Plan-Guided Text Simplification with Extended Contexts

Pascal Mathas, Jan Bakker, Jaap Kamps
Institute for Logic, Language and Computation (ILLC)
University of Amsterdam
Amsterdam, The Netherlands
pascal.mathas@student.uva.nl, j.bakker@uva.nl, kamps@uva.nl

Abstract

In this paper, we investigate the impact of increasing context lengths (one to five paragraphs) on plan-following accuracy in plan-guided text simplification. Plan-guided models simplify text according to sentence-level operation labels such as copy, rephrase, split, and delete. Previous work fine-tunes BART with target reading-level and sentence-level operation tokens to perform this task. We find that BART's plan-following accuracy on Newsela-auto drops significantly as context increases from one to five paragraphs. This means that the model becomes less reliable with longer contexts, and the quality of its outputs decreases. To address this, we propose replacing the fine-tuned BART models with a prompting-based approach using instruction-tuned Qwen models. We find that this approach not only maintains robust plan-following across all context lengths, but even at the longest context length still exceeds BART's performance at the shortest. We further provide ablations on model size and model family, showing that a minimum model capacity is required for the approach to work and that it transfers across LLM families.

Keywords: text simplification, plan-guided generation, long-context, large language models, BART

1. Introduction

Plan-guided simplification models, as proposed by Cripwell et al. (2023b), simplify text according to a plan. This plan consists of target reading levels and sentence-based operation labels. The model receives in its input the target reading level token, plus sentence-level operation tokens for each sentence. These tokens include <*copy*>, <*rephrase*>, <*split*>, and <*delete*>, and are inferred from references (oracle) or by a separate planning model designed to predict the appropriate token for each sentence (Cripwell et al., 2023b).

In this work, we show that fine-tuned plan-guided BART simplification models, introduced by Cripwell et al. (2023b), degrade in their plan-following ability as the context length increases. Since document simplification requires broader context to preserve discourse structure and handle multi-sentence operations (Alva-Manchego et al., 2019; Cripwell et al., 2023a), this degradation therefore exposes a limitation in the fine-tuned BART models. We investigate whether this gap can be addressed by replacing the fine-tuned BART models with prompting instruction-tuned large language models (LLMs).

The main contributions of our paper are as follows:

- We show that fine-tuned plan-guided BART models degrade in their plan-following accuracy as context increases.
- We show that a prompting-based approach using instruction-tuned LLMs maintains both plan-following accuracy and simplification quality at extended context lengths.

We make all code, prompts, and settings publicly available at github.com/pascalmathas/plan_simp_extended.

2. Related Work

Sentence alignment. Jiang et al. (2020) propose a neural Conditional Random Field (CRF) alignment model that leverages the sequential structure of sentences in parallel documents, combined with fine-tuned BERT to capture semantic similarity. The model aligns simple sentences to complex sentences, after which simplification operation labels can be inferred from the alignments (copy, rephrase, split, delete). In their work, the authors introduce the Newsela-auto and Wiki-auto datasets. In our work, we use the Newsela-auto dataset to train and evaluate our models, and the neural CRF aligner to align our generations back to the complex sentences.

Plan-guided simplification. The baseline BART planning model we use in this paper is adapted from Cripwell et al. (2023b). The authors introduce two models: a planning model that predicts a sequence of sentence-level operations and a generation model conditioned on these plans. They show that their plan-guided system outperforms other end-to-end approaches. In Cripwell et al. (2023a), the authors extend this work by using broader contextual information during generation. In this work, we show that plan-guided BART models degrade in plan-following ability as input context increases. Due to some missing details in the pipeline of Cripwell et al. (2023b), we follow the code from a re-

Proceedings of the Joint Workshop on Readability and Text Simplification (READIxTSAR) @ LREC 2026, pages 121–129
11 May 2026.

production of those works by Bakker and Kamps (2024).

Prompting strategies for LLM-based simplification. Finally, to address the degradation observed in BART models, we replace them with LLMs. Our approach is loosely inspired by Papandreou et al. (2025), where the authors investigate several prompting strategies. We adapt their codebase as a starting point for our LLM implementation, although our prompting strategy and focus differ.

3. Task & Experimental Setup

3.1. Data

The data used for our experiments is the Newsela-auto dataset from Jiang et al. (2020), based on the Newsela corpus by Xu et al. (2015). Each of the $1,882$ news articles in the corpus is manually rewritten at five different simplification levels. Our preprocessing approach follows that of Cripwell et al. (2023b). Since this paper only requires complex-to-simple article pairs, we pair each article version with every version corresponding to a simpler reading level, resulting in a total of $18,820$ article pairs. We split the data into training, validation, and test sets of 92.5%, 2.5%, and 5.0%, respectively.

Table 1 shows the dataset statistics of Newsela-auto after preprocessing, where $|c_i|$ is the average token length of a complex sentence, $|s_i|$ is the average token length of a simple sentence, n and k are the average number of sentences in the complex and simple documents, and p is the average number of sentences per paragraph.

	Newsela-auto
# Doc Pairs	18,820
# Para Pairs	478,479
# Sent Pairs	960,365
Avg. $\|c_i\|$	17.14
Avg. $\|s_i\|$	12.84
Avg. n	51.03
Avg. k	43.25
Avg. p	2.01

Table 1: Statistics of the Newsela-auto dataset after preprocessing, where n is # sentences in C, and k is # sentences in S and p is # sentences per paragraph in C.

Paragraphs in the Newsela-auto dataset are relatively short, containing roughly two sentences on average. This underlines the importance of evaluating on multiple input paragraphs, as real-world text is likely to contain longer paragraphs.

We infer the operation labels based on the already aligned sentences in the Newsela-auto dataset. A complex sentence is labeled as *delete* if it has no aligned counterpart, *split* if it aligns to multiple simplified sentences, and *rephrase* if it aligns to a single, different sentence. Sentences with a Levenshtein similarity of ≥ 0.92 to their aligned counterpart are labeled as *copy*. Table 2 shows the distribution of operation labels in the Newsela corpus. Our distribution differs slightly from that of Cripwell et al. (2023b), as we did not have access to their filtered version and thus used the unfiltered variant.

Data	**Copy**	**Rephrase**	**Split**	**Delete**
Newsela-auto	20.14	27.21	16.69	35.95

Table 2: Operation class distributions of Newsela-auto in percentages.

Finally, to create the paragraph-level datasets for $i = 1 \ldots 5$, each document in the data splits is divided into chunks of i paragraphs. If a document is not evenly divisible, chunks of size $i-1$ are used to distribute the remainder evenly. For example, if a document contains 7 paragraphs, then for $i = 3$, the document is chunked into paragraph groups of 3–2–2.

3.2. Planning Models

To assess how the models would perform in a real-world scenario, we also evaluate them with predicted labels instead of oracle labels. For this, we use the liamcripwell/pgdyn-plan model checkpoint made available on Huggingface, which was part of Cripwell et al. (2023a) and trained on the oracle labels. As the model checkpoint is the context-aware variant of the planning model, we first generate context representations of the test documents. We then use these contexts when predicting operation labels for the test set.

3.3. Evaluation

We evaluate the simplifications produced by our models in two ways. First, we assess simplification quality at the document level. Second, we evaluate the models' ability to follow the operation labels when simplifying sentences.

3.3.1. Alignment

To align the generated simplifications with the complex sentences, we use the neural CRF alignment model of Jiang et al. (2020). Instead of training the aligner ourselves, we use a Wiki-auto pretrained checkpoint from Bakker and Kamps (2024). The checkpoint is available on GitHub: aligner checkpoint. We align the full outputs to the inputs instead

of aligning each paragraph pair separately. This means that we group the paragraphs back into the full documents using their `pair_id`.

3.3.2. Document-Level Metrics

After generating simplifications with the models, we evaluate them by regrouping the paragraphs into documents and calculating several document-level metrics. We use two reference-based metrics: (a) SMART, which measures semantic similarity between the output and reference using sentence embeddings (Amplayo et al., 2023), and (b) SARI, which assesses simplification quality by comparing n-gram operations (additions, deletions, and unchanged) between the output, source, and reference (Xu et al., 2016). Additionally, we use the reference-free Flesch-Kincaid Grade Level (FKGL), which measures text readability (Kincaid et al., 1975).

3.3.3. Plan-Following (Micro-Recall)

To assess whether the models follow the operator labels, we first align the generated simplifications back to the complex sentences. We then assign operator labels using the same procedure described in Section 3.1. Once the labels for the generated simplifications are obtained, we compare them to the reference labels to calculate the percentage of sentences that were simplified according to the oracle plan. We refer to this metric as micro-recall, the percentage of sentences simplified according to the oracle plan.

3.4. Computational Requirements

Table 3 presents the computational requirements for our experiments. The reported time estimates correspond to all five runs together, that is, for all paragraph-level datasets $i = 1 \ldots 5$. Micro-recall time includes generation, alignment, and micro-recall computation. Generating with the LLMs is significantly faster due to the vLLM implementation.

Model	GPU	Training	Micro-recall	Doc-eval
BART (all variants)	A100	∼6h	∼12h	∼2h
Qwen-7B & Qwen-14B	A100	–	∼9h	∼2h
Qwen-32B	H100	–	∼9h	∼2h

Table 3: Computational requirements for training and inference.

3.5. Reproducibility

All fine-tuning experiments in this paper are run with a fixed random seed (42) and deterministic Torch and CUDA behavior to ensure reproducibility.

For a detailed overview of all parameters for every run, refer to the *README* and *shell scripts* in our GitHub repository.

4. Problem: BART Model Variants Degrade in Plan-Following as Context Increases

4.1. Baseline Model Setup

We train a total of 10 BART models: 5 with oracle labels (plan-guided) and 5 without. Both groups are trained on paragraph-level datasets with $i = 1 \ldots 5$. We refer to the BART models with oracle labels as O-BART$_{\text{para-}i = 1 \ldots 5}$ and the models without oracle labels as BART$_{\text{para-}i = 1 \ldots 5}$.

Both groups follow the same training procedure. The base model we fine-tune is facebook/bart-base (Lewis et al., 2019). Training is performed using the Adam optimizer with a learning rate of 2×10^{-5} and an effective batch size of 16 (batch size 8 with gradient accumulation of 2). We use mixed precision (FP16) and early stopping with a patience of 1. For both models, target reading level tokens are prepended to each paragraph, along with operation tokens for O-BART$_{\text{para}}$ following Cripwell et al. (2023b). During inference, we use beam search with a beam size of 5.

We also evaluate the performance of the BART models with predicted operation labels from the planning model. For this, we use the BART models trained with oracle labels, but during inference, we use the predicted labels. We refer to these models as $\hat{\text{O}}$-BART$_{\text{para-}i = 1 \ldots 5}$.

4.2. Baseline Analysis

Figure 1 showcases the micro-recall for the BART model variants across paragraph input lengths of 1 to 5.

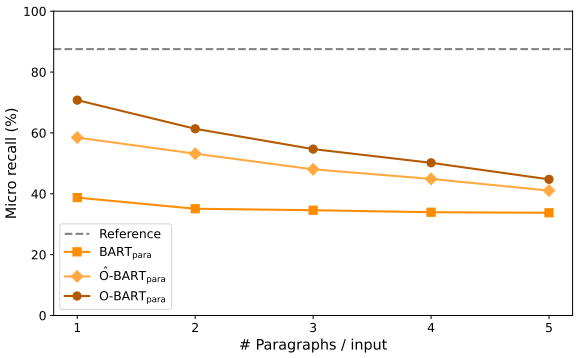

Figure 1: Plan-following micro recall for BART models across context sizes on Newsela-auto.

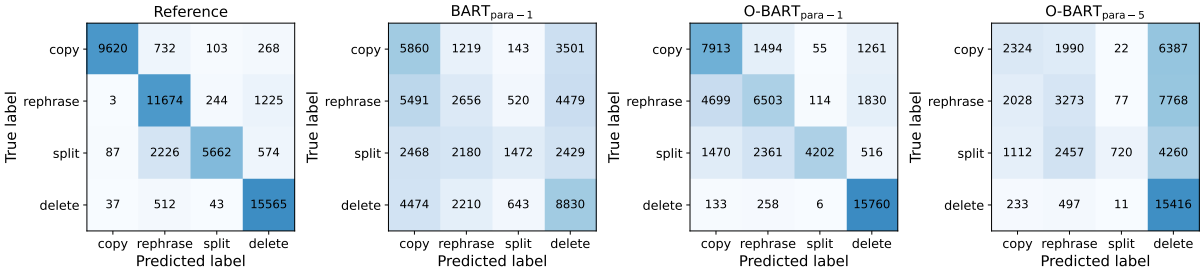

Figure 2: Confusion matrices comparing the ground-truth (oracle) operation labels against the operation labels predicted by the model for input documents, evaluated on the Newsela-auto dataset.

O-BART achieves the highest plan-following accuracy at paragraph level 1 (70.8%) but degrades as context increases, dropping to 44.7% at paragraph level 5, a loss of 26.1 percentage points. Ô-BART follows a similar decline, starting at 58.5% and falling to 41.0%, being limited by the planning model. The unguided BART baseline remains relatively flat (38.7% to 33.8%), showing that without plans the model defaults to a standard set of operations. Interestingly, at paragraph level 5, all models are relatively close to each other, indicating that increased context degrades the models' performance to almost baseline levels.

This becomes more apparent when we look at the matrices in Figure 2, where we compare the oracle labels (y-axis) to the labels predicted by the model (x-axis). For paragraph level 1, O-BART follows the copy and delete operations somewhat reliably but already struggles with rephrase and split. At paragraph level 5, copy recall drops from 73.8% to 30.9%, rephrase from 49.5% to 33.1%, and split from 49.2% to 13.7%. Nearly half of all copy and rephrase sentences are incorrectly deleted, indicating that as context grows, the model increasingly defaults to deletion rather than executing the intended operation.

In Table 4, we can observe the document-level simplification results for the BART model variants.

System	SMART ↑			FKGL ↓	SARI ↑	Length	
	P	R	F1			Tok.	Sent.
Input	58.9	64.9	61.5	7.61	19.5	1089.9	49.1
Reference	100	100	100	4.62	100	721.1	45.2
BART$_{\text{para-1}}$	57.7	53.3	55.1	6.15	38.7	712.1	37.8
O-BART$_{\text{para-1}}$	61.5	60.6	60.8	6.02	45.3	807.7	40.7
Ô-BART$_{\text{para-2}}$	60.8	54.6	57.3	6.96	44.6	847.9	33.0
Ô-BART$_{\text{para-3}}$	58.8	50.9	54.4	7.93	43.3	946.7	29.2
Ô-BART$_{\text{para-4}}$	59.7	45.8	51.6	8.25	42.3	781.6	23.5
Ô-BART$_{\text{para-5}}$	59.1	40.7	47.9	9.17	40.7	667.9	18.8
O-BART$_{\text{para-1}}$	66.1	61.0	63.4	5.93	50.3	709.8	36.5
O-BART$_{\text{para-2}}$	65.4	55.0	59.7	6.89	48.4	744.7	29.7
O-BART$_{\text{para-3}}$	63.0	51.3	56.5	7.80	47.1	826.0	26.3
O-BART$_{\text{para-4}}$	63.8	46.2	53.4	8.18	45.2	689.3	21.2
O-BART$_{\text{para-5}}$	62.9	41.0	49.4	8.99	42.6	588.5	17.0

Table 4: Results of document simplification of BART models on Newsela-auto.

We can see that the operation labels help the model produce better simplifications, as the Ô-BART models achieve higher scores than base BART across the board. The delete bias at higher paragraph levels, however, can also be seen here, as the number of output sentences drops drastically. FKGL worsens with increased paragraph input for both variants, surpassing even the input FKGL. As the model produces fewer sentences, these sentences are more complicated and longer than the input.

5. Solution: Prompting Instruction-tuned LLMs

5.1. Model & Inference

To explore whether prompting instruction-tuned LLMs results in better plan-following accuracy at extended context sizes, we replace the fine-tuned BART model with a prompting-based approach. For the prompting-based approach, we use Qwen2.5-7B-Instruct, Qwen2.5-14B-Instruct, and Qwen2.5-32B-Instruct (Team, 2024; Yang et al., 2024). Our setup is inspired by Papandreou et al. (2025), but we introduce several modifications. First, we utilize the vLLM (Kwon et al., 2023) library for increased inference speed and memory efficiency. Second, we adjust the prompt so the LLM knows to follow the four operation labels when simplifying. All models are run in bfloat16, and for generation we set the temperature to 0.2, top-p to 0.9, repetition penalty to 1.1, and max new tokens to 2048.

As a small ablation to assess the generalizability of our prompting-based pipeline to a different LLM family, we also run google/Gemma-3-27B-Instruct (Gemma-27B) (Team, 2025) at paragraph levels 1 and 5, using the same parameters as the Qwen models.

5.2. Prompting Strategies

Similar to the BART models, the LLM receives in each user prompt the target reading level to simplify to and, before each sentence, the operation labels indicating what it should do with that sentence. In the system prompt, we first indicate that this is a simplification task and that it should simplify according to the operation labels. We then explain the reading levels and operation labels. Next, we introduce several rules, for example, that <*rephrase*> should differ from its input, and that <*copy*> should produce exactly the same output. Finally, we provide three few-shot examples of the training data.

We also run Qwen-32B without access to oracle labels. For this variant, we adjust the prompt to not refer to the operation labels and remove them from the examples, while keeping the reading levels. This prompt serves as our unguided baseline for the prompting-based approach.

The two prompts that we used can be found in Appendix A and on our GitHub: *prompts*.

6. Results

Figure 3 shows the micro-recall results of the fine-tuned BART models compared to the prompting-based approach with the Qwen model variants.

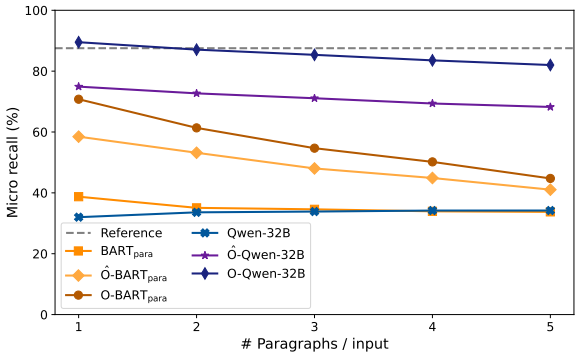

Figure 3: Plan-following micro recall for BART and Qwen-32B models across context sizes on Newsela-auto.

O-Qwen-32B achieves 89.5% micro-recall at paragraph level 1 and drops to 82.0% at 5 input paragraphs, a loss of 7.5 percentage points compared to 26.1 for O-BART over the same range. Ô-Qwen-32B similarly declines less steeply. Notably, Ô-Qwen-32B (with predicted operation labels) at 5 input paragraphs still exceeds O-BART (with oracle labels) at 1 paragraph, suggesting that the prompting-based approach is more robust to increased context even under imperfect planning. The unguided LLM performs similarly to the unguided BART model.

Input (Complex)

<copy> Those competing interests were already playing out on fire-stripped slopes. <split> Above, the woodpeckers were feasting on wood-boring beetles that began swarming dead trees while they were still smoking. <rephrase> Below, crews with chain saws and big-rig trucks were removing barriers and salvaging fallen tree trunks from roads and paths created for power lines. <copy> “We’re looking for silver linings,” Bridgman said with a sigh. <copy> “But we’re caught between extremes.”

O-BART$_{para\text{-}5}$ Simplification

Those competing interests were already playing out on fire-stripped slopes. Above, the woodpeckers were feasting on wood-boring beetles that began swarming dead trees while they were still smoking.

O-Qwen-32B-5 Simplification

Those competing interests were already playing out on fire-stripped slopes. Above, the woodpeckers were feasting on wood-boring beetles. They began swarming dead trees while they were still smoking. Below, crews with chain saws and big-rig trucks were removing barriers. They were also salvaging fallen tree trunks from roads and paths created for power lines. “We’re looking for silver linings,” Bridgman said with a sigh. “But we’re caught between extremes.”

Figure 4: Simplification comparison between O-BART$_{para\text{-}5}$ and O-Qwen-32B-5. Operation labels are marked by <>.

The ability of Qwen to follow the plan much more faithfully also becomes apparent when we look at the matrices in Figure 5, where O-Qwen-32B-1 follows the plan with high accuracy, and O-Qwen-32B-5 still maintains strong performance compared to the deletion bias of O-BART. This deletion bias is illustrated in Figure 4: O-BART deletes the final three sentences (rephrase, copy, copy) and fails to split, while O-Qwen-32B correctly executes all operations.

System	SMART ↑			FKGL ↓	SARI ↑	Length	
	P	R	F1			Tok.	Sent.
Input	58.9	64.9	61.5	7.61	19.5	1089.9	49.1
Reference	100	100	100	4.62	100	721.1	45.2
Qwen-32B	39.0	43.9	41.2	5.01	40.0	923.3	53.4
Ô-Qwen-32B-1	57.5	61.4	59.2	5.51	47.4	830.5	48.9
Ô-Qwen-32B-2	57.3	60.5	58.7	5.54	47.3	815.9	47.9
Ô-Qwen-32B-3	57.2	59.5	58.2	5.57	47.2	794.4	46.7
Ô-Qwen-32B-4	57.3	58.5	57.7	5.63	47.0	777.0	45.2
Ô-Qwen-32B-5	57.2	57.8	57.3	5.68	47.0	766.6	44.3
O-Qwen-32B-1	61.9	62.0	61.9	5.49	51.5	743.3	43.8
O-Qwen-32B-2	61.7	61.2	61.4	5.53	51.2	729.9	42.9
O-Qwen-32B-3	61.8	60.3	61.0	5.59	51.0	712.6	41.9
O-Qwen-32B-4	62.0	59.3	60.5	5.62	50.9	693.0	40.5
O-Qwen-32B-5	61.8	58.6	60.1	5.68	50.6	684.5	39.7

Table 5: Results of document simplification of LLM models on Newsela-auto.

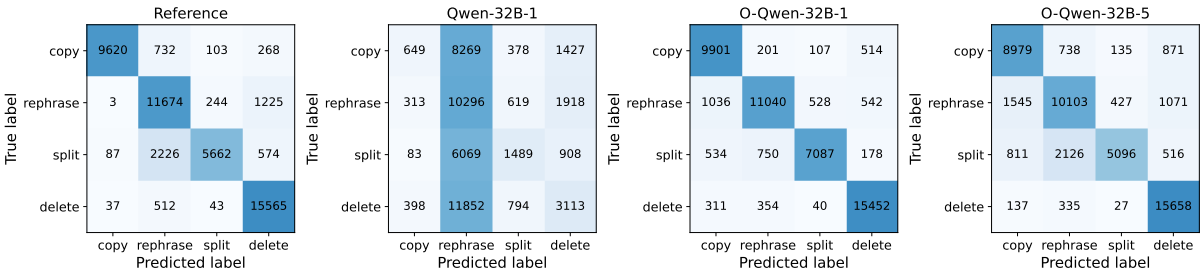

Figure 5: Confusion matrices comparing the ground-truth (oracle) operation labels against the operation labels predicted by the model for input documents, evaluated on the Newsela-auto dataset.

Finally, the document-level results in Table 5 show that both O-Qwen-32B and Ô-Qwen-32B remain stable in SMART, FKGL, and SARI scores across input levels, with only marginal drops. The Qwen-32B model with predicted operation labels performs only slightly below the model with oracle labels. Ô-Qwen-32B-5 produces more sentences than O-Qwen-32B, staying closer to the reference length. Finally, even though Qwen-32B achieves the lowest FKGL among the generated models (approaching the reference), it is semantically worse, as indicated by its lower SMART and SARI scores.

6.1. Effect of Model Size

Figure 6 shows the effect of model size on plan-following in the prompting-based approach.

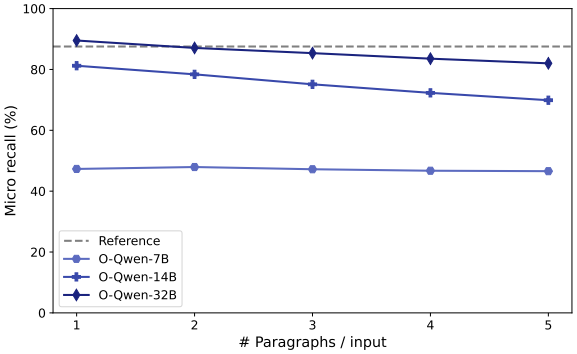

Figure 6: Plan-following micro recall for O-Qwen-7B, O-Qwen-14B, and O-Qwen-32B models across context sizes on Newsela-auto.

O-Qwen-7B is unable to execute the plan for any given input length, performing comparably to the unguided baselines. O-Qwen-14B achieves reasonable plan-following at paragraph level 1 (81.2%) but degrades to 69.9% at 5-paragraph inputs, a drop of 11.3 percentage points. O-Qwen-7B-5 achieves similar performance to the unguided baseline (SARI 40.5 vs. 40.0), indicating that the 7B model lacks the capacity to accurately simplify the input according to the operation labels.

System	SMART ↑			FKGL ↓	SARI ↑	Length	
	P	R	F1			Tok.	Sent.
Input	58.9	64.9	61.5	7.61	19.5	1089.9	49.1
Reference	100	100	100	4.62	100	721.1	45.2
Qwen-32B	39.0	43.9	41.2	5.01	40.0	923.3	53.4
O-Qwen-7B-5	49.6	54.3	51.5	6.15	40.5	834.1	47.0
O-Qwen-14B-1	59.5	59.7	59.5	5.80	49.9	749.5	43.4
O-Qwen-14B-2	59.8	58.3	59.0	6.01	49.4	726.0	40.6
O-Qwen-14B-3	59.4	56.6	57.9	6.13	48.8	702.3	38.8
O-Qwen-14B-4	59.1	54.8	56.8	6.30	48.2	678.7	36.8
O-Qwen-14B-5	58.9	53.7	56.1	6.42	47.5	662.3	35.7
O-Qwen-32B-5	61.8	58.6	60.1	5.68	50.6	684.5	39.7

Table 6: Results of document simplification of O-Qwen models on Newsela-auto.

O-Qwen-14B-1 shows comparable performance to O-BART$_{\text{para}}$ (Table 6), but degrades less as context grows. This is again reflected in the number of output sentences at 5 input paragraphs compared to O-BART$_{\text{para-5}}$ (35.7 vs. 17.0), indicating that the prompting-based approach avoids the deletion bias even at smaller model sizes. O-Qwen-32B maintains the strongest performance across the board.

6.2. Effect of Model Family

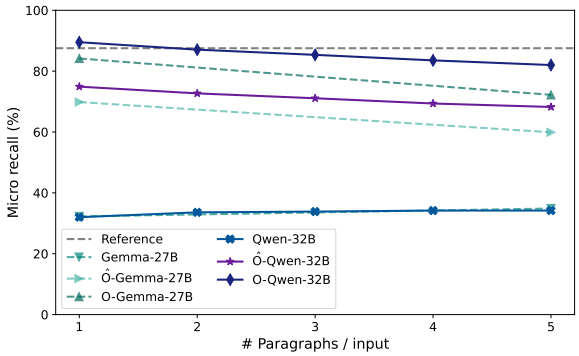

Figure 7: Plan-following micro recall for Qwen-32B and Gemma-27B model variants across context sizes on Newsela-auto.

Figure 7 and Table 7 show the results of running the same prompting-based pipeline with Gemma-27B. O-Gemma-27B-1 achieves 84.2% micro-recall, dropping to 72.2% at 5 input paragraphs, a loss of 12.0 percentage points. This is larger than O-Qwen-32B (-7.5) but well below O-BART (-26.1). Ô-Gemma-27B follows a similar pattern, declining from 69.8% to 59.9%. The document-level metrics in Table 7 show that O-Gemma-27B performs below O-Qwen-32B across SMART and SARI, and is less robust than the Qwen model family when context increases.

System	SMART ↑			FKGL ↓	SARI ↑	Length	
	P	R	F1			Tok.	Sent.
Input	58.9	64.9	61.5	7.61	19.5	1089.9	49.1
Reference	100	100	100	4.62	100	721.1	45.2
Gemma-27B	36.0	40.3	37.9	4.75	38.7	955.7	53.3
O-Gemma-27B-1	52.5	55.5	53.8	4.97	48.5	778.7	47.2
Ô-Gemma-27B-5	49.7	49.2	49.3	5.00	45.8	695.0	41.3
O-Gemma-27B-1	56.7	56.0	56.3	4.99	51.5	688.4	41.9
O-Gemma-27B-5	53.6	49.8	51.5	5.01	47.9	615.7	36.7

Table 7: Results of document simplification of Gemma-27B models on Newsela-auto.

7. Discussion

Fine-tuning vs. prompting. As shown in Section 6, the prompting-based approach achieves stronger plan-following performance than the fine-tuning approach at every paragraph input level, while also degrading less as context increases.

The degradation in the performance of the fine-tuned BART model is likely due to the difficulty of learning the sentence-level operation token semantics through fine-tuning. As more paragraphs are included, the number of operation tokens in the input grows, making the association between tokens and their corresponding sentences increasingly difficult to learn. At $i = 1$, BART sees ~ 2 operation tokens on average. At $i = 5$, it sees ~ 10. The relatively small capacity of BART ($\sim$140M parameters) likely amplifies this difficulty.

The instruction-tuned Qwen models do not have this degradation for multiple reasons. First, the meaning of each operation token is defined in the prompt rather than being learned through fine-tuning, and the models are specifically trained to follow structured instructions. Second, operation tokens are naturally interleaved with their corresponding sentences in the prompt, making the association between each token and its sentence straightforward regardless of input length. Besides this, the Qwen models are also significantly larger, the main model of this paper being at 32B parameters (vs. 140M).

Model size. BART and Qwen-32B have a substantial difference in model capacity. However, our ablation of different model sizes in Section 6.1 shows that the two approaches have very different capacity requirements. O-BART outperforms O-Qwen-7B, a model 50 times its size, demonstrating that fine-tuning can be effective at smaller scales. The prompting-based approach requires more capacity, with O-Qwen-14B being the minimum size at which the model can effectively follow operation labels and O-Qwen-32B further improving performance.

Model family. As shown in Section 6.2, the prompting-based approach transfers to Gemma-27B, though with lower plan-following and simplification quality than Qwen-32B. This gap may partly be explained by the difference in model size (27B vs. 32B), and by the fact that our pipeline was developed and tested using Qwen. It is therefore possible that Gemma-27B could perform better with, for instance, tweaked generation parameters or prompt adjustments.

Computational costs. The prompting-based approach requires more computational resources at inference time, primarily due to the larger model sizes (we used an H100 80GB for Qwen-32B). This cost is, however, somewhat mitigated by the non-existent training time and fast generation with vLLM. Additionally, O-Qwen-14B achieves comparable plan-following to O-BART while degrading far less at extended contexts, and requires only an A100 40GB.

7.1. Limitations

Dataset. The Newsela-auto dataset is proprietary. To gain access to the data, we first had to request access to the Newsela corpus from Newsela, and then contact the authors of Jiang et al. (2020) to obtain the Newsela-auto dataset. This significantly hinders reproducibility and further research.

Scope. Another limitation of our work is that we evaluate the prompting-based approach on only one dataset, which limits generalizability to other domains. Additionally, while we include Gemma-3-27B as an ablation, our main findings are based on a single LLM family (Qwen), and further validation across a broader range of models and datasets would strengthen our conclusions.

8. Conclusion

In this paper, we have shown that fine-tuned plan-guided BART models degrade in their plan-following ability as context grows. To address this, we have replaced the fine-tuned BART models with a prompting-based approach using instruction-tuned LLMs. We have shown that this approach

maintains both plan-following accuracy and simplification quality at extended context lengths, degrading far less as input context increases. Even with predicted operation labels from the planning model, the prompting-based approach at 5 input paragraphs exceeds the fine-tuned BART model with oracle labels at 1 paragraph.

Lay Summary

We look at how well different AI systems follow detailed editing plans when asked to simplify longer pieces of text. The system is given sentence-by-sentence instructions, such as "shorten this" or "keep this." Earlier models stop following those instructions once the input grows from one paragraph to several, even after extensive training. Instead, they tend to delete too much or make things up, which hurts the quality of the simplified text. The researchers test a different approach using large language models guided through prompting rather than fine-tuning. This approach handles longer texts much better: it sticks to the requested edits even when given several paragraphs at once, and produces higher-quality simplifications in every setting.

Acknowledgments

Jan Bakker and Jaap Kamps are supported by the Netherlands Organization for Scientific Research (NWO NWA # 1518.22.105). Jaap Kamps is also supported by the University of Amsterdam (AI4FinTech program) and ICAI (AI for Open Government Lab). Views expressed in this paper are not necessarily shared or endorsed by those funding the research.

Bibliographical References

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2019. Cross-sentence transformations in text simplification. In *Proceedings of the 2019 Workshop on Widening NLP*, pages 181–184.

Reinald Kim Amplayo, Peter J. Liu, Yao Zhao, and Shashi Narayan. 2023. SMART: sentences as basic units for text evaluation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jan Bakker and Jaap Kamps. 2024. Beyond sentence-level text simplification: Reproducibility study of context-aware document simplification. In *Proceedings of the Workshop on DeTermIt! Evaluating Text Difficulty in a Multilingual Context@ LREC-COLING 2024*, pages 27–38.

Liam Cripwell, Joël Legrand, and Claire Gardent. 2023a. Context-aware document simplification. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13190–13206.

Liam Cripwell, Joël Legrand, and Claire Gardent. 2023b. 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.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural crf model for sentence alignment in text simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7943–7960.

J Peter Kincaid, Robert P Fishburne Jr, 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.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Taiki Papandreou, Jan Bakker, and Jaap Kamps. 2025. Medical text simplification from jargon detection to jargon-aware prompting. In *Proceedings of the Fourth Workshop on Text Simplification, Accessibility and Readability (TSAR 2025)*, pages 36–46.

Gemma Team. 2025. Gemma 3.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

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.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

A. Prompts

A.1. Oracle Prompt

The full oracle prompt includes three few-shot examples, omitted here for brevity. The prompt with the examples can be found on our GitHub: *oracle prompt*.

```
You are a text simplification editor.  Each input sentence is numbered and
labeled with an operation.  Execute each operation and output the result with
matching sentence numbers.  Output in English only.

Reading levels indicate the target simplification level:  <RL_0> (most complex)
to <RL_4> (simplest).  Higher reading levels should produce simpler output for
REPHRASE and SPLIT operations.

Operations:
- COPY: Output the sentence with ZERO changes.  Do not fix, improve, or edit
anything.
- REPHRASE: You MUST modify the sentence.  Remove or simplify at least one
phrase, clause, or difficult word.  The result must differ from the input
while preserving the core meaning.
- SPLIT: Break into exactly 2 shorter sentences at a natural point.  Keep as
many original words as possible.
- DELETE: Delete this sentence entirely.  Do not include its number in your
output.

Output format:
- Write one numbered line per non-deleted sentence, matching input numbers.
- For SPLIT, write both result sentences on the same numbered line.
- For DELETE, skip that number.
- Output ONLY the numbered results.  No explanations.
```

A.2. Unguided Prompt

The full unguided prompt includes three few-shot examples, omitted here for brevity. The prompt with the examples can be found on our GitHub: *unguided prompt*.

```
You are a text simplification
editor.  Simplify the given text.
Output in English only.

Reading levels indicate the target
simplification level:  <RL_0> (most
complex) to <RL_4> (simplest).
Higher reading levels should
produce simpler output.
```