TMU-NLP System Using BERT-based Pre-trained Model to the NLP-TEA CGED Shared Task 2020

Hongfei Wang and Mamoru Komachi
Tokyo Metropolitan University
wang-hongfei@ed.tmu.ac.jp,komachi@tmu.ac.jp

Abstract

In this paper, we introduce our system for NLPTEA 2020 shared task of Chinese Grammatical Error Diagnosis (CGED). In recent years, pre-trained models have been extensively studied, and several downstream tasks have benefited from their utilization. In this study, we treat the grammar error diagnosis (GED) task as a grammatical error correction (GEC) problem and use a method that incorporates a pre-trained model into an encoder-decoder model to solve this problem.

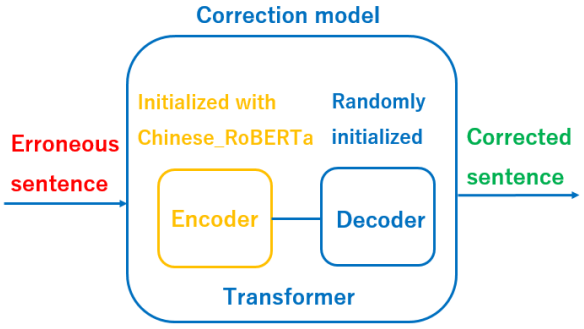

Figure 1: The structure of our system.

1 Introduction

In the CGED task, given a source sentence contains grammatical errors, systems are needed to output four kinds of results: detection level, identification level, position level, and correction level. We treat the CGED task as a GEC problem, so our system will output the corrected sentence directly. Then we use a post-processing method to generate detection, identification, and position level results.

GEC can be regarded as a sequence-to-sequence task. GEC systems receive an erroneous sentence written by a language learner and output the corrected sentence. In previous studies that adopted neural models for Chinese GEC (Ren et al., 2018; Zhou et al., 2018), the performance was improved by initializing the models with a distributed word representation, such as Word2Vec (Mikolov et al., 2013). However, in these methods, only the embedding layer of a pre-trained model was used to initialize the models.

In addition, Chinese GEC remains challenging because Chinese is a complex language. For example, there are more than 10,000 Chinese characters, and the glyphs or pronunciation of several Chinese characters are similar. Therefore, character errors introduced by Chinese learners can be variable; further, for the Chinese GEC tasks, it is necessary to utilize a pre-trained model, which captures additional information about a language. In recent years, pre-trained models based on Bidirectional Encoder Representations from Transformers (BERT) have been studied extensively (Devlin et al., 2019; Liu et al., 2019), and the performance of many downstream Natural Language Processing (NLP) tasks has been dramatically improved by utilizing these pre-trained models. In order to learn existing knowledge of a language, a BERT-based pre-trained model is trained on a large-scale corpus using the encoder of Transformer (Vaswani et al., 2017). Subsequently, for a downstream task, a neural network model is initialized with the weights learned by a pre-trained model that has the same structure and is fine-tuned on training data of the downstream task. The performance is expected to improve by using this two-stage method because downstream tasks are informed by the knowledge learned by the pre-trained model.

In this study, as shown in Figure 1, we develop a Chinese GEC model based on Transformer by initializing the encoder of Transformer with a Chinese BERT-based pre-trained model and then fine-tuning the correction model on Chinese GEC and GED corpus. Our system achieves F_1 scores of 82.00, 52.66, 22.24, 14.17 on detection level, iden-

Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications, pages 87–90
Suzhou, China, December 4, 2020.

tification level, position level and correction level evaluation.

2 Related Work

In Building Educational Applications (BEA) 2019 (Bryant et al., 2019), several teams attempted to incorporate BERT into their correction models. Kaneko et al. (2019) first fine-tuned BERT on a learner corpus and then incorporated the word probability provided by BERT into re-ranking features. Using BERT for re-ranking features, they obtained an approximately 0.7 point improvement of the $F_{0.5}$ score. Kantor et al. (2019) used BERT to solve the GEC task by iteratively querying BERT as a black box language model. They added a [MASK] token into source sentences and predicted the word represented by the [MASK] token. If the word probability predicted by BERT exceeded the threshold, the word was output as a correction candidate. Using BERT, they obtained a 0.27 point improvement of the $F_{0.5}$ score. These studies show that BERT helps improve the performance of a correction model; however, this improvement was marginal, and they did not explore the use of pre-trained models for weight initialization.

3 Method

We adopt the method from Wang et al. (2020) to construct our correction model. Additional details are introduced in the following sections.

3.1 Chinese Pre-trained Model

We use a BERT-based model as our pre-trained model. BERT is mainly trained with a task called Masked Language Model. In the Masked Language Model task, some tokens in a sentence are replaced with masked tokens ([MASK]), and the model needs to predict the replaced tokens.

In this study, we use the Chinese-RoBERTa-wwm-ext model provided by Cui et al. (2019). The main differences between Chinese-RoBERTa-wwm-ext and original BERT are as follows:

Whole Word Masking (WWM) Devlin et al. (2019) proposed a new masking method called Whole Word Masking (WWM) after proposing their original BERT, which masks entire words instead of subwords. They demonstrated that the original prediction task that only masks subwords is easy and that the performance has been improved by masking entire words. Therefore, Cui et al. (2019) adopted this method to train their Chinese pre-trained models. It should be noted that they used WordPiece (Wu et al., 2016) to preprocess Chinese sentences, and Chinese sentences are segmented into characters (not subwords) by WordPiece. Therefore, in WWM, when a Chinese character is masked, other Chinese characters that belong to the same word should also be masked. Table 1 shows an example of WWM.

[Original Sentence] 然后 **准备** 别 的 **材料** 。
[Original BERT] 然 后 **准 [MASK]** 别 的 **[MASK] 料** 。
[Whole Word Masking] 然 后 **[MASK] [MASK]** 别 的 **[MASK] [MASK]** 。
[English Translation] Then prepare for other materials.

Table 1: Example of the difference between original BERT and Whole Word Masking for Chinese sentences. The original sentence is segmented into words, whereas in original BERT and whole word masking, the sentence is segmented into characters.

Training Strategy Cui et al. (2019) followed the training strategy studied by Liu et al. (2019). Although Cui et al. (2019) referred to the training strategy from Liu et al. (2019), there are still some differences between them (e.g., they did not use dynamic masking).

Training Data In addition to Chinese Wikipedia (0.4B tokens) that was originally used to train BERT, an extended corpus (5.0B tokens), which consists of Baidu Baike (a Chinese encyclopedia) and QA data, was also used. The extended corpus has not been released due to a license issue.

3.2 Grammatical Error Correction Model

In this study, we use Transformer as our correction model. Transformer has shown excellent performance in sequence-to-sequence tasks such as machine translation and has been widely adopted in recent English GEC studies (Kiyono et al., 2019; Junczys-Dowmunt et al., 2018).

However, a BERT-based pre-trained model only uses the encoder of Transformer; therefore, it can not be directly applied to sequence-to-sequence tasks that require both an encoder and a decoder, such as GEC. Hence, we initialize the encoder of Transformer with the parameters learned by Chinese-RoBERTa-wwm-ext, and the decoder is initialized randomly. Finally, we fine-tune this initialized model on Chinese GEC data and use it as our correction model.

Parameters	
Architecture	Encoder (12-layer), Decoder (12-layer)
Learning rate	3×10^{-5}
Batch size	32
Optimizer	Adam
Loss function	cross-entropy
Dropout	0.1

Table 2: Training details for our model.

Evaluation Type	P	R	F_1
Detection	94.04	72.70	82.00
Identification	69.80	42.28	52.66
Position	34.60	16.39	22.24
Correction	22.58	10.32	14.17

Table 3: Experimental results of our model.

4 Experiments

4.1 Experimental Settings

Training Data We use the training data provided by the NLPCC 2018 Grammatical Error Correction shared task. In this task, approximately one million sentences from the language learning website Lang-8[1] are used as training data. We first segment all sentences into characters because the Chinese pre-trained model we used is character-based. In the GEC task, source and target sentences do not tend to change significantly. Considering this, we filter the training data by excluding sentence pairs that meet the following criteria: i) the source sentence is identical to the target sentence; ii) the edit distance between the source sentence and the target sentence is greater than 15; iii) the number of characters of the source sentence or the target sentence exceeds 64. Once the training data are filtered, we obtain 971,318 sentence pairs.

We also use the training data provided by the CGED task this year and in previous years. By dividing the sentence units into separate sentences, we finally obtain 56,000 sentence pairs.

Validation Data We randomly extract 5,000 sentences from the CGED training data as the validation data.

Test Data We use the official test data, which contain 2,456 sentence units. We also divide the sentence units of the test data and obtain 2,745 sentences.

Implementation We implement the Transformer model using fairseq 0.8.0.[2] and load the pre-trained model using pytorch_transformer 2.2.0.[3]

We train our model on the NLPCC training data for 20 epochs and then on CGED training data for another 20 epochs. The first 20 epochs are validated on NLPCC validation data, and the second 20 epochs are validated on CGED validation data.

We select the best model from all 40 epochs according to the loss function (cross-entropy).

We train four models using different random seeds and combine them into a 4-ensemble model.

More details on the training are provided in Table 2.

Evaluation We post-process our system outputs and obtain final outputs.

We first recover the sentence units of test source and system outputs. Then the detection result is decided by whether the output sentence is identical to the source. For identification and position, we use edit-distance to obtain the difference between source and output.

4.2 Evaluation Results

Table 3 summarizes the experimental results of our model. In all 44 submitted system results, our system is 33rd, 33rd, and 31st on detection, identification, and position evaluation. In all 25 results for the correction level, our system is 19th.

4.3 Case Analysis

Table 4 shows the sample outputs.

In the first example, both the gold edit and our system correct the spelling error 果 (fruits) to 课 (class). It appears that our system can accurately correct the error according to context because the word 果 (fruits) rarely comes after the word 汉语 (Chinese).

In the second example, the output of our system is more fluent, although the gold edit does not make any alterations to the source sentence. Our system changes the word 赏玩 into 欣赏, although the two words have a similar meaning: enjoy, a native speaker often uses 欣赏 rather than 赏玩 when the object is 景色 (scenery). It appears that our system can capture the collocation efficiently because the pre-trained model is trained with a large-scale corpus.

In the third example, our system changes the word selection error 殷勤 (attentive) to 勤奋 (working hard), which is irrelated to the context. We

[1] https://lang-8.com/
[2] https://github.com/pytorch/fairseq
[3] https://github.com/huggingface/transformers

src	星期五中午十二点钟我上汉语**果**。	我们可以悠闲地**赏玩**风景。
gold	星期五中午十二点钟我上汉语**课**。	我们可以悠闲地**赏玩**风景。
Our system	星期五中午十二点钟我上汉语**课**。	我们可以悠闲地**欣赏**风景。
Translation	I have a Chinese class at 12 o'clock on Friday.	We can enjoy the scenery leisurely.
src	他知道了我要换房子，因为我的租金太高。所以他请我来跟他一起住以便分担租金。我觉得他很**殷勤**。	
gold	……我觉得他很**善良**。	
Our system	……我觉得他很**勤奋**。	
Translation	He knew that I was going to change house because my rent was too high. So he invited me to live with him in order to share the rent. I think he is very kind.	

Table 4: Source sentence, gold edit, and output of our system.

think that this is because we divide the sentence unit, and so our system can not refer to the context that is necessary to correct this error.

5 Conclusion

In this study, we initialized the encoder of the Transformer with a Chinese pre-trained model and used this system to challenge the CGED task.

References

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *BEA@ACL*.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for Chinese BERT. *ArXiv*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. In *NAACL-HLT*.

Masahiro Kaneko, Kengo Hotate, Satoru Katsumata, and Mamoru Komachi. 2019. TMU transformer system using BERT for re-ranking at BEA 2019 grammatical error correction on restricted track. In *BEA@ACL*.

Yoav Kantor, Yoav Katz, Leshem Choshen, Edo Cohen-Karlik, Naftali Liberman, Assaf Toledo, Amir Menczel, and Noam Slonim. 2019. Learning to combine grammatical error corrections. In *BEA@ACL*.

Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. An empirical study of incorporating pseudo data into grammatical error correction. In *EMNLP-IJCNLP*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke S. Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *ArXiv*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Hongkai Ren, Liner Yang, and Endong Xun. 2018. A sequence to sequence learning for Chinese grammatical error correction. In *NLPCC*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Hongfei Wang, Michiki Kurosawa, Satoru Katsumata, and Mamoru Komachi. 2020. Chinese grammatical correction using BERT-based pre-trained model. In *AACL-IJCNLP*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*.

Junpei Zhou, Chen Li, Hengyou Liu, Zuyi Bao, Guangwei Xu, and Linlin Li. 2018. Chinese grammatical error correction using statistical and neural models. In *NLPCC*.