UPF-TALN at the CLEF 2026 SimpleText Track: Ensemble Approach for Scientific Text Simplification and Hallucination Classification

Akio Hayakawa, Horacio Saggion

Universitat Pompeu Fabra, Barcelona, Spain

Abstract

This paper reports on the UPF-TALN team's participation in the CLEF 2026 SimpleText track. We participated in Task 1.2 Document-level Scientific Text Simplification, and Task 2.2 Detect and Classify Information Distortion Errors in Simplified Sentences. We explored an ensemble approach that utilizes the outputs of multiple systems to address both tasks. For Task 1.2, we generated a candidate pool of simplified texts using various LLMs and prompts for each instance, and then selected the best candidate based on Minimum Bayes Risk (MBR) decoding with SARI as the utility function. For Task 2.2, we constructed an ensemble classifier that leverages a BERT-based classifier, a Natural Language Inference (NLI) model, LLM-based classification, and heuristic rules. Overall, while our ensemble approach showed promising performance on the SimpleText data, its effectiveness in other domains remains to be assessed.

Keywords

Natural Language Processing, Text Simplification, Hallucination Detection, CLEF 2026 SimpleText

1. Introduction

While scientific information can affect people's decisions and actions, reliable sources, such as scientific research papers, often contain complex language and require background knowledge to understand. Text Simplification is the task of rewriting a text into a simpler version, which can help people understand scientific information more easily. In text simplification, large language models (LLMs) have shown promising performance [1, 2]. However, they can also generate hallucinated information that is not present or implied by the original text, leading to harmful misinformation [3, 4]. Therefore, it is crucial to ensure both the factuality and the quality of simplification.

To address these challenges, the SimpleText track of CLEF 2026 [5] offers several tasks, including Task 1 *Text Simplification: Simplify Scientific Text* [6] and Task 2 *Controlled Creativity: Identify and Avoid Hallucination* [7]. UPF-TALN participated in **Task 1.2** *Document-level Scientific Text Simplification* and **Task 2.2** *Detect and Classify Information Distortion Errors in Simplified Sentences.* In this paper, we present our approaches and results for these tasks. For Task 1.2, we developed a generate-and-select pipeline as our core approach. This pipeline first generated a diverse set of candidates and then filtered them to create a high-quality candidate pool. From this pool, the best output was selected using Minimum Bayes Risk (MBR) decoding [8] to optimize the SARI score [9]. For Task 2.2, we relied on multiple sources of information to evaluate whether a simplified sentence contains hallucinated information, including a fine-tuned BERT classifier, a natural language inference (NLI) model, LLM-based classification, and heuristic rules. We then fed the outputs of these strategies into a meta-classifier based on a gradient boosting model for the final decision.

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

akio.hayakawa@upf.edu (A. Hayakawa); horacio.saggion@upf.edu (H. Saggion)

Table 1
Example of Document-level Simplification from Cochrane-auto train.

Abstract	Therapeutic aspiration in addition to metronidazole to hasten clinical or radiologic resolution of uncomplicated amoebic liver abscesses cannot be supported or refuted by the present evidence. The trials lack methodological rigour and adequate sample size to conclude on the presence of effectiveness of adjunctive image-guided aspiration plus metronidazole versus metronidazole alone. Further randomised trials are necessary.
Lay summary	No additional benefit has been found with percutaneous needle aspiration plus metronidazole versus metronidazole alone for uncomplicated amoebic liver abscesses in hastening clinical and radiologic resolution. However, this conclusion is based on trials with methodological flaws and with insufficient sample sizes, and requires further confirmation in larger well-designed, randomised trials.

Table 2
Comparison of data statistics between training and test data.

			Abstract			Lay Summary		
Phase	Source	# Doc.	Avg # Sent.	Avg # Word	Words / Sent.	Avg # Sent.	Avg # Word	Words / Sent.
`Train`	Cochrane-auto train	849	13.67	340.39	24.90	9.12	196.80	21.58
`Test`	Cochrane-auto 2026	175	21.02	561.11	26.69	(Not Available)		

2. Task 1: Text Simplification - Simplify Scientific Text

2.1. Task Description

Continuing from SimpleText 2025 [10], Task 1 mainly employed a dataset derived from Cochrane-auto [11]. This framework generates parallel simplification corpora by aligning abstracts with their corresponding lay summaries from identical biomedical articles in the Cochrane Library. Task 1 is composed of two subtasks: 1.1 *Sentence-level Scientific Text Simplification* and 1.2 *Document-level Scientific Text Simplification*, the goal of both being to generate simplified text from abstracts at the sentence or document level, respectively. We only participated in Task 1.2, which is closer to real-world scenarios.

The organizers provided an evaluation dataset consisting of 8,069 documents for Task 1.2 (and 48,809 sentences for Task 1.1). While the entire evaluation dataset consists of various sources, including non-English documents, non-Cochrane documents, and Cochrane with different granularities, only the *Cochrane-auto 2026* subset, a newly crawled and processed dataset in English for the 2026 shared task, was used for the final evaluation.

To evaluate the performance of the submissions, the organizers used SARI [9] as the primary metric, which measures the quality of simplification by comparing n-grams in an original text, a system output, and reference texts. Lay summaries were hidden from participants, and the evaluation was automatically conducted upon submission on the CodaBench platform.

For reference, the organizers provided a parallel training dataset, Cochrane-auto train, similar to SimpleText 2025. Table 1 shows an example, and Table 2 shows the statistics of the training and test datasets. The training set suggests that simplification decreases the average number of sentences and words by around 60%, and lay summaries for the test set should show similar trends. However, Cochrane-auto 2026 shows some disparities from the training set in terms of the average number of sentences and words, which may affect the performance of the systems.

2.2. Methodology

While the evaluation of text simplification remains an open question [12], the use of automatic metrics is well established. To maximize the score of reference-based metrics, including SARI, MBR decoding [8] has been widely used in text generation tasks [13, 14]. MBR decoding is a technique that selects the

optimal output from multiple candidates by maximizing a specific metric score.

Hayakawa et al. [14] showed promising results by applying MBR decoding to readability-controlled paragraph-level text simplification in English. Their approach first generated a diverse set of candidate simplifications and then filtered them to create a high-quality candidate pool for MBR decoding. Their generation step used various LLMs and prompts to generate candidates, and their filtering step utilized a readability classifier and a similarity score between the original and simplified texts to filter out low-quality candidates. Subsequently, MBR decoding was applied with a BERT-based similarity metric as the utility function.

While we fundamentally followed this generate-and-select pipeline, we made some modifications to adapt it to this shared task due to the large-scale and multilingual nature of the dataset, even though only a small English subset was used for the final evaluation. Specifically, we employed efficient K-means clustering [15] for the filtering step.

2.2.1. Generation

The process starts with generating a large set of initial simplification candidates for each source abstract. We used three prompts and four LLMs to generate a diverse set of candidates.

- **Prompts**: We prepared three prompts automatically generated by GPT-5.4 based on an inductive approach, which involved extracting simplification features from the training data to create instructions for the prompts. See Appendix A for details of the prompts.
- **LLMs**: We employed two open-source LLMs, Gemma-3-4B [16] and Qwen-3-4B [17], and two proprietary LLMs, GPT-5-mini [18] and Gemini-2.5-Flash-lite [19].

For each combination of prompt and LLM, we performed three simplification trials, using three different random seeds for the open-source LLMs and three trials for the proprietary LLMs. We built three types of candidate pools: one with only open-source LLMs (18 candidates per instance), one with only proprietary LLMs (18 candidates), and one with all LLMs combined (36 candidates).

2.2.2. Filtering

After the generation, we created an optimized candidate pool of up to 10 candidates for MBR decoding. We used K-means clustering to cluster candidates embeddings generated by BGE-M3 [20], which is a sentence embedding model applicable to long texts in multiple languages. We set the number of clusters to 3 to balance diversity and quality. Candidates closest to the centroid were primarily selected, with the number of selected candidates weighted by the cluster size. However, to maximize the benefits of MBR decoding, which requires a diverse candidate pool [13], we applied another filter to ensure structural diversity. A candidate was added to the pool only if its BLEU [21] against every candidate already in the pool is below a threshold of 50, following Hayakawa et al. [14].

As an ablation study, we applied two additional filtering approaches, exclusively for Cochrane-auto 2026, to measure the effectiveness of K-means clustering. The first approach is a random selection of candidates, which serves as a baseline. The second approach is an LLM-as-a-judge-based ranking. In this approach, we prompted Gemma-3-14B [16] to score the overall quality of each candidate simplification on a scale from 1 to 5. The final score was computed as the weighted average of the next-token probabilities, derived by converting the logits of each score into a probability distribution (See Appendix B for details). An optimized candidate pool was then constructed based on these scores, with the diversity-aware selection described above.

2.2.3. MBR Decoding

Finally, we applied MBR decoding to the constructed pool. MBR decoding selects the single candidate that maximizes the expected utility function against all other candidates in the set. The candidate with

Table 3
Result of Task 1.2. MBR score represents the average pairwise SARI score of the final candidate, calculated against each simplification in the filtered candidate pool.

Filtering	LLMs: Gemma-3 4B	LLMs: Qwen-3 4B	LLMs: GPT-5 mini	LLMs: Gemini-2.5 Flash-lite	# Cands. / Doc.	MBR Score	Final SARI
K-means	✓	✓			18	48.66	43.87
			✓	✓	18	49.82	**45.67**
	✓	✓	✓	✓	36	47.74	45.51
Random	✓	✓			1	-	42.56
			✓	✓	1	-	45.52
	✓	✓	✓	✓	1	-	44.23
LLM-as-a-judge Ranking	✓	✓			18	49.70	44.14
			✓	✓	18	51.00	**45.84**
	✓	✓	✓	✓	36	48.59	45.30

the highest average utility score against all other candidates in the pool was selected as the final output. The final output $\hat{y}_{MBR}$ can be expressed as:

$$\hat{y}_{MBR} = \underset{y \in \mathcal{H}}{\operatorname{argmax}}(\mathbb{E}_{\mathcal{H}}[\mathbb{E}_{y' \in \mathcal{H}}[u(y, y')]]), \tag{1}$$

where $\mathcal{H}$ is the candidate pool and $u(y, y')$ is the utility function, defined as $\mathrm{SARI}(original, y, y')$.

2.3. Results and Discussion

Table 3 shows the results of our submitted system and the ablation settings. Final SARI scores show that the system with the combination of GPT-5 and Gemini-2.5 achieved the highest performance. This suggests that these proprietary LLMs are more effective in generating high-quality candidates than smaller open-source LLMs. Interestingly, aggregating the outputs of all LLMs did not necessarily improve performance, which is contrary to previous findings [14].

Compared with random selection, applying MBR decoding with K-means filtering consistently showed improved performance, which indicates that MBR decoding can effectively select candidates with higher SARI scores.

Replacing K-means clustering with the LLM-as-a-judge-based ranking for candidate filtering further improved performance, which suggests that LLM-based ranking can be more effective in selecting high-quality candidates for MBR decoding. However, this approach requires substantially more computational resources, which can be a significant drawback when processing large-scale datasets.

Finally, we observed a non-negligible gap between the MBR score and SARI, which implies that the candidates in the pool lack sufficient similarity to the actual references. This could be due to the disparity between the training dataset and Cochrane-auto 2026, which may have affected the effectiveness of our inductive prompting approach. Furthermore, since only five examples were provided in the prompt, it may not have been sufficient to capture the characteristics of Cochrane-auto 2026. Providing more examples or using a more curated set of examples may lead to better performance.

3. Task 2: Controlled Creativity - Identify and Avoid Hallucination

3.1. Task Description

While LLMs have shown promising performance in text simplification, they can also produce problematic outputs that contain hallucinated information or gibberish texts, which can lead to harmful misinformation [3, 4]. Task 2 aims to identify and classify such hallucination or overgeneration [6]. The organizers provided a dataset containing simplified texts submitted to Task 1 in the previous year (CLEF 2025 SimpleText Track) with annotations of the types of overgenerations they contain. Task 2 is

Table 4
Hallucination categories and their descriptions.

Category	Description	# in Train	Typical Example
None	No overgeneration detected	323702 (92.33%)	-
LEAKED INSTRUCTIONS	Leaked task framing, assistant chatter, simplification rationale	8238 (2.35%)	*... Certainly! Here's a simplified version while maintaining ...*
UNGROUNDED INJECTION	Unsupported factual or interpretive additions	5735 (1.64%)	Abstract: *No other statistically significant effects on clinical or biochemical outcomes were reported on any of the evaluated interventions.* Lay Summary: *There is also no evidence that they are effective for other conditions such as rheumatoid arthritis, rheumatic fever, and psoriasis.*
GENERATION FAILURE	Catastrophic or unusable output (degenerate, truncated, corrupted)	11331 (3.23%)	*... Output: Limits: Limits: Limits: Limits ...*
REPETITIVE CONTENT	Repetition loops	1477 (0.42%)	*... We judged the Overall Risk of Bias across Trials was low. The risk of biased results was judged to be low across all trials. ...*
GROUNDED OVERGENERATION	Source-supported content beyond simplification scope	100 (0.03%)	-

composed of three subtasks, 2.1 *Identify Creative Generation at Document Level*, 2.2 *Detect and Classify Information Distortion Errors in Simplified Sentences*, and 2.3 *Avoid Creative Generation and Perform Grounded Generation by Design*. While Task 2.1 is a binary classification (hallucinated or not) task at the document level, Task 2.2 is a multi-class classification task at the sentence level, which requires classifying hallucinated sentences into six categories. We only participated in Task 2.2 to focus on exploring the factuality of text simplification outputs.

In Task 2.2, participants were expected to classify each simplified sentence into one of the six categories described in Table 4. The organizers provided a training and validation set with 350,583 and 18,638 annotated sentences, respectively, where the annotation was done automatically. Additionally, a test set with 105,001 sentences without annotations was provided. All sets contain the original abstracts used for simplification. The systems were evaluated based on the accuracy of the predicted categories only for instances with non-None ground truth.

3.2. Methodology

In text classification tasks, ensembles of multiple systems have been shown to be effective in improving performance [22, 23]. Therefore, we explored various approaches for hallucination classification and fed them into a meta-classifier model. We generated four types of features for each sentence based on the following approaches:

- **Heuristic features** (39 features): The number of words, sentences, and characters, as well as the counts of specific words (e.g., "would you", "please", "output") that are likely to be used in specific categories.
- **BERT-based features** (13 features): We fine-tuned *Clinical ModernBERT* [24], a BERT-based model pre-trained on clinical text, for the category classification. We utilized its logits, their softmax probabilities per category, and the predicted category as features.
- **NLI-based features** (3 features): We used *PubMedBERT-mnli* [25], an NLI model pretrained on biomedical text. We utilized its predicted probabilities of entailment, contradiction, and neutral

Table 5
Data distribution for subsets used in our experiments.

		Features	Meta-classifier		
	% in	Fine-tuning	Training		Evaluating
Category	Train	BERT	(a)	(b)	
None	92.33	3000	3000	5000	500
LEAKED INSTRUCTIONS	2.35	1000	1000	1000	200
UNGROUNDED INJECTION	1.64	2000	2000	2000	200
GENERATION FAILURE	3.23	1500	1500	1500	300
REPETITIVE CONTENT	0.42	300	300	300	50
GROUNDED OVERGENERATION	0.03	50	50	50	3

Table 6
Result of Task 2.2. Accuracy scores are calculated for instances with non-None ground-truth categories. "-XXX" in the second column means Full without the XXX features.

Train		F1 score on Val						Val	Test Prediction Distribution						Test
Subset	Feats.	None	LEAK	UNG	GEN	REP	OVER	Acc	(None)	LEAK	UNG	GEN	REP	OVER	Acc
(a)	Full	.929	.870	.810	.942	.577	.500	**.8659**	(.698)	.180	.056	.037	.020	.009	.7069
	- Heur.	.852	.870	.673	.946	.484	.000	.8274	(.145)	.246	.020	.040	.548	.001	.6306
	- BERT	.928	.855	.753	.947	.590	.182	.8367	(.390)	.462	.003	.046	.021	.079	.5998
	- NLI	.926	.873	.783	.947	.541	.333	.8566	(.656)	.212	.055	.041	.020	.017	.7063
	- LLM	.922	.878	.776	.942	.518	.000	.8459	(.801)	.061	.090	.032	.010	.007	**.7801**
(b)	Full	.925	.863	.780	.942	.569	.333	.8552	(.971)	.010	.002	.010	.007	.000	.2813
	- Heur.	.828	.853	.675	.940	.458	.000	.8367	(.982)	.011	.002	.004	.002	.000	.1497
	- BERT	.925	.856	.758	.940	.629	.222	.8473	(.971)	.009	.002	.010	.008	.000	.2618
	- NLI	.922	.865	.772	.944	.605	.250	.8579	(.970)	.011	.002	.009	.008	.000	.2653
	- LLM	.923	.871	.747	.942	.528	.000	.8313	(.846)	.049	.060	.022	.011	.012	.7682

as features, providing the original abstract as the premise and the simplified sentence as the hypothesis.

- **LLM-based features** (13 features): We prompted Gemma-3-14B [16] to classify the sentence into one of the six categories. Similar to the BERT-based features, we utilized its logits, their softmax probabilities per category, and the predicted category as features.

A total of 68 features were generated for each sentence, which were then fed into a meta-classifier model. We employed a LightGBM model[1] as the meta-classifier. Taking into account the size and class imbalance of the dataset, we divided the training dataset into the following subsets: a subset for fine-tuning the BERT-based classifier, a subset for training the meta-classifier, a subset for evaluating the meta-classifier, and an unused subset. For the meta-classifier training, we prepared two different versions with different class distributions. The configurations of these subsets are detailed in Table 5. Prior to the division, data points in the None category containing obvious errors (e.g., those ending with "...") were removed.

The iteration of the meta-classifier that achieved the highest macro F1 score on the validation set was selected as the final model for prediction on the test set. Furthermore, we conducted an ablation study to explore the contribution of each feature type by removing one feature type at a time. See Appendix C for details of the features and the meta-classifier.

3.3. Results and Discussion

Table 6 shows the results of our experiments. The best performance in both subsets of the validation set was achieved when all features were utilized, which indicates that our ensemble approach works effectively for hallucination classification under in-domain settings. However, the performance on

[1]https://github.com/lightgbm-org/LightGBM

the test set was drastically lower than that on the validation set. This suggests that the difference in data and category distribution between the validation and test sets has a far greater impact on the performance than the choice of features.

A critical factor behind this performance gap lies in the evaluation design, which calculates accuracy scores exclusively for instances with non-None ground-truth categories. This specific setup heavily penalizes any model that frequently assigns the None label. For instance, training with the subset (b) suffered a significant drop (Full: .2813) compared to that with (a) (.7069). This drop occurred because the meta-classifier trained with (b) predicted None for 97.1% of the test set. As detailed in Table 5, (b) contains a larger number of None instances compared to (a). This higher exposure to the majority class caused the meta-classifier to favor the None label. When applied to the test set, the meta-classifier trained with (b) assigned None to nearly all unseen data. Under the shared evaluation design, this heavy None bias automatically degraded the final performance.

The removal of LLM-based features resulted in a performance boost on the test set for both subsets. This improvement is explained by the changes in the distribution of predicted categories on the test set. When LLM-based features were removed, the prediction ratio for UNGROUNDED INJECTION increased from 5.6% to 9.0% in (a) and from 0.2% to 6.0% in (b). As shown in section C, the LLM-as-a-judge classifier suffered from a severe prediction bias, outputting consistently high probabilities for LEAKED INSTRUCTIONS and low probabilities for UNGROUNDED INJECTION. This bias functioned as a noise rather than a helpful signal. While data in the training set was automatically annotated and not always reliable, providing a more careful description of the categories in the prompt for the LLM-based classifier may lead to a more balanced prediction.

Overall, since the ground-truth annotations for the test set remain unavailable, we cannot determine whether the performance drop on the test set is due to a mismatch between the training and test data. However, our findings suggest that a more careful design of the systems is needed to obtain a genuine ability to generalize to unseen data.

4. Conclusion

In this paper, we described UPF-TALN's participation in Task 1 and 2 of the CLEF 2026 SimpleText Track. In both tasks, we explored ensemble approaches by combining multiple systems or features, which showed promising performance on the validation set. However, the performance on the test set was much lower than that on the validation set, suggesting that a more careful design of the systems is needed to obtain a genuine ability to generalize to unseen data. Additionally, these ensemble approaches are computationally expensive, which may not be ideal for real-world applications. Exploring more efficient approaches will be one of our future research directions.

Moreover, as introduced in Task 2, text simplification optimized for SARI (or other automatic metrics) does not necessarily lead to better factuality, which is a critical aspect for text simplification. Exploring better evaluation methods will be another important direction for future work.

Acknowledgments

This work is partially financed by the Ministerio de Ciencia, Innovación y Universidades, Agencia Estatal de Investigaciones: project CPP2023-010780 funded by MICIU/AEI/10.13039/501100011033 and by FEDER, UE ("Habilitando Modelos de Lenguaje Responsables e Inclusivos"). We also acknowledge support from Grant PCI2026-177545-1 funded by MICIU/AEI/10.13039/501100011033/CHIST-ERA (EU).

Declaration on Generative AI

During the preparation of this work, the author(s) used Gemini in order to: Paraphrase and reword and Improve writing style. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] X. Wu, Y. Arase, An in-depth evaluation of large language models in sentence simplification with error-based human assessment, ACM Trans. Intell. Syst. Technol. 17 (2026). URL: https://doi.org/10.1145/3744744. doi:10.1145/3744744.

[2] A. Barayan, J. Camacho-Collados, F. Alva-Manchego, Analysing zero-shot readability-controlled sentence simplification, in: O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, S. Schockaert (Eds.), Proceedings of the 31st International Conference on Computational Linguistics, Association for Computational Linguistics, Abu Dhabi, UAE, 2025, pp. 6762–6781. URL: https://aclanthology.org/2025.coling-main.452/.

[3] A. Devaraj, W. Sheffield, B. Wallace, J. J. Li, Evaluating factuality in text simplification, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 7331–7345. URL: https://aclanthology.org/2022.acl-long.506/. doi:10.18653/v1/2022.acl-long.506.

[4] A. Hayakawa, S. Bott, H. Saggion, Towards trustworthy lexical simplification: Exploring safety and efficiency with small LLMs, in: Proceedings of the 18th International Natural Language Generation Conference, Association for Computational Linguistics, Hanoi, Vietnam, 2025, pp. 215–231. URL: https://aclanthology.org/2025.inlg-main.15/.

[5] L. Ermakova, H. Azarbonyad, J. Bakker, B. Chakar, G. K. Shahi, J. Kamps, Overview of the CLEF 2026 SimpleText track: Simplify scientific texts (and nothing more), in: M. Hagen, M. Potthast, B. Stein, P. Schaer, E. Zangerle, S. MacAvaney, J. M. Struß, E. Sánchez Salido, A. Barrón Cedeño, A. García Seco de Herrera (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, 2026.

[6] J. Bakker, L. Ermakova, J. Kamps, Overview of the CLEF 2026 SimpleText Task 1: Simplify Scientific Text, in: [26], 2026.

[7] B. Chakar, J. Bakker, L. Ermakova, J. Kamps, Overview of the CLEF 2026 SimpleText Task 2: Identify and Avoid Hallucination, in: [26], 2026.

[8] S. Bickel, P. Haider, T. Scheffer, Predicting sentences using n-gram language models, in: R. Mooney, C. Brew, L.-F. Chien, K. Kirchhoff (Eds.), Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Vancouver, British Columbia, Canada, 2005, pp. 193–200. URL: https://aclanthology.org/H05-1025/.

[9] 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. URL: https://aclanthology.org/Q16-1029/. doi:10.1162/tacl_a_00107.

[10] J. Bakker, B. Vendeville, L. Ermakova, J. Kamps, Overview of the clef 2025 simpletext task 1: Simplify scientific text, Working Notes of CLEF (2025) 3147–3162.

[11] J. Bakker, J. Kamps, Cochrane-auto: An aligned dataset for the simplification of biomedical abstracts, in: M. Shardlow, H. Saggion, F. Alva-Manchego, M. Zampieri, K. North, S. Štajner, R. Stodden (Eds.), Proceedings of the Third Workshop on Text Simplification, Accessibility and Readability (TSAR 2024), Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 41–51. URL: https://aclanthology.org/2024.tsar-1.5/. doi:10.18653/v1/2024.tsar-1.5.

[12] N. Grabar, H. Saggion, Evaluation of automatic text simplification: Where are we now, where should we go from here, in: Y. Estève, T. Jiménez, T. Parcollet, M. Zanon Boito (Eds.), Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale, ATALA, Avignon, France, 2022, pp. 453–463. URL: https://aclanthology.org/2022.jeptalnrecital-taln.47/.

[13] D. Heineman, Y. Dou, W. Xu, Improving minimum Bayes risk decoding with multi-prompt, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA,

2024, pp. 22525–22545. URL: https://aclanthology.org/2024.emnlp-main.1255/. doi:10.18653/v1/2024.emnlp-main.1255.

[14] A. Hayakawa, N. Khallaf, H. Saggion, S. Sharoff, UoL-UPF at TSAR 2025 shared task a generate-and-select approach for readability-controlled text simplification, in: M. Shardlow, F. Alva-Manchego, K. North, R. Stodden, H. Saggion, N. Khallaf, A. Hayakawa (Eds.), Proceedings of the Fourth Workshop on Text Simplification, Accessibility and Readability (TSAR 2025), Association for Computational Linguistics, Suzhou, China, 2025, pp. 193–210. URL: https://aclanthology.org/2025.tsar-1.16/. doi:10.18653/v1/2025.tsar-1.16.

[15] S. Lloyd, Least squares quantization in pcm, IEEE transactions on information theory 28 (1982) 129–137.

[16] G. Team, Gemma 3 technical report, 2025. URL: https://arxiv.org/abs/2503.19786. arXiv:2503.19786.

[17] Q. Team, Qwen3 technical report, 2025. URL: https://arxiv.org/abs/2505.09388. arXiv:2505.09388.

[18] OpenAI, Openai gpt-5 system card, 2026. URL: https://arxiv.org/abs/2601.03267. arXiv:2601.03267.

[19] Google, Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL: https://arxiv.org/abs/2507.06261. arXiv:2507.06261.

[20] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, Z. Liu, Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, arXiv preprint arXiv:2402.03216 4 (2024).

[21] 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.

[22] B. W. Lee, Y. S. Jang, J. Lee, Pushing on text readability assessment: A transformer meets handcrafted linguistic features, in: M.-F. Moens, X. Huang, L. Specia, S. W.-t. Yih (Eds.), Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 10669–10686. URL: https://aclanthology.org/2021.emnlp-main.834/. doi:10.18653/v1/2021.emnlp-main.834.

[23] A. Mohammed, R. Kora, An effective ensemble deep learning framework for text classification, Journal of King Saud University-Computer and Information Sciences 34 (2022) 8825–8837.

[24] S. A. Lee, A. Wu, J. N. Chiang, Clinical modernbert: An efficient and long context encoder for biomedical text, 2025. URL: https://arxiv.org/abs/2504.03964. arXiv:2504.03964.

[25] P. Deka, A. Jurek-Loughrey, D. P, Multiple evidence combination for fact-checking of health-related information, in: The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 237–247. URL: https://aclanthology.org/2023.bionlp-1.20.

[26] E. Sánchez Salido, A. Barrón Cedeño, A. García Seco de Herrera, S. MacAvaney, J. M. Struß (Eds.), Working Notes of CLEF 2026: Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CEUR-WS.org, 2026.

[27] C.-H. Chiang, H.-y. Lee, Can large language models be an alternative to human evaluations?, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 15607–15631. URL: https://aclanthology.org/2023.acl-long.870/. doi:10.18653/v1/2023.acl-long.870.

A. Task 1: Prompts for Candidate Generation

We used three simplification prompts for LLMs. These prompts were based on an inductive approach, which involved extracting simplification features from the training data of Cochrane-auto to create instructions as a prompt. To do this, the following prompt was given to GPT-5.4.

```
You will be given several pairs of paragraphs. Each pair is composed
 of an original text and a simplified version in biomedical domain.
Your task is to analyze these pairs to find the general patterns of
simplification and write an instruction for LLMs to simplify
paragraphs similarly. Include observations on information or
phrasing that remains unchanged. Do not include examples that
contain text parts in given paragraphs. Only output your final
prompt.

Original: {Original medical abstract 1}
Simplified: {Reference lay summary 1}
Original: {Original medical abstract 2}
Simplified: {Reference lay summary 2}
Original: {Original medical abstract 3}
Simplified: {Reference lay summary 3}
Original: {Original medical abstract 4}
Simplified: {Reference lay summary 4}
Original: {Original medical abstract 5}
Simplified: {Reference lay summary 5}
```

After several trials, we picked up following three prompts from the generated outputs with some minor arrangements.

Prompt 1

```
You are given a medical research abstract. Rewrite it as a plain
language summary (PLS) for non-specialist readers, following these
patterns:

- Keep the core study facts, but simplify the wording: number of
studies, participants, setting, intervention, comparator, and main
findings should remain.
- Preserve the overall conclusions and certainty statements,
including uncertainty, low/very low/moderate quality evidence, and "
more research is needed" messages.
- Keep outcome names and main result directions unchanged, but
explain them in simpler terms. Convert technical statistical
language into plain language while retaining whether results were
similar, better, worse, unclear, or not assessed.
- Retain important study design details when relevant, such as
randomized controlled trial, cross-over, ongoing studies, or risk of
 bias, but explain these in simple terms only when they affect
interpretation.
- Mention when no studies were found for a comparison or outcome,
and when outcomes were not assessed.
- Keep disease names, treatment names, and intervention types
unchanged unless a simple explanation is needed for comprehension.
```

- If the abstract says results are uncertain, low-certainty, or very low-certainty, say this plainly and explain that the findings should be interpreted with caution.
- If the abstract compares two treatments, state the comparison clearly and avoid overloading the summary with numerical statistics unless they are essential to preserve the meaning.
- Keep phrases about comparable efficacy, no significant differences, little to no effect, and lack of evidence in the summary when those are the main conclusions.
- Preserve any mention of ongoing studies or the absence of trials for certain comparisons.
- When helpful, add brief clarifications in parentheses for specialized terms, but do not change the scientific meaning.

Simplify the abstract into short, readable paragraphs or sentences. Use direct, neutral language. Do not add new findings or opinions. Provide only PLS without any explanation or justification.

Prompt 2

Write a plain-language summary (PLS) from a medical abstract by simplifying the language while preserving the core findings and key study details.

Follow these principles:
- Keep the main study purpose, number of studies, total participants, comparison groups, and overall conclusions.
- Replace technical wording with everyday language where possible.
- Explain specialized terms briefly in parentheses when they are important for understanding.
- Keep names of interventions, diseases, outcomes, and study designs when they are central to the meaning, but make them easier to read.
- Preserve certainty statements such as low, very low, moderate, or high certainty, and clearly state when evidence is uncertain.
- Preserve whether results show no important difference, a possible benefit, or insufficient evidence.
- Keep important numerical findings only when they help convey the main result; otherwise summarize them more simply.
- Mention ongoing studies, missing evidence, or outcomes that were not assessed when these are important to the review's conclusion.
- If the abstract reports multiple outcomes, group them into a short, readable summary rather than listing every statistic.
- Keep cautious wording from the abstract, especially when evidence is limited, inconsistent, or based on small studies.
- Maintain the overall meaning of risk-of-bias or study-quality statements, but simplify the explanation.

Style guidelines:
- Use short, clear sentences.
- Prefer active, reader-friendly phrasing.
- Avoid jargon, dense statistical language, and long method descriptions unless they are needed to understand the result.

- Write for a general audience with no medical background.
- Do not overstate certainty or clinical impact.
- Do not add information not supported by the abstract.

What should usually stay unchanged in substance:
- The number of studies and participants, if reported prominently.
- The compared treatments or approaches.
- The direction of the main findings.
- The level of certainty in the evidence.
- Key limitations such as few studies, small sample sizes, or lack of long-term data.

Transform the abstract into a concise, accessible PLS that keeps the scientific meaning but is easier for non-specialists to understand. Provide only PLS without any explanation or justification.

Prompt 3

Write a plain-language summary from a medical abstract by preserving the study's core findings, numbers, and conclusions, while making the wording easier for non-specialists to understand.

Simplification rules:
- Keep the same overall meaning, especially the study question, number of studies, number of participants, main comparisons, main outcomes, and the final conclusion about certainty or uncertainty.
- Replace technical terminology with everyday language when possible, but do not remove important medical terms if they are central to the topic; instead, explain them briefly in parentheses the first time they appear.
- Convert statistical language into simpler phrases. Keep effect estimates, confidence intervals, and quality/certainty labels only if they are important to the conclusion; otherwise summarize them as "no clear difference," "may help," "evidence is uncertain," or similar plain-language wording.
- Prefer short, direct sentences and a reader-friendly flow.
- Avoid detailed methodological jargon unless it is necessary for understanding the results. For example, simplify descriptions of trial design, bias tools, and analysis methods into plain language or omit them if they do not change the main message.
- Explain acronyms and abbreviations on first use, or replace them with plain words.
- When the abstract lists several outcomes, group them into a simpler summary and keep only the most important results.
- If the abstract says outcomes were not assessed or no studies were found for a comparison, state that clearly in plain language.
- If the abstract mentions ongoing studies, include this only if it is relevant to the completeness of the evidence.
- Keep statements about uncertainty, low certainty, or lack of strong evidence, and make the uncertainty easy to understand.
- Preserve any important subgroup information, setting, or population details when they help the reader understand who the

```
results apply to.

What usually stays unchanged:
- The number of studies and participants.
- The main intervention and comparison being studied.
- The main direction of results, especially whether there was no
significant difference, a possible benefit, or uncertainty.
- Key outcome names when they are central to the review.
- Final conclusions about whether the intervention is effective,
uncertain, or needs more research.
- Important population, setting, and follow-up information when it
affects interpretation.

Style to aim for:
- Use "we found," "we included," "the studies showed," and "we are
uncertain" in a straightforward way.
- Write for a general audience, as if explaining the review to a
patient, caregiver, or non-specialist reader.
- Do not add new information not supported by the abstract.
- Do not mention the abstract's technical details unless they are
needed to understand the result.

Provide only PLS without any explanation or justification.
```

B. Task 1: Prompts for LLM-as-a-judge

We utilized LLM-as-a-judge framework [27] for filtering candidates as an ablation study. Below is the prompt we used to score the overall quality of each candidate simplification. We converted A, B, C, D, E into 5, 4, 3, 2, 1, respectively, and computed the final score as the weighted average of the next-token probabilities, derived by converting the logits of each score into a probability distribution.

```
Your task is to judge the quality of simplified medical text based
on the original medical abstract. Please evaluate the simplified
text from Very Good to Very Poor.
## Medical Abstract:
{abstract}

## Simplified Text:
{simplified}

## Option:
A) Very Good
B) Good
C) Normal
D) Poor
E) Very Poor

## Answer (One option letter ONLY):
```

C. Task 2: Meta-classifier Settings

C.1. Features

Table 7: Features used for meta-classifier. The average values of each feature in the validation set are also shown.

		Average Values in Val					
Type	Feature	None	LEAK	UNG	GEN	REP	OVER
Heuristic	# of sentences (source)	45.4	41.8	47.1	37.4	43.3	80.3
	# of words (source)	474	514	529	485	490	1006
	# of characters (source)	3007	3391	3503	3232	3282	6170
	# of sentences (output)	2.9	11.7	4.4	8.7	4.6	5.3
	# of words (output)	24	188	46	70	45	64
	# of characters (output)	158	1229	285	464	291	158
	# of "if you"	0.00	0.76	0.01	0.09	0.00	0.00
	# of "this version"	0.00	0.79	0.00	0.03	0.00	0.00
	# of "revised version"	0.00	0.17	0.00	0.03	0.00	0.00
	# of "version"	0.01	1.61	0.01	0.07	0.00	0.00
	# of "plain language"	0.00	0.01	0.00	0.00	0.00	0.00
	# of "simple"	0.00	0.07	0.01	0.00	0.00	0.00
	# of "term"	0.08	1.64	0.11	0.11	0.22	0.00
	# of "rewrit"	0.00	0.01	0.00	0.00	0.00	0.00
	# of "simpli"	0.05	1.42	0.00	0.16	0.00	0.00
	# of "exclamation"	0.00	0.99	0.00	0.14	0.00	0.00
	# of "please"	0.00	0.17	0.00	0.04	0.00	0.00
	# of "IMPORTANT"	0.05	1.09	0.00	0.13	0.00	0.00
	# of "visit"	0.00	0.03	0.10	0.00	0.02	0.00
	# of "would you"	0.00	0.22	0.00	0.01	0.00	0.00
	# of "http"	0.00	0.01	0.03	0.00	0.00	0.00
	# of "www"	0.00	0.02	0.09	0.00	0.02	0.00
	# of "okay"	0.00	0.00	0.00	0.00	0.00	0.00
	# of "note"	0.00	0.06	0.01	0.01	0.00	0.00
	# of "category"	0.00	0.03	0.00	0.00	0.00	0.00
	# of "text"	0.00	0.69	0.00	0.07	0.00	0.00
	# of "input"	0.08	0.01	0.00	1.58	0.00	0.00
	# of "output"	0.07	0.06	0.00	6.74	0.00	0.00
	# of "["	0.00	0.01	0.02	0.03	0.06	0.00
	# of "::"	0.01	0.00	0.00	0.63	0.00	0.00
	starts with "(number)"	0.01	0.01	0.00	0.00	0.00	0.00
	contains "rule (number)"	0.00	0.01	0.00	0.00	0.00	0.00
	starts with "# (number)"	0.00	0.00	0.00	0.00	0.00	0.00
	ends with "..."	0.00	0.00	0.00	0.00	0.00	0.00
	# of unique 1-gram	21.29	100.53	33.95	22.10	27.84	42.67
	# of unique 2-gram	22.48	138.16	41.95	26.80	37.06	55.00
	# of unique 3-gram	21.89	146.56	42.82	29.08	39.72	58.67
	# of unique 4-gram	21.03	150.39	42.20	30.86	39.58	59.67
	max % word overlap btw sents	0.02	0.56	0.36	0.83	0.62	0.51
NLI	% entailment	0.93	0.91	0.94	0.90	0.96	0.99
	% contradiction	0.07	0.08	0.05	0.09	0.04	0.01
	% neutral	0.00	0.01	0.01	0.01	0.00	0.00
	logit for None	1.52	-1.38	-0.51	0.39	0.41	0.85

Table 7: Features used for meta-classifier. The average values of each feature in the validation set are also shown.

Type	Feature	Average Values in Val					
		None	LEAK	UNG	GEN	REP	OVER
BERT	logit for LEAK	-1.76	4.19	-2.01	-0.52	-2.33	-3.07
	logit for UNG	1.41	-0.16	3.05	-1.59	2.32	2.51
	logit for GEN	-2.19	1.12	-2.49	5.59	-2.46	-3.19
	logit for REP	0.03	-2.06	0.41	-1.31	0.63	1.27
	logit for OVER	0.48	-1.89	0.32	-1.63	0.66	0.59
	prob for None	0.38	0.05	0.11	0.01	0.18	0.18
	prob for LEAK	0.10	0.78	0.02	0.04	0.02	0.00
	prob for UNG	0.27	0.09	0.68	0.02	0.52	0.48
	prob for GEN	0.02	0.05	0.00	0.91	0.00	0.00
	prob for REP	0.09	0.02	0.09	0.02	0.14	0.27
	prob for OVER	0.15	0.01	0.11	0.00	0.13	0.07
	predicted index	1.81	2.04	2.80	1.14	2.48	2.33
LLM	logit for None	24.95	25.80	21.31	20.13	21.19	24.62
	logit for LEAK	30.26	25.24	30.77	21.81	28.08	28.62
	logit for UNG	22.52	21.83	22.30	20.89	21.38	21.71
	logit for GEN	22.02	25.15	20.91	20.46	20.78	21.33
	logit for REP	20.54	20.68	20.63	28.91	24.07	19.42
	logit for OVER	19.92	21.13	19.22	22.81	20.41	19.29
	prob for None	0.14	0.43	0.03	0.02	0.03	0.33
	prob for LEAK	0.83	0.22	0.89	0.08	0.64	0.67
	prob for UNG	0.01	0.00	0.01	0.03	0.00	0.00
	prob for GEN	0.01	0.33	0.04	0.03	0.02	0.00
	prob for REP	0.01	0.02	0.03	0.84	0.31	0.00
	prob for OVER	0.00	0.00	0.00	0.00	0.00	0.00
	predicted index	1.76	0.80	2.00	3.64	2.56	1.33

C.2. Meta-classifier

We employed a LightGBM model as the meta-classifier. We trained the model with the following hyper-parameters: n_estimators=1000, learning_rate=0.02, objective='multiclass', class_weight='balanced'. An iteration with the best macro F1 score on the validation set was selected as the final model for prediction on the test set.