LM-Combiner: A Contextual Rewriting Model for Chinese Grammatical Error Correction

Yixuan Wang[1], **Baoxin Wang**[1,2], **Yijun Liu**[1], **Dayong Wu**[2], **Wanxiang Che**[1,*]
[1]Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, China
[2]State Key Laboratory of Cognitive Intelligence, iFLYTEK Research, China
{yixuanwang, yijunliu, car}@ir.hit.edu.cn
{bxwang2, dywu2}@iflytek.com

Abstract

Over-correction is a critical problem in Chinese grammatical error correction (CGEC) task. Recent work using model ensemble methods based on voting can effectively mitigate over-correction and improve the precision of the GEC system. However, these methods still require the output of several GEC systems and inevitably lead to reduced error recall. In this light, we propose the LM-Combiner, a rewriting model that can directly modify the over-correction of GEC system outputs without a model ensemble. Specifically, we train the model on an over-correction dataset constructed through the proposed K-fold cross inference method, which allows it to directly generate filtered sentences by combining the original and the over-corrected text. In the inference stage, we directly take the original sentences and the output results of other systems as input and then obtain the filtered sentences through LM-Combiner. Experiments on the FCGEC dataset show that our proposed method effectively alleviates the over-correction of the original system (+18.2 Precision) while ensuring the error recall remains unchanged. Besides, we find that LM-Combiner still has a good rewriting performance even with small parameters and few training data, and thus can cost-effectively mitigate the over-correction of black-box GEC systems (e.g., ChatGPT).

Keywords: Grammatical Error Correction, Language Model, Text Rewriting

1. Introduction

Grammatical error correction (GEC) is a formally simple but challenging task (Wang et al., 2020; Bryant et al., 2022), which aims to identify and correct grammatical errors present in a sentence. As a basic application task, it has a wide range of applications in areas such as search engines, automatic speech recognition (ASR) systems, and writing assistants (Omelianchuk et al., 2020). In terms of model architecture, the mainstream approaches can be categorized into the auto-encoding Seq2Edit model and the auto-regressive Seq2Seq model.

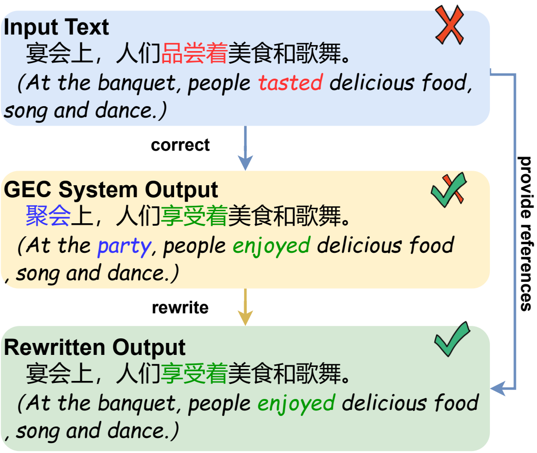
Input Text
宴会上，人们品尝着美食和歌舞。
(At the banquet, people tasted delicious food, song and dance.)
correct
GEC System Output
聚会上，人们享受着美食和歌舞。
(At the party, people enjoyed delicious food , song and dance.)
rewrite
Rewritten Output
宴会上，人们享受着美食和歌舞。
(At the banquet, people enjoyed delicious food , song and dance.)
provide references

Figure 1: An example of the problem of over-correction, where red represents grammatical errors, blue represents over-correction, and green represents correct changes. LM-Combiner can directly rewrite the system output with reference to the original sentence, filtering out the over-corrections.

Over-correction has always been a challenge in GEC tasks (Tang et al., 2023), which can seriously affect the precision rate of the GEC system. As shown in Figure 1, the over-correction problem is that the error correction system modifies the correct part of a sentence to some other expressions. Although sometimes these expressions don't differ much from the meaning of the original sentence, as a correction system, excessive modification of the input can still cause annoyance to the user. Compared to English GEC, Chinese GEC faces a more severe over-correction problem due to the lack of training data and more difficult errors. Specifically, the previous Chinese GEC task datasets are mainly sourced from non-native learners, with low-quality and inconsistently annotated training sets. In addition to disfluencies such as spelling errors in English, most of the errors in CGEC involve syntactic and semantic information, which are difficult and make the model prone to false corrections. The above factors result in the precision of the same baseline model on the Chinese dataset usually being only about half of the rate on the English dataset. It can be said that over-correction is a key difficulty

[*] Email corresponding.

LREC-COLING 2024, pages 10675–10685

of CGEC task and deserves a deeper study.

Nowadays, model ensemble is a primary solution to the problem of over-correction. Li et al. (2018); Liang et al. (2020) view error correction as different types of edit labels and vote to integrate the system based on the labels. Zhang et al. (2022a) integrate multiple architectures of CGEC systems through the method of label voting, improving the precision rate significantly. Tang et al. (2023) integrate the outputs of multiple error correction systems at different granularities by computing the perplexity (PPL) through a language model to obtain the final output. While the above methods can improve the final precision rate, they all suffer from two key problems that need to be solved. **(1)** ***Excessive Cost.*** As ensemble methods, they typically require the results of several models, leading to greater costs in the training phase and longer time in the inference phase. **(2)** ***Reduced Recall.*** Current methods for alleviating over-correction all lead to a significant decrease in error recall rate, which seriously affects the usability of the correction system. Voting methods inevitably lead to some decrease in recall, and PPL-based methods can't make accurate judgements on various domain datasets without fine-tuned LMs.

To better mitigate the problem of over-correction, we propose the **LM-Combiner**, a trainable LM-based text rewriting model. It can filter the output of a GEC system without a model ensemble, significantly reducing the problem of over-correction while ensuring as much error recall as possible. In summary, we decouple the over-correction problem from the Chinese grammatical error correction task and treat it as a post-processing rewriting task. Different from the model ensemble methods, the rewriting model simply takes the original sentence and the result of a single GEC system as inputs, and directly outputs suitable combinations of the two sentences as results.

Specifically, we design the LM-Combiner at the data and model level to ensure its effectiveness. At the data level, we propose an overcorrected dataset construction method based on the idea of k-fold cross validation. We divide the training set multiple times, use parts for the model training, and inference on the remaining data to obtain naturally overcorrected sentences. In addition to this, we propose the gold labels merging approach to further decouple the correction task and the rewriting tasks, so that the LM-Combiner only needs to select from the over-correction and right correction in output sentences of GEC systems. At the model level, we are inspired by Tang et al. (2023) to further explore the application of causal language models to the Chinese grammatical error correction task. Compared to directly using PPL as a criterion, we find that after fine-tuning on the corresponding domain dataset as a rewriting model, GPT2 can better retain the right correction while filtering over-correction, resulting in higher recall.

We evaluate the proposed method on the FCGEC dataset (Xu et al., 2022) sourced from a native speaker corpus. With the rewriting of the LM-Combiner, we improve the precision of the baseline model by 18.2 points, while ensuring that the recall remain basically unchanged, and the $F_{0.5}$ improves by 5.8 points to reach the level of SOTA. Besides, experiments show that LM-Combiner has small requirements on model size and data quantity, and can achieve excellent results just by training with base-level models and thousand-level data quantity.

The main contributions of this paper can be summarized as follows:

- We propose a novel rewriting model, LM-Combiner, which can effectively mitigate over-correction of the existing GEC systems without model ensemble.
- We propose k-fold cross inference, a construction method for over-correction data. It can stably construct over-corrected sentences for LM-Combiner training from existing parallel corpora.
- Experiments show that the proposed rewriting method can greatly improve the precision of the GEC system while maintaining the recall constant.
- We also find that the LM-Combiner achieves good rewriting results even with small parameters and few training data, which provides a cost-saving solution to alleviate the over-correction of existing black-box GEC systems.

We will release our code and model[1].

2. Method

The core of our proposed method is a rewriting model, which can alleviate over-correction in the original GEC system by direct rewriting. The workflow of the correction-rewriting framework is shown in Figure 2. Inspired by the recent use of LM for model ensemble (Tang et al., 2023) in the CGEC domain, we train an LM rewriting model that uses only the original sentence and the output of a single system as inputs, which can filter out over-correction and retain as many right correction as possible. Specifically, we try the application of causal LM on CGEC (Section 2.1), based on which we propose a rewriting model LM-Combiner (Section 2.2) and an over-correction data construction method (Section 2.3) for the model training.

[1] https://github.com/wyxstriker/LM-Combiner

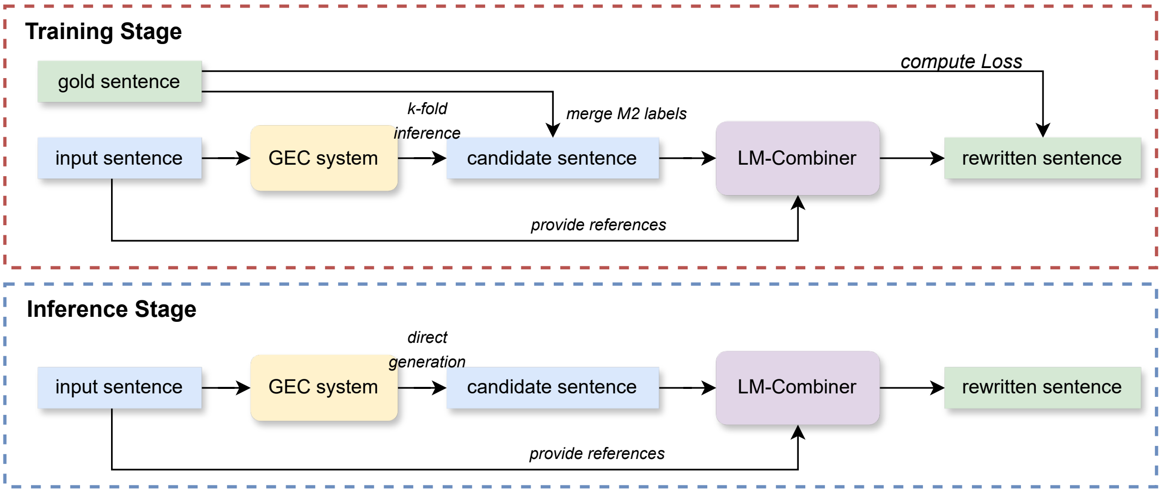

Figure 2: The flowchart of our error correction-rewriting framework. In the training phase, we construct candidate sentences containing GEC systems over-correction by k-fold cross inference and gold labels merging (see Section 2.3 for details). Then, we train the model to generate gold sentences based on the original and candidate sentences (Section 2.2). In the inference phase, LM-Combiner directly rewrites the system output based on the original sentence.

2.1. Causal LM For CGEC

The current Seq2Seq-based GEC models are mainly implemented by considering grammatical error correction as a neural machine translation task (Junczys-Dowmunt et al., 2018). Therefore it is natural to use models with encoder-decoder architecture (Bart (Lewis et al., 2019), T5 (Raffel et al., 2020), etc.) to synthesize the capabilities of NLU and NLG for text error detection and correction. Recently, many causal LM-based models (Brown et al., 2020; Wei et al., 2021; Touvron et al., 2023) with large-scale corpora and parameters have achieved excellent results on various natural language processing tasks including CGEC. It is meaningful to explore the application of relatively small-scale causal LMs like GPT2 (Radford et al., 2019) to CGEC task.

For the CGEC task, one of the most obvious ways to use causal LMs is letting the model continue to write the modification result based on the original sentence input. The inputs to the model during the training phase S can be formulated as:

$$S = \text{<sos>}X_1X_2...X_m\text{<sep>}Y_1Y_2...Y_n \qquad (1)$$

where X represents the sentence to be corrected of length m and Y represents the correct sentence of length n. <sos> represents the start of generation, and <sep> marks the completion of input and prompts the model to start generating results. The training labels are obtained by shifting the input as in the traditional LM task, and in order to ensure that the model learns to correct errors, the final training objective of the model is the loss of the correct sentence part, which can be formulated as:

$$\mathcal{L}_{Causal} = \sum_{k=i}^{j} -log(P(t_k|t_0t_1...t_{k-1};\theta)) \qquad (2)$$

where θ is the set of parameters of the language model, i represents the start index of the correct sentence Y, j represents the end index of the correct Y, and t_i represents the ith token in the model inputs like Equation 1. Although the experiments in table 1 show that the causal LM lacks the ability to correct errors on CGEC compared to the traditional Bart model, its higher precision rate inspires us to employ it as a rewriting model to alleviate the over-correction problem.

2.2. LM-Combiner Model

Based on the performance of causal LM on the CGEC dataset, we propose the text rewriting model LM-Combiner to deal with the over-correction of the original GEC system. As shown in figure 3, LM-Combiner takes the original sentence and the potentially overcorrected candidate sentences as inputs and directly generates the rewritten sentence as the final output of the GEC system. The candidate sentences are the outputs of the GEC system, and this method can be regarded as a kind of soft ensemble of the original sentences and the output sentences of a single model. We first describe the details of LM-Combiner at the model level in this section, and the specific training data construction methods are presented in Section 2.3. Unlike model ensemble methods based on PPL, LM-Combiner is trained to generate rewritten correct sentences directly from contextual inputs (inputs and outputs of the GEC system). Similar to Section 2.1, we adopt causal LM as the backbone of our approach. The inputs to the model S can be formulated as:

$$S = \text{<sos>} X_{src} \text{<cat>} X_{candi} \text{<sep>} Y_{tgt} \quad (3)$$

where X_{src} represents the original input sentence, X_{candi} represents the error correction result of the existing model, and Y_{tgt} represents the correct gold sentence. The meaning of the special token is the same as in Equation 1, and <cat> is used as a split label between the original and candidate sentences. Like the normal GEC model, in the training phase LM-Combiner only calculates the loss of the correct sentence part, which can be formulated as:

$$\mathcal{L}_{Combiner} = \sum_{k=i}^{j} -log(P(t_k|t_0 t_1 ... t_{k-1}; \theta)) \quad (4)$$

where θ is the set of parameters of the language model, i and j are the start and end indices of the sentence Y_{tgt}.

The structure of the LM-Combiner is relatively simple and straightforward, and the key to this model's performance is the way in which the sentence X_{candi} is obtained during the training phase. The method works only if the X_{candi} conforms to the distribution during the testing phase that corrects a certain amount of error but has a partly over-correction problem.

2.3. Dataset Construction

Over-corrections obtaining The main objective in the data construction phase is generating candidate sentences containing right correction and over-correction for each parallel corpus sample. Due to data exposure, it is not possible to obtain high-quality over-correction cases by directly inferring on the corpus with a fully trained model. To address this issue, we propose a data construction method based on k-fold cross inference. The specific process is shown in Algorithm 1. Firstly, we randomly divide the training set into K copies. Subsequently, we use the model obtained by training on k-1 copies to infer partial candidate sentences on the remaining data. Eventually, with many iterations, we get the candidate sentences of the full training set that correspond to the same distribution as the testing phase. Specifically, for the FCGEC dataset, we find that setting k to 4 already achieves good results.

```
Algorithm 1: K-fold Cross Inference
  Input: $D = \{(x_1, y_1), ..., (x_n, y_n)\}$, where $x_i$
         is the original sentence with the error,
         $y_i$ is the corrected sentence. The
         hyper parameter $K$.
  Output: $D_{candi} =$
          $\{(x_1, z_1, y_1), ..., (x_n, z_n, y_n)\}$, where
          $z_i$ is the candidate sentence that
          contains corrective modifications
          and over-corrections.
  $D_{candi} \leftarrow \{\}$;
  Randomly divide $D$ into $K$ copies $D_{Split}$;
  foreach $D_i$ in $D_{Split}$ do
      $D_{train}$ = $D - D_i$;
      Train on $D_{train}$ to get the model $\theta_i$;
      Obtain the inference result $Z_i$ of the
        model $\theta_i$ on $D_i$;
      Merge $Z_i$ as candidate sentences with
        $D_i$ to get $D_{merge}$;
      $D_{candi} = D_{candi} \cup D_{merge}$;
  Return $D_{candi}$;
```

Gold Labels Merging With k-fold cross inference, we ensure that the model always infers on data not used for training. This allows us to obtain the same distribution of over-correction as in the test phase, but at the same time doesn't guarantee that the model corrects all errors. Because the training goal is correct sentences, if there is missing error correction in the candidate sentence it will make the rewriting model still have to learn a part of the error correction task. In order to be able to completely decouple the two tasks of error correction and rewriting, we add correct corrections to the candidate sentences through MaxMatch (Dahlmeier and Ng, 2012) (M2) labels, so that the rewriting model only needs to complete the task of filtering over corrections and correct corrections in the candidate sentences. Specifically, we integrate the M2 labels of the candidate and gold sentences, prioritize the labels of the gold sentence when indexing conflicts, and collaborate the final merged set of M2 labels to the original sentence to obtain the final candidate sentence.

Inference stage In the inference phase, we directly use the real output of the error correction system as candidate sentences. We expect that the LM-Combiner trained on the above data can compare the original and candidate sentences to filter over corrections and retain right corrections.

3. Experiment

3.1. Settings

Dataset Restricted by the lack of data, previous CGEC tasks mainly use labeled datasets collected from Chinese as a Foreign Language (CFL) learner sources. However, Tang et al. (2023) have discov-

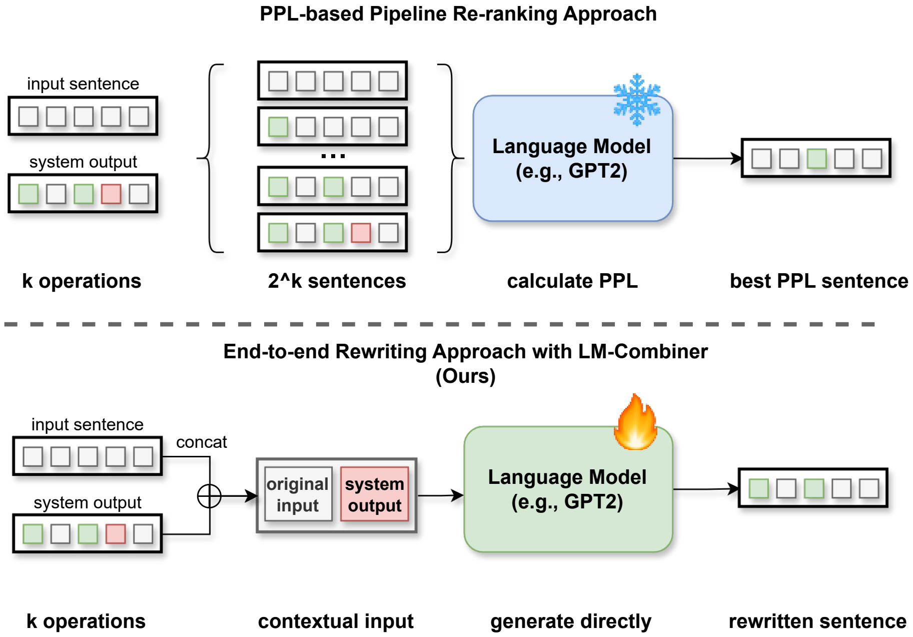

Figure 3: Comparison between the PPL-based approach and our approach. Both methods take the original sentence and the output of GEC system as input. In the figure, **gray** squares represent unmodified tokens, **green** squares represent rightly corrected tokens, and **red** squares represent overcorrected tokens. Existing work using PPL to rerank different candidate sentences can improve the precision rate of the system, but the judgment is not accurate enough because the LM is not trained on the domain data, leading to reduced recall. The LM-Combiner, trained on constructed candidate sentences, is better able to distinguish over-correction and generate results with higher recall end-to-end.

ered by way of human inspection that there is a distributional inconsistency between CFL corpus labeling distributions and native speakers, which may lead to unrealistic metrics. In recent years, more and more scholars have been working on the construction of CGEC datasets for native speaker corpora, Xu et al. (2022) provide a large-scale multi-reference corpus named FCGEC sourced from native speakers. Compared to CFL, the CGEC dataset from native speakers is more standardized and has higher annotation quality, but also includes more complex grammatical errors. We adopt the FCGEC dataset as the main dataset for our experiments, which contains 36,340 sentences of training data, 2,000 sentences of validation set, and 3,000 sentences of test set.

Evaluation metrics We follow Zhang et al. (2022a)'s setup by using character-level edit metrics to measure the error correction performance of each model. For the validation set experiments, we use the official evaluation tool ChERRANT [2] to evaluate the model based on correction span's P/R/F0.5. As for the test set, we obtain the same evaluation metrics by submitting the system results in CodaLab [3] online platform.

Model selection Reference to mainstream methods of CGEC, our main experiment adopts the model of Bart (Lewis et al., 2019) and GPT2 (Radford et al., 2019) architectures as the backbone network. We use the Chinese Bart model trained by Shao et al. (2021) and the series of Chinese GPT2 models trained by Zhao et al. (2019, 2023) to obtain a good performance on the CGEC task. Referring to other related work based on the Seq2Seq model (Zhang et al., 2022a; Li et al., 2023a), we chose Bart-Large and the equivalent scaled GPT2-medium as the backbone in the main experiment in order to make a fair comparison, and the LM-Combiner also uses the same settings as the GPT2 baseline.

Model hyperparameters As a general optimization method, in order to compare the enhancement effect more intuitively, we don't employ some common training techniques in the GEC field (e.g., Src-drop (Junczys-Dowmunt et al., 2018), label-smoothing (Szegedy et al., 2016), etc.) in the

[2] ChERRANT is a Chinese GEC evaluation tool that refers to ERRANT, the mainstream GEC evaluation tool in English.

[3] https://codalab.lisn.upsaclay.fr/competitions/8020

model training phase. For both models, we use the AdamW (Loshchilov and Hutter, 2017) optimizer with 5e-5 learning rate, and 32 batch size for training. We use the polynomial strategy as a warm-up strategy for learning rate. Considering the difference in model architectures, the maximum sentence lengths of the Bart and GPT2 models are 256 and 512. In the testing phase, both generative models inference using beam search with a beam size of 4.

3.2. Baseline Approaches

We select several common methods with Seq2Edit and Seq2Seq architectures as baseline models, and pick the one with the largest recall as the system output to validate the effectiveness of the rewriting model. Our adoption of Chinese GEC model is largely referenced by Zhang et al. (2022a); Xu et al. (2022)'s related work.

- **LaserTagger** (Malmi et al., 2019) is a text generation method based on editing operations that improves the inference speed and reduces the data requirements of the model for the text generation task.
- **PIE** (Awasthi et al., 2019) leverages the power of pre-trained models to efficiently correct grammatical errors through iterative edit tag prediction.
- **GECToR** (Omelianchuk et al., 2020) further refines the custom token-level edit tags to map more diverse errors.
- **STG** (Xu et al., 2022) completes complex grammatical error correction by pipelining three self-encoding models, Switch, Tagger, and Generator, and achieves the SOTA on the FCGEC dataset by jointly training three models.
- **Bart** (Lewis et al., 2019; Zhang et al., 2022a) model has achieved good results on the CGEC task with its denoising pre-training task, and can be used as a representative of the Seq2Seq model.
- **GPT2** (Radford et al., 2019) model is typically used for generative tasks, and we implemente a GPT model for CGEC as a baseline model following the methodology of Section 2.1.

As a post-processing method, our rewriting model can also be understood as an ensemble of the original sentence and the output of a single system. Although a single model can't be integrated using traditional voting ensemble methods, the fine-grained PPL-based model ensemble method proposed by Tang et al. (2023) can still be used as a baseline model for post-processing methods. Specifically, we replicate three different granularity ensemble approaches based on the same scale of GPT2.

Method	FCGEC-test		
	P	R	$F_{0.5}$
LaserTagger*	36.60	31.16	35.36
PIE*	29.15	29.77	29.27
GECToR (Chinese)*	30.68	21.65	28.32
STG*	48.19	37.14	45.48
Bart-Chinese-large	37.49	38.87	37.76
GPT2-medium	56.71	24.79	45.10
Bart-Chinese-large	37.49	38.87	37.76
+ Sentence-level	55.26	20.23	41.04
+ Edit-level	**58.22**	24.12	45.39
+ Edit-combination	58.16	25.63	46.38
+ LM-Combiner (Ours)	55.67	**39.04**	**51.30**

Table 1: Experimental results of our method on the FCGEC test set. Results with * are reported from the original paper (Xu et al., 2022). The first group indicates common Seq2Edit models, the second group indicates Seq2Seq models, and in the last group we choose the highest recall Bart model as a baseline and list some LM-based post-processing methods.

- **Sentence-level** makes a judgment directly from the PPL of the original and output sentences, and only retains sentences with lower perplexity.
- **Edit-level** makes a judgment based on the impact of each editing operation on the PPL of the original sentence, and retains only those operations that reduce the PPL of the original sentence.
- **Edit-combination** permutes all the editing operations and selects the sentence with the lowest PPL among them as the final output as shown in Figure 3.

3.3. Main Results

Table 1 shows the comparison of the performance among different models on the FCGEC test set. In order to maximize the verification of the performance of the LM-Combiner, we chose the output of the highest recall Bart model as the rewriting input. As shown in the table, through the rewriting of our LM-Combiner model, we make the output of the original error-correction system substantially improve the precision by 18.2 points while the recall remains basically unchanged, and the $F_{0.5}$ metric improves by 5.8 points compared to the SOTA model.

Compared to the PPL-based methods, LM-Combiner does better in recall retention due to

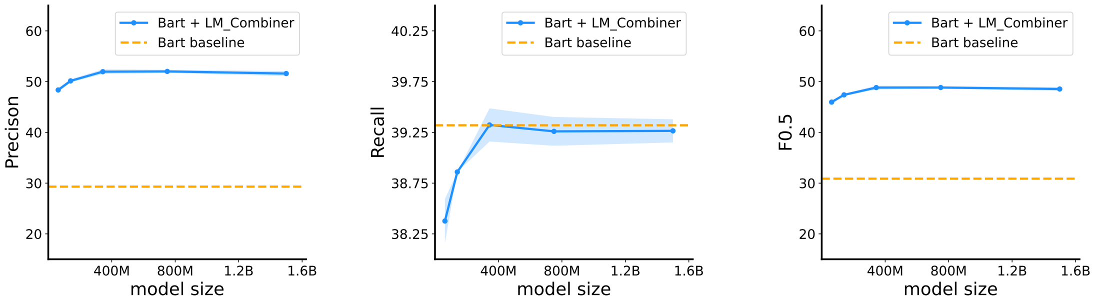

Figure 4: The effect of model size for LM-Combiner on FCGEC valid. The Bart baseline is the system metric without LM-Combiner rewriting. For a more accurate evaluation, we average the results of 5 experiments for each size of the model, and the floating part of the figure shows the standard deviation of the metrics.

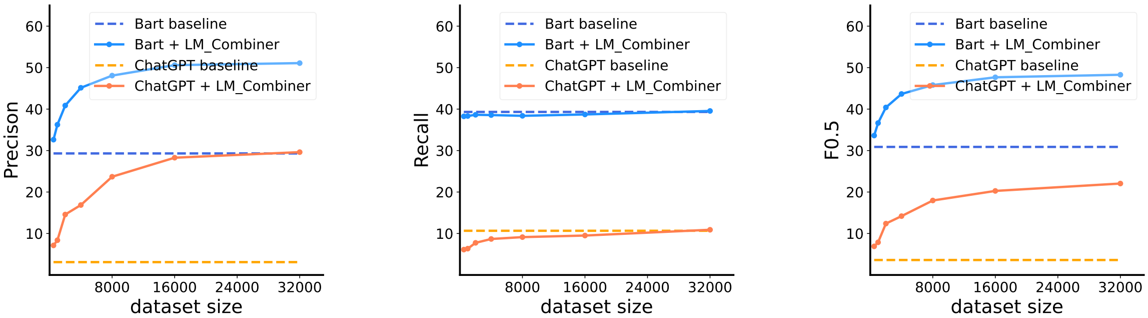

Figure 5: The effect of training dataset size for LM-Combiner on FCGEC valid. The baseline method represents the metrics for each system without the use of LM-Combiner.

the fine over-correction dataset construction. Although both using GPT2 as the backbone network, the PPL-based approach suffers from the problem of inconsistency between the domains of the pre-trained corpus and the target corpus, which makes the model filter out too much right correction in the re-ranking phase. Conversely, by constructing a real over-correction dataset under the target domain for training, the LM-Combiner is able to better learn the relationship between over-correction and right correction in the target domain, and thus improves the precision with essentially no decrease in recall.

4. Analysis

In this section we will validate and analyze the details of the LM-Combiner through experiments.

4.1. Effect of Model Scale

By decoupling the CGEC task, the LM-Combiner model only needs to complete the rewriting task without considering the performance of error correction. For the simpler rewriting task, we wonder if its rewriting performance is strongly correlated with the scale of the model, thus we use five scales of GPT2, small, base, medium, large, and xlarge, respectively, as the backbone network of the LM-Combiner for the experiments.

As shown in Figure 4, all scales of rewriting model can relatively improve the precision of the GEC system. The reduction in error recall from rewriting the model becomes smaller and smaller as the model size increases. In addition to this, we can find that a small-level 62M model can still improve precision by about 18 points compared to the baseline model and essentially preserve the recall of the original system. For the insignificant change in rewriting performance with model scale, we analyse that this is because the difficulty of the decoupled rewriting task is lower compared to the error correction task, which makes it possible for small models to perform well.

4.2. Effect of Data Quantity

According to the data construction method in Section 2.3, we have obtained the over-correction training set totaling 36,340 sentences of the entire FCGEC training data. However, in practice it is still a large number in a new domain. We want to know what amount of parallel corpus will enable us to train a rewriting model works reasonably well through data construction. Thus, we randomly sample subsets of different sizes from the constructed training set to validate the effect of rewriting the model.

Besides that, Li et al. (2023b); Fang et al. (2023) have evaluated the effectiveness of LLMs (e.g., ChatGPT) on the CGEC task, and the experiments show that there is also a large amount of over-correction in LLMs using the zero-shot and few-shot methods. Therefore, we follow Fang et al. (2023)'s approach and also obtain the results of the ChatGPT model on FCFEC for rewriting the model's training as a way to validate the ability of the LM-Combiner for black-box correction systems. Since there is no data leakage, we directly use the error correction results of ChatGPT as candidate sentences instead of the cross inference method in Section 2.3.

The experimental results are shown in Figure 5, LM-Combiner trained at all scales of data amounts is able to alleviate the over-correction problem of the original system to varying degrees. In particular, thousands of domain training corpora are sufficient to obtain a rewriting model that performs well, both for Bart model and the ChatGPT. Consistent with Li et al. (2023b)'s evaluation, ChatGPT doesn't perform well on the native speaker CGEC task, with metrics even lower than the Bart baseline model. Nevertheless, LM-Combiner can still be considered as a low-cost post-processing model, which can effectively relieve over-correction of various GEC systems (including the black-box ChatGPT) on domain-specific datasets.

4.3. Importance of Gold Labels Merging

As described in Section 2.3, after acquiring the overcorrected data, we merge the gold labels with the overcorrected labels based on the M2 labels as a way to completely decouple the error correction task. To verify the effect of label merging, we conducted experiments on the original training set and the training set with gold labels merging, respectively. The experimental results are shown in Table 2, where the gold label merging enables LM-Combiner to learn the rewriting task better and retain a higher recall. It can be said that fully decoupling correction and rewriting tasks by gold labels merging is the key for LM-Combiner to maintain high recall.

Method	FCGEC-valid		
	P	R	$F_{0.5}$
Bart-Chinese-large	29.31	39.32	30.88
+LM-C wo merging	**54.02**	36.07	49.13
+LM-C w merging	53.56	**39.25**	**49.92**

Table 2: Experimental results on the effectiveness of gold label merging. LM-C represents the LM-Combiner model, and merging represents the gold labels merging operation.

5. Related Work

Compared to the English GEC, the Chinese GEC is just getting started (Tang et al., 2023). Early CGEC tasks are mainly researched in the field of non-native language learning, which has a large error rate, and many CFL datasets such as Lang8, CGED (Rao et al., 2020), and NLPCC18 (Zhao et al., 2018) are proposed. On this basis, Zhang et al. (2022) sample and organise the annotation of several CFL datasets, correct the existing annotation problems in them, and propose the MuCGEC dataset with multi-source references. Recently, more and more scholars (Xu et al., 2022; Ma et al., 2022) have noticed the problems with CFL datasets and propose a series of datasets based on native speakers' grammatical errors, posing a greater challenge to the CGEC task.

The CGEC task has received increasing attention in recent years. Responding to the lack of data, Zhao and Wang (2020) propose a dynamic mask strategy for data augmentation and improve the robustness of the model. Yue et al. (2022) generate high-quality grammatical errors to complete the data augmentation by conditional non-autoregressive error generation model. In terms of model architecture, Zhang et al. (2022b) extract the syntactic hidden representation by graph convolutional neural network and incorporate the syntactic information into the GEC system to further improve the error correction performance. Li et al. (2023a) fuse the models of the two paradigms in the form of templates and improve the precision of the Seq2Seq model with the help of Seq2Edit model through the detection and correction framework.

Previous researchers have also attempted to explore the potential of causal LMs in GEC tasks. Yasunaga et al. (2021) determine the grammatical correctness of a sentence with the help of the PPL of PLMs, and implements a unsupervised GEC framework by assuming that the sentence with the smallest perplexity within a particular set is the correct sentence. Similarly, Tang et al. (2023) use the PPL of pre-trained models as a model ensemble method to re-rank the outputs of multiple models.

The large language model represented by ChatGPT (Ouyang et al., 2022) is developing rapidly, and there have been some recent related evaluation work (Li et al., 2023b; Fang et al., 2023) on LLM on CGEC tasks. The results indicate that LLM suffers from serious over-correction problems. Recently Vernikos et al. (2023) use the T5 model for soft aggregation of multiple outputs from LLM, but there are still some common problems of ensemble methods. In view of this, LM-Combiner is a good solution to alleviate the over-correction problem by directly rewriting individual system outputs without the need for model ensemble.

6. Conclusion

In this paper, we propose LM-Combiner, a general rewriting model based on a causal language model, capable of mitigating the problem of over-correction based on the original sentences and single system outputs. We also propose k-fold cross inference to enable the construction of domain-specific over-correction data for LM-Combiner training. Experiments show that the proposed method can effectively improve the system precision while ensuring the recall rate, and it provides a low-cost over-correction solution for existing GEC systems.

7. Acknowledgement

We gratefully acknowledge the support of the National Natural Science Foundation of China (NSFC) via grant 62236004 and 62206078.

8. Bibliographical References

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. *arXiv preprint arXiv:1910.02893*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2022. Grammatical error correction: A survey of the state of the art. *Computational Linguistics*, pages 1–59.

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.

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*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. *arXiv preprint arXiv:1804.05940*.

Masahiro Kaneko, Sho Takase, Ayana Niwa, and Naoaki Okazaki. 2022. Interpretability for language learners using example-based grammatical error correction. *arXiv preprint arXiv:2203.07085*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chen Li, Junpei Zhou, Zuyi Bao, Hengyou Liu, Guangwei Xu, and Linlin Li. 2018. A hybrid system for chinese grammatical error diagnosis and correction. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 60–69.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10974–10982.

Yinghao Li, Xuebo Liu, Shuo Wang, Peiyuan Gong, Derek F Wong, Yang Gao, He-Yan 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.

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.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. *arXiv preprint arXiv:1909.01187*.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. Gector–grammatical error correction: tag, not rewrite. *arXiv preprint arXiv:2005.12592*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Gaoqi Rao, Erhong Yang, and Baolin Zhang. 2020. Overview of nlptea-2020 shared task for chinese grammatical error diagnosis. In *Proceedings of the 6th workshop on natural language processing techniques for educational applications*, pages 25–35.

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*.

Bo Sun, Baoxin Wang, Yixuan Wang, Wanxiang Che, Dayong Wu, Shijin Wang, and Ting Liu. 2023. Csed: A chinese semantic error diagnosis corpus. *arXiv preprint arXiv:2305.05183*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Chenming Tang, Xiuyu Wu, and Yunfang Wu. 2023. Are pre-trained language models useful for model ensemble in chinese grammatical error correction? *arXiv preprint arXiv:2305.15183*.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Giorgos Vernikos, Arthur Bražinskas, Jakub Adamek, Jonathan Mallinson, Aliaksei Severyn, and Eric Malmi. 2023. Small language models improve giants by rewriting their outputs. *arXiv preprint arXiv:2305.13514*.

Yu Wang, Yuelin Wang, Jie Liu, and Zhuo Liu. 2020. A comprehensive survey of grammar error correction. *arXiv preprint arXiv:2005.06600*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Xiuyu Wu and Yunfang Wu. 2022. From spelling to grammar: A new framework for chinese grammatical error correction. *arXiv preprint arXiv:2211.01625*.

Lvxiaowei Xu, Jianwang Wu, Jiawei Peng, Jiayu Fu, and Ming Cai. 2022. Fcgec: Fine-grained corpus for chinese grammatical error correction. *arXiv preprint arXiv:2210.12364*.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2021. Lm-critic: Language models for unsupervised grammatical error correction. *arXiv preprint arXiv:2109.06822*.

Tianchi Yue, Shulin Liu, Huihui Cai, Tao Yang, Shengkang Song, and Tinghao Yu. 2022. Improving chinese grammatical error detection via data augmentation by conditional error generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2966–2975.

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

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. *arXiv preprint arXiv:2204.10994*.

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.

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: 7th CCF International Conference, NLPCC 2018, Hohhot, China, August 26–30, 2018, Proceedings, Part II 7*, pages 439–445. Springer.

Zewei Zhao and Houfeng Wang. 2020. Maskgec: Improving neural grammatical error correction via dynamic masking. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1226–1233.

Zhe Zhao, Hui Chen, Jinbin Zhang, Xin Zhao, Tao Liu, Wei Lu, Xi Chen, Haotang Deng, Qi Ju, and Xiaoyong Du. 2019. Uer: An open-source toolkit for pre-training models. *EMNLP-IJCNLP 2019*, page 241.

Zhe Zhao, Yudong Li, Cheng Hou, Jing Zhao, et al. 2023. Tencentpretrain: A scalable and flexible toolkit for pre-training models of different modalities. *ACL 2023*, page 217.

9. Language Resource References

Ma, Shirong and Li, Yinghui and Sun, Rongyi and Zhou, Qingyu and Huang, Shulin and Zhang, Ding and Yangning, Li and Liu, Ruiyang and Li, Zhongli and Cao, Yunbo and others. 2022. *Linguistic rules-based corpus generation for native chinese grammatical error correction*. PID https://github.com/masr2000/CLG-CGEC.

Xu, Lvxiaowei and Wu, Jianwang and Peng, Jiawei and Fu, Jiayu and Cai, Ming. 2022. *FCGEC: Fine-Grained Corpus for Chinese Grammatical Error Correction*. PID https://github.com/xlxwalex/FCGEC.

Zhang, Yue and Li, Zhenghua and Bao, Zuyi and Li, Jiacheng and Zhang, Bo and Li, Chen and Huang, Fei and Zhang, Min. 2022. *Mucgec: a multi-reference multi-source evaluation dataset for chinese grammatical error correction*. PID https://github.com/HillZhang1999/MuCGEC.

Zhao, Yuanyuan and Jiang, Nan and Sun, Weiwei and Wan, Xiaojun. 2018. *Overview of the nlpcc 2018 shared task: Grammatical error correction*. Springer. PID https://github.com/zhaoyyoo/NLPCC2018$_{GEC}$.