Edit-Wise Preference Optimization for Grammatical Error Correction

Jiehao Liang, Haihui Yang, Shiping Gao, Xiaojun Quan*
School of Computer Science and Engineering, Sun Yat-sen University, China
{liangjh226, yanghh29, gaoshp}@mail2.sysu.edu.cn
quanxj3@mail.sysu.edu.cn

Abstract

While large language models (LLMs) have achieved remarkable success in various natural language processing tasks, their strengths have yet to be fully demonstrated in grammatical error correction (GEC). This is partly due to the misalignment between their pre-training objectives and the GEC principle of making minimal edits. In this work, we aim to bridge this gap by introducing a novel method called Edit-wise Preference Optimization (EPO). By distinguishing the importance of different tokens and assigning higher reward weights to edit tokens during preference optimization, our method captures fine-grained distinctions in GEC that traditional preference learning often overlooks. Extensive experiments on both English and Chinese datasets show that our framework consistently outperforms strong baselines, achieving state-of-the-art performance and demonstrating the advantages of LLMs in GEC.

1 Introduction

Grammatical error correction (GEC) is a task aimed at detecting and correcting potential grammatical errors in given sentences with minimal edits (Bryant et al., 2023). GEC models have widespread applications in areas such as automatic speech recognition (Liao et al., 2023), writing assistants (Knill et al., 2019), and search engines (Ye et al., 2023a). Traditional GEC methods can be divided into two categories: Sequence-to-Edit (Seq2Edit) and Sequence-to-Sequence (Seq2Seq). The Seq2Edit approach frames GEC as a sequence labeling task by predicting the appropriate edit operation for each token (Omelianchuk et al., 2020; Stahlberg and Kumar, 2020), while the Seq2Seq approach treats GEC as a monolingual machine translation task using an encoder-decoder architecture (Zhang et al., 2022b; Zhou et al., 2023b).

Src.	He might be wanted to guard the national image.
Pos.	He might want to guard the national image. ✓
Neg.	He could be wanted to guard the national image. ✗
Src.	On the other side, she don't likes cats at all.
Pos.	On the other side, she doesn't like cats at all. ✓
Neg.	On the one hand, she didn't likes cats at all. ✗

Table 1: Two examples of GEC. "Src./Pos./Neg." denote source, positive, and negative sentences. Underlined texts show edit spans, while red highlights pivot tokens.

Recently, decoder-only large language models (LLMs)[1], such as GPT series (Achiam et al., 2023) and LLaMA series (Touvron et al., 2023), have demonstrated remarkable performance and potential across various NLP tasks. However, initial studies suggest that these LLMs struggle to surpass traditional Seq2Seq models on the GEC task (Qu and Wu, 2023; Zhang et al., 2023; Yang and Quan, 2024). Moreover, LLMs often lead to overcorrection due to their pre-training objective of maximizing the likelihood of the next token, which conflicts with the GEC principle of making minimal edits (Omelianchuk et al., 2024; Davis et al., 2024).

Essentially, the above issue arises from the misalignment between LLMs and human expectations in the GEC task. In this work, we aim to enhance the GEC capabilities of LLMs by leveraging established LLM alignment techniques (Wang et al., 2024). Among these techniques, Direct Preference Optimization (DPO) (Rafailov et al., 2023) is widely used due to its effectiveness and simplicity in aligning LLMs with human intentions. DPO treats policy LLMs as reward models and optimizes them by maximizing the reward gap between positive (preferred) and negative (dispreferred) samples. In the GEC task, where differences between positive and negative samples typically involve only a

*Corresponding author.

[1]From here on, we will use "LLMs" to refer to decoder-only large language models (LLMs).

Proceedings of the 31st International Conference on Computational Linguistics, pages 3401–3414
January 19–24, 2025.

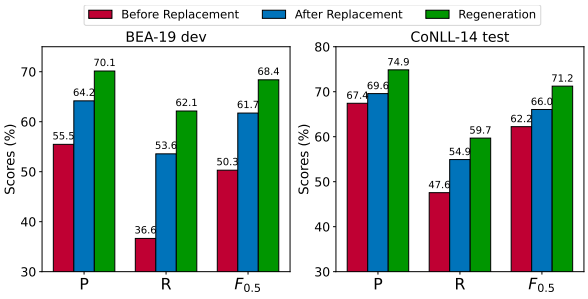

Figure 1: Results of our simulation experiment on BEA-19 dev set and CoNLL-14 test set using LLaMA2-7B. The pivot tokens in edit spans of negative samples were replaced with corrected ones, followed by the model regenerating the subsequent tokens. These results highlight the specific importance of pivot tokens.

few tokens due to the minimal edit principle, these subtle distinctions are crucial for model learning. However, DPO assumes that all tokens contribute equally during preference learning and fails to account for token-specific importance in GEC. Hence, directly applying DPO is unable to capture these fine-grained variations, which leads to suboptimal performance in GEC, as demonstrated in Table 2.

Therefore, we argue that alignment techniques should focus on increasing the reward gap between positive and negative samples specifically at the positions where edits are needed. Building on this premise, we propose Edit-wise Preference Optimization (EPO). Our approach begins with a warm-up training phase to develop an initial model with basic grammatical error correction capabilities. We then sample several candidate sentences from this model for each training input. The sentence with the largest edit distance from the ground truth is selected as the negative sample, while the ground truth itself serves as the positive sample. This sampling method does not require human annotations or external reward models, making it both cost-effective and easy to implement.

After collecting the preference pairs, we employ a GEC parsing tool (Bryant et al., 2017) to identify the differences between the sample pairs, referred to as *edit spans*, as shown in Table 1. To encourage the model to pay more attention to these edit spans during preference optimization, we assign higher reward weights to these edit tokens than other tokens. Moreover, to mitigate the risk of error accumulation during decoding, we assume that the first token of each edit span, referred to as the *pivot token*, is crucial for guiding the model in correcting subsequent edit tokens. For this reason, we further increase the reward weights of pivot tokens. To demonstrate the role of pivot tokens, we conducted a simulation experiment in which we manually corrected the pivot tokens in the negative samples and asked the model to regenerate the subsequent tokens. As shown in Figure 1, this correction and subsequent token regeneration resulted in a 6.7 and 5.2 point increase in $F_{0.5}$ score, respectively, compared to simple token replacement. These findings emphasize the critical role of pivot tokens.

To validate the effectiveness of EPO, we conduct experiments on two English and two Chinese GEC datasets. Our results show that EPO consistently outperforms baseline models, improving LLMs capabilities in grammatical error correction and achieving state-of-the-art performance among single-model approaches. Moreover, our method offers insights for tasks that require a focus on token-wise preference learning. Source code and scripts are available at https://github.com/liangjh2001/EPO-GEC.

2 Related Work

2.1 Traditional GEC Methods

Traditional GEC methods mainly fall into two categories: Seq2Edit and Seq2Seq.

Seq2Edit typically treats GEC as a sequence labeling task. Given an input sentence, Seq2Edit models predict the corresponding edit operation for each token, such as insertion, deletion, and replacement (Awasthi et al., 2019; Stahlberg and Kumar, 2020; Li et al., 2022; Yakovlev et al., 2023). As a milestone work, GECToR (Omelianchuk et al., 2020) further introduces task-specific token transformations on top of basic token-level edit operations, achieving high-precision results.

Seq2Seq treats GEC as a monolingual machine translation task solved with encoder-decoder architectures (Zhao and Wang, 2020; Rothe et al., 2021; Ye et al., 2023b; Zhou et al., 2023b). Unlike Seq2Edit which performs localized corrections, Seq2Seq aims to generate the entire corrected sentence. More advanced Seq2Seq approaches (Zhang et al., 2022b; Li et al., 2023; Fang et al., 2023a) further enhance performance by incorporating additional information, such as syntactic features or error detection signals.

2.2 LLMs for GEC

While decoder-only LLMs have achieved notable success in many NLP tasks, they face challenges

in GEC, primarily due to the conflict between their generative nature and the minimal edit principle of GEC (Coyne et al., 2023; Qu and Wu, 2023; Fang et al., 2023b; Katinskaia and Yangarber, 2024). Several studies attempt to address this challenge from two perspectives. First, Fan et al. (2023) and Zhang et al. (2023) apply supervised fine-tuning to open-source LLMs on GEC datasets to enhance task-specific performance. Second, Davis et al. (2024) and Tang et al. (2024) improve the GEC performance of closed-source models through prompt engineering. Besides, Alirector (Yang and Quan, 2024) attempts to address the overcorrection issue in LLMs by training an alignment model that takes the source sentence and the initially corrected sentence as input to produce the final target sentence. However, these LLM-based methods have yet to achieve the impressive performance in GEC seen in other NLP tasks.

In another line, some research have explored other roles of LLMs in the GEC task, such as generating explanations for corrections (Li et al., 2024; Song et al., 2024) and assessing the quality of grammatical edits (Kobayashi et al., 2024).

2.3 LLM Alignment

LLM alignment techniques can be roughly classified into three categories: supervised fine-tuning (SFT), reinforcement learning from human feedback (RLHF), and offline RLHF (Wang et al., 2024). Although simple, the performance of SFT is limited by the quality of fine-tuning data (Zhou et al., 2023a) and is also vulnerable to out-of-distribution samples (Kirk et al., 2024). Online RLHF methods face challenges like training instability and high resource demands (Yuan et al., 2023; Ethayarajh et al., 2024). Direct Preference Optimization (DPO) (Rafailov et al., 2023), a key offline RLHF method, eliminates the need for a reward model and avoids the complexities of RLHF.

Despite DPO achieving excellent results on various chat benchmarks (Tunstall et al., 2023; Dubey et al., 2024), our experiments show that it performs poorly on the GEC task. In contrast, our Edit-wise Preference Optimization (EPO) approach, designed specifically for GEC, focuses on edit-specific tokens and has shown promising results.

3 Methodology

As illustrated in Figure 2, our proposed framework consists of three main phases: supervised fine-tuning (SFT), preference pair sampling, and **E**dit-wise **P**reference **O**ptimization (EPO). In this section, we first introduce the preliminaries of DPO (§3.1). Then, we provide a detailed explanation of the proposed EPO (§3.2). Finally, we describe the complete training pipeline (§3.3).

3.1 Preliminaries

Direct Preference Optimization (DPO) (Rafailov et al., 2023) seeks to address the instability and complexity of RLHF by directly utilizing pairwise preference data for model optimization. Given a set of source sentences $\{x^i\}_{i=1}^N$ which may contain grammatical errors, along with corresponding correct (positive) sentences $\{y_w^i\}_{i=1}^N$ and incorrect (negative) sentences $\{y_l^i\}_{i=1}^N$, we construct a preference dataset denoted as $\mathcal{D} = \{x^i, y_w^i, y_l^i\}_{i=1}^N$. The objective of DPO is to maximize the reward gap between positive and negative samples:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\log\sigma\Big(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\Big)\Big], \quad (1)$$

where π_θ denotes the policy model to be optimized, π_{ref} is the reference model, σ is the sigmoid function, and β is a hyperparameter. The log-likelihood $\log\pi(y|x)$ of a sentence y given x is computed as:

$$\log\pi(y|x) = \sum_{k=1}^{K}\log\pi\left(y_k|y_{<k}, x\right). \quad (2)$$

3.2 Edit-wise Preference Optimization

As shown in Equation 2, DPO assumes all tokens contribute equally to preference optimization by assigning each a uniform reward weight of 1. However, the differences between preference pairs in GEC are often subtle, making it difficult for the model to capture these nuances. To address this issue, our EPO introduces a dynamic token weighting mechanism, which assigns varying weights based on the importance of each token. EPO helps the model capture subtle differences during preference optimization by assigning higher reward weights to tokens involved in necessary edits.

Specifically, we first employ a GEC parsing tool (Bryant et al., 2017) to identify the differences between positive and negative samples, referred to as *edit spans*. To encourage the model to focus more on these edit spans during preference optimization, we assign a higher reward weight γ ($\gamma > 1$) to

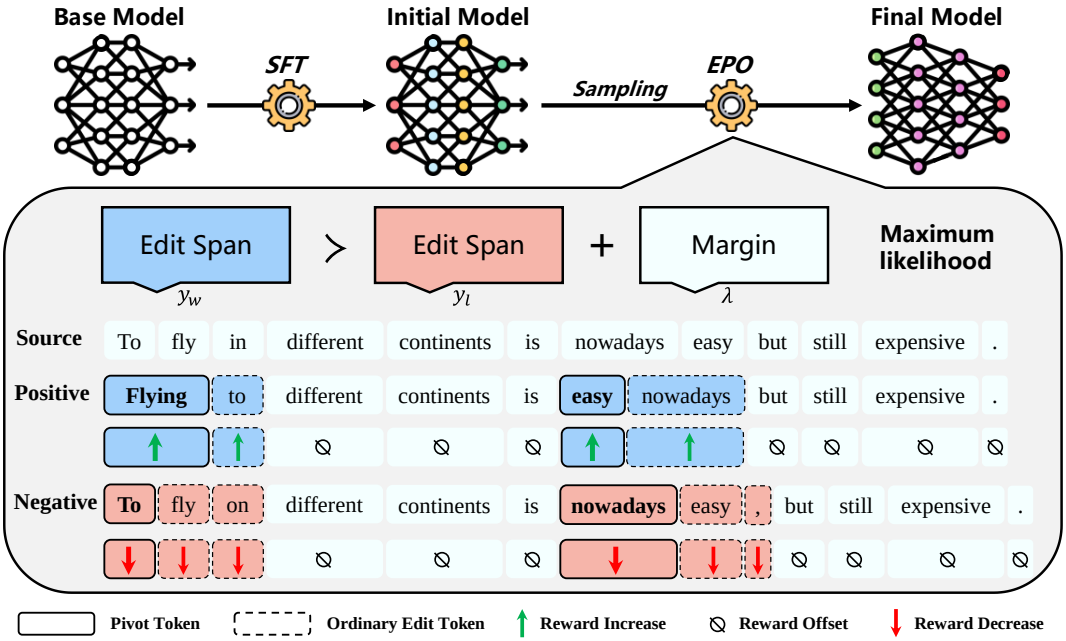

Figure 2: An overview of our proposed framework, which consists of three main phases: (1) supervised fine-tuning (SFT) to train an initial GEC model, (2) sampling from the GEC model to construct preference pairs, and (3) optimizing the GEC model on these preference pairs using the EPO method.

tokens within the edit spans, while setting the reward weights of other tokens to 1. Furthermore, to mitigate the risk of error accumulation during decoding, we treat the first token of each edit span, called the *pivot token*, as crucial for guiding the correction of subsequent tokens. To emphasize its importance, we set its reward weight to α ($\alpha > \gamma$).

By incorporating the above dynamic weight into Equation 2, the log-likelihood of the sentence y with edit information can be reformulated as:

$$\log \pi^e(y|x) = \sum_{k=1}^{K} w_k \log \pi\left(y_k|y_{<k}, x\right). \quad (3)$$

Here, w_k represents the reward weight of the k-th token and is defined as:

$$w_k = \begin{cases} \alpha, & \text{if } y_k \text{ is a pivot token,} \\ \gamma, & \text{if } y_k \text{ is an ordinary edit token,} \\ 1, & \text{otherwise.} \end{cases} \quad (4)$$

To further enhance the model's generalization and robustness, we introduce a margin λ to ensure the reward of the positive sample exceeds that of the negative sample by at least λ, similar to SimPO (Meng et al., 2024). Combining Equations 1 and 3, the EPO objective is defined as:

$$\mathcal{L}_{\text{EPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\log \sigma\Big(\beta \log \frac{\pi^e_\theta(y_w|x)}{\pi^e_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi^e_\theta(y_l|x)}{\pi^e_{\text{ref}}(y_l|x)} - \lambda\Big)\Big]. \quad (5)$$

3.3 Training Pipeline

In this subsection, we provide a detailed introduction to the EPO training pipeline. The training process is also described in Algorithm 1.

SFT Given a base model $\mathcal{M}$ and a training dataset $\hat{\mathcal{D}} = \{x^{(i)}, y^{(i)}\}_{i=1}^{M}$, we first train an initial GEC model $\mathcal{M}_{\text{SFT}}$ as follows:

$$\mathcal{L}_{\text{SFT}}(\pi_\theta) = -\mathbb{E}_{(x,y)\sim\hat{\mathcal{D}}} \log \pi_\theta(y|x), \quad (6)$$

where θ denotes the trainable parameters in $\mathcal{M}$. Through SFT, the model acquires basic grammatical error correction capabilities, laying the foundation for subsequent sampling and optimization.

Preference Pair Construction We construct the preference pairs as follows: For each data pair (x, y) from the training dataset $\hat{\mathcal{D}}$, where x is the source sentence and y is the target, we treat y as the positive sample y_w. For the negative sample y_l, we sample k predicted sentences from $\mathcal{M}_{\text{SFT}}$ using beam search with x as input and choose the one with the largest edit distance from y_w.

This strategy maximizes the differences between positive and negative samples, which facilitates preference optimization. More importantly, it does not require additional human annotations or external reward models, making it cost-effective and high-quality. Alternative preference pair construction strategies are discussed in Section 5.2.

Algorithm 1 Training pipeline

Input: Base model $\mathcal{M}$, training dataset $\hat{\mathcal{D}}$, number of samples per sentence k.
Output: Final model $\mathcal{M}^*$, preference dataset $\mathcal{D}$.

Fine-tune $\mathcal{M}$ on dataset $\hat{\mathcal{D}}$ according to Eq. 6 to obtain the initial GEC model $\mathcal{M}_{\text{SFT}}$;
for (x, y) in $\hat{\mathcal{D}}$ **do**
  Sample k predictions $\mathcal{P} = \{p_i\}_{i=1}^{k}$ from $\mathcal{M}_{\text{SFT}}$ with x as input;
  Initialize $m \leftarrow -\infty, neg \leftarrow \emptyset, pos \leftarrow y$;
  for $i = 1$ to k **do**
    Compute distance d between p_i and pos;
    if $d > m$ **then**
      $m \leftarrow d, neg \leftarrow p_i$;
    end if
  end for
  Match pos and neg using GEC parsing tools to get edit information;
  Add (x, pos, neg) to the dataset $\mathcal{D}$;
end for
Run EPO to update $\mathcal{M}_{\text{SFT}}$ on dataset $\mathcal{D}$ according to Eq. 5 to obtain the final model $\mathcal{M}^*$.

EPO Training After obtaining the preference pairs, we employ a GEC parsing tool, ERRANT[2] (Bryant et al., 2017), to extract edit spans and pivot tokens. We then apply EPO to optimize the SFT model using these preference pairs, as described in Section 3.2, to obtain the final GEC model.

4 Experiments

4.1 Datasets and Evaluation

We conduct experiments on both English and Chinese GEC datasets. For English, following Omelianchuk et al. (2020) and Zhang et al. (2022b), we use three training datasets: the FCE train set (Yannakoudakis et al., 2011), the NUCLE dataset (Dahlmeier et al., 2013), and the W&I+LOCNESS train set (Bryant et al., 2019). The BEA dev set is used for validation. For evaluation, we assess the model on the CoNLL-14 test set (Ng et al., 2014) using the M^2 Scorer (Dahlmeier and Ng, 2012), and on the BEA-19 test set (Bryant et al., 2019) with the ERRANT scorer (Bryant et al., 2017).

For the Chinese GEC experiments, we compile the training data from the Chinese Lang8 dataset (Zhao et al., 2018), the HSK dataset (Zhang, 2009), and the FCGEC train set (Xu et al., 2022), following previous work (Yang and Quan, 2024). The FCGEC dev set is used for validation, while performance is evaluated on both the FCGEC test set (Xu et al., 2022) and the NaCGEC test set (Ma et al., 2022) using the ChERRANT scorer (Zhang et al., 2022a). We report precision, recall, and $F_{0.5}$ scores for all experiments. Further details regarding these datasets are provided in Appendix A.

[2]For Chinese data, we use the ChERRANT tool from https://github.com/HillZhang1999/MuCGEC/tree/main/scorers/ChERRANT

4.2 Base Models and Baselines

Base Models For the English GEC task, we use LLaMA2 (Touvron et al., 2023) and Mistral-v0.1 (Jiang et al., 2023) as the base models, while for the Chinese GEC task, Baichuan2 (Yang et al., 2023) and Qwen2 (Yang et al., 2024) are selected. Due to resource limitations, we opt for 7B-sized models and apply LoRA fine-tuning, which updates only a small subset of the parameters. Further experimental details can be found in Appendix B.

Baselines We compare our approach with the following baselines. **Supervised Fine-tuning (SFT)** means directly fine-tuning the base models on the training data. We also implement **Direct Preference Optimization (DPO)** (Rafailov et al., 2023) using the SFT model on our constructed preference dataset, as EPO is an evolution of DPO. **Alirector** (Yang and Quan, 2024) trains an alignment model to tackle the overcorrection issue in LLMs. Since their experiments are limited to Chinese datasets, we reproduce their results on the English datasets.

Besides, we also employ the following traditional GEC baselines. GECToR (Omelianchuk et al., 2020) is a representative model of the Seq2Edit methods, while BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) are SOTA backbones of Seq2Seq GEC methods. The results of T5 and BART on the English dataset are cited from Rothe et al. (2021) and Zhang et al. (2022b), respectively. SynGEC (Zhang et al., 2022b) incorporates syntactic information into the BART model. We reproduce the results for BART and SynGEC on the Chinese datasets using our configuration. TemplateGEC (Li et al., 2023) uses detection signals from Seq2Edit models as supplementary input for Seq2Seq models, while DeCoGLM (Li and Wang, 2024) combines detection and correction tasks within a single GLM model (Du et al., 2022).

4.3 Main Results

The main results for English GEC are presented in Table 2. Our method consistently outperforms all

Method	Parameters	Data Size	CoNLL-14-test P	CoNLL-14-test R	CoNLL-14-test $F_{0.5}$	BEA-19-test P	BEA-19-test R	BEA-19-test $F_{0.5}$
Traditional GEC Baselines								
GECToR	110M	10.1M	**77.5**	40.1	65.3	**79.2**	53.9	72.4
T5-XL	3B	2.4M	-	-	67.75	-	-	73.92
T5-XXL	11B	2.4M	-	-	**68.87**	-	-	**75.88**
BART	400M	2.5M	73.6	48.6	66.7	74.0	64.9	72.0
SynGEC	110M+400M	2.5M	74.7	49.0	67.6	75.1	65.5	72.9
TemplateGEC	125M+770M	2.4M	74.8	50.0	68.1	76.8	64.8	74.1
DeCoGLM	335M	186.3M	75.1	49.4	68.0	77.4	64.6	74.4
DeGLM-CoGLM	335M+10B	0.1M	70.6	**52.7**	66.1	72.8	**67.6**	71.7
LLaMA2-7B								
+SFT†	7B	0.1M	73.86	50.61	67.64	73.53	67.60	72.26
+Alirector†	7B	0.1M	73.06	**53.03**	67.93	74.88	68.15	73.43
+DPO†	7B	0.1M	73.45	50.80	67.44	73.73	**68.19**	72.55
+EPO (ours)	7B	0.1M	**75.63**	50.94	**68.95**	**77.35**	65.87	**74.75**
Mistral-7B-v0.1								
+SFT†	7B	0.1M	74.10	53.69	68.86	74.55	69.01	73.37
+Alirector†	7B	0.1M	75.20	53.93	69.70	76.20	68.47	74.52
+DPO†	7B	0.1M	74.50	**54.04**	69.26	74.47	**69.97**	73.53
+EPO (ours)	7B	0.1M	**76.71**	52.56	**70.26**	**78.16**	68.07	**75.91**

Table 2: Results on English GEC Benchmarks. Results marked with † indicate those implemented by us; other results are taken from the original papers. The best results are highlighted in bold. Note that ensemble-based methods were excluded to ensure a fair comparison, as our approach involves only a single model.

Method	NaCGEC-test P	NaCGEC-test R	NaCGEC-test $F_{0.5}$	FCGEC-test P	FCGEC-test R	FCGEC-test $F_{0.5}$
Traditional GEC Baselines						
BART†	62.04	45.84	57.94	63.07	**39.95**	56.53
SynGEC†	62.42	**47.41**	58.71	63.75	39.78	56.89
DeCoGLM	-	-	-	55.75	37.91	50.96
DeGLM-CoGLM	-	-	-	56.09	38.02	51.22
BART-Alirector	**68.11**	43.87	**61.33**	**69.44**	36.60	**58.88**
Baichuan2-7B						
+SFT†	63.65	47.73	59.67	61.97	37.25	54.71
+Alirector	66.04	45.91	60.71	64.49	36.22	55.78
+DPO†	63.21	48.29	59.53	58.54	39.21	53.29
+EPO (ours)	**66.94**	**48.37**	**62.16**	**65.19**	**39.49**	**57.68**
Qwen2-7B						
+SFT†	64.27	49.07	60.52	62.18	42.70	56.98
+Alirector†	66.93	46.59	61.55	65.76	39.52	58.05
+DPO†	64.89	**50.15**	61.28	63.53	**42.89**	57.95
+EPO (ours)	**67.09**	49.97	**62.79**	**66.67**	41.93	**59.63**

Table 3: Results on Chinese GEC benchmarks. Results marked with † are implemented by us.

the baselines in $F_{0.5}$ score across all benchmarks, demonstrating its effectiveness and superior performance. For instance, when using Mistral-7B-v0.1 as the backbone, our EPO method improves the $F_{0.5}$ score by an average of 2.0 points across two benchmarks compared to SFT, reaching state-of-the-art performance. In contrast, DPO achieves comparable or even worse performance than SFT, which underscores the limitations of DPO in the GEC task and demonstrates the effectiveness of our approach. Besides, our method also shows superior performance compared to traditional GEC models. Nonetheless, LLaMA2-7B with EPO still lags behind T5-XXL on the BEA test set. This lag may be due not only to T5-XXL's larger parameter size and extensive training data but also to its pre-training task being better aligned with GEC.

Moreover, our results reveal several interesting observations. First, compared to most traditional GEC methods that update all parameters and rely on large amounts of training data, our EPO achieves superior performance by updating only a small subset of parameters[3] and using a smaller-scale training set. This underscores the significant potential of LLMs in the GEC task. Second, the improvement of EPO is primarily driven by an increase in precision, with some recall values showing a decline. This is encouraged in GEC since ignoring an error is less detrimental than proposing an incorrect correction (Ng et al., 2014). In other words, our EPO method can effectively mitigate the overcorrection issue of LLMs when applied to the GEC task. Third, although the performance varies across different base LLMs, the improvements achieved with EPO remain consistently similar. This suggests that more powerful LLMs could be enhanced with EPO in the GEC task.

[3]In our experiments, the LoRA rank is 32 and it updates approximately 1.1% of the parameters (around 80M for a 7B model).

Method	CoNLL-14-test P	R	$F_{0.5}$	BEA-19-test P	R	$F_{0.5}$
LLaMA2-7B						
EPO	**75.63**	50.94	**68.95**	**77.35**	65.87	**74.75**
w/o PTE	75.25	50.13	68.39	77.13	65.17	74.40
w/o PTE&ESW	73.47	**52.10**	67.90	74.39	68.13	73.05
w/o MT	74.23	51.89	68.34	76.61	66.02	74.23
w/o PTE&ESW&MT	73.45	50.80	67.44	73.73	**68.19**	72.55
Mistral-7B-v0.1						
EPO	**76.71**	52.56	**70.26**	**78.16**	68.07	**75.91**
w/o PTE	76.54	51.36	69.70	77.91	66.85	75.41
w/o PTE&ESW	74.42	**54.54**	69.36	75.23	69.51	74.01
w/o MT	75.68	54.46	70.21	76.55	68.73	74.84
w/o PTE&ESW&MT	74.50	54.04	69.26	74.47	**69.97**	73.53

Table 4: Results of ablation study, where "PTE/ESW/MT" are short for pivot token emphasis/edit span weighting/margin term, respectively.

The main results for Chinese GEC are presented in Table 3, from which we can draw conclusions similar to those for English GEC. Our EPO method outperforms most baselines in terms of the $F_{0.5}$ metric, indicating its effectiveness across different languages. However, Baichuan2-7B with EPO underperforms compared to the BART-based Alirector model on the FCGEC test set, suggesting that LLMs for GEC still require further improvement.

5 Analysis

5.1 Ablation Study

Our EPO method comprises three key components: (1) edit span weighting, which assigns a higher reward weight γ ($\gamma > 1$) to tokens within the edit spans, (2) pivot token emphasis, which assigns an even higher reward weight α ($\alpha > \gamma$) to the first token in the edit span, and (3) reward margin λ, which is designed to widen the reward gap between preference pairs. To assess the effectiveness of these components, we remove them one by one and analyze the resulting performance.

As shown in Table 4, removing the entire edit span weighting module (i.e., w/o PTE&ESW) by setting $\alpha = \gamma = 1$ leads to a significant performance drop, while removing pivot token emphasis (i.e. $\alpha = \gamma$) or the reward margin (i.e., $\lambda = 0$) results in moderate performance degradation. These results not only validate the motivation behind our approach that the model should capture and focus on the subtle difference between sample pairs during GEC preference optimization, but also demonstrate the effectiveness of each component of EPO.

5.2 Impact of Pairwise Sample Selection

In our primary experiments, we use the target sentence (*tgt*) as the positive sample and select the candidate sentence with the largest edit distance from the target sentence (*max*) as the negative sample. We employ the LLaMA2-7B model to explore how variations in the pairwise sample selection strategy affect model's performance. For clarity, we define the candidate sentence with the smallest edit distance from the target sentence as *min*, the original erroneous sentence as *src*, and a randomly selected sentence from the sampling results as *rand*.

Method	CoNLL-14-test P	R	$F_{0.5}$	BEA-19-test P	R	$F_{0.5}$
EPO	**75.63**	50.94	**68.95**	**77.35**	65.87	**74.75**
(*min*, *max*)	75.58	49.80	68.49	77.00	65.05	74.27
(*tgt*, *rand*)	73.56	51.44	67.74	75.77	66.65	73.75
(*tgt*, *src*)	41.83	61.77	44.71	33.77	69.85	37.67
(*min*, *src*)	47.48	**64.25**	50.10	39.82	**70.71**	43.64

Table 5: Results of pairwise sample selection, where the first element in parentheses indicates positive samples and the second element denotes negative samples.

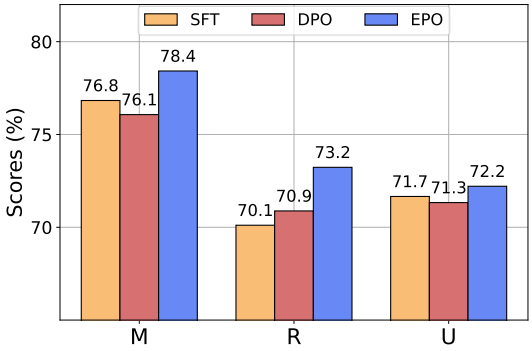

Figure 3: $F_{0.5}$ scores of SFT, DPO, and EPO on Bea-19 test set for different error categories: *missing* (M), *replacement* (R), and *unnecessary* (U).

As shown in Table 5, any deviation from our current setup leads to a decline in performance. Specifically, substituting *min* as the positive sample or *rand* as the negative sample results in a modest decrease in model performance. Moreover, using *src* as the negative sample causes a significant drop in precision, although recall increases. This suggests that such a sample selection approach may induce the model to blindly correct input sentences, leading to severe overcorrection issues.

5.3 Model Robustness

Error Robustness To verify the robustness of our method across different grammatical error types, we use LLaMA2-7B as the backbone and present the fine-grained results across three error categories: *missing* (M), *replacement* (R), and *unnecessary* (U). As shown in Figure 3, EPO consistently improves the overall $F_{0.5}$ scores across all error types compared to both SFT and DPO, with no-

Type	#Overcorrections/#Undercorrections		
	SFT	**DPO**	**EPO**
M	363/446	421/421	322/418
R	929/1269	880/1272	672/1312
U	149/203	149/206	117/215
All	1441/1918	1450/1899	1111/1945

Table 6: The number of overcorrections and undercorrections for different error types on BEA-19 test set.

	CoNLL-14-test			BEA-19-test		
Objective	**P**	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**
SFT	73.86	50.61	67.64	73.53	67.60	72.26
IPO	75.29	48.03	67.62	73.52	**68.28**	72.41
w/ PTE&ESW	75.37	49.91	68.39	77.29	65.47	74.59
KTO	75.26	48.36	67.72	74.80	68.62	73.48
w/ PTE&ESW	75.47	50.15	68.55	76.82	66.30	74.46
SimPO	74.72	49.06	67.65	73.81	68.15	72.60
w/ PTE&ESW	75.52	50.49	68.71	**77.53**	65.30	74.73
EPO	**75.63**	**50.94**	**68.95**	77.35	65.87	**74.75**

Table 7: Results of EPO on different preference optimization objectives using LLaMA2-7B.

table gains in the *missing* and *replacement* categories, highlighting EPO's robustness.

Overcorrection Mitigation As mentioned in previous sections, LLMs tend to exhibit overcorrection when applied to the GEC task. To investigate whether our method can alleviate this issue, we record the performance of LLaMA2-7B on different error types in the BEA-19 test set. As shown in Table 6, our method results in fewer overcorrections across all error categories compared to both SFT and DPO, while displaying a modest increase in undercorrections. These results suggest that the EPO method effectively mitigates the overcorrection problem in LLMs, confirming the effectiveness of edit-wise preference optimization for this task.

5.4 Variants of Preference Optimization

Theoretically, EPO can be applied to any contrastive preference optimization method. Therefore, to demonstrate the extensibility of our approach, we conduct additional experiments by applying the edit span weighting and pivot token emphasis modules to the following DPO variants: IPO (Gheshlaghi Azar et al., 2024), KTO (Ethayarajh et al., 2024), and SimPO (Meng et al., 2024)[4]. As shown in Table 7, our method consistently yields substantial performance improvements across all three objectives, demonstrating its effectiveness across various DPO variants.

[4]See Appendix C for a brief introduction to the different DPO variants.

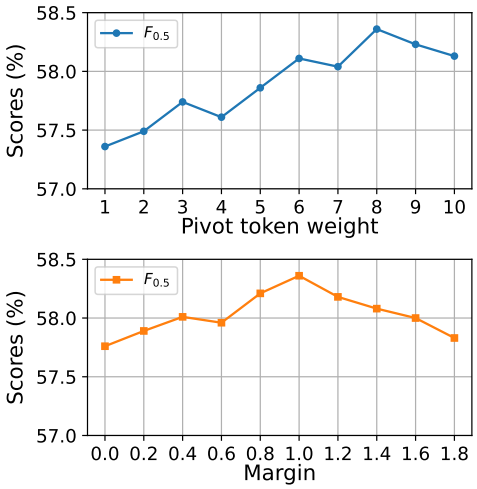

Figure 4: Results of our method on Bea-19 dev set with different values of pivot token weight and margin.

5.5 Impact of Reward Weights and Margin

The training objective of EPO involves three key hyperparameters: γ, which controls the reward weights of edit tokens; α, which controls the reward weights of pivot tokens; and λ, which represents the reward margin. To investigate their impact on model performance, we use LLaMA2-7B as the backbone and present the results of different values of α and λ on the Bea-19 dev set in Figure 4, where we change one while fixing the other[5]. As shown in the first subfigure, the $F_{0.5}$ score generally increases as the pivot token weight α rises, peaking at approximately 8. The second subfigure shows that as the margin λ increases, the $F_{0.5}$ score initially rises and then declines, reaching its peak at $\lambda = 1$. These results indicate that for LLaMA2-7B, the optimal hyperparameter configuration is $\alpha = 8$, $\gamma = 4$, and $\lambda = 1$. The optimal configurations for other models are provided in Table 10.

6 Conclusion

In this paper, we propose Edit-wise Preference Optimization (EPO), a method specifically designed to enhance the grammatical error correction (GEC) capabilities of large language models (LLMs). Unlike Direct Preference Optimization (DPO), which treats all tokens equally, EPO focuses on improving the reward gap of edit tokens between positive and negative samples to better capture their subtle differences. Moreover, to mitigate the risk of error accumulation in autoregressive models including LLMs, we assign higher reward weights to pivot tokens. We also incorporate a margin term in the

[5]For simplicity, we set γ to $\alpha/2$ as α varies.

training objective to improve generalization. Finally, we develop a simple yet effective method for constructing preference datasets. Experimental results on widely-used English and Chinese benchmarks show that EPO not only outperforms traditional GEC models but also surpasses standard LLM alignment techniques. Extensive analysis confirms that the strategies for implementing our token-wise preference optimization are critical.

Limitations

While EPO demonstrates promising results, it has certain limitations. First, our experiments are conducted using LoRA fine-tuning on 7B-scale models. Due to computational resource constraints, we did not explore larger-scale LLMs or full-parameter fine-tuning, which might yield better performance. Second, our method may require additional effort to tune four hyperparameters: β, α, γ and λ. Lastly, we focus on enhancing the grammatical error correction capabilities of LLMs, which could potentially lead to a decline in their general abilities.

Ethics Statement

Our work aims to propose a technical method to enhance the grammar error correction capabilities of LLMs, which does not raise ethical concerns. All datasets and models used in this work are publicly available, and we strictly adhere to their usage guidelines. We are committed to conducting our research in an ethical and responsible manner.

Acknowledgements

We appreciate the anonymous reviewers for their valuable comments. This work was supported by the National Natural Science Foundation of China (No. 62176270) and the Guangdong Basic and Applied Basic Research Foundation (No. 2023A1515012832).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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.

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2023. Grammatical error correction: A survey of the state of the art. *Computational Linguistics*, pages 643–701.

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. *arXiv 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.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner English: The NUS corpus of learner English. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.

Christopher Davis, Andrew Caines, O 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 and virtual meeting. Association for Computational Linguistics.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, Dublin, Ireland. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Model alignment as prospect theoretic optimization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 12634–12651. PMLR.

Yaxin Fan, Feng Jiang, Peifeng Li, and Haizhou Li. 2023. Grammargpt: Exploring open-source llms for native chinese grammatical error correction with supervised fine-tuning. *Preprint*, arXiv:2307.13923.

Tao Fang, Jinpeng Hu, Derek F. Wong, Xiang Wan, Lidia S. Chao, and Tsung-Hui Chang. 2023a. Improving grammatical error correction with multimodal feature integration. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9328–9344, Toronto, Canada. Association for Computational Linguistics.

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

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 4447–4455. PMLR.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

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.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2024. Understanding the effects of rlhf on llm generalisation and diversity. In *Proceedings of the Twelfth International Conference on Learning Representations*.

K.M. Knill, M.J.F. Gales, P.P. Manakul, and A.P. Caines. 2019. Automatic grammatical error detection of non-native spoken learner english. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8127–8131.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jiquan Li, Junliang Guo, Yongxin Zhu, Xin Sheng, Deqiang Jiang, Bo Ren, and Linli Xu. 2022. Sequence-to-action: Grammatical error correction with action guided sequence generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10974–10982.

Wei Li and Houfeng Wang. 2024. Detection-correction structure via general language model for grammatical error correction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1748–1763, Bangkok, Thailand. Association for Computational Linguistics.

Yinghao Li, Xuebo Liu, Shuo Wang, Peiyuan Gong, Derek F. Wong, Yang Gao, Heyan Huang, and Min Zhang. 2023. TemplateGEC: Improving grammatical error correction with detection template. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6878–6892, Toronto, Canada. Association for Computational Linguistics.

Yinghui Li, Shang Qin, Haojing Huang, Yangning Li, Libo Qin, Xuming Hu, Wenhao Jiang, Hai-Tao Zheng, and Philip S Yu. 2024. Rethinking the roles of large language models in chinese grammatical error correction. *arXiv preprint arXiv:2402.11420*.

Junwei Liao, Sefik Eskimez, Liyang Lu, Yu Shi, Ming Gong, Linjun Shou, Hong Qu, and Michael Zeng. 2023. Improving readability for automatic speech recognition transcription. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(5).

Shirong Ma, Yinghui Li, Rongyi Sun, Qingyu Zhou, Shulin Huang, Ding Zhang, Li Yangning, Ruiyang Liu, Zhongli Li, Yunbo Cao, Haitao Zheng, and Ying

Shen. 2022. Linguistic rules-based corpus generation for native Chinese grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 576–589, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*.

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.

Fanyi Qu and Yunfang Wu. 2023. Evaluating the capability of large-scale language models on chinese grammatical error correction task. *arXiv preprint arXiv:2307.03972*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

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.

Yixiao Song, Kalpesh Krishna, Rajesh Bhatt, Kevin Gimpel, and Mohit Iyyer. 2024. GEE! grammar error explanation with large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 754–781, Mexico City, Mexico. 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.

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.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Amos Tversky and Daniel Kahneman. 1992. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and Uncertainty*, 5:297–323.

Zhichao Wang, Bin Bi, Shiva Kumar Pentyala, Kiran Ramnath, Sougata Chaudhuri, Shubham Mehrotra, Xiang-Bo Mao, Sitaram Asur, et al. 2024. A comprehensive survey of llm alignment techniques: Rlhf, rlaif, ppo, dpo and more. *arXiv preprint arXiv:2407.16216*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Lvxiaowei Xu, Jianwang Wu, Jiawei Peng, Jiayu Fu, and Ming Cai. 2022. FCGEC: Fine-grained corpus for Chinese grammatical error correction. In *Findings of the Association for Computational Linguistics:*

EMNLP 2022, pages 1900–1918, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Konstantin Yakovlev, Alexander Podolskiy, Andrey Bout, Sergey Nikolenko, and Irina Piontkovskaya. 2023. GEC-DePenD: Non-autoregressive grammatical error correction with decoupled permutation and decoding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1546–1558, Toronto, Canada. Association for Computational Linguistics.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

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 and virtual meeting. Association for Computational Linguistics.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

Dezhi Ye, Bowen Tian, Jiabin Fan, Jie Liu, Tianhua Zhou, Xiang Chen, Mingming Li, and Jin Ma. 2023a. Improving query correction using pre-train language model in search engines. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 2999–3008, New York, NY, USA. Association for Computing Machinery.

Jingheng Ye, Yinghui Li, Yangning Li, and Hai-Tao Zheng. 2023b. MixEdit: Revisiting data augmentation and beyond for grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10161–10175, Singapore. Association for Computational Linguistics.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback. In *Advances in Neural Information Processing Systems*, volume 36, pages 10935–10950. Curran Associates, Inc.

Baolin Zhang. 2009. Features and functions of the hsk dynamic composition corpus. *International Chinese Language Education*, 4:71–79.

Yue Zhang, Leyang Cui, Deng Cai, Xinting Huang, Tao Fang, and Wei Bi. 2023. Multi-task instruction tuning of llama for specific scenarios: A preliminary study on writing assistance. *arXiv preprint arXiv:2305.13225*.

Yue Zhang, Zhenghua Li, Zuyi Bao, Jiacheng Li, Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022a. MuCGEC: a multi-reference multi-source evaluation dataset for Chinese grammatical error correction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3118–3130, Seattle, United States. Association for Computational Linguistics.

Yue Zhang, Bo Zhang, Zhenghua Li, Zuyi Bao, Chen Li, and Min Zhang. 2022b. 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.

Yuanyuan Zhao, Nan Jiang, Weiwei Sun, and Xiaojun Wan. 2018. Overview of the nlpcc 2018 shared task: Grammatical error correction. In *Natural Language Processing and Chinese Computing*, pages 439–445, Cham. Springer International Publishing.

Zewei Zhao and Houfeng Wang. 2020. Maskgec: Improving neural grammatical error correction via dynamic masking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):1226–1233.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023a. Lima: Less is more for alignment. In *Advances in Neural Information Processing Systems*, volume 36, pages 55006–55021. Curran Associates, Inc.

Houquan Zhou, Yumeng Liu, Zhenghua Li, Min Zhang, Bo Zhang, Chen Li, Ji Zhang, and Fei Huang. 2023b. 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 Datasets

The statistics of the datasets used in our English and Chinese GEC experiments are listed in Table 8. Each dataset consists of a parallel corpus containing pairs of erroneous and corrected sentences.

Dataset	Language	#Sentences	%Error	Usage
FCE-train	English	28350	62.5	SFT Stage I
NUCLE	English	57151	37.4	SFT Stage I
W&I+LOCNESS	English	34308	66.3	SFT Stage II & EPO Training
Lang8	Chinese	1,220,906	89.5	SFT Stage I
HSK	Chinese	15,6870	60.8	SFT Stage I
FCGEC	Chinese	36,341	54.3	SFT Stage II & EPO Training
BEA-19-dev	English	4384	64.3	Validation
FCGEC-dev	Chinese	2,000	55.1	Validation
CoNLL-14-test	English	1312	71.9	Testing
BEA-19-test	English	4477	-	Testing
FCGEC-test	Chinese	3,000	-	Testing
NaCGEC-test	Chinese	5,869	95.6	Testing

Table 8: Statistics of the used datasets. #Sentences denotes the number of the sentences and %Error denotes the percentage of the erroneous sentences.

B Experimental Details

B.1 Training Details

Similar to previous works (Zhang et al., 2022b; Yang and Quan, 2024), we divide the SFT process into two stages, with the training data presented in Table 8. For English GEC, we first perform SFT-stage 1 using the FCE+NUCLE dataset, followed by SFT-stage 2 on the smaller but higher-quality W&I+LOCNESS dataset. Finally, we construct the pairwise dataset based on the W&I+LOCNESS dataset and conduct EPO training.

For Chinese GEC, we use the same training set preparation as Zhang et al. (2022a) for SFT-stage 1. Specifically, we discard all error-free samples from the Lang8 and HSK datasets, replicate the HSK dataset five times, and combine it with the Lang8 dataset, resulting in a total of 1,568,885 sentence pairs. For SFT-stage 2 and EPO training, we utilize the FCGEC training set.

B.2 Instruction Templates

Table 9 presents the instruction templates used for English and Chinese GEC during the instruction fine-tuning of LLMs. Each template consists of an input field, providing the source text, and a response field, specifying the target text.

B.3 Implementation Details

Our code implementation is primarily based on the *LLaMA-Factory* project (Zheng et al., 2024) and Huggingface Transformers (Wolf et al., 2020). As in most studies, we do not calculate the loss for the prompt portion. For preference data construction, the number of samplings per training sample is set to $k = 10$. Considering the time and computational resources, we applied LoRA for efficient fine-tuning instead of full-parameter fine-tuning, updating only a small portion of the parameters. We used the Adam optimizer (Kingma and Ba, 2014) with cosine learning rate decay. We searched for the optimal value of α in {2, 4, 6, 8, 10}, γ in {1, 2, 3, 4, 5} and the margin λ in {0.5, 1.0, 1.5, 2.0} on the validation set. The hyperparameter settings are presented in Table 10. All experiments are carried out on 4 GeForce RTX 3090 24GB GPUs.

C Different DPO Variants

Besides DPO, we also apply our method to other DPO variants: IPO (Gheshlaghi Azar et al., 2024), KTO (Ethayarajh et al., 2024), and SimPO (Meng et al., 2024). Identity Preference Optimization (IPO) aims to mitigate overfitting to the preference dataset by regressing the gap between pairwise log-likelihood ratios to $\frac{1}{2\beta}$:

$$\mathcal{L}_{\text{IPO}}(\pi_\theta;\pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big(\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} - \frac{1}{2\beta}\Big)^2. \quad (7)$$

Kahneman-Tversky Optimization (KTO) enhances the DPO method by incorporating Kahneman and Tversky's prospect theory (Tversky and Kahneman, 1992) that losses outweigh equivalent gains:

$$\mathcal{L}_{\text{KTO}}(\pi_\theta,\pi_{\text{ref}}) = \mathbb{E}_{(x,y_w,y_l)\sim D}\Big[\lambda_w\sigma(z_{\text{ref}} - \beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)}) + \lambda_l\sigma(z_{\text{ref}} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)})\Big], \quad (8)$$

where $z_{\text{ref}} = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\beta\text{KL}\left(\pi_\theta(y|x)||\pi_{\text{ref}}(y|x)\right)\right]$.

Simple Preference Optimization (SimPO) replaces the reference policy reward in DPO with a length-normalized reward to reduce the discrepancy between training and inference:

$$\mathcal{L}_{\text{SimPO}}(\pi_\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\log\sigma\Big(\frac{\beta}{|y_w|}\log\pi_\theta(y_w|x) - \frac{\beta}{|y_l|}\log\pi_\theta(y_l|x) - \lambda\Big)\Big]. \quad (9)$$

Language	Instruction Template
English	Rewrite the input text into grammatically correct text. ### input:\n{Source}\n\n ### response:\n{Target}
Chinese	纠正输入句子中的语法错误，并输出正确的句子。### input:\n{Source}\n\n ### response:\n{Target} (Trans.: *Correct grammatical errors in the input sentence and output the correct sentence.*)

Table 9: Instruction templates for English and Chinese GEC, where "Trans." denotes the translation of the instruction.

Hyperparameter	English		Chinese	
Backbone	LLaMA2-7B	Mistral-7B-v0.1	Baichuan2-7B	Qwen2-7B
Batch size (SFT I)	256	256	128	128
Batch size (SFT II)	128	128	64	64
Batch size (EPO)	128	128	64	64
Max Epochs (SFT I)	5	5	3	3
Max Epochs (SFT II)	1	1	1	1
Max Epochs (EPO)	3	3	3	3
Max Length	200	200	200	200
Learning Rate (SFT I)	3×10^{-5}	3×10^{-5}	3×10^{-5}	3×10^{-5}
Learning Rate (SFT II)	3×10^{-5}	3×10^{-5}	3×10^{-4}	3×10^{-4}
Learning Rate (EPO)	5×10^{-7}	5×10^{-7}	5×10^{-7}	5×10^{-7}
Learning Rate Scheduler	Cosine	Cosine	Cosine	Cosine
Optimizer	AdamW	AdamW	AdamW	AdamW
Weight Decay	0.0	0.0	0.0	0.0
Warmup Ratio	0.1	0.1	0.1	0.1
LoRA	target modules = all linears; lora rank = 32; lora alpha = 64			
β	0.5	0.5	0.5	0.5
α	8	10	8	6
γ	4	5	4	3
λ	1	1.5	1	1
Beam Size	10	10	10	10

Table 10: Hyperparameter settings in our experiments.