UZH_Pandas at SimpleText@CLEF-2023: Alpaca LoRA 7B and LENS Model Selection for Scientific Literature Simplification

Pascal Severin Andermatt[1,*,†], Tobias Fankhauser[1,*,†]

[1]*University of Zurich (UZH), Rämistrasse 71, 8006 Zürich, Switzerland*

Abstract

In this study, we advance the field of scientific text simplification by harnessing the capabilities of Alpaca LoRA 7B [1], a large language model derivative of the 7B LLaMA [2]. We expand the dataset for Task 3 of SimpleText@CLEF-2023 [3] by integrating data from Task 2, aiming to identify complex terms in need of explanation for better text comprehension. Our methodology involves rigorous fine-tuning, prompt engineering, and the application of the LENS score [4] as a tool for model reranking and evaluation. Our findings suggest the efficacy of our approach in creating a more effective text simplification system. Our final model demonstrates expertise not only in expanding abbreviations, but also in explaining complex terms present in the input sentence. This ability allows it to create texts that are both easy to understand and simple to comprehend, making the information presented more accessible and opening the door for more efficient communication. However, the study also highlights several challenges and areas of improvement, providing a valuable contribution to future research in text simplification. Our research underscores the potential of large language models like Alpaca LoRA 7B in transforming complex terminologies into more accessible language, ultimately enhancing the public's understanding of scientific literature.

Keywords

Scientific text simplification, Generative Language Models, Automated Simplification Metrics, Prompt engineering, Alpaca LoRA 7B, LENS, SimpleText@CLEF-2023

1. Introduction

In the rapidly evolving landscape of Natural Language Processing (NLP), the simplification of intricate texts remains a prevalent challenge. To tackle this, we engage Alpaca LoRA 7B [1], a state-of-the-art large language model fine-tuned from the LLaMA architecture [2] using 52K instruction-following data [5]. This model, further refined using the Low-Rank Adaptation (LoRA) technique [6], presents an effective solution for condensing and simplifying complex scientific narratives.

This paper explores our experiment in using Alpaca LoRA 7B [1] to address Task 3 of the SimpleText CLEF shared task [3], which aims to simplify scientific abstracts given a specific

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18-21, 2023, Thessaloniki, Greece

[*] Corresponding author.

[†] These authors contributed equally.

pascalseverin.andermatt@uzh.ch (P. S. Andermatt); tobias.fankhauser@uzh.ch (T. Fankhauser)
https://github.com/pandermatt (P. S. Andermatt); https://github.com/TobiFank (T. Fankhauser)

Input *snt_id: G06.2_2805209921_6*: we have proven that transfer learning is not only applicable in this field, but it requires smaller well-prepared training datasets, trains significantly faster and reaches similar accuracy compared to the original method, even improving it on some aspects.
Output extended, simplification, complex words: Transfer learning is a technique used in machine learning which involves taking a pre-trained model and using it as a starting point for a new model. It requires smaller well-prepared training datasets, trains significantly faster and reaches similar accuracy compared to the original method, even improving it on some aspects.
Output default, simplification, default: Transfer learning is better than the original method.

Figure 1: Example of text simplification, comparing the different model approaches. The examples demonstrate how the *extended* model with the *simplification* prompt and *complex words* evaluation provides a detailed explanation of the concept *transfer learning*. In contrast, the *default* model with the *simplification* prompt and *default* evaluation provides a much simpler, yet less informative statement.

query. While previous efforts have made progress in automatic simplification of scientific texts, there remains a gap between these scholarly texts and their accessibility to the public. Our goal is to further diminish this barrier by deploying an automated text simplification system that retains critical information while reducing linguistic complexity.

Building upon the findings from prior research that demonstrated the complementary nature of Task 2 and Task 3 [7], we also explored incorporating data from Task 2 to extend the data for Task 3 of the SimpleText shared tasks. With the data from Task 2, we seek to identify terms or concepts that need explanation for understanding a passage. With this, we aim to provide valuable insight into the key elements of scientific texts that typically impede understanding. By incorporating this data into our work with Alpaca LoRA 7B, we intend to create a more attuned and efficient text simplification system that anticipates and addresses potential comprehension obstacles.

The key contributions and findings of this study are the following:

- We illustrate the benefits of dataset augmentation by showing that the integration of the Task 2 dataset (consisting of difficult terms, identified in scientific abstracts, and their corresponding explanations) into the Task 3 dataset (comprising simplified sentences) can enhance the performance and generalization capabilities of the model, improving the text simplification process.
- We demonstrate the effectiveness of the Alpaca LoRA 7B model for the text simplification task. By leveraging its instruction following capabilities, Alpaca LoRA 7B was able to effectively simplify complex linguistic constructs.
- We highlight the efficacy of the LENS score as a method for model re-ranking and evaluation in the context of text simplification. Compared to traditional metrics such as SARI, the LENS score more accurately captures the nuances of text simplification and aligns more closely with human judgement on the quality of text simplifications.

Our study marks an exploratory step towards understanding how effectively large language models like Alpaca LoRA 7B [1] can simplify complex terminologies and concepts into more accessible language. Our findings offer insights into the potential of these models and their applicability to text simplification, making a contribution to improving public understanding of the scientific literature. Figure 1 shows an example of how our model can explain complex terms like *transfer learning* and simplify the sentence.

The structure of the paper continues as follows: Section 3 discusses our *default* and *extended* datasets used in model training, their origin, the process of their integration, and their effect on the model's performance.

In Section 4, we provide a comprehensive overview of the various components and steps involved in our approach. We begin by discussing the utilized model architectures in Section 4.1, followed by an exploration of the prompt engineering process in Section 4.2. We then investigate the incorporation of *complex terms* in Section 4.3, and subsequently explain the fine-tuning procedure in Section 4.4. The evaluation methodology is presented in Section 4.5, and finally, we introduce the LENS Score in Section 4.6 as a metric for assessing the quality of our results.

Finally, sections 5-7 outline our text simplification findings, highlighting the importance of suitable approaches and strategies, and ending with prospects for future research.

2. Background

Text-to-Text Transfer Transformer (T5) T5 [8], grounded in transformer architecture, operationalized a notable approach in the field of natural language processing (NLP) by recasting all tasks as text-to-text problems. This strategy allowed a single model to address a wide variety of NLP tasks, which was an important step in the development of these technologies. Despite its advancements, T5's effectiveness is intimately tied to the quality and volume of its training data, and the model lacks the ability to genuinely understand the textual content it processes. These factors have shaped its strengths and limitations in practice [8].

Despite these limitations, the success of T5 underscores the potential of transformer-based Large Language Models (LLMs) in tasks like text simplification. LLMs generate contextually appropriate responses, enabling nuanced simplifications. Yet, they require considerable computational resources and can occasionally produce verbose or off-topic outputs.

Alpaca 7B The Alpaca 7B model [9], introduced by Stanford University, is an instruction-following language model, fine-tuned from Meta's LLaMA 7B [2]. This compact and efficient model closely parallels the capabilities of OpenAI's models, notably `text-davinci-003`, but offers a cost-efficiency alternative for academic research.

The development of Alpaca 7B addresses key challenges in training high-quality, budget-friendly instruction-following models. For that, an innovative adaptation of the self-instruct method was utilized. With 175 initial human-written instruction-output pairs, `text-davinci-003` was used to generate an additional 52,000 instruction-following demonstrations [5], which were then employed to fine-tune Alpaca 7B using Hugging Face's training framework. `text-davinci-003` refers to the third version of a text-based model developed by OpenAI [10].

Preliminary human evaluations demonstrated favorable performance of Alpaca compared to text-davinci-003 [9]. However, Alpaca does share common language model limitations, such as generating false information and perpetuating social stereotypes. Further, although 7 billion parameters are already small in terms of Large Language Models, Alpaca 7B's size remains a barrier. Hence, the need for more compact models is evident. Parameter-efficient Fine-tuning (PEFT) techniques like Low Rank Adaptation (LoRA) come into play as a potential solution for size constraints without compromising performance.

Low Rank Adaptation LoRA Emphasizing efficiency in advanced natural language processing, and because of limitations to the available hardware, Low-Rank Adaptation (LoRA) [6] was applied to the Alpaca 7B model. This technique introduces trainable rank decomposition matrices at each Transformer layer, substantially reducing the number of trainable parameters.

The impact of LoRA on Alpaca 7B has been transformational, reducing trainable parameters to a mere 16 million, while preserving model performance [1]. This reduction mitigates computational demands and cost constraints, rendering deployment of the model more feasible.

Learnable Evaluation Metric for Text Simplification (LENS) LENS was developed to address limitations in current text simplification evaluation metrics. Leveraging a modern language model, LENS is trained on the SimpEval corpus, a robust dataset featuring human ratings of text simplifications from multiple sources, including GPT-3.5. Through this method, LENS captures nuanced aspects of text simplification that conventional metrics might overlook. The crux of its functionality lies in its adaptivity: LENS adjusts and improves as it encounters more data, increasing the accuracy and relevance of its evaluations. By aligning more closely with human judgment than traditional metrics, LENS offers a promising tool for evaluating and advancing text simplification technologies, as shown by Maddela *et al.* [4].

3. Dataset

During our training procedure, we primarily utilized the Task 3 dataset for fine-tuning our models. Task 3 comprised a parallel corpus of simplified sentences originating from Medicine and Computer Science domains. The simplification was performed either by a master student in Technical Writing and Translation or by an expert duo consisting of a computer scientist and a professional translator [3]. Despite its quality, the Task 3 dataset posed a challenge in terms of data scarcity, as it consisted of only 648 sentence pairs.

In an effort to address this challenge and enhance the stability of our model, we incorporated data from Task 2. The Task 2 dataset, also drawn from Medicine and Computer Science domains, consisted of scientific abstracts sourced from the Citation Network Dataset and Google Scholar and PubMed articles focusing on muscle hypertrophy and health. These were annotated by a master student in Technical Writing and Translation, who assigned difficulty scores to extracted terms, resulting in a total of 453 annotated examples [3].

Our approach was to integrate the Task 2 dataset into the training process to assess if a larger amount of data, pointing out difficult terms, could potentially boost the model's performance. By merging Task 2 and Task 3 datasets, we aimed to not only increase the quantity of the training data but also its diversity, thereby enhancing the model's overall performance and generalization capabilities.

It is important to highlight that our approach did not involve the utilization of any supplementary data or the implementation of other data augmentation techniques. We solely relied on the data from Task 3 as the *default* dataset, and the integration of Task 2 data resulted in the *extended* dataset. In the subsequent sections, we will refer to the dataset sourced from Task 3 as the *default* dataset and the combined dataset from Task 2 and Task 3 as the *extended* dataset.

4. Methodology

In this section, we outline our research methodology involving the utilization of the Alpaca 7B model optimized by Low Rank Adaptation (LoRA) for text simplification. We employed strategies such as prompt engineering, identification of complex terms, rigorous fine-tuning, and a unique two-option evaluation process. Furthermore, we used the LENS score for assessment and model fusion to combine the strengths of multiple models. The overall setup of our submission is shown in Figure 2.

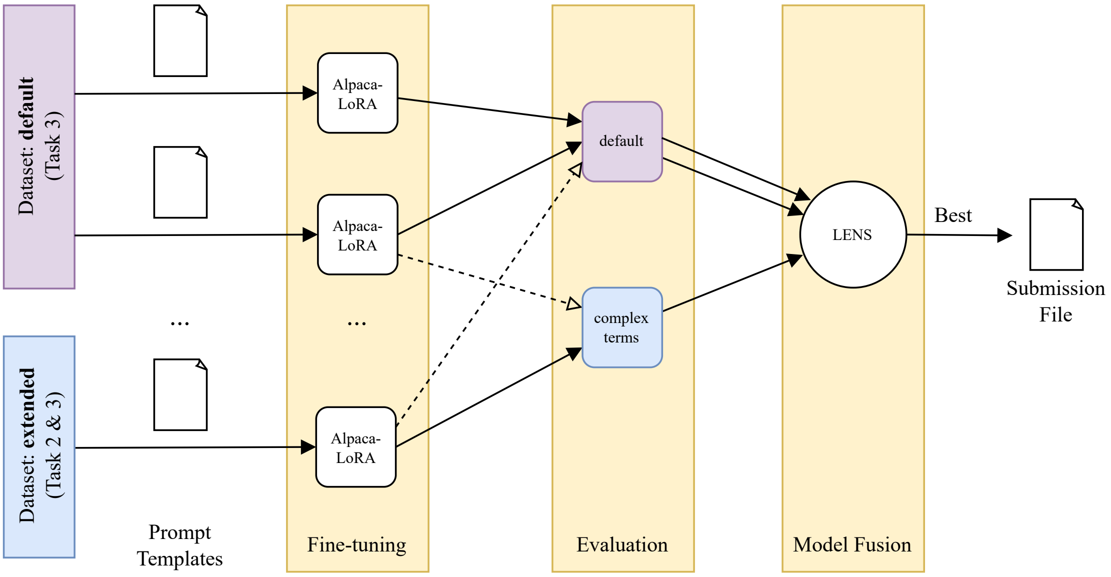


Figure 2: Schematic depiction of the training architecture. Input data, originating either from Task 3 (*default*) or a combination of Task 2 and 3 (*extended*), is paired with predefined prompt templates. The models are then fine-tuned using a wide variety of combinations of the hyperparameters. After fine-tuning, these models are evaluated on two tasks: (1. *default*) direct text simplification or (2. *complex terms*) a combination of intermediate outcomes from Task 2 (identification and explanation of challenging terms) with text simplification. Performance assessment is further conducted by computing the LENS score of text simplifications generated on the test set. The output attaining the highest LENS score was selected for the final submission.

4.1. Explored Model Architectures

In this section, we conducted tests on various language models and manually evaluated them to determine the most suitable model for the given task. We analyzed T5 and Alpaca LoRA 7B models.

Text-to-Text Transfer Transformer (T5) In our initial experimentation, we adopted T5, `t5-base` from Hugging Face, as a baseline model for the text simplification task. However, we encountered a significant limitation with this approach. The output generated by the T5 model was essentially identical to the input, which rendered it ineffective for our purposes. As a result, we swiftly rejected the idea of using T5 as our primary model.

Alpaca LoRA 7B Even without fine-tuning, the Alpaca LoRA 7B model demonstrated potential in our preliminary manual evaluations of the text simplification outputs. One example can be seen in Figure 3. Thus, we pursued further investigation and optimization of this model to potentially extend its usability in the context of text simplification.

4.2. Prompt Engineering

We conducted experiments using three distinct prompt templates: *default*, *short*, and *simplification*. The *simplification* prompt template aimed to tackle the challenge of explaining complex concepts in a way that is understandable to a 5-year-old child. By utilizing this prompt template, we sought to simplify the language and structure of the generated responses, making them more accessible and easier to comprehend. These templates were designed to provide specific instructions to guide the model's generation process and explore different approaches to improve the output quality.

For a detailed list of the prompt templates used in our experiments and their corresponding instruction prompts, please refer to Appendix A.

Input *snt_id: G06.2_2805209921_6*: we have proven that transfer learning is not only applicable in this field, but it requires smaller well-prepared training datasets, trains significantly faster and reaches similar accuracy compared to the original method, even improving it on some aspects.
Output default: Transfer learning is not only applicable in this field, but it requires smaller well-prepared training datasets, trains significantly faster and reaches similar accuracy compared to the original method, even improving it on some aspects.
Output simplification: Transfer learning is applicable in this field, trains significantly faster and reaches similar accuracy compared to the original method, even improving it on some aspects.

Figure 3: Comparison of Alpaca LoRA 7B model's text simplification output without fine-tuning. *default* prompt template shows the model's output with a default prompt which is nearly identical to the input. *simplification* prompt template presents a simplified version, showing the model's potential for text simplification without any fine-tuning.

4.3. Complex Terms Identification

First, we employ the model to identify complex terms within a given input text. We then retrieve definitions for these complex terms, providing a way to make them more understandable. Now we have a set of complex terms (e.g., *transfer learning*) and their definition. Next, we provide the model with these definitions as well as the original text to create a simplified version of the text. By integrating the explanations of complex concepts into the simplified text, we intend to minimize the need for prior knowledge, thus enhancing the text's understandability for a wider audience. The process is visualized in Figure 4. The idea of enhancing our model's performance by utilizing intermediate results draws inspiration from the *Chain-of-Thought Prompting* approach. This strategy significantly improved the outcomes of large language models in tasks like complex reasoning [11]. In the following section, we refer to this idea of using complex terms and their definitions as *complex terms*.

4.4. Fine-Tuning

The fine-tuning process was carried out on numerous models, incorporating the two distinct datasets presented in Section 3 (*default* and *extended*), three varying prompt templates presented in Section 4.2 (*default*, *short*, and *simplification*), as well as a variety of hyperparameters.

The training was subjected to a series of hyperparameter tuning experiments. In this context, the number of epochs was varied, ranging from 3 to 10, to investigate the optimal duration for training to balance the trade-off between model performance and computational efficiency.

Another important aspect of the training was the learning rate scheduler. The use of the learning rate scheduler aimed to optimize the learning rate during the training process, adapting it based on the progress of the training. Two batch sizes, 32 and 64, were considered to observe their influence on the model's learning and performance. It's worth noting that a single training epoch typically took around 8 minutes, highlighting the computationally intensive nature of the procedure. Please refer to Appendix B for detailed information on the hardware used.

In addition to the aforementioned parameters, we evaluated the performance of various input prompts using the three given prompt templates. This involved experimenting with diverse instructions and task descriptions, as these factors can significantly influence the effectiveness of the model's training and eventual performance.

4.5. Evaluation of Fine-Tuned Models

The fine-tuned models from the previous section were evaluated using two different methods: (1. *default*) where text simplification was exclusively applied to the source sentence, following

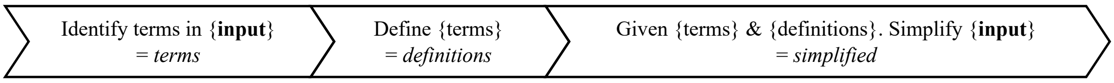

Identify terms in {**input**}
= *terms*
Define {terms}
= *definitions*
Given {terms} & {definitions}. Simplify {**input**}
= *simplified*

Figure 4: Illustration of the complex term identification pipeline. The process involves identifying and explaining difficult terms first and then performing the text simplification task using intermediate results.

the default approach, or (2. *complex terms*) where the process involved starting with complex terms, providing explanations for those terms, and leveraging all intermediate results to simplify the text, as elaborated in Section 4.3. Evaluating the fine-tuned models on the *default* dataset should naturally align with the *default* method, and similarly, the *extended* dataset with the *complex terms* approach. Nevertheless, we tried an alternative, which involves blending these approaches, illustrated by the dotted line in Figure 2.

4.6. Model Fusion

In our approach, we leverage the power of ensemble learning to tackle the task of text simplification. Our goal is to enhance the understandability and readability of text by combining the outputs of multiple models and selecting the best result for each sample. We do this with LENS [4], which forms a critical aspect of our evaluation process.

5. Results

In this section, we elaborate on the findings from our experiments. We noted that some models, like T5 or specific configurations of the Alpaca LoRA model using the *short* template with the *default* dataset, showed inadequate performance during testing. The simplifications made by these models were either identical to the input or incomplete, with some instances resulting in the repeated use of the same word. Given these limitations, we chose not to further investigate or utilize these models in our project.

One interesting result we observed was that our approach using the Alpaca LoRA 7B resulted in a 10% average reduction in sentence size. However, shorter sentences do not necessarily equate to better readability. The readability improvement rate was found to be within the range of 2%-5%. Like the T5 model, our fine-tuned models also had their share of imperfections. These included instances of incomplete translations, representation of summaries as bullet points, and in some cases, over-complication of the simplification process. The latter was particularly noticeable when the model using *complex terms* introduced too many definitions, which counterintuitively complicated the text rather than simplifying it. These are areas that require attention for improvement in subsequent iterations of our text simplification model.

The experimentation also revealed an intriguing trend where simpler methods often outperformed their more complex counterparts. For example, using the *default* dataset, coupled with fine-tuning via the *simplification* prompt template and evaluation using the *default* method, yielded a selection rate of over 50% in the model fusion phase. This showcases the potential effectiveness of simpler training and evaluation processes.

Fine-tuning with the *extended* dataset and evaluating using *complex terms* with the *simplification* prompt template resulted in a selection rate of over 20%. This further underlines the importance of customizing the training and evaluation processes based on the unique demands of text simplification tasks. This specific model stands out in simplifying and elaborating on complex terms. It provided detailed explanations which aided the simplification process. An example of this can be seen in Figure 1. Here, the model clarifies the concept of *transfer learning* before beginning the summarization. This example demonstrates how our model successfully adds new information to help to understand and simplify complex concepts.

The model also demonstrated proficiency in explaining abbreviations. For instance, it could expand www into `World Wide Web (WWW)` and p2p into `peer-to-peer (P2P)`. This capability adds another layer of utility in simplifying and clarifying complex text by demystifying unfamiliar abbreviations for the reader.

However, not all approaches proved successful. The *short* prompt template, for instance, consistently underperformed across all provided configuration choices. This stresses the necessity to strike a balance between simplicity and effectiveness in text simplification tasks. Figure 5 shows the selection rate of the models during the model fusion phase.

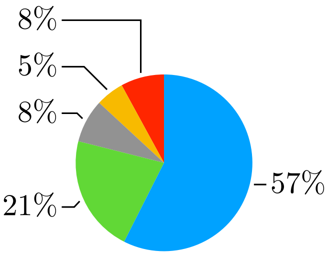
In addition to our evaluation, the official evaluation results utilizing established metrics such as SARI, BLEU, and FKGL can be found in Appendix C. These results offer a comprehensive assessment of the performance of our text simplification models.

6. Limitations

Our approach, while promising in its outcomes, is subject to certain limitations that are essential to acknowledge. One of the issues is the occasional equivalence of input and output. While this does not increase the complexity of the input, it is not the outcome we desire. Ideally, a simplification model should always render an output that is less complex and more comprehensible than the input. However, our model does not consistently ensure this.

Furthermore, some cases require additional information to simplify a sentence effectively, such as providing a simpler explanation for a complex term or concept. We attempted this with our model using the *extended* dataset and *complex terms* for evaluation. However, these results might not be incorporated into the model fusion phase. This is because our model fusion phase struggles to automatically evaluate scenarios where the simplification significantly deviates from the original input due to the lack of a reliable metric. This limitation presents a challenge in delivering accurate text simplification when necessary.

Even with the useful LENS score as a tool for evaluating simplification, it carries inherent limitations, particularly when applied to unlabeled data. For example, instances of excessive explanatory information or elaboration can be misinterpreted as hallucinations by the LENS evaluator, consequently leading to inaccurately low scores. Moreover, when employing Large Language Models (LLMs), the issue of hallucinations continues to pose a significant challenge, underscoring an area that requires further exploration and methodological refinement.



	Dataset	Prompt Template	Evaluation Method
🔵	default	simplification	default
🟢	extended	simplification	complex terms
⚫	extended	default	complex terms
🟡	default	default	default
🔴	other		

Figure 5: Selection rate of the models during the model fusion phase.

An additional limitation lies in the common misconception equating text simplification to summarization. While both share the aim of rendering complex information more digestible, they are distinct tasks. Summarization concentrates on shortening the text while retaining its core ideas, whereas simplification is dedicated to reducing complexity, which does not necessarily involve decreasing the text's length. This divergence in objectives introduces unique challenges not fully addressed by our present model.

Constraints concerning time and computational power further limit the exploration of diverse approaches and models. Identifying suitable evaluation metrics that accurately measure the effectiveness of text simplification also remains a substantial challenge. These factors underline the complexities and challenges involved in the quest for effective text simplification methodologies.

7. Conclusion

The field of text simplification presents various challenges, but our research has revealed promising pathways. Simple approaches utilizing pretrained large language models, targeted prompts, and adapted training strategies can lead to significant strides towards more effective text simplification. The findings emphasize the importance of using the right techniques and prompts to find a balance between simplicity and effectiveness in text simplification tasks.

A successful text simplification process should aim to elucidate complex words that may be unfamiliar to the reader and expand abbreviations to ensure clarity. These elements are crucial in reducing textual complexity without compromising on the richness of the information being conveyed.

Acknowledgments

We would like to take this opportunity to express our deepest gratitude and sincere appreciation to Andrianos Michail, and Simon Clematide for their unwavering support and constructive guidance during the creation of this paper.

References

[1] E. J. Wang, Alpaca-lora, 2023. URL: https://github.com/tloen/alpaca-lora.

[2] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models (2023). arXiv:2302.13971.

[3] L. Ermakova, E. SanJuan, S. Huet, O. Augereau, H. Azarbonyad, J. Kamps, Overview of simpletext - clef-2023 track on automatic simplification of scientific texts, in: Avi Arampatzis, Evangelos Kanoulas, Theodora Tsikrika, Stefanos Vrochidis, Anastasia Giachanou, Dan Li, Mohammad Aliannejadi, Michalis Vlachos, Guglielmo Faggioli, Nicola Ferro (Eds.) Experimental IR Meets Multilinguality, Multimodality, and Interaction., Proceedings of the Fourteenth International Conference of the CLEF Association, 2023.

[4] M. Maddela, Y. Dou, D. Heineman, W. Xu, Lens: A learnable evaluation metric for text simplification (2023). arXiv:2212.09739.

[5] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, H. Hajishirzi, Self-instruct: Aligning language model with self generated instructions, 2023. arXiv:2212.10560.

[6] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models (2021). arXiv:2106.09685.

[7] A. Rubio, P. Martínez, Hulat-uc3m at simpletext@ clef-2022: Scientific text simplification using bart, Proceedings of the Working Notes of CLEF (2022).

[8] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, The Journal of Machine Learning Research 21 (2020) 5485–5551.

[9] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, T. B. Hashimoto, Stanford alpaca: An instruction-following llama model, 2023. URL: https://github.com/tatsu-lab/stanford_alpaca.

[10] OpenAI, Openai: Models, 2023. URL: https://platform.openai.com/docs/models/models.

[11] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. Chi, Q. Le, D. Zhou, Chain of thought prompting elicits reasoning in large language models (2022). arXiv:2201.11903.

A. Prompt Engineering

A.1. Templates

Default Default prompt template provided by Alpaca-LoRA [1]:

```
Below is an instruction that describes a task,
paired with an input that provides further
context. Write a response that appropriately
completes the request.
###
Instruction:
{instruction}
###
Input:
{input}
###
Response:
```

Short Short prompt template without any instruction:

```
###
Instruction:
{instruction}
###
Input:
{input}
###
Response:
```

Simplification Modified prompt template specifically for simplification tasks:

```
Below is an instruction that describes a
simplification task, paired with an input
that provides further context.
Write a simple response that appropriately
completes the request. Write your response
as you would talk to a 5-year-old.
###
Instruction:
{instruction}
###
Input:
{input}
###
Response:
```

A.2. Instructions

This section shows the prompts for the two datasets used in this paper. The instructions are interpolated in the previously provided template.

Default For the default evaluation process, we used a simple instruction prompt.

```
Simplify the following sentence
```

Complex Terms For the complex terms evaluation, we used chained the model using the following two instruction prompts.

1. To identify the difficult ***terms***:

   ```
   Decide which terms (up to 5) require
   explanation and contextualization to
   help a reader understand a complex
   scientific text
   ```

2. To obtain definitions for the previously identified ***terms***:

   ```
   Provide a short (one/two sentence)
   explanations/definitions for the detected
   difficult terms: {term} in
   the context of the following sentence:
   ```

B. System Environment

This section provides an overview of the system configuration and software dependencies employed in the development and implementation of our research system.

To fine-tune and perform inference with the model, 8 cores 32 GB RAM with a 16 GB Tesla T4 GPU was utilized, providing the necessary computational resources. The model we used is `chainyo/alpaca-lora-7b` from Hugging Face, which is a LLaMA-7B fine-tuned model on the Stanford Alpaca cleaned version dataset.

C. Official Assessment

This section presents a detailed analysis of the official evaluation of the SimpleText CLEF shared task. All figures in this section show a graphical representation of the ranking of all submissions.

Figure 6 shows the FKGL scores, a measure of text complexity. Here, our ensemble model outperforms all the models we submitted, achieving the lowest FKGL score. This result is expected given the way our model fusion works, which always selects the simplest version of our models.

Figure 7 shows the SARI scores. Once again, our ensemble model stands out with the highest SARI score of all our submissions. As SARI is a critical criterion for our model fusion, this high score is not surprising. However, it is worth noting that models that clarify complex terms can deviate significantly from the target sentence, and consequently receive a lower score. This is despite the potential improvement in overall readability and comprehension.

Figures 8 and 11 show the BLEU scores and Levenshtein similarity scores respectively. Both metrics highlight a correlation between models that produce outputs that are very similar to their inputs. Since this pattern was observed in our models without fine-tuning or prompt engineering, it is expected that these models would also achieve the highest overall BLEU scores. It is clear from these figures that preserving the original sentence structure and content contributes significantly to higher scores.

In conclusion, our ensemble model, which is designed to incorporate the strengths of all our models, consistently shows superior performance across multiple evaluation metrics. However, it is important to keep in mind that optimal text simplification may involve an acceptable level of deviation from the original sentence, if it results in improved readability and comprehension for the intended audience.

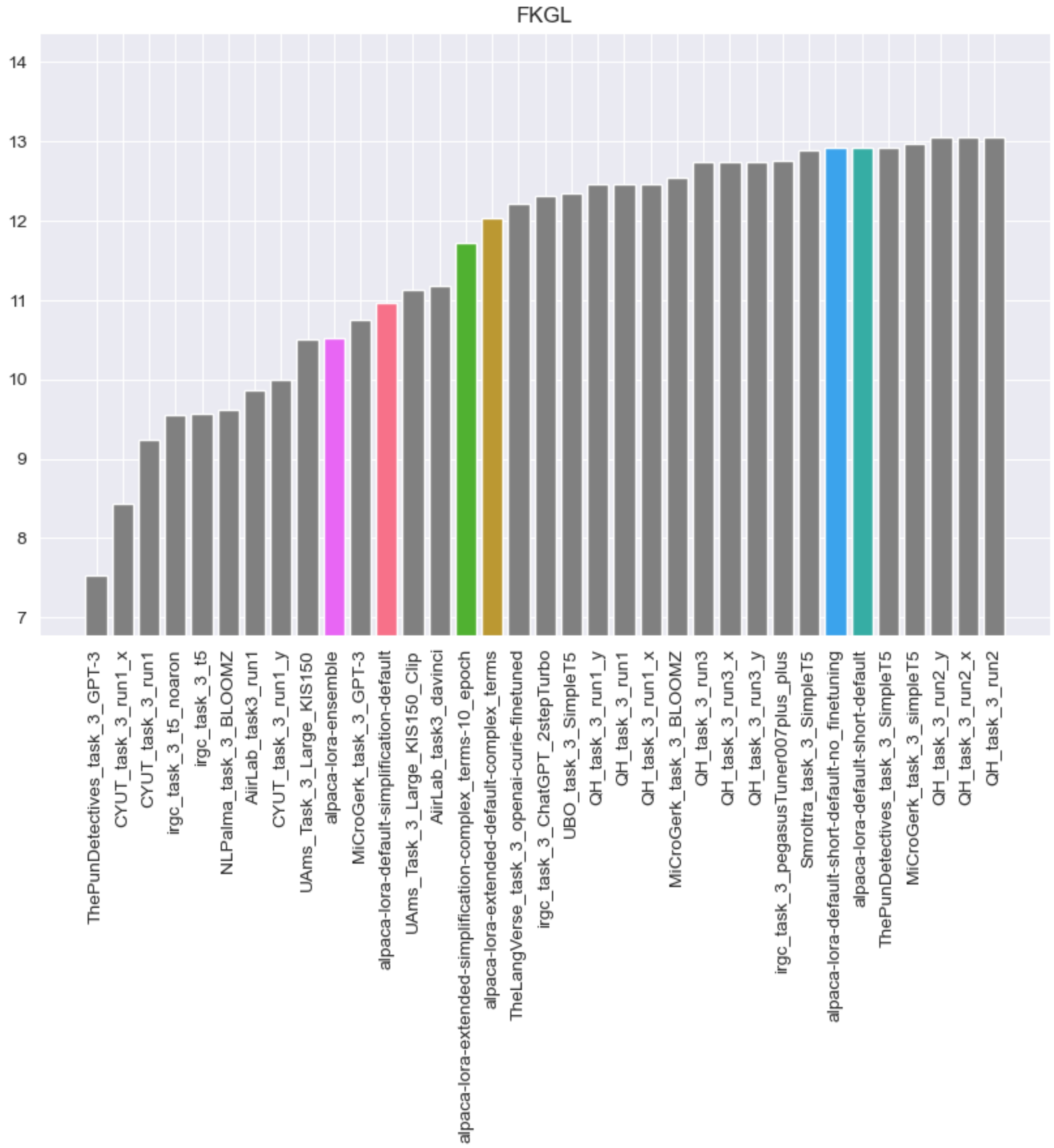


Figure 6: FKGL (Flesch-Kincaid Grade Level) scores for the text simplification models. The FKGL metric measures the grade level required to understand the text, with lower scores indicating simpler and more accessible language.

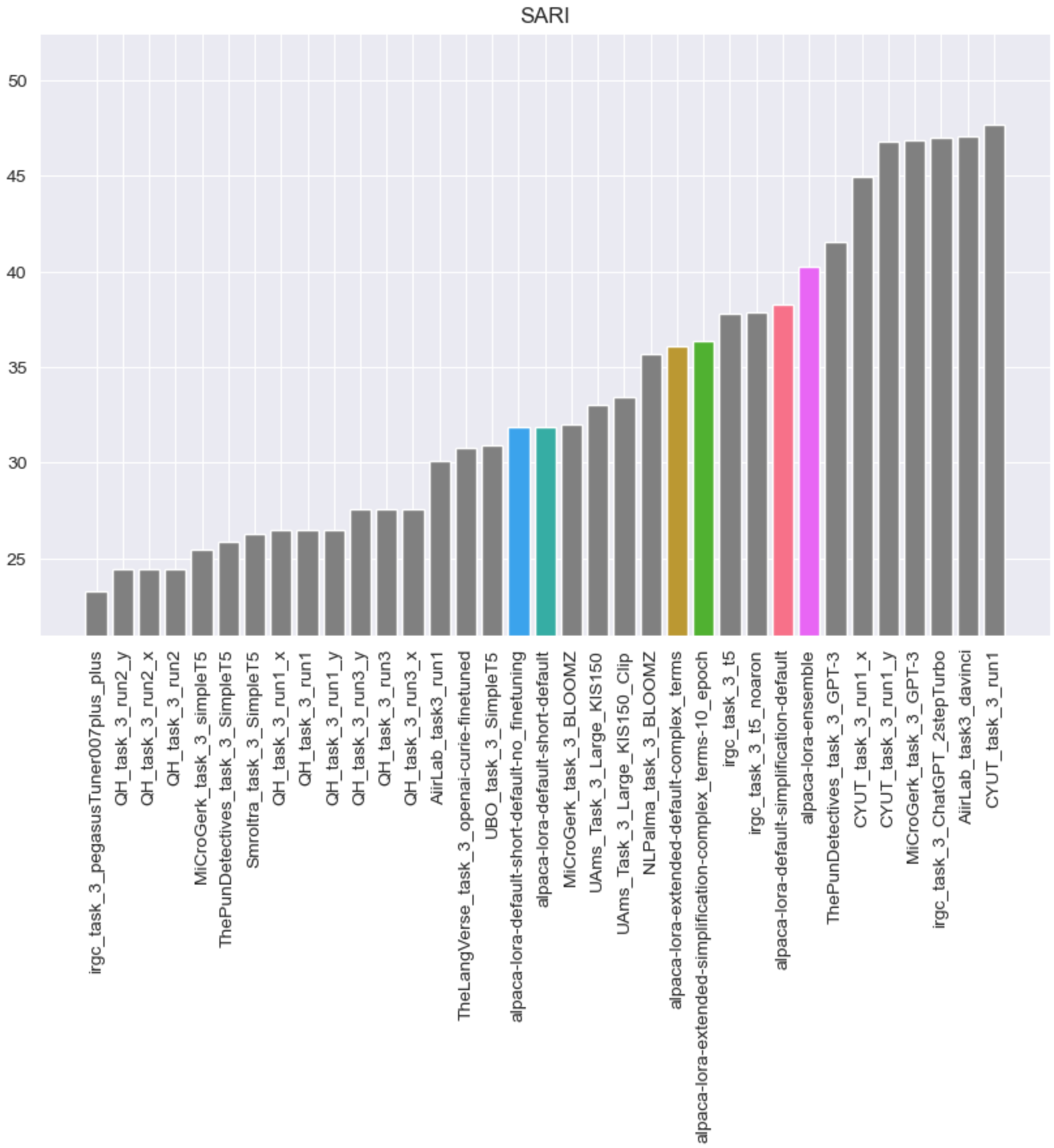


Figure 7: SARI (System-level Automatic Evaluation Metric for Text Simplification) scores for text simplification models. Higher values indicate better quality simplifications. SARI is a popular metric that evaluates the overall quality of text simplification by comparing the generated simplified text to reference simplifications.

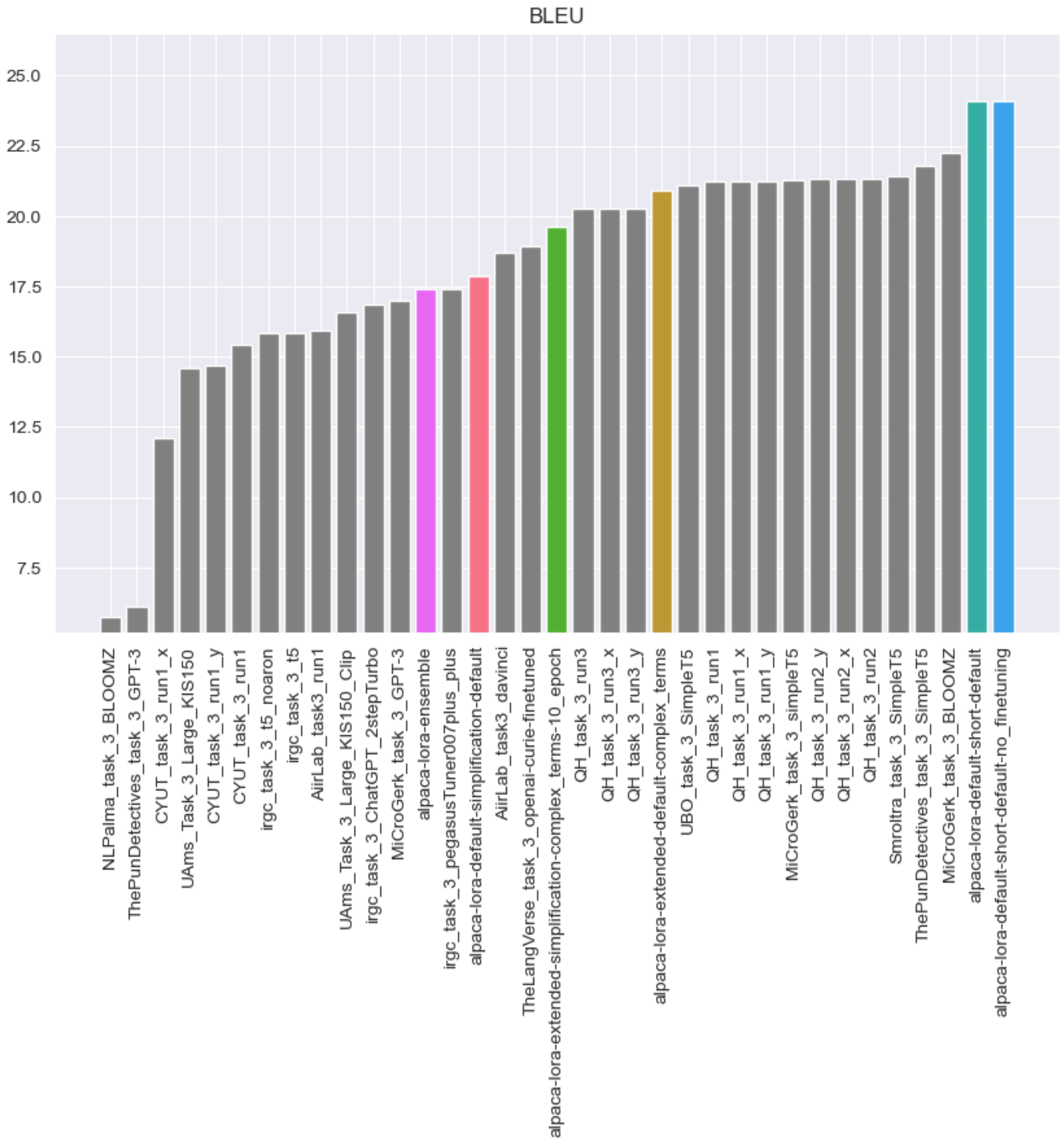


Figure 8: BLEU (Bilingual Evaluation Understudy) scores for the text simplification models. BLEU measures the overlap between the generated simplified text and reference simplifications, with higher scores indicating better similarity.

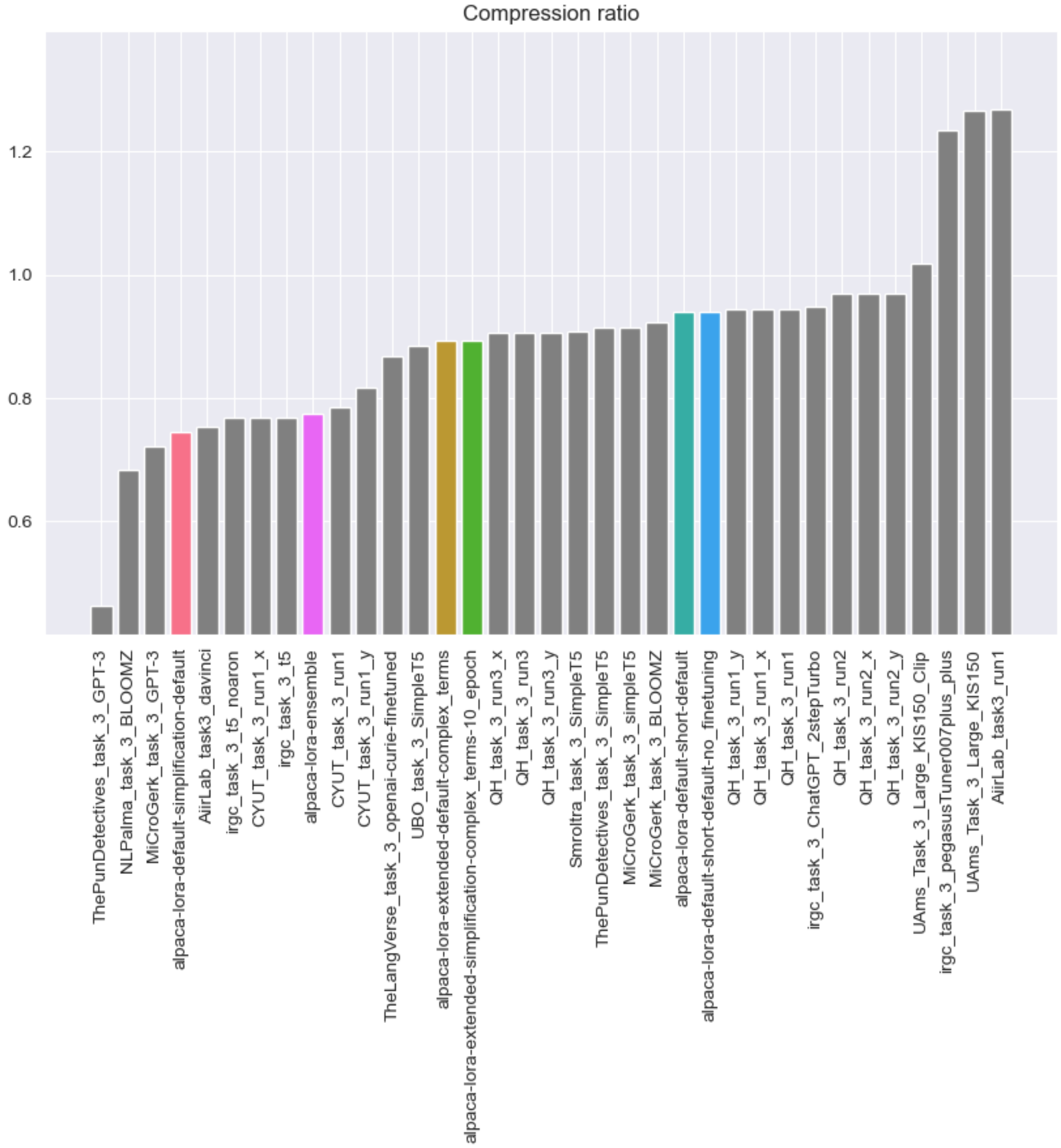


Figure 9: Compression ratios for the text simplification models. Compression ratio measures the reduction in sentence length achieved by the text simplification models, with higher values indicating more significant simplification.

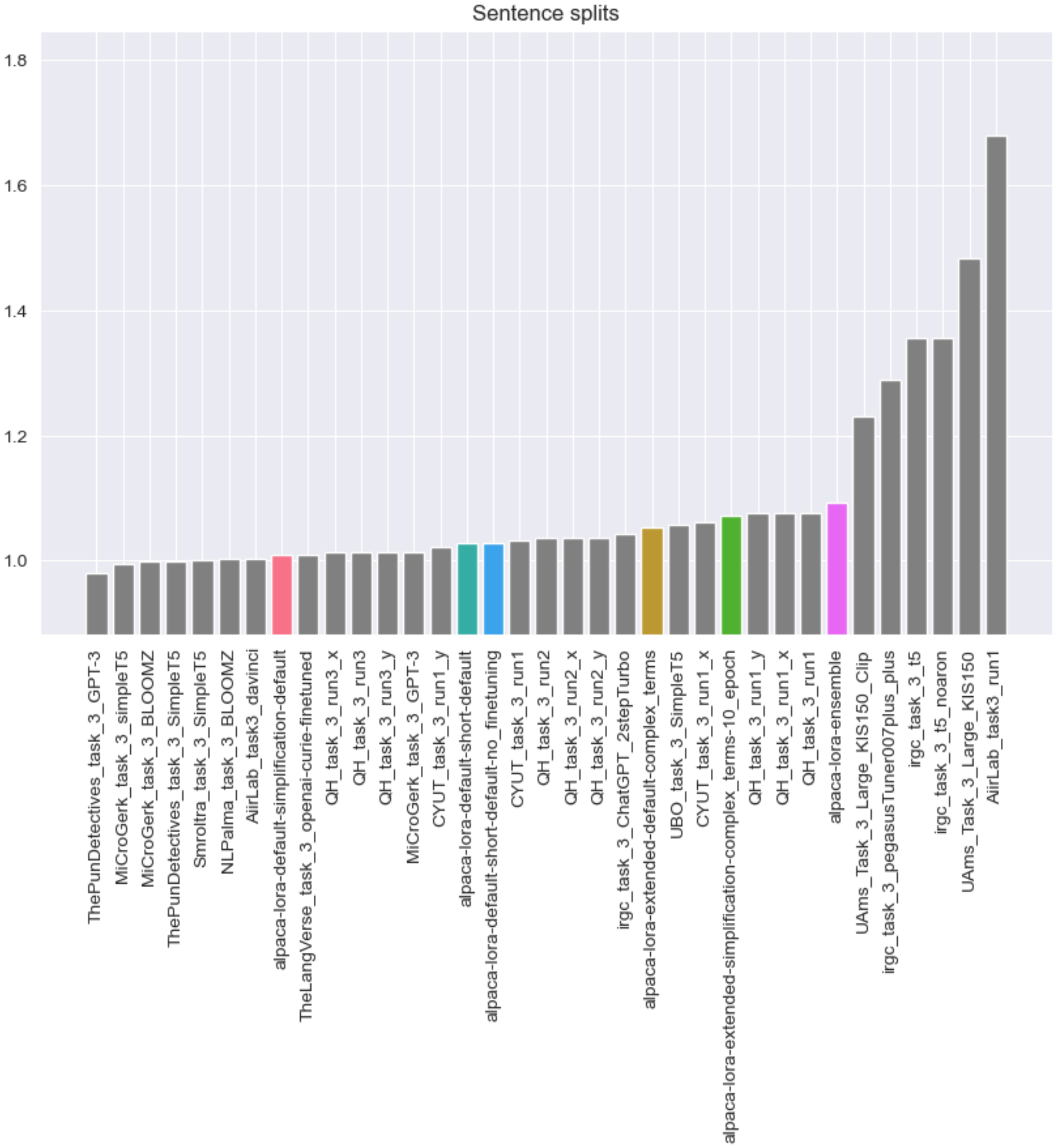


Figure 10: Number of sentence splits for the text simplification models. Sentence splits measure the extent to which the original sentences were divided during the simplification process, with lower values indicating better preservation of sentence structure.

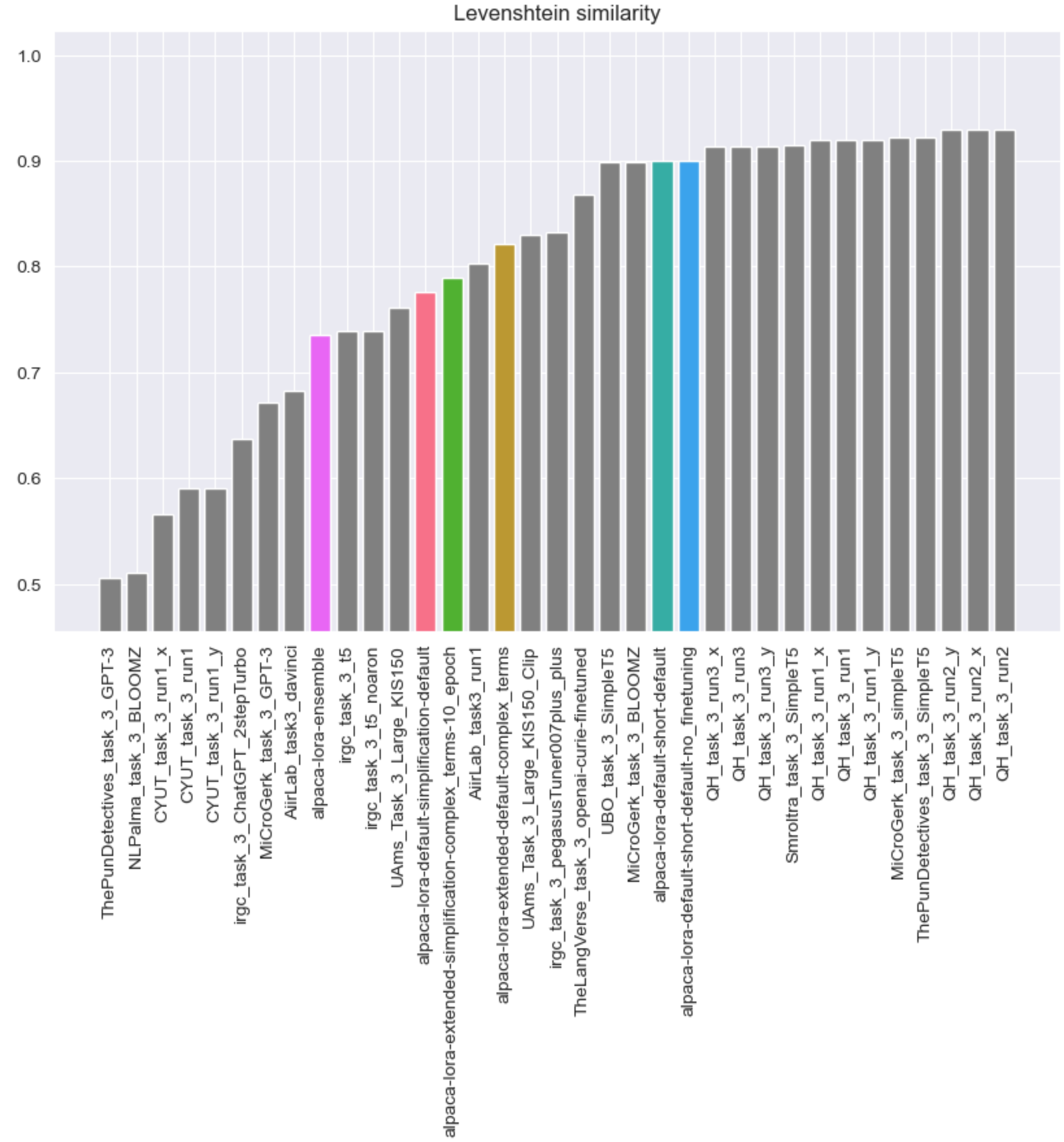


Figure 11: Levenshtein similarity scores for the text simplification models. Levenshtein similarity measures the similarity between the generated simplified text and the original text, with higher values indicating better preservation of the original content.

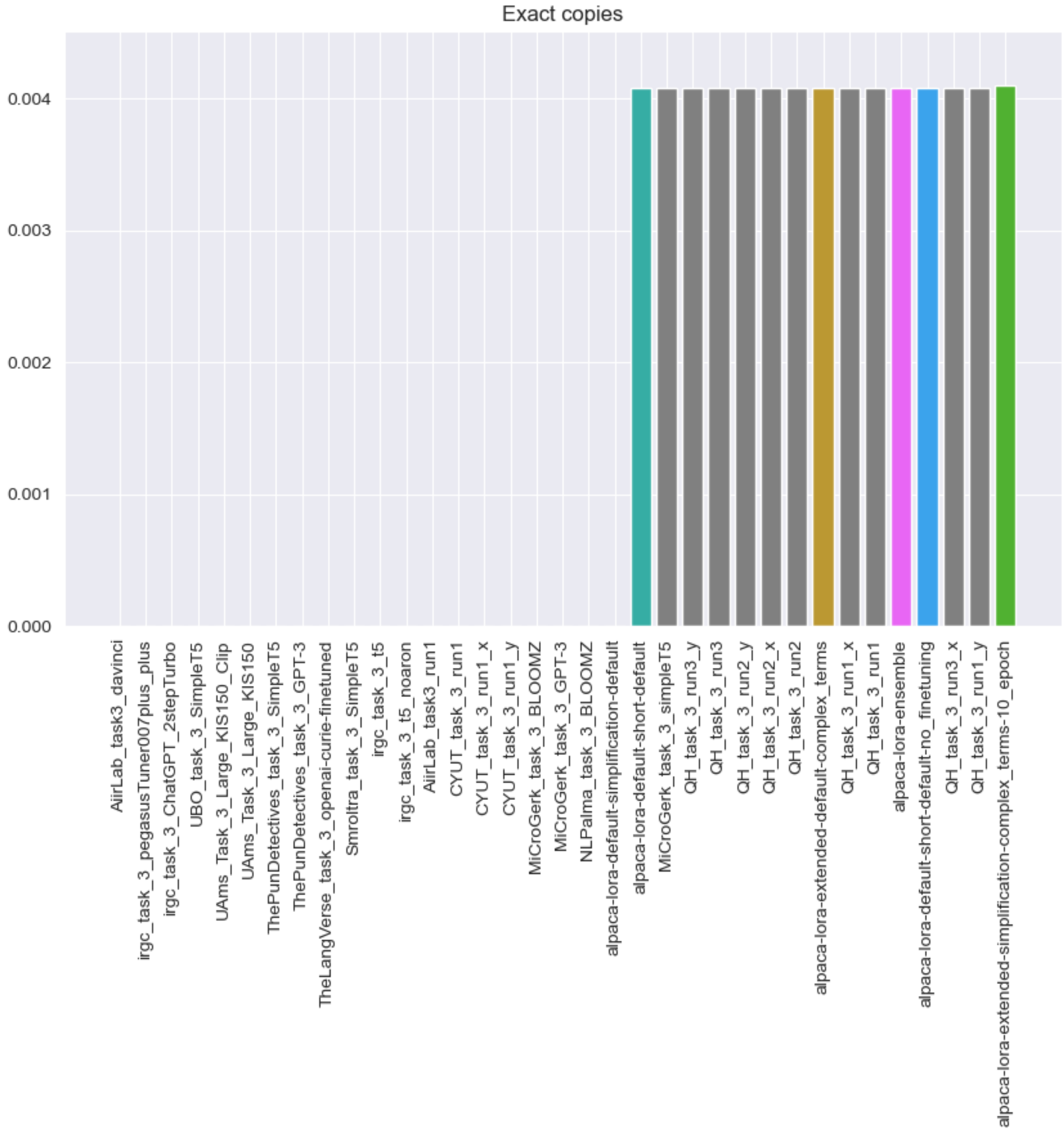


Figure 12: Percentage of exact copies for the text simplification models. This metric measures the extent to which the generated simplified text is an exact copy of the original text, with lower values indicating better paraphrasing and simplification.

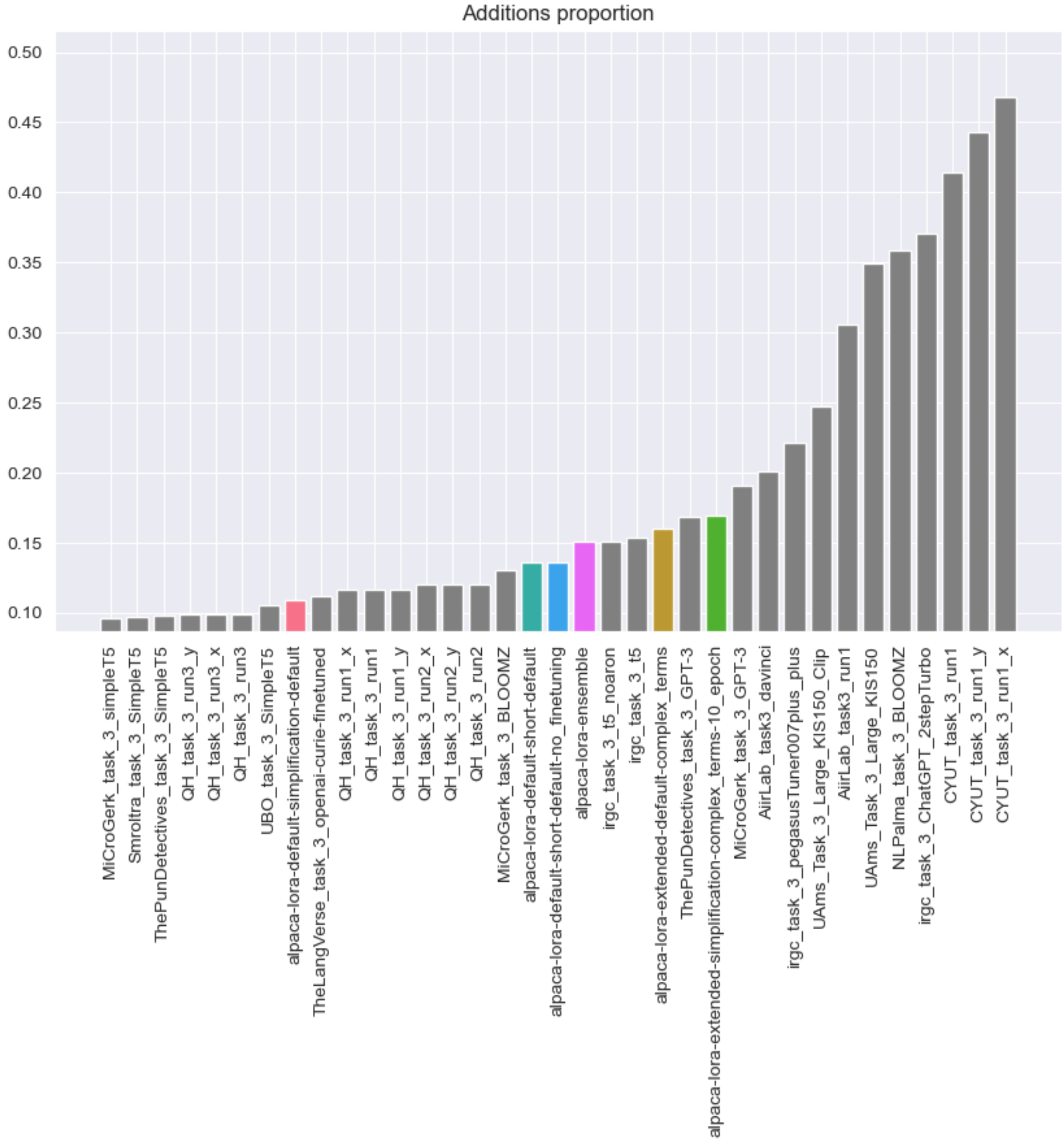


Figure 13: Proportion of additions for the text simplification models. This metric measures the extent to which additional information was introduced during the simplification process, with lower values indicating better adherence to simplicity.

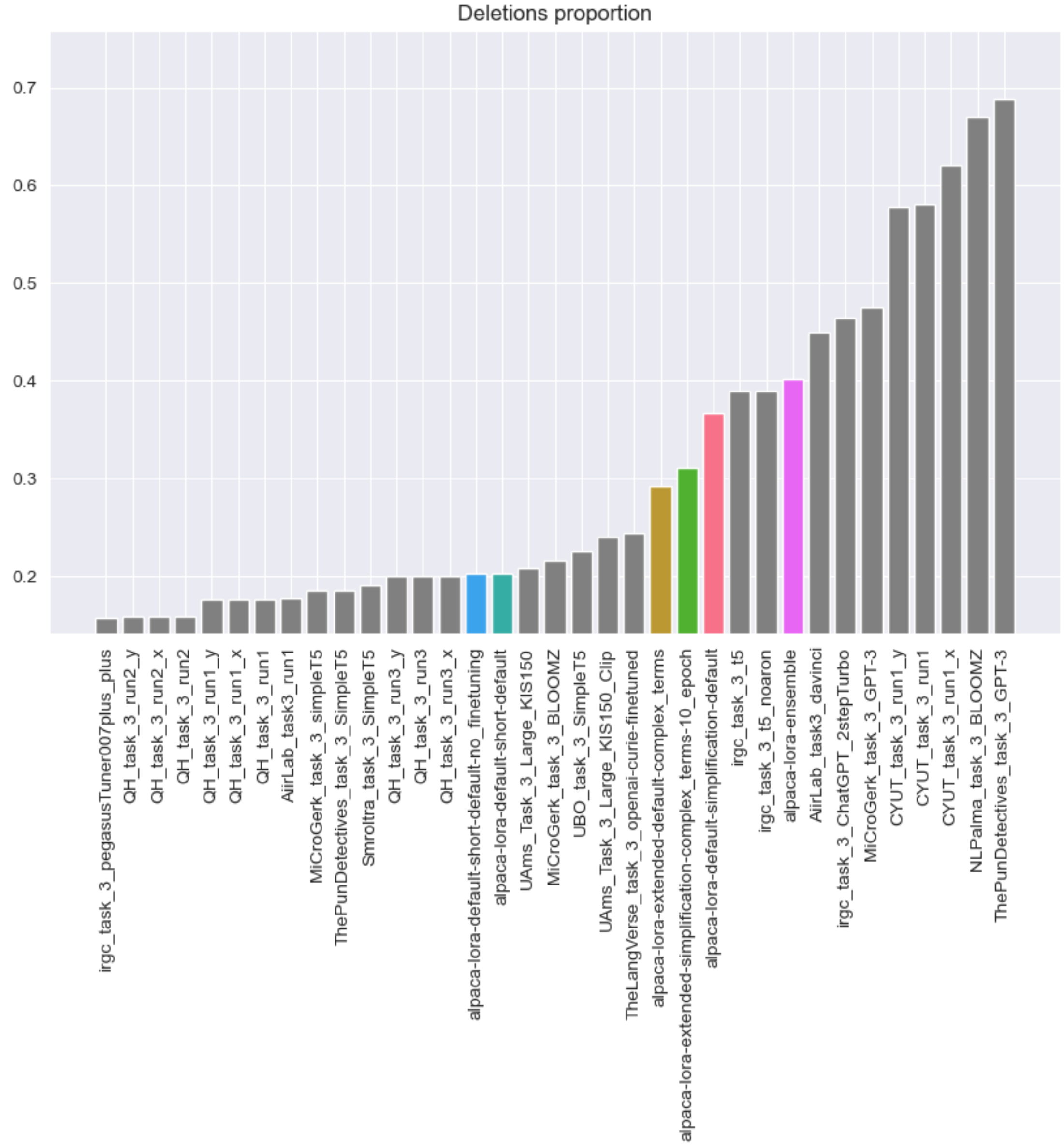


Figure 14: Proportion of deletions for the text simplification models. This metric measures the extent to which unnecessary or redundant information was removed during the simplification process, with lower values indicating better conciseness.

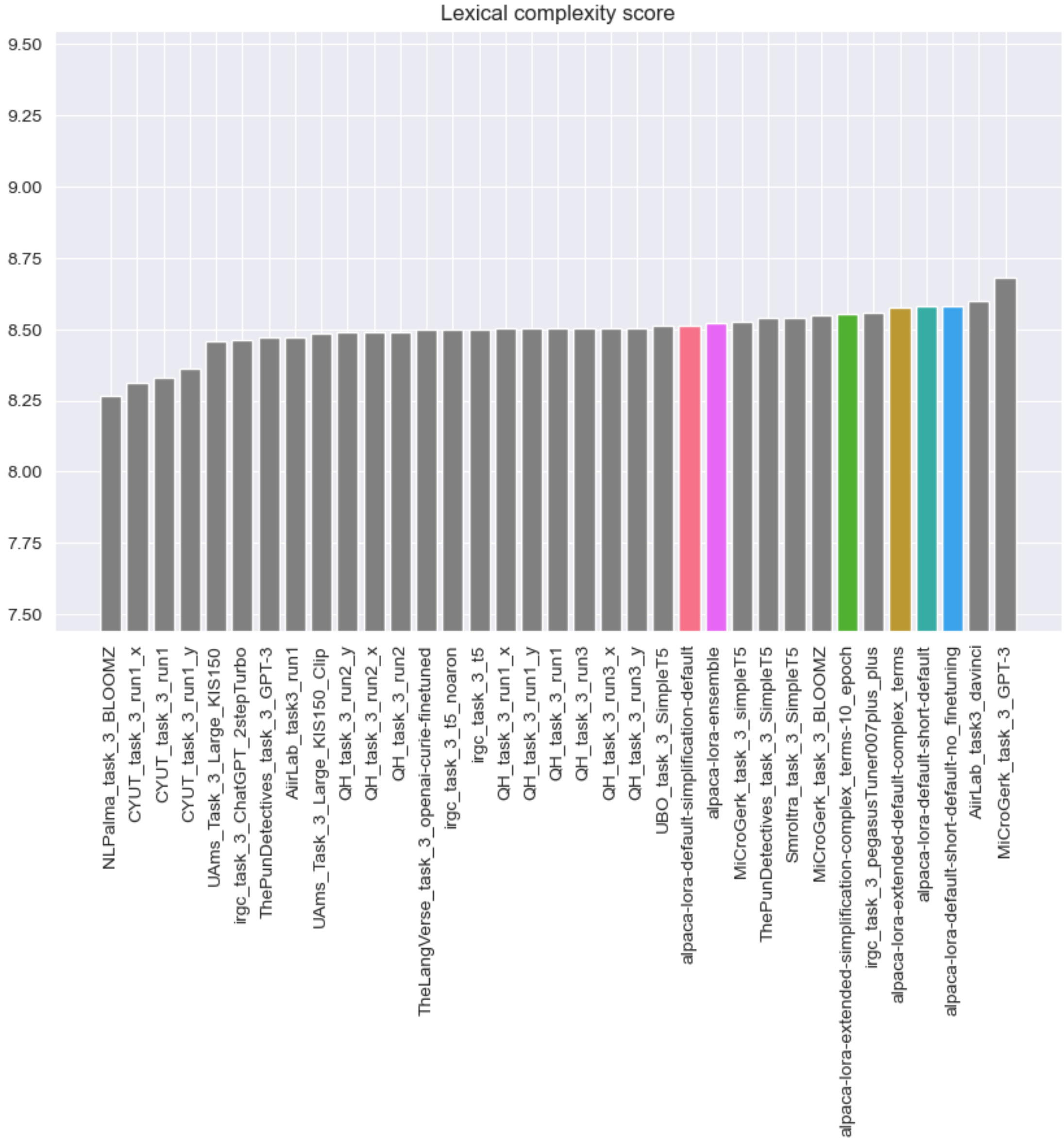


Figure 15: Lexical complexity scores for the text simplification models. This metric measures the complexity of the vocabulary used in the generated simplified text, with lower scores indicating simpler and more accessible language.