BERT Enhanced Neural Machine Translation and Sequence Tagging Model for Chinese Grammatical Error Diagnosis

Deng Liang [1], **Chen Zheng**[1], **Lei Guo**[1], **Xin Cui**[1], **Xiuzhang Xiong**[1,2],
Hengqiao Rong[1,3], **and Jinpeng Dong**[1]

[1]Beijing Waiyan Online Digital Technology Co.,Ltd.
[2]Minzu University of China
[3]Catholic University of Leuven
{liangdeng, zhengchen, guolei, cuixin, dongjp}@unipus.cn
hengqiao.rong@student.kuleuven.be 921883243@163.com

Abstract

This paper presents the UNIPUS-Flaubert team's hybrid system for the NLPTEA 2020 shared task of Chinese Grammatical Error Diagnosis (CGED). As a challenging NLP task, CGED has attracted increasing attention recently and has not yet fully benefited from the powerful pre-trained BERT-based models. We explore this by experimenting with three types of models. The position-tagging models and correction-tagging models are sequence tagging models fine-tuned on pre-trained BERT-based models, where the former focuses on detecting, positioning and classifying errors, and the latter aims at correcting errors. We also utilize rich representations from BERT-based models by transferring the BERT-fused models to the correction task, and further improve the performance by pre-training on a vast size of unsupervised synthetic data. To the best of our knowledge, we are the first to introduce and transfer the BERT-fused NMT model and sequence tagging model into the Chinese Grammatical Error Correction field. Our work achieved the second-highest F1 score at the detecting errors, the best F1 score at correction top1 subtask and the second-highest F1 score at correction top3 subtask.

1 Introduction

Recently, the pre-trained language models such as BERT (Devlin et al., 2019) obtain state-of-the-art results on a wide range of natural language processing (NLP) tasks, such as text classification, reading comprehension, machine translation (Zhu et al., 2020), etc. The English Grammatical Error Correction (GEC) task also benefits from the pre-trained language models. For example, in the work of Kaneko et al. (2020), they not only follow Zhu et al. (2020) to incorporate BERT into an Encoder-Decoder model for GEC, but also maximize the benefit by additionally training BERT on GEC corpora (BERT-fuse mask) or fine-tuning BERT as a GED model (BERT-fuse GED). Another route to improve the performance of GEC is using BERT as an encoder and incorporating it into a sequence tagging model (Malmi et al., 2019; Awasthi et al., 2019; Omelianchuk et al., 2020).

In the Chinese NLP community, a variety of pre-trained Chinese language models have been proposed and publicly available (Sun et al., 2019; Cui et al., 2019, 2020). Those models are proved to have a significant improvement in a variety of down-stream tasks, including reading comprehension, natural language inference, sentiment classification, etc.

In this paper, we apply the state-of-the-art English GEC models to the CGED task. Our CGED system consists of three types of models. We propose the position-tagging model, which is a sequence tagging model with a BERT encoder, to concentrate on the error localization task. The output label consists of 8 types of tags and indicates the start and end of each error for the input sentence, but it will not tell us how to correct it in the case of S (word selection) and M (missing word) errors. The correction-tagging model (Malmi et al., 2019; Awasthi et al., 2019; Omelianchuk et al., 2020) concentrates on the error correction task, and the output label contains 8772 types of tags. The tags reveal the editing operations for each Chinese character, e.g. KEEP, DELETE, APPEND, and REPLACE. The APPEND tags (3788 in total) and REPLACE tags (4982 in total) cover most Chinese characters.

The BERT-fused model (Zhu et al., 2020) is proposed for Neural Machine Translation (NMT) task and adaptively controls the interaction between representations from BERT and each layer of the Transformer (Vaswani et al., 2017) by using the attention mechanism. (Kaneko et al., 2020) transfers the BERT-fused model to the English GEC task and further advances it. Due to time limitations,

Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications, pages 57–66
Suzhou, China, December 4, 2020.

we only follow the training settings in (Zhu et al., 2020). Besides, we perform unsupervised data augmentation by introducing synthetic errors on a large amount of error-free corpora, then pair synthetic and original sentences to pre-train Transformers (Grundkiewicz et al., 2019).

This paper is organized as follows: Section 2 summarizes the recent developments in the field of CGED. Section 3 introduces the dataset we used to train the models, including human-annotated data and synthetic data. Section 4 is the overview of each component of our system, including BERT-fused NMT, position-tagging model, correction-tagging model, and error annotation tool. Section 5 describes our training and ensemble process. Section 6 discusses the result of our models and Section 7 concludes the paper.

2 Related Work

Zhao et al. (2015) used a statistical machine translation method to the CGED task and examined corpus-augmentation and explored alternative translation models including syntax-based and hierarchical phrase-based models. Zheng et al. (2016), Yang et al. (2017) and Liao et al. (2017) treat the CGED task as a sequence tagging problem to detect the grammatical errors. Li and Qi (2018) applied a policy gradient LSTM model to the CGED task. Fu et al. (2018b) built a CGED system based on a BiLSTM-CRF model and combined with rule-based templates to bring in grammatical knowledge. Hu et al. (2018) employed a sequence-to-sequence model and used pseudo data to pre-training the model. Li et al. (2018) designed a system for CGED which is composed of a BiLSTM-CRF model, an NMT model, and a statistical machine translation model to detect and correct the grammatical errors. A similar system (Zhou et al., 2018) achieved a competitive result in NLPCC 2018 shared task. Fu et al. (2018a) also treated the CGED task as a translation problem and used character-based and sub-word based NMTs to correct the grammatical errors. Li et al. (2019) and Ren et al. (2018) introduced the convolutional sequence-to-sequence model into the CGED task.

3 Datasets

Training data The datasets of the NLPTEA 2014∼2018 & 2020 shared task of CGED are corpora composed of parallel sentences written by Chinese as a Foreign Language (CFL) learners and their corrections. The source sentences are selected from the essay section of the computer-based TOCFL (Test of Chinese as a Foreign Language) and written-based HSK (Pinyin of Hanyu Shuiping Kaoshi, Test of Chinese Level). Before 2016, there are only TOCFL data written in traditional Chinese. In the dataset of 2016, we have both TOCFL and HSK data. We use the `opencc`[1] package to convert the traditional Chinese to simplified Chinese for the TOCFL corpus. Since 2017, only HSK data are provided that are all written in simplified Chinese.

The grammatical errors were manually annotated by native Chinese speakers. There are four kinds of errors: R (redundant word), M (missing word), S (word selection error), and W (word ordering error). Each error type has a different proportion in the corpus and each sentence may contain several errors. For example, in the CGED 2020 training set, W/S/R/M accounted for 7%, 42%, 23%, 28% of the total errors respectively. There are 2909 manually annotated errors in 1641 sentences, and only 2 sentences are error-free.

We also collect several external datasets from NLPCC 2018 GEC [2] and other resources [3]. The NLPCC 2018 GEC data contains more than 700,000 sentences and each sentence may be correct or have one or more candidate corrections.

Synthetic data We train BERT-fused NMT models in pre-training mode and no pre-training mode. For pre-training mode, the model is pre-trained on a large amount of synthetic data (Grundkiewicz et al., 2019). The other models did not use the synthetic data.

We first split each error-free sentence into words by a Chinese word segmentation tool [4], and then randomly select several words for each sentence. The number of selected word is the product of a probability which is sampled from the normal distribution and the number of words in the sentence. For each selected word, one of the four operations including substitution, deletion, insertion, and transposition is performed with a probability of 0.5, 0.2, 0.2, 0.1, which simulates the proportions of S, M, R, W errors in the CGED data.

For substitution, the selected word is replaced by a word that has a similar meaning, pronunciation,

[1] https://github.com/BYVoid/OpenCC

[2] http://tcci.ccf.org.cn/conference/2018/taskdata.php

[3] https://github.com/shibing624/pycorrector

[4] https://github.com/fxsjy/jieba

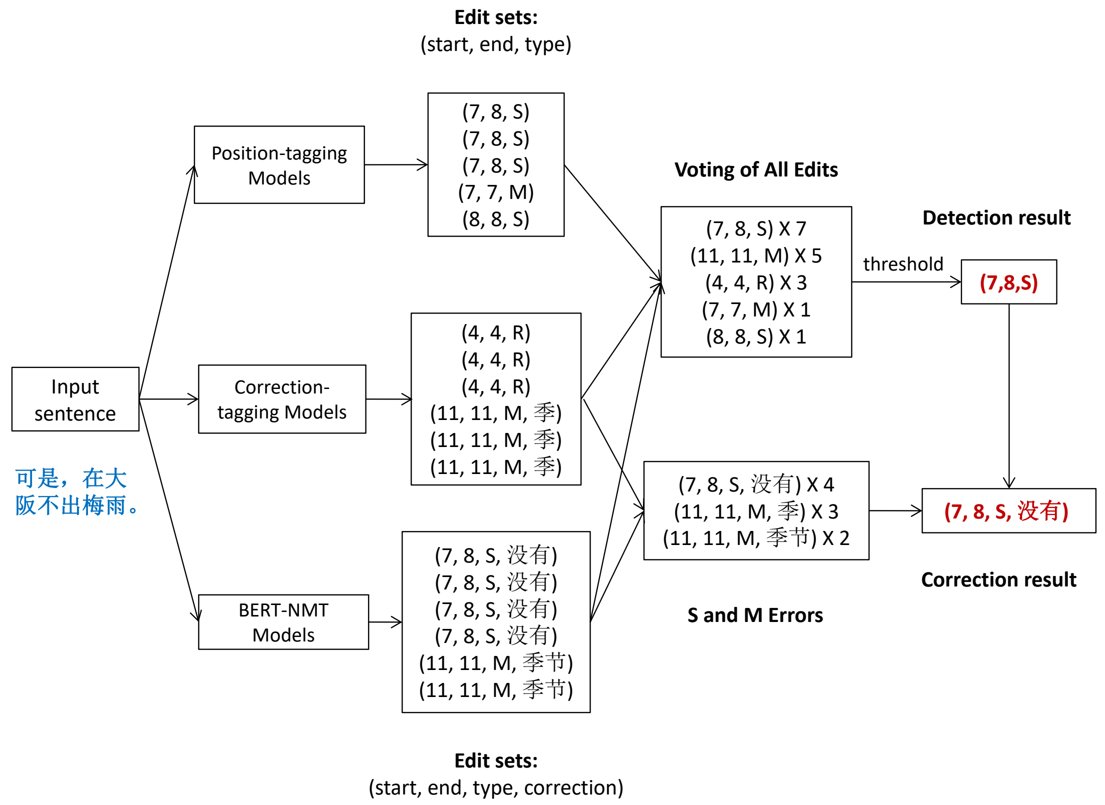

Figure 1: A demonstration of our hybrid system using a real sentence from the CGED 2020 test set. Each edit format (start, end, type, correction) stands for an error and its start position, end position, type and correction. Here, all groups of models have an equal weight 1 and the threshold is set to 7.

or shape. To simulate the confusion from similar meaning, we randomly choose a replacement from the following sources: (1) synonyms of the selected word [5] with a word similarity greater than 0.75; (2) a Chinese dictionary that we can search the word contain at least one character identical to the selected word; (3) a confusion dictionary consists of Japanese and Chinese word pairs that might be misused by Japanese learners. To mimic the confusion from similar pronunciation, we replace the selected word with a word that has the same pinyin. When introducing confusion from similar shapes, we define the similarity between two characters by their four-corner code [6].

For deletion, we simply remove the selected word. For insertion, we add on a word randomly taken from a set after the selected word. The set consists of stop words [7] and redundant words from R errors in the past CGED dataset. For transposition, we swap the selected word with the next word or with a random word in the sentence. We skip the named entities for substitution and deletion operations.

After introducing the word-level error to each error-free sentence, we introduce character-level errors by similar methods.

The corpora we used to generate synthetic data are the wiki2019zh (9.64 million sentences), the news2016zh (51.4 million sentences), the webtext2019zh (1.06 million sentences)[8] and the SogouCA (0.94 million sentences) [9].

4 System Overview

Our system consists of a sequence labeling model concentrated on the error detection subtask, and two types of error correction models aimed at generating candidate corrections.

[5] https://github.com/chatopera/Synonyms

[6] http://code.web.idv.hk/misc/four.php?i=3

[7] https://github.com/goto456/stopwords

[8] https://github.com/brightmart/nlp_chinese_corpus

[9] http://www.sogou.com/labs/resource/ca.php

Data set	# sent pairs	Source
PT	61.0 M	wiki2019zh, news2016zh
MA	1.17 M	CGED 2016∼2018 train × 5 , HSK, NLPCC
AMA	5.35 M	CGED (2014 train + test, 2015 train +test)×3, (2016 train + 2017 train)×10, 2018 train×20, HSK × 2, NLPCC × 3, webtext2019zh, SogouCA

Table 1: Summary of the three training sets we constructed to train the BERT-NMT models at different stages. The number after the multiplication sign stands for how many times the data was oversampled.

4.1 Position-tagging Model

The position-tagging model is a sequence tagging model aimed to locate grammatical errors. We use RoBERTa (Liu et al., 2019) [10] as the model's encoder then fine-tune it during training. The output tags are generated by applying a softmax layer over the encoder's logits.

Given a sequence of Chinese characters as input, the model predicts the label of each character. The output label consists of 8 types of tag, including O (correct), B-S (begin of S), I-S (middle of S), B-W (begin of W), I-W (middle of W), B-M (begin of M), B-R (begin of R), and I-R (middle of R). We extract the location and type of each error directly from the output labels. For S and M errors, the model can not give any candidate corrections.

4.2 BERT-fused NMT

The BERT-fused NMT model proposed in (Zhu et al., 2020) aims at the NMT task, we transfer the original work to the correction subtask. The BERT-fused NMT model is made up of two modules: the NMT module and the BERT module. Both modules take erroneous sentences as input. We start with training a Transformer from scratch until it converges. Then, we use the encoder and decoder of this Transformer to initialize the encoder and decoder of the NMT module. The BERT module is identical to a ready-made pre-trained BERT model.

The way to fuse the NMT module and the BERT module is to feed the representations from the BERT module (i.e. the output of the last layer of the BERT module) to each layer of the NMT module. Taking the NMT encoder as an example, the BERT-encoder attention is introduced into each NMT encoder layer and processes the representations from the BERT module. The original self-attention of each NMT encoder layer still processes the representations from the previous NMT encoder layer. The output of the BERT-encoder attention and the original self-attention are further processed by the encoder layer's original feedforward network. The NMT decoder works similarly by introducing BERT-decoder attention to each NMT decoder layer.

The parameters of the BERT-encoder attention and BERT-decoder attention are randomly initialized. During the training of the BERT-fused NMT model, the parameters of the BERT module are fixed.

4.3 Correction-tagging Model

The correction-tagging model is a sequence tagging model[11] specific to the GEC task. The output labels consist of 8772 tags, which form a large edit space. We obtain corrections by iteratively feeding a sentence to the model, getting the edit operations of each character, then editing the sentence.

To prepare the training data, we first convert the target sentence into a sequence of tags where each tag represents an edit operation on each source token. Take the following sentence pair as an example:

Source: 突 然   风 起 来 刮 了 。
Target: 突 然 刮 起 风   来   了 。

We use the minimum edit distance algorithm to align the source tokens with the target tokens. For each mapping in alignment, we collect the edit steps from the source token to the target subsequence:

突$^{\text{KEEP}}$ 然$^{\text{KEEP \& APPEND\_刮 \& APPEND\_起}}$
风$^{\text{KEEP}}$ 起$^{\text{DELETE}}$ 来$^{\text{KEEP}}$ 刮$^{\text{DELETE}}$
了$^{\text{KEEP}}$ 。$^{\text{KEEP}}$

Lastly, we leave only one edit for each source token, because in the training stage, each token can only have one label. In the case of the above example,

[10] RoBERTa-wwm-ext-large, from https://github.com/ymcui/Chinese-BERT-wwm

[11] https://github.com/grammarly/gector

突$^{\text{KEEP}}$ 然$^{\text{APPEND\_刮}}$ 风$^{\text{KEEP}}$ 起$^{\text{DELETE}}$
来$^{\text{KEEP}}$ 刮$^{\text{DELETE}}$ 了$^{\text{KEEP}}$ 。$^{\text{KEEP}}$

The correction-tagging model is a pre-trained BERT-like Transformer encoder stacked with two linear layers and softmax layers on the top.

In the inference stage, we tag and edit the sentence iteratively to obtain a fully corrected sentence. In each iteration, we apply the edits according to the output labels on the input sentence and send the edited sentence to the next iteration.

4.4 Error Classification

For the BERT-fused NMT and correction-tagging model, the final output is a corrected sentence. To match with the official submission format, we align the target sentence with the source sentence to locate the start and end of the error and classify error types.

In the field of GED, there is a widely used error annotation tool — errant (Bryant et al., 2017), which automatically annotates error type information of parallel English sentences. However, there is no such tool in the CGED task. We developed a simple rule-based annotation tool to locate the error and classify the error type. Our tool first segment the source and target sentence into words using Jieba [12], then align the source and target words based on the minimum edit distance algorithm. In each mapping, if the blocks of source and target words are not the same, our tool judges this mapping as a grammatical error and determines the position and type of this error.

However, even if we have the golden corrected sentence, there exists some ambiguity when localizing and classifying the error. For example, in the CGED 2020 training set, given the following sentence pairs:

Source: 首先通过对话来知道子女的爱好、价值观，然后一起相受拥着共同的爱好。
Target: 首先通过对话来知道子女的爱好、价值观，然后一起拥有共同的爱好。

The official result is an S error starts from the 24th character and ends at the 27th character (”相受拥着”) with a correction ”拥有”. But there may be many possible solutions that depend on the word segmentation. For example, if we split ”相受拥着” into ”相受” and ”拥着”，the result becomes an R error starts from the 24th character and ends at the 25th character and an S error starts from the 26th character and end the 27th character (”拥着”) with a substitution ”拥有”. So, it is hard to locate and classify errors unambiguously due to different word segmentation rules.

We tested our annotation tool on the CGED 2020 training data set, which are shown in Table 2. Our error annotation tool loses some precision and recall at the detection, identification, and position subtasks when annotating the error information from parallel sentences.

5 Experiments

5.1 Position-tagging Model

We trained the position-tagging models with two different combinations of CGED data and used the CGED 2016 test set as the development set. For each data combination, we tried serval models with different parameter initialization and training settings. When using CGED 2016 (HSK)∼2018 & 2020 training set and 2017 test set as the training set, we get the best performance of the F1 score on detection and identification subtask on the CGED 2018 test set. When adding the TOCFL data from 2014 to 2016 to the training set, we get the best performance of the F1 score on the position subtask(see Table 3). Four position-tagging models (two models from each data combination) are used in ensemble modeling.

5.2 BERT-fused NMT

We prepared several datasets to train the BERT-fused NMT models. The first dataset is named Pre-Training data (PT data) consisting of synthetic sentences from the wiki2019zh corpus and the news2016zh corpus. The second dataset is the Manually Annotated data (MA data) which is composed of the CGED 2016∼2018 training set, HSK, and NLPCC 2018 GEC data. We filtered out the error-free sentences in HSK and NLPCC 2018 GEC dataset and oversampled the CGED data. The last dataset is the Augmented Manually-Annotated data (AMA data) consists of oversampled MA data and synthetic sentences from the text2019zh corpus and the SogouCA corpus. See details at Table 1.

We trained BERT-fused NMT models in pre-training mode and non-pre-training mode. For non-pre-training mode, we trained the BERT-fused NMT in the following steps: (1) train a baseline

[12] https://github.com/fxsjy/jieba

	M	R	S	W	Total
Detection	0.902	0.902	0.924	0.785	1
Identification	0.909	0.914	0.930	0.801	0.899
Position	0.825	0.782	0.652	0.390	0.712

Table 2: The test results of the error annotation tool. Given an original and corrected sentence pair from CGED 2020 training set, the tool extracts the position and type of each error. We compare the output of the tool with the standard result and get the F1 scores of each error type.

Model	Detection	Identification	Position
Data comb. 1	**0.780**	**0.644**	0.399
Data comb. 2	0.776	0.641	**0.428**

Table 3: The best results of the position-tagging model on the CGED 2018 test set. The data comb. 1 is the model trained on CGED 2016 (HSK)∼2018 & 2020 training set and 2017 test set, the data comb. 2 is the model trained on more data which added TOCFL 2014∼2016 data. The former gets the best performance of the F1 score on detection and identification subtask and the latter gets the best performance on the position subtask.

	2018 test set			2020 test set		
Model name	P	R	F1	P	R	F1
BERT ∗	0.213	0.193	0.203	0.185	0.134	0.155
RoBERTa ∗	0.245	0.213	0.228	0.206	0.134	0.162
ELECTRA	0.211	0.184	0.197	0.180	0.118	0.143
XLNet	0.215	0.151	0.178	0.184	0.106	0.134
Ensemble (RoBERTa + BERT) ∗	0.237	0.227	**0.232**	0.203	0.154	**0.176**
Baseline Transformer (MA)	0.263	0.0967	0.141	0.208	0.0723	0.107
→ BERT-fused (MA) ∗	0.263	0.216	**0.256**	0.223	0.118	0.154
→ Fine-tuned on AMA ∗	0.281	0.217	0.245	0.236	0.145	**0.180**
Pre-trained Transformer (PT)	0.0953	0.0324	0.0484	0.147	0.05	0.0747
→ Fine-tuned on AMA ∗	0.219	0.218	0.219	0.184	0.135	0.155
→ BERT-fused (MA)∗	0.308	0.190	**0.235**	0.224	0.124	0.159
→ BERT-fused (AMA)	0.257	0.197	0.223	0.219	0.144	**0.174**
Ensemble	-	-	-	0.222	0.192	0.206

Table 4: The results of our correction models and the ensemble on correction top1 subtask on the CGED 2018/2020 test set. The first group shows the results of the correction-tagging model with various encoders. The second / third group shows the results of the BERT-fused NMT models in non-pre-trained / pre-trained mode. The asterisk after the model name indicates that the model participates in the final ensemble. The model BERT-fused (AMA) in the third group is not used in the ensemble stage due to the time limit of the competition, and the training was completed after the deadline. The original scores of the ensemble on the CGED 2020 test set are P = 0.2848, R = 0.1415, F1 = 0.1891. We recalculated scores after an update of the error annotation tool and got a slight improvement on the final performance.

Transformer from scratch on MA data; (2) train a BERT-fused model on MA data using the baseline Transformer trained in the previous step; (3) fine-tune the previous step's BERT-fused model on AMA data. For pre-training mode, we trained the model in the following steps: (1) pre-train a Transformer from scratch on PT data; (2) fine-tune the previous step's pre-trained Transformer on AMA data; (3) train a BERT-fused model using the fine-tuned Transformer from the previous step on MA data and AMA data respectively. In all the training steps above, we combined the CGED 2018 test set and the CGED 2020 training set as the development set.

We use the `fairseq` (Ott et al., 2019) to train Transformers and the `bert-nmt` to train BERT-fused models [13]. We use *Transformer Base* architecture to train all the Transformer models and reset the learning rate scheduler and optimizer parameters when training the fine-tuned Transformer and BERT-fused model. The parameters of the fine-tuned Transformer are used to initialize the encoder and decoder of the BERT-fused model. BERT-encoder attention and BERT-decoder attention are randomly initialized. We adopt the label smoothed cross-entropy as a loss function. The overall performance of each NMT model are listed in Table 4.

5.3 Correction-tagging Model

The training of the correction-tagging model is decomposed into two stages, which are inspired by Omelianchuk et al. (2020). The first stage uses all training sets from CGED 2014∼2018 and NLPCC 2018 as the training set and the CGED 2020 training set as the development set. For NLPCC 2018 training set, we discard the sentence that is correct or has more than one correction. The second stage fine-tunes on 80% CGED 2020 training set and takes the other 20% as the development set.

The difference between our training process and Omelianchuk et al. (2020) is that we do not use synthetic data to pre-train the model. It will be investigated in future work that if a pre-training step on a large synthetic data set can improve the performance of the current model.

We fine-tune four models using the BERT (Devlin et al., 2019), RoBERTa [14], ELECTRA (Clark et al., 2020) [15], and XLNet (Yang et al., 2019) [16] encoders. The learning rate for each model on the first stage is 2e-5, 2e-5, 4e-5, and 4e-5 respectively, and all 1e-5 on the second stage. In the first stage, we freeze the encoder's weights for the first epoch and set the learning rate to 1e-3.

We adjust several hyperparameters after fine-tuning the models. The first is a threshold of the KEEP tag probability. If the KEEP tag probability is greater than the threshold, we will not change the source token. The other hyperparameters are the threshold of sentence-level minimum error probability and the number of iterations. These hyperparameters are tuned on the CGED 2018 test set to trade-off precision and recall.

A simple ensemble of RoBERTa and BERT got an additional boost of the F1 score. We use BERT, RoBERTa, and their ensemble during the ensemble modeling.

Both the BERT-fused NMT models and correction-tagging models are character-based instead of word-based for two reasons. First, the Chinese word segmentation tools are usually trained on grammatical sentences and will generate unexpected word segmentation results when applied to erroneous sentences. Second, word-based models use a larger vocabulary dictionary and more data is needed to obtain well-trained models, which conflicts with the fact that CGED is obviously a low-resource task.

5.4 Ensemble Modeling

We adopt a weighted voting strategy inspired by Li et al. (2018). The output of position-tagging models provides the position and type of each error but lack corrections for S and M errors. The output of BERT-fused NMT models and correction-tagging models are corrected sentences and are converted into the official submission format using our annotation tool in Section 4.4.

First, we omit the corrections for S and M errors temporarily and vote to determine the result of the position and type of all the errors. We accept an error proposal only if it gets the votes more than a threshold. A sentence is treated as correct if all its error proposals are not accepted. Then, we fill

[13] https://github.com/bert-nmt/bert-nmt, the pre-trained BERT from https://huggingface.co/bert-base-chinese

[14] BERT-wwm-ext and RoBERTa-wwm-ext-large, from https://github.com/ymcui/Chinese-BERT-wwm

[15] ELECTRA-large, from https://github.com/ymcui/Chinese-ELECTRA

[16] XLNet-mid, from https://github.com/ymcui/Chinese-XLNet

Submition	Detection	Identification	Position	Correction top1	Correction top3
Run 1	0.8479	0.5893	**0.3140**	**0.1891**	**0.1876**
Run 2	0.8311	0.5791	0.3057	0.1793	0.1613
Run 3	**0.8966**	**0.6463**	0.2929	0.1785	0.1564

Table 5: The overall F1 scores of our three submissions.

the corrections. For each accepted S and M error, we rank the candidate corrections from the BERT-fused NMT models and correction-tagging models according to votes. We take the first three candidates as the final corrections. A demonstration of our ensemble strategy is showed in Figure 1.

Each group of models has different weights during voting. All the thresholds and weights are tuned on the CGED 2018 test set using grid search, aiming at obtaining the best F1 score in the correction top1 subtask. The official evaluation of our three submissions are described in Table 5. Run 1 got 1st place in the correction top1 subtask and 2nd place in the correction top3 subtask. The difference between Run 1 and Run 2 is that the hyperparameter of n-best in BERT-fused NMT models is set to 1 and 8 respectively. For Run 2 (n-best is 8), each BERT-fused NMT model generates 8 candidate sentences and all take part in the voting. Run 3 tried a different ensemble modeling which mainly focused on improving recall and got the 2nd place at the detection subtask.

6 Discussion

For the BERT-fused NMT models, the BERT-fused stage improves the F1 scores for both non-pre-training and pre-training mode (See Table 4). In the non-pre-training mode, fine-tuning on AMA data further improves the performance on the CGED 2020 test set. By comparing the Baseline Transformer at the non-pre-training mode with the Fine-tuned Transformer at the pre-training mode, we find a substantial improvement of the performance on both the CGED 2018 and 2020 test sets. This proves that the CGED task can benefit from pre-training on synthetic data. However, the best results of the non-pre-training mode surpass the pre-training mode unexpectedly after the BERT-fused stage. We will investigate the reason in the future work.

(Kaneko et al., 2020) (Zhao et al., 2019) demonstrated that the GED task can help improve the performance of the GEC task. Due to time limitations, we did not try to combine the detection and correction processes in our system, which can be further improved in the future work.

In the ensemble modeling, we found that FPR (False Positive Rate) decreased as the threshold in the voting stage increased. Our submissions did not rank high in the FPR subtask, since we focused on the detection and correction rather than the FPR subtask.

Compared to the methods proposed in the NLPTEA 2018 shared task of CGED, our system greatly improves the F1 score on correction top1 and correction top3 subtask on the CGED 2018 test set. This advance mainly comes from: (1) we not only fully exploit the Transformer model for the correction subtask, but also comprehensively incorporate the power of pre-trained BERT-based models into every subtask of the CGED task; (2) the low-resource problem in the GEC task restricts the performance of NMT models (Junczys-Dowmunt et al., 2018), and we address this by utilizing the power of pre-trained BERT models and synthesizing extensive artificial data.

7 Conclusion

In this work, we present our solutions to the NLPTEA 2020 shared task of CGED. Three kinds of models are used in our system: position-tagging models, BERT-fused NMT models and correction-tagging models. Our hybrid system achieved the second-highest F1 score in the detection subtask, the highest F1 score in the correction top1 subtask and the second-highest F1 score in the correction top3 subtask, which shows that the CGED task can benefit from the recent advances of pre-trained language models.

References

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. In *2019 Conference on Empirical Methods in Natural Language Processing*, pages 4259–4269.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 793–805.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *ICLR 2020 : Eighth International Conference on Learning Representations*.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for chinese natural language processing. *arXiv preprint arXiv:2004.13922*.

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 preprint arXiv:1906.08101*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT 2019: Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

Kai Fu, Jin Huang, and Yitao Duan. 2018a. Youdao's 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*, pages 341–350.

Ruiji Fu, Zhengqi Pei, Jiefu Gong, Wei Song, Dechuan Teng, Wanxiang Che, Shijin Wang, Guoping Hu, and Ting Liu. 2018b. Chinese grammatical error diagnosis using statistical and prior knowledge driven features with probabilistic ensemble enhancement. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 52–59.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263.

Qinan Hu, Yongwei Zhang, Fang Liu, and Yueguo Gu. 2018. Ling@cass solution to the nlp-tea cged shared task 2018. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 70–76.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 595–606.

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 *ACL 2020: 58th annual meeting of the Association for Computational Linguistics*, pages 4248–4254.

Changliang Li and Ji Qi. 2018. Chinese grammatical error diagnosis based on policy gradient lstm model. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 77–82.

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

Si Li, Jianbo Zhao, Guirong Shi, Yuanpeng Tan, Huifang Xu, Guang Chen, Haibo Lan, and Zhiqing Lin. 2019. Chinese grammatical error correction based on convolutional sequence to sequence model. *IEEE Access*, 7:72905–72913.

Quanlei Liao, Jin Wang, Jinnan Yang, and Xuejie Zhang. 2017. Ynu-hpcc at ijcnlp-2017 task 1: Chinese grammatical error diagnosis using a bi-directional lstm-crf model. In *Proceedings of the IJCNLP 2017, Shared Tasks*, pages 73–77.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. In *2019 Conference on Empirical Methods in Natural Language Processing*, pages 5053–5064.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem N. Chernodub, and Oleksandr Skurzhanskyi. 2020. Gector – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *NAACL-HLT 2019: Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 48–53.

Hongkai Ren, Liner Yang, and Endong Xun. 2018. A sequence to sequence learning for chinese grammatical error correction. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 401–410.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

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 *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5998–6008.

Yi Yang, Pengjun Xie, Jun Tao, Guangwei Xu, Linlin Li, and Si Luo. 2017. 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*, pages 41–46.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS 2019 : Thirty-third Conference on Neural Information Processing Systems*, pages 5753–5763.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. *arXiv preprint arXiv:1903.00138*.

Yinchen Zhao, Mamoru Komachi, and Hiroshi Ishikawa. 2015. Improving chinese grammatical error correction with corpus augmentation and hierarchical phrase-based statistical machine translation. In *Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications*, pages 111–116.

Bo Zheng, Wanxiang Che, Jiang Guo, and Ting Liu. 2016. Chinese grammatical error diagnosis with long short-term memory networks. In *NLP-TEA@COLING*, pages 49–56.

Junpei Zhou, Chen Li, Hengyou Liu, Zuyi Bao, Guangwei Xu, and Linlin Li. 2018. Chinese grammatical error correction using statistical and neural models. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 117–128.

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 *ICLR 2020 : Eighth International Conference on Learning Representations*.