clnn88@SimpleText 2026 - Task 1: Llama 3-based Simplification for Biomedical Text

Clyde Lanvin Njinpie Noutche[1], Christin Katharina Kreutz[1,2,*]

[1]TH Mittelhessen - University of Applied Sciences, Gießen, Germany
[2]Herder Institute, Marburg, Germany

Abstract

Automatic simplification of scientific content enables non-experts to use high-quality information in important decisions, such as personal healthcare.

The SimpleText initiative has been advancing automatic text simplification. In this year's lab, they offer two tasks, one focusing on sentence-level simplification and one tackling document-level simplification of biomedical text. We investigate the influence of different formulations of instructions in zero-shot prompting with an open source large language model (Meta-Llama-3-8B-Instruct). We produce six runs, three for both subtasks. We found considerable performance differences between our runs hinting at the importance of the exact wording used in prompts in addition to types of included instructions.

Keywords

Text Simplification, Llama 3, Prompt Engineering, Biomedical Text

1. Introduction

The number of scientific publications has been increasing rapidly in the last years. Access to high-quality information from such papers is essential for informed decision-making in healthcare, education, and public communication [1]. However, scientific texts are often difficult for non-expert readers to understand due to usage of specialized terminology and their high density of factual information.

Automatic text simplification aims to address this challenge by translating complex texts into more accessible versions while preserving their original meaning. The SimpleText@CLEF'26 lab [1, 2] offers a shared task [3] for scientific text simplification at sentence- (Task 1.1) and document-level (Task 1.2). Recent advances in Large Language Models (LLMs) have demonstrated strong capabilities in text generation and rewriting, making them promising tools for simplification tasks [4, 5, 6].

Previous approaches include prompt-based simplification, fine-tuned language models, and summarization-oriented architectures [4]. Building upon these findings, this work investigates the use of an open source LLM (Meta-Llama-3-8B-Instruct) and zero-shot prompting. In our prompts, we focus on different formulations of (comparable) instructions and their influence on simplification performance. In total we submit six runs, three runs for the sentence-level and three runs for the document-level subtasks.

2. Related Work

Several works closely related to our approach stem from the previous year's iteration of the SimpleText text simplification task [4].

The SINAI team [7] used GPT-4.1 and experimented with zero-shot prompts defining the role or persona of the LLM that appear rather extensive and describing the objective of simplification clearly. The UM_FHS team [8] compared direct prompt-based simplification using GPT-4.1 against fine-tuned

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany
*Corresponding author.
ckreutz@acm.org (C. K. Kreutz)
https://kreutzch.github.io (C. K. Kreutz)
0000-0002-5075-7699 (C. K. Kreutz)

models, examining whether lightweight specialized models could achieve competitive performance. In their prompts they define the role of the LLM clearly and clearly list the guidelines for simplification to follow. Team THM [9] first marked complex scientific terms via keyphrase identification before instructing small GPT or Gemini models in a zero-shot setting with defined personas to simplify only these complex terms. The UBOnlp team [10] used GPT-4o in a zero-shot setting. They observed that sentence-level simplification often relied on information deletion, while document-level simplification benefited from sentence splitting and the addition of explanations. Team UoA [11] compared a fine-tuned BART model with a zero-shot Llama-based approach guided by a jargon-aware prompt. The LIS team [12] used Mistral 7B supported by a Retrieval-Augmented Generation (RAG) component. Their approach incorporated simplified medical definitions from the MedSimplify Glossary to support the simplification of complex medical terminology. The DS@GT team [13] used few-shot prompting. Their approach introduced an intermediate planning step to identify terms and sentence structures requiring simplification before generating the final output, and employed document summarization to support document-level rewriting. The AIIR-Lab team [14] used fine-tuned versions of Gwen, Llama and Mistral. The SimpLIA team [15] experimented with multiple open source models (Llama, Gemma and Mistral) and prompt formulations such as few-shot in a multi-stage pipeline including keyword extraction, summarization, and simplification. Team DUTH [16] used FLAN-T5-Large and BART-SAMSum.

Following several teams [7, 8, 9], we clearly define a persona in our prompts and give detailed instruction on what to consider in the simplification. Following some approaches from last year's SimpleText lab [7, 9, 10] we also issue zero-shot prompts. Contrasting some previous approaches [7, 8, 10], we do not use closed source LLMs but rather follow the idea of trying to use cheap ones [9] or minimal resources. In our case this means using the open source Llama model similar to SimpLIA [15] or LIS [12] instead of (smaller) closed source ones. Contrasting some approaches [8, 11, 14], we do not fine-tune our models. We also do not incorporate steps supporting the LLM with identifying [9] or resolving [12] complex keyphrases.

3. Dataset

The test datasets provided by the SimpleText lab organizers [1, 2, 3] consist of texts from biomedical abstracts and Cochrane systematic reviews (`Cochrane-auto 2026`). For the sentence-level simplification (Task 1.1), the test dataset contains 48,800 texts and for the document-level simplification (Task 1.2), the test dataset contains 8,069 texts. In the sentence-level dataset 11,765 duplicated (and 37,044 unique) entries are contained. The document-level dataset contains 6,923 unique entries and 1,146 duplicates. Each entry consists of a unique ID and a complex scientific text to simplify. The texts are characterized by biomedical jargon, complex sentence structures, and a high density of factual information. In the sentence-level dataset, there is no context provided to embed the short snippets.

4. Method

Our approach relies on simple zero-shot prompting.

4.1. Task 1.1 - Simplification of Sentences

Table 1 contains the prompts for the simplification of sentences (Task 1.1). In general, `P1` focuses on preserving meaning, medical facts and data points, `P2` targets a high school-level reader and `P3` emphasizes lexical and syntactic simplification by replacing technical terminology with more accessible language.

All three prompts contain the following components in different formulations: The definition of a persona, preserving meaning, simplifications should be in English and an instruction to output only the simplifications, no fluff text.

Prompt ID	Prompt text
P1	You are an expert at the SimpleText Lab. Simplify the given complex sentence for a lay audience. Do not change the original meaning, medical facts, or data points. Respond ONLY with the simplified sentence in English, without any introductory or concluding remarks.
P2	Act as an expert text simplifier for SimpleText Task 1.1. Adapt the following scientific sentence so it can be easily understood by a non-expert general reader (high school level). Keep all essential data, comparisons, and metrics unchanged. Replace difficult scientific and medical terms with simpler alternatives whenever possible. Do not hallucinate, omit, or add information. Return ONLY the simplified English sentence.
P3	You are a scientific communicator for the SimpleText Lab. Rewrite the provided sentence by replacing complex academic or technical jargon with simpler terms and breaking down heavy, compound syntactic structures. Maintain strict factual accuracy. Output ONLY the final simplified English sentence.

Table 1
Prompts for simplification of sentences.

Prompt ID	Prompt text
P4	You are working on scientific text simplification for the SimpleText Lab. Rewrite the following scientific paragraph into simple and clear English for people without scientific knowledge while keeping the original meaning. Replace difficult medical and scientific words with easier words. Keep all important results, numbers, comparisons, and facts. Preserve the logical order and structure of the paragraph. Split long and difficult sentences when needed to improve readability. Make the text easier to read without changing the scientific meaning. Do not add, remove, exaggerate, or invent information. Keep the text coherent and scientifically accurate. Return ONLY the simplified paragraph.
P5	Act as a technical editor for SimpleText. Condense and simplify the following complex academic paragraph. Remove redundant jargon, abstract phrasings, and unnecessarily dense sentence constructions while retaining all key findings, scientific claims, and supporting evidence. Preserve all numerical values, comparisons, and conclusions exactly as presented in the original text. The output must be a cohesive, accurate, and easy-to-read English paragraph. Output ONLY the simplified paragraph.
P6	You are an expert at SimpleText Lab Task 1.2. Rewrite the entire scientific paragraph for a lay audience as a coherent and easy-to-read text. Focus on the paragraph as a whole rather than on individual sentences. Improve the overall flow of information, strengthen connections between ideas, and organize the content in a logical order that is easy for non-experts to follow. Present the main findings clearly, then support them with the most relevant information. Use plain and accessible language throughout the paragraph while preserving all important conclusions and factual information. Remove or simplify technical details that are not essential for understanding the main message, but do not omit key findings or introduce new information. Ensure that the final text reads like a naturally written explanation for the general public. Write the output in English. Respond ONLY with the simplified paragraph.

Table 2
Prompts for simplification of documents.

There are slight differences in the contained components of prompts. P1 is the only prompt to not indicate a replacement of complex jargon. P2 contains an explicit instruction not to add or omit information. P3 is the only prompt to not explicitly define a target audience for the simplification but instructs to break down complex sentence structures.

4.2. Task 1.2 - Simpliciation of Documents

Table 2 contains the prompts for the simplification of documents (Task 1.2). In general, P4 focused on improving readability by replacing complex terms while preserving the original structure, P5 aimed to reduce redundancy and dense academic phrasing, P6 intended to highlight the target audience.

All three prompts contain the following components in different formulations: The definition of a

persona, preserving meaning, an explicit instruction not to add or omit information, the instruction to produce coherent simplifications, simplifications should be in English and an instruction to output only the simplifications, no fluff text.

There are slight differences between prompts beyond wording. P4 is the only prompt to explicitly instruct to not to modify the logical order and structure of text but to split sentences. This prompt also explicitly instructs to replace difficult terms. P5 does not mention the logical order explicitly and P6 instructs to improve the flow and logic of content. P5 is the only prompt to not explicitly define the target audience for the simplification. P6 contains a phrase on handling the complete document at once.

4.3. Implementation

Our approach was implemented in Python via Google Colab using the Hugging Face `transformers` library. As LLM we used the open source `meta-llama/Meta-Llama-3-8B-Instruct` model. Given the computational constraints of deploying an 8-billion parameter model on standard cloud infrastructure, we applied 4-bit quantization using the `bitsandbytes` library. The model weights were loaded using the Normal Float 4 (`nf4`) data type with double quantization, optimizing VRAM consumption while preserving mathematical precision through a `torch.bfloat16` compute data type.

To maximize GPU utility in the simplification step, we implemented a parallel batch processing mechanism with a `BATCH_SIZE` of 20. Left-padding was enforced on the tokenizer to ensure alignment during batch generation.

To prevent runtime crashes or disconnects, we monitor progress by hashing unique structural keys (`pair_id_para_id_sent_id`) and performing real-time incremental saving to Google Drive after each batch. GPU memory leakage was prevented by enforcing explicit garbage collection (`gc.collect()`) and clearing the PyTorch CUDA cache (`torch.cuda.empty_cache()`) at the end of each iteration.

The source code is available at GitHub[1].

5. Submitted Runs and Results

We submitted six runs, three runs for Task 1.1 (sentence simplification) and three runs for Task 1.2 (document simplification). Our Runs are named `cln88_Task1.2_P4` for example with the leftmost component giving our team name `cln88`, the middle component indicating the task and the rightmost component indicating the used prompt by its ID. We only use the prompt IDs to indicate runs in the following.

Table 3 contains the automatic evaluation metrics provided by CodaBench assessing various dimensions of text simplification. Specifically, **SARI** [17] measures the goodness of words added, deleted, and kept against references. **BLEU** [18] evaluates n-gram precision and fluency. The **FKGL** (Flesch-Kincaid Grade Level) indicates the readability of a text on a grade-level (lower scores mean easier text). **Compr.** evaluates the compression ratio, **Sent. Split** indicate the ratio at which sentences are split. **Levenshtein** distance computes the edit distance between the complex and simplified text. **Copies** indicated if there are exact copies. Proportions of **Additions** and **Deletions** are contained. **Lex. Complex.** indicates the lexical complexity of the simplification.

The run achieving the highest SARI score, the main evaluation measure in this shared task, is using prompt P3 and operates on sentences instead of documents.

For BLEU P3 achieves the best results even though all runs produce very low scores. These low scores are not surprising, as BLEU operates on an n-gram level but LLMs tend to rephrase texts completely, thus breaking up neighbouring n-grams.

Even though P2 targets high school readers, its FKGL is the highest overall while P3's FKGL is the lowest without explicit instruction to adhere a specific age or grade level. In P3 it is explicitly stated that heavy compound structures in sentences should be broken down. This component of breaking down complex sentence structures is also present in P4 which also achieved a lower FKGL score. As

[1] https://github.com/kreutzch/CLNN88-SimpleText-26

Metric	Task 1.1 (Sentence)			Task 1.2 (Document)		
	P1	P2	P3	P4	P5	P6
SARI	44.5	44.41	**44.93**	43.56	**43.8**	42.72
BLEU	6.73	6.52	**7.74**	**4.97**	4.91	3.99
FKGL	13.95	14.53	**10.41**	**10.62**	11.38	14.22
Compr.	1.08	1.18	**0.79**	0.4	0.39	**0.35**
Sent. Split	1.04	1.12	1.02	0.54	0.51	0.38
Levenshtein	0.52	0.51	0.49	0.39	0.39	0.37
Copies	0.0	0.0	0.0	0.0	0.0	0.0
Additions	0.53	0.55	0.41	0.14	0.13	0.09
Deletions	0.46	0.42	0.61	0.75	0.75	0.77
Lex. Compl.	8.38	8.41	**8.26**	**8.44**	8.6	8.78

Table 3
Official CodaBench automatic evaluation results for the three submitted runs across both tasks.

FKGL does not consider complexity of words but rather number of words, sentences and syllables, this observation seems to fit the instructions of the prompts. The number of additions and deletions are comparable for runs from a subtask with the exception of P3. Here again, the same explanation might hold.

The compression rate for the document-level simplification is comparable, indicating a considerable reduction of length of text. On the sentence-level both P1 and P2 increase text length while P3 vastly reduces text size. While P1 and P2 explicitly mention not to change numerical information, the call for meaning preservation is not as strict in P3.

Sentences are split at comparable levels on the sentence-level task while P6 achieves a lower score than P4 and P5. This means fewer sentences are contained in simplifications produced by P6. This prompt contains the instruction to focus on the document as a whole instead of individual sentences which could have contributed to this behaviour.

For Levenshtein and lexical complexity we found comparable scores for the three runs from a subtask. The simplifications do not contain exact copies of the original source texts.

All of our prompts used for one subtask contain roughly the same components of instruction in different formulations with few instructions being present in only one or two of the prompts. We assumed to achieve comparable results across evaluation measures for the three prompts of a subtask but found considerable differences. These differences rather stem from the exact formulations of prompts than their contained components.

6. Conclusion

In this work, we used prompt engineering and the open source LLM Llama 3 (8B) to simplify biomedical sentences (Task 1.1) and documents (Task 1.2). Our three prompts for both sub-tasks contained roughly the same components of instruction in different formulations. Nevertheless, we found considerable differences in achieved scores, highlighting the importance of exact formulation of prompts.

In future work, we plan to further explore the exact formulations of components in prompts as well as their effect on linguistic features of resulting simplifications. Additionally, few-shot prompt engineering could introduce in-context medical examples.

Declaration on Generative AI

During the preparation of this work, the authors used the ChatGPT in order to translate a French version of the manuscript to the English write-up. After using this service, the authors reviewed and

edited the content as needed and take full responsibility for the publication's content.

Acknowledgments

We thank the SimpleText organisers for continuously organising the shared task and bringing together the community to advance the field of automatic text simplification.

References

[1] L. Ermakova, H. Azarbonyad, J. Bakker, G. K. Shahi, B. Vendeville, J. Kamps, CLEF 2026 SimpleText Track - Simplify Scientific Text (and Nothing More), in: ECIR '26, 2026, pp. 215–224. URL: https://doi.org/10.1007/978-3-032-21321-1_31. doi:10.1007/978-3-032-21321-1_31.

[2] L. Ermakova, H. Azarbonyad, J. Bakker, B. Chakar, G. K. Shahi, J. Kamps, Overview of the CLEF 2026 SimpleText track: Simplify scientific texts (and nothing more), in: CLEF '26, 2026.

[3] J. Bakker, L. Ermakova, J. Kamps, Overview of the CLEF 2026 SimpleText Task 1: Simplify Scientific Text, in: CLEF '26, 2026.

[4] L. Ermakova, H. Azarbonyad, J. Bakker, B. Vendeville, J. Kamps, CLEF 2025 SimpleText Track, in: ECIR'25, 2025, pp. 425–433.

[5] J. Bakker, B. Vendeville, L. Ermakova, J. Kamps, Overview of the CLEF 2025 SimpleText Task 1: Simplify Scientific Text, in: CLEF '25, 2025, pp. 4167–4185. URL: https://ceur-ws.org/Vol-4038/paper_344.pdf.

[6] N. Largey, R. Maarefdoust, S. Durgin, B. Mansouri, SimpleText Best of Labs in CLEF-2024: Application of Large Language Models for Scientific Text Simplification, in: CLEF '25, 2025, pp. 128–141. doi:10.1007/978-3-032-04354-2_9.

[7] J. Collado-Montañez, J. A. O. Zambrano, C. Espin-Riofrio, A. Montejo-Ráez, SINAI in SimpleText CLEF 2025: Simplifying Biomedical Scientific Texts and Identifying Hallucinations Using GPT-4.1 and Pattern Detection, in: CLEF Working Notes '25, 2025, pp. 4225–4239. URL: https://ceur-ws.org/Vol-4038/paper_348.pdf.

[8] P. Kocbek, G. Stiglic, UM_FHS at the CLEF 2025 SimpleText Track: Comparing No-Context and Fine-Tune Approaches for GPT-4.1 Models in Sentence and Document-Level Text Simplification (2025) 4301–4310. URL: https://ceur-ws.org/Vol-4038/paper_354.pdf.

[9] N. Hofmann, J. Dauenhauer, N. O. Dietzler, I. D. Idahor, C. K. Kreutz, THM@SimpleText 2025 - Task 1.1: Revisiting Text Simplification based on Complex Terms for Non-Experts, in: CLEF Working Notes '25, 2025, pp. 4276–4285. URL: https://ceur-ws.org/Vol-4038/paper_352.pdf.

[10] B. Vendeville, L. Ermakova, P. D. Loor, J. Kamps, UBOnlp Report at the SimpleText lab of CLEF 2025, in: CLEF Working Notes '25, 2025, pp. 4363–4375. URL: https://ceur-ws.org/Vol-4038/paper_360.pdf.

[11] T. Papandreou, J. Bakker, J. Kamps, University of Amsterdam at the CLEF 2025 SimpleText Track, in: CLEF Working Notes '25, 2025, pp. 4356–4362. URL: https://ceur-ws.org/Vol-4038/paper_359.pdf.

[12] A. A. N. Djoudi, S. Nouali, M. Aabid, I. Badache, A. Chifu, P. Bellot, LIS at SimpleText 2025: Enhancing Scientific Text Accessibility with LLMs and Retrieval-Augmented Generation, in: CLEF Working Notes '25, 2025, pp. 4348–4355. URL: https://ceur-ws.org/Vol-4038/paper_358.pdf.

[13] K. C. Marturi, H. H. Elwazzan, LLM-Guided Planning and Summary-Based Scientific Text Simplification: DS@GT at CLEF 2025 SimpleText, in: CLEF Working Notes '25, 2025, pp. 4338–4347. URL: https://ceur-ws.org/Vol-4038/paper_357.pdf.

[14] N. Largey, D. Wu, B. Mansouri, AIIRLab Systems for CLEF 2025 SimpleText: Cross-Encoders to Avoid Spurious Generation, in: CLEF Working Notes '25, 2025, pp. 4311–4323. URL: https://ceur-ws.org/Vol-4038/paper_355.pdf.

[15] Y. Gallina, T. Jiménez, S. Huet, University of Avignon at the CLEF 2025 SimpleText Track: Guided Medical Abstract Simplification, in: CLEF Working Notes '25, 2025, pp. 4263–4275. URL: https://ceur-ws.org/Vol-4038/paper_351.pdf.

[16] G. Arampatzis, A. Arampatzis, DUTH at CLEF 2025 SimpleText Track: Tackling Scientific Text Simplification and Hallucination Detection, in: CLEF Working Notes '25, 2025, pp. 4211–4224. URL: https://ceur-ws.org/Vol-4038/paper_347.pdf.
[17] W. Xu, C. Napoles, E. Pavlick, Q. Chen, C. Callison-Burch, Optimizing Statistical Machine Translation for Text Simplification, Transactions of the Association for Computational Linguistics 4 (2016) 401–415. doi:10.1162/tacl_a_00107.
[18] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a Method for Automatic Evaluation of Machine Translation, in: ACL '02, 2002, pp. 311–318. doi:10.3115/1073083.1073135.