-tuned models—remarkably fine-tuned LLaMA—highlighting their strong rewriting capability compared to traditional models like Dress-LS, which retain a higher proportion of copied content.

Additionally, the red (Additions) and cyan (Deletions) segments show that the fine-tuned models engage in more balanced and substantive edits. For example, fine-tuned LLaMA and Qwen demonstrate nearly symmetrical proportions of additions and deletions, indicating that the models are not merely replacing individual words (as in shallow lexical simplification) but are actively restructuring sentences by inserting relevant information and removing redundant or complex segments, thereby engaging in more meaningful and substantive simplification.

Finally, the Levenshtein similarity (gray segments) and lexical complexity scores indicate that our fine-tuned models strike a desirable balance between adequacy and fluency. The longer gray bars in fine-tuned models signify sufficient divergence from the source while maintaining coherence. These visual patterns affirm the benefits of fine-tuning LLMs in sentence simplification, aligning more closely with human-like simplification behaviors.

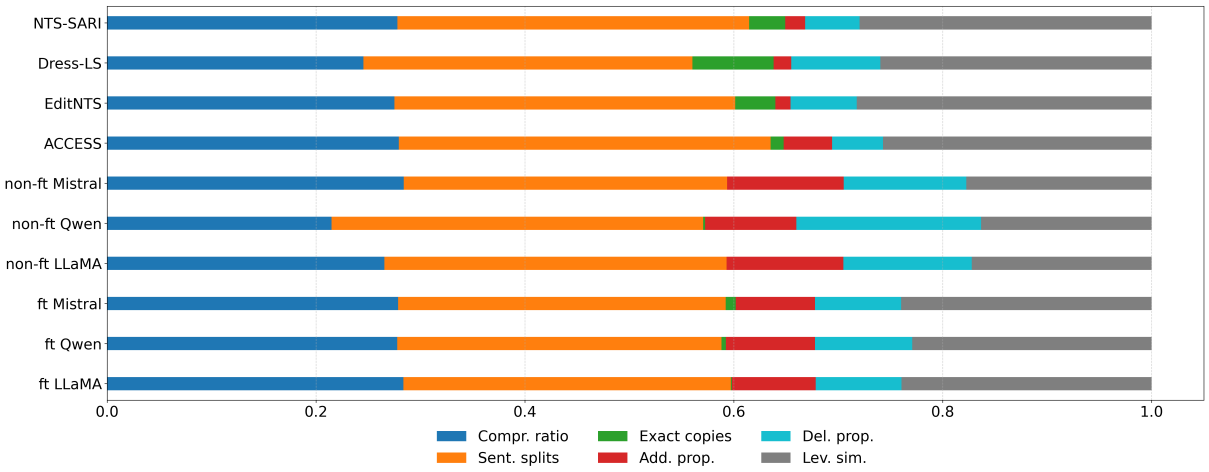

Figure 3: Distribution of Structural and Lexical Metrics Across Simplification Models.

In summary, these results validate the strength of our instruction-based fine-tuning approach. By structurally isolating CWI and LS within modular prompt designs, our models consistently outperform their non-fine-tuned counterparts and match or exceed the performance of existing state-of-the-art lexical simplification systems across various evaluation criteria, including simplification depth, fluency, and adequacy. Further illustrations are detailed in Appendix A, which includes three representative cases comparing traditional models, non-fine-tuned LLMs, and fine-tuned LLMs. The analysis demonstrates how instruction tuning enables more fluent, semantically faithful, and structurally appropriate simplifications across varied sentence types.

6 Conclusion

This study proposes a lexical-level sentence simplification system based on LLMs that emulate the traditional LS pipeline. We revise the CWIG3G2 dataset to develop this system and construct a new instruction–answer corpus, LS-I-A, for fine-tuning LLMs. Experimental results demonstrate that fine-tuning significantly enhances LLM performance in CWI and LS tasks. On LS tasks, our system achieves performance comparable to several state-of-the-art models. Despite the strong performance, our system still requires further improvements. Notably, while LS-I-A performs well in practice, it lacks reasons for explaining the selection of complex words and the generation of simplified alternatives. As a result, the system does not support Chain-of-Thought reasoning in either the identification or substitution stages, limiting the interpretability of its outputs. In future work, we aim to refine the LS-I-A dataset and incorporate strategies that enhance the system's reasoning capabilities.

Limitations

Our study has several limitations. First, the LS-I-A dataset is designed to fine-tune LLMs by mimicking the traditional LS pipeline, identifying complex words, and providing substitutions. However, it offers no explanations or rationales for simplifying choices, lacking interpretability. Moreover, the dataset does not consider stylistic factors; it primarily emphasizes semantic preservation and grammatical correctness, neglecting aspects such as tone or fluency. Second, although fine-tuning improves LLM performance on simplification tasks, the generation process remains largely uninterpretable and challenging to control. As a result, models can still produce hallucinations or overly aggressive simplifications that distort the original meaning. Finally, current evaluation metrics are insufficient for thoroughly assessing simplification quality. Sentence simplification involves multiple dimensions that are not comprehensively captured by existing metrics. In particular, SARI is heavily reference-dependent and can be biased by the lexical and stylistic preferences in the reference simplifications.

Acknowledgments

We gratefully acknowledge the support of Ulster University and Shandong Jianzhu University. Their generous assistance and institutional backing have been instrumental in enabling this research and contributing to its successful completion.

References

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020. Asset: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. *arXiv preprint arXiv:2005.00481*.

Kumbhar Atharva, Sonawane Sheetal, Kadam Dipali, and Mulay Prathamesh. 2023. Casm-context and something more in lexical simplification. In *Proceedings of the 20th International Conference on Natural Language Processing (ICON)*, pages 506–515.

Anthony Baez and Horacio Saggion. 2023. Lsllama: Fine-tuned llama for lexical simplification. In *Proceedings of the Second Workshop on Text Simplification, Accessibility and Readability*, pages 102–108.

Mohammad Dehghan, Dhruv Kumar, and Lukasz Golab. 2022. Grs: Combining generation and revision in unsupervised sentence simplification. *arXiv preprint arXiv:2203.09742*.

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. Editnts: An neural programmer-interpreter model for sentence simplification through explicit editing. *arXiv preprint arXiv:1906.08104*.

Daniel Ferrés, Horacio Saggion, and Xavier Gómez Guinovart. 2017. An adaptable lexical simplification architecture for major ibero-romance languages. In *Proceedings of the first workshop on building linguistically generalizable NLP systems*, pages 40–47.

Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 63–68.

Yinhan Jiang and 1 others. 2023. Mistral 7b. https://mistral.ai/news/announcing-mistral-7b/.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

Dhruv Kumar, Lili Mou, Lukasz Golab, and Olga Vechtomova. 2020. Iterative edit-based unsupervised sentence simplification. *arXiv preprint arXiv:2006.09639*.

John SY Lee and Chak Yan Yeung. 2019. Personalized substitution ranking for lexical simplification. In *Proceedings of the 12th international conference on natural language generation*, pages 258–267.

Christopher D Manning. 2009. *An introduction to information retrieval*.

Louis Martin, Benoît Sagot, Eric de la Clergerie, and Antoine Bordes. 2019. Controllable sentence simplification. *arXiv preprint arXiv:1910.02677*.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 85–91.

Kai North, Tharindu Ranasinghe, Matthew Shardlow, and Marcos Zampieri. 2024. Multils: An end-to-end lexical simplification framework. In *Proceedings of the Third Workshop on Text Simplification, Accessibility and Readability (TSAR 2024)*, pages 1–11.

Gustavo Paetzold and Lucia Specia. 2017. 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.

Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2020. Lsbert: A simple framework for lexical simplification. *arXiv preprint arXiv:2006.14939*.

Regina Stodden. 2024. Easse-de: Easier automatic sentence simplification evaluation for german. *arXiv preprint arXiv:2404.03563*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, and 1 others. 2025. Qwen2. 5-1m technical report. *arXiv preprint arXiv:2501.15383*.

Seid Muhie Yimam, Sanja Štajner, Martin Riedl, and Chris Biemann. 2017. Cwig3g2-complex word identification task across three text genres and two user groups. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 401–407.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. *arXiv preprint arXiv:1703.10931*.

Case 1	She was among the first doctors to object to cigarette smoking around children, and drug use in pregnant women.
NTS+SARI	She was among the first doctors to object to cigarette smoking around children, and drugs use in pregnant women.
ACCESS	She was one of the first doctors to find out how to cigarette smoking around children, and use in pregnant women.
EditNTS	She was among the first doctors to object to cigarette smoking around children, and drug use in pregnant women.
Dress-LS	She was among the first doctors to make cigarette smoking around children .
non-ft Mistral	She was a first medical professional to oppose smoking near kids and drug misuse during pregnancy.
non-ft Qwen	She was among the earliest doctors to oppose smoking near kids and taking drugs while pregnant.
non-ft LLaMA	She was among the first doctors who opposed cigarette smoking near children, and drug use when women were expecting a baby.
ft Mistral	She was among the first doctors to protest against smoking around children, and drug use in expecting women.
ft Qwen	She was among the first doctors to protest smoke around children, and drug use by expecting mothers.
ft LLaMA	She was among the first doctors to protest smoking around children, and drug use in expecting women.
Case 2	They are culturally akin to the coastal peoples of Papua New Guinea.
NTS+SARI	They are culturally akin to the coastal communities of Papua New Guinea.
ACCESS	They are to the coastal peoples of Papua New Guinea, Papua New Guinea.
EditNTS	They are akin to the coastal peoples of Papua New Guinea.
Dress-LS	They are culturally referring to the coastal peoples of Papua New Guinea.
non-ft Mistral	They have similar cultures to the people who live near the coasts of Papua New Guinea.
non-ft Qwen	They are similar to people who live near the coast in Papua New Guinea.
non-ft LLaMA	The coastal peoples of Papua New Guinea have much in common with them.
ft Mistral	They are similar in culture to the near-sea people of Papua New Guinea.
ft Qwen	They are similar to the people near the sea in Papua New Guinea.
ft LLaMA	They are similar in culture to the people near the sea in Papua New Guinea.
Case 3	These attacks may have been psychological in origin rather than physical.
NTS+SARI	These attacks may have been psychological in origin than physical.
ACCESS	These attacks may have been different in origin rather than physical.
EditNTS	These attacks may have been psychological in origin and physical.
Dress-LS	These attacks may have been psychological in origin rather than physical.
non-ft Mistral	These actions could have started mentally instead of physically.
non-ft Qwen	These assaults might have started in people's minds instead of causing direct harm.
non-ft LLaMA	These attacks may be caused by emotional pain instead of causing physical harm.
ft Mistral	These attacks may have been meant to scare people rather than hurt them.
ft Qwen	These attacks may have been mental in source rather than physical.
ft LLaMA	These attacks may have been mental cause rather than physical.

Table 3: Lexical simplification examples across three cases. Our instruction-tuned models produce simpler, more fluent outputs while retaining the original meaning.

A Appendix

Three representative examples are used to analyze the behavioral differences across simplification systems. The comparison includes outputs from traditional models, non-fine-tuned LLMs, and instruction-fine-tuned LLMs.

The first case contains multiple complex expressions joined by coordination. Traditional models such as Dress-LS and EditNTS perform primarily local edits, often deleting or replacing single-word units. This strategy frequently results in incomplete simplifications or loss of important content. Non-fine-tuned LLMs produce more syntactically varied outputs, but these are often semantically inaccurate. Instruction-fine-tuned models apply edits across multiple spans, preserving the core meaning while improving surface fluency. The instruction–answer format used during training presents simplification as a global transformation task, requiring the model to operate over the entire sentence rather than isolated tokens.

The second case includes a long noun phrase with generalized referents. Outputs from traditional models exhibit minimal restructuring and mostly retain the original phrasing. Non-fine-tuned LLMs attempt paraphrasing but often generate repetitive or verbose alternatives. Instruction-fine-tuned models produce simpler constructions that reduce lexi-

cal and syntactic complexity without distorting the original meaning. The presence of sentence-level, span-aligned annotations in the LS-I-A dataset provides direct supervision for such multi-span transformations, encouraging broader structural adjustment rather than surface-level replacement.

The third case features contrastive lexical elements and implicit logical relations. Traditional models often simplify only part of a contrastive expression or leave the contrast unclear, reducing semantic clarity in the output. Non-fine-tuned LLMs vary in output quality, often generating inconsistent or logically disjointed results. Instruction-fine-tuned models consistently preserve the contrast and simplify the associated expressions in a controlled manner. The simplification objective in training is framed around semantic preservation under minimal complexity, which supports stable handling of discourse-level relationships in cases involving contrast or attribution.

The observed differences across systems correspond to their supervision regimes. Traditional models rely on local alignment or rule-based editing, which constrains their capacity for structural rewriting. Non-fine-tuned LLMs lack explicit task grounding and produce unstable outputs. Instruction-fine-tuned LLMs receive training on task-specific instructions and span-level supervision, which enables more consistent simplification at both lexical and structural levels.