Master's Thesis

Chinese Grammatical Error Correction Using Pre-trained Model

Hongfei Wang

April 15, 2021

Graduate School of Systems Design
Tokyo Metropolitan University

A Master's Thesis
submitted to Graduate School of Systems Design,
Tokyo Metropolitan University
in partial fulfillment of the requirements for the degree of
Master of Computer Science

Hongfei Wang

Thesis Committee:

Associate Professor Mamoru Komachi	(Supervisor)
Professor Toru Yamaguchi	(Co-supervisor)
Professor Yasufumi Takama	(Co-supervisor)

Chinese Grammatical Error Correction Using Pre-trained Model*

Hongfei Wang

Abstract

In recent years, pre-trained models have been extensively studied, and several downstream tasks have benefited from their utilization. In this study, we develop the Chinese GEC models based on Transformer with a pre-trained model using two methods: first, by initializingthe encoder with the pre-trained model (BERT-encoder); second, by utilizing the technique proposed by Zhu et al. (2020), which uses the pre-trained model for additional features (BERT-fused). On the Natural Language Processing and Chinese Computing (NLPCC) 2018 Grammatical Error Correction shared task test set, our single models obtain $F_{0.5}$ scores of 33.66 and 30.63 respectively, which is higher than the performance of ensemble models developed by the top team of the shared task. Moreover, using a 4-ensemble model, we obtain an $F_{0.5}$ score of 37.47, which is a state-of-the-art result of the task. We annotate the error types of the development data; the results show that word-level errors dominate all error types, and sentence-level errors remain challenging and require a stronger approach.

Keywords:

Chinese grammatical error correction, pre-trained models, NLP application

*Master's Thesis, Graduate School of Systems Design,
Tokyo Metropolitan University, April 15, 2021.

Contents

List of Figures

1 Introduction

Grammatical error correction (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 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. 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. Using this two-stage method, the performance is expected to improve because downstream tasks are informed by the knowledge learned by the pre-trained model.

Recent works (Kaneko et al., 2020; Kantor et al., 2019) show that BERT helps improve the performance on the English GEC task. As the Chinese pre-trained models are developed and released continuously (Cui et al., 2020; Zhang et al., 2019), the Chinese GEC task may also benefit from using those pre-trained models.

In this study, as shown in Figure 1.1, we develop a Chinese GEC model based on Transformer with a pre-trained model using two methods: first, by initializing

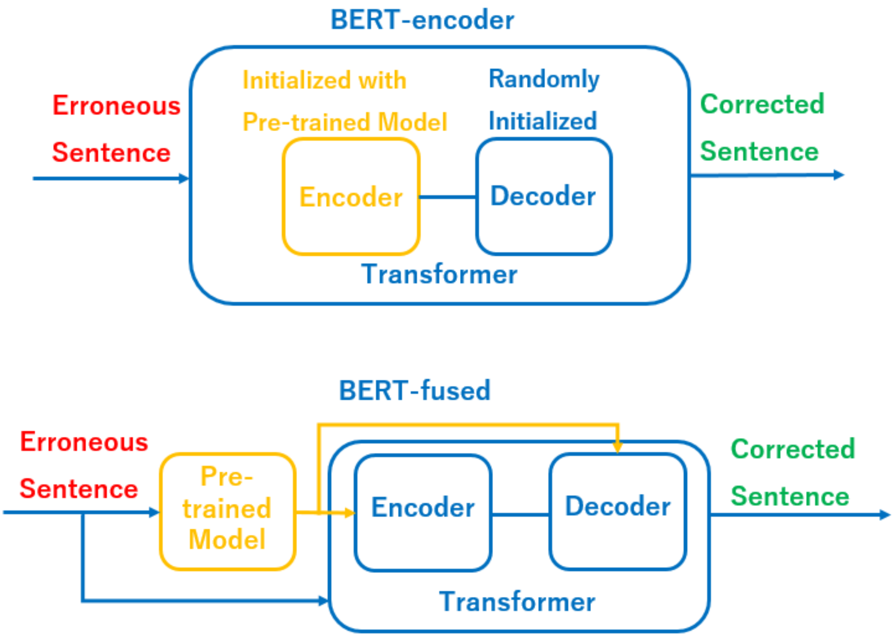

Figure 1.1: Two methods for incorporating a pre-trained model into the GEC model.

the encoder with the pre-trained model (BERT-encoder); second, by utilizing the technique proposed by Zhu et al. (2020), which uses the pre-trained model for additional features (BERT-fused); on the Natural Language Processing and Chinese Computing (NLPCC) 2018 Grammatical Error Correction shared task test dataset (Zhao et al., 2018), our single models obtain $F_{0.5}$ scores of 33.66 and 30.63 respectively, which is higher than the performance of ensemble models developed by the top team of the shared task. Moreover, using a 4-ensemble model, we obtain an $F_{0.5}$ score of 37.47, which is a state-of-the-art result of the Chinese GEC task. We also annotate the error types of the development data; the results show that word-level errors dominate all error types, and sentence-level errors remain challenging and require a stronger approach.

This thesis is organized into the following chapters. Chapter 2 describes the background knowledge of the encoder-decoder model, the Transformer model and the pre-trained model. Chapter 3 provides an overview of the related works. Chapter 4 describes how we construct our Chinese grammatical error correction models. Chapter 5 describes the experimental settings, the evaluation results, and the comparison with previous works. Chapter 6 provides the sample sentences of our model and the analysis of error types. Chapter 7 concludes the thesis.

2 Background of Encoder-Decoder, Transformer and BERT

In this chapter, we will simply introduce the core mechanisms of the encoder-decoder models, the Transformer model and the BERT model.

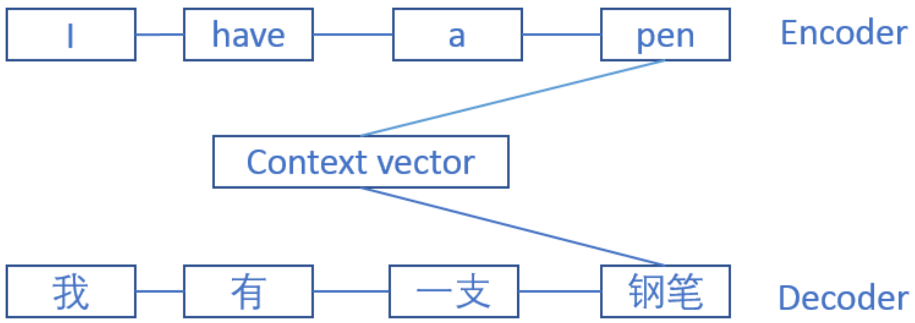

Figure 2.1: The model state when decoder outputs the token 钢笔.

2.1 Encoder-Decoder Models

There are many NLP tasks that can be treated as a sequence-to-sequence problem. For example, in machine translation tasks, the input is a sentence of source language, and the output is a sentence of target language. And in question answering tasks, the input is a sentence of question, and the output is a sentence of answer. The encoder-decoder models are designed to solve these sequence-to-sequence problems. Sutskever et al. (2014) constructed the encoder-decoder using Recurrent Neural Network (RNN). The encoder receives a source sequence and encodes it from left to right. The last hidden state of the encoder is used as a context vector. Then the decoder outputs the tokens of the target sequence one by one according to the context vector and hidden states of words that have been output by decoder. Take the neural machine translation (NMT) as an example: Assume that we want to translate the English sentence *I have a pen* into the Chinese sentence 我 有 一支 钢笔. The decoder outputs the token 钢笔 according to the context vector (i.e. the hidden state of *pen*) and the hidden state of token 一支 (illustrated in Figure 2.1).

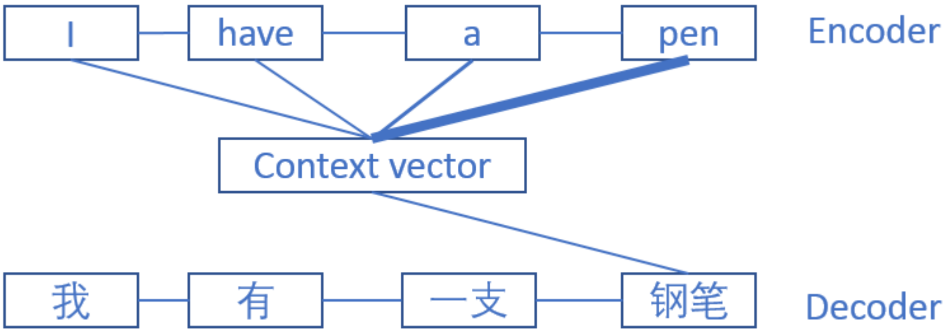

Figure 2.2: The model state when decoder outputs the token 钢笔. The lines between context vector and each token in encoder represent the attention.

Since the encoder-decoder models need to compress all information from the source sequence, it is difficult for RNN to encode long sequences. To solve this problem, Bahdanau et al. (2015) proposed the attention mechanism. This mechanism is based on a concept that when humans output each token in the target sequence, they concentrate on the different parts of the source sequence. Consider the NMT example again. When we output the token 钢笔 (which means pen in English) of the target sequence, it is natural to concentrate more on the token *pen*. And the attention is a score that represents how well the two tokens match. For example, the attention score for the token pair (pen, 钢笔) should be higher than other (source-token, 钢笔) pairs (illustrated in Figure 2.2). By doing so, the context vector changes dynamically, and hence the decoder can concentrate on different parts of the source sequence when outputs each token of a target sequence.

2.2 Transformer

The RNN models are restricted by the sequential computation: the RNN processes the sentence from left to right, and the hidden states of words depend on the previous hidden state. This sequential nature precludes parallelization within training examples, especially for long sequences. Considering this inherent nature of RNN, Vaswani et al. (2017) developed a encoder-decoder model called Transformer based solely on attention mechanism instead of complex RNN that were broadly adopted by previous encoder-decoder models. The Transformer allows for more training parallelization, and can capture the global dependencies efficiently.

There are three kinds of attention mechanisms in Transformer: self-attention for the encoder, self-attention for the decoder, and encoder-decoder attention.

- Self-attention for the encoder is based on a concept that we should refer to other parts of the sentence when we encode a token of the sentence. Consider the example *I have a pen* again. When we encode the token *pen*, except the token itself, we concentrate more on the token *a* than other tokens because the token *a* is the article that modifies *pen*, hence they have stronger relation than others (illustrated in Figure 2.3).

- The only difference between self-attention for the decoder and for the encoder is that when we output the current token, tokens after the current token should be masked in training because, in a practical scene, the model outputs the target sequence from left to right and the model does not receive the information from the right side of the current token.

- The encoder-decoder attention is just as we mentioned in section 2.1.

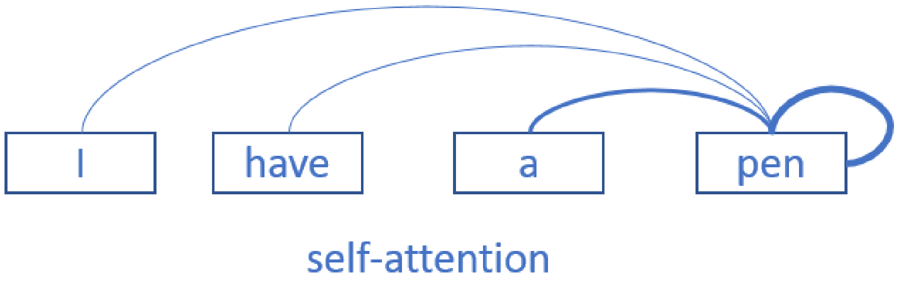

Figure 2.3: The self-attention when encodes the token *pen*. The curves represent attention.

2.3 BERT

BERT is a pre-trained model developed by Devlin et al. (2019) using the encoder side of the Transformer.

The main goal of training a pre-trained model is to learn existing knowledge of a language and the downstream tasks can benefit from this knowledge. The pre-trained model is first trained on a large-scale corpus using pre-training tasks. 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. Using this two-stage method, the performance is expected to improve because downstream tasks are informed by the knowledge learned by the pre-trained model.

Devlin et al. (2019) designed two pre-training tasks for BERT: The first one is the Masked Language Model (MLM) task, which is inspired by the cloze task. In the MLM task, some tokens in a sentence are replaced with masked tokens ([MASK]), and the model has to predict the replaced tokens. Unlike previous works, they do not use unidirectional pre-training tasks because they argue that bidirectional tasks can receive information from both left and right sides hence can capture the context more efficiently. The second one is the Next Sentence Prediction (NSP) task. This task is designed to train a model that understands sentence relationships. The model takes sentence A and sentence B as an input and then it predicts whether sentence B is the next sentence of A or not.

3 Related Works of GEC

In this chapter, we will introduce previous works of Chinese GEC, and English GEC which utilize BERT.

3.1 Chinese Grammatical Error Correction

Given the success of the shared tasks on English GEC at the Conference on Natural Language Learning (CoNLL) (Ng et al., 2013, 2014), a Chinese GEC shared task was performed at the NLPCC 2018. In this task, approximately one million sentences from the language learning website Lang-8* were used as training data and two thousand sentences from the PKU Chinese Learner Corpus (Zhao et al., 2018) were used as test data. Here, we briefly describe the three methods with the highest performance.

First, Fu et al. (2018) combined a 5-gram language model-based spell checker with subword-level and character-level encoder-decoder models using Transformer to obtain five types of outputs. Then, they re-ranked these outputs using the language model. Although they reported a high performance, several models were required, and the combination method was complex.

Second, Ren et al. (2018) utilized a convolutional neural network (CNN), such as in Chollampatt and Ng (2018). However, because the structure of the CNN is different from that of BERT, it cannot be initialized with the weights learned by the BERT.

Last, Zhao and Wang (2020) proposed a dynamic masking method that replaces the tokens in the source sentences of NLPCC 2018 Grammar Error Correction shared task training data with other tokens (e.g. [PAD] token). They achieved

*https://lang-8.com/

comparatively high results on the shared task without using any extra knowledge. This is a data augmentation method that can be a supplement for our study.

3.2 English Grammatical Error Correction Using BERT

In Building Educational Applications (BEA) 2019 English Grammatical Error Correction Shared Task (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.

Kaneko et al. (2020) first fine-tuned BERT using a Grammatical Error Diagnosis task, and then incorporated the fine-tuned BERT into the correction model by using method proposed by Zhu et al. (2020). They showed the effectiveness of BERT on the English GEC task, and achieved comparatively high results.

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.

4 Method of Incorporating the Chinese Pre-trained Model into GEC Model

In this chapter, we will describe the details of the Chinese pre-trained Model used in this work, and the two methods about how we incorporate the Chinese pre-trained Model into our GEC models.

4.1 Chinese Pre-trained Model

In this study, we use the Chinese-RoBERTa-wwm-ext model provided by Cui et al. (2020), which is a BERT-based pre-trained model. 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. (2020) adopted this method to train their Chinese pre-trained models. In WWM, when a Chinese character is masked, other Chinese characters that belong to the same word should also be masked. Table 4.1 shows an example of WWM.
- **Training Strategy**: Cui et al. (2020) followed the training strategy studied by Liu et al. (2019). Although Cui et al. (2020) referred to the training

[*] `https://github.com/google-research/bert`

[Source 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 4.1: Example of the difference between original BERT and Whole Word Masking for Chinese sentences. The source sentence is segmented into words, whereas in original BERT and whole word masking, the sentence is segmented into characters.

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, extended corpus (5.0B tokens), which consists of Baidu Baike (a Chinese encyclopedia) and QA data, was also used. Extended corpus has not been released due to a license issue.

4.2 Grammatical Error Correction Model

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

However, a BERT-based pre-trained model only uses the encoder of Transformer; therefore, it cannot be directly applied to sequence-to-sequence tasks that require both an encoder and a decoder, such as GEC. Hence, we incorporate the encoder-decoder model with the pre-trained model in two ways as described in the following subsections.

4.2.1 BERT-encoder

We initialize the encoder of Transformer with the parameters learned by Chinese-RoBERTa-wwm-ext; the decoder is initialized randomly. Finally, we fine-tune the initialized model on Chinese GEC data.

4.2.2 BERT-fused

Zhu et al. (2020) proposed a method that uses a pre-trained model as the additional features. In this method, input sentences are fed into the pre-trained model and the pre-trained model outputs the encoded vector representations. Then, the representations from the pre-trained model will interact with the encoder and decoder by using attention mechanism. Kaneko et al. (2020) verified the effectiveness of this method on English GEC tasks.

5 Experiments

In this chapter, we will provide the details of our experiments and the comparison with previous works.

5.1 Experimental Settings

5.1.1 Data

In this study, we use the data provided by the NLPCC 2018 Grammatical Error Correction shared task. We first segment all sentences into characters because the Chinese pre-trained model we used is character-based.

The training data consist of 1.2 million sentence pairs extracted from the language learning website Lang-8.

Because the NLPCC 2018 Grammatical Error Correction shared task did not provide development data, we opted to randomly extract 5,000 sentences from the training data as the development data following Ren et al. (2018).

The test data consist of 2,000 sentences extracted from the PKU Chinese Learner Corpus. According to Zhao et al. (2018), the annotation guidelines follow the minimum edit distance principle (Nagata and Sakaguchi, 2016), which selects the edit operation that minimizes the edit distance from the original sentence.

5.1.2 Model

We implement the Transformer model using fairseq 0.8.0.[*] and load the pre-trained model using pytorch_transformer 2.2.0.[†]

We then train the following models based on Transformer.

[*] https://github.com/pytorch/fairseq

[†] https://github.com/huggingface/transformers

- **Baseline**: A plain Transformer model that is initialized randomly without using a pre-trained model.
- **BERT-encoder**: The correction model introduced in Section 4.2.1.
- **BERT-fused**: The correction model introduced in Section 4.2.2. We use the implementation provided by Zhu et al. (2020).[‡]

Finally, we train a 4-ensemble BERT-encoder model and a 4-ensemble BERT-fused model.

More details on the training are provided in the Table 5.1.

5.1.3 Evaluation

As the evaluation is performed on word-unit, we strip all delimiters from the system output sentences and segment the sentences using the pkunlp[§] provided in the NLPCC 2018 Grammatical Error Correction shared task.

Based on the setup of the NLPCC 2018 Grammatical Error Correction shared task, the evaluation is conducted using *MaxMatch* (M2).[¶] The *MaxMatch* algorithm computes the phrase-level edits between the source sentence and the system output. Then it finds the overlaps between the system edits and gold edits.

5.2 Evaluation Results

Table 5.2 summarizes the experimental results of our models. We run the single models four times, and report the average score. For comparison, we also cite the best single model result of Zhao and Wang (2020) and the results of the models developed by two teams in the NLPCC 2018 Grammatical Error Correction shared task.

The performances of BERT-encoder and BERT-fused are significantly superior to that of the baseline model and are comparable to those achieved by the two

[‡] `https://github.com/bert-nmt/bert-nmt`

[§] `http://59.108.48.12/lcwm/pkunlp/downloads/libgrass-ui.tar.gz`

[¶] `https://github.com/nusnlp/m2scorer` teams in the NLPCC 2018 Grammatical Error Correction shared task, indicating the effectiveness of adopting the pre-trained model.

The BERT-encoder (4-ensemble) model yields an $F_{0.5}$ score nearly 7 points higher than the highest-performance model in the NLPCC 2018 Grammatical Error Correction shared task. However, there is no improvement for the BERT-fused (4-ensemble) model compared with the single BERT-fused model. We find that the performance of the BERT-fused model depends on the warm-up model. Compared with Kaneko et al. (2020) using a state-of-the-art model to warm-up their BERT-fused model, we did not use a warm-up model in this work. The performance noticeably drops when we try to warm-up the BERT-fused model from a weak baseline model, therefore, the BERT-fused model may perform better when warmed-up from a stronger model (e.g., the model proposed by Zhao and Wang (2020)).

For Zhao and Wang (2020), they achieved best recall and comparatively high $F_{0.5}$ score using single model.

Baseline	
Architecture	Encoder (12-layer), Decoder (12-layer)
Learning rate	1×10^{-5}
Batch size	32
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1 \times 10^{-8}$)
Max epochs	20
Loss function	cross-entropy
Dropout	0.1
BERT-encoder	
Architecture	Encoder (12-layer), Decoder (12-layer)
Learning rate	3×10^{-5}
Batch size	32
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1 \times 10^{-8}$)
Max epochs	20
Loss function	cross-entropy
Dropout	0.1
BERT-fused	
Architecture	Transformer (big)
Learning rate	3×10^{-5}
Batch size	32
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$)
Max epochs	20
Loss function	label smoothed cross-entropy ($\epsilon_{ls} = 0.1$)
Dropout	0.3

Table 5.1: Training details for each model.

[Our models]	Precision	Recall	$F_{0.5}$
Baseline	25.14	14.34	21.85
BERT-encoder	39.78	20.84	33.66
BERT-fused	36.91	18.23	30.63
BERT-encoder (4-ensemble)	47.20	20.54	**37.47**
BERT-fused (4-ensemble)	38.29	17.55	30.97
[Best Single Model]			
Zhao and Wang (2020)	44.36	**22.18**	36.97
[NLPCC 2018]			
Fu et al. (2018)	35.24	18.64	29.91
Ren et al. (2018)	41.73	13.08	29.02
Ren et al. (2018) (4-ensemble)	**47.63**	12.56	30.57

Table 5.2: Experimental results on the NLPCC 2018 Grammatical Error Correction shared task.

6 Analysis of System Outputs and Error Types

In this chapter, we will analyze the system outputs and error types, provide the performance of our models on each error type.

6.1 System Outputs

Table 6.1 shows the sample outputs.

src	**持 别** 是 北京 ， 没有 “ 自然 ” 的 感觉 。
gold	**特别** 是 北京 ， 没有 “ 自然 ” 的 感觉 。
baseline	**持 别** 是 北京 ， 没有 “ 自然 ” 的 感觉 。
BERT-encoder	**特别** 是 北京 ， 没有 “ 自然 ” 的 感觉 。
Translation	**Especially** in Beijing, there is no *natural* feeling.
src	人们 在 一 辈子 **经验** 很多 事情 。
gold	人们 在 一 辈子 **经历** 很多 事情 。
baseline	人们 在 一辈子 **经历** 了 很多 事情 。
BERT-encoder	人们 一辈子 **会 经历** 很多 事情 。
Translation	People **experience** many things in their lifetime.

Table 6.1: Source sentence, gold edit, and output of our models.

In the first example, the spelling error 持别 is accurately corrected to 特别 (which means *especially*) by the proposed model, whereas it is not corrected by the baseline model. Hence, it appears that the proposed model captures context more efficiently by using the pre-trained model through the WWM strategy.

In the second example, the output of the proposed model is more fluent, although the correction made by the proposed model is different from the gold edit. The proposed model not only changed the wrong word 经验 (which usually means the noun *experience*) to 经历 (which usually means the verb *experience*), but also added a new word 会 (*would, could*); this addition makes the sentence more fluent. It appears that the proposed model can implement additional changes to the source sentence because the pre-trained model is trained with a large-scale corpus. However, this type of change may affect the precision because the gold edit in this dataset followed the principle of minimum edit distance (Zhao et al., 2018).

6.2 Error Types

To understand the error distribution of Chinese GEC, we annotate 100 sentences of development data and obtain 130 errors (one sentence may contain more than one error). We refer to the annotation of the HSK learner corpus* and adopt five categories of error: B, CC, CQ, CD, and CJ. B denotes character-level errors, which are mainly spelling and punctuation errors. CC, CQ, and CD are word-level errors, which are word selection, missed word, and redundant word errors, respectively. CJ denotes sentence-level errors which contain several complex errors, such as word order and lack of subject errors. Several examples are presented in Table 6.2. Based on the number of errors, it is evident that word-level errors (CC, CQ, and CD) are the most frequent.

Table 6.3 lists the detection and correction results of the BERT-encoder and BERT-fused models for each error type. The two models perform poorly on sentence-level errors (CJ), which often involve sentence reconstructions, demonstrating that this is a difficult task. For character-level errors (B), the models achieve better performance than for other error types. Compared with the correction performance, the systems indicate moderate detection performance, demonstrating that the systems address error positions appropriately. With respect to the difference in performance of the two systems on each error type, we can conclude that BERT-encoder performs better on character-level errors (B), and

*`http://hsk.blcu.edu.cn/`

Error Type	Number of errors	Examples
B	9	最后 ， 要 关主{**关注**} 一些 关于 天气 预报 的 新闻 。 (Finally, pay attention to some weather forecast news.)
CC	35	有 一 天 晚上 他 下 了 决定{**决心**} 向 富丽堂皇 的 宫殿 里 走 ， 偷偷 的{**地**} 进入 宫内 。 (One night he decided to walk to the magnificent palace, and sneaked in it secretly.)
CQ	30	在 上海 我 总是 住 NONE{**在**} 一家 特定 NONE{**的**} 酒店 。 (I always stay in the same hotel in Shanghai.)
CD	21	我 很 喜欢 念{**NONE**}读 小说 . (I like to read novels.)
CJ	35	...... 但是 同时 也 对 环境 问题{**NONE**} 日益 严重 造成 了{**造成 了 日益 严重 的**} 空气 污染 问题 。 (But on the meanwhile, it also aggravated the environmental problem of air pollution.)

Table 6.2: Examples of each error type. The underlined tokens are detected errors that should be replaced with the tokens in braces.

BERT-fused performs better on other error types.

Type	Detection			Correction		
	Precision	Recall	$F_{0.5}$	Precision	Recall	$F_{0.5}$
BERT-encoder						
B	**80.0**	**55.6**	**73.5**	**80.0**	**55.6**	**73.5**
CC	62.5	31.4	52.2	43.8	20.0	35.4
CQ	65.0	43.3	59.1	45.0	30.0	40.9
CD	58.3	28.6	48.3	50.0	28.6	43.5
CJ	56.5	42.9	53.1	4.3	2.9	3.9
BERT-fused						
B	**80.0**	44.4	**69.0**	**80.0**	44.4	**69.0**
CC	61.9	42.9	56.9	38.1	22.9	33.6
CQ	69.0	**63.3**	67.8	44.8	**46.7**	45.2
CD	71.4	42.9	63.0	57.1	38.1	51.9
CJ	63.2	34.3	54.1	15.8	8.6	13.5

Table 6.3: Detection and correction performance of BERT-encoder and BERT-fused models on each type of error.

7 Conclusion

In this study, we incorporated a pre-trained model into an encoder-decoder model using two methods on Chinese GEC tasks. The experimental results demonstrate the usefulness of the BERT-based pre-trained model in the Chinese GEC task. Additionally, our error type analysis showed that sentence-level errors remain to be addressed.

For future consideration, a majority of the methods proposed in the NLPCC 2018 Grammatical Error Correction shared task are simply based on the methods of English GEC; however, Chinese GEC has its own characteristics. For example, spelling errors mainly arise from the similarity of the glyph and pronunciation, and sentence-level errors often depend on word order. Hence, we plan to study and improve the Chinese GEC system while considering these characteristics, using methods such as incorporating glyph embeddings into the system (Meng et al., 2019) or adopting the neural model whose positional embeddings can capture word order more efficiently (Wang et al., 2020).

Acknowledgements

I want to thank my supervisor, Komachi sensei, he opens the door of research for me and gives many useful comments for my research. It is my honor to be a member of Komachi lab.

I also want to thank my two mentors, Kurosawa san and Katsumata san, who are alumni of Komachi lab. They kindly helped me to do experiments and write the thesis. Without their help, I could not complete my research.

Last, I want to thank my parents. They give me mental and financial supports. Without them, I could not complete my research.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations.* 15 pages.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Workshop on Innovative Use of NLP for Building Educational Applications.* 24 pages.

Shamil Chollampatt and Hwee Tou Ng. 2018. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proceedings of the AAAI Conference on Artificial Intelligence.* 8 pages.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Findings of Conference on Empirical Methods in Natural Language Processing.* 11 pages.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics.* 14 pages.

Kai Fu, Jun Huang, and Yitao Duan. 2018. Youdao's winning solution to the NLPCC-2018 task 2 challenge: A neural machine translation approach to Chinese grammatical error correction. In *The CCF International Conference on Natural Language Processing and Chinese Computing.* 10 pages.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics*. 11 pages.

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 *Proceedings of the Workshop on Innovative Use of NLP for Building Educational Applications*. 6 pages.

Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. 7 pages.

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 *Proceedings of the Workshop on Innovative Use of NLP for Building Educational Applications*. 10 pages.

Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. An empirical study of incorporating pseudo data into grammatical error correction. In *Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*. 7 pages.

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*. 13 pages.

Yuxian Meng, Wei Wu, Fei Wang, Xiaoya Li, Ping Nie, Fan Yin, Muyu Li, Qinghong Han, Xiaofei Sun, and Jiwei Li. 2019. Glyce: Glyph-vectors for Chinese character representations. In *Conference on Neural Information Processing Systems*. 13 pages.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Conference on Neural Information Processing Systems*. 9 pages.

Ryo Nagata and Keisuke Sakaguchi. 2016. Phrase structure annotation and parsing for learner English. In *Proceedings of5 the Annual Meeting of the Association for Computational Linguistics*. 11 pages.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Conference on Natural Language Learning*. 14 pages.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *CoNLL*. 12 pages.

Hongkai Ren, Liner Yang, and Endong Xun. 2018. A sequence to sequence learning for Chinese grammatical error correction. In *The CCF International Conference on Natural Language Processing and Chinese Computing*. 10 pages.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Conference on Neural Information Processing Systems*. 9 pages.

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 *Conference on Neural Information Processing Systems*. 15 pages.

Benyou Wang, Donghao Zhao, Christina Lioma, Qiuchi Li, Peng Zhang, and Jakob Grue Simonsen. 2020. Encoding word order in complex embeddings. In *International Conference on Learning Representations*. 15 pages.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. 11 pages.

Yuanyuan Zhao, Nan Jiang, Weiwei Sun, and Xiaojun Wan. 2018. Overview of the NLPCC 2018 shared task: Grammatical error correction. In *The CCF International Conference on Natural Language Processing and Chinese Computing.* 7 pages.

Zewei Zhao and Houfeng Wang. 2020. MaskGEC: Improving neural grammatical error correction via dynamic masking. In *Proceedings of the AAAI Conference on Artificial Intelligence.* 8 pages.

Junpei Zhou, Chen Li, Hengyou Liu, Zuyi Bao, Guangwei Xu, and Linlin Li. 2018. Chinese grammatical error correction using statistical and neural models. In *The CCF International Conference on Natural Language Processing and Chinese Computing.* 12 pages.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. 2020. Incorporating BERT into neural machine translation. In *International Conference on Learning Representations.* 18 pages.

Publication List

[1] Hongfei Wang, Michiki Kurosawa, Satoru Katsumata and Mamoru Komachi. 2020. 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.*

[2] Hongfei Wang and Mamoru Komachi. 2020. TMU-NLP System Using BERT-based Pre-trained Model to the NLP-TEA CGED Shared Task 2020. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications.* (no peer-review)

[3] Hongfei Wang, Michiki Kurosawa, Satoru Katsumata and Mamoru Komachi. 2020. Chinese Grammatical Error Correction Using BERT-based Pre-trained Model. In *The 26th Annual Proceedings of Association for Natural Language Processing (Japan).* (no peer-review)