FOSU-YZH-Group at the SimpleText 2026 Track: Statistics-Aware Prompting for Task 1.1 and Exploratory Overgeneration Detection for Task 2

Zonghua Yang[1], Haoliang Qi[1,*] and Yusheng Yi[1]

[1]*Foshan University, Foshan, China*

Abstract

The CLEF 2026 SimpleText Track asks systems to simplify scientific text, and our main submission addresses Task 1.1, where complex English sentences from Cochrane biomedical abstracts must be rewritten as plain-language sentences. This problem is challenging because biomedical sentences often contain parenthetical statistical expressions, such as risk ratios, confidence intervals, heterogeneity statistics, and p-values, which should not be copied verbatim but should also not be removed without preserving the main conclusion. For Task 1.1, our system combines statistics-aware prompting with heuristic candidate selection and achieved a SARI score of 47.01 on the English Cochrane-auto 2026 main evaluation subset; we also report seven exploratory Task 2 submissions for overgeneration detection, with the best source-aware binary run achieving a CodaBench document-level F1 score of 0.7716. These results suggest that explicit handling of statistical expressions is useful for biomedical sentence simplification, while Task 2 overgeneration detection benefits from supervised classification and development-set threshold calibration.

Keywords

Biomedical Text Simplification, Scientific Text Simplification, Sentence-Level Simplification, Large Language Models, Statistics-Aware Prompting, Candidate Selection, SimpleText 2026 Track

1. Introduction

The SimpleText 2026 Track focuses on making scientific texts easier to understand. Task 1 asks participating systems to simplify scientific text [1]. Task 1.1 is a sentence-level scientific text simplification task in which the input is a complex English sentence from a Cochrane biomedical abstract and the output is a corresponding plain-language sentence [2]. Cochrane plain language summaries are intended for non-expert readers and aim to communicate the questions and main findings of systematic reviews. Therefore, the task is different from general stylistic rewriting and from summarization that only seeks shorter text. A system must reduce linguistic complexity while preserving the comparison, effect direction, and uncertainty of the biomedical evidence.

Automatic text simplification has been studied through several types of methods. Shardlow categorizes automated text simplification into lexical, syntactic, statistical machine translation, and hybrid techniques [3]. Al-Thanyyan and Azmi further review text simplification in terms of resources, corpora, methods, and evaluation settings [4]. These studies provide a general framework for text simplification. Evaluation metrics such as SARI measure the relation between the system output, the source sentence, and the reference simplification in terms of add, keep, and delete operations [5]. Recent work revisiting SARI in the era of modern large language models also suggests that simplification metrics should be interpreted together with system behavior and task-specific analysis [12]. Recent large language models can perform rewriting and simplification through natural-language instructions, and instruction-following ability affects how well models follow user intent [6].

Cochrane biomedical sentence simplification includes a more specific problem: how to handle parenthetical statistical expressions. We use this term to refer to statistical details placed in parentheses,

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

*Corresponding author.

2112571041@stu.fosu.edu.cn (Z. Yang); haoliangqi163@163.com (H. Qi); yiys@fosu.edu.cn (Y. Yi)

0009-0004-3108-841X (Z. Yang); 0000-0003-1321-5820 (H. Qi); 0009-0006-7098-3681 (Y. Yi)

including risk ratios, odds ratios, confidence intervals, I^2 values, p-values, numbers of studies, and numbers of participants. These expressions are not ordinary difficult words. They encode effect direction, group difference, and uncertainty. If a system copies these expressions directly, the output is still not plain language. If it deletes the parenthetical expression without replacement, the output may omit key conclusions. At the same time, factual consistency is an important concern in text simplification, because simplified outputs may introduce unsupported information or omit important information [7]. Therefore, in this task, parenthetical statistical expressions should be treated as a controlled rewriting problem rather than as ordinary lexical simplification.

Our system is designed around this problem. It uses statistics-aware prompting to make the rewriting of parenthetical statistical expressions explicit. It then uses heuristic candidate selection to choose among outputs generated by DeepSeek-V4-Pro and deepseek-chat prompt variants. This paper describes the Task 1.1 data, the submitted Task 1.1 system, the prompt variants, the candidate selection method, the official results, and a small set of analyses based on submitted variants. In response to the track-level preference that a single team paper should document the team's work within the track, we also report our exploratory Task 2 submissions for overgeneration detection. The main contribution and analysis of this paper remain focused on Task 1.1.

2. Experimental Setup

Our experiments had two objectives. The first objective was to improve Task 1.1 biomedical sentence simplification by explicitly rewriting parenthetical statistical expressions and selecting among multiple prompt-based candidates. The second objective was to report our exploratory Task 2 submissions for overgeneration detection, following the track preference that a single team paper should document the team's work within the track.

2.1. Task and Data

We participated in SimpleText 2026 Track Task 1.1. The input consists of individual complex sentences extracted from Cochrane biomedical abstracts. Each record contains `pair_id`, `source`, `language`, `para_id`, `sent_id`, and `complex`. The system is required to preserve these fields and add `prediction` and `run_id`.

The main score displayed on CodaBench is based on the English Cochrane-auto 2026 main evaluation subset, defined by:

```
source = Cochrane-auto 2026
language = en
```

This subset contains 3,679 sentences. The complete test file contains 48,809 records, including Cochrane full 2026, Cochrane-auto 2025, Cochrane-auto val, Cochrane-auto test, SimpleText2024, Medline, and multilingual pilot data. According to the task requirement, a Task 1 submission should provide predictions for all records in the complete test file. Therefore, our final JSON submission contains predictions for all 48,809 records.

Our core system is designed for English biomedical sentence simplification. Statistics-aware prompting, prompt variants, and heuristic candidate selection are based on English biomedical expressions and English plain-language rewriting. The complete test file includes non-English and pilot data. These data are included for complete submission, but the method design and analysis in this paper focus on the English Cochrane-auto 2026 main evaluation subset.

2.2. System Overview

The final system consists of three main components: statistics-aware prompting, candidate generation, and heuristic candidate selection. Figure 1 shows the inference pipeline. Given an input sentence, the system first applies a statistics-aware prompt. It then generates four candidate simplifications using

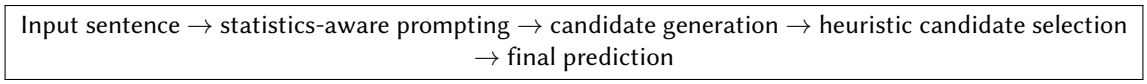

Figure 1: Inference pipeline of the submitted Task 1.1 system. The system applies statistics-aware prompting, generates candidates using DeepSeek-V4-Pro and deepseek-chat prompt variants, and uses heuristic candidate selection to produce the final prediction.

Table 1
Candidate sources used in the submitted system.

Candidate	Model and prompt	Description
`base_pro`	DeepSeek-V4-Pro + main prompt	Default candidate
`chat_original`	deepseek-chat + main prompt	Same prompt, different model
`context_chat`	deepseek-chat + context prompt	Resolves local references
`r2_cochrane`	deepseek-chat + cautious prompt	Cautious biomedical wording

DeepSeek-V4-Pro and deepseek-chat prompt variants. Finally, heuristic candidate selection filters and scores the candidates. The DeepSeek-V4-Pro output is kept by default, and an alternative candidate is selected only when it exceeds the current best candidate by a replacement margin of 0.45.

2.3. Statistics-Aware Prompting

Cochrane sentences in Task 1.1 often contain parenthetical statistical expressions. For example:

```
PPIs may reduce ulcer complications compared with placebo
(RR 0.33, 95% CI 0.10 to 1.07; P = 0.30; I2 = 18%;
5 studies, 4394 participants; low-certainty evidence).
```

The parenthetical expression contains a risk ratio, confidence interval, p-value, heterogeneity statistic, number of studies, number of participants, and evidence certainty. For non-expert readers, these numbers and symbols are not intuitive. If the parenthetical expression is copied directly, the output remains complex. If the parenthetical expression is only removed, the output may lose information about uncertainty or the possibility that the difference is due to chance.

To address this problem, we use a statistics-aware prompt. The central rule is:

```
HOW TO HANDLE STATISTICS IN PARENTHESES:
- Remove the detailed numbers (RR, CI, I2, p-values).
- REPLACE them with a short plain-language conclusion
  (e.g., "the risk was reduced by about 80%",
  "there was little to no difference between groups").
- This replacement is REQUIRED, not optional.
  It is NOT considered "adding new facts" - it's part of simplification.
```

This rule defines the rewriting of parenthetical statistical expressions as part of simplification. The model is not required to preserve symbols such as RR, CI, I^2, and p-values, but it should preserve the main conclusion supported by the parenthetical statistical expression. This may include risk reduction, risk increase, little or no difference between groups, or uncertainty of the evidence.

2.4. Candidate Generation

For each sentence in the main evaluation subset, the system generates four candidates. Table 1 lists the candidate sources used in the submitted system.

The default candidate, `base_pro`, is generated by DeepSeek-V4-Pro with the main statistics-aware prompt. The `chat_original` candidate uses the same prompt but is generated with deepseek-chat. The `context_chat` candidate uses a context-trigger prompt, where local context is used only to resolve missing references such as pronouns, group names, or comparison targets. The `r2_cochrane`

candidate uses a cautious biomedical wording prompt variant, which encourages expressions such as "may reduce", "little or no difference", and "the evidence is uncertain" when supported by the source sentence.

All prompt variants use deterministic inference with temperature 0.0. The model output is a JSON object, and the field `step3_draft` is extracted as the candidate simplification. The complete prompt variants are provided in Appendix A.

2.5. Heuristic Candidate Selection

The goal of heuristic candidate selection is to select the final prediction without using test references. The selector uses `base_pro` as the default candidate. Alternative candidates must first pass filtering and are then assigned a heuristic score.

The filtering rules are:

1. The candidate must not be empty.
2. The candidate must contain at least two characters.
3. The candidate length must not exceed 1.15 times the source length.
4. If the source sentence has more than 12 words, the candidate length must not be less than 0.35 times the source length.
5. The candidate must contain no more than two sentences.

After filtering, the system uses a deterministic rule-based scoring function. Let `r` be the ratio between the candidate word count and the source word count. The score mainly considers the following factors:

1. **Target compression ratio:** candidates closer to 0.73 times the source length receive a higher score.
2. **Statistical-symbol handling:** if the source contains parenthetical statistical expressions or statistical symbols such as RR, OR, `CI`, `I2`, I-squared, p-value, `95%`, `confidence interval`, `risk ratio`, or `odds ratio`, candidates that remove these symbols are rewarded, while candidates that keep them are penalized.
3. **Copy handling:** copying a long source sentence is penalized; copying a very short sentence is not heavily penalized.
4. **Length penalties:** candidates that are too short or too long are penalized.
5. **Plain-language biomedical cues:** candidates containing expressions such as `little or no difference`, `reduced`, `increased`, `risk`, `side effects`, `study`, `studies`, `treatment`, `vaccine`, `people`, or `uncertain` receive a small reward.

The selection rule is:

```
Select alternative candidate yi only if:
score(x, yi) > score(x, current_best) + 0.45.
Otherwise, keep the current best candidate.
```

The value 0.45 is the replacement margin. It makes the selector conservative: the DeepSeek-V4-Pro output is kept by default, and an alternative candidate is selected only when it is clearly better under the heuristic score. This value is not a theoretical optimum, but an empirical boundary used in the submitted system. The full heuristic candidate selection procedure is given in Appendix B.

2.6. Complete Submission and Reproducibility Settings

The final submission file contains predictions for all 48,809 records. The English Cochrane-auto 2026 main evaluation subset uses the candidate generation and heuristic candidate selection pipeline described above. For records outside the main evaluation subset, we used deepseek-chat with the same general prompt-based simplification procedure to generate predictions and satisfy the complete-submission

Table 2
System settings used for the final submitted run.

Item	Setting
Task	SimpleText 2026 Track Task 1.1
Level	Sentence-level
Main evaluation subset	Cochrane-auto 2026, English
Default model	DeepSeek-V4-Pro
Candidate model	deepseek-chat
API	DeepSeek OpenAI-compatible API
Temperature	0.0
Fine-tuning	No
External retrieval	No
Additional training data	No
Final run id	`FOSU-YZH-Group_Task11_hybrid_safe_selector_fixstats`
Output field	`prediction`
Submission format	One JSON file in a zip archive

requirement. These additional records are not included in the main CodaBench score reported in this paper.

The system does not use fine-tuning, external retrieval, or additional training data. All candidates are generated through prompt-based inference with temperature 0.0.

2.7. Exploratory Task 2 Submissions

We also participated in SimpleText 2026 Task 2, which focuses on identifying and avoiding hallucination or overgeneration in simplified scientific text [9]. In Task 2, the input contains a source biomedical abstract and a simplified version split into sentences. The system must identify whether simplified sentences introduce content that is not supported by the source. Our submissions mainly targeted the source-aware binary setting, denoted as Task 2.1b, where the source abstract is available and each sentence is assigned a binary label.

For Task 2 development, we used the official training and development data released by the organizers. The training set contains 21,057 examples and 350,583 sentences, while the development set contains 1,120 examples and 18,638 sentences. These data were used for classifier training, hard-negative construction, and development-set threshold selection.

Our Task 2 system was developed as an exploratory classification pipeline rather than as the main contribution of this paper. We converted the document-level data into sentence-level classification examples and used the development set to select decision thresholds. We also constructed hard-negative training examples to make the classifier less sensitive to superficial lexical overlap and more sensitive to unsupported additions. The main classifiers were based on DeBERTa-v3-large and RoBERTa-large [10, 11]. For the best run, we used a weighted soft-voting ensemble, with a larger weight assigned to the DeBERTa model and a smaller weight assigned to the RoBERTa model. The ensemble weight and decision threshold were selected on the development set before producing the CodaBench submission.

Task 2 was not used in the Task 1.1 candidate generation or heuristic candidate selection pipeline. We report it here for completeness because the same team submitted runs to Task 2 during the SimpleText 2026 Track.

3. Experimental Results

Unless otherwise stated, all official scores reported in this section correspond to the final submitted run `FOSU-YZH-Group_Task11_hybrid_safe_selector_fixstats`. We report the main result first, followed by two ablation analyses based on submitted variants: candidate-source ablation and replacement-margin ablation.

Table 3
Official results of the final system on the English Cochrane-auto 2026 main evaluation subset. Lex. Compl. denotes lexical complexity score.

Reference	SARI	BLEU	FKGL	Comp.	Split	Lev.	Add.	Del.	Lex. Compl.
simple_original	**47.01**	11.97	10.55	0.70	1.04	0.55	0.26	0.58	8.50
simple_auto	46.23	12.42	10.59	0.67	1.05	0.55	0.23	0.59	8.47

Table 4
Candidate-source ablation based on submitted variants. The final system uses all alternative candidates.

Setting	Alternative candidates allowed	SARI
Default prompt only	None	46.87
Selection with chat candidate only	`chat_original`	46.95
Selection without cautious wording candidate	`chat_original`, `context_chat`	46.93
Selection without context candidate	`chat_original`, `r2_cochrane`	46.90
Final selection	`chat_original`, `context_chat`, `r2_cochrane`	**47.01**

3.1. Main Result

Table 3 reports the official results on the English Cochrane-auto 2026 main evaluation subset.

The final system achieved a SARI score of 47.0145 under the `simple_original` setting, rounded to 47.01 in Table 3. Compared with the single DeepSeek-V4-Pro prompt run, the final system reduced the compression ratio from 0.75 to 0.70, reduced additions from 0.28 to 0.26, and increased deletions from 0.56 to 0.58. This indicates that heuristic candidate selection produced slightly shorter outputs and reduced some added content, although the overall improvement was modest.

The final run includes explicit statistical-symbol handling in heuristic candidate selection. This allows the selection stage to consider whether statistical symbols such as RR, OR, CI, and p-values are preserved or removed. Together with statistics-aware prompting, this forms the final system's mechanism for handling parenthetical statistical expressions.

3.2. Candidate-Source Ablation

Table 4 reports a candidate-source ablation based on submitted variants. The first row is the default DeepSeek-V4-Pro prompt run without heuristic candidate selection. The remaining rows use the same selection framework but change which alternative candidates are allowed.

The full candidate set achieved the highest SARI score. Using only the `chat_original` candidate improved over the default prompt-only run, but remained below the final system. Removing either the cautious biomedical wording candidate or the context-trigger candidate also reduced SARI. These results suggest that the gain does not come from adding more candidates alone. Instead, the alternative candidates are useful when they provide local replacements that satisfy the heuristic selection constraints.

3.3. Replacement-Margin Ablation

Table 5 reports a replacement-margin ablation based on submitted variants. All rows use the same candidate sources. The only changed factor is the replacement margin used by heuristic candidate selection.

Larger replacement margins make heuristic candidate selection more conservative, but the score decreases. This result suggests that some replacements selected by the final system contribute to the final SARI score. If the replacement margin is too high, the system keeps more default candidates and loses some useful local replacements. This comparison is based on submitted variants and is reported as an ablation analysis, not as direct tuning against test references.

Table 5
Replacement-margin ablation based on submitted variants. Larger margins make the selector more conservative.

Setting	Replacement margin	SARI
Final selection	0.45	**47.01**
Stricter selection	0.55	46.87
Strictest selection	0.65	46.86

Table 6
A statistics-aware prompting example from the final submission.

Type	Text
Source	PPIs may reduce ulcer complications compared with placebo (RR 0.33, 95% CI 0.10 to 1.07; P = 0.30; I2 = 18%; 5 studies, 4394 participants; low-certainty evidence).
Prediction	PPIs may reduce ulcer complications compared with placebo, but the evidence is uncertain and the difference could be due to chance.

Table 7
A heuristic candidate selection example from the final submission.

Type	Text
Source	There is insufficient evidence to support the use of older agents like alkylating agents or cytokine inhibitors.
`base_pro`	There is not enough evidence to support using older treatments, such as some chemotherapies or drugs that suppress the immune system.
Selected prediction	There is not enough evidence to recommend using older drugs for treatment.
Selected from	`context_chat`

3.4. Development Runs

During development, the initial prompt achieved 46.21 SARI. Adding explicit statistics-handling instructions improved the deepseek-chat run to 46.64, and the DeepSeek-V4-Pro run with the same main prompt achieved 46.87. The final submitted variant with heuristic candidate selection achieved 47.01. These development results indicate that the main improvement comes from making parenthetical statistical expressions explicit in the prompt and then applying conservative candidate selection.

3.5. Case Studies

Table 6 shows a real example of statistics-aware prompting from the final submission.

This example is from CD014585, `para_id = 4`, `sent_id = 8`. The source sentence contains RR, 95% CI, p-value, and I^2. The final output removes these statistical symbols and preserves a plain-language conclusion: PPIs may reduce ulcer complications, but the evidence is uncertain and the difference could be due to chance. This example corresponds to the core rule in the prompt: remove detailed statistical values and replace them with a short conclusion.

Table 7 shows an example of heuristic candidate selection.

This example is from CD015494, `para_id = 5`, `sent_id = 15`. The `base_pro` output expands "alkylating agents or cytokine inhibitors", making the candidate longer than the source sentence. The selected `context_chat` output preserves the main conclusion that there is not enough evidence to recommend older drugs for treatment and avoids a long explanation. This example shows that heuristic candidate selection can choose an alternative candidate that better matches the compression constraint when the default candidate is too long. We do not interpret this example as a human judgment of medical accuracy.

3.6. Exploratory Task 2 Results

Table 8 summarizes the seven Task 2 submissions made by our team. The best submitted Task 2 run was `FOSU-YZH-Group_Task21b_DeBERTa075_R.zip`, which achieved a CodaBench document-level F1 score of 0.7716. The run name indicates the Task 2.1b source-aware binary setting and a DeBERTa-based ensemble configuration. The next best run, `FOSU-YZH-Group_Task21b_DeBERTa_V3_L.zip`, achieved 0.7704. Other DeBERTa and DeBERTa–RoBERTa variants achieved scores between 0.7635 and 0.7682, while an early Task 2.2b DocAgent-based detection run achieved 0.4039.

Table 8
Exploratory Task 2 submissions on CodaBench. The score is the document-level F1 score reported by the Task 2 leaderboard.

CodaBench ID	Setting and method	Score
726350	Task 2.1b, DeBERTa/RoBERTa ensemble	0.7716
722892	Task 2.1b, DeBERTa-v3-large	0.7704
724878	Task 2.1b, DeBERTa/RoBERTa ensemble	0.7682
724221	Task 2.1b, DeBERTa/RoBERTa variant	0.7676
723317	Task 2.1b, DeBERTa-v3-large	0.7635
721209	Task 2.1b, DeBERTa-v3-large	0.7146
716798	Task 2.2b, DocAgent-based detection	0.4039

These results show that the best Task 2 submissions came from source-aware binary overgeneration detection with DeBERTa-based classifiers. The Task 2.2b DocAgent-based run was substantially weaker in this preliminary setting. Since the Task 2 work was developed as a separate exploratory classification pipeline and was not used to improve the Task 1.1 simplification outputs, we keep the main analysis of this paper focused on Task 1.1.

4. Discussion and Conclusions

This paper reports the FOSU-YZH-Group system for SimpleText 2026 Track Task 1.1. The system uses statistics-aware prompting for parenthetical statistical expressions in Cochrane biomedical sentences. The prompt requires the model to remove detailed statistical values and replace them with a plain-language conclusion supported by the source sentence. The final submission further applies heuristic candidate selection: the DeepSeek-V4-Pro output is used as the default candidate, multiple deepseek-chat prompt variants are used as alternative candidates, and an alternative candidate is selected only when it passes filtering and exceeds the replacement margin.

On the English Cochrane-auto 2026 main evaluation subset, the final submitted run `FOSU-YZH-Group_Task11_hybrid_safe_selector_fixstats` achieved a SARI score of 47.01. Candidate-source and replacement-margin ablations based on submitted variants show that the final selection setting outperforms the default prompt-only run, the chat-only selection setting, the setting without the cautious biomedical wording candidate, the setting without the context-trigger candidate, and more conservative replacement-margin settings. We also reported seven exploratory Task 2 submissions, where the best source-aware binary overgeneration detection run achieved a CodaBench document-level F1 score of 0.7716. Future work may calibrate heuristic candidate selection using development references and include human evaluation of factual accuracy in parenthetical statistical-expression simplification. For Task 2, our exploratory submissions suggest that source-aware overgeneration detection benefits from supervised classification models and development-set threshold calibration, while multi-class overgeneration categorization remains more difficult in our preliminary experiments.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62276064). We thank the CLEF 2026 SimpleText Track organizers for providing the benchmark datasets, evaluation platform, task documentation, and review feedback.

Declaration on Generative AI

During the preparation of this work, the authors used *ChatGPT* in order to support **Drafting content**, **Abstract drafting**, **Paraphrase and reword**, **Improve writing style**, and **Grammar and spelling check**. The authors used DeepSeek-V4-Pro and deepseek-chat as part of the submitted Task 1.1 system to generate sentence simplification candidates. The authors also used classifier-based pretrained language models, including DeBERTa-v3-large and RoBERTa-large, during exploratory Task 2 development. After using these tools and services, the authors reviewed and edited the content as needed and take full responsibility for the final paper.

References

[1] Liana Ermakova, Hosein Azarbonyad, Jan Bakker, Berkay Chakar, Gautam Kishore Shahi, and Jaap Kamps. Overview of the CLEF 2026 SimpleText Track: Simplify Scientific Texts (and Nothing More). In Matthias Hagen et al. (Eds.), *Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026)*. Lecture Notes in Computer Science. Springer-Verlag, 2026.

[2] Jan Bakker, Liana Ermakova, and Jaap Kamps. Overview of the CLEF 2026 SimpleText Task 1: Simplify Scientific Text. In Eva Sánchez Salido et al. (Eds.), *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2026)*. CEUR Workshop Proceedings, CEUR-WS.org, 2026.

[3] Matthew Shardlow. A Survey of Automated Text Simplification. *International Journal of Advanced Computer Science and Applications*, Special Issue on Natural Language Processing, 4(1), 2014. DOI: 10.14569/SpecialIssue.2014.040109.

[4] N. S. Al-Thanyyan and A. M. Azmi. Automated Text Simplification: A Survey. *ACM Computing Surveys*, 54(2), Article 43, 2021. DOI: 10.1145/3442695.

[5] Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. Optimizing Statistical Machine Translation for Text Simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415, 2016.

[6] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[7] Ashwin Devaraj, William Sheffield, Byron C. Wallace, and Junyi Jessy Li. Evaluating Factuality in Text Simplification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 7331–7345, 2022. DOI: 10.18653/v1/2022.acl-long.506.

[8] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[9] Berkay Chakar, Jan Bakker, Liana Ermakova, and Jaap Kamps. Overview of the CLEF 2026 SimpleText Task 2: Identify and Avoid Hallucination. In Eva Sánchez Salido et al. (Eds.), *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2026)*. CEUR Workshop Proceedings, CEUR-WS.org, 2026.

[10] Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTaV3: Improving DeBERTa using

ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. arXiv preprint arXiv:2111.09543, 2021.
[11] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692, 2019.
[12] Nico Hofmann, Julian Dauenhauer, Nils Ole Dietzer, Idehen Daniel Idahor, and Christin Katharina Kreutz. Lexical Simplification for Scientific Texts: Revisiting SARI in the Era of Modern LLMs. In Matthias Hagen et al. (Eds.), *Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026)*. Lecture Notes in Computer Science. Springer-Verlag, 2026.

A. Prompt Variants Used for Candidate Generation

A.1. Main Statistics-Aware Prompt

The main statistics-aware prompt was used for both `base_pro` and `chat_original`. The `base_pro` candidate was generated with DeepSeek-V4-Pro, while the `chat_original` candidate was generated with deepseek-chat.

```
You are a medical text simplifier.
Rewrite the [Target Sentence] into plain English that a non-expert reader can understand.

HOW TO HANDLE STATISTICS IN PARENTHESES:
- Remove the detailed numbers (RR, CI, I2, p-values).
- REPLACE them with a short plain-language conclusion (e.g., "the risk was reduced by about
    80%", "there was little to no difference between groups").
- This replacement is REQUIRED, not optional. It is NOT considered "adding new facts" - it's
     part of simplification.

OTHER RULES:
1. Output exactly ONE sentence unless the original is very long (over 40 words) and has
    clearly separate ideas - then you may split into TWO.
2. Replace uncommon medical abbreviations (RCT, AE, TCV) with everyday words. Common terms (
    vaccine, treatment, study) can stay.
3. Keep the sentence shorter than the original, but make sure the key finding is clearly
    stated.

Output STRICT JSON:
{
  "step1_context_analysis": "I will simplify by...",
  "step3_draft": "Your simplified sentence."
}
```

A.2. Context-Trigger Prompt

The context-trigger prompt was used to generate the `context_chat` candidate. Local context was provided only for resolving missing references in the target sentence, not for summarizing surrounding sentences.

```
You are a medical text simplifier.
Rewrite the [Target Sentence] into plain English that a non-expert reader can understand.

You may use [Local Context] only to resolve missing references, such as what "this", "these",
     "they", "control", "intervention", or "groups" refer to.
Do NOT summarize the context.
Do NOT add facts from the context unless they are needed to make the target sentence
    understandable.
The simplified sentence must mainly reflect the [Target Sentence].
```

```
HOW TO HANDLE STATISTICS IN PARENTHESES:
- Remove the detailed numbers (RR, CI, I2, p-values).
- REPLACE them with a short plain-language conclusion (e.g., "the risk was reduced by about
    80%", "there was little to no difference between groups").
- This replacement is REQUIRED, not optional. It is NOT considered "adding new facts" - it's
     part of simplification.

OTHER RULES:
1. Output exactly ONE sentence unless the original is very long (over 40 words) and has
    clearly separate ideas - then you may split into TWO.
2. Replace uncommon medical abbreviations (RCT, AE, TCV) with everyday words. Common terms (
    vaccine, treatment, study) can stay.
3. Keep the sentence shorter than the original, but make sure the key finding is clearly
    stated.
4. Avoid copying unnecessary details from [Local Context].

Output STRICT JSON:
{
  "step1_context_analysis": "I will simplify by...",
  "step3_draft": "Your simplified sentence."
}
```

A.3. Cautious Biomedical Wording Prompt

The cautious biomedical wording prompt was used to generate the `r2_cochrane` candidate. It encourages cautious biomedical wording when such wording is supported by the source sentence. The prompt includes Cochrane-style expressions, but it is not an official Cochrane guideline prompt.

```
You are a medical text simplifier.
Rewrite the [Target Sentence] into plain English for a non-expert reader.

STYLE:
- Use cautious biomedical plain language.
- Prefer Cochrane-style wording when supported by the sentence:
  "may reduce", "probably reduces", "may increase",
  "makes little or no difference", "there is little or no difference",
  or "the evidence is uncertain".
- Keep the key finding and comparison.
- Avoid unnecessary details.

HOW TO HANDLE STATISTICS IN PARENTHESES:
- Remove detailed numbers such as RR, CI, I2, and p-values.
- Replace them with a short plain-language conclusion.
- Prefer qualitative conclusions over exact percentages.
- Only mention an approximate percentage when it is directly clear from the sentence.
- This replacement is part of simplification and is not considered adding new facts.

OTHER RULES:
1. Output exactly ONE sentence unless the original is very long and has clearly separate
    ideas.
2. Replace uncommon medical abbreviations with everyday words.
3. Keep the output shorter than the original.

Output STRICT JSON:
{
  "step1_context_analysis": "I will simplify by...",
  "step3_draft": "Your simplified sentence."
}
```

B. Heuristic Candidate Selection

The submitted `hybrid_safe_selector_fixstats` used a deterministic rule-based candidate selector. The selector used the DeepSeek-V4-Pro output as the default candidate and replaced it only when an alternative candidate exceeded the current best candidate by a fixed replacement margin.

```
Input:
- source sentence x
- default candidate y0 from DeepSeek-V4-Pro
- alternative candidates y1, y2, y3 from deepseek-chat variants

Candidate filtering:
Reject candidate yi if:
1. yi is empty;
2. character length of yi is less than 2;
3. word_count(yi) / word_count(x) > 1.15;
4. word_count(x) > 12 and word_count(yi) / word_count(x) < 0.35;
5. yi contains more than two sentences.

Candidate scoring:
Let r = word_count(yi) / word_count(x).

score(x, yi) starts from 0 and adds:

1. Compression score:
  max(0, 1 - abs(r - 0.73) / 0.35)

2. Statistical-symbol handling:
  If x contains statistical symbols such as RR, OR, CI, I2, I-squared, p-values,
  95%, confidence interval, risk ratio, or odds ratio:
    add 0.6 if yi removes such symbols;
    subtract 0.8 if yi keeps such symbols.

3. Copy handling:
  If yi is identical to x:
    add 0.2 when word_count(x) <= 6;
    subtract 0.4 otherwise.

4. Length penalties:
  subtract 0.5 if r < 0.45 and word_count(x) > 12;
  subtract 0.3 if r > 0.95.

5. Plain-language biomedical cue:
  add 0.15 if yi contains at least one of:
  "little or no difference", "reduced", "increased", "risk",
  "side effects", "study", "studies", "treatment",
  "vaccine", "people", or "uncertain".

Selection:
1. Compute score(x, y0) for the default candidate.
2. For each alternative candidate yi that passes filtering:
    compute score(x, yi).
3. Replace the current best candidate with yi only if:
    score(x, yi) > score(x, current_best) + 0.45.
4. Otherwise keep the current best candidate.

Output:
- selected candidate prediction.
```

This selector does not use references from the test set. It is used only to choose among candidates generated by the prompt variants.