# Alirector: Alignment-Enhanced Chinese Grammatical Error Corrector

Anonymous ACL submission

## Abstract

Chinese grammatical error correction (CGEC) faces serious overcorrection challenges when employing autoregressive generative models such as sequence-to-sequence (Seq2Seq) models and decoder-only large language models (LLMs). While previous methods aim to address overcorrection in Seq2Seq models, they are difficult to adapt to decoder-only LLMs. In this paper, we propose an alignment-enhanced corrector for the overcorrection problem that applies to both Seq2Seq models and decoder-only LLMs. Our method first trains a correction model to generate an initial correction of the source sentence. Then, we combine the source sentence with the initial correction and feed it through an alignment model for another round of correction, aiming to enforce the alignment model to focus on potential overcorrection. Moreover, to enhance the model's ability to identify nuances, we further explore the reverse alignment of the source sentence and the initial correction. Finally, we transfer the alignment knowledge from two alignment models to the correction model, instructing it on how to avoid overcorrection. Experimental results on three CGEC datasets demonstrate the effectiveness of our approach in alleviating overcorrection and improving overall performance.

## 1 Introduction

Chinese grammatical error correction (CGEC) (Zhao et al., 2018), which aims to identify and correct potential grammatical errors in given Chinese sentences while adhering to the principle of minimal editing, has broad applications in scenarios such as writing assistant and search engine (Wang et al., 2021). Chinese grammatical errors can be basically categorized into component missing, component redundancy, improper collocation, and improper word order (Ma et al., 2022), which are similar to those in English but tend to be more intricate due to the complexities of Chinese grammar.

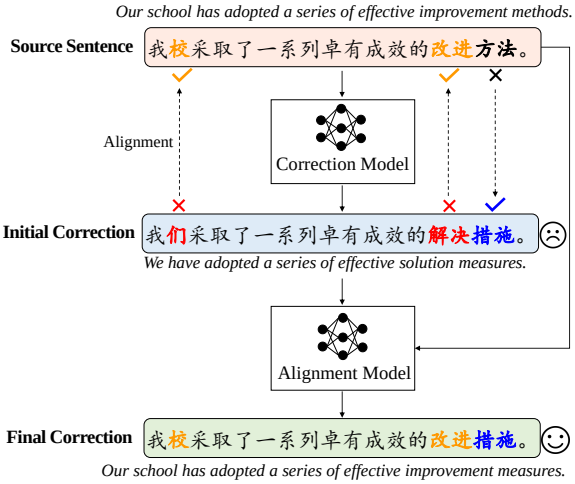


Figure 1: An illustration of addressing overcorrection through alignment of the source sentence and the initial correction. Overcorrected characters and their error-free counterparts are highlighted in red and orange, respectively. Correct edits are highlighted in blue.

Existing CGEC methods can be mainly divided into three categories: sequence-to-edit (Seq2Edit), sequence-to-sequence (Seq2Seq), and decoder-only large language models (LLMs). Seq2Edit methods treat CGEC as a sequence tagging task by predicting token-level edit operations (Liang et al., 2020; Zhang et al., 2022a). While offering fast inference and robust error detection, these methods may compromise text fluency and exhibit weak migration ability due to the reliance on language-specific vocabulary (Li et al., 2022). Seq2Seq methods tackle CGEC using neural machine translation techniques (Fu et al., 2018; Zhao and Wang, 2020) and excel in generating fluent sentences but often lack controllability. More recently, decoder-only LLMs have demonstrated breakthrough performance in various NLP tasks, showing significant potential in CGEC (Fang et al., 2023; Qu and Wu, 2023). However, research suggests that decoder-only LLMs still fall short of surpassing lightweight state-of-the-art models (Zhang et al., 2023).

Besides, Seq2Seq models and decoder-only

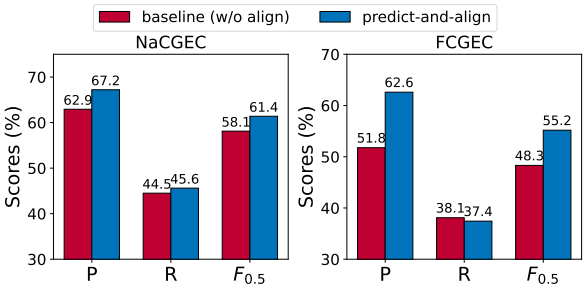


Figure 2: Preliminary results of predict-and-align on NaCGEC (Ma et al., 2022) and FCGEC (Xu et al., 2022) datasets with Baichuan2-7B model (Yang et al., 2023).

LLMs may suffer from severe overcorrection issues, resulting in the modification of error-free characters of the source sentence (Park et al., 2020), as illustrated in Figure 1. This can be attributed to the tendency of these generative models to generate target sequences with higher probabilities and replace low-frequency words with more frequent ones (Li et al., 2022). While increasing the number of training examples empirically alleviates this problem, obtaining high-quality annotated examples remains a challenge. Previous studies have explored mitigating overcorrection in Seq2Seq models. Among them, a copy module can be incorporated to enable the direct copying of correct tokens from source sentences to output sentences (Zhao et al., 2019). Another approach involves integrating error detection results from a Seq2Edit model into a Seq2Seq correction model (Li et al., 2023a). However, these methods prove challenging to migrate to decoder-only LLMs due to differences in their architectures. Given the emerging breakthroughs of LLMs in various NLP tasks, there is an urgent need to explore their potential in CGEC, where the overcorrection problem presents a significant obstacle.

To fill this gap, we first explore a two-stage *predict-and-align* method for mitigating overcorrection caused by Seq2Seq models and decoder-only LLMs. As illustrated in Figure 1, we first train a correction model to generate an initial correction of the source sentence. Then, we combine the source sentence with the initial correction and feed it through an alignment model for another round of correction. The alignment model is tasked not only with copying correct edits in the initial correction but also retaining error-free characters in the source sentence, thereby reducing overcorrections. Preliminary results in Figure 2 show that the two-stage method substantially enhances the overall performance of the original correction model.[1]

[1]More preliminary results are provided in Section 4.3.

The above predict-and-align method requires deploying two models during inference, which is inefficient in terms of both time and storage. Therefore, we propose to enhance the correction model with knowledge acquired from the alignment model, resulting in an **ali**gnment-enhanced cor**rector** (Alirector) better at alleviating the overcorrection problem. Moreover, previous studies (Lu et al., 2022; Qin et al., 2023) have shown that language models are sensitive to the ordering of the input sequence. Hence, we train another alignment model to explore the reverse combination of the source sentence and the initial correction. For knowledge transfer, we apply KL-divergence to constrain the output distributions of the correction model and the two alignment models, guiding the correction model on how to avoid overcorrection. Note that the proposed alignment method applies to both Seq2Seq models and decoder-only LLMs.

Extensive experiments were conducted on three CGEC datasets, and the experimental results demonstrate that our method achieves substantial improvements over baselines and effectively alleviates the overcorrection problem. Among various findings, our in-depth analysis reveals that the alignment information between the source sentence and the initial correction is crucial for mitigating overcorrection and improving the robustness of the correction model. Besides, we confirm that current decoder-only LLMs underperform Seq2Seq models, which warrants further investigation.

## 2 Related Work

### 2.1 Traditional CGEC Methods

Traditional CGEC methods typically follow the approaches used in English GEC, which are broadly categorized into Seq2Edit and Seq2Seq methods.

Seq2Edit methods (Awasthi et al., 2019; Omelianchuk et al., 2020; Liang et al., 2020; Zhang et al., 2022a) treat GEC as a sequence editing task, which predicts token-level edit operations for the input sentence. PIE (Awasthi et al., 2019) utilizes BERT to iteratively predict edit labels. GECToR (Omelianchuk et al., 2020) further extends the tag vocabulary with fine-grained edit tags. Liang et al. (2020) and Zhang et al. (2022a) explore adapting GECToR for CGEC tasks. The strengths of Seq2Edit methods lie in its high inference efficiency and strong error detection performance. However, they rely heavily on manually designed vocabularies and language-specific lexical rules,

limiting their adaptability (Li et al., 2022).

On the other hand, Seq2Seq methods (Zhao et al., 2019; Zhao and Wang, 2020; Kaneko et al., 2020; Zhang et al., 2022b) employ encoder-decoder models inspired by neural machine translation to model the GEC task, where the encoder encodes the source sentence and the decoder sequentially generates the target tokens. While Kaneko et al. (2020) further adapts pre-trained knowledge into the encoder-decoder model, Zhang et al. (2022b) explore incorporating syntax information. Besides, efforts have been made to combine Seq2Edit and Seq2Seq to enhance the inference efficiency (Chen et al., 2020) or improve the correction results (Yuan et al., 2021; Li et al., 2022, 2023a).

### 2.2 LLMs for GEC

With the success of LLMs across various NLP tasks, researchers have explored their potential for CGEC. Recent studies (Fang et al., 2023; Li et al., 2023b; Qu and Wu, 2023; Fan et al., 2023) assess the performance of diverse LLMs, including both closed-source and open-source models, on the CGEC task. Fang et al. (2023) evaluate ChatGPT's performance on CGEC through in-context learning, highlighting its ability to generate fluent sentences but also its susceptibility to overcorrection. Fan et al. (2023) explore open-source LLMs for CGEC via instruction tuning (Ouyang et al., 2022). Zhang et al. (2023) suggest that fine-tuned LLMs still struggle to match the performance of existing state-of-the-art lightweight GEC models. Besides, some research endeavors (Kaneko and Okazaki, 2023; Song et al., 2023) aim at generating explanations for corrections utilizing LLMs' powerful capability. While these studies often overlook the overcorrection issue, our work presents a novel approach capable of mitigating overcorrection in LLMs.

### 2.3 Overcorrection in GEC

Seq2Seq models tend to generate sentences with higher probabilities and replace infrequent words with more frequent ones, leading to overcorrection. Previous works (Zhao et al., 2019; Li et al., 2022, 2023a) expore various approaches to relieve this problem. Zhao et al. (2019) employ a copy module to directly copy the correct tokens from the source sentence to the output sentence. Li et al. (2022) propose a sequence-to-action module based on the seq2seq model to generate a token-level action sequence. Li et al. (2023a) propose a two-stage approach by integrating detection results from a Seq2Edit model into a Seq2Seq correction model. While these methods are challenging when applied to decoder-only LLMs due to architectural differences, the approach proposed in this work applies to both Seq2Seq models and decoder-only LLMs.

## 3 Methodology

As depicted in Figure 3, our **ali**gnment-enhanced cor**rector** (Alirector) for Chinese grammatical error correction (CGEC) comprises three main steps to build. First, we train a correction model to generate an initial correction of the source sentence. Second, we perform bidirectional alignment by combining the source sentence with the initial correction forward and backward respectively, and passing each combination through an alignment model for another round of correction. Third, we employ knowledge distillation to transfer the knowledge from the two alignment models to the correction model. In the following sections, we first formulate the CGEC task and introduce the correction model in Section 3.1. Then, we delve into the details of the alignment models in Section 3.2 and specify the knowledge distillation in Section 3.3.

### 3.1 Correction Model

Given a source sentence $X = \{x_1, x_2, .., x_m\}$ that may contain grammatical errors, the goal of CGEC is to identify and correct the potential grammatical errors within $X$ and output the corresponding gold sentence $Y = \{y_1, y_2, .., y_n\}$. The models we investigate to implement the correction model include Transformer-based (Vaswani et al., 2017) Seq2Seq models and decoder-only LLMs.

**Seq2Seq** The training objective of the Seq2Seq correction model is to minimize the negative log-likelihood (NLL) loss (Williams and Zipser, 1989):

$$\mathcal{L}_{\text{gec}} = \sum_{t=1}^{n} -\log P(y_t | y_{<t}, X; \theta_1), \quad (1)$$

where $y_{<t}$ represents the tokens preceding time step $t$, and $\theta_1$ denotes the trainable parameters.

**Decoder-only LLMs** The input to the decoder-only correction model is formulated by converting $X$ and $Y$ into a natural language sequence $Z$ with an instruction template $\mathcal{T}_{\text{gec}}(X, Y)$:[2]

$$\begin{aligned} Z &= \mathcal{T}_{\text{gec}}(X, Y) \\ &= \{\overbrace{z_1, ..., z_{i-1}}^{instruction}, \overbrace{z_i, ..., z_j}^{X}, \overbrace{z_{j+1}, ..., z_{j+n}}^{Y}\}. \end{aligned} \quad (2)$$

[2]Instruction templates are provided in Appendix A.

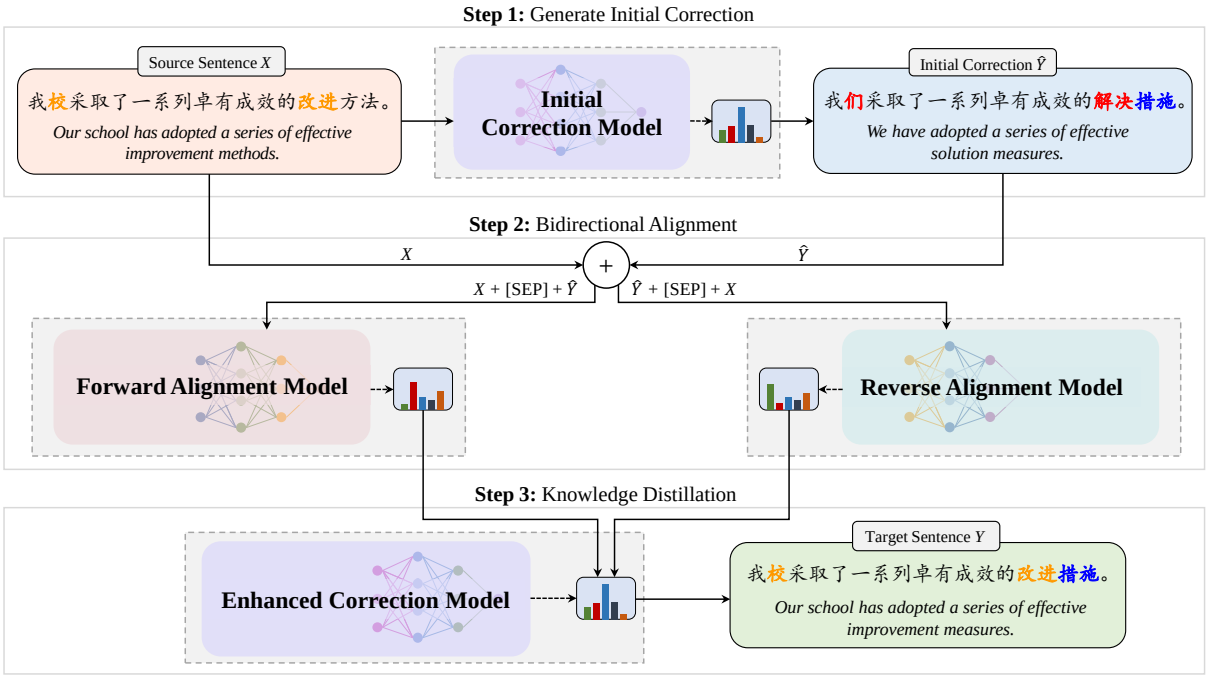


Figure 3: An overview of our proposed framework, which comprises three main steps. First, we train a correction model to generate an initial correction of the source sentence. Second, we perform bidirectional alignment by combining the source sentence with the initial correction forward and backward respectively, and passing each combination through an alignment model for another round of correction. Third, we employ knowledge distillation to transfer the knowledge from the two alignment models to the correction model.

Then, we compute the NLL loss on the target tokens as the training objective:

$$\mathcal{L}_{\text{gec}} = \sum_{t=j+1}^{j+n} -\log P(z_t|z_{<t};\theta_2). \quad (3)$$

### 3.2 Alignment Model

The purpose of our alignment model is to mitigate potential overcorrections in the initial correction generated by the aforementioned correction model. This is achieved by using a separate dataset and training the alignment model to predict the target sentence based on alignment information between the source sentence and the initial correction. Similar to the correction model, both Seq2Seq models and decoder-only LLMs can be employed to build the alignment model. However, to reduce the difficulty of transferring knowledge from the alignment model to the correction model, we require the two stages to share the same architecture.

**Input Construction** We use $\hat{Y}$ to represent the initial output generated by the correction model for the source sentence $X$. Then, we construct the input to the alignment model based on $X$ and $\hat{Y}$ as follows. For Seq2Seq models, we simply concatenate $X$ and $\hat{Y}$ separated by "[SEP]" as the input, denoted as $X_{\text{align}} = X + [\text{SEP}] + \hat{Y}$. As for decoder-only LLMs, we follow Eq. (2) and construct the input by transforming $X$, $\hat{Y}$ and $Y$ into a natural language sequence $W$ using another instruction template $\mathcal{T}_{\text{align}}$:

$$\begin{aligned} W &= \mathcal{T}_{\text{align}}(X, \hat{Y}, Y) \\ &= \{..., \overbrace{w_i, ...}^{X}, \overbrace{w_j, ..., w_k}^{\hat{Y}}, \overbrace{w_{k+1}, ..., w_{k+n}}^{Y}\}. \end{aligned} \quad (4)$$

**Training Objective** The alignment model aims to predict $Y$ based on the alignment of $X$ and $\hat{Y}$. For Seq2Seq models, the training objective is:

$$\mathcal{L}_{\text{align}} = \sum_{t=1}^{n} -\log P(y_t|y_{<t}, X_{\text{align}};\theta_3), \quad (5)$$

where $\theta_3$ denotes the trainable parameters in the alignment model. As for decoder-only LLMs, the NLL loss is computed on the target tokens in $W$:

$$\mathcal{L}_{\text{align}} = \sum_{t=k+1}^{k+n} -\log P(w_t|w_{<t};\theta_4). \quad (6)$$

**Bidirectional Alignment** Previous studies (Lu et al., 2022; Qin et al., 2023) have shown that language models are sensitive to input ordering. Motivated by this, we further introduce bidirectional alignment by incorporating a reverse alignment model, which takes the combined source and initial

correction in reverse order as input. For example, the input to the Seq2Seq-based reverse alignment model is $\hat{Y} + [\text{SEP}] + X$. Intuitively, the reverse alignment model may capture different information compared to the forward alignment model between the source sentence and the initial correction. Our empirical analysis in Section 5.2 also demonstrates that combining these two alignment models helps alleviate the impact of overcorrection and improves the overall robustness of the correction model.

### 3.3 Bidirectional Alignment Distillation

The alignment models described above can be employed alongside the correction model in a two-stage predict-and-align paradigm to mitigate overcorrection. However, this approach presents two potential issues. Firstly, deploying both the correction model and the two alignment models during inference increases both time and storage requirements. Secondly, the correction model and the alignment models are trained separately, overlooking the possibility of mutual enhancement. To address these issues, we propose enhancing the correction model with knowledge distilled from the alignment models, guiding the correction model to avoid overcorrection, as well as eliminating the need for the alignment models during inference.

**Knowledge Distillation** We consider the two alignment models as the teachers and the correction model as the student for knowledge distillation (Hinton et al., 2015). During this process, only the parameters of the correction model are updated, while the parameters of the alignment models remain fixed. For training, we construct inputs for both the correction model and the alignment models following the methods introduced in Section 3.1 and Section 3.2, and obtain the final output logits over the target tokens. Let $z^c$, $z^f$, and $z^r$ denote the output logits from the correction model, the forward alignment model, and the reverse alignment model, respectively. We use KL-divergence as the distillation objective. The forward and reverse alignment distillation losses are defined as:

$$\begin{aligned} \mathcal{L}^f_{\text{kd}} &= \mathcal{D}_{KL}(\sigma(\frac{z^f}{\tau})||\sigma(\frac{z^c}{\tau})) \\ \mathcal{L}^r_{\text{kd}} &= \mathcal{D}_{KL}(\sigma(\frac{z^r}{\tau})||\sigma(\frac{z^c}{\tau})), \end{aligned} \tag{7}$$

where $\tau$ is the temperature, $\sigma$ is the softmax function, and $\mathcal{D}_{KL}(\cdot)$ denotes the KL-divergence.

The overall distillation loss is the weighted sum of these two distillation losses:

$$\mathcal{L}_{\text{kd}} = \alpha\mathcal{L}^f_{\text{kd}} + (1-\alpha)\mathcal{L}^r_{\text{kd}}, \tag{8}$$

where $\alpha \in (0, 1)$ is a hyperparameter.

**Overall Objective** To train the correction model, we formulate the overall objective by combining the GEC loss with the alignment distillation loss:

$$\mathcal{L} = \mathcal{L}_{\text{gec}} + \beta\mathcal{L}_{\text{kd}}, \tag{9}$$

where $\beta$ is a hyperparameter that controls the importance of the distillation loss. More training details are provided in Appendix C.1.

## 4 Experiments

### 4.1 Datasets and Metrics

Based on the data sources, the datasets utilized in our experiments fall into two categories: i) datasets sourced from Chinese-as-a-Second-Language (CSL) learner texts, and ii) datasets sourced from Chinese native speaker texts. For CSL learner data, following previous works (Zhang et al., 2022b; Li et al., 2023a), we employ a combination of the Chinese Lang8 dataset (Zhao et al., 2018) and the HSK dataset (Zhang, 2009) as our training set, MuCGEC-Dev (Zhang et al., 2022a) as the development set, and NLPCC18-Test (Zhao et al., 2018) as the test set. For native speaker data, we first randomly partition 1000 samples from the FCGEC (Xu et al., 2022) training set as the development set, with the remainder used for training. For testing, we utilize FCGEC-Dev and NaCGEC-Test (Ma et al., 2022) as our test sets[3]. Further details regarding the datasets can be found in Appendix B.

For evaluation metrics, we follow previous work and report word-level *precision* (P)/*recall* (R)/*F-measure* ($F_{0.5}$) performance on NLPCC18-Test using the official MaxMatch scorer (Dahlmeier and Ng, 2012) and PKUNLP word segmentation tool provided by Zhao et al. (2018). For FCGEC-Dev and NaCGEC-Test, we report the character-level P/R/$F_{0.5}$ scores using the ChERRANT scorer[4].

### 4.2 Base Models and Baselines

As previously mentioned, the proposed method applies to both Seq2Seq models and decoder-only LLMs. For Seq2Seq, we choose Transformer-large and Chinese BART-large (Shao et al., 2021) as the

[3]We use FCGEC-Dev here since we can not access the gold labels of FCGEC-Test.

[4]https://github.com/HillZhang1999/MuCGEC/tree/main/scorers/ChERRANT

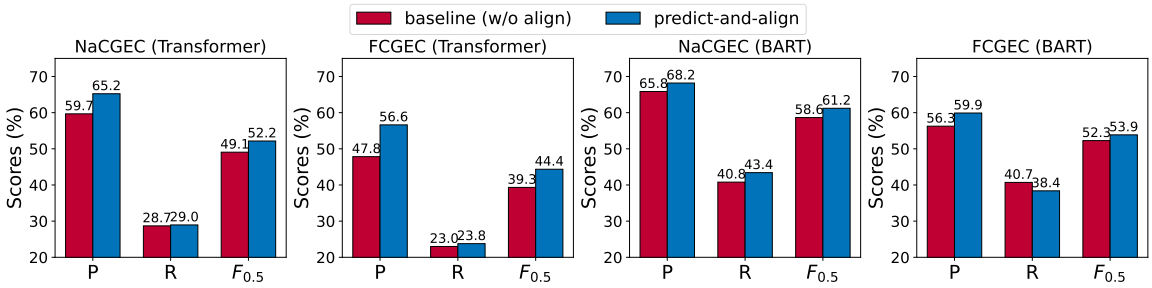


Figure 4: Preliminary results of predict-and-align on NaCGEC and FCGEC datasets with Transformer and BART.

base models. For decoder-only LLMs, we choose Baichuan2-7B (Yang et al., 2023), a powerful Chinese LLM, and Chinese-LLaMA2-7B[5], which is obtained by incremental training of LLaMA2 (Touvron et al., 2023b) with Chinese corpus.

For comparison, we first employ the following Seq2Seq models as baselines. **Vanilla Fine-tuning (FT)** means directly fine-tuning the base models on the entire training set. **TemplateGEC** (Li et al., 2023a) constructs a detection template to integrate the Seq2Edit and Seq2Seq methods. **SynGEC** (Zhang et al., 2022b) incorporates syntax information into Seq2Seq models. **Copy** (Zhao et al., 2019) employs a copy mechanism for Seq2Seq models to directly copy unchanged words from the source sentence to the target sentence. Besides, we also employ decoder-only baselines. Except for vanilla fine-tuning, we implement the copy method (Zhao et al., 2019) in decoder-only LLMs for comparison. The implementation details and hyperparameter settings are presented in Appendix C.2.

### 4.3 Preliminary Results

As mentioned earlier, we conducted preliminary experiments of the predict-and-align method on NaCGEC and FCGEC datasets. In addition to the results shown in Figure 2 using Baichuan2-7B, we also present the results with two Seq2Seq models, namely Transformer-large and BART-large, in Figure 4. From these results, we observe that after alignment, both Baichuan2-7B and Transformer exhibit a substantial performance improvement, especially in precision, revealing the potential of alignment in enhancing overall performance and mitigating overcorrection. While BART's performance improvement may not be as remarkable as Baichuan2, the alignment approach still demonstrates favorable enhancement. More analysis and discussion regarding the potential of the alignment method are presented in Appendix D.

### 4.4 Main Results

The main results are shown in Table 1.[6] We note that our Alirector consistently outperforms all baselines in $F_{0.5}$ across all the datasets, demonstrating the effectiveness of this method. In contrast, the Copy method even underperforms vanilla fine-tuning in some cases. Besides, Alirector achieves considerable improvements in precision across all the datasets, highlighting the efficacy of this approach in mitigating overcorrection. Further analysis regarding the effect of Alirector on reducing overcorrection is presented in Section 5.1.

Moreover, we make several interesting observations. First, despite the notable enhancement achieved by Alirector, the decoder-only LLMs of Baichuan2 and Chinese-LLaMA2 still struggle to outperform BART. This can be attributed to the fact that BART's pre-training involves a series of denoising tasks utilizing strategies like token masking, token deletion and text infilling, which are naturally suitable for the CGEC/GEC task (Lewis et al., 2020; Wang et al., 2023). Second, different models exhibit varying degrees of improvement by employing Alirector, with decoder-only LLMs generally experiencing more notable improvements than Seq2Seq models. The performance of Chinese-LLaMA2 is much worse than Baichuan2, which may be attributed to their different capabilities achieved through pre-training. Third, Alirector yields more pronounced improvements on the FCGEC and NaCGEC datasets than on the NLPCC18 dataset. We attribute this discrepancy to the differing error distributions across datasets. The errors in NLPCC18, derived from Chinese-as-a-Second-Language learners, are less common, while the errors in FCGEC and NaCGEC, stemming from native speakers, exhibit more prevalent patterns that are easier for the model to learn.

[5] https://huggingface.co/Linly-AI/Chinese-LLaMA-2-7B-hf

[6] More experimental results can be found in Appedix E.

| Model | Method | NLPCC18-Test | | | NaCGEC-Test | | | FCGEC-Dev | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | P | R | $F_{0.5}$ | P | R | $F_{0.5}$ | P | R | $F_{0.5}$ |
| Transformer | Vanilla FT | 42.37 | 23.49 | 36.50 | 59.67 | 28.69 | 49.07 | 47.83 | 22.99 | 39.33 |
| | TemplateGEC | 42.00 | 22.20 | 35.60 | - | - | - | - | - | - |
| | SynGEC | 41.44 | **28.28** | 37.91 | 51.45 | **39.69** | 48.57 | 38.00 | **32.18** | 36.67 |
| | Copy | 43.16 | 23.58 | 37.01 | 64.61 | 26.42 | 50.12 | 48.95 | 19.77 | 37.79 |
| | Alirector | **45.98** | 22.87 | **38.25** | **65.44** | 31.27 | **53.70** | **57.86** | 24.15 | **45.23** |
| BART | Vanilla FT | 50.63 | 31.83 | 45.28 | 65.85 | 40.79 | 58.64 | 56.26 | **40.71** | 52.27 |
| | TemplateGEC | **54.50** | 27.40 | 45.50 | - | - | - | - | - | - |
| | SynGEC | 49.96 | 33.04 | 45.32 | 63.76 | **47.41** | 59.65 | 53.11 | 39.45 | 49.67 |
| | Copy | 51.25 | 32.55 | 45.97 | 66.67 | 41.88 | 59.61 | 58.55 | 38.46 | 53.01 |
| | Alirector | 51.76 | **33.49** | **46.67** | **68.11** | 43.87 | **61.33** | **58.78** | 39.15 | **53.42** |
| Baichuan2-7B | Vanilla FT | 51.69 | 27.92 | 44.17 | 62.93 | 44.50 | 58.12 | 51.77 | 38.10 | 48.31 |
| | Copy | 51.56 | **28.53** | 44.39 | 62.27 | 44.20 | 57.56 | 53.47 | 35.51 | 48.56 |
| | Alirector | **52.27** | 27.14 | **45.01** | **66.04** | **45.91** | **60.71** | **58.55** | **39.74** | **53.49** |
| Chinese-LLaMA2-7B | Vanilla FT | 45.85 | 27.44 | 40.43 | 61.93 | 30.31 | 51.24 | 50.15 | 26.19 | 42.39 |
| | Copy | 46.53 | **27.93** | 41.06 | 62.15 | 30.54 | 51.49 | 48.04 | 28.35 | 42.18 |
| | Alirector | **47.43** | 26.96 | **41.18** | **62.60** | **32.90** | **53.03** | **52.64** | **28.47** | **45.00** |

Table 1: Overall results on NLPCC18-Test, NaCGEC-Test, and FCGEC-Dev datasets. The results of TemplateGEC (Li et al., 2023a) and SynGEC (Zhang et al., 2022b) on NLPCC18 are cited from the original papers, and other results including Copy (Zhao et al., 2019) are implemented by us. Best results are highlighted in bold.

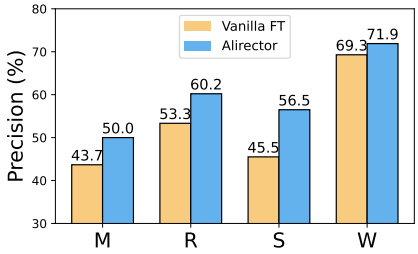


Figure 5: Results of precision for different error types, including *missing* (M), *redundant* (R), *substitution* (S), and *word-order* (W), on the FCGEC-Dev test set.

| Type | #Overcorrections / #Undercorrections | |
|---|---|---|
| | Vanilla FT | Alirector |
| M | 129 / 259 | 113 (-12.4%) / 245 |
| R | 203 / 181 | 152 (-25.1%) / 183 |
| S | 91 / 226 | 67 (-26.4%) / 215 |
| W | 39 / 140 | 34 (-12.8%) / 141 |
| All | 462 / 806 | 366 (-20.8%) / 784 |

Table 2: The number of overcorrections and undercorrections reduced by BiAlign over direct fine-tuning for different error types on FCGEC-Dev.

## 5 Analysis

### 5.1 Overcorrection Mitigation

To further verify the effectiveness of Alirector in mitigating overcorrection, we use Baichuan2-7B as the backbone and present fine-grained precision results across the four categories of CGEC errors, including *missing* (M), *redundant* (R), *substitution* (S), and *word-order* (W), in Figure 5. Moreover, we present in Table 2 the number of overcorrections and undercorrections that Alirector reduces on the four error types compared to Baichuan2-7B. The results depicted in Figure 5 and summarized in Table 2 demonstrate that Alirector significantly enhances precision for all error types while notably decreasing the number of overcorrections without deteriorating undercorrection, particularly for the *redundant* and *substitution* types. These findings support the effectiveness of Alirector in mitigating overcorrection induced by generative language models and in enhancing the robustness of our method across different error types. For a more intuitive illustration of Alirector's effectiveness, we provide a case study in Appendix F.

### 5.2 Ablation Study

To investigate the contribution of key components of our approach, we conduct in-depth ablation experiments on NaCGEC-Test and FCGEC-Dev datasets using BART and Baichuan2-7B.

**Distillation from Alignment Models** We first ablate different alignment distillation components in turn to analyze their contribution. As shown in Table 3, while removing either forward distillation $\mathcal{L}_{\text{kd}}^{f}$ or reverse distillation $\mathcal{L}_{\text{kd}}^{r}$ causes noticeable performance degradation, there is a significant performance drop after removing the overall distillation loss $\mathcal{L}_{\text{kd}}$, particularly in recall and $F_{0.5}$. This indicates that bidirectional alignment contributes more to performance improvement through knowl-

| Method | NaCGEC-Test | | | FCGEC-Dev | | |
|---|---|---|---|---|---|---|
| | P | R | $F_{0.5}$ | P | R | $F_{0.5}$ |
| **BART** | | | | | | |
| **Alirector** | 68.11 | **43.87** | **61.33** | 58.78 | **39.15** | **53.42** |
| *w/o* $\mathcal{L}_{\text{kd}}^{f}$ | **68.30** | 40.44 | 60.03 | **59.70** | 36.46 | 52.95 |
| *w/o* $\mathcal{L}_{\text{kd}}^{r}$ | 68.19 | 43.41 | 61.21 | 59.56 | 37.41 | 53.25 |
| *w/o* $\mathcal{L}_{\text{kd}}$ | 67.17 | 40.79 | 59.48 | 56.26 | 40.71 | 52.27 |
| *disc.* source | 65.44 | 41.87 | 58.82 | 57.62 | 38.69 | 52.48 |
| *disc.* predict | 67.93 | 39.53 | 59.40 | 59.22 | 35.08 | 52.05 |
| **Baichuan2-7B** | | | | | | |
| **Alirector** | 66.04 | **45.91** | **60.71** | **58.55** | **39.74** | **53.49** |
| *w/o* $\mathcal{L}_{\text{kd}}^{f}$ | 65.92 | 43.72 | 59.84 | 57.88 | 38.57 | 52.62 |
| *w/o* $\mathcal{L}_{\text{kd}}^{r}$ | **66.91** | 40.99 | 59.40 | 55.99 | 36.66 | 50.65 |
| *w/o* $\mathcal{L}_{\text{kd}}$ | 62.93 | 44.50 | 58.12 | 51.77 | 38.10 | 48.31 |
| *disc.* source | 59.98 | 49.46 | 57.53 | 51.46 | 39.22 | 48.44 |
| *disc.* predict | 66.05 | 41.78 | 59.18 | 53.47 | 35.51 | 48.56 |

Table 3: Results of ablation study on NaCGEC-Test and FCGEC-Dev, where "*disc.*" is short for "discard".

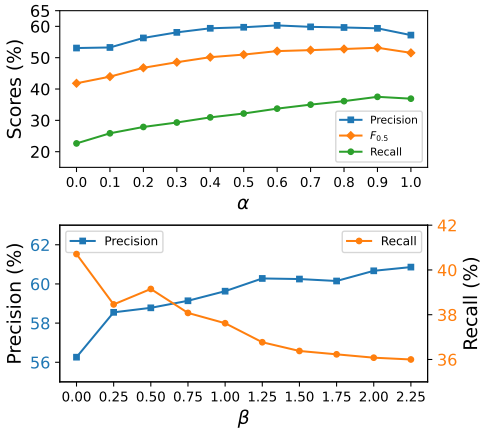


Figure 6: Results of our method on FCGEC development set with different values of $\alpha$ and $\beta$ that control the weight of forward and reverse alignment losses.

edge distillation compared to unidirectional alignment. Moreover, the notable drops in precision when removing any of the forward or reverse distillation loss suggest that the alignment distillation is essential for our method to mitigate overcorrection.

**Input of Alignment Models** To further investigate the effect of the alignment between the source sentence and the initial correction, we conduct additional experiments by ablating the source sentence or initial correction from the input of the alignment models during training Alirector. To keep the format of the input, we ablate the source sentence by replacing it with the initial correction, e.g., $\hat{Y} + [\text{SEP}] + \hat{Y}$ for Seq2Seq. Similarly, we construct the input as $X + [\text{SEP}] + X$ when ablating the initial correction. As shown in Table 3, we observe that ablating the source sentence causes an obvious decline in precision while ablating the initial correction leads to a notable drop in recall. These findings highlight the role of alignment in reducing both overcorrection and undercorrection.

### 5.3 Impact of $\alpha$ and $\beta$

The training objective of Alirector involves $\alpha$ to control the weight of forward and reverse alignment losses, as well as $\beta$ to balance between the original GEC loss and the distillation loss. To investigate their impact on model performance, we use BART as the backbone and show the results of different values of $\alpha$ and $\beta$ on the FCGEC development set in Figure 6, where we change one while fixing the other. From the first subfigure, we observe that as $\alpha$ increases, the P/R/$F_{0.5}$ scores consistently rise and achieve the best results around 0.9. The second subfigure shows that as $\beta$ increases, the precision rises accordingly while recall gradually falls. This trend indicates that $\beta$ plays a role in balancing between precision and recall. Similar trends can be observed when other models are employed, and the optimal values of $\alpha$ and $\beta$ are provided in Table 6.

## 6 Conclusion

In this paper, we first investigate a predict-and-align method that effectively leverages alignment information between the source sentence and the initial correction to alleviate the overcorrection issue in CGEC. Then, we propose transferring knowledge from the alignment process to enhance the correction model, resulting in an improved model termed Alirector. Experimental results on three CGEC datasets showcase the efficacy of our approach in mitigating overcorrection for both Seq2Seq models and decoder-only LLMs. Detailed analysis further demonstrates the effectiveness of this method across various error types, as well as the pivotal role of alignment in enhancing performance.

Broadly speaking, the overcorrection challenge falls within the realm of uncontrollability of generative language models. Besides straightforward efforts to acquire more high-quality training data or employ specific pre-training strategies such as BART, this study introduces an alignment-based method that has demonstrated effectiveness in addressing this issue. Despite the improvement of our approach for decoder-only LLMs, their performance in CGEC still lags behind that of the strongest Seq2Seq models, even though they are smaller in size, which contradicts their outstanding performance in other NLP tasks. In future research, we will further exploring enhancing the performance of decoder-only LLMs for CGEC.

## Limitations

The potential limitations of our work are threefold. First, we have exclusively validated our approach on Chinese GEC datasets. However, our approach is language-independent, and it can be investigated in other languages. Second, our approach incurs additional training costs, as training alignment models and performing knowledge distillation are required. Third, our experiments are confined to 7B-scale LLMs using the QLoRA efficient fine-tuning technique. Due to computational resource constraints, we have not explored the impact of larger-scale LLMs and full-parameter fine-tuning, which may lead to improved performance.

## Ethics Statement

This work aims to propose a technical method to mitigate overcorrection caused by Seq2Seq models and decoder-only LLMs in Chinese grammatical error correction, which will not cause ethical issues. All datasets and models used in this work are publicly available, and we adhere strictly to their usage policies. We are committed to conducting our research in an ethical and responsible manner.

## References


Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4260–4270, Hong Kong, China. Association for Computational Linguistics.

Mengyun Chen, Tao Ge, Xingxing Zhang, Furu Wei, and Ming Zhou. 2020. Improving the efficiency of grammatical error correction with erroneous span detection and correction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7162–7169, Online. Association for Computational Linguistics.

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.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

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.

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. In *Natural Language Processing and Chinese Computing*, pages 69–80, Cham. Springer Nature Switzerland.

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. *arXiv preprint arXiv:2304.01746*.

Kai Fu, Jin Huang, and Yitao Duan. 2018. Youdao's winning solution to the nlpcc-2018 task 2 challenge: A neural machine translation approach to chinese grammatical error correction. In *Natural Language Processing and Chinese Computing*, pages 341–350, Cham. Springer International Publishing.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

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, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4248–4254, Online. Association for Computational Linguistics.

Masahiro Kaneko and Naoaki Okazaki. 2023. Controlled generation with prompt insertion for natural language explanations in grammatical error correction. *arXiv preprint arXiv:2309.11439*.

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

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.

Yinghao Li, Xuebo Liu, Shuo Wang, Peiyuan Gong, Derek F. Wong, Yang Gao, Heyan Huang, and Min Zhang. 2023a. 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, Haojing Huang, Shirong Ma, Yong Jiang, Yangning Li, Feng Zhou, Hai-Tao Zheng, and Qingyu Zhou. 2023b. On the (in) effectiveness of large language models for chinese text correction. *arXiv preprint arXiv:2307.09007*.

Deng Liang, Chen Zheng, Lei Guo, Xin Cui, Xiuzhang Xiong, Hengqiao Rong, and Jinpeng Dong. 2020. BERT enhanced neural machine translation and sequence tagging model for Chinese grammatical error diagnosis. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 57–66, Suzhou, China. Association for Computational Linguistics.

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

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.

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.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Chanjun Park, Yeongwook Yang, Chanhee Lee, and Heuiseok Lim. 2020. Comparison of the evaluation metrics for neural grammatical error correction with overcorrection. *IEEE Access*, 8:106264–106272.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, et al. 2023. Large language models are effective text rankers with pairwise ranking prompting. *arXiv preprint arXiv:2306.17563*.

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*.

Yunfan Shao, Zhichao Geng, Yitao Liu, Junqi Dai, Hang Yan, Fei Yang, Li Zhe, Hujun Bao, and Xipeng Qiu. 2021. Cpt: A pre-trained unbalanced transformer for both chinese language understanding and generation. *arXiv preprint arXiv:2109.05729*.

Yixiao Song, Kalpesh Krishna, Rajesh Bhatt, Kevin Gimpel, and Mohit Iyyer. 2023. Gee! grammar error explanation with large language models. *arXiv preprint arXiv:2311.09517*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Hongfei Wang, Michiki Kurosawa, Satoru Katsumata, Masato Mita, and Mamoru Komachi. 2023. Chinese grammatical error correction using pre-trained models and pseudo data. *ACM Transactions on Asian*

*and Low-Resource Language Information Processing*, 22(3):1–12.

Yu Wang, Yuelin Wang, Kai Dang, Jie Liu, and Zhuo Liu. 2021. A comprehensive survey of grammatical error correction. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 12(5):1–51.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

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.

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

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.

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.

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

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.

## A Instruction Templates

In our experiments, we explored various instruction templates and observed that the choice of instruction templates has a limited impact on the experimental results, particularly when the amount of training data is sufficient. Table 4 presents the instruction templates $\mathcal{T}_{\text{gec}}$ and $\mathcal{T}_{\text{align}}$ used in our experiments for tuning LLMs. The instruction template comprises an input field that provides the source text and a response field that denotes the target text.

| **LLM** | **Instruction for correction model $\mathcal{T}_{\text{gec}}$** |
|---|---|
| Baichuan2 | 纠正输入句子中的语法错误，并输出正确的句子。<br>(Trans.: *Correct grammatical errors in the input sentence and output the correct sentence.*)<br>Input: {Source}<br>Response: {Target/Output} |
| Chinese-LLaMA2 | ### Instruction: 纠正输入句子中的语法错误，并输出正确的句子。<br>(Trans.: *Correct grammatical errors in the input sentence and output the correct sentence.*)<br>### Input: {Source}<br>### Response: {Target/Output} |
| **LLM** | **Instruction for alignment model $\mathcal{T}_{\text{align}}$** |
| Baichuan2 | 对齐输入中用"\t"分隔的两个句子，并输出没有语法错误的句子。<br>(Trans.: *Align the two sentences separated by "\t" in the input and output the sentence without grammatical errors.*)<br>Input: {Source} \t {Initial Correction}<br>Response: {Target/Output} |
| Chinese-LLaMA2 | ### Instruction: 对齐输入中用"\t"分隔的两个句子，并输出没有语法错误的句子。<br>(Trans.: *Align the two sentences separated by "\t" in the input and output the sentence without grammatical errors.*)<br>### Input: {Source} \t {Initial Correction}<br>### Response: {Target/Output} |

Table 4: Instruction templates for the correction model and alignment models, where "Trans." denotes the translation of the instruction.

## B Datasets

The statistics of the datasets used in our experiments are shown in Table 5. For CSL learner data, we adopted the same training set as Zhang et al. (2022a), which involves discarding all samples without grammatical errors in the Lang8 and HSK datasets and replicating the HSK dataset five times and combining with the Lang8 dataset, resulting in a total of 1,568,885 sentence pairs.

| **Train** | **Source** | **#Sent** | **#Error** |
|---|---|---|---|
| **Lang8** | Learner | 1,220,906 | 1,092,285 (89.5%) |
| **HSK** | Learner | 15,6870 | 95,320 (60.8%) |
| **FCGEC** | Native | 35,341 | 19,183 (54.3%) |
| **Dev** | **Source** | **#Sent** | **#Error** |
| **MuCGEC-Dev** | Learner | 2,467 | 2,409 (97.6%) |
| **FCGEC** | Native | 1,000 | 563 (56.3%) |
| **Test** | **Source** | **#Sent** | **#Error** |
| **NLPCC18-Test** | Learner | 2,000 | 1,983 (99.2%) |
| **FCGEC-Dev** | Native | 2,000 | 1,101 (55.1%) |
| **NaCGEC-Test** | Native | 5,869 | 5,612 (95.6%) |

Table 5: Statistics of the used CGEC datasets. **#Sent** denotes the number of the sentences and **#Error** denotes the number (the percentage) of the erroneous sentences.

## C Experimental Details

### C.1 Training Details

**Training on Native Speaker Datasets** Since FCGEC contains only 35,341 training samples, which is insufficient for model training, we performed continuous training on the FCGEC training set with the model trained on the CSL learner data.

**Training of Alignment Models** As described in Section 3.2, before training the alignment models, we need to obtain initial corrections using an initial correction model. For this purpose, we divided the training data into two parts, one for training the initial correction model and the other for training the alignment models. In the case of CSL learner data, 80%[7] of the training samples are randomly selected to train the initial correction model. Then, this correction model is used to generate initial corrections for the remaining training samples. These initial corrections along with their corresponding source and target sentences are used to train the alignment models. For the native speaker datasets, we used the correction model trained on the CSL learner data to generate initial corrections on the FCGEC training set.

**Training of Alirector** As outlined in Section 3.3, we perform knowledge distillation using the correction model as the student and the alignment models as the teachers. For this training, we used the same training set as that used for training the teachers. The student was initialized with the weights of the well-trained initial correction model.

[7]We experimented with different ratios, including 4:6, 5:5, and 8:2, and found that 8:2 works the best.

| Hyperparameter | NLPCC18 | | FCGEC/NaCGEC | |
|---|---|---|---|---|
| | | **Seq2Seq** | | |
| Backbone | Transformer-large | BART-large | Transformer-large | BART-large |
| Batch size | 1024 | 1024 | 256 | 256 |
| Max Epochs | 20 | 10 | 20 | 10 |
| Max Length | | 128 (Source); 128 (Target) | | |
| Learning Rate | $3 \times 10^{-4}$ | $3 \times 10^{-5}$ | $3 \times 10^{-5}$ | $3 \times 10^{-5}$ |
| Warmup Steps | 3000 | 1000 | 100 | 100 |
| Dropout | 0.3 | 0.1 | 0.3 | 0.1 |
| Dropout-Src | 0.2 | 0.2 | 0.2 | 0.2 |
| $\alpha$ | 0.7 | 0.5 | 0.5 | 0.9 |
| $\beta$ | 1.0 | 1.5 | 0.5 | 0.5 |
| $\tau$ | 1 | 1 | 1 | 1 |
| Beam Size | 10 | 10 | 10 | 10 |
| | | **LLMs** | | |
| Backbone | Baichuan2-7B | Chinese-LLaMA2-7B | Baichuan2-7B | Chinese-LLaMA2-7B |
| Batch size | 1024 | 1024 | 256 | 256 |
| Max Epochs | 3 | 5 | 10 | 10 |
| Max Length | | 192 (GEC); 256 (Alignment) | | |
| Learning Rate | $3 \times 10^{-5}$ | $3 \times 10^{-4}$ | $3 \times 10^{-5}$ | $3 \times 10^{-5}$ |
| Warmup Steps | 1000 | 1000 | 100 | 100 |
| LoRA | target modules = all linears; lora rank = 8; lora alpha = 16, lora dropout = 0.05 | | | |
| $\alpha$ | 0.3 | 0.5 | 0.3 | 0.5 |
| $\beta$ | 1.5 | 2.0 | 0.5 | 1.0 |
| $\tau$ | 1 | 1 | 1 | 1 |
| Beam Size | 10 | 10 | 10 | 10 |

Table 6: Hyperparameter settings in our experiments.

### C.2 Implementation Details

For Seq2Seq model training, following Zhang et al. (2022b), we utilized the Dropout-Src technique (Junczys-Dowmunt et al., 2018) that applies dropout on input embeddings for alleviating over-fitting. As for LLMs tuning, considering the time and computational resources, we applied QLoRA (Dettmers et al., 2023) for efficient fine-tuning instead of full-parameter fine-tuning. Our code implementation mainly follows the *Alpaca LoRA* project[8], and is based on the Huggingface Transformers (Wolf et al., 2020) and bitsandbytes[9] (Dettmers et al., 2022) toolkit in Pytorch. We searched for the optimal value of $\alpha$ in {0.1, 0.3, 0.5, 0.7, 0.9}, $\beta$ in {0.5, 1.0, 1.5, 2.0} and the temperature $\tau$ in {1, 2, 3, 4, 5} on the development set. We used the Adam optimizer (Kingma and Ba, 2014) and polynomial learning rate decay. The hyperparameter settings are presented in Table 6. All experiments are carried out on 8 GeForce RTX 4090 24GB GPUs.

[8] https://github.com/tloen/alpaca-lora

[9] https://github.com/TimDettmers/bitsandbytes

| Method | NaCGEC-Test | | | FCGEC-Dev | | |
|---|---|---|---|---|---|---|
| | **P** | **R** | **$F_{0.5}$** | **P** | **R** | **$F_{0.5}$** |
| **Vanilla FT** | 62.93 | 44.50 | 58.12 | 56.26 | 40.71 | 52.27 |
| **predict-and-align** | **67.21** | 45.61 | **61.39** | **62.60** | 37.43 | **55.18** |
| *repl.* src+src | 66.05 | 41.78 | 59.18 | 61.50 | 32.25 | 52.06 |
| *repl.* pred+pred | 59.98 | **49.56** | 57.53 | 50.59 | **42.48** | 48.73 |
| **Alirector** | 66.04 | 45.91 | 60.71 | 58.55 | 39.74 | 53.49 |

Table 7: Results of the potential of alignment on FCGEC-Dev, where "*repl.*" is short for "replace".

## D Potential of Alignment

The alignment models are introduced to mitigate overcorrection by leveraging the alignment information between the source sentence and the initial correction. To demonstrate the potential of the alignment models, we employed a BART-based alignment model to align the predictions of a Baichuan2-based correction model, and present the comparison results between vanilla fine-tuning (i.e., without alignment) and predict-and-align method in Table 7. From the results, we note that predict-and-align improves precision and $F_{0.5}$ by a large margin compared to vanilla fine-tuning. Notably, predict-and-align even outperforms our Alirector, highlighting the effectiveness and the potential of the two-stage alignment method. Moreover, when

only source information is retained in the input (namely *repl.* src+src), we observe high precision but low recall, while retaining only prediction information (namely *repl.* pred+pred) exhibits the opposite trend. This observation once again emphasizes the role of alignment in reducing both overcorrection and undercorrection.

# E Additional Experimental Results

## E.1 Results on FCGEC-Test

We conducted additional experiments on FCGEC-Test[10], the test set of FCGEC (Xu et al., 2022), for more comprehensive evaluation. As shown in Table 8, our Alirector improves the P/$F_{0.5}$ score by 5.59/1.77 over vanilla fine-tuning on BART, while the improvement is 4.37/2.25 on Baichuan2-7B. In contrast, the Copy method has only a minor improvement over vanilla fine-tuning.

| Model | Method | FCGEC-Test | | |
|---|---|---|---|---|
| | | P | R | $F_{0.5}$ |
| BART | Vanilla FT | 63.85 | **40.16** | 57.11 |
| | Copy | 65.31 | 39.45 | 57.74 |
| | Alirector | **69.44** | 36.60 | **58.88** |
| Baichuan2-7B | Vanilla FT | 60.12 | **37.21** | 53.53 |
| | Copy | 62.14 | 35.47 | 54.01 |
| | Alirector | **64.49** | 36.22 | **55.78** |

Table 8: Results on FCGEC-Test.

## E.2 Results on More LLMs

We also implemented our Alirector method on other Chinese LLMs, namely Yi-6B[11], a Chinese LLM that employs the same architecture as LLaMA (Touvron et al., 2023a), and ChatGLM3-6B[12], a chat model based on GLM (Du et al., 2022). The results on NLPCC18-Test are shown in Table 9. We observe that our Alirector method improves the precision/$F_{0.5}$ score over vanilla fine-tuning by 3.62/1.75 on Yi-6B and 3.14/1.05 on ChatGLM3-6B, respectively. This highlights the generalizability of Alirector across other LLMs. However, the performance of Yi-6B and ChatGLM3-6B lags significantly behind that of Baichuan2-7B. This discrepancy can be attributed to the different capabilities achieved through pre-training.

[10]FCGEC-Test provides online evaluation at https://codalab.lisn.upsaclay.fr/competitions/8020.

[11]https://huggingface.co/01-ai/Yi-6B

[12]https://huggingface.co/THUDM/chatglm3-6b-base

| Model | Method | NLPCC18-Test | | |
|---|---|---|---|---|
| | | P | R | $F_{0.5}$ |
| Yi-6B | Vanilla FT | 50.61 | **25.22** | 42.13 |
| | Copy | 50.11 | 25.01 | 41.73 |
| | Alirector | **54.23** | 24.89 | **43.88** |
| ChatGLM3-6B | Vanilla FT | 48.61 | **26.46** | 41.64 |
| | Copy | 49.64 | 25.23 | 41.59 |
| | Alirector | **51.75** | 25.11 | **42.69** |

Table 9: Results on Yi-6B and ChatGLM3-6B.

# F Case Study

We provide four examples in Table 10 to illustrate the effectiveness of our Alirector in mitigating overcorrection. We can note that the vanilla fine-tuned model often tends to overcorrect by modifying the error-free characters. In contrast, Alirector is able to correct all the errors in the sentence while preserving the error-free characters. These cases intuitively show that Alirector learns to identify and correct the potential errors in the sentence while actively avoiding overcorrection as much as possible.

| | |
|---|---|
| **Source** | 在过去一年，我校采取了一系列卓有成效的改进方法。<br>In the past year, our school has adopted a series of effective improvement methods. |
| **Target** | 在过去一年，我校采取了一系列卓有成效的改进措施。<br>In the past year, our school has adopted a series of effective improvement measures. |
| **Vanilla FT** | 在过去一年，我们采取了一系列卓有成效的解决措施。(✗)<br>In the past year, we have adopted a series of effective solution measures. |
| **Alirector** | 在过去一年，我校采取了一系列卓有成效的改进措施。(✓)<br>In the past year, our school has adopted a series of effective improvement measures. |
| **Source** | 他已经回国，现就任于北大医学部体育教授。<br>He has returned to China and is now serving in a professor of physical education in the Peking University Health Science Center. |
| **Target** | 他已经回国，现就任于北大医学部，担任体育教授。<br>He has returned to China and is now serving as a professor of physical education in the Peking University Health Science Center. |
| **Vanilla FT** | 他已经回国，现就任于北大医学部~~体育教授~~。(✗)<br>He has returned to China and is now serving in the Peking University Health Science Center. |
| **Alirector** | 他已经回国，现就任于北大医学部，担任体育教授。(✓)<br>He has returned to China and is now serving as a professor of physical education in the Peking University Health Science Center. |
| **Source** | 斯诺登虽然决定了自己的住处，而且出于安全考虑他不会公布住址。<br>Snowden has decided where he is going to live, though, and he will not release his address for security reasons. |
| **Target** | 斯诺登虽然决定了自己的住处，但是出于安全考虑他不会公布住址。<br>Snowden has decided where he will live, but he will not release his address for security reasons. |
| **Vanilla FT** | 虽然斯诺登决定了自己的住处，但是出于安全考虑他不会公布住址。(✗)<br>Although Snowden has decided where he will live, he will not release his address for security reasons. |
| **Alirector** | 斯诺登虽然决定了自己的住处，但是出于安全考虑他不会公布住址。(✓)<br>Snowden has decided where he will live, but he will not release his address for security reasons. |
| **Source** | 这样不仅有助于维护国家安全和社会稳定，而且有利于提高工作效率，有利于金融机构落实存款实名制。<br>This will not only help maintain national security and social stability, but also help improve work efficiency and help financial institutions implement the real-name deposit system. |
| **Target** | 这样不仅有利于提高工作效率，有利于金融机构落实存款实名制，而且有助于维护国家安全和社会稳定。<br>This will not only help improve work efficiency, but also help financial institutions implement the real-name deposit system and help maintain national security and social stability. |
| **Vanilla FT** | 这样不仅~~有利于提高工作效率~~，有利于金融机构落实存款实名制，而且有助于维护国家安全和社会稳定。(✗)<br>This will not only help financial institutions implement the real-name deposit system, but also help maintain national security and social stability. |
| **Alirector** | 这样不仅有利于提高工作效率，有利于金融机构落实存款实名制，而且有助于维护国家安全和社会稳定。(✓)<br>This will not only help improve work efficiency, but also help financial institutions implement the real-name deposit system and help maintain national security and social stability. |

Table 10: A case study of vanilla fine-tuning and our Alirector using Baichuan2-7B on FCGEC-dev and NaCGEC-Test, where overcorrected characters and their error-free counterparts are highlighted in red and orange, respectively, and correct edits are highlighted in blue.