applied sciences

MDPI

Article

A Chinese Grammatical Error Correction Method Based on Iterative Training and Sequence Tagging

Hailan Kuang, Kewen Wu *, Xiaolin Ma * and Xinhua Liu

Hubei Key Laboratory of Broadband Wireless Communication and Sensor Networks, School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China; kuanghailan@whut.edu.cn (H.K.); liuxinhua@whut.edu.cn (X.L.)

* Correspondence: wkw@whut.edu.cn (K.W.); maxiaolin0615@whut.edu.cn (X.M.)

Abstract: Chinese grammatical error correction (GEC) is under continuous development and improvement, and this is a challenging task in the field of natural language processing due to the high complexity and flexibility of Chinese grammar. Nowadays, the iterative sequence tagging approach is widely applied to Chinese GEC tasks because it has a faster inference speed than sequence generation approaches. However, the training phase of the iterative sequence tagging approach uses sentences for only one round, while the inference phase is an iterative process. This makes the model focus only on the current sentence's current error correction results rather than considering the results after multiple rounds of correction. In order to address this problem of mismatch between the training and inference processes, we propose a Chinese GEC method based on iterative training and sequence tagging (CGEC-IT). First, in the iterative training phase, we dynamically generate the target tags for each round by using the final target sentences and the input sentences of the current round. The final loss is the average of each round's loss. Next, by adding conditional random fields for sequence labeling, we ensure that the model pays more attention to the overall labeling results. In addition, we use the focal loss to solve the problem of category imbalance caused by the fact that most words in text error correction do not need error correction. Furthermore, the experiments on NLPCC 2018 Task 2 show that our method outperforms prior work by up to 2% on the F0.5 score, which verifies the efficiency of iterative training on the Chinese GEC model.

Keywords: Chinese grammatical error correction; sequence tagging; iterative training; conditional random field; focal loss

Citation: Kuang, H.; Wu, K.; Ma, X.; Liu, X. A Chinese Grammatical Error Correction Method Based on Iterative Training and Sequence Tagging. *Appl. Sci.* **2022**, *12*, 4364. https://doi.org/10.3390/app12094364

Academic Editors: Katarzyna Antosz, Valentino Santucci, Jose Machado, Yi Ren, Rochdi El Abdi, Dariusz Mazurkiewicz, Marina Ranga, Pierluigi Rea, Vijaya Kumar Manupati, Emilia Villani and Erika Ottaviano

Received: 10 February 2022
Accepted: 22 April 2022
Published: 26 April 2022

1. Introduction

In recent years, rapid development in China has encouraged many people to learn Chinese. The number of foreigners who regard Chinese as their second language is constantly increasing. Because the grammatical structure and usage habits of different languages are very different, learners of Chinese are prone to grammatical errors when writing Chinese. In addition, with the booming of the Internet, a huge amount of textual information has appeared in our lives in the form of news reports, blogs, emails and chat logs. Inevitably, there are many grammatically incorrect texts due to the lack of secondary review. Moreover, text generated by methods such as machine translation, speech recognition and image text recognition may contain errors. These Chinese grammatical errors not only seriously affect the user experience but also affect the computer's understanding of the semantics, causing the computer to make incorrect judgments. Therefore, the question of how to detect and correct grammatical errors in Chinese texts deserves further research. Grammatical error correction (GEC) refers to generating grammatically correct sentences from incorrect sentences, thereby enhancing the user experience and reducing the cost of manual verification. Although GEC can be used in many languages, in this paper, we focus only on Chinese GEC.

There are two typical types of grammar error correction methods used in this academic field: sequence generation and sequence tagging. Because the sequence generation method is similar to neural machine translation (NMT), the authors in [1] used the NMT approach based on sequence-to-sequence (Seq2Seq) models to address the GEC problem and made some progress. However, the sequence generation method requires sequential decoding, resulting in low efficiency. The sequence tagging approach was first applied to the GEC task in [2]. Sequence tagging methods predict the corresponding edit operation, while sequence generation methods predict the correct sentence directly. For this reason, the sequence tagging method allows us to better observe and control the correction process. Compared with the sequence generation method, the sequence tagging method not only has stronger interpretability and controllability but also has a faster processing speed.

The sequence tagging model can only tag each token once, so only one edit operation can be performed on each token by the model. However, many errors cannot be corrected in a single edit. The current state-of-the-art method GECToR [3] uses multiple iterations to tag and correct sentences, correcting only the more significant errors in each round. However, each iteration relies on the error correction results of the previous round, which leads to exposure bias. To address the problem, inspired by Google's TeaForN [4], we propose a Chinese GEC method based on iterative training and sequence tagging (CGEC-IT) to extend the GECToR model, to improve the effectiveness of the Chinese GEC task.

Our contributions are summarized as follows:

1. We propose a Chinese GEC method based on iterative training, which aligns the model training process with the inference process to solve the problem of exposure bias caused by using iterative inference.
2. We design a sequence labeling method that combines direct prediction with prediction using conditional random fields (CRF) [5], allowing the model to focus more on the tagging results of the whole sequence.
3. We implement a focal loss [6] penalty strategy for the loss function to alleviate the class imbalance problem in the classification problem.
4. Experiments based on NLPCC 2018 Task 2 [7] demonstrate the robustness and effectiveness of our approach.

2. Related Work

Before the rise of deep learning, almost all Chinese GEC researchers used methods based on rules and statistics [8,9]. Most of these methods extracted features manually, making it easy to produce deviation. The effect of these methods was not ideal, due to the complexity of Chinese grammar. As deep learning began to dominate in various fields, text error correction gradually began to be combined with deep learning. In 2014, Yu et al. [10] organized a shared task for Chinese grammatical error diagnosis (CGED), which greatly assisted the research on Chinese GEC. There are already many ways to deal with Chinese GEC issues. At present, the most typical methods for dealing with Chinese GEC are sequence generation methods and sequence tagging methods.

The sequence generation approach directly generates the correct sentence from the original sentence, similarly to a machine translation task. Hu et al. [11] used a Seq2Seq model for Chinese GEC and pre-trained the model using pseudo data. Since there may be multiple errors in a sentence which cannot be corrected by one round of editing operations, Lichtarge et al. [12] proposed an iterative error correction strategy. This method takes the results with the highest confidence generated in the first round as the input of the second round, and repeats the process until there is no need for further correction or until the maximum number of cycles is reached. Since there is a large intersection between the original text and the target text of the Seq2Seq model, generating a large number of correct words will waste resource. Ren et al. [13] introduced the ConvSeq2Seq [14] model for the Chinese GEC task, which enabled the parallel execution of the decoding process and improved the decoding speed. Their single model achieved a 29.0% F0.5 measure on the NLPCC 2018 test set. Chen et al. [15] divided the GEC task into two

subtasks: error span detection (ESD) and error span correction (ESC). ESD uses sequence tagging binary classification to identify the location of the incorrect character, and then ESC uses the Seq2Seq model to correct errors only for the wrong locations. The inference time for this method is only half of that for Seq2Seq used directly, and the F0.5 measure is 28.4% on the NLPCC 2018 test set. With the rise of pre-trained language models, the pre-trained model BERT [16] achieved better results in multiple NLP tasks. Zhu et al. [17] proposed a BERT-fused method that uses BERT to extract the features of input sequences, and then fuses the representation with each encoder and decoder layer of the NMT model through an attention mechanism. Wang et al. [18] experimented on the Chinese GEC task using BERT-fused and used BERT to initialize the Transformer [19] encoder and randomly initialize the decoder, achieving better results than BERT-fused on the NLPCC 2018 dataset. Kaneko et al. [20] fine-tuned BERT using the GEC corpus and then trained the BERT-based Seq2Seq model with the original input and the BERT output. Li and Shi [21] proposed an end-to-end non-autoregressive sequence prediction model for Chinese GEC, to solve the problem that the label space of the sequence labeling method is large and requires multiple rounds of decoding. They achieved good results on a fixed-length dataset.

GEC methods based on sequence tagging are intended to solve the problems of slow inference speed and poor controllability of generation. In GEC, the intersection of the error text and the correct text is particularly large, so it can be related to the way that the string edit distance is related to how the incorrect text can be transformed into the correct text with fewer token edit operations. Awasthi et al. [2] proposed a parallel iterative editing (PIE) model which defines editing operations such as copy, append, delete, replace and case-change. By comparing the error sentence with the correct sentence, the editing method is obtained, and then each word is marked, iteratively refining the prediction to capture dependencies. This method greatly improves the error correction speed, but it has not been applied to Chinese to date. Malmi et al. [22] proposed a sequence tagging error correction method, LaserTagger, which set up two labels for retain and delete, and added phrase or word labels based on these two labels. The entire prediction space was twice the size of the set thesaurus. The model uses BERT as the encoder and Transformer as the decoder for sequence tagging. However, LaserTagger does not seem to work well in Chinese; it only achieved a 19.9% F0.5 measure on the NLPCC 2018 test set. Omekianchuk et al. [3] proposed the GECToR model and defined more than 5000 tags for grammatical operations, including basic tags such as KEEP, DELETE, 1167 APPEND_X and 3802 REPLACE_X, as well as 29 new tags for grammatical changes in English, such as singular and plural conversion, tense conversion and so on. Firstly, each token of the wrong sentence is labeled with a defined label and then trained by using the sequence tagging model based on BERT, and the iterative inference is used for multiple rounds of error correction. This has achieved the best effect in English GEC. In the NLPTEA 2020 [23] competition, Liang et al. [24] applied the GECToR model to Chinese GEC for the first time. They removed the 29 grammar change tags for English and constructed 8772 basic tags for common Chinese characters. Their joint model achieved the highest F1 value on the NLPTEA 2020 error correction top1 subtask. We experimented with the GECToR model on the NLPCC 2018 test set and achieved a 29.8% F0.5 measure.

Due to the high cost of GEC data annotation, it is difficult to obtain large-scale training data. Many studies related to grammar error correction have focused on how to automatically construct large-scale training data. It has been shown that the use of data augmentation methods can effectively improve the performance of grammar error correction models. Zhao and Wang [25] proposed a Chinese error correction method called MaskGEC, which uses a dynamic random mask to mask and replace some words, to expand the dataset. Surprisingly, the MaskGEC achieved a good score of 36.9% on the F0.5 measure on the NLPCC 2018 test dataset. Tang et al. [26] proposed a data augmentation method that fuses word and character levels. They used the expanded six million sentence pairs to train the model and achieved a 39.1% F0.5 measure on the NLPCC 2018 test set. Because our focus was on the training strategy and model improvement, we expected the

model to achieve better results on the same sized dataset. Therefore, no comparisons are made with methods that use data augmentation.

Among the above methods, the sequence generation methods such as Seq2Seq, ConvSeq2Seq, ESD and BERT-fused can be applied without modification to other languages. Only the training data of the corresponding language needs to be used for training. In contrast, sequence tagging methods such as PIE, LaserTagger, GECToR, etc. must set specific edit tags for other languages. Good or bad edit tag setting has a large impact on the performance of the model, so the settings cannot be directly applied to other languages. For the NLPCC 2018 dataset, the best-performing sequence generation method is BERT-fused, with a 29.9% F0.5 value. GECToR is the best-performing sequence labeling method, with a 29.8% F0.5 value, and the inference speed of GECToR is much faster than that of BERT-fused.

3. Methodology

CGEC-IT extends the GEC model based on iterative sequence tagging. Its framework is shown in Figure 1. The model is divided into a Transformer encoder, a grammatical error detector and a grammatical error corrector. The input of the Transformer encoder is an incorrect sentence $X = (x_1, x_2, \ldots, x_N)$, where x_i represents the i-th token in the sentence and N is the length of X. The Transformer encoder converts each token in the input sentence to a vector using BERT, to obtain a bidirectional semantic representation of the text. Both the grammatical error detector and the grammatical error corrector are sequence tagging models, whose input is the output of the Transformer encoder. They differ in the size of the tag set: the grammatical error detector has only two tags, tag correct or tag incorrect, whereas the size of the tag set of the grammatical error corrector depends on the number of defined edit operations.

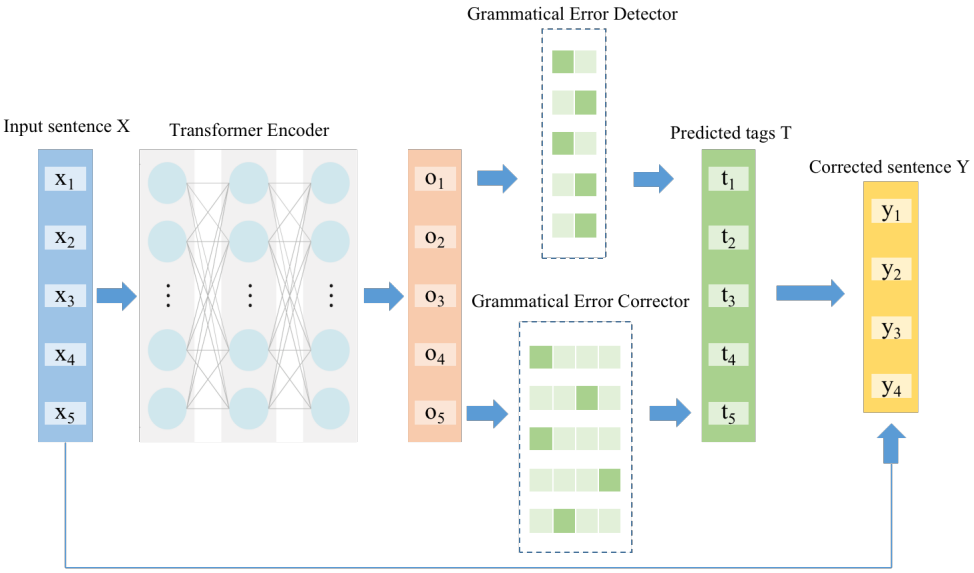

Figure 1. An overview of our framework.

3.1. Chinese Correct Tag Set

The tagging error correction approach requires careful analysis and processing of the data and requires a high level of tag definition. The key to this approach is to design suitable output tags that indicate the differences between the source and target sentences. The number of Chinese words is over 400,000, of which there are about 50,000 in common use. However, there are only about 7000 commonly used characters in Chinese, so the tag sets used by the tagging error correction model for Chinese use Chinese characters rather than Chinese words. In this study, we first extracted the minimum edits of characters based

on the modified Levenshtein distances, and then converted these edits to tags. There are four main types of tags in the grammatical error corrector:

- KEEP indicates that the character is correct and does not need to be modified.
- DELETE indicates that the character is redundant and should be deleted.
- APPEND_X indicates that the current character is correct, but a character X needs to be added after the current character. In this study, 4177 values of X are used in APPEND_X tags.
- REPLACE_X indicates that the current character is wrong and should be replaced with the character X. In this study, 3283 values of X are used in REPLACE_X tags.

The error correction tag set consists of 7462 tags, forming a large editing space. The error checking tag set consists of just two tags, correct and incorrect, where correct corresponds to KEEP tags and incorrect corresponds to tags other than KEEP.

3.2. Deep Bidirectional Transformer Encoder

Transformer layers (Vaswani et al., 2017) are particularly suitable for bidirectional semantic modeling. In this study, we use multilayer Transformer layers to model the bidirectional semantics of the input sequence and a pre-trained BERT model to initialize the multilayer Transformer. For each character of the source sentence $X = [x_1, \ldots, x_n]$, the initial input of the original semantic representation according to the input requirements of BERT is as follows:

$$x_i = \left[emb_{word[i]}; emb_{pos[i]}; emb_{seg[i]}\right] \tag{1}$$

where $emb_{word[i]}$ is the word embedding, $emb_{pos[i]}$ is the positional embedding and $emb_{seg[i]}$ is the segment embedding. Here, ";" is the vector splicing operation.

The output of the model is a sequence of word vectors whose length is fixed and equal. Furthermore, the output features are used in the grammatical error detector and grammatical error corrector. This can be formalized as follows:

$$O = (o_1, o_2, \ldots, o_N) = BERT(x_1, x_2, \ldots x_N) \tag{2}$$

3.3. Sequence Tagging Method Based on CRF

Both the grammatical error detector and the grammatical error corrector serve to label each token according to the feature vector output of the Transformer encoder. The sequence tagging problem can be simply understood as one of classifying each character, and the number of categories is the size of the tag set. The only difference between the grammatical error detector and the corrector is that the set of tags differs between the two. The grammatical error detector uses only two tags, while the grammatical error corrector uses the 7462 tags mentioned in Section 3.1. Since this is a classification task, it is easy to consider classifying directly, after encoding with a fully-connected layer and then activating it with the softmax function. When the length of the sequence is N, and the size of the label set is L, the formula is as follows:

$$h_i = o_i^{\mathsf{T}} W + b \tag{3}$$

$$P_{\mathsf{sm}}(t_i) = softmax(h_i) \tag{4}$$

where W is an $N \times L$ parameter matrix and b is a parameter matrix of size L. P_{sm} is a probability distribution, based on which we can obtain the tagging result T for each character:

$$\begin{aligned} T &= (t'_1, t'_2, \ldots, t'_N) \\ t'_i &= argmax(P_{\mathsf{sm}}(t_i)) \end{aligned} \tag{5}$$

However there are associations between the output tag sequences. Label-by-label softmax does not take into account such output-level contextual associations but instead puts these associations into the coding level for learning. Particularly when there are multiple errors in a sentence, it is necessary to comprehensively consider the correction

results of each point, to achieve better results. If an input sentence has N tokens and each token has L possibilities for its label, then there are theoretically L^N different outputs. As Figure 2 shows, the goal is to select the correct one from L^N paths, which means that if it were considered as a classification problem, it would be a problem of selecting one of the L^N classes! It is natural to consider using linear-chain CRF [5] to ensure the model takes this overall association into account.

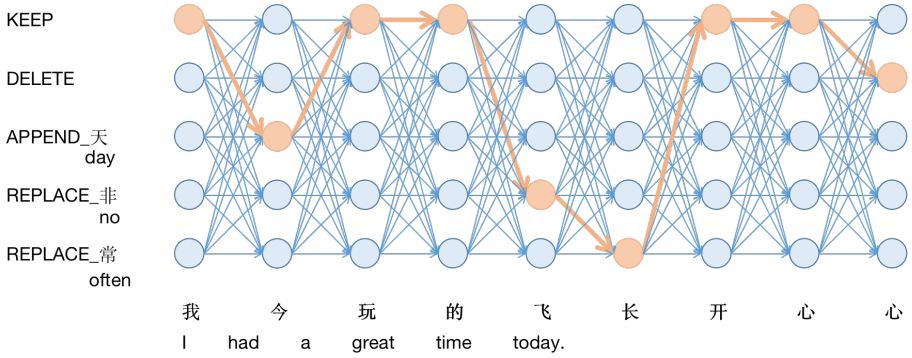

Figure 2. Each point represents a label possibility, the line between the points indicates the association between the labels and each of the labeled outcomes corresponds to a complete path on the diagram.

In a CRF network with input sequence $H = (h_1, \ldots, h_N)$ and target sequence $T = (t_1, \ldots, t_N)$, the probability distribution of linear-chain CRF is as follows:

$$P_{\text{crf}}(T \mid H) = \frac{1}{Z(H)} exp\left(s(t_1; H) + \sum_{i=1}^{N-1} [f(t_i, t_{i+1}) + s(t_{i+1}; H)]\right) \tag{6}$$

where $Z(H)$ is the normalization factor and $s(ti; H)$ is the fraction of t at position i, which can be obtained from Equations (3) and (4). In addition, $f(ti, ti+1)$ is a parameter matrix of size $L \times L$ to be trained and L is the size of the tag set. In order to solve the problem that calculating all paths requires more resources, we refer to [27] and improve the computational efficiency by selecting the k tags with the highest probability at each position for calculation instead of all the tags in the tag set.

3.4. Training with Joint Focal Loss

Using softmax direct prediction gives the best tagging result for each token, while CRF is more concerned with the quality of the overall tagging result. To improve the capability of the model, direct prediction and CRF are combined for better results. In the grammatical error detector, there are only two tags in the tag set: correct and incorrect. The activity diagram of the loss function is shown in Figure 3. We employ the cross-entropy error as the loss function for direct prediction:

$$dL_{\text{sm}} = \frac{1}{N} \sum_{i=1}^{N} -[t_i \cdot \log(p_i) + (1 - t_i) \cdot \log(1 - p_i)] \tag{7}$$

where p_i is the probability that the token at position i is tagged as correct. In addition, t_i is the tag of the token at position i, which has a value of 1 when the label is correct and has a value of 0 when the label is incorrect. The grammatical error corrector is similar to the detector, except that there are more tags in the corrector and we also use a cross-entropy loss function for direct prediction:

$$cL_{\text{sm}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{L} t_{ic} \log(p_{ic}) \tag{8}$$

where L is the number of tags and t_ic is 1 if the true tag of the i-th token is equal to the c-th tag in the tag set and 0 otherwise. In addition, p_{ic} is the predicted probability that the i-th token is labeled as the c-th tag in the tag set.

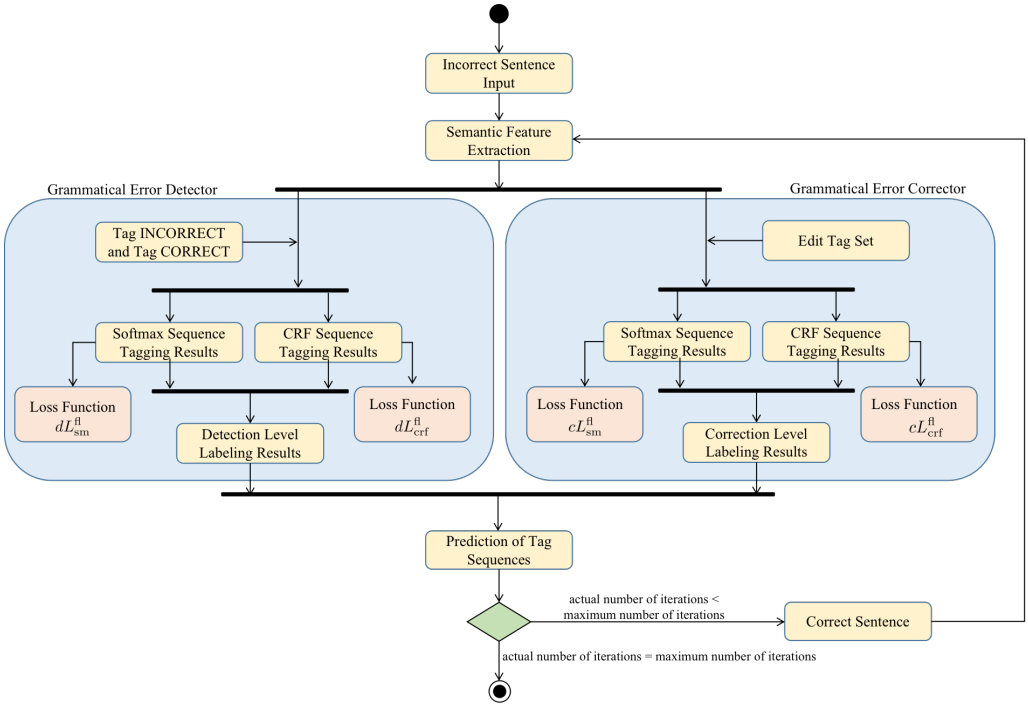

Figure 3. The activity diagram of the joint focal loss function.

The grammatical error corrector and the detector use a similar objective function for the CRF model, both calculated via maximum likelihood estimation. The optimization objective is:

$$dL_{\text{crf}} = -\log P_{\text{crf}}(T \mid H) \tag{9}$$

$$cL_{\text{crf}} = -\log P'_{\text{crf}}(T \mid H) \tag{10}$$

In the formulas above, P_{crf} and P'_{crf} are obtained through Equation (6). Then, the final loss function is:

$$L = dL_{\text{sm}} + dL_{\text{crf}} + cL_{\text{sm}} + cL_{\text{crf}} \tag{11}$$

Since the number of correct tokens in a sentence is much higher than the number of incorrect tokens, the error detector labels more correct tags than incorrect tags, and the error corrector labels far more KEEP tags than other tags. Category imbalance is essentially a reflection of the difference in classification difficulty. If a label is highly dominant in the training data, the model will prefer to use the highly dominant label during inference. In order to solve this category imbalance problem, this paper uses a focused penalty strategy [6] to ensure the model pays less attention to the more dominant labels. The loss function after using focal loss is:

$$\begin{aligned}
dL_{\text{sm}}^{\text{fl}} &= -\frac{1}{N}\sum_{i=1}^{N}(1-p_i)^{\gamma}[t_i \cdot \log(p_i) + (1-t_i)\cdot \log(1-p_i)] \\
cL_{\text{sm}}^{\text{fl}} &= \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{L}(1-p_ic)^{\gamma} t_{ic} \log(p_{ic}) \\
dL_{\text{crf}}^{\text{fl}} &= -(1-P_{\text{crf}}(T \mid H))^{\gamma} \log P_{\text{crf}}(T \mid H) \\
cL_{\text{crf}}^{\text{fl}} &= -\left(1-P'_{\text{crf}}(T \mid H)\right)^{\gamma} \log P'_{\text{crf}}(T \mid H) \\
L^{\text{fl}} &= dL_{\text{sm}}^{\text{fl}} + cL_{\text{sm}}^{\text{fl}} + dL_{\text{crf}}^{\text{fl}} + cL_{\text{crf}}^{\text{fl}}
\end{aligned} \tag{12}$$

where γ is a hyperparameter that controls the weight of the penalty.

3.5. Iterative Training Strategy

The sentence may contain multiple errors, and a single correction may not correct them all. Consequently, the baseline [3] uses an iterative decoding strategy that selects a candidate with a high confidence level in each round as the input for the next round, and keeps decoding until the input no longer needs to be corrected. The structure of iterative inference is shown in Figure 4.

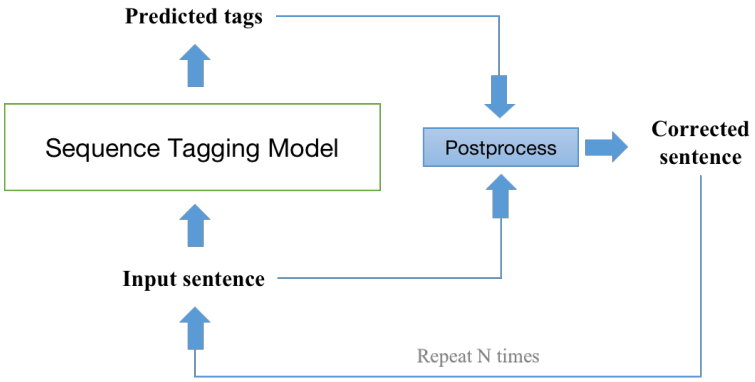

Figure 4. The structure of iterative inference.The corrected sentence, which results from the processing of the input sentence and the predicted tags, is used as the input sentence for the next round.

Due to the iterative inference process, the correction result of the model in the current round depends on the output of the previous round, whereas the baseline is only trained with the first edited tag sequence, and the connection between multiple rounds of error correction cannot be learned. This imbalance in the training and inference process can lead to problems such as exposure bias in left-to-right sequence generation. The difference is that the sequence generation relies on the prediction result of the previous token when predicting the next token, whereas the baseline relies on the predicted sentences from the previous round to make its predictions. To address this issue, we propose a dynamic iterative training method that is consistent with iterative prediction. Since the output of each iteration of training must be compared with the target label to calculate the loss function, but the length of the input changes in each round, the edit label generated from the original sentence of the training data and the target sentence cannot be directly taken as the target label for each iteration. To solve this problem, dynamically generated target tags are used for each iteration. Specifically, the target label for each iteration is generated from the input and target sentences of the current iteration. The structure of the iterative training is shown in Figure 5. The average loss in each round is taken as the final loss.

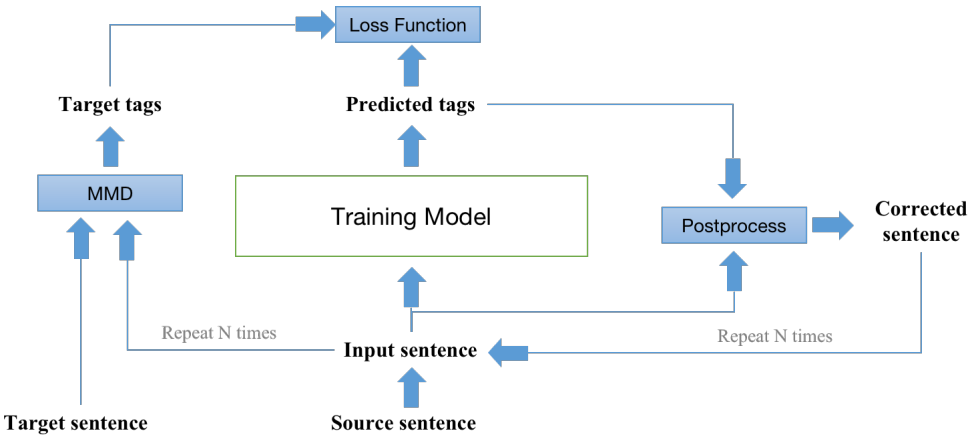

Figure 5. The structure of iterative training.

4. Experiments

4.1. Datasets and Evalution Metrics

The experiments in this paper were carried out on the dataset provided by the NLPCC 2018 Chinese Grammar Error Correction Sharing Task [7]. The dataset consisted of 717,241 items of training data and 2000 items of test data. The training data were collected from Lang-8, which is a multilingual language-learning platform. In the training data, each incorrect sentence corresponds to multiple correct sentences. We split these to obtain a total of 1,349,769 sentence pairs and randomly selected 5000 sentence pairs as the validation set. The remaining 1,344,769 sentence pairs were used as the training set. The test set consisted of 2000 annotated sentences, and these annotations followed the minimum edit distance principle. Table 1 shows the details of the experimental data.

Table 1. Details of NLPCC 2018 test set

Split	Sentence	Token src	Token tgt	Error Token	Redundant Errors	Missing Errors	Ordering Errors	Selection Errors
Train	1.3 M	23.7 M	25.0 M	-	-	-	-	-
Dev	5000	99.7 K	103.2 K	-	-	-	-	-
Test	2000	58.9 K	-	4371	911	1147	176	2137

The evaluation method was consistent with the grammar correction task of NLPCC 2018, using the MaxMatch (M2) tool for evaluation. The evaluation metrics were precision, recall and F0.5 measure.

4.2. Parameters Setting

The deep learning framework used in this experiment was PyTorch. We initialized the Transformer encoder using a pre-trained model. The hyperparameters setting for CGED-IT was the same as for GECToR. The pre-training model used was the Chinese-roberta-wwm-ext model [28]. The batch size for training was set to 64, and that for testing was set to 128. The optimizer used AdamW, with 1 epoch for warm-up learning and a warm-up learning rate of 1×10^{-3}. The maximum number of iterations for both training and inference was set to 3 after parameter tuning. The focal loss hyperparameter γ for CGEC-IT was set to 0.5 after parameter tuning.

4.3. Overall Results

To evaluate the effectiveness of our model for Chinese GEC, we conducted experiments on the NLPCC 2018 dataset. Table 2 shows the performance of our approach compared with recent models for Chinese GEC. The first column shows some Chinese GEC methods, and the second column shows the time required to decode the 2000 sentences of the NLPCC 2018 test dataset. The GEC results are presented in the last three columns, i.e., precision, recall and F0.5. The models in the top group are all classical single models for the Chinese GEC task. The second group shows the three best-performing teams in the NLPCC 2018 Chinese grammar error correction competition. The best results were all obtained using the joint model. The bottom group includes the baseline model and the approach proposed in this paper. In order to verify the effectiveness of the several improvement methods proposed in this paper, we used the baseline model for experiments using each of the three improvement methods, and the experimental results are also shown in the bottom group.

The GECToR model achieved a 29.8% F0.5 measure. There is a gap between this baseline model and the leading Chinese GEC models such as the YouDao model [29], the AliGM model [30] and the BLCU model [13]. However, our model achieved an F0.5 score of 31.8%, which was higher than the best-performing BLCU model [13]. It is worth noting that the top three models in NLPCC 2018 Task 2 were ensemble models, but our single model still outperformed them. Our model does not conflict with the ensemble method, and it is theoretically possible to further improve the performance using this method. Moreover, our proposed approach is comparable to the baseline in terms of decoding speed and far better than the autoregressive models such as Seq2Seq, BERT-encode and BERT-fused.

We conducted separate experiments on each of the three proposed improvements based on GECToR, and the experimental results showed that all three improvements improved the model performance. The CRF layer increased the focus of the GECToR model on the overall annotation results, which improved the F0.5 measure by 0.2%. The use of focus loss mitigated label classification imbalances, increasing false positives and resulting in a 0.9% reduction in the recall, while significantly increasing the precision by 1.5%. The iterative training method solved the exposure bias problem that exists in GECToR and led to the most obvious improvement in performance, with a 1.6% improvement in the F0.5 measure. We concluded that the GECToR method does have an exposure bias problem, and that solving the problem by iterative training can further improve the performance of the model. Finally, the combination of the above three improvements resulted in a 2% improvement in the F0.5 measure, which verifies the effectiveness of our proposed methods.

Table 2. Estimation Results on the NLPCC2018 dataset (2000).

System	Time (in Seconds)	P	R	F0.5
Seq2Seq [11]	63	36.9	14.4	28.1
ConvSeq2Seq [13]	31	41.7	13.1	29.0
LaserTaggger [22]	13	25.6	10.5	19.9
ESC [15]	29	37.3	14.5	28.4
BERT-encode [18]	>63	32.7	22.2	29.8
BERT-fused [18]	>63	32.1	23.6	29.9
YouDao [29]	-	35.2	18.6	29.9
AliGM [30]	-	41.0	13.8	29.4
BLCU [13]	-	47.6	12.6	30.6
Baseline [3]	10	33.6	20.4	29.8
+CRF	11	33.6	21.0	30.0
+fl	10	35.1	19.5	30.3
+iterative train	10	35.8	21.1	31.4
CGEC-IT	11	36.2	21.3	31.8

The value of the focal loss hyperparameter γ can affect the performance of the model. In this study, several different hyperparameters with different focal losses were tried, and the experimental results are shown in Figure 6. Finally, we chose 0.5 as the value of the focal loss hyperparameter γ for CGEC-IT.

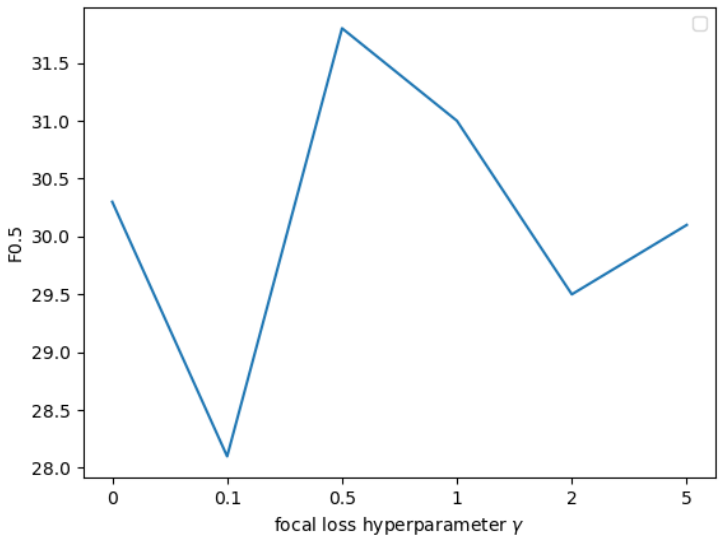

Figure 6. Tuning for focal loss hyperparameter γ for CGEC-IT.

We also investigated the effect of different numbers of iterations on the performances of the baseline method and the method described in this paper, and the experimental results are shown in Figure 7. It can be seen that the highest value of F0.5 was obtained when the maximum number of iterations was three. Continuing to increase the number of iterations

did not improve the results further, but the inference time still increased. Therefore, the maximum number of iterations used in this study was three.

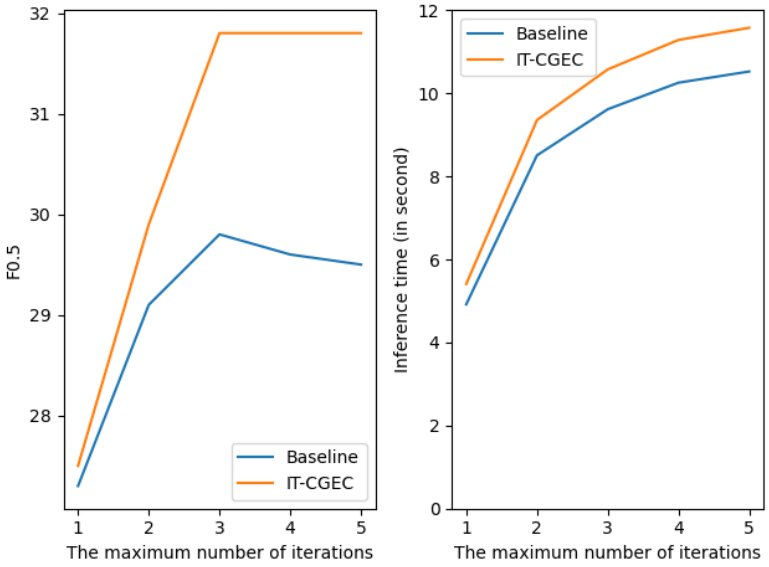

Figure 7. F0.5 measure and inference time for baseline and CGEC-IT using different maximum numbers of iterations.

5. Conclusions

In this paper, the grammatical error correction task was viewed as a sequence tagging task, which has greater interpretability as well as faster decoding speed compared to the generation task. To address the problem of exposure bias due to the mismatch between the training and inference phases of the sequence tagging model GECToR, the proposed approach used dynamically generated target labels for iterative training. Multiple errors in the same sentence may be related to each other, and therefore a CRF layer was added for sequence annotation, to focus the model more closely on the overall annotation results. During the training phase, focal loss was introduced to alleviate the severe mismatch in classification labels caused by the vast majority of words in a sentence that do not require correction. CGEC-IT outperformed the models in previous studies on the F0.5 measure on the NLPCC 2018 dataset, and its usefulness was demonstrated on the Chinese GEC task. Due to the fast decoding speed of this method, CGEC-IT can be applied to real-time Chinese text error correction, to reduce the cost of manual review. Future attempts will be made to apply the method to other languages. There will also be a focus on the effect of other features of Chinese such as pinyin on the Chinese GEC task, allowing for further development of the Chinese GEC model.

Author Contributions: Conceptualization, H.K. and K.W.; methodology, H.K.; software, K.W.; validation, X.M.; data analysis, K.W. and X.M.; original draft preparation, X.M. and X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 61772088).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yuan, Z.; Briscoe, T. Grammatical error correction using neural machine translation. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 380–386.
2. Awasthi, A.; Sarawagi, S.; Goyal, R.; Ghosh, S.; Piratla, V. Parallel Iterative Edit Models for Local Sequence Transduction. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language, Hong Kong, China, 3–7 November 2019; pp. 4260–4270.

3. Omelianchuk, K.; Atrasevych, V.; Chernodub, A.; Skurzhanskyi, O. GECToR–Grammatical Error Correction: Tag, Not Rewrite. In Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications, Online, 10 July 2020; pp. 163–170.
4. Goodman, S.; Ding, N.; Soricut, R. TeaForN: Teacher-Forcing with N-grams. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 8704–8717.
5. Lafferty, J.D.; McCallum, A.; Pereira, F.C. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williamstown, MA, USA, 28 June–1 July 2001.
6. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
7. Zhao, Y.; Jiang, N.; Sun, W.; Wan, X. Overview of the nlpcc 2018 shared task: Grammatical error correction. In Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing, Qingdao, China, 13–17 October 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 439–445.
8. Wu, C.H.; Liu, C.H.; Harris, M.; Yu, L.C. Sentence correction incorporating relative position and parse template language models. *IEEE Trans. Audio Speech Lang. Process.* **2009**, *18*, 1170–1181.
9. Yu, C.H.; Chen, H.H. Detecting word ordering errors in Chinese sentences for learning Chinese as a foreign language. In Proceedings of the COLING 2012, Mumbai, India, 8–15 December 2012; pp. 3003–3018.
10. Yu, L.C.; Lee, L.H.; Chang, L.P. Overview of grammatical error diagnosis for learning Chinese as a foreign language. In Proceedings of the 1st Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA-14), Nara, Japan, 30 November 2014; pp. 42–47.
11. Hu, Q.; Zhang, Y.; Liu, F.; Gu, Y. Ling@ cass solution to the nlp-tea cged shared task 2018. In Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications, Melbourne, Australia, 19 July 2018; pp. 70–76.
12. Lichtarge, J.; Alberti, C.; Kumar, S.; Shazeer, N.; Parmar, N.; Tong, S. Corpora Generation for Grammatical Error Correction. *arXiv* **2019**, arXiv:1811.01710.
13. Ren, H.; Yang, L.; Xun, E. A sequence to sequence learning for Chinese grammatical error correction. In Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing, Qingdao, China, 13–17 October 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 401–410.
14. Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; Dauphin, Y.N. Convolutional sequence to sequence learning. In Proceedings of the International Conference on Machine Learning (PMLR 2017), Sydney, Australia, 6–11 August 2017; pp. 1243–1252.
15. Chen, M.; Ge, T.; Zhang, X.; Wei, F.; Zhou, M. 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), Online, 16–20 November 2020; pp. 7162–7169.
16. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
17. Zhu, J.; Xia, Y.; Wu, L.; He, D.; Qin, T.; Zhou, W.; Li, H.; Liu, T. Incorporating BERT into Neural Machine Translation. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
18. Wang, H.; Kurosawa, M.; Katsumata, S.; Komachi, M. Chinese Grammatical Correction Using BERT-based Pre-trained Model. In Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, Suzhou, China, 5–6 December 2020; pp. 163–168.
19. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
20. Kaneko, M.; Mita, M.; Kiyono, S.; Suzuki, J.; Inui, K. 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, Online, 5–10 July 2020; pp. 4248–4254.
21. Li, P.; Shi, S. Tail-to-Tail Non-Autoregressive Sequence Prediction for Chinese Grammatical Error Correction. *arXiv* **2021**, arXiv:2106.01609.
22. Malmi, E.; Krause, S.; Rothe, S.; Mirylenka, D.; Severyn, A. Encode, Tag, Realize: High-Precision Text Editing. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 5054–5065.
23. Rao, G.; Yang, E.; Zhang, B. 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, Suzhou, China, 4 December 2020; pp. 25–35.
24. Liang, D.; Zheng, C.; Guo, L.; Cui, X.; Xiong, X.; Rong, H.; Dong, J. 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, Suzhou, China, 4 December 2020; pp. 57–66.
25. Zhao, Z.; Wang, H. MaskGEC: Improving neural grammatical error correction via dynamic masking. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 1226–1233.

26. Tang, Z.; Ji, Y.; Zhao, Y.; Li, J. Chinese Grammatical Error Correction enhanced by Data Augmentation from Word and Character Levels. In Proceedings of the 20th Chinese National Conference on Computational Linguistics, Hohhot, China, 13–15 August 2021; pp. 813–824.
27. Sun, Z.; Li, Z.; Wang, H.; He, D.; Lin, Z.; Deng, Z. Fast structured decoding for sequence models. *arXiv* **2019**, arXiv:1910.11555.
28. Cui, Y.; Che, W.; Liu, T.; Qin, B.; Wang, S.; Hu, G. Revisiting Pre-Trained Models for Chinese Natural Language Processing. *arXiv* **2020**, arXiv:2004.13922.
29. Fu, K.; Huang, J.; Duan, Y. Youdao's Winning Solution to the NLPCC-2018 Task 2 Challenge: A Neural Machine Translation Approach to Chinese Grammatical Error Correction. In Proceedings of the 7th CCF International Conference (NLPCC 2018), Hohhot, China, 26–30 August 2018.
30. Zhou, J.; Li, C.; Liu, H.; Bao, Z.; Xu, G.; Li, L. Chinese grammatical error correction using statistical and neural models. In *Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing, Qingdao, China, 13–17 October 2018*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 117–128.