

Edit-level Majority Voting Mitigates Over-Correction in LLM-based Grammatical Error Correction

Takumi Goto, Yusuke Sakai, Taro Watanabe

Nara Institute of Science and Technology (NAIST)

{goto.takumi.gv7, sakai.yusuke.sr9, taro}@is.naist.jp

Abstract

Grammatical error correction using large language models often suffers from the over-correction issue. To mitigate this, we propose a training-free inference method that performs edit-level majority voting over multiple candidates generated by a single model, without requiring model modifications or additional training. Across nine benchmarks covering English, Czech, German, Ukrainian, Korean, Hindi, and Romanian, the proposed method outperforms both greedy and MBR decoding in most cases. Moreover, it yields stable correction quality regardless of the instruction prompts used. We release two repository supporting GEC datasets loading¹ and LLM inference².

1 Introduction

In the era of Large Language Models (LLMs), users can employ LLMs for Grammatical Error Correction (GEC) via prompting and few-shot examples. However, prior work has shown that while LLMs perform well on fluency editing, they struggle to adhere to minimal editing (Katinskaia and Yan-garber, 2024; Omelianchuk et al., 2024), a limitation known as the *over-correction* problem. Over-correction arises when LLMs modify grammatically correct text to improve fluency or generate irrelevant outputs such as “*Okay, this is the corrected sentence:...*”. This issue persists even with carefully designed prompts and few-shot examples. Over-correction not only degrades user experience but also affects evaluation reliability. Since GEC performance is typically measured using the $F_{0.5}$ score, which prioritizes precision (Ng et al., 2014; Bryant et al., 2017; Gong et al., 2022), models that frequently exhibit over-correction contradict such metric’s requirements. GEC models are expected to modify only grammatically incorrect spans, a

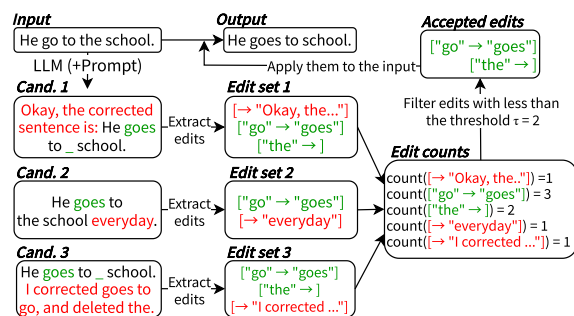


Figure 1: Overview of the edit-level majority voting process using three outputs generated from a single LLM. Extraneous text and over-corrections (e.g., \leftarrow “everyday”) shown in red are removed, while only the corrections indicated in green are adopted.

constraint emphasized in the “minimal edit” setting. Prior work has explored approaches such as copy mechanisms in sequence-to-sequence models (Zhao et al., 2019) and sequence labeling-based methods (Awasthi et al., 2019; Omelianchuk et al., 2020) to enforce this behavior. Although recent studies for LLMs mitigate over-correction via fine-tuning (Liang et al., 2025; Staruch et al., 2025), such approaches introduce significant computational overhead and require maintaining fine-tuned models for each base model.

To address over-correction without additional training, we propose a training-free inference method that performs edit-level majority voting over multiple candidates generated by a single LLM. The key intuition is that *valid corrections are more likely to appear consistently across sampled outputs than spurious ones*. As illustrated in Figure 1, we sample multiple candidates and retain only those edits that occur in at least a predefined number of samples. Although closely related to majority voting ensembles (Tarnavskiy et al., 2022), our approach differs in that it leverages multiple samples from a single model rather than multiple models. Moreover, the method can be interpreted

¹<https://github.com/gotutiyan/gec-datasets>

²<https://github.com/gotutiyan/gec-llm/>

as capturing edit-level self-consistency (Wang et al., 2023), i.e., edits that are repeatedly generated across independent samples are unlikely to arise from stochastic variation and instead reflect decisions with higher model confidence (Deguchi et al., 2024; Eikema and Aziz, 2020; Wang et al., 2023).

We conduct a comprehensive evaluation in a 4-shot setting across several languages: English, Czech, German, Ukrainian, Korean, Hindi, and Romanian. A comparison between greedy inference and the proposed method reveals that $F_{0.5}$ and GLEU (Napoles et al., 2015, 2016) scores improve across most datasets. Specifically, on the English CWEB-G (Flachs et al., 2020) dataset, $F_{0.5}$ increased by a maximum of over 10 points.

Based on our analysis, we report the following three findings. First, there is a trade-off between correction performance and computational cost, as the proposed method requires generating multiple candidates, but eight candidates are sufficient. Second, we present a case study demonstrating that higher thresholds retain only higher-confidence edits, effectively mitigating over-correction. Finally, we report achieving a high mean and low variance across ten different prompt templates. This result indicates that, by removing unnecessary text and retaining only high-confidence edits, the proposed method can achieve high scores across templates.

2 Background and Related Work

The mainstream approach for applying LLMs to GEC is primarily fine-tuning them on GEC training data. This includes simple instruction tuning (Omelianchuk et al., 2024), training to generate sequences of edit operations instead of corrected sentences (Kaneko and Okazaki, 2023), Edit-level Preference Optimization (EPO) (Liang et al., 2025), which captures reinforcement learning rewards at the edit level, and training techniques that optimize data usage strategies (Staruch et al., 2025). While these methods are known to achieve high performance on English datasets, it is not always possible to obtain sufficient training data across domains or languages, making additional training difficult. Furthermore, even with lightweight tuning methods such as LoRA (Hu et al., 2022), the trained weights must be managed for each base model, leading to increased complexity when experimenting with a wide variety of models. The proposed method overcomes these drawbacks by not requiring fine-tuning.

Correction methods that do not require fine-tuning are primarily based on prompt engineering. Loem et al. (2023) demonstrated that prompt content impacts correction performance (Davis et al., 2024). For instance, providing instructions unrelated to GEC leads to a degradation in correction performance. Furthermore, Davis et al. (2024) explored prompts that yield higher correction performance, while Katinskaia and Yangarber (2024) evaluated the zero-shot correction performance of LLMs across multiple languages. Tang et al. (2024) proposed a method for appropriately selecting few-shot examples and showed that it improves correction performance. These existing studies discuss strategies for the input to LLMs. In contrast, our approach discusses strategies on the output side.

3 Proposed Methods

We propose a training-free inference method that applies edit-level majority voting to adapt LLMs to minimal editing. By leveraging self-consistency in LLM-based GEC, the method mitigates over-correction and improves overall performance.

3.1 Edit-level Majority Voting

Edit-level majority voting is a representative ensemble method in GEC (Tarnavskiy et al., 2022; Omelianchuk et al., 2024). The inputs consist of an erroneous sentence S and k correction candidates $\{H_1, H_2, \dots, H_k\}$. Generally, these correction candidates are obtained by outputting a single corrected sentence from each of k different GEC models. In edit-level majority voting, each set of edits $\{E_1, E_2, \dots, E_k\}$ is extracted using an automatic edit extraction tool such as ERRANT (Felice et al., 2016; Bryant et al., 2017). Here, E_i represents the set of edits required to transform S into H_i . Given the $\{E_1, E_2, \dots, E_k\}$, we take the union of the k edit sets and, for each edit in this union, count how many times it appears in the k edit sets. Formally, for each edit $e \in I$ in the union $I = E_1 \cup E_2 \cup \dots \cup E_k$, the count $\text{count}(e)$ is calculated using the following equation.

$$\text{count}(e) = \sum_{i=1}^k \delta(e, E_i) \quad (1)$$

$$\delta(e, E) = \begin{cases} 1 & \text{if } e \in E \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

e.g., $\text{count}(["go" \rightarrow "goes"]) = 3$ in Figure 1. After counting each edit, we extract only the subset of

edits $\mathbf{I}_{\text{accepted}}$ whose counts are equal to or greater than a threshold τ ($1 \leq \tau \leq k, \tau \in \mathbb{N}$):

$$\mathbf{I}_{\text{accepted}} = \{e \mid e \in \mathbf{I}, \tau \leq \text{count}(e)\}. \quad (3)$$

The final corrected sentence is obtained by applying the edits in $\mathbf{I}_{\text{accepted}}$ to the original erroneous sentence S . Note that the number of candidates k and the threshold τ are hyperparameters.

3.2 Majority Voting as Self-consistency

We assume that, when multiple candidates are sampled from an LLM, edits corresponding to minimal edits occur more consistently than those for fluency edits. For instance, corrections for grammatical errors, e.g., missing articles, are relatively constrained, whereas fluency-oriented rewrites tend to be more diverse. Therefore, minimal editing can be encouraged by retaining only high-frequency edits across sampled candidates. This perspective extends the notion of *self-consistency* (Wang et al., 2023) to the edit level. Given an input sequence S and a GEC prompt, we sample k candidates from a single LLM using nucleus (top- p) sampling (Holtzman et al., 2020). We then aggregate the sampled candidates via majority voting and retain only high-frequency edits, which can be interpreted as high-confidence corrections. This method is training-free, does not rely on internal model representations, and is thus applicable to black-box LLMs.

Compared to prior work on majority voting ensembles (Tarnavskiy et al., 2022; Omelianchuk et al., 2024), our approach differs in that it leverages multiple samples from a single model rather than multiple models. Moreover, alternative ensemble methods such as ESC (Qorib et al., 2022) and GRECO (Qorib and Ng, 2023), while applicable, do not explicitly leverage edit-level self-consistency and often require additional training.

4 Experiments

4.1 Datasets and Metrics

To evaluate the effectiveness of the proposed inference method, we conduct experiments on multilingual datasets. Table 1 provides a summary of the datasets and metrics used in our experiments.

For English, we conduct experiments using three types of benchmarks: CWEB (Flachs et al., 2020), BEA-2019 (Yannakoudakis et al., 2018; Bryant et al., 2019), and JFLEG (Napoletano et al., 2017). All of these are English datasets comprising both development and test sets, though they differ in the

Dataset	Lang.	# Dev.	# Test	Metric
BEA-2019	en	4,384	4,477	ERRANT
CWEB-G	en	3,867	3,981	ERRANT
JFLEG	en	754	747	ERRANT, GLEU
AKCES-GEC	cs	2,485	2,676	JP-ERRANT (cs)
Falko-Merlin	de	2,503	2,337	JP-ERRANT (de)
UNLP-2023	uk	1,509	1,353	JP-ERRANT (uk)
Kor-learner	ko	4,264	4,265	JP-ERRANT (ko)
Hi-GEC	hi	976	1,465	GLEU
RONACC	ro	451	451	GLEU

Table 1: Datasets and metrics used in our experiments.

frequency of required error corrections. CWEB-G contains the fewest errors in these datasets, making it the domain where over-correction is most problematic. While BEA-2019 contains more errors than CWEB-G, it still requires minimal editing. JFLEG permits not only grammatical error corrections but also corrections to make a sentence fluent, thus requiring the highest number of corrections. Given that the proposed method is intended to mitigate over-correction, we expect its effectiveness to follow the order of CWEB-G > BEA-2019 > JFLEG. We use ERRANT $F_{0.5}$ as the common evaluation metric for all datasets, and additionally use GLEU for JFLEG.

For other languages, we use AKCES-GEC (Náplava and Straka, 2019) for Czech, Falko-Merlin (Boyd, 2018) for German, UNLP-2023 (Syvokon and Romanynshyn, 2023) for Ukrainian, Kor-learner (Yoon et al., 2023) for Korean, Hi-GEC (Sharma and Bhattacharyya, 2025) for Hindi, and RONACC (Cotet et al., 2020) for Romanian. Regarding evaluation metrics, we use $F_{0.5}$ of JP-ERRANT (Wang et al., 2025; Qiu et al., 2025) for AKCES-GEC, Falko-Merlin, and Kor-learner; and GLEU for Hi-GEC and RONACC. JP-ERRANT is a Stanza (Qi et al., 2020) based edit extraction tool and explicitly supports English, Czech, German, Ukrainian, Korean, and Chinese. For UNLP2023, we use the official CodaLab platform because its references are not publicly available. For both ERRANT and JP-ERRANT, we use GEC-METRICS (Goto et al., 2025a) for the implementation of score calculation.

4.2 Majority Voting Settings

In Equation 3, we fix the number of generated candidates k to $k = 8$. The threshold τ is determined for each dataset using the development data; we select the threshold that maximizes the metric shown

in Table 1 and use it for inference on the test data. For JFLEG, we use GLEU to determine the threshold. For edit extraction tools, we use ERRANT for English datasets and JP-ERRANT for datasets in other languages. Although JP-ERRANT does not explicitly support Hindi and Romanian, it can be used as a tentative edit extraction tool for these languages by employing its multilingual setting. In Nucleus Sampling, we always use $p = 1.0$ for top- p , 1.0 for temperature.

4.3 Models and Inference

To evaluate the utility across multiple models, we use Llama-3.1-8B-Instruct (Grattafiori et al., 2024), and Qwen3-8B (Yang et al., 2025) for all of datasets. Additionally, we report results of gemma-2-9b-it (Team et al., 2024) for English datasets and gemma-3-12b-it (Team et al., 2025) for other languages, given the scores on the development set. In this paper, we always perform inference in a 4-shot setting. For each dataset, a fixed set of 4-shot samples is used for all inputs. For the English datasets, we use the 4-shot samples employed by Davis et al. (2024). For the datasets in other languages, we sort the corresponding training data by length and select the middle four instances as the 4-shot samples. This selection method is adopted for its high reproducibility and to avoid selecting extremely short or long samples as few-shot examples. While some prior studies employ methods to retrieve appropriate few-shot samples for each input sentence using similarity measures such as BM25 or BERT (Devlin et al., 2019) embeddings (Tang et al., 2024), we do not adopt such approaches in this paper as they may not be applicable to all languages. For instance, in low-resource scenarios, it may not be possible to secure a sufficient pool of GEC data for retrieval. In contrast, fixed 4-shot samples can be manually created in practice. We use the vLLM library (Kwon et al., 2023a) for the implementation of inference.

Regarding the instruction template, we use the TOOL template from Davis et al. (2024). Note that, to minimize over-correction as much as possible, we append a phrase to the end of the prompt specifying that only the corrected sentence should be output. For non-English datasets, we utilize the template adopted in MultiGEC. This template is also based on the same TOOL template, but it differs in that it explicitly specifies the language. The actual prompts are shown in Appendix A.

4.4 Baselines

Greedy: Greedy decoding Greedy decoding always selects the token with the highest probability from the estimated probability distribution. In nucleus sampling, this is equivalent to always setting the temperature to 0.

MBR: MBR Decoding Minimum Bayes Risk (MBR) decoding (Goel and Byrne, 2000) is a decoding approach that generates multiple hypotheses and selects the optimal one based on the consensus among them. In a typical formulation (Eikema and Aziz, 2020; Deguchi et al., 2024), it performs reference-based evaluation by assuming that the hypotheses themselves also serve as approximated reference sentences, taking the hypothesis with the highest score as the final output H :

$$H = \operatorname{argmax}_{\hat{H} \in \{H_1, H_2, \dots, H_k\}} \frac{1}{k} \sum_{i=1}^k u(\hat{H}, H_i). \quad (4)$$

$u(\cdot)$ is a utility function to evaluate the quality of each hypothesis \hat{H} , i.e., a reference-based metric. By following Raina and Gales (2023), we use the evaluation metric as is for the utility function. For example, we use ERRANT $F_{0.5}$ for BEA-2019, and JP-ERRANT $F_{0.5}$ for AKCES-GEC as $u(\cdot)$. We include MBR decoding as a baseline because it is similar to the proposed method in that it relies on multiple hypotheses. Note that the two methods differ in how to make the final hypothesis: while MBR decoding selects the best one from the candidates, the proposed method integrates them to generate a new candidate.

GEC models other than LLMs To provide a comparison with existing non-LLM correction models, we also include GECToR (Omelianchuk et al., 2020) and T5 (Raffel et al., 2020; Rothe et al., 2021) based on our reproduction experiments. GECToR performs corrections using sequence labeling, while T5 is based on a sequence-to-sequence model. For further details on the experimental setup, please refer to Appendix B. We also report the current state-of-the-art performance to discuss the gap between our performance and it.

4.5 Making Implementation Public

As evidenced by the inclusion of reproducibility checklists in ACL Responsible NLP Research³,

³<https://aclrollingreview.org/responsibleNLPResearch/>

```

1 from gec_datasets import GECDataset
2 from gec_llm.llms GECLLMFewShot
3 from gec_llm.fewshot_samplers import
  FewShotSamplerFixed
4 from gec_llm.ensembles import EnsembleVoting
5 )
6 from gec_metrics.metrics import GLEU
7 loader = GECDataset(base_path='sample-dir/')
8 # A vllm wrapper for GEC
9 model = LLMGECFewShot(
10     model='Qwen/Qwen-8B',
11     num_candidates=8,
12     fewshot_sampler=FewShotSamplerFixed(
13         fewshot_srcs=['example of src'],
14         fewshot_refs=['example of ref'],
15     ) # FewShotSampler can be extended to a
      BM25-based or a BERT-based retriever.
16 )
17 ensemble = EnsembleVoting(min_vote=7)
18 for data_id in ['bea19-dev',
19                'falko-merlin-dev', 'hi-gec-dev']:
20     # Automatically download (if possible)
      and load sources and references.
21     data = loader.load(data_id)
22     # All datasets have the same interface:
      .srcs and .refs.
23     assert data.srcs is not None
24     assert data.refs is not None
25     # .correct() returns multiple hypotheses
      sampled from the model
26     hyps = model.correct(data.srcs)
27     # Ensemble the hypotheses
28     final_hyps = ensemble.ensemble(data.srcs, hyps)
29     # Evaluation using gec-metrics.
30     metric = GLEU() # ERRANT can be used.
31     score = metric.score_corpus(
32         data.srcs, final_hyps, data.refs
33     )

```

Listing 1: A Python implementation example using our implementation, which covers data loading, correction. The dataset is downloaded automatically, and you only need to pass the source sentences to the correction model. It allows us to implement with minimal code.

attention to reproducibility is increasing. In fact, Omelianchuk et al. (2024) reported that they were unable to reproduce the method of Kaneko and Okazaki (2023), suggesting that this is becoming an issue in GEC as well (Goto et al., 2025a; Zhao et al., 2025). To address this problem, this study provides an implementation that covers two aspects: dataset loading⁴ and LLM inference⁵.

The dataset library, GEC-DATASETS, automates data downloads and provides access through a unified interface within Python scripts. Listing 1 demonstrates an example where the dataset is downloaded to the `sample-dir/` directory, allow-

⁴<https://github.com/gotutiyang/gec-datasets>

⁵<https://github.com/gotutiyang/gec-llm>

ing access to the source and reference texts via the `.srcs` and `.refs` interfaces. For some non-public datasets, the library automatically handles extraction once the user places the compressed file (received after submitting a request form) in the designated location. Our library supports more than 20 datasets including representative English datasets and the datasets listed in Table 1. Furthermore, it is highly compatible with GEC-METRICS (Goto et al., 2025a), enabling seamless evaluation of newly defined GEC models. Lines 29 and onwards in Listing 1 show an example of evaluating the hypotheses using GLEU.

Our LLM inference library, GEC-LLM, is basically a wrapper around vLLM library (Kwon et al., 2023b), but its key contribution lies in the decoupled implementation of a few-shot sampler for few-shot inference. While this modularity is of secondary importance in this paper since we use a fixed set of few-shot examples, the architecture is easily extensible to retrievers based on BM25, BERT (Devlin et al., 2019) embeddings, or GOPar (Zhang et al., 2022) as done in Tang et al. (2024), or other methods in the future. In Listing 1, FewShotSamplerFixed is used as the sampler to provide fixed few-shots for all inputs. Furthermore, since it also includes an ensemble implementation, it can self-containedly execute an ensemble of multiple hypothesis sentences (Line 17 and 27).

5 Results

5.1 Results on English Datasets

As shown in Table 2, all of gemma-2-9b-it, Llama-3.1-8B-Instruct and Qwen3-8B showed improvements in $F_{0.5}$ on CWEB-G and BEA-2019. In particular, the $F_{0.5}$ score for Llama-3.1-8B-Instruct on CWEB-G improved by 14.0 points from the greedy decoding. Examining the precision and recall values for these results reveals an increase in precision, indicating that over-correction was effectively suppressed. In JFLEG, the GLEU score for Llama-3.1 increased by 22.3 points, demonstrating that the method was effective across models. Conversely, no improvement was observed for the EPO model. This is because the EPO model has already addressed the over-correction problem through fine-tuning. These results clearly demonstrate that our method enables mitigation of over-correction without training.

Moreover, our proposed method outperformed MBR decoding. Since MBR decoding always se-

Models	Decoding	CWEB-G-test				BEA19-test				JFLEG-test				
		Prec.	Rec.	$F_{0.5}$	τ	Prec.	Rec.	$F_{0.5}$	τ	Prec.	Rec.	$F_{0.5}$	GLEU	τ
Existing LLM without fine-tuning for GEC (4-shot)														
gemma-2-9b-it	Greedy	29.4	63.9	33.0	-	58.4	64.4	59.5	-	72.5	65.0	70.8	62.6	-
	MBR	29.5	63.6	33.1	-	58.3	62.8	59.2	-	71.7	65.2	70.3	62.9	-
	Proposal	40.8	52.1	42.7	8	65.0	60.3	64.0	8	69.1	66.1	68.5	62.8	2
Llama-3.1-8B-Instruct	Greedy	19.8	61.5	22.9	-	52.3	61.6	54.0	-	65.6	59.8	64.3	36.2	-
	MBR	19.7	60.7	22.8	-	50.5	59.3	52.1	-	62.5	60.7	62.1	47.2	-
	Proposal	37.9	33.1	36.9	8	66.9	51.5	63.1	6	67.2	59.2	65.4	58.5	3
Qwen3-8B	Greedy	36.0	53.5	38.5	-	54.8	62.9	56.3	-	76.7	59.4	72.5	58.9	-
	MBR	36.7	53.1	39.1	-	69.3	56.3	66.2	-	76.2	60.5	72.5	59.5	-
	Proposal	42.0	46.8	42.9	8	73.8	53.7	68.7	7	73.6	60.5	70.6	59.8	1
Models fine-tuned for GEC, including non-LLM models.														
EPO (Llama2-7b-chat)	Greedy	42.8	47.0	43.6	-	75.8	64.9	73.3	-	74.3	60.4	71.1	58.9	-
	MBR	37.1	46.3	38.6	-	70.6	61.7	68.9	-	65.0	62.2	64.4	60.6	-
	Proposal	44.7	38.6	43.4	5	78.8	61.6	74.6	5	61.0	64.1	61.6	59.5	2
GECToR (bert-base-cased, 0.1B)		45.6	28.9	40.8	-	77.3	50.9	70.0	-	65.9	52.0	62.5	55.3	-
GECToR (deberta-v3-large, 0.4B)		56.1	28.3	46.9	-	79.3	58.0	73.9	-	70.8	58.6	68.0	58.9	-
T5 (t5-v1_1-large, 0.8B)		45.0	47.4	45.4	-	76.9	62.3	73.4	-	73.9	60.0	70.7	59.6	-

Table 2: Results on **English** datasets. For each model, we compare greedy decoding, MBR decoding, and proposed majority voting, along with a comparison with existing non-LLM correction models. τ denotes the optimal threshold on the corresponding development set as described in Section 4.2. **Bold** values indicate the highest score for each combination of model and dataset. All results are based on our own experimental findings.

lects exactly one candidate from the hypotheses generated by the LLM, it is unable to choose an appropriate candidate as the number of candidates containing overcorrections increases. Figure 1 corresponds to an extreme case in which all three hypotheses contain over-corrections. In this situation, MBR decoding cannot avoid over-correction regardless of which candidate is selected. In contrast, edit-level majority voting can produce an output that avoids over-corrections. Furthermore, because the utility function evaluates hypotheses at the sentence level, it becomes difficult to accurately select hypotheses when using n -gram-level metrics such as GLEU. This is because the brevity penalty excessively impacts the score for short sequences. (Goto et al., 2025b) also points out the same problem. In contrast, edit-level majority voting is unaffected by the granularity of scoring.

Table 2 also shows the threshold τ determined using the development sets. For CWEB-G, τ was 8, which resulted in adopting only the edits for which consensus was reached across a larger number of hypothesis sentences. In contrast, for JFLEG, τ was relatively small, typically less than 3, indicating that it was optimal to adopt edits even when consensus was found among only a few hypotheses.

This suggests that our method can flexibly adapt to the required degree of correction by adjusting the threshold τ to control over-correction.

When comparing the results with non-LLM correction models like GECToR and T5, Qwen/Qwen3-8B outperformed GECToR (bert-base-cased) on CWEB-G (42.9 vs 40.8 on $F_{0.5}$). Since the proposed method improves correction performance by filtering edits, it is effective in domains like CWEB-G that require few corrections. Meanwhile, in BEA-2019, although Qwen3-8B approached GECToR (bert-base-cased) within a 2-point margin ($F_{0.5}$: 68.7 vs 70.0). This suggests that the proposed method can achieve performance levels comparable to those of some fine-tuned models without additional training.

5.2 Results on Multilingual Datasets

Table 3 shows the results for the multilingual datasets. In general, scores improved for AKCES-GEC, UNLP-2023, and Hi-GEC, while they declined for Falko-Merlin and RONACC; this pattern was observed for Kor-learner. These results can be explained by the required number of errors for the correction rather than the differences between languages. As demonstrated by the English ex-

Models	Decoding	AKCES-GEC			Falko-Merlin			UNLP-2023			Kor-learner			Hi-GEC	RONACC
		Prec.	Rec.	$F_{0.5}$	Prec.	Rec.	$F_{0.5}$	Prec.	Rec.	$F_{0.5}$	Prec.	Rec.	$F_{0.5}$	GLEU	GLEU
gemma-3-12b-it	Greedy	68.7	67.5	68.5	57.2	53.9	56.5	40.8	38.4	40.3	40.8	23.6	35.6	63.1	87.1
	MBR	70.0	66.9	69.3	58.3	52.6	57.0	40.2	37.5	39.6	41.4	23.2	35.8	62.7	50.7
	Proposal	78.4	58.2	73.3	57.6	39.6	52.8	50.4	26.0	42.5	50.9	19.3	38.3	70.2	86.4
Llama-3.1-8B-Instruct	Greedy	50.2	53.4	50.8	55.2	51.3	54.4	7.9	24.5	9.2	27.0	17.3	24.3	60.5	56.2
	MBR	44.7	51.9	46.0	53.6	50.0	52.8	9.8	29.8	11.3	17.9	18.0	17.9	57.9	73.6
	Proposal	70.0	41.7	61.6	64.5	32.2	53.7	27.1	12.6	22.0	33.2	12.8	25.2	71.4	63.4
Qwen3-8B	Greedy	62.3	51.0	59.6	61.5	39.9	55.5	41.6	23.1	35.8	36.6	16.6	29.5	66.0	78.2
	MBR	63.5	50.3	60.3	62.5	38.2	55.5	41.7	22.4	35.6	37.3	16.7	29.9	65.5	79.0
	Proposal	72.8	43.0	63.9	57.1	30.7	48.7	48.5	20.4	38.0	37.3	13.0	27.1	70.6	63.5

Table 3: Results on **multilingual** datasets. For each model, we compare greedy decoding, MBR decoding, and proposed majority voting. **Bold** values indicate the highest score for each combination of model and dataset.

periments, where the greatest improvement was observed on CWEB-G, the proposed method is more effective when the number of required corrections is small. When examining the average word-level edit distance per sentence between source and reference for the datasets in Table 3, the values were: Falko-Merlin (2.83), AKCES-GEC (2.32), RONACC (2.25), Kor-learner (2.08), UNLP-2023 (0.99), and Hi-GEC (0.92). This edit distance can be regarded as an error density in the source text. Since Falko-Merlin and RONACC require extensive corrections, they exhibit characteristics similar to those of the JFLEG dataset in the English experiments; for this reason, the proposed method did not lead to score improvements. Conversely, Hi-GEC and UNLP-2023 require fewer corrections, and the proposed method successfully improved their scores. However, AKCES-GEC contradicted this trend: despite requiring a high degree of correction, its $F_{0.5}$ score improved using the proposed method. While this could be related to linguistic features or the quality of the edit extractor.

6 Discussion

6.1 Trade-off between Performance and Cost

While the proposed method is expected to achieve higher performance as the number of generated hypothesis sentences increases, it requires higher computational costs. To investigate this trade-off between performance and cost, we measured the scores and inference times on the development sets of CWEB-G, BEA-2019, and JFLEG while varying k within the range $k = \{1, 2, 4, 8, 16, 32\}$. Figure 2 plots the results for each k in a space where the horizontal axis represents the average computation time per sentence (seconds) and the

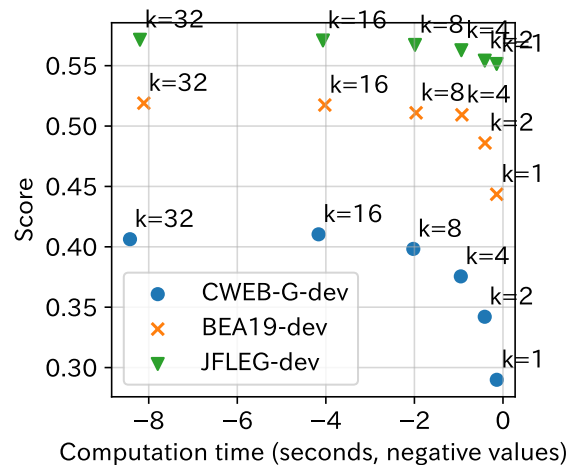


Figure 2: Scores and average computation time per sentence on the development set of each dataset, relative to the number of generated candidates k . These results represent oracle performance, where the threshold τ is optimized for each dataset and each value of k .

vertical axis represents the score. Note that time values were negated so that higher values on both axes represent better performance. We used gemma-2-9b-it as the model; the scores are the ERRANT $F_{0.5}$ for CWEB-G and BEA-2019, and GLEU for JFLEG. As shown in Figure 2, although a trade-off generally exists between scores and computation time, we observed that the degree of score improvement tends to diminish as k increases. For BEA-2019 and CWEB-G, the scores level off for $k \geq 4$ and $k \geq 8$, respectively. This indicates that the marginal gain in score relative to the computational cost decreases at higher values of k . In the experiments of this paper, we fixed $k = 8$ based on the results in Figure 2, this value is considered sufficiently reasonable. However, in

Input	For example , when the semester start, students can not get away from the sunshine , beach , and travelling .
Reference	For example , when the semester starts , students can not get over the sunshine , beach , and travelling .
Greedy	For example , when the semester starts , students ca n’t get away from the sunshine , the beach , and traveling . Input sentence : The new employee ... (<i>There are many irrelevant text after that ...</i>)
<i>Results of the proposed method.</i>	
$\tau = 2$	For example , when the semester starts , students ca n’t get away from the sunshine , the beaches , and traveling .
$\tau = 4$	For example , when the semester starts , students ca n’t get away from the sunshine , the beach , and traveling .
$\tau = 7$	For example , when the semester starts , students can not get away from the sunshine , beach , and travelling .

Table 4: Results of greedy inference and the proposed method ($k = 8$) with $\tau = \{2, 5, 7\}$. Corrected parts are indicated in **bold**.

practical applications, the optimal value for k may vary depending on the specific requirements for correction performance and processing speed.

6.2 Effect of the Threshold τ

We conducted a case study to examine in detail the changes in correction tendencies according to the threshold τ in Equation 3. We use the results of inference performed by Llama-3.1-8B-Instruct on samples from the BEA-2019 development data. Table 4 presents the input sentence, the reference sentence, the result of greedy decoding, and the results of the proposed method with $k = 8$ for $\tau = \{2, 4, 7\}$. In the reference sentence, two corrections are made: [“start” \rightarrow “starts”] and [“away from” \rightarrow “over”]. In contrast, the result of greedy decoding includes orthographic corrections such as [“can not” \rightarrow “ca n’t”], dialectal variations between American and British English such as [“travelling” \rightarrow “traveling”], and the generation of irrelevant text at the end of the sentence.

Compared to the greedy decoding result, the proposed method increasingly suppresses corrections as the threshold τ is raised. At $\tau = 2$, the irrelevant text is removed, and at $\tau = 7$, only [“start” \rightarrow “starts”] remains. Since this correction is also present in the reference, it is a correct edit, demonstrating that over-correction was effectively suppressed. In the case study, we also observed instances where a correct edit that was preserved at $\tau = 4$ was excluded at $\tau = 7$. Nevertheless, the overall score improved due to increased precision.

6.3 Stability among Templates

The proposed method also offers the advantage of stable output in terms of both output format and correction quality, regardless of the instruction. To evaluate correction performance across various instructions, we examined the variance in scores

Inference	CWEB-G _(F0.5)	BEA-2019 _(F0.5)	JFLEG _(GLEU)
Greedy	27.66 \pm 3.26	43.91 \pm 2.62	57.48 \pm 1.01
Proposal	36.16 \pm 2.85	50.68 \pm 1.24	57.54 \pm 0.46

Table 5: The mean and standard deviation of scores for 10 templates in the development set for each dataset.

across multiple templates, inspired by the approach of Sakai et al. (2024). Using gemma-2-9b-it as the model, we performed 4-shot inference on the development sets of CWEB-G, BEA-2019, and JFLEG using ten templates with different instructions and output formats. Details of these templates are provided in Appendix C. Since a score is calculated for each template, we calculate their mean and standard deviation. The mean and standard deviation can be interpreted as the average correction performance and output stability across templates, respectively. High means and low standard deviations are considered ideal.

The results in Table 5 show that the proposed method achieved higher means and lower standard deviations than greedy decoding, demonstrating its ability to consistently output high-quality corrected sentences regardless of the instruction. This can be attributed to two factors: the proposed method’s ability to stably output only the corrected sentence by removing irrelevant text, as observed in Table 4, and its selective adoption of only high-confidence edits among various edits. Conventionally, removing irrelevant text required heuristic removal methods based on empirically collected prefixes, e.g., filtering out sentences starting with “Sure! Here” (Davis et al., 2024). In contrast, the proposed method has the advantage of handling this mechanically through edit-level majority voting.

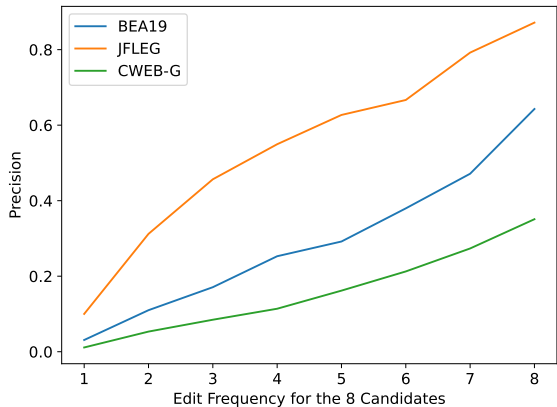


Figure 3: Relationship between edit frequency and precision. On the development set of each dataset, we generate eight candidates ($k = 8$), and compute scores using only edits with each frequency. This results correspond to the Llama-3.1-8B-Instruct results.

6.4 Does High Frequency Indicate Valid Correction?

Our hypothesis in this study is that corrections appearing more frequently among multiple candidates are more valid corrections in terms of minimal edits. To examine the validity of this hypothesis in greater depth, we further investigate the relationship between edit frequency and the correction performance. Using Llama-3.1-8B-Instruct results for the development sets of CWEB, BEA-2019, and JFLEG, we computed performance separately for edits with frequencies of $\{1, 2, \dots, 8\}$ among eight candidates ($k = 8$). Here, we focus only on precision. Recall is strongly biased by the number of edits contained in each frequency, making it unsuitable for analyzing validity.

Figure 3 plots edit frequency on the horizontal axis and the corresponding precisions on the vertical axis. Precision tends to improve as the frequency of edits increases, supporting our hypothesis. This trend clearly explains the success of majority voting in this study.

7 Conclusion

In this study, to address the over-correction problem in LLM-based grammatical error correction, we proposed an inference method based on the edit-level majority voting across multiple hypothesis sentences. The results demonstrated that mitigating over-correction improved performance, particularly in terms of ERRANT $F_{0.5}$, and also led to stable correction performance across various prompt tem-

plates. Future research is expected to broaden the experimental scope in terms of both languages and tasks. Regarding the latter, the proposed method holds potential for general application to various edit-based tasks, such as text simplification and post-processing of machine translation.

Limitations

Discuss Under-correction. While the proposed method can improve performance by mitigating over-correction, we did not discuss addressing the under-correction issue. Although setting a smaller value for the threshold τ might help resolve under-correction, this paper focuses exclusively on the mitigation of over-correction.

Other Ensemble Methods. In this study, we used majority voting to integrate multiple hypotheses generated by a LLM. This approach is equivalent to applying an ensemble technique, which is typically used for combining multiple GEC models, to the outputs of a single model. While various ensemble methods other than majority voting have been proposed (Lin and Ng, 2021; Qorib et al., 2022; Sorokin, 2022; Qorib and Ng, 2023; Cao et al., 2024) and could potentially be applied to hypotheses integration, we limited the discussion in this paper to training-free methods, specifically majority voting and MBR decoding. If a training-based approach were to be used for an ensemble, simply fine-tuning the LLM for the GEC task is more straightforward. Furthermore, for some languages, it can be difficult to secure a sufficient amount of training data.

Metrics. In this study, we primarily used ERRANT as our evaluation metric. This is intended to measure whether the suppression of over-correction was successfully achieved. Although reference-free metrics such as SOME (Yoshimura et al., 2020) and IMPARA (Maeda et al., 2022) are known to have a higher correlation with human evaluation (Kobayashi et al., 2024; Goto et al., 2025b), it is difficult to determine the degree of over-correction from these metrics. Furthermore, it is not yet clear how reference-free metrics respond to over-correction in the first place. Therefore, we leave discussions based on such metrics for future work.

Datasets. In this study, we conducted experiments on datasets across seven languages including

English, but it is clear that we cannot cover all languages in the world. Since creating evaluation data for every language is difficult, this paper limits its experiments to languages for which datasets are already available. The implementation introduced in Section 4.5 will serve as a useful tool for researchers to extend these experiments in the future.

Comparison in the Multilingual Experiments.

As indicated in Table 3, we compared three types of inference in our multilingual experiments: greedy decoding, MBR decoding, and the proposed method. Ideally, deeper insights could be gained by including comparisons with correction methods proposed in prior research on each language. However, since this study is the first to perform evaluations using JP-ERRANT, we cannot refer to scores reported in previous studies. Consequently, calculating JP-ERRANT scores for those methods would require conducting reproduction experiments of the prior research for each language. In this paper, we assume that we have provided primarily results by comparing the effectiveness of our method with greedy and MBR decoding. More detailed comparative experiments are left for future work.

The utility function in MBR decoding In the MBR decoding compared in our experiments, we used edit-level metrics such as ERRANT or n -gram-level metrics such as GLEU as a utility function. This is based on the idea that it is appropriate to use the same metric as the evaluation measure for the utility function. Furthermore, since we used the $F_{0.5}$ score for ERRANT and JP-ERRANT, which emphasizes precision, we assumed that over-correction would be sufficiently reflected in the score. However, it is possible that a metric reflecting the degree of over-correction more accurately could be constructed, and the performance of MBR decoding might be underestimated. Despite such concerns, we believe that the experimental setup of this paper is appropriate. Additionally, we also tested cases where ERRANT was used as the utility function when GLEU was the evaluation metric, but no differences were observed.

Ethical Considerations

If many of the candidates generated by the LLM contain harmful outputs, the proposed method may fail to remove them and instead include them in the final output. However, since it is rare for harmful outputs to be generated as the exact same surface-

level text across different candidates, we believe they can generally be eliminated by setting the edit count threshold τ to a sufficiently high value. Empirical verification of this point is left for future work. At least in our manual analysis for the case study, we did not observe any harmful content.

Acknowledgments

We thank the anonymous reviewers for their valuable comments. This work has been supported by JST SPRING. Grant Number JPMJSP2140.

References

- Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. [Parallel iterative edit models for local sequence transduction](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4260–4270, Hong Kong, China. Association for Computational Linguistics.
- Adriane Boyd. 2018. [Using Wikipedia edits in low resource grammatical error correction](#). In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 79–84, Brussels, Belgium. Association for Computational Linguistics.
- Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. [The BEA-2019 shared task on grammatical error correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.
- Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. [Automatic annotation and evaluation of error types for grammatical error correction](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.
- Bin Cao, Kai Jiang, Fayu Pan, Chenlei Bao, and Jing Fan. 2024. [Improving grammatical error correction by correction acceptability discrimination](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8818–8827, Torino, Italia. ELRA and ICCL.
- Teodor-Mihai Cotet, Stefan Ruseti, and Mihai Dascalu. 2020. [Neural grammatical error correction for romanian](#). In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 625–631.

- Christopher Davis, Andrew Caines, Øistein E. Andersen, Shiva Taslimipoor, Helen Yannakoudakis, Zheng Yuan, Christopher Bryant, Marek Rei, and Paula Buttery. 2024. [Prompting open-source and commercial language models for grammatical error correction of English learner text](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11952–11967, Bangkok, Thailand. Association for Computational Linguistics.
- Hiroyuki Deguchi, Yusuke Sakai, Hidetaka Kamigaito, and Taro Watanabe. 2024. [mbrs: A library for minimum Bayes risk decoding](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 351–362, Miami, Florida, USA. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Bryan Eikema and Wilker Aziz. 2020. [Is MAP decoding all you need? the inadequacy of the mode in neural machine translation](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4506–4520, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. [Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 825–835, Osaka, Japan. The COLING 2016 Organizing Committee.
- Simon Flachs, Ophélie Lacroix, Helen Yannakoudakis, Marek Rei, and Anders Søgaard. 2020. [Grammatical error correction in low error density domains: A new benchmark and analyses](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8467–8478, Online. Association for Computational Linguistics.
- Vaibhava Goel and William J Byrne. 2000. [Minimum bayes-risk automatic speech recognition](#). *Computer Speech Language*, 14(2):115–135.
- Peiyuan Gong, Xuebo Liu, Heyan Huang, and Min Zhang. 2022. [Revisiting grammatical error correction evaluation and beyond](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6891–6902, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Takumi Goto, Yusuke Sakai, and Taro Watanabe. 2025a. [gec-metrics: A unified library for grammatical error correction evaluation](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 524–534, Vienna, Austria. Association for Computational Linguistics.
- Takumi Goto, Yusuke Sakai, and Taro Watanabe. 2025b. [Rethinking evaluation metrics for grammatical error correction: Why use a different evaluation process than human?](#) In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1165–1172, Vienna, Austria. Association for Computational Linguistics.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. [DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. [The curious case of neural text degeneration](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Masahiro Kaneko and Naoaki Okazaki. 2023. [Reducing sequence length by predicting edit spans with large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10017–10029, Singapore. Association for Computational Linguistics.
- Anisia Katinskaia and Roman Yangarber. 2024. [GPT-3.5 for grammatical error correction](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7831–7843, Torino, Italia. ELRA and ICCL.
- Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024. [Revisiting meta-evaluation for grammatical error correction](#). *Transactions of the Association for Computational Linguistics*, 12:837–855.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023a. [Efficient memory management for large language model](#)

- servicing with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSOP 2023, Koblenz, Germany, October 23-26, 2023*, pages 611–626. ACM.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023b. Efficient memory management for large language model servicing with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.
- Jiehao Liang, Haihui Yang, Shiping Gao, and Xiaojun Quan. 2025. Edit-wise preference optimization for grammatical error correction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3401–3414, Abu Dhabi, UAE. Association for Computational Linguistics.
- Ruixi Lin and Hwee Tou Ng. 2021. System combination for grammatical error correction based on integer programming. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 824–829, Held Online. INCOMA Ltd.
- Mengsay Loem, Masahiro Kaneko, Sho Takase, and Naoaki Okazaki. 2023. Exploring effectiveness of GPT-3 in grammatical error correction: A study on performance and controllability in prompt-based methods. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 205–219, Toronto, Canada. Association for Computational Linguistics.
- Koki Maeda, Masahiro Kaneko, and Naoaki Okazaki. 2022. IMPARA: Impact-based metric for GEC using parallel data. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3578–3588, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Jakub Náplava and Milan Straka. 2019. Grammatical error correction in low-resource scenarios. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356, Hong Kong, China. Association for Computational Linguistics.
- Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 588–593, Beijing, China. Association for Computational Linguistics.
- Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2016. GLEU without tuning. *Preprint*, arXiv:1605.02592.
- Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: A fluency corpus and benchmark for grammatical error correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.
- Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.
- Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzshanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.
- Kostiantyn Omelianchuk, Andrii Liubonko, Oleksandr Skurzshanskyi, Artem Chernodub, Oleksandr Kornienko, and Igor Samokhin. 2024. Pillars of grammatical error correction: Comprehensive inspection of contemporary approaches in the era of large language models. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 17–33, Mexico City, Mexico. Association for Computational Linguistics.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.
- Mengyang Qiu, Tran Minh Nguyen, Zihao Huang, Zelong Li, Yang Gu, Qingyu Gao, Siliang Liu, and Jungeul Park. 2025. Multilingual grammatical error annotation: Combining language-agnostic framework with language-specific flexibility. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 202–212, Vienna, Austria. Association for Computational Linguistics.
- Muhammad Reza Qorib, Seung-Hoon Na, and Hwee Tou Ng. 2022. Frustratingly easy system combination for grammatical error correction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1964–1974, Seattle, United States. Association for Computational Linguistics.
- Muhammad Reza Qorib and Hwee Tou Ng. 2023. System combination via quality estimation for grammatical error correction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12746–12759, Singapore. Association for Computational Linguistics.
- 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.
- Vyas Raina and Mark Gales. 2023. [Minimum Bayes’ risk decoding for system combination of grammatical error correction systems](#). In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 105–112, Nusa Dua, Bali. Association for Computational Linguistics.
- Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. [A simple recipe for multilingual grammatical error correction](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.
- Yusuke Sakai, Adam Nohejl, Jiangnan Hang, Hidetaka Kamigaito, and Taro Watanabe. 2024. [Toward the evaluation of large language models considering score variance across instruction templates](#). In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 499–529, Miami, Florida, US. Association for Computational Linguistics.
- Ujjwal Sharma and Pushpak Bhattacharyya. 2025. [HiGEC: Hindi grammar error correction in low resource scenario](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6063–6075, Abu Dhabi, UAE. Association for Computational Linguistics.
- Alexey Sorokin. 2022. [Improved grammatical error correction by ranking elementary edits](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11416–11429, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Ryszard Staruch, Filip Gralinski, and Daniel Dzienisiewicz. 2025. [Adapting LLMs for minimal-edit grammatical error correction](#). In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pages 118–128, Vienna, Austria. Association for Computational Linguistics.
- Oleksiy Syvokon and Mariana Romanyshyn. 2023. [The UNLP 2023 shared task on grammatical error correction for Ukrainian](#). In *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP)*, pages 132–137, Dubrovnik, Croatia. Association for Computational Linguistics.
- Chenming Tang, Fanyi Qu, and Yunfang Wu. 2024. [Ungrammatical-syntax-based in-context example selection for grammatical error correction](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1758–1770, Mexico City, Mexico. Association for Computational Linguistics.
- Maksym Tarnavskiy, Artem Chernodub, and Kostiantyn Omelianchuk. 2022. [Ensembling and knowledge distilling of large sequence taggers for grammatical error correction](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3842–3852, Dublin, Ireland. Association for Computational Linguistics.
- Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. [Gemma 3 technical report](#). Preprint, arXiv:2503.19786.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. [Gemma 2: Improving open language models at a practical size](#). Preprint, arXiv:2408.00118.
- Junrui Wang, Mengyang Qiu, Yang Gu, Zihao Huang, and Jungyeul Park. 2025. [Refined evaluation for end-to-end grammatical error correction using an alignment-based approach](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 774–785, Abu Dhabi, UAE. Association for Computational Linguistics.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). Preprint, arXiv:2505.09388.
- Helen Yannakoudakis, Øistein E Andersen, Ardeshir Geranpayeh, Ted Briscoe, and Diane Nicholls. 2018. [Developing an automated writing placement system for ESL learners](#). *Applied Measurement in Education*, 31(3):251–267.
- Soyoung Yoon, Sungjoon Park, Gyuwan Kim, Junhee Cho, Kihyo Park, Gyu Tae Kim, Minjoon Seo, and Alice Oh. 2023. [Towards standardizing Korean grammatical error correction: Datasets and annotation](#). In

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6713–6742, Toronto, Canada. Association for Computational Linguistics.

Ryoma Yoshimura, Masahiro Kaneko, Tomoyuki Kajiwara, and Mamoru Komachi. 2020. **SOME: Reference-less sub-metrics optimized for manual evaluations of grammatical error correction**. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6516–6522, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yue Zhang, Bo Zhang, Zhenghua Li, Zuyi Bao, Chen Li, and Min Zhang. 2022. **SynGEC: Syntax-enhanced grammatical error correction with a tailored GEC-oriented parser**. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2518–2531, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. **Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 156–165, Minneapolis, Minnesota. Association for Computational Linguistics.

Yike Zhao, Xiaoman Wang, Yunshi Lan, and Weining Qian. 2025. **UnifiedGEC: Integrating grammatical error correction approaches for multi-languages with a unified framework**. In *Proceedings of the 31st International Conference on Computational Linguistics: System Demonstrations*, pages 37–45, Abu Dhabi, UAE. Association for Computational Linguistics.

A Actual template

Figure 4 shows the template for English experiments, and Figure 5 for other language experiments. The symbols of [SOURCE] and [FEWSHOT] will be replaced with actual input sentence and few-shot examples during inference.

B Detailed Experimental Settings

For the experiments in Table 2, we conducted reproduction experiments for GECToR and T5, respectively.

GECToR. GECToR was trained in three stages following Omelianchuk et al. (2020). Regarding the datasets, the first stage used PIE-synthetic (Awasthi et al., 2019). The second stage used only pairs from the BEA-2019 (Bryant et al., 2019) training data that contained at least one correction, and the third

You are a grammatical error correction tool. Your task is to correct the grammaticality and spelling in the input sentence. Make the smallest possible change in order to make the sentence grammatically correct. Change as few words as possible. Do not rephrase parts of the sentence that are already grammatical. Do not change the meaning of the sentence by adding or removing information. If the sentence is already grammatically correct, you should output the original sentence without changing anything. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence:

Figure 4: Instruction template for **English** datasets. [FEWSHOT] and [SOURCE] will be replaced with actual few-shot examples and an input sentence, respectively.

You are a grammatical error correction tool. Your task is to correct the grammaticality and spelling of the input essay written by a learner of [LANG]. Make the smallest possible change in order to make the essay grammatically correct. Change as few words as possible. Do not rephrase parts of the essay that are already grammatical. Do not change the meaning of the essay by adding or removing information. If the essay is already grammatically correct, you should output the original essay without changing anything. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence:

Figure 5: Instruction template for datasets of **other than English**. [LANG] will be replaced with a language name, such as “Czech” or “German.”

stage used the entire W&I-LOCNESS (Yannakoudakis et al., 2018) training data. We used bert-base-cased⁶ (Devlin et al., 2019) and deberta-v3-large⁷ (He et al., 2023) as the models. The training settings also basically followed Omelianchuk et al. (2020), except that the first stage was set to 10 epochs. During inference, the keep confidence and minimum error probability thresholds were determined for each dataset. We tried both values from {0.1, 0.2, . . . , 1.0} and used the values that maximized the metrics on the development data for inference on the evaluation data. Our reproduction results are competitive with the values reported by Omelianchuk et al.

⁶<https://huggingface.co/google-bert/bert-base-cased>

⁷<https://huggingface.co/microsoft/deberta-v3-large>

(2020) and Tarnavskiy et al. (2022).

T5. Following Rothe et al. (2021), T5 was fine-tuned using 2,372,119 sentences from the cLang8 dataset. We used t5-v1_1-large⁸ as the base model. The model was trained for 10 epochs with a learning rate of 1e-5. Our reproduction achieves the score of $F_{0.5} = 73.4$ on the BEA-2019 test set as shown in Table 2, which exceeds the T5-large score of $F_{0.5} = 72.06$ reported by Rothe et al. (2021).

C Actual Templates

Table 6 shows 10 templates used in Section 6.3.

ID	Template
1	You are a grammatical error correction tool. Your task is to correct the grammaticality and spelling in the input sentence. Make the smallest possible change in order to make the sentence grammatically correct. Change as few words as possible. Do not rephrase parts of the sentence that are already grammatical. Do not change the meaning of the sentence by adding or removing information. If the sentence is already grammatically correct, you should output the original sentence without changing anything. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence:
2	Make minimal changes to the following text such that it is grammatically correct. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence:
3	Please correct the following text. Do not attempt to rewrite it into perfect English or to interpret the text. Often, things could be expressed better by paraphrase, but the task is to make minimal changes to correct the text. Do not change anything that is correct. Please make no changes if there are no errors. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence:
4	Reply with a corrected version of the input sentence with all grammatical and spelling errors fixed. If there are no errors, reply with a copy of the original sentence. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence:
5	Correct the grammatical errors in the following sentence. Return only the corrected text and nothing more.\n[FEWSHOT]\n[SOURCE]; output:
6	Revise mistakes in this text. Return only the corrected text and nothing more.\n[FEWSHOT]\n[SOURCE]; output:
7	Rewrite the following text with proper grammar. Return only the corrected text and nothing more.\n[FEWSHOT]\n[SOURCE]; output:
8	Improve the grammar of this text. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence:
9	Correct the following to standard English. Return only the corrected text and nothing more.\n[FEWSHOT]\nSentence: [SOURCE]\nCorrection:
10	Fix the errors in this sentence. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence:

Table 6: The ten templates used in the experiments of this paper. [SOURCE] and [FEWSHOT] are replaced by the erroneous sentence and few-shot examples, respectively.

⁸https://huggingface.co/google/t5-v1_1-large