Example-Guided Prompting for Document-Level Text Simplification

Ariel Perstin
SCE, Beer Sheva, Israel
arielperstin10@gmail.com

Ilan Shtilman
SCE, Beer Sheva, Israel
shtilmanilan@gmail.com

Michael Färber
ScaDS.AI, TUD
Dresden, Germany
michael.faerber@tu-dresden.de

Marina Litvak
SCE, Beer Sheva, Israel
ScaDS.AI, TUD
Dresden, Germany
marinal@sce.ac.il

Abstract

Document-level text simplification requires large language models (LLMs) to rewrite complex documents while preserving meaning, readability, and discourse coherence. Although prompt-based LLMs have shown promising performance, they often produce inconsistent simplifications because textual instructions alone provide limited guidance for complex document-level transformations. We investigate whether retrieved document-simplification examples can improve document-level generation by augmenting prompts with examples selected from a parallel simplification corpus. This example-guided prompting approach enables LLMs to exploit relevant simplification patterns without task-specific fine-tuning. Experiments on the OneStopEnglish corpus using multiple state-of-the-art LLMs show that incorporating retrieved examples generally improves simplification quality over prompt-only generation and achieves competitive or superior performance compared with representative supervised (T5) and planning-based (PlanSimp) document simplification systems. Furthermore, we find that the benefits of example-guided prompting vary across LLMs, suggesting that effective use of retrieved examples depends on a model's ability to integrate contextual information during generation.

1 Introduction

Large language models (LLMs) have become an effective approach for text simplification, enabling high-quality rewriting through instruction following and in-context learning without task-specific training (Brown et al., 2020; Liu et al., 2023; Cohen et al., 2026). While recent research has primarily focused on sentence-level simplification, many practical applications, such as educational resources, news articles, and technical documents, require simplifying entire documents while preserving semantic fidelity, discourse coherence, and an appropriate level of readability. Compared with sentence-level rewriting, document-level simplification requires coordinated transformations across multiple sentences while maintaining a consistent simplification style throughout the document.

Prompt-based LLMs provide a flexible solution, yet textual instructions alone often provide insufficient guidance for complex document-level transformations, leading to inconsistent simplification quality (Cohen et al., 2026; Maddela et al., 2023; Alfear et al., 2024). This raises the question of whether retrieved examples can provide more effective guidance than textual instructions alone.

Retrieval-Augmented Generation (RAG) has successfully improved many generation tasks by incorporating relevant external context during inference (Lewis et al., 2020). Unlike knowledge-intensive applications, document-level simplification requires guidance on how to transform text rather than additional factual information. We therefore investigate retrieval of document–simplification examples.

To this end, we propose **Example-Guided Prompting (EGP)**, an inference-time approach that augments a strong prompt with retrieved examples selected from a parallel simplification corpus. Instead of relying solely on textual instructions, the model is exposed to examples illustrating how similar documents have previously been simplified, enabling it to exploit relevant simplification patterns without additional training or parameter updates.

We evaluate EGP on the OneStopEnglish corpus (Vajjala and Lučić, 2018) using multiple state-of-the-art LLMs. Building on a strong prompt formulation, we investigate whether retrieved examples improve document-level simplification compared with prompt-only generation. We further compare EGP with two representative document simplification baselines: T5 (Raffel et al., 2020) and PlanSimp, a document-level planning approach pro-

Proceedings of the 19th International Natural Language Generation Conference: Short Papers, pages 9–15
October 17-21, 2026 ©2026 Association for Computational Linguistics

posed by (Cripwell et al., 2023), using automatic measures of simplification quality (Xu et al., 2016), semantic preservation (Maddela et al., 2023; Zhang et al., 2019), and readability (Flesch, 1948; Kincaid et al., 1975).

Our experiments demonstrate that EGP improves prompt-only generation for most of the evaluated models, achieves competitive or superior performance compared with representative state-of-the-art simplification systems, and reveals that the effectiveness of example-guided prompting depends on the underlying LLM.

Our contributions are summarized as follows: (1) We present a systematic study of Example-Guided Prompting (EGP) for document-level text simplification, examining whether semantically similar complex-simple demonstrations can improve the performance of different LLMs without task-specific training[1]. (2) We demonstrate that EGP generally improves prompt-only generation across several state-of-the-art LLMs and achieves competitive or superior performance compared with representative supervised and planning-based document simplification systems. (3) We provide an empirical analysis of when EGP is most effective, showing that its benefits depend on the underlying LLM and involve a trade-off between semantic preservation and readability.

2 Background

Text simplification aims to transform complex text into simpler versions while preserving meaning and improving accessibility for diverse audiences (Siddharthan, 2006; Xu et al., 2015). Early approaches relied on lexical substitution, syntactic rewriting, statistical machine translation (Siddharthan, 2006; Pavlick and Callison-Burch, 2016; Xu et al., 2016), and later supervised neural models (Martin et al., 2020; Sheang and Saggion, 2021). More recently, large language models (LLMs) have enabled instruction-based simplification without task-specific fine-tuning, providing a flexible alternative to supervised approaches (Brown et al., 2020; Liu et al., 2023; Cohen et al., 2026). However, prompt-based simplification remain highly sensitive to prompt formulation and often produce inconsistent simplifications, motivating research on improving the controllability of LLM-based simplification (Cohen et al., 2026; Maddela et al., 2023; Alfear et al., 2024).

Prior work has demonstrated that the choice and organization of in-context examples can substantially influence model performance (Liu et al., 2022; Lu et al., 2022; Rubin et al., 2022). Consequently, several studies have investigated retrieval-based demonstration selection, including approaches that select examples based on their relevance or semantic similarity to the target input (Liu et al., 2022; Rubin et al., 2022; Margatina et al., 2023). Unlike RAG applications (Parvez et al., 2021; Zhao et al., 2026, 2025; Cai, 2025), where generation is improved by incorporating relevant external information during inference (Lewis et al., 2020), these works use retrieval for example-based transformation guidance rather than factual grounding. Building on this established paradigm, we investigate retrieval-based example selection specifically for document-level text simplification, using complex–simple text pairs as demonstrations and systematically evaluating its effect across different LLMs and prompting strategies.

3 Example-Guided Prompting

Figure 1 illustrates the EGP pipeline. Given a complex input document, EGP retrieves semantically similar document-simplification examples and incorporates them into the prompt together with the target document. The resulting prompt is passed to the LLM, which generates the simplified output.

4 Experiments

4.1 Experimental Setup

We evaluate the proposed approach on the OneStopEnglish corpus (Vajjala and Lučić, 2018), using the advanced-to-elementary simplification setting.

Experiments are conducted using five state-of-the-art LLMs representing both commercial and open-weight models: GPT-4o-mini, Claude Haiku 4.5, Gemini 2.0 Flash, Sonar, and Llama 3.2.

As a preliminary experiment, we compare three prompting strategies (zero-shot, structured, and constraint-based) and select the best-performing prompt for all subsequent experiments. We then compare prompt-only generation with EGP.

For EGP, we retrieve the top-3 most similar document–simplification examples from a parallel simplification corpus using dense retrieval. The retrieval component uses dense embeddings and cosine similarity to identify semantically similar

[1]Our code and data are available at https://github.com/arielperstin10/LLM-simplification-PE.

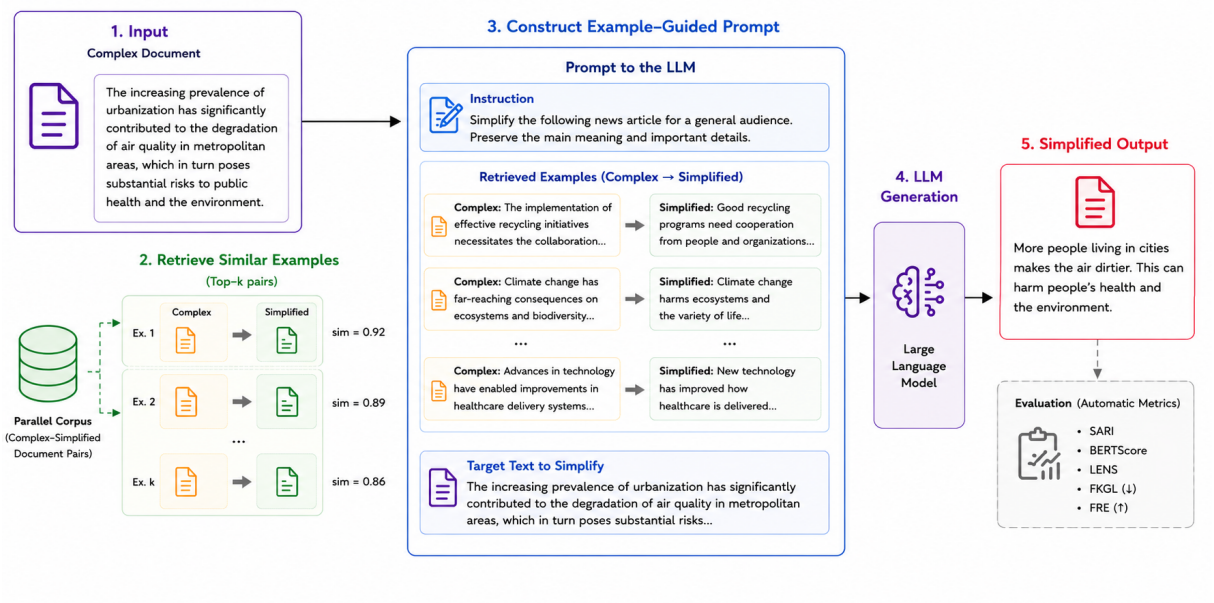


Figure 1: Overview of the proposed EGP framework. Given a complex input document, the system retrieves semantically similar document–simplification examples from a parallel corpus and incorporates them into the prompt together with the target document. The LLM uses these examples as guidance for document-level simplification, and the generated output is evaluated using automatic measures of simplification quality, semantic preservation, and readability.

documents. The evaluated document is excluded from the retrieval pool to prevent data leakage.

We evaluate the generated simplifications using complementary automatic metrics that capture the principal objectives of document-level text simplification. SARI (Xu et al., 2016) measures simplification quality, BERTScore (Zhang et al., 2019) and LENS (Maddela et al., 2023) assess semantic preservation, and FKGL (Kincaid et al., 1975) together with FRE (Flesch, 1948) quantify readability. This combination enables us to analyze the trade-off between producing simpler text and preserving the original meaning.

We additionally compare our approach with representative supervised (T5[2]) and planning-based (PlanSimp) simplification systems (Raffel et al., 2020; Cripwell et al., 2023).

4.2 Prompt Selection

As shown in Appendix Table A1, constraint-based prompting consistently achieves the highest SARI while maintaining competitive semantic preservation and readability across all evaluated models. Consequently, we adopt this prompting strategy as the prompt-only baseline in subsequent experiments.

[2]T5-large-text-simplification

4.3 Effect of EGP

First, we examine whether EGP improves prompt-only generation. Table 1 compares prompt-only generation with the proposed EGP approach. Incorporating retrieved document-simplification examples generally improves simplification quality for four of the five evaluated LLMs. Claude Haiku 4.5 achieves the largest improvement, increasing the SARI score from 40.78 to 44.84 while simultaneously improving semantic preservation measured by BERTScore and LENS. Similar improvements are observed for Gemini 2.0 Flash and Sonar, whereas GPT-4o-mini shows a smaller but consistent gain.

Second, we compare EGP against representative supervised and planning-based simplification systems. EGP consistently achieves competitive or superior performance. Claude Haiku 4.5 + EGP obtains the highest SARI score (44.84), exceeding both PlanSimp (42.02) and T5 (30.0). Gemini 2.0 Flash also outperforms PlanSimp, while Sonar reaches comparable performance without supervised training or explicit planning. The superiority of EGP is even more evident in semantic preservation. GPT-4o-mini and Claude Haiku 4.5 substantially outperform both T5 and PlanSimp according to LENS, improving over the planning-based baseline by more than 20 points. These results indicate

Table 1: Comparison between the prompt-only baseline and EGP. Each cell reports the EGP result, with the change relative to the prompt-only baseline shown in parentheses. Positive values indicate improvement for SARI, BERTScore, LENS, and FRE, while negative values indicate improvement for FKGL.

Model	**SARI**	**BERTScore**	**LENS**	**FKGL↓**	**FRE↑**
T5	30.0	0.86	46.89	1.33	-3.11
PlanSimp	42.02	0.89	48.63	-4.66	16.96
GPT-4o-mini	43.78 (+0.39)	0.89 (+0.00)	**72.38** (+1.28)	-4.28 (+0.34)	15.85 (-1.12)
Gemini 2.0 Flash	42.51 (+3.89)	0.89 (+0.02)	64.93 (+4.20)	-7.83 (+0.55)	**31.69** (-2.20)
Sonar	42.29 (+1.36)	0.89 (+0.02)	49.65 (+10.09)	-6.23 (+0.62)	21.33 (-3.15)
Claude Haiku 4.5	**44.84** (**+4.06**)	**0.90** (+0.02)	70.72 (**+11.62**)	-5.13 (+1.67)	16.75 (-8.84)
Llama 3.2	28.53 (-5.78)	0.79 (-0.07)	35.92 (-36.72)	2.38 (+6.37)	-15.93 (-30.95)

that retrieved examples help preserve document meaning considerably better than existing supervised or planning-based simplification approaches. These results are particularly notable because both T5 and PlanSimp are task-specific document simplification systems, whereas EGP relies solely on inference-time prompting with retrieved examples.

The impact of retrieved examples is model-dependent. While Claude Haiku 4.5, Gemini 2.0 Flash, and Sonar effectively exploit retrieved simplification examples, Llama 3.2 experiences a noticeable performance degradation. This suggests that successful EGP depends not only on retrieval quality but also on an LLM's ability to integrate contextual examples during generation.

We also observe a trade-off between simplification and semantic preservation. EGP generally improves semantically oriented metrics, particularly BERTScore and LENS, but does not always yield the largest readability gains as measured by FKGL and FRE. Retrieved examples appear to encourage simplifications that remain closer to the source document while preserving more of its meaning, highlighting the need to balance readability and semantic fidelity in document-level simplification.

5 Discussion

Our results indicate that EGP improves prompt-only generation for most of the evaluated models, without requiring task-specific fine-tuning.

An important finding of this study is that the effectiveness of EGP is highly model-dependent. While Claude Haiku 4.5, Gemini 2.0 Flash, Sonar, and GPT-4o-mini successfully exploit retrieved examples, Llama 3.2 exhibits a substantial performance degradation across all evaluation metrics. This observation suggests that retrieval quality alone is insufficient; successful example-guided generation also depends on an LLM's ability to integrate retrieved examples into the generation process. Understanding which architectural or training characteristics enable this capability remains an important direction for future research.

Our experiments also reveal an interesting trade-off between readability and semantic preservation. Although EGP consistently improves SARI, BERTScore, and LENS, it generally produces smaller readability gains measured by FKGL and FRE. This suggests that retrieved examples encourage the model to generate more conservative simplifications that remain closer to the source document. Rather than aggressively simplifying the input, the model appears to prioritize preserving meaning and discourse structure while applying transformations that resemble those observed in retrieved examples. Consequently, EGP shifts the balance toward higher semantic fidelity at the cost of slightly smaller readability improvements.

Our findings suggest that retrieved examples can serve not only as a source of external knowledge, as in conventional RAG, but also as a mechanism for guiding complex text generation tasks. This perspective may extend beyond text simplification to other controlled generation problems, where examples illustrate desirable transformations rather than provide factual information.

6 Conclusion

We investigate an example-guided prompting approach for document-level text simplification. Experiments demonstrate that EGP generally improves prompt-only generation across the evaluated models, while achieving competitive or superior performance compared with representative supervised and planning-based baselines. More broadly, our findings suggest that retrieved examples provide an effective mechanism for controlling text generation through transformation guidance rather than factual grounding.

An important next step is to complement the current automatic evaluation with human assessment of simplicity, meaning preservation, fluency, and

usefulness to target readers. We also plan to investigate alternative demonstration-selection strategies, the number and diversity of retrieved examples, and evaluation across additional simplification datasets and domains.

Limitations

This work has several limitations that should be considered when interpreting the results.

First, the experiments were conducted using a single benchmark dataset, OneStopEnglish (Vajjala and Lučić, 2018), which contains educational news articles written in English. Although this corpus is widely used for evaluating document-level text simplification, the observed improvements may not directly generalize to other domains, languages, or writing styles.

Second, our evaluation relies primarily on automatic metrics. While SARI, BERTScore, LENS, FKGL, and FRE capture complementary aspects of simplification quality, they cannot fully reflect human judgments of readability, coherence, factual consistency, or usefulness for different target audiences. Human evaluation is therefore necessary to better understand how retrieved examples influence the overall quality and usability of generated simplifications.

Third, the proposed approach retrieves examples from a fixed parallel simplification corpus using a single retrieval configuration. Different retrieval strategies, retrieval corpus sizes, similarity measures, or example selection mechanisms may substantially influence the effectiveness of EGP. In particular, the relationship between retrieval quality and generation quality deserves further investigation.

Finally, the effectiveness of EGP was shown to vary considerably across LLMs. While several instruction-following models consistently benefited from retrieved examples, one model exhibited substantial performance degradation. Understanding why some models integrate retrieved examples more effectively than others remains an open research question and may provide valuable insights into the interaction between retrieval and large language model generation.

Although the proposed approach focuses on document-level text simplification, the idea of guiding generation through retrieved examples may also be applicable to other controlled text generation tasks. Evaluating the generality of this paradigm across different applications represents an important direction for future work.

References

Noof Abdullah Alfear, Dimitar Kazakov, and Hend Al-Khalifa. 2024. Meta-evaluation of sentence simplification metrics. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11229–11235.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yunzhen Cai. 2025. A comprehensive study on code completion for large language models. In *2025 8th International Conference on Advanced Algorithms and Control Engineering (ICAACE)*, pages 2506–2510. IEEE.

Eilam Cohen, Itamar Bul, Danielle Inbar, and Omri Loewenbach. 2026. Simplify-this: A comparative analysis of prompt-based and fine-tuned llms. *arXiv preprint arXiv:2601.05794*.

Liam Cripwell, Joël Legrand, and Claire Gardent. 2023. Document-level planning for text simplification. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 993–1006.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

J. Peter Kincaid, Robert P. Fishburne, Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical Report Research Branch Report 8-75, Naval Technical Training Command, Millington, TN.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd workshop on knowledge extraction and integration for deep learning architectures*, pages 100–114.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM computing surveys*, 55(9):1–35.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098.

Mounica Maddela, Yao Dou, David Heineman, and Wei Xu. 2023. 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)*, pages 16383–16408.

Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. 2023. Active learning principles for in-context learning with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5011–5034.

Louis Martin, Éric Villemonte De La Clergerie, Benoît Sagot, and Antoine Bordes. 2020. Controllable sentence simplification. In *Proceedings of the twelfth language resources and evaluation conference*, pages 4689–4698.

Md Rizwan Parvez, Wasi Ahmad, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. 2021. Retrieval augmented code generation and summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2719–2734.

Ellie Pavlick and Chris Callison-Burch. 2016. Simple ppdb: A paraphrase database for simplification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 143–148.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 2655–2671.

Kim Cheng Sheang and Horacio Saggion. 2021. Controllable sentence simplification with a unified text-to-text transfer transformer. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 341–352.

Advaith Siddharthan. 2006. Syntactic simplification and text cohesion. *Research on Language and Computation*, 4(1):77–109.

Sowmya Vajjala and Ivana Lučić. 2018. Onestopenglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 297–304.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, Jie Jiang, and Bin Cui. 2026. Retrieval-augmented generation for AI-generated content: A survey. *Data Science and Engineering*, pages 1–29.

Xiuyuan Zhao, Tiejiang Sun, Shaochen Ren, Jingyun Yang, and Yang Liu. 2025. Rag-based ai agents for enterprise software development: Implementation patterns and production deployment. *Frontiers in Artificial Intelligence Research*, 2(3):501–520.

A Example Appendix

A.1 Prompts

A.1.1 Zero-shot Prompt Template

You are given a news article written at an advanced reading level.

Your task is to rewrite the text in simpler language for an elementary-level reader.

Preserve the original meaning and main ideas.

Use clear and short sentences.

Text:

`{INPUT_TEXT}`

A.1.2 Structured Prompt Template

You are given a news article written at an advanced reading level.

Your task is to rewrite the text for an elementary-level reader by following these steps:

Table 2: Results for different prompt strategies.

Model	Strategy	SARI	BERTScore	$FKGL_{\Delta}$	FRE_{Δ}	LENS
Llama 3.2	Zero-shot	34.04	0.85	-3.41	14.59	73.91
	Structured	34.13	0.85	-3.02	11.26	64.72
	Constraint	34.20	0.85	-4.19	16.75	69.45
Gemini 2.0 Flash	Zero-shot	33.82	0.85	-7.42	31.08	67.58
	Structured	34.01	0.85	-5.76	23.71	64.22
	Constraint	38.14	0.87	**-8.52**	**34.78**	57.53
GPT-4o-mini	Zero-shot	39.66	0.88	-3.80	15.00	74.65
	Structured	37.47	0.87	-3.73	14.08	72.34
	Constraint	**43.52**	**0.89**	-4.93	18.16	69.07
Sonar	Zero-shot	37.45	0.87	-6.81	26.88	54.90
	Structured	35.44	0.85	-6.69	25.95	52.55
	Constraint	39.89	0.86	-6.80	23.46	31.25
Claude Haiku 4.5	Zero-shot	35.45	0.86	-5.30	19.73	55.23
	Structured	36.26	0.86	-5.13	17.96	51.71
	Constraint	39.78	0.87	-6.58	23.53	53.67

1. Identify the main ideas in the text.
2. Remove or shorten secondary details that are not essential.
3. Replace complex or technical words with simpler alternatives.
4. Split long or complex sentences into shorter ones.
5. Verify that all original facts and main ideas are preserved.

Text:

`{INPUT_TEXT}`

A.1.3 Constraint Prompt Template

You are given a news article written at an advanced reading level.

Your task is to rewrite the text for an elementary-level reader under the following constraints:

- Do not add, remove, or change factual information.
- Keep all names, places, organizations, and numbers exactly as in the original text.
- Use common, high-frequency words suitable for an elementary-level reader.
- Limit sentence length to a maximum of 12–15 words.
- Prefer simple and direct sentence structures.

Text:

`{INPUT_TEXT}`