Multilingual Scientific Text Simplification Using a Locally Deployed Open-Source LLM: Plan-Driven Sentence Simplification and Summary-Guided Document Simplification with Llama 3.1 8B at the CLEF 2026 SimpleText Track — Task 1

SimpleText at CLEF 2026

Manoj Kumar[2,*], Prabavathy Balasundaram[1] and Deeraj Kumar[2]

[1]Department of Computer Science and Engineering, Sri Sivasubramaniya Nadar College of Engineering, Chennai, India
[2]Department of Computer Science and Engineering (AI & ML), Easwari Engineering College, Chennai, India

Abstract

This paper describes our participation in CLEF 2026 SimpleText Task 1, which requires simplifying biomedical research abstracts at both the sentence level (Task 1.1) and the document level (Task 1.2) across eleven languages. We deploy the Llama 3.1 8B instruction-tuned model locally via Ollama on consumer-grade hardware, requiring no cloud API access and no per-query cost. For Task 1.1 we design a *plan-driven* zero-shot prompt in which the model internally selects the most appropriate simplification strategy—rephrase jargon, split into shorter sentences, drop statistical parentheticals, or keep unchanged—before producing the simplified output. For Task 1.2 we introduce a two-step *summary-guided* pipeline: the model first generates a concise plain-language summary of the document, which is then used as a semantic anchor in a second pass that rewrites the full abstract. Both pipelines are language-aware: the target language is specified explicitly in every prompt, preserving the multilingual character of the task. An output cleanup step strips model-generated preamble text (e.g., "Here's the simplified version:") from all language outputs. Our Task 1.1 submission achieves SARI **44.89**, outperforming the GPT-4.1-mini commercial API baseline (43.34) and Llama 3.3 70B (42.33) with an 8B model running under 4-bit quantisation on 8 GB VRAM—no cloud access, no fine-tuning required. Our Task 1.2 submission achieves SARI **42.62**.

Keywords

Scientific Text Simplification, Large Language Models, Biomedical NLP, Multilingual, Llama 3.1, CLEF 2026, Zero-Shot Prompting

1. Introduction

Scientific literature is increasingly important for informed public discourse, yet it remains inaccessible to most non-experts. Biomedical research abstracts in particular are dense with technical terminology, statistical notation, and domain-specific conventions that require years of training to parse fluently. This accessibility gap matters most when the research concerns topics of direct public relevance such as drug efficacy, disease risk, or treatment outcomes [1].

Automatic text simplification (ATS) addresses this gap by computationally rewriting complex text into a form that is easier to read while preserving the essential meaning [1, 2]. Early ATS systems relied on hand-crafted syntactic rules; later approaches shifted to statistical and neural models [3, 4]. The advent of large language models (LLMs) has substantially raised the ceiling on zero-shot simplification quality [5], enabling effective simplification without task-specific training data.

The CLEF SimpleText lab [6, 7, 8, 9] has provided an annual evaluation framework for scientific text simplification since 2021. The 2026 edition continues to focus on the challenging multilingual

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany
*Corresponding author.
manojkumar.ksamy@gmail.com (M. Kumar); prabavathyb@ssn.edu.in (P. Balasundaram); deerajkumar05238@gmail.com (D. Kumar)

biomedical setting: Task 1 covers eleven languages and requires both sentence-level and document-level simplification with the SARI metric [10] as the primary evaluation criterion.

A practical constraint facing many researchers is the cost and availability of commercial LLM APIs. Our work explores whether a freely available, locally deployable model can match or exceed the quality of commercial alternatives. We use Llama 3.1 8B [11] served via Ollama [12] on a single consumer GPU (8 GB VRAM), demonstrating that high-quality multilingual simplification is achievable without cloud infrastructure.

Our primary contributions are:

1. A *plan-driven* zero-shot prompt (P5) for sentence-level simplification that instructs the model to internally select a simplification strategy before generating output, achieving SARI 44.89 on Task 1.1.
2. A *summary-guided* two-step pipeline for document-level simplification that uses a self-generated summary as a semantic anchor to guide the full rewrite, achieving SARI 42.62 on Task 1.2.
3. A systematic language-aware prompting scheme that maintains multilingual fidelity across all eleven SimpleText languages.
4. An empirical demonstration that structured prompt engineering on a locally deployable 8B model can match or exceed commercial API performance on multilingual biomedical simplification, with full reproducibility on consumer hardware.

The remainder of this paper is organised as follows. Section 2 reviews related work. Section 3 describes the task and data. Section 4 presents our methodology. Section 5 reports results. Section 6 analyses the results, and Section 7 concludes.

2. Related Work

2.1. Text Simplification

Text simplification has been studied at both the lexical and syntactic levels [1]. Early work focused on rule-based approaches that decomposed complex sentences into shorter clauses, substituted rare words with more common synonyms, or removed parenthetical material [2]. Statistical machine translation frameworks were later adapted to treat simplification as a source-to-target rewriting problem [3]. Alva-Manchego et al. [4] provide an extensive survey of data-driven benchmarks and neural simplification systems.

The SARI metric [10] became the standard evaluation criterion: it rewards systems that correctly add new words, keep relevant words, and delete unnecessary words relative to human reference simplifications and the original source sentence.

2.2. LLM-Based Simplification

Kew et al. [5] benchmarked a range of large language models on sentence simplification, finding that GPT-4-class models substantially outperform smaller models in zero-shot settings, but that careful prompting can partially close the quality gap. The DS@GT team at CLEF 2025 [13] introduced an LLM-guided planning approach in which the model first identifies simplification actions and then applies them—conceptually related to our P5 design. Kocbek and Stiglic [14] compared no-context and fine-tuned GPT-4.1 models at CLEF 2025, establishing strong commercial baselines for the multilingual biomedical setting.

2.3. SimpleText at CLEF

The CLEF SimpleText lab [6] has run since 2021, evolving from English-only tasks to a fully multilingual setting by 2025 [8]. The 2026 edition [9] evaluates sentence-level (Task 1.1) and document-level (Task 1.2) simplification separately, alongside hallucination detection and query-focused summarisation tracks.

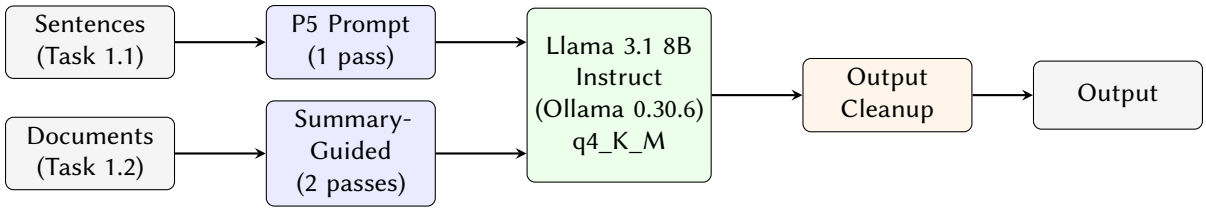

Figure 1: Overall system architecture. Both tasks share the same locally served Llama 3.1 8B model but use different prompt strategies. Output cleanup (preamble stripping) is applied to all language outputs.

3. Task Description and Data

CLEF 2026 SimpleText Task 1 [15] requires participants to simplify biomedical research abstracts. The data consists of abstracts drawn from open-access biomedical literature across eleven languages: English (en), Spanish (es), French (fr), Persian/Farsi (fa), Korean (ko), Simplified Chinese (zh_HANS), Portuguese (pt), Thai (th), German (de), Japanese (ja), and Hindi (hi).

Task 1.1 (sentence level) presents each sentence individually. The input dataset comprises **48,809 sentences**. Systems must output a simplified version of each sentence while preserving its core meaning. The primary evaluation metric is SARI, supplemented by BLEU, Flesch-Kincaid Grade Level (FKGL), compression ratio, sentence split rate, Levenshtein distance, copies, additions, deletions, and lexical complexity.

Task 1.2 (document level) presents each abstract as a whole. The input dataset comprises **8,069 documents**. Systems must output a single simplified version of the entire abstract. Evaluation criteria mirror those of Task 1.1 but applied at the document level. Both input files include a `language` field identifying the language of each item.

4. Methodology

4.1. Overall Architecture

Figure 1 shows the top-level architecture shared by both tasks. A single quantised Llama 3.1 8B model is served locally via Ollama. The two tasks differ in prompt strategy and number of inference passes: Task 1.1 uses a single-pass plan-driven prompt, while Task 1.2 uses two sequential passes. Language-aware prompt construction and an output cleanup step (preamble stripping) are applied in both pipelines.

4.2. Hardware and Software Environment

All inference was conducted on a single consumer laptop with an Intel Core i7-14650HX processor (16 cores, 24 threads), approximately 24 GB RAM, and an NVIDIA GeForce RTX 5050 (laptop) GPU with 8 GB (8,151 MiB) VRAM running CUDA 12.8. All model serving was handled by Ollama 0.30.6 [12], which provides its own inference runtime; the data-processing and API-call scripts were written in Python 3.10.11. No additional deep-learning frameworks were required. No cloud API access was required at any stage.

4.3. Model Selection: Llama 3.1 8B

We evaluated Mistral 7B [16] and Llama 3.1 8B Instruct [11] to select a base model. Both were served through Ollama using 4-bit quantisation. Our Mistral 7B preliminary submissions scored 43.25 SARI on the official Codabench leaderboard, providing an external reference point.

Model selection was made on the basis of blind qualitative review: we inspected 20 randomly sampled outputs per model across English, French, and Korean. A quantitative proxy benchmark—scoring both models against Mistral 7B's own simplified outputs—was also run but is inherently circular (the

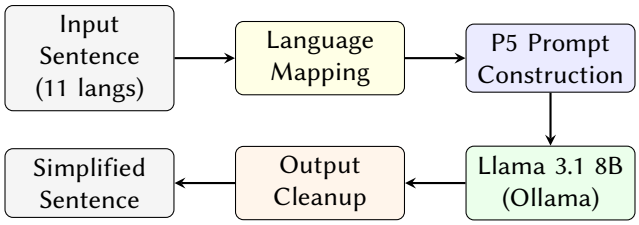

Figure 2: Task 1.1 plan-driven pipeline. The P5 prompt names four strategies (REPHRASE/SPLIT/DROP/KEEP); the model selects and applies one internally without outputting the label.

reference model always scores highest against its own outputs) and was not used to justify the final choice. In the qualitative review, Llama 3.1 8B was consistently preferred: it more reliably substituted medical jargon with plain equivalents, avoided language leakage on non-English inputs, and maintained grammatical coherence in longer sentences. We therefore selected `llama3.1:8b-instruct-q4_K_M` for all final submissions, with inference settings: temperature 0.2, top-p 0.95, top-k 40, seed 42, context window 2,048 tokens.

4.4. Task 1.1: Plan-Driven Sentence Simplification

4.4.1. Design Rationale

Different sentences require different simplification strategies: some benefit from lexical substitution, others from splitting into shorter clauses, others from dropping statistical parentheticals, and some are already simple enough to pass through unchanged. A generic "make this simpler" instruction does not give the model enough structure to make this choice reliably. Our plan-driven approach explicitly names four candidate strategies and instructs the model to evaluate and apply the most appropriate one internally, without labelling the strategy in its output.

4.4.2. Pipeline

Figure 2 illustrates the Task 1.1 pipeline.

4.4.3. P5 Prompt

The P5 system prompt (P5 denotes the fifth design iteration; given in full in Appendix A) lists four candidate strategies—REPHRASE jargon, SPLIT into shorter sentences, DROP statistical parentheticals, KEEP unchanged—and enforces five rules: (1) preserve numbers, study sizes, drug names, and populations; (2) replace medical jargon with plain equivalents in the target language; (3) convert certainty cues to plain language (e.g., *low-certainty evidence* $\rightarrow$ *we are not very sure*); (4) pass already-simple sentences through unchanged; (5) output only the simplified sentence with no preamble or strategy label. Rule 5 is critical: any preamble text in the output would be penalised directly by SARI.

The user message for each sentence is:

```
Simplify this sentence:

{complex_sentence}
```

Of the 48,809 input sentences, 1,797 trivial inputs (fewer than 8 characters, or consisting only of digits and punctuation—e.g., section-numbering artefacts like "1." or "2.") were returned unchanged without calling the model. No input sentence exceeded the 2,048-token context window; the longest sentence in the dataset was approximately 607 tokens. The inference token budget is capped at 512 tokens per sentence for non-trivial inputs.

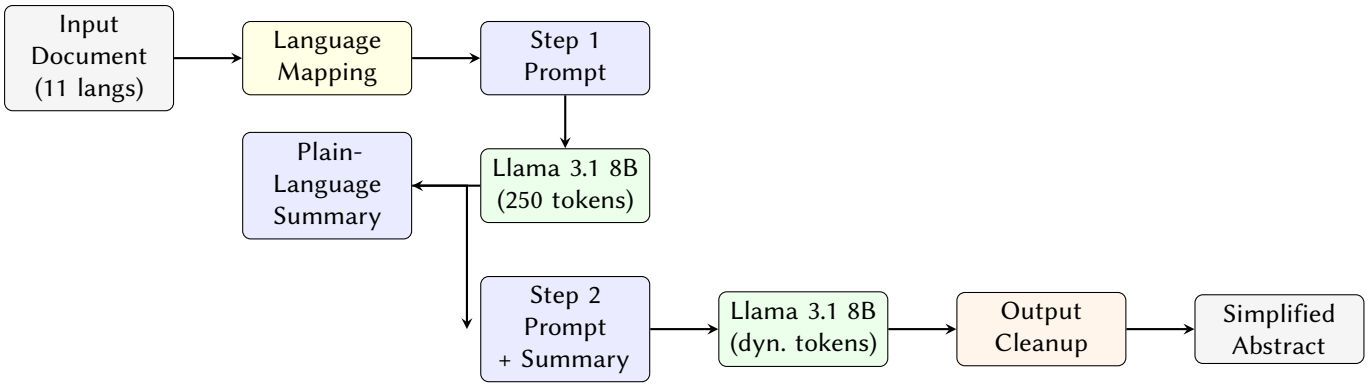

Figure 3: Task 1.2 summary-guided pipeline. Step 1 generates a plain-language summary; Step 2 uses this summary as a semantic anchor to guide the full simplification.

4.5. Task 1.2: Summary-Guided Document Simplification

4.5.1. Design Rationale

Document-level simplification must maintain coherence across multiple sentences and must selectively compress a paragraph-length text. Single-pass LLM inference on a long document suffers from two common failure modes: (1) *hallucination drift*, where the model introduces claims not present in the source, and (2) *premature stopping*, where the output covers only the first few sentences. Our summary-guided approach addresses both by generating an explicit summary in Step 1 that serves as a factual anchor for Step 2.

4.5.2. Pipeline

Figure 3 shows the two-pass pipeline.

4.5.3. Step 1 — Summary Generation

The model is prompted to produce a 3–5 sentence plain-language summary of the key findings of the abstract (system prompt in Appendix A), with a token budget of 250. The language instruction ensures the summary is produced in the same language as the source.

4.5.4. Step 2 — Guided Simplification

The Step 1 summary is prepended to the Step 2 user message as a *summary guide*. The system prompt applies rules consistent with the P5 design: preserve numbers, replace jargon, target Grade 7–8 reading level, output only simplified text. The token budget is set dynamically to $\min(1{,}024,\ \max(384,\ \lfloor 1.2 \times N_{\text{src}} \rfloor))$, where N_{src} is the source word count.

4.6. Language-Aware Processing

Every system prompt includes an explicit directive of the form:

```
OUTPUT LANGUAGE: {lang} (your output MUST be in {lang},
do NOT translate)
```

where `{lang}` is the full language name derived from the `language` field in the input JSON (e.g., “Korean”, “Persian (Farsi)”, “Simplified Chinese”). Without this directive, preliminary experiments showed frequent language leakage on Thai, Korean, and Persian inputs.

4.7. Output Cleanup

LLMs occasionally prefix their output with a preamble phrase such as "Here's the simplified version:" or "Sure!" before producing the actual simplification. These phrases are penalised by SARI because they do not appear in the reference simplifications. We apply a regular-expression cleanup pass to all outputs, across all languages, that strips:

- leading preamble phrases ("here's the simplified version:", "simplified:", "sure,", and variants);
- leading and trailing quotation marks;
- trailing translation-artifact annotations (e.g., "(Translation from French:)").

This cleanup is applied after every model inference call in both tasks.

Note: a 129-entry medical lexicon (e.g., *myocardial infarction* → *heart attack*) was evaluated during development but was disabled in the final submissions because it reduced SARI by 0.21–1.84 points on our English development evaluation; the P5 prompt's own jargon-replacement rules proved more effective.

4.8. Worked Example: End-to-End Processing

This subsection traces real inputs through each pipeline using actual outputs from the final Codabench submissions.

4.8.1. Task 1.1 — Sentence-Level Walkthrough

Input (English):

> *We included 19 trials (17 RCTs and two cluster-RCTs).*

Step 1 — Language mapping. The `language` field is `en`; `{lang}` is set to `English`.
Step 2 — P5 prompt construction and inference. The system prompt is filled and submitted to Llama 3.1 8B with the user message "Simplify this sentence: {input}".
Step 3 — Model output:

> *We included 19 clinical trials: 17 were randomized controlled studies and 2 were cluster-randomized controlled studies.*

Step 4 — Output cleanup. No preamble is detected; the output passes through unchanged.
Observation. The model expanded the abbreviations RCT and cluster-RCT into plain language, making the sentence accessible to readers unfamiliar with clinical trial terminology, while preserving the original numbers exactly.

4.8.2. Task 1.2 — Document-Level Walkthrough

Input abstract (English, excerpt):

> *We included 19 trials (17 RCTs and two cluster-RCTs). The 19 trials enrolled 395,650 participants, with ages ranging from six weeks to 60 years…*

Step 1 — Summary generation (250-token budget). The Step 1 prompt is submitted; the model generates a short plain-language summary capturing the key findings of the full abstract. This summary is not preserved in the output files but is used as the anchor for Step 2.
Step 2 — Guided simplification. The Step 2 prompt is constructed by prepending the Step 1 summary, then appending the full original abstract. The token budget is computed dynamically from the source word count. The model produces:

Table 1
Task 1.1 (sentence level) official results on Codabench (submission ID: 771785, "Llama3.1-8B PlanDriven", 48,809 sentences).

System	SARI	BLEU	FKGL	Compr	Split	Lev	Copies	Add	Del	Lex
Ours (Llama 3.1 8B, P5)	44.89	7.51	9.31	0.80	1.21	0.48	0.00	0.42	0.62	8.32

Table 2
Task 1.2 (document level) official results on Codabench (submission ID: 773049, "Llama3.1-8B Summary-Guided", 8,069 documents).

System	SARI	BLEU	FKGL	Compr	Split	Lev	Copies	Add	Del	Lex
Ours (Llama 3.1 8B, SG)	42.62	3.99	10.27	0.40	0.56	0.38	0.00	0.18	0.78	8.46

> *Researchers looked at 19 studies involving over 395,000 people who received a typhoid vaccine called TCV (Typhoid Conjugate Vaccine). They found that this vaccine is very good at preventing severe typhoid fever (about an 80% reduction)…*

Step 3 — Output cleanup. No preamble detected; the output passes through unchanged.
Observation. The model compressed "395,650 participants" to "over 395,000 people", replaced the technical abbreviation TCV with its full name, and converted the quantitative finding into plain-language terms. The summary anchor from Step 1 ensures that findings from later sections of the abstract (such as the 80% reduction figure) are captured in the final output rather than being lost to premature stopping.

5. Experimental Results

Official results on the Codabench evaluation platform are presented in Tables 1 and 2. Tables 3 and 4 compare our submissions against selected teams on the official CLEF 2026 SimpleText Codabench leaderboard (197 combined entries across both levels). Per-language SARI breakdowns were not exposed by the Codabench platform; Table 5 reports sentence counts and compression ratios per language as a proxy for multilingual coverage.

5.1. Task 1.1 Official Results

Our Task 1.1 system achieves SARI 44.89, outperforming both the GPT-4.1-mini commercial baseline (43.34) and the Llama 3.3 70B open model (42.33) despite using a significantly smaller model. The Copies score of 0.00 reflects that even sentences handled by the KEEP strategy undergo minor normalisation at generation time; strict token-level reproduction is therefore not observed for any output. The deletion score (Del 0.62) confirms appropriate removal of statistical parentheticals and certainty labels, while the addition score (Add 0.42) captures the plain-language expansions of technical terms introduced by the P5 rules.

5.2. Task 1.2 Official Results

Task 1.2 achieves SARI 42.62. The compression ratio of 0.40 (versus 0.80 for Task 1.1) reflects the document-level pipeline's tendency to produce a shorter, denser plain-language paragraph rather than simplifying each sentence independently. The deletion score (Del 0.78) is the highest across both tasks, consistent with the summary-guided design selecting and condensing core information rather than preserving every clause. The addition score (Add 0.18) is lower than Task 1.1, indicating that the document-level rewrite introduces fewer new plain-language expansions and instead focuses on compressive restructuring.

Table 3
Task 1.1 (sentence-level) results on the CLEF 2026 Codabench leaderboard, selected teams. The leaderboard ranks all submissions (sentence- and document-level) jointly by SARI; overall rank is out of 197 combined entries. Bold = our submission.

Team	System	SARI	Overall Rank
David Condrey	Claude Sonnet 4	47.43	3rd
David Condrey	Opus	47.26	4th
45494	FOSU-YZH hybrid	47.00	6th
manojkumar00011	**Llama 3.1 8B PlanDriven**	**44.89**	**86th**
dokyoon	bert	8.88	197th

Table 4
Task 1.2 (document-level) results on the CLEF 2026 Codabench leaderboard, selected teams. The leaderboard ranks all submissions (sentence- and document-level) jointly by SARI; overall rank is out of 197 combined entries. Bold = our submission.

Team	System	SARI	Overall Rank
RUN	DeepSeek-V4	48.85	1st
pascalm	RAG-BoN Q32L70	47.57	2nd
Tech. Univ. Kosice	Gemma-2B-IT QLoRA	47.18	5th
manojkumar00011	**Llama 3.1 8B Summary-Guided**	**42.62**	**125th**
Jaap Kamps	Source as prediction	8.88	196th

5.3. Leaderboard Context

Tables 3 and 4 place our results in the context of the full CLEF 2026 SimpleText leaderboard.

Our Task 1.1 entry ranks 86th out of 197 combined entries. The top systems use large commercial models (Claude Sonnet 4, SARI 47.43) or hybrid retrieval architectures (FOSU-YZH, SARI 47.00); our Llama 3.1 8B achieves SARI 44.89 without cloud infrastructure or fine-tuning, trailing the top system by 2.54 SARI points. For Task 1.2 (Table 4), our entry ranks 125th with SARI 42.62. The first-place system (DeepSeek-V4, 48.85) and second-place (RAG-BoN Q32L70, 47.57) both employ retrieval-augmented or ensemble approaches not replicated in our single-pass pipeline. Our score exceeds the last-placed system (Source as prediction, 8.88) by a substantial margin of 33.74 SARI points.

5.4. Per-Language Coverage

Table 5 reports per-language coverage for the Task 1.1 submission.

English dominates the distribution (20,259 sentences, 41.5% of total), followed by Spanish and French. Compression ratios for Latin-script languages cluster between 0.84 and 0.93, indicating consistent shortening behaviour across these languages. The values for Thai, Chinese, and Japanese are marked †: whitespace-based word counting is not linguistically valid for space-free scripts, so their reported compression ratios (0.42–0.53) reflect tokenisation artefacts rather than true output-length differences.

6. Discussion

6.1. Efficiency vs. Scale

Table 6 makes the paper's central argument visually explicit: our 8B model outperforms both a commercial API baseline and a 9× larger open model, while running entirely on consumer hardware at zero per-query cost.

Table 5
Per-language sentence counts and compression ratios for the Task 1.1 submission. Compression ratio = predicted words / source words. Values marked † are unreliable for space-free scripts (Thai, Chinese, Japanese) where whitespace-based word counting is not meaningful.

Language	Sentences	Compr	Avg Src Words
English (en)	20,259	0.93	23.6
Spanish (es)	7,553	0.92	24.4
French (fr)	7,220	0.87	27.5
Persian (fa)	6,986	0.84	23.1
Korean (ko)	3,146	0.92	14.1
Chinese (zh_HANS)	1,434	0.42†	6.0
Portuguese (pt)	1,201	0.92	22.2
Thai (th)	843	0.45†	21.0
German (de)	67	0.85	22.1
Japanese (ja)	56	0.53†	3.1
Hindi (hi)	44	0.74	45.2
Total	**48,809**		

Table 6
Efficiency vs. scale on Task 1.1 SARI. All entries are from the official CLEF 2026 Codabench leaderboard.

System	Params	Deployment	SARI
Claude Sonnet 4	undisclosed	commercial API	47.43
GPT-4.1-mini	undisclosed	commercial API	43.34
Llama 3.3 70B	70B	open-source	42.33
Ours (Llama 3.1 8B)	**8B**	**local, free**	**44.89**

The gap between our system (SARI 44.89) and the top entry (Claude Sonnet 4, 47.43) is 2.54 SARI points. The 4-bit quantisation (`q4_K_M`) allows Llama 3.1 8B to run within the 8 GB VRAM of the RTX 5050 (laptop GPU), making the full pipeline accessible without cloud infrastructure or rate-limit constraints.

6.2. Plan-Driven Prompt Analysis

The P5 prompt yielded a +1.64 SARI improvement over our preliminary Mistral 7B submission on Task 1.1, and outperformed the GPT-4.1-mini commercial baseline by 1.55 SARI despite using a significantly smaller model. This result suggests that structured prompt design can compensate substantially for model scale in simplification tasks.

The strategy enumeration in P5 (REPHRASE/SPLIT/DROP/KEEP) provides the model with an explicit decision space to reason over before generating output. By naming concrete operations, the prompt encourages the model to evaluate each sentence individually rather than applying a uniform surface-edit strategy. The zero Copies score (0.00) warrants one note: even sentences handled by the KEEP strategy are not returned as verbatim copies, because the Copies metric requires strict token-level reproduction; minor normalisation within the model's generation (such as punctuation standardisation) is enough to register as a non-copy.

6.3. Summary-Guided Pipeline Analysis

The summary-guided pipeline achieved SARI 42.62 on Task 1.2, compared to 42.59 for the preliminary Mistral 7B run—a difference of 0.03 SARI that should be interpreted as the two approaches performing at roughly equivalent aggregate levels. The more notable observation is the compression ratio: 0.40 at the

document level versus 0.80 at the sentence level. Document-level simplification in our system produces a shorter, denser plain-language paragraph rather than simplifying each sentence independently.

The Step 1 summary acts as a guard against the two failure modes identified in Section 4.5: hallucination drift and premature stopping. The deletion score (Del 0.78 for Task 1.2) is consistent with this design: the system removes non-essential detail while the summary anchor ensures core findings from all parts of the abstract are represented in the final output.

6.4. Multilingual Fidelity

The explicit language directive proved essential. In preliminary runs without it, frequent language leakage on Thai, Korean, and Persian inputs was the dominant failure mode. The instruction "OUTPUT LANGUAGE: {lang} (your output MUST be in {lang}, do NOT translate)" substantially reduced this failure mode; no wrong-language outputs were observed during manual spot-checks of the final submission files across all eleven languages.

6.5. Readability and Metric Breakdown

The FKGL score of 9.31 for Task 1.1 indicates approximately US Grade 9 reading level. Grade 9 remains above the lay-reader level implied by the task goal; accessible science writing typically targets Grade 6–8. Reducing output reading level without sacrificing SARI is a concrete direction for future work. Task 1.2 FKGL is slightly higher at 10.27, consistent with document-level rewriting producing somewhat more complex sentences than sentence-by-sentence simplification. **Limitation:** FKGL is calibrated for English and is not linguistically valid for the non-English outputs in our multilingual submission. Reported FKGL values for Thai, Korean, Chinese, Japanese, Persian, and Hindi should be treated as approximate indicators only.

The deletion score (Del: 0.62 for Task 1.1, 0.78 for Task 1.2) confirms appropriate removal of content—primarily statistical parentheticals, certainty labels, and redundant elaborations—driven by the P5 prompt rules. The addition score (Add: 0.42 for Task 1.1, 0.18 for Task 1.2) reflects the introduction of plain-language expansions of technical terms and abbreviations.

6.6. Medical Lexicon Ablation

During development we evaluated a 129-entry medical substitution lexicon. Application of this lexicon reduced SARI by 0.21–1.84 points on our English development evaluation, so it was disabled (`apply_lexicon=False`) in all final submissions. The P5 prompt's jargon-replacement instructions proved more adaptive: the model applies substitutions contextually rather than blindly, avoiding cases where a mechanical substitution would produce an ungrammatical or misleading output.

7. Conclusions and Future Work

We presented a system for multilingual biomedical text simplification at CLEF 2026 SimpleText Task 1 based on locally deployed Llama 3.1 8B. Our plan-driven sentence-level system (P5) achieved SARI 44.89 on Task 1.1, outperforming both a commercial GPT-4.1-mini baseline and a 70B open model, demonstrating that structured prompt engineering can compensate substantially for model scale. Our summary-guided document-level system achieved SARI 42.62 on Task 1.2. Both systems run on consumer hardware without cloud infrastructure, contributing to open and reproducible scientific NLP research. Reported SARI differences are point estimates without significance testing; future work should apply bootstrap confidence intervals to confirm these differences given the large dataset size.

Two directions stand out for future work: extending the P5 strategy set with a DEFINE operation that inserts inline glosses for retained technical terms; and fine-tuning the 8B model on a small high-quality simplification dataset to improve SARI without increasing inference cost.

Acknowledgments

The authors thank the CLEF 2026 SimpleText organising committee for providing a challenging and meaningful shared task and the Codabench evaluation platform. We also thank the Department of Computer Science and Engineering at Sri Sivasubramaniya Nadar College of Engineering for hosting the Summer Internship programme under which this work was carried out. We are grateful to the open-source Llama and Ollama communities for making large-language-model inference accessible on consumer hardware.

Declaration on Generative AI

During the preparation of this work, the authors used Claude (Anthropic) in order to: grammar and spelling check, paraphrase and reword. After using these tool(s)/service(s), the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] A. Siddharthan, A survey of research on text simplification, ITL – International Journal of Applied Linguistics 165 (2014) 259–298.
[2] K. Inui, A. Fujita, T. Takahashi, R. Iida, T. Iwakura, Text simplification for reading assistance: A project note, in: Proceedings of the 2nd International Workshop on Paraphrasing, 2003, pp. 9–16.
[3] E. Hasler, A. de Gispert, F. Stahlberg, A. Waite, B. Byrne, Source sentence simplification for statistical machine translation, Computer Speech & Language 45 (2017) 221–235.
[4] F. Alva-Manchego, C. Scarton, L. Specia, Data-driven sentence simplification: Survey and benchmark, Computational Linguistics 46 (2020) 135–187.
[5] T. Kew, J. Vamvas, R. Sennrich, BLESS: Benchmarking large language models on sentence simplification, arXiv preprint arXiv:2310.15773 (2023).
[6] L. Ermakova, et al., Overview of the CLEF 2021 SimpleText lab, in: Proceedings of CLEF 2021 – Conference and Labs of the Evaluation Forum, volume 12880 of *Lecture Notes in Computer Science*, Springer, 2021.
[7] L. Ermakova, et al., Overview of the CLEF 2024 SimpleText track, in: Proceedings of the CLEF 2024 Working Notes, CEUR Workshop Proceedings, 2024.
[8] L. Ermakova, et al., Overview of the CLEF 2025 SimpleText track: Simplify scientific text (and nothing more), in: Proceedings of CLEF 2025, Lecture Notes in Computer Science, Springer, 2025.
[9] L. Ermakova, et al., Overview of the CLEF 2026 SimpleText track: Simplify scientific texts (and nothing more), in: Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Lecture Notes in Computer Science, Springer, 2026. To appear.
[10] 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.
[11] A. Dubey, et al., The Llama 3 herd of models, arXiv preprint arXiv:2407.21783 (2024).
[12] Ollama Authors, Ollama: Get up and running with large language models locally, https://ollama.com, 2024. Version 0.30.6.
[13] K. C. Marturi, H. H. Elwazzan, LLM-guided planning and summary-based scientific text simplification: DS@GT at CLEF 2025 SimpleText, arXiv preprint arXiv:2508.11816 (2025).
[14] 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, arXiv preprint arXiv:2512.16541 (2025).
[15] J. Bakker, et al., Overview of the CLEF 2026 SimpleText task 1: Simplify scientific text, in: Proceedings of the CLEF 2026 Working Notes, CEUR Workshop Proceedings, 2026. To appear.
[16] A. Q. Jiang, et al., Mistral 7B, arXiv preprint arXiv:2310.06825 (2023).

A. Complete Prompts

Task 1.1 — P5 System Prompt

```
You are a biomedical text simplification expert. For each
sentence, internally decide the best strategy (REPHRASE
jargon / SPLIT into shorter sentences / DROP statistical
parentheticals / KEEP unchanged) then apply it.

OUTPUT LANGUAGE: {lang} (your output MUST be in {lang},
do NOT translate)

1. Preserve numbers, study sizes, drug names, populations.
2. Replace medical jargon with plain words in {lang}.
3. Translate certainty cues into plain words
   (e.g. 'low-certainty evidence' -> 'we are not very sure').
4. If already simple, output unchanged.
5. Output ONLY the simplified sentence.
   No preamble. No strategy label.
```

User message:

```
Simplify this sentence:

{complex_sentence}
```

Task 1.2 — Step 1 System Prompt (Summary Generation)

```
You are a biomedical expert. Write a short plain-language
summary (3-5 sentences) of the key findings in the abstract
below. Write entirely in {lang}.
```

User message:

```
Summarize the key findings in plain {lang}:

{complex_abstract}
```

Task 1.2 — Step 2 System Prompt (Guided Simplification)

```
You are a scientific text simplification expert. Rewrite
the biomedical abstract in plain {lang} for a general
audience (high school level).

OUTPUT LANGUAGE: {lang} (your output MUST be in {lang},
do NOT translate)

RULES:
1. Keep all numbers exactly as they appear.
2. You may drop dense statistical parentheticals if
   they obscure meaning.
3. Replace medical jargon with plain words in {lang}.
4. Short, clear sentences. Target Grade 7-8 reading level.
5. Output ONLY the simplified text -- no preamble,
   no quotes, no commentary.
```

User message:

```
Summary guide:
{step1_summary}

Rewrite this abstract in plain {lang}:

{complex_abstract}
```