HULAT-UC3M at CLEF 2026 SimpleText: Controlled Generation for Scientific Text Simplification with Hallucination Control

SimpleText at CLEF 2026

Pablo Anegón Núñez[1], Paloma Martínez[1] and Lourdes Moreno[1]

[1]*Universidad Carlos III de Madrid, Computer Science and Engineering Department, Av. de la Universidad 30, 28911 Leganés, Madrid, Spain*

Abstract

This paper describes the participation of HULAT-UC3M in the CLEF 2026 SimpleText track on scientific text simplification. We address sentence-level simplification (Task 1.1), document-level simplification (Task 1.2), and controlled generation (Task 2.3). The implemented system follows a shared controlled-generation architecture with two model variants: `Qwen2.5-3B-Instruct` fine-tuned with QLoRA and `FLAN-T5 Large` fine-tuned as a sequence-to-sequence model. Both variants use task-specific prompts with explicit factual constraints and a threshold-based mechanism that detects semantic, factual, syntactic, lexical, and numerical risks.

For the submitted sentence-level run reported in this paper, the official Codabench evaluation obtained a SARI of 42.79 and BLEU of 5.72 against the `simple_original` references, and a SARI of 40.49 and BLEU of 4.30 against `simple_auto`. The FKGL scores were 12.80 and 12.98, respectively. The output contained no exact source copies and achieved sentence-split ratios above 1.24, indicating consistent rewriting and sentence segmentation. Paired normal-versus-controlled Task 2.3 scores are not reported because they were not included in the available evaluation artifacts.

Keywords

scientific text simplification, controlled generation, hallucination control, faithfulness, CLEF 2026 SimpleText, large language models

1. Introduction

Access to scientific information has expanded dramatically, but having a document available is not the same as being able to understand it. Scientific and, in particular, biomedical texts are written for expert readers: they rely on technical vocabulary, long sentences, and a high density of information packed into few words [1, 2]. Automatic text simplification aims to reduce this barrier by rewriting a complex text into a clearer version while preserving its meaning [3].

Modern simplification systems are increasingly based on generative models built on the Transformer architecture [4]. These models produce fluent rewrites, but the same flexibility brings a well-known risk: they can generate output that looks correct yet is not faithful to the source. This phenomenon is usually referred to as hallucination or factual distortion [5]. In scientific text the tolerance for such errors is very low: changing a number, dropping a negation, removing a condition such as "only if", or turning "may reduce" into "reduces" can change the technical meaning of a sentence entirely.

The CLEF 2026 SimpleText track reflects this concern directly: besides scientific text simplification, it includes a controlled-creativity line explicitly aimed at identifying and avoiding hallucinated or under-supported content [1]. Our work follows this view. Rather than accepting the first generated output, we treat simplification as a controlled process in which an evaluator inspects each candidate and a retry mechanism regenerates it when faithfulness or quality signals are not met.

In this paper we describe our participation in three subtasks: sentence-level simplification (Task 1.1), document-level simplification (Task 1.2), and a controlled-generation comparison (Task 2.3). Our main contributions are: (i) a generation-plus-evaluation pipeline with explicit faithfulness signals and

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

pablo.anegon@alumnos.uc3m.es (P. A. Núñez); pmf@inf.uc3m.es (P. Martínez); lmoreno@inf.uc3m.es (L. Moreno)

threshold-based routing; (ii) two generator variants, based on `Qwen2.5-3B-Instruct` with QLoRA and `FLAN-T5 Large` with sequence-to-sequence fine-tuning, driven by task-specific prompts with explicit factual constraints; and (iii) an analysis of the trade-off between simplification and meaning preservation.

2. Related Work

Automatic text simplification has moved from rule-based and statistical methods to neural sequence-to-sequence models and, more recently, large language models [6, 7, 8]. Encoder–decoder models such as BART [8] and T5 [7] are well suited to rewriting because they condition the output on an input sequence; instruction-tuned variants such as FLAN-T5 [9] extend this further. Earlier work by our group already explored abstractive simplification for SimpleText using BART [10].

Evaluation remains a central difficulty. SARI was proposed specifically for simplification, rewarding correct add/keep/delete operations [3], while BLEU [11] and ROUGE [12] only measure surface overlap. BERTScore estimates semantic similarity through contextual embeddings [13], and MeaningBERT targets meaning preservation between sentences [14]. Several studies show that automatic metrics may penalise valid reformulations and do not capture whether a text is actually clearer or faithful [15, 16]. The problem of hallucination in generation has been surveyed extensively and is particularly relevant for scientific content [5]. Recent work on accessible text generation also stresses the need to combine automatic signals with control and human-oriented validation [17].

3. Task and Data

CLEF SimpleText is a shared task focused on improving access to scientific information [1]. The scientific-simplification data builds on the Cochrane-auto corpus, derived from biomedical abstracts and their plain-language summaries from Cochrane systematic reviews, with alignments at sentence, paragraph and document level [2, 1].

Table 1
Reported dataset split used for Task 1.

Item	Value
Training sentences	9,939
Training documents	849
Validation	591
Test sentences	48809
Test documents	8069

4. System Description

4.1. Overview

Our system is organised around two main components: a *generator*, which produces candidate simplifications, and an *evaluator*, which inspects each candidate and decides whether it is accepted or sent back for regeneration. Figure 1 summarises the pipeline.

4.2. Base model and fine-tuning

The system was implemented with two generator variants following the same controlled-generation architecture. The main submitted configuration uses `Qwen/Qwen2.5-3B-Instruct`, a decoder-only

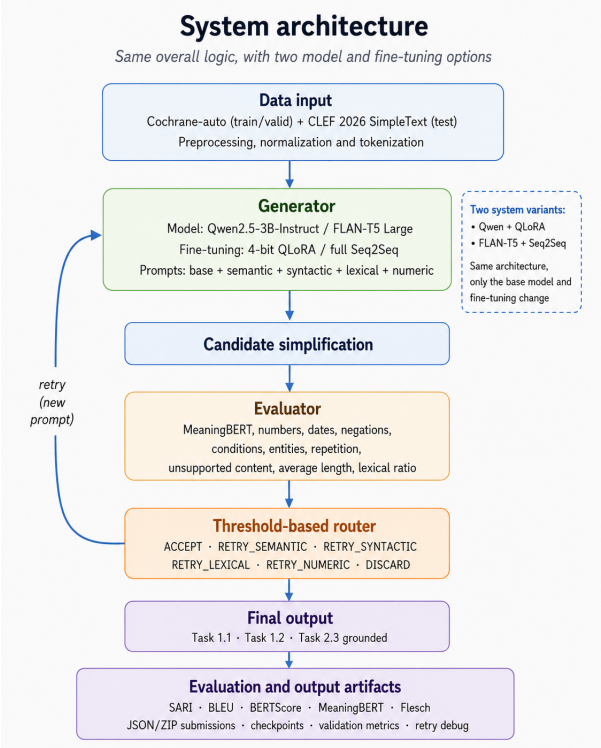

Figure 1: Shared controlled-generation architecture used by the Qwen+QLoRA and FLAN-T5+Seq2Seq variants.

causal language model loaded through `AutoModelForCausalLM`. The task is framed as instruction-following generation: given a complex scientific input and an instruction, the model produces a simpler version. This model is fine-tuned with QLoRA: the base model is loaded with 4-bit NF4 quantisation and low-rank adapters are trained over the attention and feed-forward projection layers. After training, the adapter is merged into a bfloat16 copy of the base model for faster inference.

A second internal variant uses `FLAN-T5 Large`, an encoder–decoder sequence-to-sequence model. This variant is fine-tuned as a standard text-to-text generation model and is used as an internal comparison with the Qwen-based submission. Both variants share the same general pipeline: candidate generation, automatic evaluation, threshold-based routing, and retry with task-specific prompts.

During Qwen fine-tuning, each training instance is expanded across the prompt templates used by the controlled pipeline: base, semantic, syntactic, lexical, and numeric. This allows the model to learn not only the initial plain-language rewriting instruction, but also the specialised retry instructions used when factual, syntactic, lexical, or numerical issues are detected. The loss is computed only over the target simplification, while the system and user prompt tokens are masked.

4.3. Prompting strategies

We use several prompt templates that condition the type of simplification: a *base* prompt for plain-language biomedical rewriting with explicit factual constraints, a *semantic* prompt for meaning preservation, a *syntactic* prompt for sentence restructuring, a *lexical* prompt for vocabulary simplification, and a *numeric* prompt for cases in which numbers, percentages, dates, or statistical values are at risk of being changed. For document-level inputs, the base and retry prompts are augmented with document-level requirements aimed at preserving coherence, order of ideas, numerical comparisons, limitations, and uncertainty.

4.4. Evaluator and faithfulness signals

The evaluator combines reference-based metrics (SARI, BLEU, BERTScore, MeaningBERT) with reference-free signals designed to catch frequent errors in scientific simplification: loss of negation; preservation of numbers and dates; preservation of conditions; preservation of entities; unsupported-content ratio; repeated-n-gram ratio; sentence length and split ratio; and difficult-word ratio.

The local faithfulness signals and retry decisions are applied to English items, where the heuristics for negation, entities, unsupported tokens, and readability are calibrated. All input items are nevertheless preserved in the submission JSON files. Non-English items are handled conservatively to avoid uncontrolled translation or factual distortion.

4.5. Thresholds and routing

Each candidate is routed according to threshold-based Event–Condition– Action rules. The main decisions are ACCEPT, RETRY_SEMANTIC, RETRY_SYNTACTIC, RETRY_LEXICAL, RETRY_NUMERIC, and DISCARD. Table 2 reports the thresholds assigned in the executable code. Some syntactic and lexical conditions are combined rather than applied independently; these combinations are stated in the action column.

Table 2
Routing thresholds of the retry loop, as implemented in the executable code.

Signal	Accepted criterion	Routing if violated
Repeated 4-gram ratio	≤ 0.35	DISCARD
Non-Latin script detected	false	DISCARD
MeaningBERT	≥ 70.0	RETRY_SEMANTIC
Negation loss	$= 0$	RETRY_SEMANTIC
Number/date recall	≥ 0.95	RETRY_NUMERIC
Condition recall	≥ 0.90	RETRY_SEMANTIC
Entity recall	≥ 0.90	RETRY_SEMANTIC
Compression ratio (sentence)	0.40–1.20	Below: RETRY_SEMANTIC; above: RETRY_SYNTACTIC
Compression ratio (document)	0.60–1.10	Below: RETRY_SEMANTIC; above: RETRY_SYNTACTIC
Mean sentence length (sentence)	≤ 20 words	RETRY_SYNTACTIC when the split ratio is also < 1.10 or punctuation complexity is > 0.50
Mean sentence length (document)	≤ 22 words	RETRY_SYNTACTIC when punctuation complexity is also > 0.50
Lexical simplification (sentence)	≥ 0.05	RETRY_LEXICAL when difficult-word ratio is also > 0.25
Lexical simplification (document)	≥ 0.04	RETRY_LEXICAL when difficult-word ratio is also > 0.25

Table 3
Strict faithfulness filter used for the Task 2.3 grounded run. If any criterion fails, the source text is returned instead of the simplification.

Signal	Accepted criterion
MeaningBERT	≥ 45.0
Negation loss	$= 0$
Number/date recall	$= 1.0$
Entity recall	≥ 0.90
Repeated 4-gram ratio	≤ 0.30
Non-Latin script detected	false
Unsupported-token ratio	≤ 0.30

Table 4
Main hyperparameters used for Qwen2.5 fine-tuning and generation.

Parameter	Value
Base model	`Qwen/Qwen2.5-3B-Instruct`
Model architecture	Decoder-only causal language model
Fine-tuning method	QLoRA with 4-bit NF4 quantisation
Prompt expansion during fine-tuning	Base, semantic, syntactic, lexical, and numeric prompts
Quantisation configuration	NF4, double quantisation, bfloat16 computation
Training epochs	3
Learning rate	2×10^{-4}
Train micro-batch size	1
Gradient accumulation steps	32
Effective train batch size	32
Sentence inference batch size	16
Document inference batch size	4
Weight decay	0.01
Warm-up ratio	0.03
Learning-rate scheduler	Cosine
Optimiser	8-bit AdamW
LoRA rank (r)	32
LoRA alpha	64
LoRA dropout	0.05
LoRA target modules	`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj`
Sentence input / output limit	512 / 160 tokens
Document input / output limit	2048 / 1536 tokens
Beam size	1
Sampling / temperature	Disabled / 1.0
Repetition penalty	1.08; 1.10 for semantic retries; document 1.03
No-repeat n-gram size	3; 4 for semantic retries; 0 for documents
Maximum attempts	3
Validation split	5% of the training data
Precision during training	bfloat16 with gradient checkpointing
Inference configuration	LoRA adapter merged into the base model in bfloat16
Random seed	42
Hardware	NVIDIA GeForce RTX 4070 Ti

4.6. Retry mechanism

When a candidate fails its criteria, the system regenerates it with the prompt matching the detected problem, up to a maximum of three attempts. All candidates, signals, decisions, and selected outputs are logged for traceability.

5. Experimental Setup

A single submitted configuration is run: the QLoRA-fine-tuned model uses the base plain-language prompt for the initial generation, while the controlled pipeline can route problematic English candidates to semantic, syntactic, lexical, or numeric retry prompts. The same predictions are reused to build the Task 2.3 grounded run, in which any candidate that fails the strict faithfulness filter (Table 3) is replaced by the source text. Generation and fine-tuning use Hugging Face Transformers [18] and PyTorch [19].

The simplification model is based on `Qwen/Qwen2.5-3B-Instruct`, adapted to the scientific text simplification task using QLoRA. During fine-tuning, the base model is loaded using 4-bit NF4 quantisation, while low-rank adapters are trained over the attention and feed-forward projection layers. After training, the LoRA adapter is merged into a bfloat16 version of the base model to improve inference

speed. Training uses a micro-batch size of one and 32 gradient accumulation steps, resulting in an effective batch size of 32 on a single GPU.

6. Results

Codabench returned two official evaluations for the same submitted sentence-level prediction file. The two rows differ only in the reference set used by the scorer: `simple_original` and `simple_auto`. They must therefore not be interpreted as two separate model configurations. Tables 5 and 6 report the complete official output.

Table 5
Official Codabench results for the submitted sentence-level run.

Reference set	BLEU	SARI	FKGL	Compression	Sentence splits	Levenshtein sim.	Exact copies
`simple_original`	5.7211	**42.7868**	12.7994	1.0410	1.2423	0.4781	0.0000
`simple_auto`	4.3000	40.4882	12.9775	1.0149	1.2673	0.4789	0.0000

Table 6
Editing behaviour reported by Codabench for the sentence-level submitted run.

Reference set	Additions proportion	Deletions proportion	Lexical complexity
`simple_original`	0.4796	0.5371	8.5927
`simple_auto`	0.4642	0.5435	8.6346

The best official score is obtained against `simple_original`, with a SARI of 42.79, which is 2.30 points higher than the score obtained against `simple_auto`. BLEU follows the same pattern, increasing from 4.30 to 5.72. In contrast, readability is very stable across the two reference sets: FKGL is 12.80 with `simple_original` and 12.98 with `simple_auto`.

The compression ratios of 1.0410 and 1.0149 show that the generated text is, on average, slightly longer than the source rather than aggressively compressed. Sentence-split ratios above 1.24 indicate that the system regularly divides complex sentences into multiple simpler sentences. Moreover, the exact-copy rate is zero and Levenshtein similarity remains close to 0.48 under both evaluations, confirming that the system does not merely reproduce the source. The additions and deletions proportions also show substantial rewriting, while their similarity across reference sets suggests that the observed difference in SARI is mainly caused by the choice of reference rather than a change in system behaviour.

The separate validation artifact supplied during development contains one aggregate sentence-level result and one aggregate document-level result for the fine-tuned model. These internal results are reported below for diagnostic purposes and must not be confused with the official Codabench scores. BLEU and BERTScore in these tables use the $[0, 1]$ scale returned by the local evaluation implementation.

Table 7
Internal Task 1.1 sentence-level validation results.

Config.	SARI	BLEU	BERTScore F1	MeaningBERT	Flesch
Provided validation run	32.6259	0.0369	0.8794	53.2699	29.7183

The internal document-level validation run has slightly higher SARI and MeaningBERT than the internal sentence-level run, but much lower Flesch readability and lower BERTScore F1. Paired normal

Table 8
Internal Task 1.2 document-level validation results.

Config.	SARI	BLEU	BERTScore F1	MeaningBERT	Flesch
Provided validation run	33.6910	0.0464	0.8452	61.3273	11.6903

Table 9
Reference-free diagnostic metrics from the internal validation artifact.

Level	Mean sentence length	Split ratio	Lexical simplification	Difficult-word ratio
Sentence	21.7479	1.0319	0.0265	0.5181
Document	20.3752	0.3876	0.0113	0.5250

Table 10
Contextual comparison between the official Qwen submission and the FLAN-T5 previous internal experiment. Results are reported on different evaluation settings and are therefore not directly comparable.

Model	Evaluation setting	Level	SARI	BLEU	MeaningBERT	Readability
Qwen2.5-3B + QLoRA	Official CLEF test, `simple_original`	Sent.	42.79	5.72	–	12.80 FKGL
Qwen2.5-3B + QLoRA	Official CLEF test, `simple_auto`	Sent.	40.49	4.30	–	12.98 FKGL
FLAN-T5 + Seq2Seq	Previous internal experiment, exp3	Sent.	38.17	–	76.10	42.80 Flesch
FLAN-T5 + Seq2Seq	Previous internal experiment, exp3	Doc.	36.22	–	76.88	42.29 Flesch

and controlled metrics required for Task 2.3 were not included in the available artifacts; consequently, no Task 2.3 numerical comparison is reported here.

Table 10 provides a contextual comparison between the official Qwen2.5-3B + QLoRA submission and the FLAN-T5 internal experiment. The FLAN-T5 rows correspond to the fine-tuned model with the base/transformative prompt. This comparison must be interpreted with caution: the Qwen results come from the official CLEF 2026 SimpleText test evaluation, whereas the FLAN-T5 results come from a previous internal experimental split. Therefore, the table is not intended as a strict leaderboard comparison, but as an overview of the behaviour of both model variants under their respective evaluation settings. In addition, FKGL and Flesch are different readability metrics: lower FKGL values and higher Flesch values indicate easier text.

7. Discussion

Several observations follow from the design and from the comparison between runs. First, the metrics must be read jointly: a high similarity score may simply mean the output barely changed the source, while a very high simplification score may hide loss of relevant information. This is why we combine SARI with semantic and faithfulness signals rather than optimising a single number.

Second, the official Codabench results show that evaluation depends substantially on the selected reference set. The same output changes from a SARI of 42.79 with `simple_original` to 40.49 with `simple_auto`, despite identical predictions. At the same time, the zero exact-copy rate, Levenshtein similarity near 0.48, and sentence-split ratio above 1.24 indicate that the model performs active rewriting instead of copying. The internal validation results additionally reveal a level-dependent trade-off: the document-level run has slightly higher SARI and MeaningBERT than the sentence-level run, but much

lower Flesch readability and lower BERTScore F1. The paired Task 2.3 output files are still required to determine whether routing and retries improve faithfulness relative to the normal run.

Third, the main limitations are: (i) the faithfulness checks are heuristics and can produce false positives and false negatives; (ii) document-level simplification (Task 1.2) is constrained by the input length limits of the model, which complicates the handling of long documents; and (iii) reference-based metrics may penalise valid reformulations.

8. Conclusions and Future Work

We presented a controlled-generation system for scientific text simplification in the CLEF 2026 SimpleText track. The system combines candidate generation with automatic faithfulness checks and a retry mechanism aimed at reducing factual distortion. The official sentence-level evaluation shows that the Qwen2.5-3B + QLoRA submission performs active rewriting, with no exact source copies and a SARI of 42.79 against the `simple_original` reference set.

Future work will focus on stronger factuality verification, including NLI-based consistency checks, improved document-level handling for long inputs, and a more direct evaluation of the controlled-generation setting once paired Task 2.3 results are available.

Declaration on Generative AI

Generative AI tools were used to improve the wording in English and to edit some LaTeX text. The authors checked and edited the final version. The authors take full responsibility for the content of the paper.

Acknowledgments

This work has been supported by grant PID2023-148577OB-C21 (Human-Centered AI: User-Driven Adapted Language Models – HUMAN-AI) by MICIU/AEI/10.13039/501100011033 and by FEDER/UE.

References

[1] L. Ermakova, H. Azarbonyad, J. Bakker, G. K. Shahi, B. Vendeville, J. Kamps, CLEF 2026 SimpleText track: Simplify scientific text (and nothing more), in: Advances in Information Retrieval – 48th European Conference on Information Retrieval, ECIR 2026, Proceedings, Part IV, Springer, 2026, pp. 215–224. doi:10.1007/978-3-032-21321-1_31.

[2] J. Bakker, J. Kamps, Cochrane-auto: An aligned dataset for the simplification of biomedical abstracts, in: Proceedings of the Third Workshop on Text Simplification, Accessibility and Readability (TSAR 2024), Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 41–51. doi:10.18653/v1/2024.tsar-1.5.

[3] W. Xu, C. Napoles, E. Pavlick, Q. Chen, C. Callison-Burch, Optimizing statistical machine translation for text simplification, Transactions of the Association for Computational Linguistics 4 (2016) 401–415. doi:10.1162/tacl_a_00107.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, volume 30, 2017, pp. 5998–6008.

[5] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, P. Fung, Survey of hallucination in natural language generation, ACM Computing Surveys 55 (2023) 1–38. doi:10.1145/3571730.

[6] H. Saggion, E. Gómez-Martínez, E. Etayo, A. Anula, L. Bourg, Text simplification in simplext: Making text more accessible, Procesamiento del Lenguaje Natural 47 (2011) 341–342.

[7] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of Machine Learning Research 21 (2020) 1–67.

[8] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020, pp. 7871–7880. doi:10.18653/v1/2020.acl-main.703.

[9] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, J. Wei, Scaling instruction-finetuned language models, Journal of Machine Learning Research 25 (2024) 1–53.

[10] A. Rubio, P. Martínez, HULAT-UC3M at SimpleText@CLEF-2022: Scientific text simplification using BART, in: Working Notes of CLEF 2022 – Conference and Labs of the Evaluation Forum, volume 3180 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 2845–2851.

[11] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, BLEU: A method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2002, pp. 311–318. doi:10.3115/1073083.1073135.

[12] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 74–81.

[13] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, BERTScore: Evaluating text generation with BERT, in: International Conference on Learning Representations, 2020.

[14] D. Beauchemin, H. Saggion, R. Khoury, MeaningBERT: Assessing meaning preservation between sentences, Frontiers in Artificial Intelligence 6 (2023) 1223924. doi:10.3389/frai.2023.1223924.

[15] F. Alva-Manchego, C. Scarton, L. Specia, The (un)suitability of automatic evaluation metrics for text simplification, Computational Linguistics 47 (2021) 861–889. doi:10.1162/coli_a_00418.

[16] M. Maddela, Y. Dou, D. Heineman, W. Xu, LENS: A learnable evaluation metric for text simplification, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 16383–16408. doi:10.18653/v1/2023.acl-long.905.

[17] L. Moreno, P. Martínez, A human-in/on-the-loop framework for accessible text generation, in: Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), European Language Resources Association, Palma, Mallorca, Spain, 2026, pp. 7236–7247. doi:10.63317/2gtngp2nmx63.

[18] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, 2020, pp. 38–45. doi:10.18653/v1/2020.emnlp-demos.6.

[19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, volume 32, 2019, pp. 8024–8035.

A. Prompt templates

For reproducibility, this appendix reproduces the prompt templates used by the generator. The placeholder `texto` is replaced at run time by the input text. For the numeric retry prompt, `numeros` is replaced by the list of numerical expressions extracted from the source text. For document-level inputs, the document-level suffix in Listing 7 is inserted into the base prompt through the $\texttt{doc}_{s}uffixplaceholder$.

Listing 1: System prompt.

```
You rewrite biomedical and scientific text in plain language. You DO NOT add information,
    you DO NOT summarize, you DO NOT introduce yourself or the document. You output ONLY
    the rewritten version of the input, with the same scope: a sentence stays one or a few
     sentences, a paragraph stays a paragraph, a document stays a document. Always answer
    in the same language as the input.
```

Listing 2: Base prompt.

```
Rewrite the following biomedical text in plain language for a non-expert reader.

STRICT RULES:

* Do NOT add introductory phrases like 'We found', 'We identified', 'We included', 'This
    review', 'This study', 'The authors', 'According to', 'In this paper'. These phrases
    must not appear in your output unless they already appear in the input.
* Do NOT add interpretation, context, background, or explanations beyond the input.
* Do NOT translate the input. Answer in the SAME language as the input.
* Keep EVERY number, percentage, date, statistic, p-value, confidence interval, name,
    abbreviation (RCT, RR, CI, OR, HR, etc.) EXACTLY as written.
* Keep negations ('not', 'no', 'never', 'without'), conditions ('if', 'unless'), and
    comparisons ('more than', 'less than', 'compared to').
* You MAY: split long sentences, reorder for clarity, replace very technical words with
    simpler synonyms when the meaning is preserved.
* The output length should be similar to the input length. Do not produce a summary.
  {doc_suffix}
  Input text:
  {texto}

Plain-language version (same language, same length, same facts):
```

Listing 3: Semantic retry prompt.

```
Rewrite the following biomedical text preserving its EXACT meaning.

RULES:

* Do not change numbers, dates, names, entities.
* Do not change negations, conditions, or comparisons.
* Use clearer language only where it does not risk meaning loss.

Output only the rewritten text.

Text:
{texto}

Rewritten text:
```

Listing 4: Syntactic retry prompt.

```
Simplify the structure of the following biomedical text.
```

```
RULES:

* Split long sentences into shorter ones.
* Reduce punctuation complexity.
* Keep all factual information (numbers, names, negations).
* Do not add new information.

Output only the simplified text.

Text:
{texto}

Simplified text:
```

Listing 5: Lexical retry prompt.

```
Simplify the vocabulary of the following biomedical text.

RULES:

* Replace difficult or technical words with simpler ones when possible.
* If a technical term is essential, keep it and add a brief simple explanation.
* Do not change numbers, dates, names, entities, negations.
* Do not add information not in the original.

Output only the simplified text.

Text:
{texto}

Simplified text:
```

Listing 6: Numeric retry prompt.

```
The previous rewrite changed numerical values. Rewrite the text below in plain language
    but copy EVERY number, percentage, dose, p-value, confidence interval and statistical
    value EXACTLY as written, without rounding, approximating or rephrasing them.

Numbers that MUST appear verbatim in your output: {numeros}

Other rules:

* Answer in the same language as the input.
* Do not add introductory phrases.
* Do not change names, abbreviations, or entities.
* Output only the rewritten text.

Input:
{texto}

Rewritten text:
```

Listing 7: Document-level suffix.

```
DOCUMENT-LEVEL REQUIREMENTS:

* Keep the original order of ideas.
* Preserve key findings, numerical comparisons, limitations, and uncertainty.
* Do not collapse the document into one sentence.
* Keep the document coherent.
```