Combining N-gram Statistical Model with Pre-trained Model to Correct Chinese Sentence Error

G. L. He, C. Y. Chi and Y. Y. Zhan

***Abstract*—There have been a fund of studies on Chinese Grammatical Error Correction (CGEC) since it was proposed by NLPCC 2018 shared task 2. In previous studies, most researchers regarded this task as a Neural Machine Translation (NMT) task, which treated erroneous sentences as source-language and correct sentences as target-language. But this method relies on large-scale parallel corpus. In recent years, Bidirectional Encoder Representations from Transformers (BERT) and its variants have made an exciting breakthrough on various NLP tasks and inspire NLP practitioners to explore the utilization of pre-trained model. However, BERT performs better on Natural Language Understanding (NLU) benchmarks (e.g., SQuAD v1.1), the applications on generative tasks are inadequate. In NLP-TEA CGED Shared Task 2020, many methods based on BERT Pre-trained model have emerged. Unlike CGED tasks, whose purpose is to detect error position and error types in a sentence, are usually regarded as sequence labelling or binary classification problem. CGEC is a sequence generation task. In this study, we leverage n-gram statistical language model as a spelling checker and BERT-based pre-trained model as the encoder in sequence-to-sequence (seq2seq) structure to solve CGEC problem. Our baseline is Transformer. The experimental results demonstrate that our method outperforms the other three participating teams but also some latest methods, and we analyze how different checkpoints affect our results.**

***Index Terms*—Chinese grammatical error correction, BERT, N-gram, Pre-trained model**

I. Introduction

GRAMMATICAL error corrcetion (GEC) has long been a question of great interest in a wide range of natural language processing (NLP) fields. GEC is the task of correcting different kinds of errors in texts or sentences, such as spelling, punctuation, grammatical, and word choice errors. The aim of a GEC system is to take a potentially erroneous sentence as input and expect to transform it to its corrected version (As shown in Table 1). Since English is the most widely used language in the world, a considerable amount of shared tasks have been put forward for English GEC research, most notably with the CoNLL-2013, CoNLL-2014 [1], [2] and BEA-2019 [3] shared task. Compared with English GEC, Chinese grammatical error correction (CGEC) has a much shorter history, due to the fact that Chinese characters are far different from English ones. In earlier studies, because of the lack of corpus of corresponding sentence pairs, the researches for CGEC were first came up with its previous stage: Chinese grammatical error diagnosis (CGED) , such as IJCNLP-2017 [4], NLPTEA-2018 [5]. In these tasks, participants were required to detect where the errors probably are in sentences, and the most representative method is conditional random fields (CRF) and long short term memory (LSTM) network [6], [7] proposed by Xie et al. and Zheng et al. The CGEC task is more challenging and valuable than CGED task, but it was not until NLPCC 2018 shared task [8] raised this challenge that NLP researchers had the opportunity to participate and develop CGEC.

Recently, pre-training approaches, such as ELMo [9], GPT-2 [10], BERT [11], XLNet [12] and RoBERTa [13], which utilize large amount of unlabeled data to capture enriched contextual representations lead to marvelous improvements on natural language understanding (NLU) tasks, like SQuAD [14] and CoQA [15]. But very little work has been done to applying such pre-training techniques to sequence-to-sequence (seq2seq) models. For the GEC task, extensive researches have shown that treating GEC task as a sequence-to-sequence neural machine translation (NMT) task is feasible [16]–[18]. However, training an NMT system mentioned above requires considerable parallel corpus, otherwise it will cause insufficient network training. So we come up with the idea that leveraging both pre-trained model and seq2seq structure to solve CGEC problem, in order to acquire more existing knowledge of Chinese learned by a pre-trained model and output the result via seq2seq structure. Moreover, some recent researches have proven that BERT works well in English sentence correction tasks [19], [20], maybe Chinese GEC task will also work as well.

Inspired by Qiu's work [21] and Fu's work [18], a GEC task consists of two phases: The first stage aims at solving simple problems, such as spelling error, which could be done with a spelling checker by statistical strategy. In the second stage, more complex errors like grammatical error will be solved by neural network.

In this study, we leverage n-gram statistical language model as a spelling checker and BERT-based pre-trained model as the encoder in seq2seq structure. At the first stage, we utilize n-gram language model and SIGHAN 2013 CSC Datasets [22] to detect and verify the spelling error in training dataset. At the second stage, we propose a new seq2seq structure based on Transformer [23] that is compatible with Chinese pre-trained model. We modify slightly on Trans-

Manuscript received September 6, 2021; revised February 19, 2022. This work was supported in part by the University of Science and Technology Liaoning under Grant LKDYC201917.

G. L. He is a student of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China (e-mail: heguanlin00@gmail.com).

C. Y. Chi is a professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China (corresponding author, phone: 86-13941249890; e-mail: chichengying@ustl.edu.cn).

Y. Y. Zhan is a student of College of Science and Health, Technological University Dublin, Dublin D08 X622, Ireland (e-mail: D16123420@mytudublin.ie).

TABLE I
TYPICAL EXAMPLES FOR FOUR TYPES OF ERROR. ACCORDING TO NLPTEA 2016 SHARED TASK, CHINESE GRAMMATICAL ERRORS ARE DIVIDED INTO FOUR CATEGORIES: REDUNDANT WORDS (DENOTED AS "R"), WORD ORDERING ERRORS ("W"), MISSING WORDS ("M") AND WORD SELECTION ERRORS ("S"). THE SOURCE SENTENCE IS THE SENTENCE CONTAINING THE ERROR, WHILE THE TARGET SENTENCE IS THE CORRECTED ONE.

Error Type	source sentence	target sentence
R	他们是离婚了，所以不一起住 They are have divorced, so they don't live together	他们离婚了，所以不一起住 They are divorced, so they don't live together
W	我非常快乐，跟妹妹再想去唱卡拉OK I am very happy, and my sister wants to sing karaoke again	我非常快乐，想跟妹妹再去唱卡拉OK I am very happy and want to go singing karaoke with my sister again
M	请这个句子对不对? Please this sentence correct?	请问这个句子对不对? Excuse me, is this sentence correct?
S	好想跟她再见 I want to say goodbye to her	好想跟她再会 I want to see her again

former's encoder and leverage BERT-initialized pre-trained model as the encoder, paired with a randomly initialized decoder, since Rothe et al [24] have proven that the encoder part of BERT is crucial to the sequence generation task, while decoder is not. Our models obtain $F_{0.5}$ scores of 33.21 and 35.72 respectively, which perform better than the top three teams of the shared task, demonstrating that our work is valuable and effective.

The rest of the paper is organized as follows. Chapter II first gives a brief overview of the recent related work of GEC. The third chapter is concerned with the methodology used for this study. Chapter IV describes the experimental setup and results. Chapter V presents the findings of this research and Chapter VI makes a conclusion of the paper.

II. RELATED WORK

Early GEC missions usually used a rule-based approach [25] or classifier-based approach [26], but these methods can only modify specific concentrated errors in the text. To address various types of errors, Brockett et al. [27] first proposed the idea that GEC tasks can be regarded as translation tasks, and then the Statistical Machine Translation (SMT) method is widely accepted in the field of grammatical error correction. Felice et al. [28] propose a hybrid system which consists of a rule-based system and a SMT system. Xie et al. [29] introduced deep neural network to text correction for the first time, and proposed a sequence-to-sequence-based character-level grammar correction system. At that time, this type of SMT method further promoted the grammatical error correction method to a new level on English benchmark test data sets.

With the widespread application of deep learning, along with existing research recognises the critical role played by Neural Machine Translation (NMT) in the field of GEC. Sun et al. [30] employ Convolutional Neural Network (CNN) for English grammatical error correction. Yuan and Briscoe [16] propose a two-step method to the unregistered word problem. Chollampatt et al. [17] improved grammatical correction system by using a multilayer convolutional encoder-decoder neural network, and their strategy outperforms all prior neural approaches. Ge et al. [31] use a strategy called "fluency booast learning", which eliminates the defects of canonical seq2seq models.

Unlike the extensive research on English GEC task [1], [2], [17], [30], there is relatively little research on Chinese GEC task. Prior to 2018, the main focus was on Chinese grammatical error diagnosis (CGED) [32], [33]. For instance, Zheng applies sequence annotation methods (CRF, LSTM) to the task of grammatical error detection, and further improves the results through model ensemble methods. Until 2018, a Chinese GEC shared task was first performed at NLPCC 2018, which accelerated the progress of CGEC in this field. In this competition, the winning solution was proposed by Fu et al [18]. They regarded the CGEC task as translation task and proposed a phased solution: After removing simple errors by a statistics-based approach, they input the corrected sentence into the neural network to correct the remaining complex errors. Besides, Zhou et al. [34] used a stack approach to combine different kinds of error correction models, including rule-based error correction model, statistics-based error correction model, and deep learning-based error correction model. First obtain the candidate sets on the low-level error correction model, and then combine the candidate sets on the high-level model.

Last year, at NLP-TEA CGED Shared Task 2020, many methods based on BERT Pre-trained model have emerged. Cheng et al [35] used BERT to train a binary classification model to detect whether a sentence has errors, in their study, they trained a classification model by BERT pre-trained model. Cao et al [36] applied the idea of sequence labeling to the error detection task. The experimental baseline model used BiLSTM-CRF, and the role of BERT was to obtain the embedding layer representation of the input text. Liang et al [37] accomplished a hybrid model for CGED task, integrating position-tagging model and correction-tagging model. With position-tagging model, they trained a sequence tagging model using RoBERTa as the model's encoder. Zan et al [38] added an error correction step based on the experiment of Cao et al [36]. In the detection stage, the combination of BiLSTM and BERT is also used, and in the correction stage, an n-gram language model and a sequence-to-sequence neural network are used.

III. SYSTEM AND METHODS

In this study, as shown in Figure 1, we establish two models, respectively statistical model and seq2seq model. For the statistical model, we utilize n-gram language model and SIGHAN 2013 CSC Dataset to detect and verify the spelling errors in training dataset. For the seq2seq model, we leverage BERT-variants pre-trained model as the encoder in sequence-to-sequence (seq2seq) structure, in our proposed structure, we inherited the Transformer model and made a small change on this basis. On the encoder side, we train a BERT model to generate large semantic representations, while on the decoder side, we use a conventional Transformer

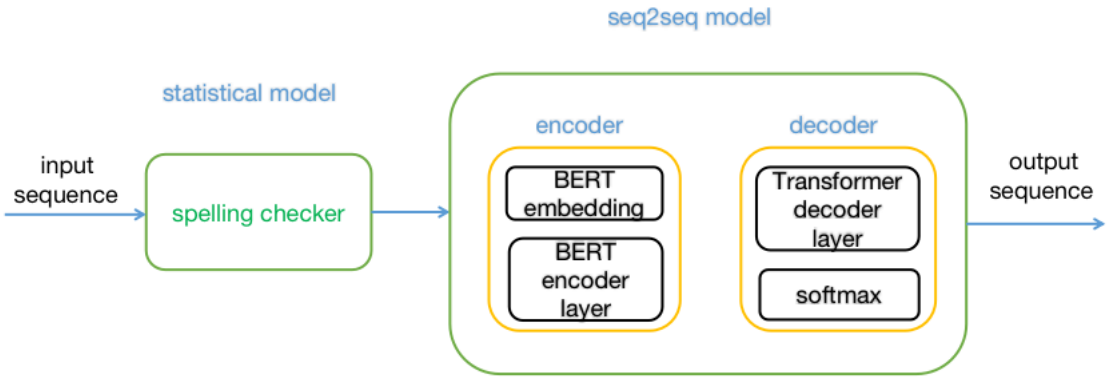

Fig. 1. The illustration of our model: a spelling checker based on statistical model and a BERT-based encoder sequence-to-sequence structure.

decoder and random initialization strategy. In the following subsections, we will elaborate on the methods we use.

A. Spelling Checker and N-gram Model

Different from English, there are two cases of spelling errors in Chinese, namely the spelling errors of the characters and the one of the phonetics. The reason for this phenomenon is the two mainstream Chinese input method editors, Wubi and Pinyin. To handle this problem, SIGHAN 2013 CSC provides similar shape and similar pronunciation character sets. For example, the set of similar shape of the character "丁" and the set of similar pronunciation of the character "又" are listed as follows:

- *Similar Shape:* 丁,叮打汀歹可仃
- *Similar Pronunciation:* 又,幼黝宥右柚有

N-gram is a method that facilitates statistical calculation. Its principle is to slide and slice on the text sequence. The sliced part is called a gram, and N is the width of the sliding window. Assuming that the occurrence probability of the current word is only related to the words preceding it, the probability of the entire sentence can be calculated.

If we have a sequence of m words (or a sentence), we want to calculate the probability $p(w_1, w_2, ..., w_m)$, according to the chain rule, we can get $p(w_1, w_2, ..., w_m) = p(w_1) * p(w_2|w_1) * p(w_3|w_1, w_2) * ... * p(w_m|w_1, ..., w_{m-1})$. Using the assumption of Markov chain, that is, the state of the current word is only related to the first few words adjacent to it, so that the length of the above formula can be greatly reduced. which is $p(w_1, w_2, ..., w_m) = p(w_i|w_{i-n+1,...,w_{i-1}})$. In this study, we choose a trigram model, that is n equals to 3 (1),

$$P(w_1, w_2, ..., w_m) = \prod_{i=1}^{m} P(w_i|w_{i-2}w_{i-1}) \quad (1)$$

Next, in the given training corpus, use Bayes' theorem to calculate all the above conditional probability values (2).

$$P(w_i|w_{i-2}, w_{i-1}) = \frac{C(w_{i-2}, w_{i-1}, w_i)}{C(w_{i-2}, w_{i-1})} \quad (2)$$

Here is the illustration of using Similar Character Set (SCS) and language model to correct simple errors: First, segment the input sentence into words, for words that do not appear in the dictionary, each character is replaced by SCS and a candidate word set is generated. Next, use the N-gram model to select the sentence with the lowest perplexity. See Algorithm 1 for details.

Algorithm 1 simple error correction

Input: input sequence (S), language model (LM), dictionary (D), Similar Character Set (SCS)
Output: corrected sequence

```
S* ⇐ input sequence has been segmented
C[] ⇐ candidate substitution word set
for each word w in S* do
    if w ∉ dictionaryD then
        for each character c in w do
            C[i] += replaced by SCS
            target output=LM.score(min perplexity(C[i]))
        end for
    else
        continue
    end if
end for
corrected sequence = target output

return corrected sequence
```

B. ATTENTION MECHANISM

The attention mechanism is inspired by the way of human thinking. When doing translation tasks, the content of the part to be translated can be judged according to the key words in the context. In the attention model, when we translate the current word, we can find the key points according to all the words in the source sentence, and combine the translation content of the previous text to get the translation result of the part to be translated, so that when our decoder predicts the target translation, we can refer to all the information of the encoder, not only the fixed-length hidden vector in the original model, and will not lose long-range information.

For an input sequence $X = (x_1, x_2, ..., x_t)$, we use the RNN structure to get the hidden state in the encoder $h = (h_1, h_2, ..., h_t)$. Suppose the hidden state of the current decoder is s_{t-1}, the relationship between each position of the input sequence j and the current output position can be calculated, shown as (3),

$$e_{tj} = a(s_{t-1}, h_j) \quad (3)$$

Written in the corresponding vector form is (4):

$$\vec{e_t} = (a(s_{t-1}, h_1), ..., a(s_{t-1}, h_t)) \tag{4}$$

where a is a correlation operator, such as the common form of multiplication, weighted multiplication, etc. For $\vec{e_t}$, perform a softmax operation to normalize it to get the distribution of attention (5):

$$\vec{\alpha_t} = softmax(\vec{e_t}) \tag{5}$$

The expanded form is (6):

$$\alpha_{tj} = \frac{exp(e_{tj})}{\sum_{k=1}^{T} exp(e_{tk})} \tag{6}$$

Using $\vec{\alpha_t}$ we can perform weighted summation to get the corresponding context vector, as shown in the formula (7):

$$\vec{c_t} = \sum_{j=1}^{T} \alpha_{tj} h_j \tag{7}$$

From this, we can calculate the decoder's next hidden state $s_t = f(s_{t-1}, y_{t-1}, c_t)$ and the output of that position $p(y_t|y1, ..., y_{t-1}, \vec{x}) = g(y_{i-1}, s_i, c_i)$

Since the attention mechanism is so effective, Vaswani et al. [23] proposed a self-attention architecture called Transformer, which is completely based on attention mechanisms and get rid of the sequential structure like RNN. Due to its marvelous performance in seq2seq task, we build our model based on Transformer.

C. BERT

Bert is based on Transformer's deep bidirectional language representation model, it is a bidirectional encoder network based on Transformer structure. One of its characteristics is that all layers are pre-trained in conjunction with context. Bert's goal is to generate a pre-trained language model, so Encoder mechanism is the only element. The working principle of BERT's Masked Language Model task is shown in Figure 2. Given an input sequence $W = (W_1, W_2, ..., W_m)$, before feeding the sequence into BERT, 15 percent of the words in the input sequence are covered with a [MASK] token, then perform embedding operations on each token. After doing this we send the sequence to Bert and get the encoder's output $O = (O_1, O_2, ..., O_m)$, then we multiply the output vectors by the embedding matrix, transforming them into the same dimension as vocabulary. The model calculates the probability of each word in the vocabulary through the softmax function to predict the word at the [MASK] token.

D. Gradient Accumulation

In order to solve the problem of insufficient GPU memory, we used the gradient accumulation mechanism in our experiment. During the training process of the neural network, a gradient clearing will be called after the update is completed. The way to accumulate gradients is to delay the call, and then call the reverse update and gradient clearing after a few batches. By delaying the update of the parameters, the same effect as using a larger batch size can be achieved.

IV. Experiments

A. DATASET

In this study, we use NLPCC 2018 shared task's official dataset, which is available at https://github.com/zhaoyyoo/NLPCC2018_GEC/. The raw training data has a total of 717,241 rows, each row consists of four columns, namely sen_id, num_correct, orig_sen and corrections. We extract the last two columns to construct sentence pairs. It is worth noting that there may be zero or more corrected sentences for an orig_sen, depending on the value of num_correct, after processing the training data, we get 1,097,190 sentence pairs. Next, we evaluate the generated sentence pairs, and we use the data filtering methods commonly used in machine translation to filter the data [39]. We exclude sentence pairs that meet the following rules:

- *The length of source sentence or target sentence if greater than 200.*
- *The edit distance between source sentence and target sentence is greater than 15.*
- *The source sentence is twice the length of the target sentence, or vice versa.*

After filtering the training data, we obtain 980,152 sentence pairs.

The test data was provided by NLPCC 2018 shared task, which contains 2,000 sentences. Since NLPCC 2018 shared task does not provide a validation set, we use the method of Ren et al [40]. Randomly select 5000 items from the training set to form the validation set.

In addition, in this experiment we used the HSK Chinese Proficiency Test corpus as data augmentation to verify the impact of the size of the training data on the results. We use the same criteria as above to filter the data, moreover, because traditional Chinese exists in the HSK data set, we also use the zhconv tool to convert the traditional Chinese into simplified Chinese. In total, we use 1,215,069 sentence pairs for training. We crawled over 300 Chinese news websites and obtained over 700GB data to train n-gram statistical model, our dictionary is also generated from these data.

B. Model

In our experiments, we used https://github.com/pytorch/fairseq to implement the Transformer model and the model we mentioned in Chapter III. We inherited the Transformer(self-attention) networks of fairseq and made some changes to train the following models:

1) Transformer: First, we train an unmodified Transformer model according to the paper [23] as the baseline of our experiment. Before training, we split all training sentence pairs into characters, due to the fact that the bert pre-training model is based on characters. The following model training also uses characters as the basic unit.

The dimension of the embedding layer of the encoder and decoder is set to 512, and the dimension of the FNN layer of the encoder is 2048. In order to unify the variables, we set both the encoder and decoder to 12 layers, because we will train a 12-layer bert model later, and the number of heads for Transformer self-attention is set to 8. In the training phase, we set the initial learning rate to 1e-5, we adopt the Adam optimizer in our model with the value of β_1 and β_2 are 0.9 and 0.999 respectively, and choose reduce_lr_on_plateau as

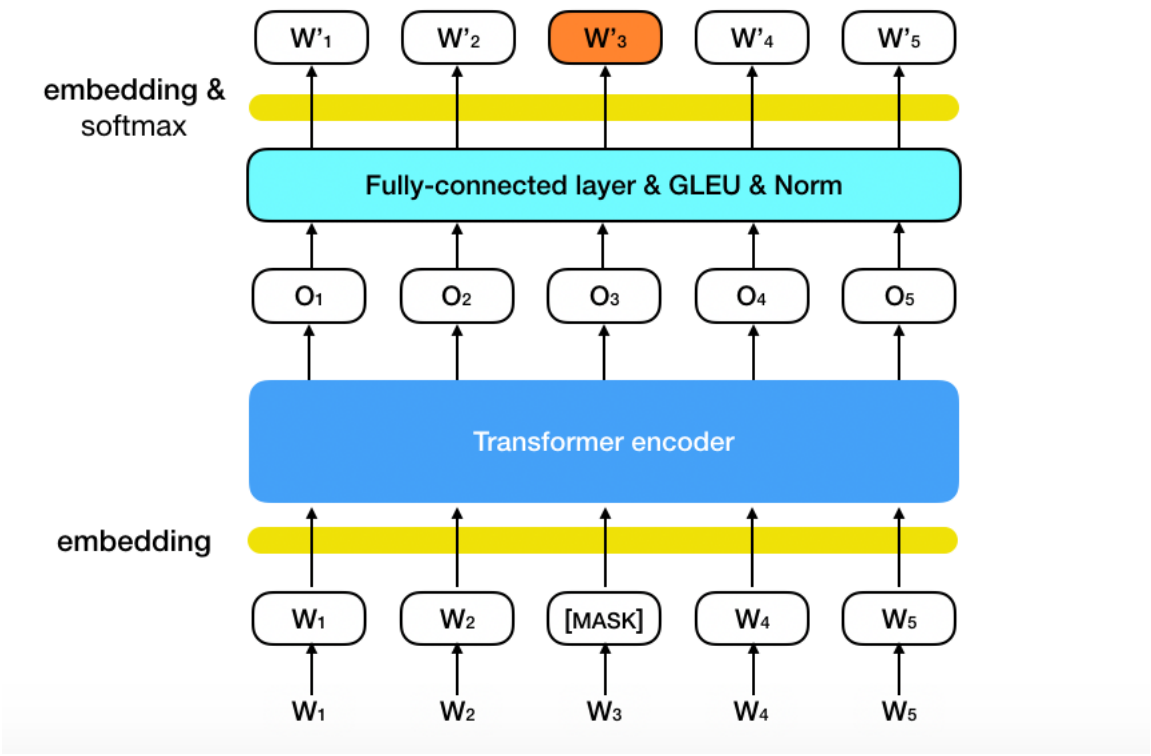

Fig. 2. The illustration of BERT's Masked LM task, where $w = (w_1, ..., w_5)$ denotes input sequence, while $w = (w'_1, ..., w'_5)$ denotes predicted sequence.

the way we update the learning rate, which means that the learning rate is updated when the loss of the validation set no longer drops. The dropout probability was set to 0.1, and the batch size was set to 2048 tokens. In the inference phase, we set the beam size to 12.

2) Ours: Specifically, our model loads the Bert model into the encoder and randomly initializes a decoder. We use pytorch_transformers to load BERT-Base-Chinese pre-trained checkpoint. Except that we set the batch size to 16 during the training process, other parameters are the same as the baseline model. In our experimental environment, it took about three days to train BERT model, and the experimental environment will be listed in Appendix A.

The details of the model parameters can be found in Appendix B.

C. EVALUATION

In this study, We use M^2 Scorer to evaluate performances. MaxMatch (M^2) algorithm is a commonly used text matching algorithm. The effect of the model is judged by phrase-level prediction result of the model and calculating the maximum overlap with the gold standard. The precision, recall, and F-score measure between the set of system edits $e = \{e_1, e_2..., e_n\}$ and the set of gold edits $g = \{g_1, g_2, ..., g_n\}$ for all sentences are computed as following formulas:

For Precision:

$$P = \frac{\sum_{i=1}^{n} |e_i \cap g_i|}{\sum_{i=1}^{n} |e_i|} \tag{8}$$

For Recall:

$$R = \frac{\sum_{i=1}^{n} |e_i \cap g_i|}{\sum_{i=1}^{n} |g_i|} \tag{9}$$

For F-score:

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \tag{10}$$

And we use NLPCC 2018 official evaluation criteria $F_{0.5}$, which means the importance of precision is twice the recall.

D. RESULTS

As table 2 shows, we compare our results with the results of the top three teams' work in the NLPCC 2018 shared task and some researches recently. From the results shown in the table, on a single model, our experimental results are 3.3 points higher than the first-ranked team, and on ensemble model, our experimental results even amazingly achieve 5.81 points higher than the first-ranked team. Compared to state-of-art work, our results are still insufficient, but the gap is not too big (near 1.25 points).And in terms of recall rate, our result is 0.13 points higher than it. Our experimental process is detailed as follows:

First of all, we are trying to find out which of the different granularities of word segmentation, namely character level, sub-word level and word level, works best. The results in Table 3 show that character-level word segmentation works best in this task. Therefore, the character-level Transformer is used as the experimental baseline in subsequent experiments.

Next, as is illustrated in table 4, we leverage BERT and its variants pre-trained model as the encoder in our structure, and analyze how different pre-trained model affect our results. Among them, alBERT reduces the amount of parameters on the basis of BERT and is a lightweight BERT. RoBERTa uses a larger amount of data, batch size, and epoch on the basis of BERT, removes the NSP task and adds a

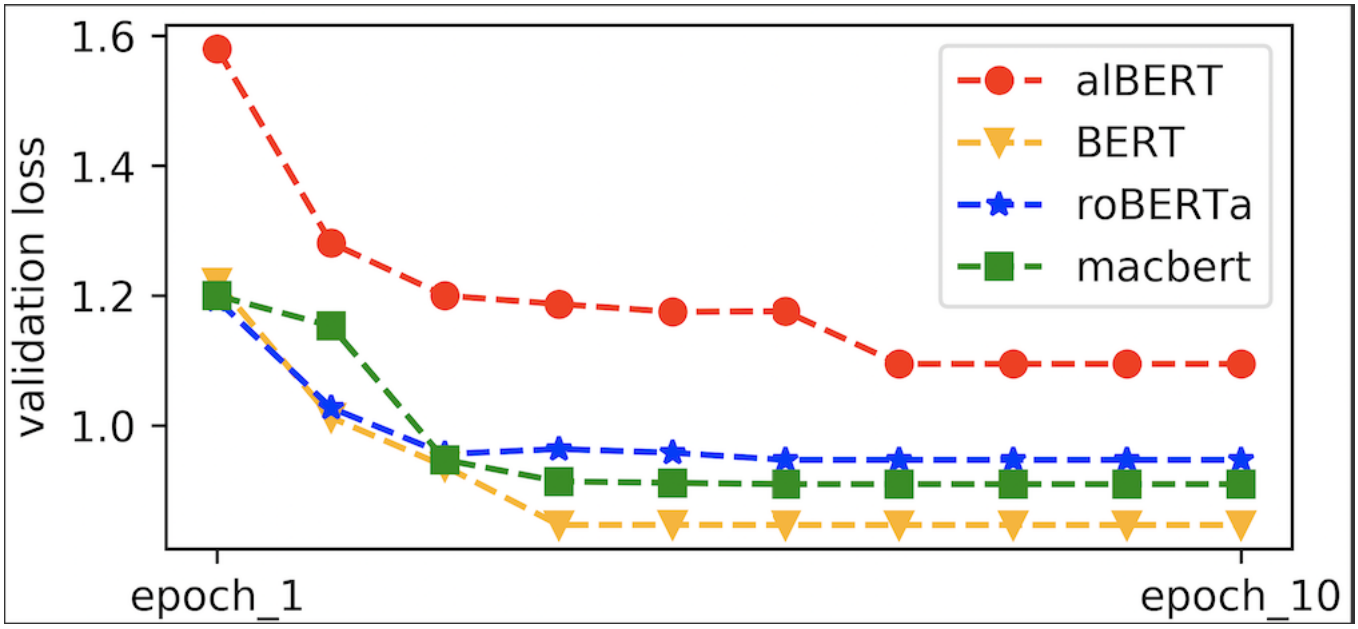

Fig. 3. The curve of the relationship between the number of iterations and the loss on the validation set.

dynamic mask mechanism. MacBERT improves the original masking strategy of BERT and proposes the Whole Word Masking strategy, which simultaneously masks the characters belonging to the same word during random masking. At the same time, the masked part is no longer represented by the [MASK] mark, but is randomly replaced by other words. For each of those BERT-variant pre-trained model, shown in figure 3, we record the curve of the relationship between the number of epoch and the loss on the validation set, which shows that, when epoch is close to 10, the loss tends to be stable, and also, that is the reason why we set the parameter epoch to 10.

The results indicate that MacBERT's performance is prior to other Pre-trained model in our CGEC system, for this reason, we use MacBERT as the pre-training model in the following experiments.

We train a character-level plain Transformer as our baseline for comparison. Next, we add spelling checker to the experiment and the result is slightly improved (1.28 points), in addition, the use of data augmentation can slightly improve the results. Then we use MacBERT pre-trained model as encoder, the result shows that the method of adding a pre-trained model on the encoder side is much better than the normal Transformer encoder. Subsequently, we also added the spelling checker to the experiment, and the results showed that the result of adding the spelling checker was 0.82 higher than that of not adding it. Finally, inspired by Wang et al [41], we use four different random seeds to train our models, and in the inference stage, we specify multiple model files, in this attempt, we achieve the best results.

TABLE II
EXPERIMENTAL RESULTS OF OUR METHODS COMPARED WITH THE OTHER TEAMS' IMPLEMENT IN NLPCC 2018 SHARED TASK AND RECENT RESEARCHES.

System	P	R	$F_{0.5}$
[NLPCC 2018]			
Fu et al (2018) [18]	35.24	18.64	29.91
Zhou et al (2018) [34]	41.00	13.75	29.36
Ren et al (2018) [40]	41.73	13.08	29.02
Ren et al ensemble (2018) [40]	47.63	12.56	30.57
[SOTA result]			
Zhao et al (2020) [42]	**44.36**	22.18	**36.97**
[Recent Researches]			
Wang et al (2020) [41]	32.67	22.19	29.76
Wang et al ensemble (2020) [41]	41.94	22.02	35.51
Ours	39.15	20.67	33.21
Ours ensemble	42.04	**22.31**	35.72

TABLE III
TRANSFORMER MODEL RESULTS OF DIFFERENT GRANULARITY SEGMENTATION.

Transformer model	P	R	$F_{0.5}$
Word level model	22.15	10.32	18.02
Subword level model	23.20	10.89	18.92
Character level model	**28.92**	**14.42**	**24.08**

V. DISCUSSION

The method proposed in this paper makes full use of the powerful semantic representation of the BERT pre-training model, and uses probability methods in statistics to correct shallow errors in sentences. For these four types of error mentioned in table 1, our system made a great prediction. For example, figure 4 shows the predicted sentence (also the correct sentence) "我可以去你家吃饭吗?" after the input sentence "我坷以去你家吃饭了吗?", is processed by the system. First, correct the error type "S" through the statistical model, that is, change "坷以" to "可以". Then use the seq2seq model to correct the error type "R", that is, to delete the redundant " 了".

TABLE IV
A SERIES OF PRE-TRAINED BERT MODELS THAT AFFECT THE RESULTS.

Pre-trained model	P	R	$F_{0.5}$
BERT-initialize	37.42	18.99	31.34
alBERT-initialize	27.28	10.10	20.36
roBERTa-initialize	37.28	**20.84**	32.20
MacBERT-initialize	**38.47**	19.85	**32.39**

TABLE V
THE COMPARISON EXPERIMENT, IN WHICH SC REFERS TO SPELLING CHECKER, ENS REFERS TO MODEL FUSION (ENSEMBLE).

System	P	R	$F_{0.5}$
Transformer	28.92	14.42	24.08
Transformer + SC	29.31	16.48	25.36
Transformer + SC + HSK	29.45	17.06	25.71
MacBERT-initialize	38.47	19.85	32.39
MacBERT-initialize + SC	39.15	20.67	33.21
MacBERT-initialize + SC + HSK	39.21	20.75	33.29
MacBERT-initialize + ENS	41.13	18.97	33.34
MacBERT-initialize + SC + ENS	**42.04**	**22.31**	**35.72**

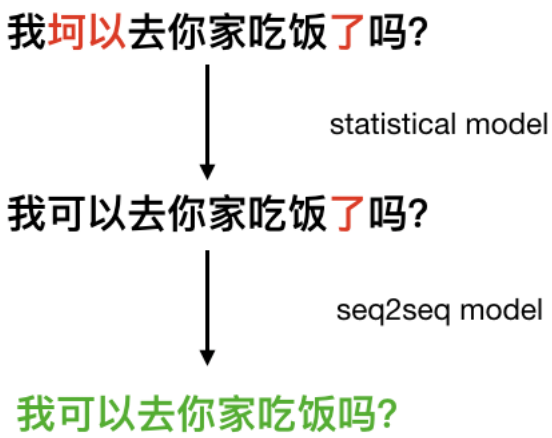

Fig. 4. An input sentence containing error type R("了") and error type S("坷以") corrected by our system.

But the shortcomings of our proposed method are also obvious, it takes too long to train the BERT pre-training model once, and it is inconvenient to adjust the parameters in time according to the results, although the effect is improved, the time cost is high. Moreover, There are long sentences in the test set, but we have to eliminate some long sentences (as we did in Chapter IV) in the process of constructing the training set, because of the limited GPU memory. However, the use of long sentences for training will further increase the requirements for GPU memory, maybe using a more advanced GPU will improve the experimental results and training speed at the same time.

VI. CONCLUSION

In this work, we leverage n-gram statistical language model as a spelling checker, and BERT-based pre-trained model as the encoder in seq2seq structure for Chinese grammatical error correction task, we have proved our method's powerful effect through our experiments. First, we use spelling checker based on n-gram language model to remove spelling errors. Next, we use the seq2seq model to correct grammatical errors and make sentences more fluent. The experimental results show that although our method still has a certain gap with sota work, it is also an effective method.

APPENDIX A
EXPERIMENTAL ENVIRONMENT

TABLE VI
HARDWARE INFORMATION AND SOFTWARE VERSION OF THE EXPERIMENTAL ENVIRONMENT.

Operating System	CentOS Linux release 7.6.1810 (Core)
CPU	Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz
Graphics Processing Unit	NVIDIA Corporation GP102 [GeForce GTX 1080 Ti]
Graphics Memory	11178 MiB
PyTorch Version	1.6.0+cu101
Python	3.6.5

APPENDIX B
MODEL AND TRAINING PARAMETERS

TABLE VII
DETAILS OF MODEL STRUCTURE AND TRAINING PROCESS.

Transformer	
Embedding dimension	512
FNN dimension	2048
Encoder layer	12
Decoder layer	12
Multi-heads	8
Initial learning rate	1e-5
Optimizer	Adam ($\beta_1 = 0.9$,$\beta_2 = 0.999$)
Learning rate scheduler	Fixed
Dropout	0.1
Batch size	2048
Max epoch	10

BERT-initialized	
Embedding dimension	512
FNN dimension	2048
Encoder layer	12
Decoder layer	12
Multi-heads	8
Initial learning rate	3e-5
Optimizer	Adam ($\beta_1 = 0.9$,$\beta_2 = 0.999$)
Learning rate scheduler	Fixed
Dropout	0.1
Batch size	32
Max epoch	10

REFERENCES

[1] T.-h. Kao, Y.-w. Chang, H.-w. Chiu, T.-H. Yen, J. Boisson, J.-c. Wu, and J. S. Chang, "Conll-2013 shared task: Grammatical error correction nthu system description," in *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, 2013, pp. 20–25.

[2] H. T. Ng, S. M. Wu, T. Briscoe, C. Hadiwinoto, R. H. Susanto, and C. Bryant, "The conll-2014 shared task on grammatical error correction," in *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, 2014, pp. 1–14.

[3] C. Bryant, M. Felice, Ø. E. Andersen, and T. Briscoe, "The BEA-2019 shared task on grammatical error correction," in *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 52–75. [Online]. Available: https://www.aclweb.org/anthology/W19-4406

[4] G. Rao, B. Zhang, E. Xun, and L.-H. Lee, "Ijcnlp-2017 task 1: Chinese grammatical error diagnosis," in *Proceedings of the IJCNLP 2017, Shared Tasks*, 2017, pp. 1–8.

[5] G. Rao, Q. Gong, B. Zhang, and E. Xun, "Overview of nlptea-2018 share task chinese grammatical error diagnosis," in *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, 2018, pp. 42–51.

[6] Y. Yang, P. Xie, J. Tao, G. Xu, L. Li, and S. Luo, "Alibaba at IJCNLP-2017 task 1: Embedding grammatical features into lstms for chinese grammatical error diagnosis task," in *Proceedings of the IJCNLP 2017, Shared Tasks, Taipei, Taiwan, November 27 - December 1, 2017, Shared Tasks*, C. Liu, P. Nakov, and N. Xue, Eds. Asian Federation of Natural Language Processing, 2017, pp. 41–46. [Online]. Available: https://www.aclweb.org/anthology/I17-4006/

[7] B. Zheng, W. Che, J. Guo, and T. Liu, "Chinese grammatical error diagnosis with long short-term memory networks," in *Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016)*, 2016, pp. 49–56.

[8] Y. Zhao, N. Jiang, W. Sun, and X. Wan, "Overview of the nlpcc 2018 shared task: Grammatical error correction," in *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer, 2018, pp. 439–445.

[9] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[10] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[12] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *arXiv preprint arXiv:1906.08237*, 2019.

[13] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[14] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for squad," *arXiv preprint arXiv:1806.03822*, 2018.

[15] S. Reddy, D. Chen, and C. D. Manning, "Coqa: A conversational question answering challenge," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 249–266, 2019.

[16] Z. Yuan and T. Briscoe, "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*, 2016, pp. 380–386.

[17] S. Chollampatt and H. T. Ng, "A multilayer convolutional encoder-decoder neural network for grammatical error correction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[18] K. Fu, J. Huang, and Y. Duan, "Youdao'Ă'Źs winning solution to the nlpcc-2018 task 2 challenge: a neural machine translation approach to chinese grammatical error correction," in *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer, 2018, pp. 341–350.

[19] M. Kaneko, M. Mita, S. Kiyono, J. Suzuki, and K. Inui, "Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction," *arXiv preprint arXiv:2005.00987*, 2020.

[20] Y. Kantor, Y. Katz, L. Choshen, E. Cohen-Karlik, N. Liberman, A. Toledo, A. Menczel, and N. Slonim, "Learning to combine grammatical error corrections," *arXiv preprint arXiv:1906.03897*, 2019.

[21] Z. Qiu and Y. Qu, "A two-stage model for chinese grammatical error correction," *IEEE Access*, vol. 7, pp. 146 772–146 777, 2019.

[22] S.-H. Wu, C.-L. Liu, and L.-H. Lee, "Chinese spelling check evaluation at SIGHAN bake-off 2013," in *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*. Nagoya, Japan: Asian Federation of Natural Language Processing, Oct. 2013, pp. 35–42. [Online]. Available: https://aclanthology.org/W13-4406

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[24] S. Rothe, S. Narayan, and A. Severyn, "Leveraging pre-trained checkpoints for sequence generation tasks," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 264–280, 2020.

[25] G. E. Heidorn, K. Jensen, L. A. Miller, R. J. Byrd, and M. S. Chodorow, "The epistle text-critiquing system," *IBM Systems Journal*, vol. 21, no. 3, pp. 305–326, 1982.

[26] N.-R. Han, M. Chodorow, and C. Leacock, "Detecting errors in english article usage with a maximum entropy classifier trained on a large, diverse corpus." in *LREC*, 2004.

[27] C. BROCKETT, "Correcting esl errors using phrasal smt techniques," in *Proc. 21th International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia, July 2006*, 2006, pp. 249–256.

[28] M. Felice, Z. Yuan, Ø. E. Andersen, H. Yannakoudakis, and E. Kochmar, "Grammatical error correction using hybrid systems and type filtering." Association for Computational Linguistics, 2014.

[29] Z. Xie, A. Avati, N. Arivazhagan, D. Jurafsky, and A. Y. Ng, "Neural language correction with character-based attention," *arXiv preprint arXiv:1603.09727*, 2016.

[30] C. Sun, X. Jin, L. Lin, Y. Zhao, and X. Wang, "Convolutional neural networks for correcting english article errors," in *Natural Language Processing and Chinese Computing*. Springer, 2015, pp. 102–110.

[31] T. Ge, F. Wei, and M. Zhou, "Fluency boost learning and inference for neural grammatical error correction," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1055–1065.

[32] Y. Yang, P. Xie, J. Tao, G. Xu, L. Li, and L. Si, "Alibaba at ijcnlp-2017 task 1: Embedding grammatical features into lstms for chinese grammatical error diagnosis task," in *Proceedings of the IJCNLP 2017, Shared Tasks*, 2017, pp. 41–46.

[33] B. Zheng, W. Che, J. Guo, and T. Liu, "Chinese grammatical error diagnosis with long short-term memory networks," in *Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016)*. Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 49–56. [Online]. Available: https://www.aclweb.org/anthology/W16-4907

[34] J. Zhou, C. Li, H. Liu, Z. Bao, G. Xu, and L. Li, "Chinese grammatical error correction using statistical and neural models," in *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer, 2018, pp. 117–128.

[35] Y. Cheng and M. Duan, "Chinese grammatical error detection based on BERT model," in *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*. Suzhou, China: Association for Computational Linguistics, Dec. 2020, pp. 108–113. [Online]. Available: https://aclanthology.org/2020.nlptea-1.15

[36] Y. Cao, L. He, R. Ridley, and X. Dai, "Integrating BERT and score-based feature gates for Chinese grammatical error diagnosis," in *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*. Suzhou, China: Association for Computational Linguistics, Dec. 2020, pp. 49–56. [Online]. Available: https://aclanthology.org/2020.nlptea-1.7

[37] D. Liang, C. Zheng, L. Guo, X. Cui, X. Xiong, H. Rong, and J. Dong, "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: Association for Computational Linguistics, Dec. 2020, pp. 57–66. [Online]. Available: https://aclanthology.org/2020.nlptea-1.8

[38] H. Zan, Y. Han, H. Huang, Y. Yan, Y. Wang, and Y. Han, "Chinese grammatical errors diagnosis system based on BERT at NLPTEA-2020 CGED shared task," in *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*. Suzhou, China: Association for Computational Linguistics, Dec. 2020, pp. 102–107. [Online]. Available: https://aclanthology.org/2020.nlptea-1.14

[39] A. S. Koyuncugil and N. Ozgulbas, "Statistical roots of machine learning, deep learning, artificial intelligence, big data analytics and data mining," in *Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering and Computer Science*, 2019, pp. 22–24.

[40] H. Ren, L. Yang, and E. Xun, "A sequence to sequence learning for chinese grammatical error correction," in *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer, 2018, pp. 401–410.

[41] H. Wang, M. Kurosawa, S. Katsumata, and M. Komachi, "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*, 2020. [Online]. Available: https://www.aclweb.org/anthology/2020.aacl-main.20

[42] Z. Zhao and H. Wang, "Maskgec: Improving neural grammatical error correction via dynamic masking," in *Proceedings of the AAAI*

Conference on Artificial Intelligence, vol. 34, no. 01, 2020, pp. 1226–1233.

GUANLIN HE was born in Anshan, Liaoning, China in 1996. He received the B.S. degree from the University of Science and Technology LiaoNing, where he is currently pursuing the M.S. degree. His mentor's main research interests include natural language processing and deep learning.

CHENGYING CHI is a professor of computer science at the University of Science and Technology, Liaoning. Her current research interests are information retrieval, data mining, and distributed database systems.

YUNYUN ZHAN is a student of College of Science and Health, Technological University Dublin, Dublin, D08 X622, Ireland.