SPU_DeepSimplify at the CLEF 2026 SimpleText Track: A Planning-Based Approach to Scientific Text Simplification

Raneem Rabih[1,*], Riad Sonbol[1,2] and Wesam Alsohle[1,2]

[1]Faculty of Artificial Intelligence Engineering, Syrian Private University, Damascus, Syria
[2]Department of Informatics, Higher Institute for Applied Sciences and Technology, Damascus, Syria

Abstract

This paper presents our participation in Task 1 of the SimpleText 2026 Track on scientific text simplification. To address these challenges, we propose a planning-based framework that separates sentence analysis from text generation to improve controllability and meaning preservation. The system combines terminology-aware preprocessing, semantic planning, strategy selection, and controlled generation. A lightweight Llama 3.1 8B model is used for semantic planning and simplification strategy selection, providing efficient structured analysis, while a larger Llama 3.3 70B model is employed for final controlled generation to maximize simplification quality and fluency. The planner determines the most suitable operation, such as rephrasing, splitting, deleting, or keeping the sentence unchanged, before the generator produces the final simplified output. Experimental results show that the proposed approach achieves competitive performance, reaching a SARI score of 45.00. The findings suggest that explicit planning can improve the transparency and effectiveness of scientific text simplification.

Keywords

Scientific text simplification, Semantic planning, Controlled generation, Large language models, Readability

1. Introduction

Scientific and medical text simplification is important for making complex information accessible to non-specialist readers [1, 2]. Scientific and medical texts often contain specialized terminology, long sentence structures, and dense factual or statistical information. These characteristics make them difficult to understand for students, patients, caregivers, and members of the general public [3]. At the same time, simplification in these domains must be handled carefully, because removing uncertainty markers, changing numerical information, or omitting key findings may alter the intended meaning.

This paper describes our participation in Task 1 of the SimpleText 2026 Track, which focuses on sentence-level scientific text simplification [5, 6]. Given a complex scientific or medical sentence, the goal is to generate a simpler version that remains faithful to the source sentence.

Recent work in scientific and medical text simplification has explored several directions. Prompt-based approaches use large language models directly through carefully designed instructions and can produce fluent simplifications without task-specific training. Fine-tuning approaches adapt pretrained models to simplification datasets, but they require additional computational resources and may not always outperform strong prompting methods. Lexical and terminology-supported approaches focus on identifying difficult terms and providing simpler alternatives or short contextual explanations. Plan-driven approaches are especially relevant to our work because they attempt to select or describe the intended simplification operation before generating the final output. These directions show that scientific text simplification requires both language generation ability and control over the transformation applied to the source sentence.

Motivated by these observations, we propose a planning-based simplification pipeline. Instead of applying a single direct prompt to all input sentences, the system separates the simplification process into several stages. First, terminology-aware preprocessing detects scientific and medical terms. These terms are either replaced with simpler alternatives when appropriate or used to construct contextual

CLEF 2026 Working Notes, 21–24 September 2026, Jena, Germany
*Corresponding author.
raneem_rabih@spu.edu.sy (R. Rabih)

guidance for later stages. Then, a semantic planning stage analyzes the sentence and produces a structured plan that describes the simplification needs of the input. In our final configuration, this planning stage uses Llama 3.1 8B as the planner model. Based on the generated plan, a decision layer selects the appropriate operation, such as rephrasing, splitting and simplifying, deleting secondary information, or keeping the sentence unchanged. Finally, a controlled generation stage uses Llama 3.3 70B as the executor model to produce the simplified sentence according to the selected operation and the available terminology context.

The main motivation behind this design is to improve control and interpretability in scientific text simplification. Direct use of large language models can produce fluent outputs, but it may also lead to over-simplification, unnecessary deletion, or meaning shifts. By separating terminology processing, semantic planning, strategy selection, and final generation, the proposed system aims to make the simplification process more traceable and better aligned with the goal of preserving scientific meaning.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 describes the data. Section 4 presents the proposed approach. Section 5 describes selective post-processing. Section 6 presents the results and discussion. Section 7 concludes the paper and outlines future work.

2. Related Work

Scientific text simplification has recently been addressed using several types of methods, including prompt-based large language models, supervised fine-tuning, plan-driven simplification, and terminology-supported approaches. These directions are reflected in recent SimpleText shared-task systems and overview papers [6, 7].

A common direction in recent SimpleText systems is the use of zero-shot or few-shot prompting with large language models. These approaches rely on the model's existing ability to rewrite complex sentences without additional task-specific training. Previous systems used models such as GPT, LLaMA, Mistral, and Gemma with prompts designed to preserve meaning, reduce complexity, or explain difficult terms [8, 9, 10, 11]. Such systems showed that strong language models can produce competitive simplifications, especially when the prompt includes clear constraints and simplification guidelines. However, direct prompting may not always provide enough control over the type of transformation applied to each sentence.

Other systems used fine-tuning or supervised learning to adapt pretrained models to scientific or medical simplification data. Fine-tuning can improve task specialization because the model learns from simplification examples. However, this approach requires training data, computational resources, and careful tuning. Previous SimpleText systems explored fine-tuned or instruction-tuned models such as GPT-2, BioBART, T5, and GPT-4.1 variants [12, 13, 14], and achieved competitive results, but they did not consistently outperform strong prompt-based methods. This suggests that system design and prompting constraints can be as important as model adaptation.

Plan-driven and structural simplification methods are especially relevant to our work. Instead of rewriting all sentences with the same instruction, these methods first identify the intended simplification operation, such as deletion, splitting, or rephrasing. This makes the generation process more explicit and easier to control. Prior plan-guided systems showed that selecting a simplification strategy before generation can improve interpretability and help the model apply more suitable transformations to different sentence types [15, 16, 17].

Terminology-supported simplification is another important direction in scientific and medical domains. Technical terms are one of the main barriers for non-specialist readers. Some previous systems used complex word identification, lexical simplification, or external glossaries such as MedSimplify to guide the model toward clearer terminology [18, 19, 20]. These approaches show that terminology support can improve simplification when it is used carefully. However, external knowledge must be controlled so that the system does not add unsupported information.

Overall, previous work shows that scientific text simplification requires more than fluent rewriting.

Effective systems need to preserve meaning, handle technical terminology, and choose the appropriate type of simplification for each sentence. Our approach builds on these observations by combining terminology-aware preprocessing, semantic planning, and controlled generation. Unlike direct prompting methods, the proposed pipeline separates sentence analysis from final generation, allowing the system to select a suitable simplification operation before producing the final output.

3. Data Description

The official test data used in this work were provided as part of Task 1 of the SimpleText 2026 Track. The task focuses on sentence-level scientific text simplification, where each input item is a complex scientific or medical sentence that must be rewritten into a simpler version while preserving its original meaning. The official input file contains 48,809 sentence-level instances.

The data were provided in JSON format. Each instance includes metadata fields that identify the source document and the sentence position within the text. The main fields are `pair_id`, `source`, `language`, `para_id`, `sent_id`, and `complex`. The `pair_id` identifies the corresponding document pair, while `para_id` and `sent_id` identify the paragraph and sentence position. The `source` field indicates the data source, and the `language` field specifies the language of the sentence. The `complex` field contains the original complex sentence that the system must simplify.

The official file contains multilingual data. English represents the largest subset with 20,259 instances, followed by Spanish, French, Persian, Korean, Chinese, Portuguese, Thai, German, Japanese, and Hindi. The data also come from several sources, including Cochrane full 2026, Cochrane-auto 2025, Cochrane-auto 2026, Cochrane-auto validation and test sets, Medline, and SimpleText2024. The largest source is Cochrane full 2026, with 35,970 instances.

To better understand the input data, we computed descriptive statistics over the complex source sentences. Sentence length was measured using whitespace-based word counts. The average sentence length was 23.01 words, with a median of 19 words and a standard deviation of 21.50. The shortest sentence contained 1 word, while the longest sentence contained 450 words. The first and third quartiles were 10 and 31 words, respectively, showing that most sentences are relatively short or medium length, while a smaller portion consists of very long sentences.

Figure 1 shows the distribution of source sentence lengths. The distribution is right-skewed: most sentences fall within the 11–50 word range, while fewer sentences exceed 100 words. This observation is relevant to the proposed system because sentence length and structural complexity influence the selected simplification strategy. Short or already clear sentences may be kept unchanged, while longer or denser sentences may require rephrasing or splitting and simplification. Therefore, the descriptive analysis supports the design choice of using a strategy selection layer rather than applying the same rewriting operation to all inputs.

In this work, the system mainly uses the `complex` field as the input sentence for simplification. The `pair_id`, `para_id`, and `sent_id` fields are preserved in the final submission to keep the required alignment with the official data. In addition, paragraph and sentence identifiers are used when available to retrieve the true next sentence for semantic planning, rather than assuming that the next row in the file always corresponds to the next sentence in the original paragraph.

An example of one input item is shown below:


```
{
  "pair_id": "CD015746",
  "source": "Cochrane-auto 2026",
  "language": "en",
  "para_id": 0,
  "sent_id": 0,
  "complex": "We included 19 trials (17 RCTs and two cluster-RCTs)."
}
```

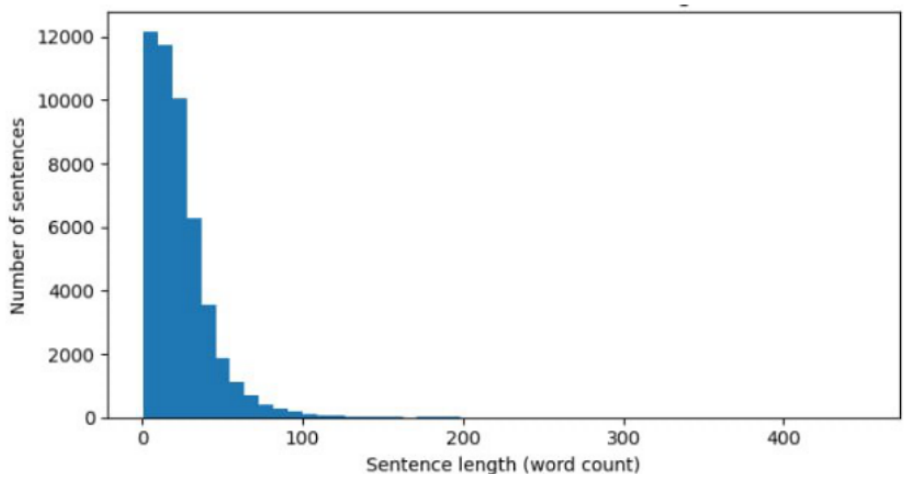

Figure 1: Distribution of source sentence lengths measured by word count.

Table 1
Descriptive statistics of source sentence lengths.

Statistic	Value
Total records	48,809
Minimum length	1
Maximum length	450
Mean length	23.01
Median length	19.00
Standard deviation	21.50
25th percentile	10.00
75th percentile	31.00

Table 1 summarizes the main descriptive statistics of the official input data based on source sentence length. The values show that most sentences are short to medium length, while a smaller number of instances are considerably longer, which supports the need for different simplification strategies.

4. Approach

The proposed system is designed as a controlled pipeline for sentence-level scientific text simplification. As shown in Figure 2, the system does not rely on a single direct rewriting prompt. Instead, it processes each input sentence through a sequence of stages: terminology support, semantic planning, JSON plan generation, strategy selection, execution, and final output generation. This design is motivated by the fact that scientific and medical sentences can be difficult for different reasons. Some sentences contain specialized terminology, others require rephrasing or splitting, and some are already simple enough to be kept unchanged.

The system receives a complex scientific or medical sentence as input and produces a simplified sentence as output. First, the terminology support component identifies relevant scientific or medical terms that may require clarification. Then, the semantic planning stage analyzes the sentence and produces a structured JSON plan describing its simplification needs. Based on this plan, the strategy selection layer chooses one of four operations: rephrase, split and simplify, delete, or keep as is/ignore. The selected operation is then passed to the execution stage, which generates the final simplified output under constraints intended to preserve the original scientific meaning.

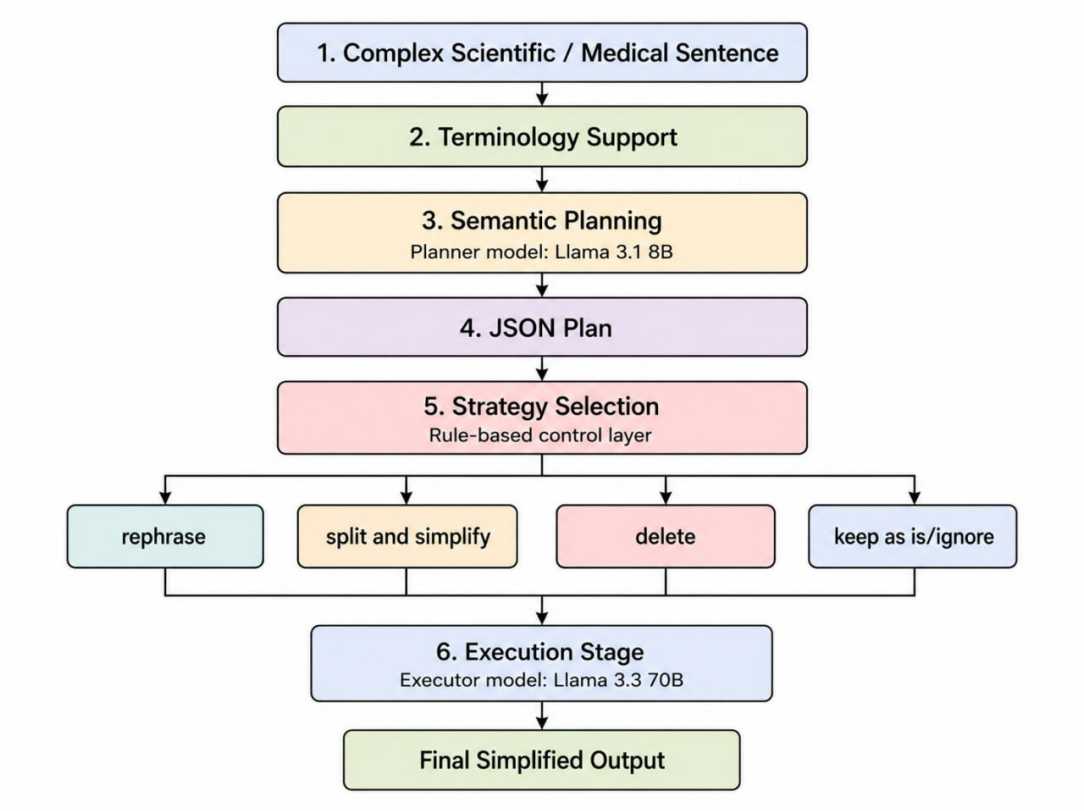

Figure 2: Overview of the proposed system simplification pipeline.

4.1. Model Selection and Inference Setting

The system uses pretrained instruction-tuned large language models as its main reasoning and generation components. The models are used in inference mode without fine-tuning or updating their weights. During execution, the system sends controlled prompts to the selected models and receives generated outputs, while no gradient computation is performed. Therefore, the model parameters remain unchanged throughout the project.

In our implementation, the models were accessed through the OpenRouter API. OpenRouter was used only as an inference provider for sending prompts and receiving model outputs. It was not used for training, fine-tuning, or modifying model weights. This setup allowed the system to run the planning and generation stages through API-based inference while keeping the model behavior controlled through prompt design and rule-based decision layers.

Two different models are used in two different roles. The first model, Llama 3.1 8B, is used for semantic planning. In this stage, the model analyzes the input sentence and returns a structured JSON plan that describes simplification-related properties, such as terminology, structural complexity, statistical details, and whether the sentence is already simple. The second model, Llama 3.3 70B, is used for controlled generation. In this stage, the model produces the final simplified sentence according to the strategy selected by the control layer.

Using two specialized models allows the system to balance efficiency and generation quality. The smaller model is suitable for structured analysis and planning, while the larger model provides stronger language generation ability for the final simplification stage. As a result, the system does not treat the language model as a single black-box simplifier. Instead, the models are integrated into a broader controlled pipeline that separates analysis from execution, constrains model behavior, and makes intermediate decisions more interpretable and traceable.

4.2. Terminology Support Component

Scientific and medical terminology is a major source of complexity in the target texts. A general-purpose language model may simplify such terms inconsistently or may choose an unsafe replacement if it is not guided carefully. For this reason, the proposed system includes a terminology-support component based on the MedSimplify glossary. MedSimplify provides biomedical terms with corresponding definitions or simpler explanations, which makes it useful for supporting lexical simplification in scientific and medical sentences.

This component provides glossary-based hints for difficult biomedical terms. It is not a full retrieval-augmented generation system, since it does not retrieve long passages or external documents. Its role is narrower and more controlled: it identifies relevant MedSimplify terms in the input sentence and provides short definitions or safe plain-language equivalents to guide the model during terminology substitution and simplification. The glossary context is used only for terms that already appear in the source sentence, so it supports faithful simplification without introducing new external information.

The terminology-support component is also connected to the prompt design. When one or more medical terms are detected in the input sentence, the system builds a short terminology context and passes it to the generation prompt. This context contains only terms that appear in the source sentence and their corresponding plain-language hints. For example, a detected term such as "hypertension" may be provided to the model as "hypertension: high blood pressure." The prompt instructs the model to use these hints only when they preserve the original meaning and to avoid adding unsupported medical information.

Example of a terminology-aware generation prompt:

```
Terminology context:
- hypertension: high blood pressure
- tachycardia: rapid heart rate

Requirements:
- Preserve the original meaning.
- Replace difficult medical terms with simpler alternatives when appropriate.
- Do not add unsupported facts.
- Output only the simplified sentence.

Sentence: Patients with hypertension and tachycardia were included in the trial.
```

4.3. Semantic Planning Stage

The semantic planning stage is responsible for analyzing the input sentence before the final simplification is generated. Instead of asking the language model to simplify the sentence directly, this stage asks the planner model to inspect the sentence and describe its simplification-related properties in a structured form. In the final system configuration, Llama 3.1 8B is used as the planner model.

The planner receives the source sentence, the terminology context when available, and the next sentence when paragraph-level context can be retrieved. It then returns a structured JSON plan that indicates whether the sentence is already simple, whether it contains technical terms, whether it is structurally complex, whether it contains secondary statistical details, whether abbreviation expansion may be required, whether it depends on the next sentence, and whether it is redundant given the next sentence.

Example of a planner prompt:

```
You are a scientific text simplification planner.
Analyze the following scientific sentence.
Do not simplify it directly.
Return valid JSON only.

Your analysis must indicate:
- whether the sentence is already simple;
- whether it contains technical or scientific terms;
- whether it is structurally complex;
- whether it contains secondary statistical details;
- whether abbreviation expansion may be required;
- whether the sentence depends on the next sentence for interpretation;
- whether the sentence is redundant given the next sentence;
- a short reason explaining the planning decision.

Sentence:
{sentence}

Next sentence, if available:
{next_sentence}
```

This planning stage makes the system more interpretable because the simplification decision is not hidden inside the final generation step. The generated plan is used by the strategy selection layer to decide whether the sentence should be rephrased, split and simplified, deleted, or kept unchanged. If the planner output is invalid or incomplete, the system falls back to a safe default plan so that the pipeline can continue without stopping.

4.4. Use of Next-Sentence Context

Some scientific sentences are difficult to interpret in isolation. They may refer to information in the following sentence, or they may be redundant with nearby text. To support this analysis, the planner can receive the true next sentence when paragraph and sentence identifiers are available.

The system does not assume that the next row in a sampled dataset is the true next sentence. Instead, it constructs a lookup table using paragraph and sentence identifiers. This ensures that the next-sentence context is retrieved from the original paragraph structure rather than from the sampled row order.

The next sentence is used only as planning context. It helps the planner detect dependency or redundancy, but the final simplification remains based on the original input sentence and the selected strategy.

4.5. Strategy Selection Layer

The strategy selection layer converts the planner output and sentence-level signals into a concrete simplification decision. This layer combines semantic information from the planner with surface-level features such as word count, comma count, clause markers, numerical patterns, evidence wording, and detected terminology.

The system selects one of four main strategies. The role of this layer is to prevent the system from applying the same transformation to all sentences. Scientific sentences differ in the type of complexity they contain. A short sentence with a difficult term may need rephrasing, while a long multi-clause sentence may benefit from splitting. A sentence containing secondary statistical details may require compression, but only if it does not contain essential scientific information.

The deletion decision is intentionally conservative. A sentence is selected for omission only when it appears to contain secondary information and does not contain important context such as trials, studies, participants, follow-up, outcomes, effects, evidence, or comparisons. This restriction is necessary because excessive deletion can damage meaning preservation.

The strategy selector is rule-based, but it functions as a control layer around the language model rather than as a replacement for it. The LLM provides semantic planning signals, while the rule-based layer makes the final decision explicit, traceable, and easier to evaluate.

4.6. Execution Stage and Controlled Generation

The execution stage is responsible for producing the final simplified output. It receives the original sentence, the selected strategy, and the terminology context when available. This stage is designed to apply terminology handling before the final sentence-level transformation. This ordering helps the system clarify difficult terms before applying rephrasing, splitting, or omission.

If a valid terminology context is available, the system first performs an LLM-based terminology substitution step. This step attempts to replace difficult scientific or medical terms with safer plain-language alternatives. If the substitution changes the sentence, the substituted version becomes the working sentence for the final simplification strategy. If no valid terminology context is available, or if substitution does not improve the sentence, the original sentence remains the working sentence.

After this step, the selected strategy is applied to the working sentence. The execution prompts are strategy-specific and include explicit constraints to preserve the original scientific meaning, avoid unsupported information, keep important numerical and medical details, and preserve uncertainty markers. This is important because changing words such as "may," "probably," or "low-certainty evidence" can make the scientific claim stronger than intended.

Example of a strategy-specific executor prompt:

```
You are an expert scientific text simplifier.
The selected strategy is REPHRASE.
Rewrite the following sentence using simpler and clearer language.

Requirements:
- Preserve the original scientific meaning.
- Do not add new information.
- Do not remove important medical facts.
- Preserve uncertainty words such as may, probably, suggests, and evidence.
- Output only the simplified sentence.
```

For the rephrase strategy, the executor rewrites the sentence using simpler and clearer wording without making major structural changes. This strategy is used for moderately complex sentences or terminology-heavy sentences that do not require splitting or omission.

For the split_and_simplify strategy, the executor may divide a long or structurally complex sentence into one or two shorter sentences when this improves readability. The prompt does not force splitting in every case. Instead, it allows splitting only when it makes the sentence clearer.

For the delete strategy, the executor is instructed to remove only unnecessary or secondary information. This strategy is used conservatively. If the sentence contains important scientific content related to studies, participants, outcomes, evidence, effects, or comparisons, the model is instructed to produce a brief simplified version instead of omitting the content entirely.

For the keep_as_is strategy, the output is preserved when the sentence is already sufficiently simple or when modification may be unsafe. If terminology substitution has already produced a safer working sentence, the system may return the substituted version. Otherwise, the original sentence is kept unchanged.

4.7. Output Validation

The system also uses output-control instructions to reduce invalid or verbose generations. The executor is instructed to return only the simplified sentence and not to include explanations, introductions, or comments. This is important because some short inputs, headings, or incomplete fragments may cause the model to produce meta-responses instead of a valid simplification. If an invalid or empty output is still produced, fallback rules are applied after generation.

```
Output only the simplified sentence.
Do not explain your answer.
Do not include introductions or comments.
If simplification is unsafe, keep the sentence unchanged.
```

4.8. Prompt Engineering

Prompt engineering is used across the proposed pipeline because the models are used without fine-tuning. Instead of using one direct prompt for all sentences, the system uses stage-specific prompts for planning, terminology-aware generation, strategy-controlled execution, and output validation. This design allows the system to separate sentence analysis from final generation and to apply different constraints depending on the selected operation.

5. Post-Processing

After obtaining the outputs of the main pipeline, we applied an additional selective post-processing stage. This stage was applied to the predictions generated by the previous system, not directly to the original input data. It was not applied to all outputs. Instead, it targeted the weakest outputs of the main pipeline, especially predictions that were identical or nearly identical to the original complex sentence after basic text normalization. This normalization included lowercasing, removing extra spaces, and standardizing whitespace.

These cases were selected because they indicate that the main pipeline did not perform an effective simplification. In the official run, this selection produced approximately 3,661 candidate sentences. Applying post-processing only to these cases reduced the risk of modifying outputs that had already been simplified successfully, while focusing the additional generation step on sentences that still needed improvement.

The post-processing prompt asked the model to rewrite the selected predictions for a general reader with approximately Grade 8 reading ability. It emphasized preserving scientific and medical facts, keeping key entities and outcomes, replacing difficult terms with simpler wording when possible, removing redundancy, splitting long or nested sentences, and avoiding unsupported additions. After generation, safety checks were applied to reject invalid candidates, including empty outputs, prompt leakage, outputs that were too short or too long, and candidates that still copied the source sentence. If a post-processed output failed these checks, the previous prediction was retained.

6. Results and Discussion

Table 2 presents the evaluation results of the proposed system variants. We report the results of the main pipeline and the final system after applying selective post-processing. Since the post-processing step was applied only to a limited set of outputs, the differences between the two variants are small. However, the final system achieved the highest SARI score, increasing from 44.91 to 45.00.

For the main pipeline, the system achieved a SARI score of 44.91, indicating strong performance in the main simplification operations: keeping important content, deleting unnecessary information, and adding appropriate wording. After selective post-processing, SARI increased slightly to 45.00. BLEU decreased from 11.61 to 11.46, which suggests that some final outputs moved farther from the reference wording. This is not necessarily negative, since valid simplifications may use wording different from the reference. Therefore, BLEU is interpreted as a secondary similarity indicator rather than the main simplification measure.

The readability and compression metrics changed only slightly. FKGL decreased from 10.22 to 10.19, suggesting a small improvement in readability, while the compression ratio increased from 0.73 to 0.74, meaning that the final outputs became slightly longer but remained shorter than the original sentences.

Table 2
Results of the proposed system.

Method	SARI	BLEU	FKGL	Compr.	Split	Lev.	Copy	Add.	Del.	Lex.
Main system	44.91	11.61	10.22	0.73	1.10	0.58	0.00	0.24	0.53	8.58
Final system	45.00	11.46	10.19	0.74	1.10	0.57	0.00	0.25	0.54	8.57

The sentence split score remained stable at 1.10. Levenshtein similarity decreased from 0.58 to 0.57, indicating slightly more rewriting after post-processing. Exact copies remained 0.00 in both variants, showing that the system did not simply copy the input sentences. Additions and deletions changed from 0.24 and 0.53 to 0.25 and 0.54, respectively, which indicates a small increase in rewriting. Lexical complexity decreased slightly from 8.58 to 8.57.

7. Conclusion

This paper described our participation in Task 1 of the SimpleText 2026 Track, focusing on sentence-level scientific text simplification. We proposed a planning-based pipeline that separates the simplification process into terminology support, semantic planning, strategy selection, controlled generation, and selective post-processing. The system uses Llama 3.1 8B as a planner to produce structured JSON-based sentence analysis, and Llama 3.3 70B as an executor to generate the final simplified output according to the selected strategy. This design aims to make the simplification process more controlled and interpretable than a single direct rewriting prompt.

The experimental results show that the proposed system achieved strong performance, with the final system reaching a SARI score of 45.00. The results also indicate that selective post-processing produced a small improvement over the main pipeline while keeping the overall system behavior stable. Future work could explore stronger terminology resources, more detailed factual consistency checks, and adaptive strategy selection methods. In addition, further evaluation could investigate how different prompt designs affect readability, meaning preservation, and hallucination risk in scientific and medical text simplification.

Acknowledgments

The authors would like to thank the organizers of the SimpleText 2026 Track for providing the task, data, evaluation platform, and detailed submission guidelines. No external funding was received for this work.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT for language editing, grammar correction, text organization, and improving the clarity of selected sections of the manuscript. The authors also used an AI image generation tool to create illustrative pipeline figures. After using these tools, the authors reviewed and edited the generated content as needed and take full responsibility for the publication's content.

References

[1] F. Alva-Manchego, L. Martin, C. Scarton, and L. Specia. Automated Text Simplification: A Survey. *ACM Computing Surveys*, 53(6).

[2] R. Gotlieb, C. Praska, S. Hendrickson, et al. Accuracy in Patient Understanding of Common Medical Phrases. *JAMA Network Open*, 5(11):e2242972, 2022.

[3] H. Van, D. Kauchak, and G. Leroy. AutoMeTS: The Autocomplete for Medical Text Simplification. In *Proceedings of COLING 2020*, pages 1424–1434.
[4] M. Maddela, F. Alva-Manchego, and W. Xu. Controllable Text Simplification with Explicit Paraphrasing. In *Proceedings of NAACL-HLT*, 2021.
[5] L. Ermakova et al. Overview of CLEF 2026 SimpleText Track: Simplify Scientific Texts (and Nothing More). In M. Hagen et al. (Eds.), *Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of CLEF 2026*. LNCS. Springer-Verlag, 2026.
[6] J. Bakker et al. Overview of the CLEF 2026 SimpleText Task 1: Simplify Scientific Text. In E. Sanchez Salido et al. (Eds.), *Working Notes of CLEF 2026*. CEUR Workshop Proceedings, CEUR-WS.org, 2026.
[7] J. Bakker, B. Vendeville, L. Ermakova, and J. Kamps. Overview of the CLEF 2025 SimpleText Task 1: Simplify Scientific Text. In *CLEF 2025 Working Notes*. CEUR Workshop Proceedings, CEUR-WS.org.
[8] J. Collado-Montanez, J. A. Ortiz-Zambrano, C. Espin-Riofrio, and A. Montejo-Raez. SINAI in SimpleText CLEF 2025: Simplifying Biomedical Scientific Texts and Identifying Hallucinations Using GPT-4.1 and Pattern Detection. In *CLEF 2025 Working Notes*. CEUR Workshop Proceedings, CEUR-WS.org.
[9] Y. Gallina, T. Jimenez, and S. Huet. University of Avignon at the CLEF 2025 SimpleText Track: Guided Medical Abstract Simplification. In *CLEF 2025 Working Notes*. CEUR Workshop Proceedings, CEUR-WS.org, 2025.
[10] B. Vendeville, L. Ermakova, P. De Loor, and J. Kamps. UBOnlp Report at the SimpleText Lab of CLEF 2025. In *CLEF 2025 Working Notes*. CEUR Workshop Proceedings, CEUR-WS.org.
[11] M. M. Aguero-Torales, C. Rodriguez Abellan, and C. A. Castano Moraga. Sentence-level Scientific Text Simplification With Just a Pinch of Data. In *CLEF 2025 Working Notes*. CEUR Workshop Proceedings, CEUR-WS.org.
[12] A. A. Dongre, A. Vaadiraaju, and A. K. Madasamy. NITK SCaLAR Lab at the CLEF 2025 SimpleText Track: Transformer-Based Models for Biomedical Sentence Simplification. In *CLEF 2025 Working Notes*. CEUR Workshop Proceedings, CEUR-WS.org.
[13] P. Kocbek and G. Stiglic. UM_FHS at the CLEF 2025 SimpleText Track: Comparing No-Context and Fine-Tune Approaches for GPT-4.1 Models. In *CLEF 2025 Working Notes*. CEUR Workshop Proceedings, CEUR-WS.org, 2025.
[14] G. Arampatzis and A. Arampatzis. DUTH at CLEF 2025 SimpleText Track: Tackling Scientific Text Simplification and Hallucination Detection. In *CLEF 2025 Working Notes*. CEUR Workshop Proceedings, CEUR-WS.org, 2025.
[15] K. C. Marturi and H. H. Elwazzan. LLM-Guided Planning and Summary-Based Scientific Text Simplification: DS@GT at CLEF 2025 SimpleText. In *CLEF 2025 Working Notes*. CEUR Workshop Proceedings, CEUR-WS.org, 2025.
[16] T. Papandreou, J. Bakker, and J. Kamps. University of Amsterdam at the CLEF 2025 SimpleText Track. In *CLEF 2025 Working Notes*. CEUR Workshop Proceedings, CEUR-WS.org, 2025.
[17] A. Vora, T. Chaudhari, S. Hotha, and S. Sonawane. S-3 Pipeline for Biomedical Text Simplification. In *CLEF 2025 Working Notes*. CEUR Workshop Proceedings, CEUR-WS.org, 2025.
[18] A. A. Nait Djoudi, S. Nouali, M. Aabid, I. Badache, A.-G. Chifu, and P. Bellot. LIS at SimpleText 2025: Enhancing Scientific Text Accessibility with LLMs and Retrieval-Augmented Generation. In *CLEF 2025 Working Notes*. CEUR Workshop Proceedings, CEUR-WS.org, 2025.
[19] N. Hofmann, J. Dauenhauer, N. O. Dietzler, I. D. Idahor, and C. K. Kreutz. THM@SimpleText 2025 - Task 1.1: Revisiting Text Simplification based on Complex Terms for Non-Experts. In *CLEF 2025 Working Notes*. CEUR Workshop Proceedings, CEUR-WS.org, 2025.
[20] Anyantd. MedSimplify: Definitions of Biomedical Jargon in Plain Language. https://github.com/Anyantd/MedSimplify/.