Elsevier at CLEF 2026 SimpleText: Deconstructing Scientific Text Simplification with Direct and Multi-Stage LLM Approaches

Elsevier at CLEF 2026

Artemis Çapari, Hosein Azarbonyad, Zubair Afzal and Georgios Tsatsaronis

Elsevier, Amsterdam

Abstract

The CLEF 2026 SimpleText track focuses on making scientific text easier to understand for non-expert readers while preserving the reliability and accuracy of the source content. In this paper, we describe our approach to the scientific text simplification tasks, where we experiment with a range of prompting strategies using large language models. We begin by testing direct prompting approaches in both zero-shot and few-shot settings, and then explore several multi-stage variants that break the simplification process into smaller and more controlled steps. These include a plan-based approach, in which the model first generates structure and terminology guidance before producing a final simplified version, and a draft-based approach, in which terminology and sentence structure are simplified separately and then combined in a final step.

A central part of our work is the exploration of strategies for understanding which patterns and aspects of scientific text simplification are most effective. In addition to standard few-shot prompting, we experiment with selecting best-match examples for each input rather than relying only on randomly chosen demonstrations. We also investigate prompt guidance derived from analyses of simplification patterns in the gold training data, with the goal of capturing useful rewriting strategies and making the prompts more targeted and better aligned with the task. Across all settings, our main objective is to improve readability while preserving the original meaning of the source text. Our strongest configurations differ by task granularity: at the sentence level, pattern-derived guidance combined with relevance-aligned best-match retrieval performs best, whereas at the document level, direct one-pass prompting with diverse demonstrations is strongest and multi-stage decomposition does not improve over direct rewriting.

Keywords

Scientific Text Simplification, Scientific Documents, Scholarly Document Processing,

1. Introduction

Reliable scientific evidence is critical for informed decision-making, but scientific writing is often difficult for non-expert readers due to dense terminology, long sentence structures, and compressed reporting of methods and outcomes [1].

Text simplification for scientific communication therefore requires a careful balance: outputs must become more accessible without changing conclusions, quantitative evidence, or uncertainty language.

The CLEF 2026 SimpleText track targets this exact challenge, with Task 1.1 focusing on sentence-level simplification and Task 1.2 on full-document simplification in the Cochrane-auto setting [2]. In this paper, we report Elsevier's participation in both subtasks and present a unified GPT-4.1-based pipeline designed to compare direct prompting, decomposition-based generation, and retrieval-informed few-shot conditioning under a common setup [2].

Building on our previous SimpleText participation, our approach is motivated by two practical questions. First, does decomposition of simplification into intermediate steps improve controllability and quality relative to direct one-pass rewriting? Second, does relevance-aligned few-shot retrieval outperform random pooled sampling from the same candidate pool? To address these questions, we run a broad experimental matrix across legacy prompts (from our CLEF 2024 setup [3]), newer prompt

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany

versions, prompt-guidance variants derived from gold-pattern analysis, and decomposition ablations that isolate structure-focused and terminology-focused transformations.

Our main contributions are as follows:

- a unified framework for both sentence- and document-level simplification,
- a comparative evaluation of direct, plan-based, and draft-based decomposition strategies,
- an analysis of relevance-aligned few-shot retrieval versus random pooled sampling baselines,
- and an analysis of prompt-guidance derived from simplification patterns in training data.

The remainder of the paper is organized as follows. Section 2 describes the related work. Section 3 describes the methodology and system design. Section 4 details the experimental setup. Section 5 presents results and analysis for Task 1.1 and Task 1.2. Section 6 concludes with key findings and future work directions.

2. Related Work

Classical text simplification distinguishes lexical simplification (replacing difficult terms with simpler alternatives) from syntactic simplification (rewriting sentence structure to reduce grammatical complexity) [4]. Early lexical simplification relied on rule-based substitution, from frequency-based replacement [5] to more context-aware methods using context vectors and BERT-based ranking [6, 7, 8], while data-driven approaches learned substitution rules from large corpora [9, 10, 4]. Syntactic simplification similarly progressed from handcrafted and parser-based sentence-splitting rules [11] and cohesion- and multilingual-aware methods [12, 13, 14, 15] to data-driven statistical models [16, 17].

Neural approaches then shifted simplification toward end-to-end generation, first with sequence-to-sequence models [18] and attention mechanisms [19, 20], and later with large pre-trained language models adapted for controllable rewriting [21, 22]. This trajectory established the current paradigm in which simplification quality depends on both model capacity and control over rewrite behavior.

More recently, instruction-tuned large language models have enabled simplification through prompting rather than task-specific training, with zero-shot and few-shot in-context prompting, and retrieval of relevant demonstrations, emerging as practical control mechanisms. In parallel, decomposition and plan-then-write strategies that separate content planning from surface realization have been explored to improve controllability in generation. Our work situates itself in this line, comparing direct prompting, decomposition, and relevance-aligned few-shot retrieval within a single controlled setup.

A recurring difficulty in this area is evaluation. Reference-based metrics such as SARI [23] and BLEU [24], together with the reference-free FKGL readability score [25], capture complementary and sometimes conflicting aspects of simplification quality, and none directly measures whether the simplified text remains faithful to the source. This motivates our use of editing-behavior statistics and a semantic-consistency (NLI) diagnostic alongside the official metrics.

3. Methodology

3.1. Task Setting and Data

We address two subtasks from CLEF 2026 SimpleText Task 1: sentence-level simplification (Task 1.1) and document-level simplification (Task 1.2) [2]. Inputs are drawn from the Cochrane-auto setting, where Task 1.1 provides sentence units with paragraph/sentence indices and Task 1.2 provides full abstract-level documents.

Our outputs follow the official JSON schema with per-instance predictions and run identifiers.

3.2. Common System Design

Our methodology treats scientific simplification as a controlled transformation problem with two competing objectives: (i) maximize accessibility and (ii) preserve epistemic faithfulness to the source.

We model this as a set of intervention families applied to the same underlying LLM (GPT-4.1), so that observed performance differences can be attributed to prompting strategy rather than model identity.

Across both tasks, every method follows the same high-level constraints: preserve claims, quantitative evidence, and uncertainty statements; reduce lexical and structural complexity for non-expert readers; and avoid unsupported additions or over-interpretation.

This shared constraint set allows controlled cross-method comparisons by varying one methodological dimension at a time (for example, context use, few-shot selection, or decomposition design).

3.3. Task 1.1 Methods (Sentence-Level)

For Task 1.1, we evaluate two base methods:

- `sentence_only`: simplify each sentence independently,
- `sentence_with_context`: simplify the target sentence with full-document context provided for consistency.

Both methods are tested in zero-shot and few-shot settings. Legacy CLEF 2024 prompting is retained and reported as a baseline. In newer prompt versions, constraints are explicit about preserving factual claims, numbers, and uncertainty qualifiers (for example, may, likely, probably, little to no difference).

The sentence-level design explicitly tests whether document context improves local rewriting quality or instead introduces contextual interference. This comparison is central for understanding when context helps sentence simplification versus when it harms precision.

3.4. Task 1.2 Methods (Document-Level)

For Task 1.2, we evaluate multiple method families:

- `direct`: one-pass full-document simplification,
- `decomposed` (plan-based):
    1. structure-planning stage,
    2. terminology-planning stage,
    3. constrained final rewrite stage,
- `decomposed_dual` (draft-based):
    1. terminology-focused draft,
    2. structure-focused draft,
    3. conservative final merge stage.

Methodologically, these variants reflect two simplification philosophies: direct rewriting assumes the model can jointly optimize readability and faithfulness in one pass, whereas decomposed rewriting assumes explicit intermediate representations improve controllability and reduce objective interference.

The two decomposed families differ in what their intermediate stages produce. In the plan-based variant, the first two stages do not rewrite the abstract: the structure-planning stage returns a plan (which sentences to split, reorder, or compress, which information order to preserve, and which numeric and uncertainty items must be kept), and the terminology-planning stage returns a glossary mapping jargon and acronyms to plain-language definitions. Only the final stage produces simplified text: it rewrites the source using both the plan and the glossary.

In the draft-based variant, each of the first two stages produces a complete simplified abstract rather than a plan: a terminology-focused pass replaces jargon while leaving sentence structure largely intact, and a structure-focused pass splits and reorders sentences while retaining technical terms. A final merge stage then combines the two drafts, resolving conflicts conservatively toward the original abstract.

Both decomposed families apply the same prompt guidance overlay as the direct methods and end with a grounding-validation step that, on failure, triggers a single repair pass re-prompting the model to restore numeric values and uncertainty qualifiers without adding new claims.

To isolate component effects, we additionally analyze structure-only and terminology-only variants as factor-isolation probes that estimate the individual contribution of syntactic restructuring versus lexical/terminological substitution in full-document simplification.

3.5. Prompt Guidance from Gold Simplification Patterns

Prompt guidance is constructed with an explicit two-stage pipeline.

Stage 1: Pattern mining from training pairs. We mine sentence-level and document-level parallel training pairs using a deterministic analysis script. For each pair, we compute compression ratio, sentence-splitting behavior, numeric-token preservation/overlap, and long-word reduction. We then aggregate these into corpus-level summary statistics (means/medians/rates).

In parallel, we mine lexical transformation signals using token-set differences between complex and simplified texts. This produces: (i) frequently dropped complex tokens, (ii) frequently added simpler tokens, and (iii) candidate complex-to-simple substitution pairs filtered by support thresholds. We also measure hedge/uncertainty preservation for a predefined phrase list (for example, may, probably, little to no, uncertain), and compute per-phrase preservation rates.

To support prompt synthesis, the mining step also samples a few short complex-to-simple example pairs, grouped by how much they shorten the text (high, moderate, or light compression). The script outputs both JSON and Markdown artifacts containing all summaries, lexical patterns, hedge statistics, sampled pairs, and derived heuristic guidance; the mined artifact is summarized in Appendix C.

Stage 2: System prompt synthesis. Subsequently, GPT-4.1 consumes the mined pattern artifact and produces a structured JSON output with guidance rules, candidate jargon substitutions and must-preserve constraints; the synthesis prompt is reproduced in Appendix D (GSYN). In our submitted runs, we evaluate two integration variants: (i) appending only guidance rules to handcrafted system prompts, and (ii) structured-guidance prompting that injects guidance rules together with jargon substitutions and must-preserve constraints into the handcrafted prompt template.

This component tests whether corpus-derived instruction signals improve prompt specificity beyond manual prompt engineering while keeping the baseline prompt families directly comparable.

3.6. Few-Shot Demonstration Selection

We evaluate whether relevance-aligned demonstration retrieval improves performance beyond standard pooled few-shot sampling.

The random pooled baseline draws demonstrations from a filtered candidate pool (quality/length filtering and overlap exclusion). The relevance-aligned condition ranks candidates with a composite similarity score combining lexical overlap, numeric-token overlap, and complexity-profile similarity (weights: content = 0.65, complexity = 0.20, numeric = 0.15), followed by near-duplicate filtering for diversity. Lexical overlap is computed as Jaccard similarity between source content-token sets, and numeric-token overlap is computed as Jaccard similarity between extracted numeric-token sets. The complexity profile is a three-value vector computed from the source complex text: normalized length $(min(word\_count, 120)/120)$, long-word ratio (fraction of tokens with length >= 8), and numeric-token ratio (fraction of tokens containing at least one digit). Complexity similarity between evaluation instance and candidate is then 1 minus the mean absolute difference across these three components, clipped to [0, 1], so higher values indicate closer structural/lexical difficulty profiles.

This setup treats pooled seeded sampling as a reference condition and estimates the performance gain from relevance-aligned few-shot retrieval. Conceptually, this tests a core retrieval-augmentation hypothesis: demonstrations should be more useful when they are closer to the target instance in content, numeric profile, and complexity characteristics.

4. Experiments

4.1. Experimental Scope and Reproducibility

We evaluate both subtasks under a unified experimental protocol: Task 1.1 (sentence-level simplification) and Task 1.2 (document-level simplification).

All compared runs use the same base model family (GPT-4.1) and the same submission/evaluation pipeline, with controlled variation in prompting strategy, few-shot retrieval strategy, and decomposition design. This design isolates method effects under a fixed model configuration.

For reproducibility, each reported result is tied to a fixed run configuration and archived submission artifact. In total, we evaluate 32 runs: 13 sentence-level and 19 document-level. All runs use a single fixed random seed (123) and are reported as single-run results; we do not estimate run-to-run variance, so small differences between closely ranked configurations should be read as indicative rather than definitive.

4.2. Data, Inputs, and Output Schema

Experiments use the official CLEF 2026 SimpleText Task 1 data in the Cochrane-auto setting [2]. Task 1.1 inputs are sentence records with paragraph/sentence indexing metadata, while Task 1.2 inputs are full-document records. Output format follows the organizer JSON schema with run identifiers and per-instance predictions. Task 1.1 outputs retain sentence-level indexing fields required by the evaluation platform.

4.3. Method Families and Run Groups

We group runs by intervention family so that each comparison isolates a specific methodological choice. Tables 1 and 2 define the comparison groups used throughout the Results section.

Table 1
Task 1.1 run groups

Family	Description	Prompt IDs
Legacy baselines	CLEF 2024 prompt variants (sentence-only and sentence-with-context).	S1, S6
Enhanced direct prompting	Newer prompt versions in zero-shot and few-shot settings.	S2, S3, S4, S5
Prompt-guided prompting	Runs with guidance mined from gold simplification patterns.	S3+G1S, S4+G1S
Structured-guidance prompting	Runs injecting guidance + jargon substitutions + must-preserve constraints into the prompt template.	S3+G2S, S4+G2S
Few-shot retrieval variants	Random pooled sampling versus relevance-aligned best-match retrieval under matched prompt families.	S4

Table 2
Task 1.2 run groups

Family	Description	Prompt IDs
Direct document simplification	One-pass baseline/enhanced prompt variants.	D1, D2, D3, D4, D5
Decomposed plan-based	Structure plan ->terminology plan ->final rewrite.	D6, D7, D8, D9
Decomposed dual-draft	Terminology draft ->structure draft ->final merge.	D10, D11, D12, D13
Factor-isolation variants	Structure-only and terminology-only variants.	D10, D14
Prompt-guided prompting	Document-level guidance from mined simplification patterns.	D4+G1D, D5+G1D
Structured-guidance prompting	Runs injecting guidance + jargon substitutions + must-preserve constraints into the prompt template.	D4+G2D, D5+G2D
Few-shot retrieval variants	Random pooled sampling versus relevance-aligned best-match retrieval under matched prompt families.	D5, D9

4.4. Few-Shot Selection

We compare two few-shot retrieval conditions, summarized in Table 3.

Table 3
Few-shot retrieval conditions

Condition	Selection mechanism	Controls
Random pooled sampling	Demonstrations are randomly sampled from a filtered candidate pool (quality/length filtering and overlap exclusion with evaluation identifiers).	Reproducibility is maintained through fixed run configuration and deterministic sampling controls.
Relevance-aligned best-match retrieval	Demonstrations are ranked by a composite similarity score using lexical overlap, numeric overlap, and complexity-profile similarity.	Near-duplicate filtering preserves demonstration diversity; the composite-similarity components and weights are defined in Section 3.6.

5. Results and Analysis

Tables 4-5 report the core quantitative results used in this paper. We present one main comparison table for Task 1.1, one main comparison table for Task 1.2, and one focused ablation table for document-level decomposition variants.

Across subtasks, method effects transfer with different magnitudes. To interpret these differences, we read the metrics as complementary rather than interchangeable signals. SARI rewards edit operations (additions, deletions, and retained tokens) that agree with the reference simplifications, so it credits a system for actively transforming the source toward the reference [23]. BLEU instead rewards n-gram overlap with the reference surface form, so it favors outputs that stay close to reference wording and penalizes aggressive rewriting even when that rewriting is appropriate [24]. FKGL is a surface readability score driven by sentence length and word length, and is independent of the reference, so it can be optimized by short sentences and short words regardless of meaning preservation [25]. The editing-behavior metrics (compression ratio, sentence splitting, additions, deletions, and lexical complexity) describe how an output was produced and explain the SARI/BLEU/FKGL trade-offs we observe. We use these behavioral metrics to explain why specific runs lead on specific metrics. These explanations are observational, drawn from aggregate run-level statistics rather than controlled interventions on a single factor.

5.1. Task 1.1: Sentence-Level Simplification

Table 4
Main Task 1.1 results

Method family	Run description	SARI	BLEU	FKGL	Compr.	Sent. Split	Levenshtein	Copies	Additions	Deletions	Lex. Compl.
Prompt-guided	FS3 + best-match	42.07	12.81	8.12	0.69	1.07	0.62	0.00	0.20	0.49	8.42
Structured-guidance	FS3 + best-match	41.73	12.98	8.20	0.69	1.05	0.62	0.00	0.20	0.48	8.39
Prompt-guided	Zero-shot	41.47	11.76	9.71	0.78	0.99	0.63	0.00	0.25	0.44	8.42
Structured-guidance	Zero-shot direct	40.56	11.56	9.85	0.81	0.99	0.64	0.00	0.26	0.41	8.45
Structured-guidance	FS3 + random pooled	39.30	12.20	8.97	0.81	1.08	0.67	0.00	0.23	0.38	8.42
Legacy	Context + FS3 best-match	38.50	12.41	10.61	0.80	0.98	0.68	0.00	0.18	0.37	8.60
Non-guided enhanced	FS3 + best-match	38.03	12.33	9.25	0.77	1.08	0.69	0.00	0.16	0.38	8.58
Legacy	Sentence-only zero-shot	37.13	8.33	10.91	0.99	1.11	0.64	0.00	0.38	0.34	8.39
Non-guided enhanced	Sentence-with-context baseline	36.13	11.24	11.50	0.90	0.98	0.71	0.00	0.23	0.31	8.56
Non-guided enhanced	FS3 + random pooled	34.94	11.18	10.06	0.94	1.15	0.74	0.00	0.23	0.26	8.52
Non-guided enhanced	Sentence-only baseline	34.79	10.83	11.01	0.89	1.00	0.74	0.00	0.21	0.28	8.55

The highest-SARI sentence run is prompt-guided FS3 + best-match (42.07), followed by structured-guidance FS3 + best-match (41.73). Retrieval quality has a clear effect in non-guided enhanced runs: best-match outperforms random pooled sampling by +3.09 SARI (+1.15 BLEU, -0.81 FKGL), with lower Levenshtein distance (0.69 vs 0.74) and lower Additions (0.16 vs 0.23). Prompt guidance further improves

the best-match setup (+4.04 SARI, +0.48 BLEU, -1.13 FKGL relative to non-guided best-match) and lowers lexical complexity (8.42 vs 8.58). Structured guidance substantially strengthens the non-guided random few-shot setup (+4.36 SARI, +1.02 BLEU, -1.09 FKGL relative to non-guided FS3 + random pooled), while its best-match variant gives the highest BLEU among the top sentence runs.

Legacy context-aware prompting remains competitive (SARI 38.50) and serves as a historical anchor. The SARI ranking at sentence level tracks how much each run actually edits the source. The weakest runs are the non-guided sentence-only and FS3 + random pooled baselines, which barely transform the input (compression ratio 0.89 and 0.94, deletions 0.28 and 0.26) and therefore leave much of the source jargon in place, yielding low SARI (34.79 and 34.94) despite competitive BLEU. Guidance and structured guidance raise SARI primarily by increasing the share of reference-aligned edits: the top guided run compresses more and deletes more (compression 0.69, deletions 0.49) while keeping lexical complexity low (8.42), which is exactly the behavior the references reward. This also explains why guidance helps the random few-shot setup more than the best-match setup (+4.36 vs +4.04 SARI over their non-guided counterparts): best-match demonstrations already supply an aligned editing pattern, so the explicit constraints are partly redundant, whereas under noisier random demonstrations the constraints provide the simplification signal the examples lack. The BLEU/SARI split among the top runs follows the same logic: the structured-guidance best-match run attains the highest BLEU (12.98) because it edits slightly more conservatively (deletions 0.48), keeping more reference n-grams, while the prompt-guided run trades a small amount of BLEU for the most reference-aligned deletions and the top SARI. This lexical-complexity effect is consistent with the guidance mechanism: structured guidance roughly halves the survival of the mined jargon source terms in the output relative to the non-guided run (0.572 to 0.233), with prompt-guided guidance intermediate (0.300) (Appendix B, Table 11), confirming that the guidance removes the targeted complex tokens rather than merely paraphrasing around them.

5.2. Task 1.2: Document-Level Simplification

Table 5
Main Task 1.2 results

Method family	Run description	SARI	BLEU	FKGL	Compr.	Sent. Split	Levenshtein	Copies	Additions	Deletions	Lex. Compl.
Non-guided enhanced	Direct FS3 + random pooled	45.33	6.92	9.59	0.40	0.61	0.43	0.00	0.10	0.71	8.42
Structured-guidance	FS3 + random pooled direct	45.20	4.54	9.13	0.33	0.53	0.37	0.00	0.11	0.78	8.32
Structured-guidance	FS3 + best-match direct	44.82	4.10	8.82	0.30	0.51	0.36	0.00	0.09	0.79	8.31
Structured-guidance	Zero-shot direct	44.31	5.50	8.87	0.40	0.62	0.40	0.00	0.16	0.76	8.24
Non-guided enhanced	Direct FS3 + best-match	44.25	4.38	9.69	0.31	0.52	0.39	0.00	0.06	0.75	8.56
Prompt-guided	Zero-shot	44.19	4.96	8.87	0.37	0.59	0.38	0.00	0.15	0.77	8.26
Prompt-guided	FS3 + best-match	43.94	2.78	9.17	0.26	0.45	0.34	0.00	0.07	0.81	8.45
Legacy	FS3	43.50	5.81	11.05	0.45	0.56	0.41	0.00	0.18	0.72	8.50
Legacy	Zero-shot	42.98	6.29	10.42	0.50	0.62	0.42	0.00	0.21	0.70	8.48
Non-guided enhanced	Direct baseline	40.70	11.15	8.75	0.64	0.92	0.59	0.00	0.17	0.50	8.55

The highest document-level SARI is obtained by direct FS3 + random pooled prompting (45.33), with structured-guidance FS3 + random pooled close behind (45.20). In contrast to sentence-level behavior, best-match retrieval lowers performance in the non-guided direct setup (-1.08 SARI, -2.54 BLEU, +0.10 FKGL versus random pooled), while the structured-guidance direct setup shows a smaller SARI drop from random pooled to best-match (-0.38) and lower FKGL (8.82). The structured-guidance zero-shot direct run reaches 44.31 SARI with FKGL 8.87 and the lowest lexical complexity among top document runs (8.24), indicating a strong readability profile under zero-shot conditions. Prompt-guided runs remain competitive, with the FS3 + best-match variant showing the lowest Levenshtein distance (0.34) among the listed document runs.

Why does retrieval help at sentence level but not at document level? A direct diagnostic on the retrieval signal itself provides the first part of the answer. Using the same composite similarity that drives best-match selection (content 0.65, complexity 0.20, numeric 0.15), we measure how strongly the top-ranked demonstration stands out from the candidate pool for each evaluation instance (Appendix A, Table 8). At sentence level the best match is well separated from the pool: its mean top-1 similarity

Table 6
Task 1.2 decomposition and factor-isolation ablations

Variant	SARI	BLEU	FKGL	Compr.	Sent. Split	Levenshtein	Copies	Additions	Deletions	Lex. Compl.
Direct FS3 + random pooled	45.33	6.92	9.59	0.40	0.61	0.43	0.00	0.10	0.71	8.42
Decomposed FS3 + random pooled	37.80	8.99	11.66	0.89	0.95	0.63	0.00	0.30	0.39	8.49
Decomposed zero-shot	34.67	8.38	12.12	1.27	1.43	0.68	0.01	0.41	0.22	8.49
Decomposed FS3 + best-match	33.70	8.85	12.16	1.20	1.34	0.70	0.01	0.38	0.23	8.53
Decomposed-dual zero-shot	30.39	11.27	13.76	1.00	0.97	0.80	0.00	0.18	0.19	8.74
Terminology-only zero-shot	28.42	10.09	15.75	1.17	0.98	0.80	0.00	0.27	0.14	8.69
Structure-only zero-shot	22.23	11.85	11.16	0.96	1.13	0.90	0.00	0.06	0.10	8.83

is 0.4474, scores span a wide range (0.149 to 1.000, including near-exact matches), and the top match scores 81.6% above the pool average. At document level the same procedure is far less discriminative: the mean top-1 similarity is only 0.3107, the top-1 scores are nearly flat across instances (0.260 to 0.359), and the best match exceeds the pool average by only 19.5%. In other words, at document level the "best" match is barely more relevant than a randomly drawn candidate, so best-match retrieval cannot reliably surface a genuinely aligned abstract. A per-component breakdown (Appendix A, Table 9) localizes the failure: the lexical/content component, which carries most of the matching signal, collapses from a +599.5% best-over-pool lift at sentence level to +59.3% at document level, while the complexity component is near-saturated and uninformative in both tasks (best-over-pool lift +7.6% and +1.9%). This low discriminability is expected: a long, heterogeneous abstract is a high-dimensional object, and bag-of-token overlap rarely isolates a single closely matching document, whereas a short sentence is far more likely to have a near-twin in the candidate pool.

The editing-behavior metrics explain the second part: why the weak document-level retrieval signal is not merely neutral but mildly harmful. When best-match selection does latch onto a similar abstract, the most similar abstracts in this corpus are themselves heavily compressed, so conditioning on them pushes the model toward more aggressive deletion (best-match compression 0.31 and deletions 0.75 versus random pooled 0.40 and 0.71). Beyond a point, additional deletion removes content that the references retain, which lowers SARI keep/precision and sharply reduces BLEU (4.38 versus 6.92). Random pooled demonstrations, being more diverse, induce a more moderate edit rate that better matches the reference editing distribution over a long, heterogeneous document. By contrast, a best-matched sentence is a short unit whose numeric and complexity profile closely matches the target, so it calibrates a local edit without much risk of discarding content. This also clarifies the contrast with the document direct baseline, which records the highest BLEU (11.15) but a lower SARI (40.70): it preserves the most source text (compression 0.64), staying close to the reference surface for BLEU while performing too few of the edits that SARI rewards. In this run set, the document task favors moderate, content-preserving compression, consistent with the strong performance of direct one-pass prompting with diverse demonstrations.

The sentence and document tasks also reward different operating points, and the gold references show why directly. Measured on the cochrane-auto test references, the document-level gold simplification compresses the whole abstract to a mean length ratio of 0.588 (median 0.582), that is, a large share of source content is removed. The sentence-level references behave differently: 38.4% of source sentences are deleted outright in the reference alignment, while the sentences that are retained are rewritten at essentially unchanged length (median length ratio 1.00, mean 1.14, the latter inflated by occasional added definitions). Document simplification is therefore distributed compression across the abstract, whereas sentence simplification is closer to a keep-or-delete decision followed by near-length-preserving lexical rewriting of the retained sentences. This asymmetry is the core reason the same intervention transfers with different magnitude: a method tuned to local, length-preserving sentence rewriting under-compresses documents, and a method tuned to document-level content removal over-edits individual sentences. Notably, the highest-SARI document systems compress more than the references on average (system compression 0.40 and below versus reference 0.588), so SARI here rewards high deletion recall even past the reference length, the same aggressive deletion that depresses BLEU.

5.3. Decomposition and Factor-Isolation Findings

Decomposition variants do not outperform direct rewriting on SARI in this run set. Under matched enhanced few-shot settings, decomposed FS3 + random pooled is -7.53 SARI behind direct FS3 + random pooled, with higher BLEU but substantially higher FKGL, higher Levenshtein distance (0.63 vs 0.43), and markedly higher Additions (0.30 vs 0.10). Factor-isolation runs show asymmetric behavior: terminology-only outperforms structure-only on SARI by 6.19 points, but with lower BLEU, much higher FKGL, and higher compression (1.17 vs 0.96).

The decomposition penalty is consistent with length expansion. The decomposed pipelines route the document through intermediate planning or drafting stages that introduce explanatory material, and the editing-behavior metrics show the result directly: decomposed runs have compression ratios at or above 1.0 (1.27 for decomposed zero-shot, 1.20 for best-match) and high additions (0.41 and 0.38), meaning the output is as long as or longer than the source. Expansion runs against the reference simplifications, which are shorter, so SARI falls while FKGL rises as sentences accumulate clauses. The reason decomposition still earns higher BLEU than direct rewriting is the same mechanism viewed from the other side: by adding and retaining material, these outputs preserve more reference n-grams locally even while diverging from the references' overall compression. The factor-isolation contrast sharpens the account. Terminology-only editing substitutes hard words for plain ones, which are precisely the token-level operations the references and SARI reward, so it scores higher SARI (28.42) even though its added definitions inflate length and produce the worst FKGL in the table (15.75). Structure-only editing splits and reorders sentences but leaves technical vocabulary in place (additions 0.06, the lowest in the table); with little lexical change, few of its edits match the simpler reference tokens, so SARI is lowest (22.23) despite a more favorable FKGL (11.16). Decomposing FKGL into its two drivers on the actual outputs (Appendix B, Table 10) supports the mechanism for the terminology-only versus structure-only gap: structure-only averages about 10.1 words per sentence because it splits sentences, whereas terminology-only averages about 13.1 and does not split, while syllables per word are essentially identical across variants (about 1.68). Lower FKGL is therefore primarily associated with sentence splitting rather than lexical substitution, which is why the terminology-focused variant cannot match the structural variant's readability score even though it scores higher on SARI. In short, at document level lexical simplification drives SARI while structural change mainly affects readability metrics, and the two do not substitute for each other.

5.4. Semantic Consistency

To complement task metrics with a standard semantic faithfulness signal, we score source-to-output NLI using a pretrained cross-encoder (`cross-encoder/nli-distilroberta-base`), treating the source as premise and the generated output as hypothesis.

Table 7
NLI-based semantic consistency diagnostic (source premise -> output hypothesis).

Level	Run	Entailment	Contradiction	Neutral
Document	Direct baseline	0.137	0.126	0.737
Document	Direct FS3 + random pooled	0.154	0.097	0.749
Document	Structured-guidance FS3 + random pooled	0.120	0.114	0.766
Document	Direct FS3 + best-match	0.194	0.137	0.669
Document	Prompt-guided FS3 + best-match	0.149	0.143	0.709
Sentence	FS3 + random pooled (non-guided)	0.788	0.015	0.197
Sentence	Sentence-only baseline	0.836	0.024	0.141
Sentence	FS3 + best-match (non-guided)	0.892	0.016	0.092
Sentence	Prompt-guided FS3 + best-match	0.834	0.017	0.149

Sentence-level outputs show high entailment and low contradiction across runs, while document-

level outputs show much higher neutral rates. We read this asymmetry mainly as a limitation of the diagnostic rather than as clean evidence about faithfulness: sentence-pair NLI is unreliable for long, heavily compressed abstracts, where source–output pairs are truncated to the encoder's sequence limit and a shorter summary is readily labelled neutral rather than entailed. The document-level scores also do not vary monotonically with editing intensity, so they cannot be used to rank runs by faithfulness. We therefore treat NLI only as a coarse secondary check and deliberately avoid strong claims that any particular configuration is more or less faithful than another; substantiating such claims would require a targeted human or reference-based faithfulness evaluation, which we leave to future work.

5.5. Analysis Across Tasks

Across both subtasks, three practical patterns emerge, and each is explained by how a method edits rather than by the method label alone. First, in this run set, direct rewriting is the strongest document-level strategy under SARI-based ranking, because it achieves high deletion recall against a reference distribution that itself removes substantial content (reference compression 0.588), without the length expansion that decomposition introduces; indeed the top systems compress somewhat past the reference length. Second, relevance-aligned few-shot retrieval helps at sentence level but not at document level. The underlying cause is a retrieval-quality gap: the composite similarity is highly discriminative for short sentences (best match 81.6% above the pool average) but nearly flat for long abstracts (only 19.5% above the pool average), so document-level best-match cannot reliably identify a genuinely aligned demonstration, and when it does, the most similar abstracts are heavily compressed and bias long inputs toward over-compression and content loss. Third, both prompt guidance and structured guidance provide substantial gains over non-guided sentence baselines, because they convert near-copying baselines into active, reference-aligned editors; their document-level effect is smaller because the strong document baselines already operate near the reference editing distribution.

The metric-level trade-offs follow the same mechanism. SARI favors runs whose edit operations match the references, BLEU favors runs that retain reference wording, and FKGL favors short surface forms; no single run maximizes all three, so the run that leads on SARI (most reference-aligned editing) is rarely the run that leads on BLEU (most conservative editing) or the run that leads on FKGL (most surface-shortened). This is why, for example, the document direct baseline tops BLEU while a more aggressive few-shot run tops SARI, and why structured-guidance runs lead on lexical complexity and FKGL without leading on SARI. As a secondary observation, NLI consistency differs by granularity (Section 5.4): sentence-level outputs show high entailment and low contradiction, while document-level outputs show much higher neutral rates.

These findings support a task-specific recommendation: prioritize guided and retrieval-enhanced prompting at sentence level, while preserving strong direct baselines and readability-aware alternatives at document level. For model selection, use SARI/BLEU/FKGL jointly and treat NLI consistency as an auxiliary safety diagnostic when comparing otherwise similar runs. A practical future direction is to enrich retrieval with behavior-aware signals (for example, expected compression/deletion tendency of candidate demonstrations) in addition to lexical, numeric, and coarse complexity matching, especially for document-level selection where current discriminability is low. Because our explanations are post-hoc and based on aggregate behavioral metrics, we treat them as mechanisms consistent with the data rather than as causally isolated effects

6. Conclusion

We reported our participation in the CLEF 2026 SimpleText Task 1, covering sentence-level (Task 1.1) and document-level (Task 1.2) scientific text simplification [2]. Using a single base model (GPT-4.1) under a fixed submission and evaluation pipeline, we conducted a controlled comparison of prompting strategies along three methodological axes: direct versus decomposition-based rewriting, relevance-aligned versus random pooled few-shot retrieval, and pattern-derived prompt guidance versus non-guided prompting.

In total we evaluated 32 runs (13 sentence-level and 19 document-level), with method effects interpreted through complementary task metrics (SARI, BLEU, FKGL) and editing-behavior statistics.

Our main findings are as follows:

Granularity determines which strategy wins. At sentence level, guided prompting combined with relevance-aligned best-match retrieval produced the strongest runs (top SARI 42.07). At document level, direct one-pass prompting with diverse (random pooled) demonstrations was strongest (top SARI 45.33), and decomposition did not improve SARI in this run set.

Retrieval value depends on demonstration discriminability. Best-match retrieval improved sentence-level outputs (+3.09 SARI over random pooled in the non-guided setting) but degraded non-guided document-level outputs (-1.08 SARI). A retrieval diagnostic explains this asymmetry: the composite similarity is highly discriminative for short sentences (top match 81.6% above the pool average) but nearly flat for long abstracts (only 19.5% above), so document-level best-match cannot reliably surface a genuinely aligned demonstration and, when it does, biases long inputs toward over-compression.

Guidance converts near-copying baselines into reference-aligned editors. Both prompt guidance and structured guidance produced substantial sentence-level gains (up to +4.36 SARI over non-guided baselines) and measurably reduced survival of targeted jargon terms, while their document-level effect was smaller because strong direct baselines already edit near the reference distribution. Metric trade-offs are systematic, not incidental. No single run maximized SARI, BLEU, and FKGL jointly; reference-aligned editing (SARI), surface retention (BLEU), and surface shortening (FKGL) reward different operating points, which is why, for example, the document direct baseline led on BLEU while a more aggressive few-shot run led on SARI.

These results indicate a task-specific recommendation: prioritize guided and retrieval-enhanced prompting at sentence level, while preserving strong direct baselines and readability-aware alternatives at document level, and use SARI/BLEU/FKGL jointly along with a semantic-consistency check when comparing otherwise similar runs.

Future work follows directly from these findings. The most promising direction is behavior-aware demonstration retrieval that augments lexical, numeric, and complexity matching with the expected editing behavior of candidate demonstrations (for example, their compression and deletion tendencies), particularly for document-level selection where current discriminability is low. A controlled study that varies compression rate at a fixed prompt would also test the over-compression account more directly, and extending the comparison to additional base models would assess how far these prompting effects generalize beyond GPT-4.1.

Declaration on Generative AI

During the preparation of this work, the authors used Claude Code to set up experiments and Claude Cowork to improve writing. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] P. Plavén-Sigray, G. J. Matheson, B. C. Schiffler, W. H. Thompson, The readability of scientific texts is decreasing over time, Elife 6 (2017) e27725.

[2] L. Ermakova, H. Azarbonyad, J. Bakker, G. K. Shahi, B. Vendeville, J. Kamps, Clef 2026 simpletext track: Simplify scientific text (and nothing more), in: European Conference on Information Retrieval, Springer, 2026, pp. 215–224.

[3] A. Çapari, H. Azarbonyad, Z. Afzal, G. Tsatsaronis, Enhancing scientific document simplification through adaptive retrieval and generative models., in: CLEF (Working Notes), 2024, pp. 3206–3229.

[4] S. S. Al-Thanyyan, A. M. Azmi, Automated text simplification: a survey, ACM Computing Surveys (CSUR) 54 (2021) 1–36.

[5] J. Carroll, G. Minnen, Y. Canning, S. Devlin, J. Tait, Practical simplification of english newspaper text to assist aphasic readers, in: AAAI-98 workshop on integrating artificial intelligence and assistive technology, Madison, WI, 1998, pp. 7–10.
[6] S. Bott, L. Rello, B. Drndarević, H. Saggion, Can spanish be simpler? lexsis: Lexical simplification for spanish, in: COLING, 2012, pp. 357–374.
[7] O. Biran, S. Brody, N. Elhadad, Putting it simply: a context-aware approach to lexical simplification, in: ACL, 2011, pp. 496–501.
[8] J. Qiang, Y. Li, Y. Zhu, Y. Yuan, X. Wu, Lexical simplification with pretrained encoders, in: AAAI, volume 34, 2020, pp. 8649–8656.
[9] C. Scarton, Horacio saggion, automatic text simplification. synthesis lectures on human language technologies, april 2017. 137 pages, isbn: 1627058680 9781627058681, Natural Language Engineering 26 (2020) 489–492.
[10] W. Coster, D. Kauchak, Simple english wikipedia: a new text simplification task, in: ACL, 2011, pp. 665–669.
[11] R. Chandrasekar, C. Doran, S. Bangalore, Motivations and methods for text simplification, in: COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics, 1996.
[12] A. Siddharthan, Syntactic simplification and text cohesion, Research on Language and Computation 4 (2006) 77–109.
[13] A. Siddharthan, Text simplification using typed dependencies: A comparision of the robustness of different generation strategies, in: The 13th European Workshop on Natural Language Generation, 2011, pp. 2–11.
[14] D. Ferrés, M. Marimon, H. Saggion, A. AbuRa'ed, Yats: yet another text simplifier, in: NLDB, Springer, 2016, pp. 335–342.
[15] C. Scarton, A. P. Aprosio, S. Tonelli, T. M. Wanton, L. Specia, Musst: A multilingual syntactic simplification tool, in: IJCNLP, 2017, pp. 25–28.
[16] Z. Zhu, D. Bernhard, I. Gurevych, A monolingual tree-based translation model for sentence simplification, in: COLING, 2010, pp. 1353–1361.
[17] K. Woodsend, M. Lapata, Learning to simplify sentences with quasi-synchronous grammar and integer programming, in: EMNLP, 2011, pp. 409–420.
[18] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, Advances in neural information processing systems 27 (2014).
[19] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473 (2014).
[20] S. Nisioi, S. Štajner, S. P. Ponzetto, L. P. Dinu, Exploring neural text simplification models, in: ACL, 2017, pp. 85–91.
[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).
[22] L. Martin, B. Sagot, E. de la Clergerie, A. Bordes, Controllable sentence simplification, arXiv preprint arXiv:1910.02677 (2019).
[23] 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.
[24] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: P. Isabelle, E. Charniak, D. Lin (Eds.), Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 2002, pp. 311–318. URL: https://aclanthology.org/P02-1040/. doi:10.3115/1073083.1073135.
[25] J. P. Kincaid, R. P. Fishburne Jr, R. L. Rogers, B. S. Chissom, Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel, Technical Report, 1975.

A. Best-Match Retrieval Diagnostics

We quantify how well relevance-aligned best-match retrieval identifies a closely matching demonstration at each granularity. For every evaluation instance, we score all candidates in the corresponding filtered pool with the same composite similarity used at run time (content 0.65, complexity 0.20, numeric 0.15) and record the top-ranked candidate. The diagnostic is computed over all evaluation instances against the full candidate pools (sentence: 3,679 instances, 7,061 candidates; document: 175 instances, 367 candidates); it is exact rather than sampled.

Table 8
Best-match retrieval discriminability under the composite similarity used for selection. "Lift over pool average" is the mean top-1 similarity relative to the mean similarity of all candidates for the same instance.

Quantity	Sentence	Document
Mean top-1 similarity	0.4474	0.3107
Top-1 range (min-max)	0.149-1.000	0.260-0.359
Top-1 IQR (p25-p75)	0.367-0.506	0.299-0.321
Mean pool-average similarity	0.2464	0.2600
Best-match lift over pool average	+81.6%	+19.5%

Table 9 decomposes the composite into its three components for the candidate selected by the composite score. For each component we report the value at the selected best match, the mean value across the candidate pool, and the best-over-pool lift, which indicates how much discriminative signal that component contributes.

Table 9
Component-wise breakdown of the composite best-match similarity (values for the composite-selected candidate)

Component	Sentence best	Sentence pool-avg	Sentence lift	Document best	Document pool-avg	Document lift
Lexical/content (Jaccard)	0.2567	0.0367	+599.5%	0.1566	0.0983	+59.3%
Complexity profile	0.9443	0.8780	+7.6%	0.9782	0.9598	+1.9%
Numeric overlap (Jaccard)	0.6115	0.3129	+95.4%	0.0889	0.0275	+223.9%

Two observations follow. First, the lexical/content component supplies most of the usable matching signal, and it is far weaker at document level: its best-over-pool lift drops from +599.5% (sentence) to +59.3% (document), and its absolute best-match value falls from 0.2567 to 0.1566. Second, the complexity component is near-saturated in both tasks (best similarity above 0.94 with small lifts of +7.6% and +1.9%), so it contributes little discrimination, particularly for documents where nearly all abstracts share a similarly high complexity profile. The numeric component has a large relative lift at document level (+223.9%) but a very small absolute value (best 0.0889), reflecting sparse numeric overlap across long abstracts. Together these components explain why document-level best-match selection is close to uninformative: the one strongly discriminative signal (lexical overlap) degrades over long, heterogeneous inputs, and the remaining components cannot compensate.

B. Output-Based Diagnostics

These diagnostics are computed directly on the run outputs (source and prediction text), independent of the official scorer, to localize the mechanisms discussed in Section 5.

Table 10
FKGL driver decomposition for document factor-isolation runs. We report words per sentence and syllables per word to isolate which component drives readability differences; official FKGL values are reported in Table 6

Run	Words/sentence	Syllables/word
Direct FS3 + random pooled	15.90	1.661
Terminology-only zero-shot	13.11	1.689
Structure-only zero-shot	10.06	1.683

Table 11
Survival of mined jargon source terms in outputs: the fraction of source occurrences of the guidance's jargon-substitution complex terms that still appear in the output. Lower means more targeted terms were removed.

Level	Run	Jargon survival
Document	Non-guided FS3 + random pooled	0.547
Document	Structured-guidance FS3 + random pooled	0.251
Sentence	Non-guided FS3 + best-match	0.572
Sentence	Prompt-guided FS3 + best-match	0.300
Sentence	Structured-guidance FS3 + best-match	0.233

C. Mined Simplification Patterns

Stage 1 of the guidance pipeline (Section 3.5) mines the gold training pairs with a deterministic script and writes a JSON/Markdown artifact. Tables 12–14 summarize the components of that artifact that feed the Stage 2 synthesis prompt; the full artifact is released with our code.

Table 12
Stage 1 mined corpus-level statistics from the gold training pairs (sentence vs. document level).

Statistic	Sentence pairs	Document pairs
Training pairs	7,082	849
Mean compression ratio	1.060	0.593
Median compression ratio	1.000	0.568
Numeric-token preservation rate	0.721	0.080
Mean numeric-token overlap	0.752	0.233
Sentence-splitting rate	0.012	0.008
Mean long-word reduction (tokens)	1.02	40.95

Table 13
Top candidate jargon→plain substitutions mined by token-set differencing, with support counts. Document-level candidates are noisier because whole-abstract differencing frequently pairs unrelated co-dropped and co-added tokens.

Sentence pairs		Document pairs	
Substitution	**Count**	**Substitution**	**Count**
participants → people	140	confidence → found	102
included → found	116	adverse → side	67
adverse → side	90	reported → found	54
mortality → death	57	difference → found	54
studies → study	41	outcomes → found	51
identified → found	40	participants → found	49

Table 14
Stage 1 mined hedge/uncertainty preservation rates in the gold training pairs: the fraction of source occurrences of each phrase retained in the reference simplification.

Hedge phrase	Sentence	Document
may	0.66	0.63
probably	0.66	0.71
uncertain	0.67	0.62
could	0.53	0.42
unclear	0.49	0.30
no evidence	0.57	0.52
little or no	0.63	0.62
low-certainty evidence	0.17	0.21
very low-certainty evidence	0.14	0.12

D. Prompt Templates

D.1. Guidance Synthesis Prompt

Stage 2 converts the mined artifact (Appendix C) into structured guidance with a single LLM call. The system prompt is shown in GSYN; the paired user message contains the compact mined artifact as JSON (corpus statistics, top dropped/added tokens, candidate jargon substitutions, hedge-preservation rates, and up to nine sampled complex-to-simple pairs), and `{level}` is instantiated as either `sentence` or `document`. The `rules`, `jargon_substitutions`, and `must_preserve` fields of the returned JSON populate the guidance overlays below.

GSYN - System prompt used in Stage 2 to synthesize structured guidance from the mined pattern artifact

```
You are an expert in plain-language medical writing. Given mined statistics and example
    complex->simple sentence pairs from a medical simplification corpus, you must produce
    concrete, actionable guidance that a downstream LLM can follow when simplifying medical
    text for a general audience. This guidance is for the {level}-level task. Your output
    MUST be a single valid JSON object with these fields:
  rules: array of short imperative bullet strings (8-15 items).
  jargon_substitutions: array of objects {complex, simple} for safe lexical swaps.
  must_preserve: array of strings describing what must always be preserved.
  system_prompt: a concise system prompt (<= 220 words) suitable for a medical text
      simplification model, integrating the rules above.
Do not include any text outside the JSON object.
```

D.2. Guidance Overlays

G1S - Prompt-guidance overlay used in sentence-level prompt-guided runs

```
Additional simplification guidance:
- Use short, clear sentences.
- Replace medical jargon with common words when possible.
- Keep important numbers and statistics.
- Preserve the main findings and outcomes.
- Keep uncertainty or hedging language (e.g., may, probably) when present.
- Remove unnecessary details about study design unless essential for understanding.
- Summarize long or complex lists into simpler forms.
- Avoid technical terms for study quality unless needed for meaning.
- Use 'people' instead of 'participants' when appropriate.
- Use 'side effects' instead of 'adverse events.'
- Use 'death' or 'deaths' instead of 'mortality.'
- Do not add new information not present in the original sentence.
```

```
- Keep the meaning and intent of the original sentence.
- Use active voice when possible.
- Avoid repeating information.
```

G1D - Prompt-guidance overlay used in document-level prompt-guided runs

```
Additional simplification guidance:
- Summarize the main findings and key points in clear, simple language.
- Reduce document length by omitting unnecessary details and technical explanations.
- Avoid or replace complex medical jargon with common, everyday words.
- Use short sentences and simple sentence structures.
- Explain or omit statistical terms and technical phrases unless essential for understanding.
- Replace or remove words like 'confidence interval', 'adverse', and 'outcomes' with simpler
    alternatives or explanations.
- Use concrete terms such as 'side effects' instead of 'adverse events', and 'death' instead
    of 'mortality'.
- Focus on what was found, who was studied, and the main results.
- Preserve hedging language (e.g., 'may', 'probably', 'uncertain') when it is important for
    conveying uncertainty.
- Remove or simplify references to study design details unless critical for understanding.
- Use words like 'people' instead of 'participants' and 'treatment' instead of 'intervention'
    where appropriate.
- Avoid including raw numbers or statistics unless they are essential for understanding the
    main message.
- Do not introduce new information or interpretations not present in the original text.
- Maintain the original meaning and intent of the document.
- Ensure the summary is accessible to a general audience with no medical background.
```

G2S - Structured-guidance overlay used in sentence-level structured-guidance runs

```
Additional simplification guidance:
- Use short, clear sentences.
- Replace medical jargon with common words when possible.
- Keep important numbers and statistics.
- Preserve the main findings and outcomes.
- Keep uncertainty or hedging language (e.g., may, probably) when present.
- Remove unnecessary details about study design unless essential for understanding.
- Summarize long or complex lists into simpler forms.
- Avoid technical terms for study quality unless needed for meaning.
- Use 'people' instead of 'participants' when appropriate.
- Use 'side effects' instead of 'adverse events.'
- Use 'death' or 'deaths' instead of 'mortality.'
- Do not add new information not present in the original sentence.
- Keep the meaning and intent of the original sentence.
- Use active voice when possible.
- Avoid repeating information.

Must preserve:
- Main medical finding or outcome
- Key numbers and statistics
- Uncertainty or hedging language (e.g., may, probably, unclear)
- Any mention of side effects or harms
- The overall meaning and intent of the original sentence

Preferred jargon substitutions:
- participants -> people
- adverse -> side
- adverse events -> side effects
- mortality -> death
- mortality -> deaths
- included -> found
```

```
- identified -> found
- studies -> study
- intervention -> treatment
- outcomes -> results
- Apply substitutions only when meaning remains unchanged and context is appropriate.
```

G2D - Structured-guidance overlay used in document-level structured-guidance runs

```
Additional simplification guidance:
- Summarize the main findings and key points in clear, simple language.
- Reduce document length by omitting unnecessary details and technical explanations.
- Avoid or replace complex medical jargon with common, everyday words.
- Use short sentences and simple sentence structures.
- Explain or omit statistical terms and technical phrases unless essential for understanding.
- Replace or remove words like 'confidence interval', 'adverse', and 'outcomes' with simpler
    alternatives or explanations.
- Use concrete terms such as 'side effects' instead of 'adverse events', and 'death' instead
    of 'mortality'.
- Focus on what was found, who was studied, and the main results.
- Preserve hedging language (e.g., 'may', 'probably', 'uncertain') when it is important for
    conveying uncertainty.
- Remove or simplify references to study design details unless critical for understanding.
- Use words like 'people' instead of 'participants' and 'treatment' instead of 'intervention'
    where appropriate.
- Avoid including raw numbers or statistics unless they are essential for understanding the
    main message.
- Do not introduce new information or interpretations not present in the original text.
- Maintain the original meaning and intent of the document.
- Ensure the summary is accessible to a general audience with no medical background.

Must preserve:
- The main findings and conclusions of the document.
- Any important hedging or uncertainty about the results.
- The overall meaning and intent of the original text.
- Essential information about who was studied and what was found.

Preferred jargon substitutions:
- confidence -> found
- interval -> found
- included -> found
- adverse -> side
- reported -> found
- difference -> found
- outcomes -> found
- evidence -> found
- participants -> people
- mortality -> death
- Apply substitutions only when meaning remains unchanged and context is appropriate.
```

D.3. Task 1.1: Sentence-Level Prompts

S1 - Legacy sentence-only zero-shot prompt

```
System prompt:
You are a helpful assistant.
User prompt:
### TASK ###
Simplify the language used in this sentence from a scientific article so that it can be
    understood by the general audience.
```

```
Focus on simplifying the sentence structure and replacing scientific jargon with everyday
    language.
### REQUEST ###
- Sentence:
{sentence}
- Simplified Sentence:
```

S2 - Enhanced sentence-only baseline (v1, zero-shot).

```
System prompt:
You simplify biomedical sentences for non-expert readers.
Rules:
1) Preserve meaning and conclusions.
2) Preserve all numbers and technical qualifiers (may, likely, probably, little to no
    difference).
3) Do not add new facts.
4) Keep it concise and clear.
Return only the simplified sentence.

User prompt:
Target sentence:
{sentence}

Simplified sentence:
```

S3 - Enhanced sentence-only zero-shot (v3).

```
System prompt:
# Objective
Simplify one biomedical sentence for non-expert readers.

# Hard Constraints
1) Preserve all factual meaning.
2) Preserve all numbers, ranges, and statistical values in meaning.
3) Preserve uncertainty qualifiers exactly in meaning (e.g., may, likely, probably, little to
    no difference).
4) Do not add, infer, or speculate.
5) Keep wording plain and concise.

# Style
- Prefer one to two short sentences.
- Replace jargon where possible without changing meaning.

# Output Format
- Return only the simplified sentence.
- No bullets, headings, labels, or explanations.

User prompt:
Target sentence:
{sentence}

Simplified sentence:
```

S4 - Sentence-only few-shot (FS3; used with random pooled or best-match retrieval).

```
System prompt:
You simplify biomedical sentences for non-expert readers.
Follow instructions literally.
Rules:
```

```
1) Preserve meaning and conclusions.
2) Preserve all numbers and uncertainty qualifiers (may, likely, probably, little to no
    difference).
3) Do not add new facts.
4) Prefer short clear sentences and plain wording.
Return only the simplified sentence.

User prompt:
Few-shot examples:
Example 1
Complex: {ex1.complex}
Simple: {ex1.simple}

Example 2
Complex: {ex2.complex}
Simple: {ex2.simple}

Example 3
Complex: {ex3.complex}
Simple: {ex3.simple}

Target sentence:
{sentence}

Simplified sentence:
```

S5 - Sentence-with-context baseline (zero-shot).

```
System prompt:
You simplify one biomedical sentence for non-expert readers.
You are given full-document context only to improve consistency.
Rules:
1) Preserve meaning and conclusions of the target sentence.
2) Preserve all numbers and technical qualifiers (may, likely, probably, little to no
    difference).
3) Do not add new facts.
4) Keep it concise and clear.
Return only the simplified sentence.

User prompt:
Document context:
{document_context}

Target sentence to simplify:
{sentence}

Simplified sentence:
```

S6 - Sentence-with-context few-shot.

```
System prompt:
You simplify one biomedical sentence for non-expert readers.
You are given document context and few-shot examples.
Follow instructions literally.
Rules:
1) Preserve meaning and conclusions of the target sentence.
2) Preserve all numbers and uncertainty qualifiers (may, likely, probably, little to no
    difference).
3) Do not add new facts.
4) Keep language concise and clear.
Return only the simplified sentence.
```

```
User prompt:
Few-shot examples:
Example 1
Complex: {ex1.complex}
Simple: {ex1.simple}
...

Document context:
{document_context}

Target sentence to simplify:
{sentence}

Simplified sentence:
```

D.4. Task 1.2: Document-Level Direct Prompts

D1 - Legacy direct zero-shot.

```
System prompt:
You are a helpful assistant.

User prompt:
### TASK ####
You are going to simplify a given abstract intended for an academic audience to a text that is
     understandable to the general audience.
### INSTRUCTIONS ####
1. Identify Key Concepts ...
2. Simplify Language ...
3. Break Down Complex Ideas ...
4. Avoid Jargon ...
### REQUEST ###
Remember: the goal is not to oversimplify or distort the scientific abstract, but to make it
    accessible and understandable
to more people.
- Abstract:
{document}
- Simplified abstract:
```

D2 - Legacy direct few-shot (FS3).

```
System prompt:
You are a helpful assistant.

User prompt:
<same legacy task/instructions as D1>

### EXAMPLE 1 ###
- Abstract:
{ex1.complex}
- Simplified Abstract:
{ex1.simple}

### EXAMPLE 2 ###
- Abstract:
{ex2.complex}
- Simplified Abstract:
{ex2.simple}
```

```
### REQUEST ###
- Abstract:
{document}
- Simplified abstract:
```

D3 - Enhanced direct baseline (v1, zero-shot).

```
System prompt:
You simplify full biomedical abstracts for non-expert readers.
Rules:
1) Preserve meaning, key findings, and conclusions.
2) Preserve numbers and uncertainty qualifiers (may, likely, probably, little to no difference
    ).
3) Do not add new facts.
4) Use short clear sentences and plain language.
Return only the simplified document text.

User prompt:
Original abstract:
{document}

Simplified abstract:
```

D4 - Enhanced direct zero-shot (v3).

```
System prompt:
# Objective
Simplify a full biomedical abstract into plain language.

# Hard Constraints
1) Preserve key findings and conclusions.
2) Preserve all numeric/statistical content in meaning.
3) Preserve uncertainty qualifiers in meaning.
4) Do not add unsupported claims.

# Style
- Split long sentences when helpful.
- Use plain wording and concise phrasing.

# Output Format
- Return only the simplified abstract text.
- No headings, bullet lists, or explanations.

User prompt:
Original abstract:
{document}

Simplified abstract:
```

D5 - Enhanced direct few-shot (FS3).

```
System prompt:
You simplify full biomedical abstracts for non-expert readers.
Follow instructions literally.
Rules:
1) Preserve meaning, key findings, and conclusions.
2) Preserve numbers and uncertainty qualifiers (may, likely, probably, little to no difference
    ).
```

```
3) Do not add new facts.
4) Use short clear sentences and plain language.
Return only the simplified abstract text.

User prompt:
Few-shot examples:
Example 1
Complex: {ex1.complex}
Simple: {ex1.simple}
...

Original abstract:
{document}

Simplified abstract:
```

D.5. Task 1.2: Decomposed Plan-Based Prompts

D6 - Structure-plan stage.

```
System prompt (v3):
# Objective
Plan structural simplification for a biomedical abstract.

# Required Output
Return concise bullets covering exactly:
1) Sentences to split/reorder/compress.
2) Information order to preserve.
3) Numeric/statistical items that must be preserved.
4) Uncertainty terms that must be preserved.

# Constraints
- Do not simplify or rewrite the abstract.
- Do not add content not present in the abstract.

User prompt:
### TASK ###
You are preparing a STRUCTURE PLAN for simplifying one biomedical abstract for non-expert
    readers.
...
### OUTPUT FORMAT ###
Return ONLY valid JSON:
{
  "structure_actions": [ ... ],
  "global_constraints": [ ... ]
}
```

D7 - Terminology-plan stage.

```
System prompt (v3):
# Objective
Create a terminology aid for simplifying a biomedical abstract.

# Required Output
Return concise bullets; each bullet includes:
- term/acronym
- one-sentence plain-language definition
- simple phrasing to use in final rewrite
```

```
# Constraints
1) Keep definitions generally accepted and conservative.
2) Do not add document-specific claims not in source.
3) Keep entries brief.

User prompt:
### TASK ###
You are preparing a TERMINOLOGY PLAN for simplifying one biomedical abstract for non-expert
    readers.
...
### OUTPUT FORMAT ###
Return ONLY valid JSON:
{
  "glossary": [ ... ]
}
```

D8 - Decomposed final merge, zero-shot.

```
System prompt (v3):
# Objective
Produce the final simplified biomedical abstract using structure and terminology guidance.

# Hard Constraints
1) Preserve findings and conclusions.
2) Preserve numbers/statistics and uncertainty qualifiers in meaning.
3) Do not add or infer new facts.

# Style
- Plain, concise language.
- Split long sentences where useful.

# Output Format
- Return only the simplified abstract text.
- No headings, bullets, or explanations.

User prompt:
### INPUT ABSTRACT ###
{document}
### VALIDATED STRUCTURE PLAN JSON ###
{structure_points_json}
### VALIDATED TERMINOLOGY PLAN JSON ###
{terminology_notes_json}
...
Simplified abstract:
```

D9 - Decomposed final merge, few-shot (FS3).

```
System prompt:
You simplify a biomedical abstract using provided guidance and few-shot examples.
Follow instructions literally.
Rules:
1) Preserve meaning, key findings, and conclusions.
2) Preserve numbers and uncertainty qualifiers (may, likely, probably, little to no difference
    ).
3) Do not add new claims.
4) Keep language plain and concise.
Return only the simplified abstract text.

User prompt:
<same as D8, plus>
### FEW-SHOT EXAMPLES ###
```

```
Example 1
Complex: {ex1.complex}
Simple: {ex1.simple}
...
```

D.6. Task 1.2: Decomposed Dual-Draft and Factor-Isolation Prompts

D10 - Dual terminology draft (also used for terminology-only factor isolation).

```
System prompt:
You simplify biomedical terminology in a full abstract for non-expert readers.
Do not perform major structural rewrites.
Rules:
1) Preserve findings, numbers/statistics, and uncertainty qualifiers in meaning.
2) Replace jargon/acronyms with plain wording where possible.
3) Keep sentence boundaries and information order mostly unchanged.
4) Do not add claims or interpretation.
Return only the terminology-simplified abstract text.

User prompt:
### TASK ###
Rewrite this abstract by simplifying terminology only.
...
Terminology-simplified abstract:
```

D11 - Dual structure draft (also used for structure-only factor isolation).

```
System prompt:
You simplify sentence structure in a full biomedical abstract for non-expert readers.
Do not aggressively simplify terminology.
Rules:
1) Preserve findings, numbers/statistics, and uncertainty qualifiers in meaning.
2) Improve readability via splitting/reordering/compressing sentence structure.
3) Keep technical terms when possible; focus on structure, not glossary rewriting.
4) Do not add claims or interpretation.
Return only the structure-simplified abstract text.

User prompt:
### TASK ###
Rewrite this abstract by simplifying sentence structure only.
...
Structure-simplified abstract:
```

D12 - Dual final merge, zero-shot.

```
System prompt:
You produce the final simplified biomedical abstract from three inputs:
- original abstract
- terminology-simplified draft
- structure-simplified draft

Rules:
1) Preserve all findings, numbers/statistics, and uncertainty qualifiers in meaning.
2) Use the two drafts only as guidance; resolve conflicts conservatively toward the original.
3) Do not add claims, causes, recommendations, or interpretation.
4) Output one clean final abstract in plain language.
Return only the final simplified abstract text.
```

```
User prompt:
### ORIGINAL ABSTRACT ###
{document}
### TERMINOLOGY-SIMPLIFIED DRAFT ###
{terminology_simplified}
### STRUCTURE-SIMPLIFIED DRAFT ###
{structure_simplified}
...
Final simplified abstract:
```

D13 - Dual final merge, few-shot.

```
System prompt:
You produce the final simplified biomedical abstract from:
- the original abstract
- a terminology-simplified draft
- a structure-simplified draft

Rules:
1) Preserve findings, numbers/statistics, and uncertainty qualifiers in meaning.
2) Use few-shot examples for style only, never for content.
3) Resolve conflicts conservatively toward the original abstract.
4) Do not add claims or interpretation.
Return only the final simplified abstract text.

User prompt:
<same as D12, plus few-shot examples block>
```

D14 - Structure-only factor-isolation output.

```
D14 uses the same prompt pair as D11, but the structure draft is treated as the final output.
```