Leveraging What's Overfixed: Post-Correction via LLM Grammatical Error Overcorrection

Taehee Park[*1] **Heejin Do**[*2] **Gary Geunbae Lee**[1,3]
[1] Graduate School of Artificial Intelligence, POSTECH, South Korea
[2] ETH Zurich, ETH AI Center
[3] Department of Computer Science and Engineering, POSTECH, South Korea
{taehpark, gblee}@postech.ac.kr heejin.do@ai.ethz.ch

Abstract

Robust supervised fine-tuned small Language Models (sLMs) often show high reliability but tend to undercorrect. They achieve high precision at the cost of low recall. Conversely, Large Language Models (LLMs) often show the opposite tendency, making excessive overcorrection, leading to low precision. To effectively harness the strengths of LLMs to address the recall challenges in sLMs, we propose Post-Correction via Overcorrection (PoCO), a novel approach that strategically balances recall and precision. PoCO first intentionally triggers overcorrection via LLM to maximize recall by allowing comprehensive revisions, then applies a targeted post-correction step via fine-tuning smaller models to identify and refine erroneous outputs. We aim to harmonize both aspects by leveraging the generative power of LLMs while preserving the reliability of smaller supervised models. Our extensive experiments demonstrate that PoCO effectively balances GEC performance by increasing recall with competitive precision, ultimately improving the overall quality of grammatical error correction.

1 Introduction

Grammatical error correction (GEC) has become a pivotal task in natural language processing, particularly for language learning applications and writing assistance tools (Tetreault and Leacock, 2014; Calò et al., 2021; Kaneko et al., 2022; Katinskaia and Yangarber, 2023; Liang et al., 2023; Alsaweed and Aljebreen, 2024). Traditional yet robust fine-tuned small Language Models (sLMs), which have been primarily used for GEC, tend to undercorrect, achieving high precision at the cost of lower recall (Figure 1). Improving recall is essential in educational settings, as it supports the primary goal of corrective feedback by raising students' awareness of their mistakes (Chen and Liu, 2021).

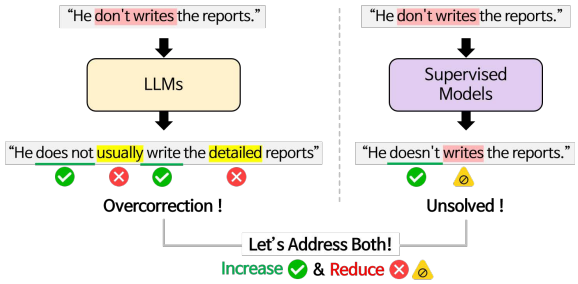

Figure 1: Our main motivation: LLMs exhibit high recall but low precision due to overcorrection, while smaller supervised models show high precision but low recall, possibly missing errors. PoCO leverages the strengths of LLMs to address the weaknesses of sLMs.

However, existing methods to reduce overcorrection often result in significantly decreased recall, limiting the usability of correction systems (Wang et al., 2024). This challenge is particularly evident with sLMs, which often exhibit conservative correction tendencies, e.g., making minimal changes or missing errors that require more substantial correction (Awasthi et al., 2019; Stahlberg and Kumar, 2020; Katsumata and Komachi, 2020; Omelianchuk et al., 2020; Rothe et al., 2021; Tarnavskyi et al., 2022). In contrast, Large Language Models (LLMs) exhibit the opposite tendency, e.g., making excessive modifications to input text (Loem et al., 2023; Fang et al., 2023; Wu et al., 2023; Coyne et al., 2023). Although LLMs have garnered notable attention for their potential in GEC, owing to their ability to understand and generate human-like text (Katinskaia and Yangarber, 2024; Zeng et al., 2024), their overcorrection often leads to low precision, distorting the intended meaning of the original text and reducing the model's reliability.

To fully exploit the strengths of LLMs while overcoming the recall challenges in sLMs, we propose Post-Correction via Overcorrection (PoCO), a novel approach that enables effective utilization of LLM outputs within sLM fine-tuning. PoCO is

[*]Equal contribution.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 28184–28196
November 4-9, 2025

a two-step method that first intentionally triggers overcorrection to maximize the possible correction via LLM and then applies a post-correction step to refine and repair the excessive changes. By leveraging the enriched generative power of LLMs while preserving the reliability of sLMs, PoCO harmonizes recall and precision. Thereby, we aim to enhance GEC quality by ensuring comprehensive error revision and high-quality post-corrections.

We conduct experiments to compare PoCO with robust supervised sLMs as well as few-shot, zero-shot, and fine-tuned LLMs. Our results show that PoCO successfully achieves superior recall scores across all evaluations and maintains competitive precision on both BEA19 development and test sets compared to robust supervised sLMs, highlighting the efficacy of our balancing strategy. Notably, when compared with LLMs, PoCO exhibits competitive performance and even higher $F_{0.5}$ scores in the BEA 19 test set.

To further validate our approach, we compare PoCO with quality estimation-based filtering techniques like GRECO (Qorib and Ng, 2023), as well as robust GEC ensemble systems, such as ESC (Qorib et al., 2022). The results show that our method outperforms GRECO in a single-model setting and also demonstrates superior performance when integrated into ensemble systems. Our novel contributions are as follows:

- We address low-recall issues in robust supervised fine-tuned sLMs with PoCO by leveraging the LLMs' overcorrection tendency, a marked limitation of LLM-based GEC. Consequently, PoCO achieves the highest recall among robust fine-tuned single models.
- To mitigate the typically low precision challenges inherent in LLMs, which reduce the reliability of model outputs, PoCO integrates a novel training strategy centered on a *recovered target*. By incorporating the recovered target, PoCO consistently shows improved precision across all datasets, indicating the efficacy of directly recovering erroneous corrections.
- We investigate the effectiveness of PoCO within a quality estimation system and a regression-based ensemble system, where it demonstrates high-performing results. In ESC settings, the experimental evidence indicates that our model can perform better using fewer models, further highlighting its efficiency.

2 Related Work

GEC Single-model frameworks for GEC primarily follow two mainstream approaches: edit-based and Seq2Seq methods. The edit-based approach predicts edit operations (Malmi et al., 2019; Awasthi et al., 2019; Stahlberg and Kumar, 2020; Omelianchuk et al., 2020; Tarnavskyi et al., 2022), while the Seq2Seq method, inspired by machine translation, treats GEC as a monolingual translation task where erroneous text is rewritten into corrected text (Xie et al., 2016; Yuan and Briscoe, 2016; Ji et al., 2017; Junczys-Dowmunt et al., 2018; Chollampatt and Ng, 2018). Recent advancements, such as Rothe et al. (2021), have demonstrated the effectiveness of Seq2Seq models like T5 (Raffel et al., 2020), which generate error-free sentences directly from erroneous inputs, achieving strong performance.

Various other efforts have been made to enhance GEC by incorporating additional components into a strong single-model correction system (Yuan et al., 2021; Yasunaga et al., 2021; Sorokin, 2022; Zhou et al., 2023). Recently, ensemble methods that combine multiple models have become increasingly prevalent to further enhance performance (Tarnavskyi et al., 2022; Omelianchuk et al., 2024; Qorib et al., 2024). For instance, Qorib and Ng (2023) proposed GRECO, a quality estimation system that scores edits using word label and gap label, while ESC (Qorib et al., 2022) combines robust GEC models using logistic regression to score each edit. However, despite various attempts, these approaches reveal a clear limitation of low recall. While ensembling enhances precision—a key advantage of supervised fine-tuning—it still fails to address the issue of low recall. This suggests that simply ensembling LMs is insufficient to overcome this fundamental limitation.

LLM for GEC While Large Language Models (LLMs) have shown remarkable success across various domains, their application to GEC has faced challenges, particularly in precision. Prompting-based approaches using models like GPT and LLaMA (Coyne et al., 2023; Wu et al., 2023; Katinskaia and Yangarber, 2024; Yang and Quan, 2024) are revealed to struggle with overcorrection, leading to unreliable edits. Studies have explored optimizing prompts (Loem et al., 2023) and employing techniques like Chain-of-Thought (CoT) (Fang et al., 2023), but precision remains a key limitation. To mitigate this, recent high-

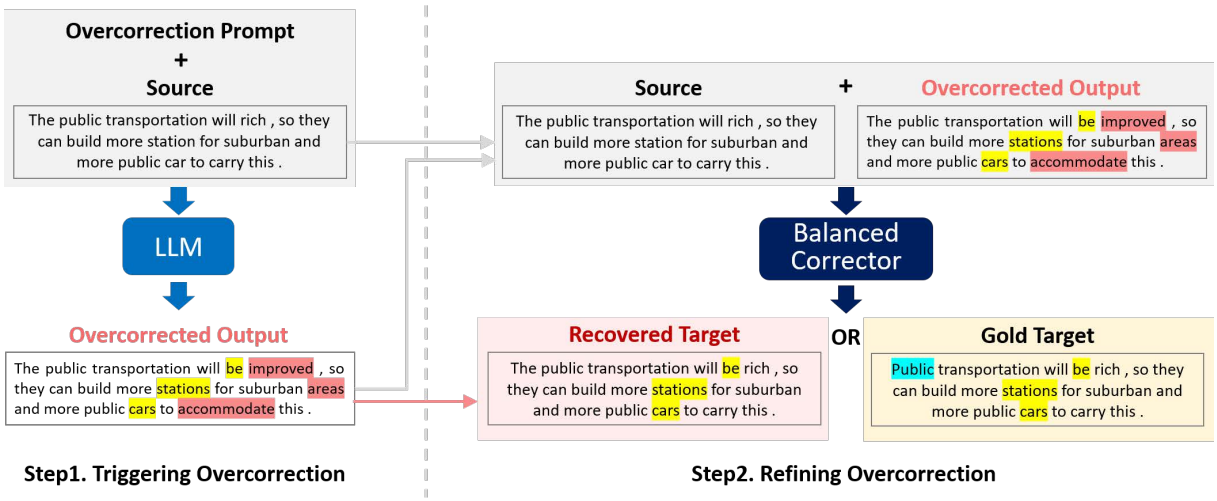

Figure 2: Overview of the POCO Framework: The yellow highlight shows the parts of the LLM-overcorrected output that match the gold target, while the pink highlight indicates the parts not in the gold target. The blue color marks the parts of the gold target missing from the LLM output. The "Recovered Target" includes only the yellow-highlighted portions (matching the gold target), with the rest restored to the original source text.

performing approaches fine-tune LLMs such as LLaMA-7B, LLaMA-13B, T5-11B, and UL2-20B (Omelianchuk et al., 2024), improving overall GEC performance. However, even fine-tuned LLMs tend to exhibit lower precision compared to sLMs, having tendency toward overcorrection.

We propose a novel method that addresses the low recall of sLMs by leveraging complementary strengths of LLMs, which effectively mitigates the limitations of existing methods.

3 PoCO

PoCO leverages the overcorrection tendencies of LLMs to address the low recall commonly observed in robust supervised sLMs, while simultaneously preserving their precision through a targeted post-correction step. Our approach consists of two stages: 1) *Triggering Overcorrection*, which prompts LLM to make extensive corrections, and 2) *Post Correction*, which then refines the overcorrections for improving the overall score. The overall structure of our method is presented in Figure 2.

3.1 Triggering Overcorrection

At this stage, our primary goal is to *improve recall* by leveraging LLMs' tendency to generate extensive corrections. While inducing overcorrection may temporarily reduce precision and impact overall performance, we address this issue in the next step. Here, our focus is on mitigating the conservative correction (low recall) of the supervised fine-tuned model by intentionally triggering overcorrection.

To achieve high recall, we incorporate the instruction "`find as many errors as you can`" to maximize the model's correction capability and actively induce overcorrection. This intentional strategy helps capture a broader range of potential errors, which can be refined in later stages. Unlike prior approaches that passively suffer from overcorrection as an unintended side effect, we systematically integrate it as a deliberate step in our pipeline by explicitly encouraging overcorrection for the GEC task.

Our prompting approach builds on the method proposed by Fang et al. (2023), which utilizes a detailed prompt that suppresses the model's ability to modify with zero-shot CoT (Kojima et al., 2022) reasoning to improve overall performance. However, since restricting the model's correction ability contradicts our objective, we removed the phrase "`keeping the original sentence structure unchanged as much as possible`" from their prompt. This modification allows the model to make more flexible corrections, effectively assisting in overcoming the limitations of overly conservative correction approaches in the next step. Table 1 provides the details of our prompt.

3.2 Post Correction

The generated overcorrected outputs would possibly exhibit high recall but low precision. Therefore, to directly enhance the precision, we suggest a double-target training strategy. Our approach

CoT Prompt
"Please identify and correct any grammatical errors in the following sentence indicated by <input> ERROR </input> tag, you need to comprehend the sentence as a whole before identifying and correcting any errors step by step while keeping the original sentence structure unchanged as much as possible. Afterward, output the corrected version directly without any explanations. Remember to format your corrected output results with the tag <output> Your Corrected Version </output>. Please start: <input> INPUT </input>:"
Overcorrection CoT Prompt
"Please identify and correct any grammatical errors in the following sentence, indicated by <input> ERROR </input> tag. You need to comprehend the sentence as a whole and find as many errors as you can before identifying and correcting them step by step. Afterward, output the corrected version directly without any explanations. Remember to format your corrected output results with the tag <output> Your Corrected Version </output>. Please start: <input> INPUT </input>:"

Table 1: The original CoT prompt (Fang et al., 2023) and our overcorrection triggering prompt used.

utilizes two types of targets: (1) the ground truth (gold) target, which is manually annotated by human annotators, and (2) the *recovered target*, which is generated by our novel method.

Each target serves a distinct purpose. Training with the gold target encourages the model to simultaneously learn two tasks: The first objective is to restore overcorrected words, and the second is to correct new errors that the LLM failed to address in the first stage. Providing a gold target can also effectively improve precision while preserving a substantial portion of recall, making it a practical approach to mitigating overcorrection. However, since precision is often prioritized over recall in the GEC task, we introduce an additional technique, *recovered target*, to further enhance precision, which was intentionally reduced in Section 3.1.

Recovered target prioritizes restoring overcorrected parts during training, rather than primarily focusing on correcting new errors. It is constructed by selectively incorporating correct edits from the gold target, but only within the regions modified by the LLM. To guide the model toward more precise corrections, we create two training pairs for each source sentence: one with the gold target and one with the recovered target. This simple yet effective integration improves model precision by guiding it to reverse overcorrections while preserving valid edits. By jointly training the model on both the gold and recovered targets, PoCO achieves this balance without significantly compromising recall and precision. When training the model, we pair the original source sentences with the LLM-generated sentences using the following format, which serves as the input to the encoder-decoder model:

```
source : source sentence
overcorrect : LLM-generated sentence
```

Our proposed recovered target with the gold target during training could achieve two key objectives: (1) Enabling the model to correct errors that the LLM failed to fix using the gold target (2) significantly improving precision, which drops substantially in Step 1 (Triggering Overcorrection), by guiding the model to concentrate more on recovering overcorrected parts. In addition, we suggest two training strategies. In the PoCO-*Seq* setting, the model is first pre-trained on the gold target and then sequentially fine-tuned on the recovered target. In the PoCO-*Mix* setting, training is conducted using a mixture of gold and recovered target sentences.

4 Experiment

4.1 Datasets

We use the Clang-8 (Rothe et al., 2021)[1] and W&I + LOCNESS (Bryant et al., 2019)[2] datasets for model training. Clang-8 is a refined version of the Lang-8 Corpus of Leaner English (Lang-8) (Tajiri et al., 2012)[3] dataset. We select the Clang-8 dataset for pretraining following the approach used in previous sequence-to-sequence models (Rothe et al., 2021). Note that in pretraining, we only use the Clang-8 dataset without using LLM overcorrection results. Then, to fine-tune our model with LLM-generated overcorrected outputs, we used only the W&I + LOCNESS dataset; thus, triggering the overcorrection step is also conducted solely on this dataset.

For evaluation, we tested our model on the BEA-19 (Bryant et al., 2019) dev and test datasets and the CoNLL-14 (Ng et al., 2014) test set. The precison, recall, $F_{0.5}$ scores are computed using Errant scorer (Bryant et al., 2017) and the M2 scorer (Dahlmeier and Ng, 2012), respectively. The BEA-19 test set can be evaluated on the BEA-19 Shared Task platform[4].

[1] https://github.com/google-research-datasets/clang8?tab=readme-ov-file

[2] https://www.cl.cam.ac.uk/research/nl/bea2019st/data/wi+locness_v2.1.bea19.tar.gz

[3] https://sites.google.com/site/naistlang8corpora

[4] https://www.cl.cam.ac.uk/research/nl/bea2019st

4.2 Settings

LLMs We attempted to replicate the results of previous studies by using GPT-3.5-turbo-0613, the model that shows strong performance in grammatical error correction. However, since OpenAI[5] officially ended support for this version, the model used in the previous study, GPT-3.5-Turbo-0613, is no longer accessible. Thus, we conducted our experiments using one of the available versions of the GPT-3.5 series, GPT-3.5-Turbo-0125 (`GPT3.5-CoT-0125`). To ensure a fair evaluation, we compare the implementation with and without our overcorrection strategy, using the same version of the model to validate the effectiveness of our approach. In previous studies, the GPT-3.5 model generally outperformed the GPT-4 model in terms of recall. Therefore, in the process of inducing overcorrection, our experiments were conducted exclusively using the GPT-3.5 model. The temperature is set to 1.

Fine-tuning Our model is primarily based on the encoder-decoder architecture. In particular, we conducted all the experiments based on the open-source T5 model[6], which had been pretrained on the Clang-8 dataset (Rothe et al., 2021). The target data is described in Section 3.2. The model is fine-tuned with a learning rate of 1e-4, batch size of 64, and 10 epochs.

4.2.1 Baseline models

Fine-tuned sLMs For robust fine-tuning models, we include not only single models such as Seq2Edits (Stahlberg and Kumar, 2020), Tagged-Corruption (Stahlberg and Kumar, 2021), and MoECE-GS-Large (Qorib et al., 2024), but also approaches that enhance performance by attaching auxiliary scorer like EditScorer (Sorokin, 2022). For EditScorer, CTC-Copy, and GECToR-2024, we report the results re-evaluated by Omelianchuk et al. (2024). EditScorer, originally proposed by Sorokin (2022), is evaluated based on its open-source[7] implementation, using the GECoR-XLNet$^{(L)}$ option from Tarnavskyi et al. (2022) with a RoBERTa-Large encoder as the scorer. CTC-Copy is initially introduced by Zhang et al. (2023), and the reported performance in our paper is the implementation of official code[8] using a RoBERTa encoder.

[5] https://platform.openai.com/docs/models
[6] https://github.com/gotutiyan/gec-t5
[7] https://github.com/AlexeySorokin/EditScorer
[8] https://github.com/yzhangcs/ctc-copy

Model	CoNLL-14 test			BEA-19 dev		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$
Chat-LLama2-7B [1]	42.9	47.3	43.7	19.1	34.1	21.0
Chat-LLama2-13B [1]	49.1	56.1	50.4	30.6	45.0	32.7
GPT3.5-0613 [1]	56.2	57.7	56.5	37.4	50.6	39.4
GPT4 [1]	59.0	55.4	58.2	42.5	45.0	43.0
GPT3.5-CoT-0613 [1]	56.0	58.7	56.5	36.4	50.8	38.5
GPT3.5-CoT-0613 [2]	55.8	58.5	56.3	-	-	-
GPT3.5-CoT-0125 [2]*	59.2	53.9	58.0	52.8	51.7	52.6
GPT3.5-CoT-O.C-0125	55.0	57.9	55.5	48.2	54.8	49.4

Table 2: Performance of LLM Zero-shot on the CoNLL-14 test and BEA-19 dev datasets. [1] indicates results reported by Omelianchuk et al. (2024), while [2] refers to results following the prompt proposed by Fang et al. (2023). * denotes our implementation using the currently available version of GPT (GPT-3.5-0125), and O.C denotes the inclusion of our overcorrection strategy. P and R are precision and recall, respectively.

GECToR-2024 is the method originally proposed by Omelianchuk et al. (2020), but reported performance in our result is a new version of GECTor where additional training was applied to GECToR-RoBERTa$^{(L)}$ to further enhance performance.

LLMs When comparing our overcorrection method, we got results reported by Omelianchuk et al. (2024) and Katinskaia and Yangarber (2024). Since Fang et al. (2023) did not explicitly specify the used GPT version, we report results from Katinskaia and Yangarber (2024) using the same prompt with Fang et al. (2023). In post-correction, we compare our model against two types of LLM baselines: (1) LLMs that enhance performance through few-shot prompting and (2) LLMs fine-tuned specifically for the GEC task. GPT-3.5 few-shot prompting baselines were referenced from Fang et al. (2023), and the 16-shot GPT-3 setting results were taken from Loem et al. (2023). For the fine-tuned LLM baseline, we use the results reported in Omelianchuk et al. (2024), where each fine-tuned LLMs were leveraged in an ensemble to achieve state-of-the-art performance.

5 Result

5.1 Triggering Overcorrection

The results in Table 2 highlight the effectiveness of our triggering overcorrection prompting method in improving recall while maintaining competitive precision. Our method shows consistent improvement in recall scores compared to the prompt proposed in Fang et al. (2023) when applied to GPT-3.5-Turbo-0125 across both the CoNLL-14 test and

Model	CoNLL-14 Test			BEA-19 Dev			BEA-19 Test			Training Data Size (#tokens)
	Precision	Recall	$F_{0.5}$	Precision	Recall	$F_{0.5}$	Precision	Recall	$F_{0.5}$	
Seq2Edits	69.9	44.4	62.7	-	-	-	72.7	62.9	70.5	-
Tagged-Corruption	72.8	49.5	66.6	59.5	41.3	54.7	72.1	64.4	70.4	-
EditScorer	**78.5**	39.4	65.5	**67.3**	36.1	57.4	**81.0**	56.1	74.4	96.57M
CTC-Copy	72.6	47.0	65.5	58.3	38.0	52.7	71.7	59.9	69.0	30.24M
GECToR-2024	75.0	44.7	66.0	64.6	37.2	56.3	77.7	59.0	73.1	124.57M
MoECE-GS-Large	74.3	50.2	**67.8**	-	-	56.4	76.9	64.5	74.1	28.00M
PoCO-*Seq*	70.2	48.9	64.6	63.0	42.4	57.4	78.6	64.3	75.3	28.62M
PoCO-*Mix*	69.5	**51.9**	65.1	62.3	**45.3**	**57.9**	78.0	**67.8**	**75.7**	28.62M

Table 3: Performance comparison between supervised fine-tuned models and PoCO (large) across different test sets.

Model	CoNLL-14 Test			BEA-19 Dev			BEA-19 Test			
	Precision	**Recall**	**$F_{0.5}$**	**Precision**	**Recall**	**$F_{0.5}$**	**Precision**	**Recall**	**$F_{0.5}$**	**Var**
Triggered Overcorrection	55.1	57.9	55.6	48.3	54.8	49.4	51.9	69.4	54.6	-
PoCO-base-*Gold*	66.2	**51.4**	62.6	56.9	**44.1**	53.8	73.1	**67.9**	72.0	-
PoCO-base-*Recovered*	69.0	44.0	62.0	61.8	36.9	54.4	78.0	59.8	73.6	-
PoCO-base-*Seq*	**69.4**	44.5	62.4	**63.0**	38.2	55.8	**78.8**	60.2	74.2	-
PoCO-base-*Mix*	69.2	49.1	**63.2**	62.5	43.2	**57.4**	77.2	65.7	**74.6**	74.47 ± 0.22
PoCO-large-*Gold*	68.6	**55.1**	**65.4**	59.0	**47.9**	56.3	73.9	**69.7**	73.0	-
PoCO-large-*Recovered*	69.8	48.9	64.3	62.2	41.4	56.5	78.2	63.4	74.7	-
PoCO-large-*Seq*	**70.2**	48.9	64.6	**63.0**	42.4	57.4	**78.6**	64.3	75.3	-
PoCO-large-*Mix*	69.5	51.9	65.1	62.3	45.3	**57.9**	78.0	67.8	**75.7**	75.64 ± 0.1

Table 4: Results comparing the impact of different target combinations. Triggered overcorrection indicates the results of GPT3.5-CoT-O.C-0125 in Table 2. Variance values (Var) of the BEA-19 test are computed across three independent runs with different random seeds.

BEA-dev datasets. Furthermore, excluding recall derived from the unavailable GPT-3.5-Turbo-0613, our approach achieves the highest recall in both datasets.

5.2 PoCO Main Results

Comparison with sLMs We successfully address the low recall issue of robust, smaller, supervised fine-tuning models. As shown in Table 3, our model achieves the highest recall on the CoNLL-14 test set and BEA-19 dev and test sets. Additionally, it also records the highest $F_{0.5}$ scores on both BEA-19 dev and test sets. Additionally, our approach demonstrates advantages in terms of data efficiency. Our method pretrains the model using the CLang-8 dataset, the dataset used in MoECE-GS-Large (Qorib et al., 2024), and fine-tunes on a relatively small W & I dataset with a token size of 628.7k. Unlike previous studies that attempted to enhance model performance by generating large-scale synthetic data, our approach demonstrates that fine-tuning a robust model with a small dataset can achieve significant performance improvements.

Effects of recovered targets In addition to the two methods proposed in Section 3.2, we also conduct experiments using only the gold target and the recovered target, respectively. The results for each training approach are presented in Table 4. Our findings clearly demonstrate the advantages of training with the recovered targets we proposed. When trained solely with gold targets, the model generally exhibits high recall, but when compared to using the recovered target for training, the precision tends to show low scores, ultimately leading to a decline in $F_{0.5}$ score. In contrast, training exclusively with recovered targets leads to a substantial improvement in precision scores, significantly enhancing the reliability of the model. When the recovered target is combined with the gold target, it consistently achieves the best performance on $F_{0.5}$ scores across nearly all experiments. Notably, in the PoCO-*Mix* setting, both precision and recall improved in a balanced manner. Additionally, there is a significant difference in recall between the base model and the large model, with the large model generally being more effective in preserving the high recall achieved by the LLM. For precision, the experimental results using the recovered target show minimal variation across different model sizes (T5-base vs. T5-large). This indicates that

Model	CoNLL-14 Test			BEA-19 Dev			BEA-19 Test		
	Precision	Recall	$F_{0.5}$	Precision	Recall	$F_{0.5}$	Precision	Recall	$F_{0.5}$
GPT-3.5-1-shot-CoT (Fang et al., 2023)	52.0	58.1	53.1	42.5	**55.6**	44.6	34.6	69.7	38.4
GPT-3.5-3-shot-CoT (Fang et al., 2023)	51.3	**62.4**	53.2	-	-	-	34.0	70.2	37.9
GPT-3.5-5-shot-CoT (Fang et al., 2023)	50.9	61.8	52.8	-	-	-	32.4	69.9	36.3
GPT-3-16-shot (Loem et al., 2023)	-	-	57.1	-	-	-	-	-	57.4
Chat-LLaMa-2-7B-FT (Omelianchuk et al., 2024)	75.5	46.8	67.2	58.3	46.0	55.3	72.3	67.4	71.2
Chat-LLaMa-2-13B-FT (Omelianchuk et al., 2024)	**77.3**	45.6	**67.9**	59.8	46.1	56.4	74.6	67.8	73.1
T5-11B (Omelianchuk et al., 2024)	70.9	56.5	67.5	60.9	51.1	**58.6**	73.2	**71.2**	72.8
UL2-20B (Omelianchuk et al., 2024)	73.8	50.4	67.5	60.5	48.6	57.7	75.2	70.0	74.1
PoCO-*Seq*	70.2	48.9	64.6	**63.0**	42.4	57.4	**78.6**	64.3	75.3
PoCO-*Mix*	69.5	51.9	65.1	62.3	45.3	57.9	78.0	67.8	**75.7**

Table 5: Performance comparison between LLM-based models and PoCO (based on the T5-large model) across different test sets. **Bold** and underlined texts indicate the highest and the second highest value, respectively.

No. of Base Systems	ESC			No. of Base Systems	Ours (ensemble)		
	Precision	Recall	$F_{0.5}$		Precision	Recall	$F_{0.5}$
T5-Large + GECTor XLNet	80.2	61.1	75.5	PoCO-*Mix* + Triggered Overcorrection	84.6	61.8	78.8
+ GECToR Roberta	84.7	59.0	77.9	+ PoCO-*Gold*	85.2	61.5	79.1
+ Riken & Tohoku	86.2	59.4	79.0	+ GECTor XLNet	85.8	63.2	80.0
+ UEDIN-MS	86.2	61.1	79.6	+ GECTor Roberta	86.1	62.9	80.2
+ Kakao & Brain	86.7	60.9	**79.9**	+ Riken & Tohoku	86.2	63.6	**80.5**
				+ UEDIN-MS	87.6	62.6	81.1
				+ Kakao & Brain	87.5	63.0	**81.2**

Table 6: Comparison of the original ESC and our ensemble results using ESC with different base system combinations. Models in each row are incrementally added. Our PoCO results (PoCO-*Mix*, PoCO-*Gold*) correspond to large-model settings.

our method consistently contributes to improving precision regardless of model scale, demonstrating its robustness and effectiveness across different parameter sizes.

Reference-free evaluation We additionally report reference-free evaluation using GPT-4.1 as an automatic judge, following the framework of Kobayashi et al. (2024) with minor modifications to the BEA-19 dev setup. GPT-4.1 directly assessed 50 randomly sampled system outputs on a 1–5 scale for *grammaticality*, *fluency*, and *meaning preservation*, thereby capturing aspects often missed by overlap-based metrics (Grundkiewicz et al., 2015; Sakaguchi et al., 2016; Yoshimura et al., 2020; Gong et al., 2022; Kobayashi et al., 2024; Östling et al., 2024). Both PoCo-*Mix* and PoCo-*Seq* achieved consistently high scores (over 4.7 in all dimensions), with both reaching 4.98 in meaning preservation, confirming that our method produces outputs of high linguistic quality and semantic fidelity. Details are shown in Appendix A.

6 Discussions and Analysis

Comparison with LLMs Table 5 demonstrates the effectiveness of our method compared to LLM-based few-shot and fine-tuning approaches. Providing few-shot prompting to LLMs can enhance recall, but it does not help to improve precision, which remains a critical limitation. However, our approach, which leverages supervised sLMs with 770M parameters, exhibits competitive performance even when compared to directly fine-tuned LLMs with over 7 billion parameters. Notably, our approach achieves the highest precision on both the BEA-19 test and dev sets, and even obtaining the highest $F_{0.5}$ score on the BEA-19 test set.

PoCO with ESC Recently, models that achieve strong performance in GEC on $F_{0.5}$ scores are primarily based on ensemble methods which combines multiple models. We aimed to demonstrate the effectiveness of our model by applying it to a high-performing ESC system, which ensembles robust fine-tuned model by using a simple linear regression method. To show a direct comparison with previously published results, we conducted experiments exclusively on the BEA-19 test set. To ensure model diversity, we incorporated three models: LLM with an overcorrection prompt, PoCO-*Gold*, and PoCO-*Mix*. As shown in Table 6, applying our proposed three models resulted in superior perfor-

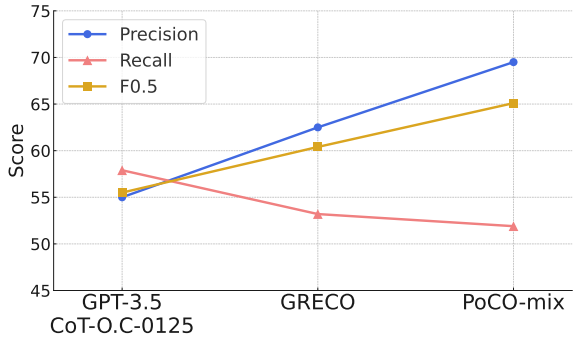

Figure 3: Illustration of performance comparison with GRECO, showing that PoCO substantially outperforms in both precision and $F_{0.5}$ on the CoNLL-14 test set.

mance compared to the three models used in the existing ESC system. Furthermore, when we incorporated the GECToR-XLNet model used in the ESC system into ours, we achieved a performance of 80.0 using only four models, surpassing the 79.9 score obtained with the original six-model ensemble. When combining our models with those used in the ESC system, we achieved a performance of 81.2. This result is competitive with the 81.4 scores reported in a state-of-the-art study (Omelianchuk et al., 2024), which uses fine-tuned LLMs for the ensemble.

PoCO vs GRECO To evaluate the effectiveness of PoCO against existing quality estimation methods, we conduct additional experiments using the top-5 GPT-generated corrections instead of the Riken&Tohoku outputs. As shown in Figure 3, PoCO-*Mix* outperforms GRECO on the CoNLL-14 test set, achieving notably higher precision (+7.0) and $F_{0.5}$ (+4.9). Although its recall is slightly lower than that of GRECO, PoCO-*Mix* still remarkably improves recall over its comparison sLMs (Table 3), demonstrating a better precision–recall trade-off and superior $F_{0.5}$ score overall.

LLMs for post-correction We investigated whether LLMs alone could effectively perform post-correction without relying on our proposed PoCO method. To evaluate this, we conducted experiments using GPT-4o-mini and GPT-4-Turbo, testing their ability to correct overcorrections made by LLMs. In particular, we offer information on overcorrection and undercorrection when providing a fine-tuned T5-base model's output to LLM, so that it can recognize what to recover. Additionally, inspired by the findings of (Ryu et al., 2024), which demonstrated that highlighting key

LLM Post Correct Prompt
Please identify and correct any grammatical errors in the source sentence while avoiding unnecessary changes (overcorrections) and insufficient edits (undercorrections). Correction Types: - Source: The original sentence before any corrections. - Overcorrect: Contains unnecessary modifications but also many correct edits, leading to high recall but low precision. - Undercorrect: Makes fewer incorrect changes but misses valid corrections, resulting in higher precision but lower recall. Hint Tags: - <R>...</R>: Indicates a replaced part of the sentence. - <M>...</M>: Marks an addition that wasn't in the source. - <U>...</U>: Highlights a removed part of the original sentence. Your task is to provide the best possible correction for the given source sentence, ensuring proper grammar and clarity while preserving its intended meaning. Use the "Overcorrect" (high recall) and "Undercorrect" (high precision) sentences, along with hint tags, to guide your edits. Input Data: Source: {{Source}} Overcorrect: {{Overcorrected output}} Undercorrect: {{Undercorrected output}} Perfect Correction:

Table 7: The prompt for Post-correction with LLMs.

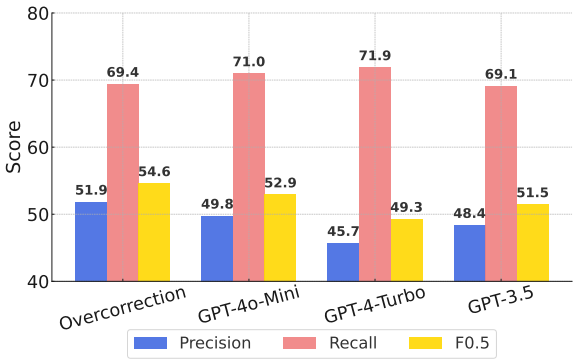

Figure 4: Precision, recall, and $F_{0.5}$ scores on BEA-19 test set when using LLMs (*GPT-4o-Mini, GPT-4-Turbo, and GPT-3.5*) for post-correction after our triggered *Overcorrection*, instead of our PoCO method.

areas improves model performance, we tagged crucial segments with the following tags: <R>, <M>, <U>, which indicate *Replace, Missing, Unnecessary* words, respectively. Specifically, we marked modifications between the source sentence and the LLM-overcorrected output, as well as between the source sentence and the high-precision T5 output, giving the model explicit correction cues. The full prompt for this experiment is detailed in Table 7.

We expected this setup to somewhat assist LLMs in balancing under-correction and overcorrection, producing a well-revised output; however, the results in Figure 4 indicate different results. Instead of improving, LLM-based correction actually led to a drop in overall performance. While recall increased, precision declined, making the final out-

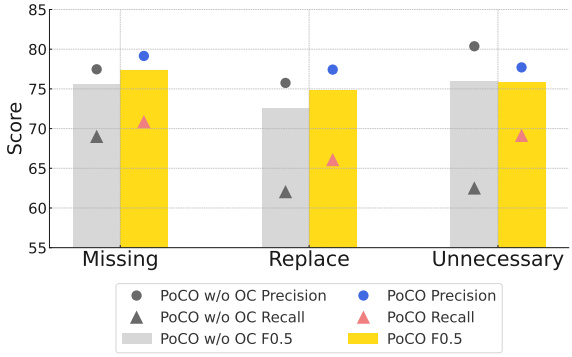

Figure 5: Performance comparison of PoCO w/o overcorrection and PoCO by error type (*Missing, Replace, Unnecessary*).

put less reliable. Notably, GPT-4-Turbo suffered an even greater performance drop compared to GPT-4o-mini. These findings further support a key takeaway that LLMs alone struggle with overcorrection in GEC tasks, as previously reported in prior studies (Loem et al., 2023; Fang et al., 2023; Wu et al., 2023; Coyne et al., 2023). PoCO, on the contrary, successfully tackles this issue, proving to be far more effective than simply relying on LLM-based correction.

Analysis of error types We further analyze results by error type, i.e., *Missing*, *Replace*, and *Unnecessary* (Figure 5). *Missing* means essential words are omitted, *Replace* refers to errors where an incorrect word or form must be substituted, and *Unnecessary* indicates redundant words that should be deleted. These results are based on the official BEA-19 Shared Task evaluation, where system submissions automatically provide error-type–level performance. We removed the intentional overcorrection step and compared the resulting model with the full PoCO pipeline on the BEA-19 test set. Our overcorrection strategy consistently improves recall across all error types, fulfilling its intended goal. While *Unnecessary* errors show a moderate trade-off with precision, the gains in recall remain substantial. Notably, in *Missing* and *Replace* cases, recall gains were accompanied by higher precision, showing that first guiding the model toward overcorrection is particularly effective at recovering omitted content and fixing incorrect tokens.

7 Conclusion

In this work, we introduced Post-Correction via Overcorrection (PoCO), a novel framework designed to address the precision-recall trade-off in grammatical error correction. By leveraging LLM-driven overcorrection to maximize recall and introducing targeted post-correction using fine-tuned smaller models to refine erroneous outputs, PoCO effectively balances precision and recall. Our extensive experiments demonstrate that PoCO enhances overall GEC performance by successfully mitigating overcorrection while maintaining the precision robustness of supervised models. These findings highlight the potential of integrating LLMs and smaller models for more accurate and reliable grammatical error correction.

Limitations

We have explored LLMs' ability to handle overcorrection; however, there is currently a lack of dedicated studies investigating how different prompting techniques influence overcorrection in LLMs. Consequently, we adopted prompting methods from prior studies, which primarily focused on general GEC rather than explicitly addressing overcorrection. While our approaches provide a reasonable baseline for guiding overcorrection, they may not be fully optimized.

Also, our findings indicate that LLMs alone face challenges in effectively correcting overcorrection errors in GEC tasks. This highlights the necessity of further research into prompting strategies that can more effectively guide LLMs. Given the lack of systematic studies on how different prompting techniques influence LLMs' ability to mitigate overcorrection errors, we anticipate that exploring more targeted and adaptive prompting methods will enhance LLM performance in post-correction tasks, leading to improvements in both recall and precision in GEC outputs.

Acknowledgments

This research was partly supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2025-RS-2020-II201789) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation; 47.5%), by Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2025 (Project Name: Development of an AI-Based Korean Diagnostic System for Efficient Korean Speaking Learning by Foreigners, Project Number: RS-2025-02413038; 47.5%),

and by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2019-II191906, Artificial Intelligence Graduate School Program (POSTECH); 5%).

Ethical Statement

This study exclusively utilizes publicly available datasets of grammatical error correction, where personal or other sensitive information is not included, adhering to ethical guidelines and policies.

References

Waad Alsaweed and Saad Aljebreen. 2024. Investigating the accuracy of chatgpt as a writing error correction tool. *Int. J. Comput.-Assist. Lang. Learn. Teach.*, 14(1):1–18.

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.

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.

Eduardo Calò, Léo Jacqmin, Thibo Rosemplatt, Maxime Amblard, Miguel Couceiro, and Ajinkya Kulkarni. 2021. GECko+: a grammatical and discourse error correction tool. In *Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 3 : Démonstrations*, pages 8–11, Lille, France. ATALA.

Wen Chen and Guo-qiang Liu. 2021. Effectiveness of corrective feedback: Teachers' perspectives. *Iranian Journal of Language Teaching Research*, 9(1):23–42.

Shamil Chollampatt and Hwee Tou Ng. 2018. A multilayer convolutional encoder-decoder neural network for grammatical error correction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Steven Coyne, Keisuke Sakaguchi, Diana Galvan-Sosa, Michael Zock, and Kentaro Inui. 2023. Analyzing the performance of gpt-3.5 and gpt-4 in grammatical error correction. *Preprint*, arXiv:2303.14342.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.

Tao Fang, Shu Yang, Kaixin Lan, Derek F. Wong, Jinpeng Hu, Lidia S. Chao, and Yue Zhang. 2023. Is chatgpt a highly fluent grammatical error correction system? a comprehensive evaluation. *Preprint*, arXiv:2304.01746.

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.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Edward Gillian. 2015. Human evaluation of grammatical error correction systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 461–470, Lisbon, Portugal. Association for Computational Linguistics.

Jianshu Ji, Qinlong Wang, Kristina Toutanova, Yongen Gong, Steven Truong, and Jianfeng Gao. 2017. A nested attention neural hybrid model for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 753–762, Vancouver, Canada. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 595–606, New Orleans, Louisiana. Association for Computational Linguistics.

Masahiro Kaneko, Sho Takase, Ayana Niwa, and Naoaki Okazaki. 2022. Interpretability for language learners using example-based grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7176–7187, Dublin, Ireland. Association for Computational Linguistics.

Anisia Katinskaia and Roman Yangarber. 2023. Grammatical error correction for sentence-level assessment in language learning. In *Proceedings of the 18th*

Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023), pages 488–502, Toronto, Canada. 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.

Satoru Katsumata and Mamoru Komachi. 2020. Stronger baselines for grammatical error correction using a pretrained encoder-decoder model. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 827–832, Suzhou, China. Association for Computational Linguistics.

Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024. Large language models are state-of-the-art evaluator for grammatical error correction. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 68–77, Mexico City, Mexico. Association for Computational Linguistics.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Kai-Hui Liang, Sam Davidson, Xun Yuan, Shehan Panditharatne, Chun-Yen Chen, Ryan Shea, Derek Pham, Yinghua Tan, Erik Voss, and Luke Fryer. 2023. ChatBack: Investigating methods of providing grammatical error feedback in a GUI-based language learning chatbot. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 83–99, Toronto, Canada. Association for Computational Linguistics.

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. *Preprint*, arXiv:2305.18156.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. 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 5054–5065, Hong Kong, China. 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 Skurzhanskyi. 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 Skurzhanskyi, Artem Chernodub, Oleksandr Korniienko, 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.

Robert Östling, Katarina Gillholm, Murathan Kurfalı, Marie Mattson, and Mats Wirén. 2024. Evaluation of really good grammatical error correction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6582–6593, Torino, Italia. ELRA and ICCL.

Muhammad Reza Qorib, Alham Fikri Aji, and Hwee Tou Ng. 2024. Efficient and interpretable grammatical error correction with mixture of experts. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 17127–17138, Miami, Florida, USA. 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.

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.

Sangwon Ryu, Heejin Do, Yunsu Kim, Gary Geunbae Lee, and Jungseul Ok. 2024. Key-element-informed sllm tuning for document summarization. In *Interspeech 2024*, pages 1940–1944.

Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel Tetreault. 2016. Reassessing the goals of grammatical error correction: Fluency instead of grammaticality. *Transactions of the Association for Computational Linguistics*, 4:169–182.

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.

Felix Stahlberg and Shankar Kumar. 2020. Seq2Edits: Sequence transduction using span-level edit operations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5147–5159, Online. Association for Computational Linguistics.

Felix Stahlberg and Shankar Kumar. 2021. Synthetic data generation for grammatical error correction with tagged corruption models. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 37–47, Online. Association for Computational Linguistics.

Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. Tense and aspect error correction for ESL learners using global context. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 198–202, Jeju Island, Korea. Association for Computational Linguistics.

Maksym Tarnavskyi, 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.

Joel Tetreault and Claudia Leacock. 2014. Automated grammatical error correction for language learners. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Tutorial Abstracts*, pages 8–10, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Yixuan Wang, Baoxin Wang, Yijun Liu, Dayong Wu, and Wanxiang Che. 2024. LM-combiner: A contextual rewriting model for Chinese grammatical error correction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10675–10685, Torino, Italia. ELRA and ICCL.

Haoran Wu, Wenxuan Wang, Yuxuan Wan, Wenxiang Jiao, and Michael Lyu. 2023. Chatgpt or grammarly? evaluating chatgpt on grammatical error correction benchmark. *Preprint*, arXiv:2303.13648.

Ziang Xie, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y. Ng. 2016. Neural language correction with character-based attention. *Preprint*, arXiv:1603.09727.

Haihui Yang and Xiaojun Quan. 2024. Alirector: Alignment-enhanced Chinese grammatical error corrector. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2531–2546, Bangkok, Thailand. Association for Computational Linguistics.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2021. LM-critic: Language models for unsupervised grammatical error correction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7752–7763, Online and Punta Cana, Dominican Republic. 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.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386, San Diego, California. Association for Computational Linguistics.

Zheng Yuan, Shiva Taslimipoor, Christopher Davis, and Christopher Bryant. 2021. Multi-class grammatical error detection for correction: A tale of two systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8722–8736, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Min Zeng, Jiexin Kuang, Mengyang Qiu, Jayoung Song, and Jungyeul Park. 2024. Evaluating prompting strategies for grammatical error correction based on language proficiency. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6426–6430, Torino, Italia. ELRA and ICCL.

Yu Zhang, Yue Zhang, Leyang Cui, and Guohong Fu. 2023. Non-autoregressive text editing with copy-aware latent alignments. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7075–7085, Singapore. Association for Computational Linguistics.

Houquan Zhou, Yumeng Liu, Zhenghua Li, Min Zhang, Bo Zhang, Chen Li, Ji Zhang, and Fei Huang. 2023. Improving Seq2Seq grammatical error correction via decoding interventions. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7393–7405, Singapore. Association for Computational Linguistics.

A Reference-free Evaluation

The full prompt used for the reference-free evaluation described in Section 5.3 is provided in Table 8. For the trait-specific instruction, we directly adopted the instruction prompts from Kobayashi et al. (2024). However, unlike their setup which presents multiple target sentences at once, our evaluation was conducted by providing a single source sentence and a single target sentence for scoring. The complete results are reported in Table 9.

Modified Reference-free Evaluation Prompt

The goal of this task is to rank the presented targets based on the quality of the sentences. The source input consists of a single sentence written by an English learner. Please assign a score from 1 point to 5 points to each target based on the quality of the sentence (note that you can assign the same score multiple times).
{trait-specific instruction}
source
{source input}
target sentences
{target sentence}
output format
The output should be a markdown code snippet formatted in the following schema, including the leading and trailing "```json" and "```":
```json {
"target_score": int // assigned score for target sentence
}
```

Table 8: Prompts used for reference-free evaluation. We adapt the original essay-level setting into a sentence-level prompt suitable for BEA-19 dev evaluation.

Model	Fluency	Grammaticality	Meaning
PoCO-*Mix*	4.70	4.78	4.92
PoCO-*Seq*	4.72	4.72	4.92
PoCO-*Recovered*	4.74	4.80	4.92
PoCO-*Gold*	4.68	4.80	4.96

Table 9: Reference-free evaluation results (1–5 scale) across different PoCO-large variants.