A Hybrid System for NLPTEA-2020 CGED Shared Task

Meiyuan Fang[*], Kai Fu[*], Jiping Wang, Yang Liu, Jin Huang, Yitao Duan
NetEase Youdao Information Technology (Beijing) Co., LTD
{fangmeiyuan, fukai, wangjp, liuyang, huangjin, duan}
@youdao.com

Abstract

This paper introduces our system at NLPTEA2020 shared task for CGED, which is able to detect, locate, identify and correct grammatical errors in Chinese writings. The system consists of three components: GED, GEC, and post processing. GED is an ensemble of multiple BERT-based sequence labeling models for handling GED tasks. GEC performs error correction. We exploit a collection of heterogenous models, including Seq2Seq, GECToR and a candidate generation module to obtain correction candidates. Finally in the post processing stage, results from GED and GEC are fused to form the final outputs. We tune our models to lean towards optimizing precision, which we believe is more crucial in practice. As a result, among the six tracks in the shared task, our system performs well in the correction tracks: measured in F1 score, we rank first, with the highest precision, in the TOP3 correction track and third in the TOP1 correction track, also with the highest precision. Ours are among the top 4 to 6 in other tracks, except for FPR where we rank 12. And our system achieves the highest precisions among the top 10 submissions at IDENTIFICATION and POSITION tracks.

1 Introduction

With the rapid growth of online education platforms and the advance of natural language processing (NLP) techniques, recent years have seen an increased interest in automatic Grammatical Error Diagnosis (GED) and Grammatical Error Correction (GEC). Shared tasks such as CoNLL-2013, CoNLL-2014 and BEA-2019 (Ng et al., 2013, 2014; Bryant et al., 2019) were held to correct grammatical errors in essays written by learners of English as a Foreign Language (EFL). State-of-the-art GEC systems for EFL learners have achieved impressive $F_{0.5}$ scores of 66.5 on CoNLL-2014 (test) and 73.7 on BEA-2019 (test).

Despite the great success of English GEC systems, Chinese Grammatical Error Detection (CGED) and Correction (CGEC) applications yet remain relatively unexplored. Chinese, on the other hand, is quite different from western languages such as English: There are more than 3,000 commonly used Chinese characters, while English has only 26 in total; Chinese uses tones to indicate various meanings, while English uses them to express emotions; Chinese emphasizes the meaning of expressions, usually resulting in short sentences without complex structure often seen in English. Due to the large number of complex characters and flexible sentence structures, Chinese is considered one of the most difficult languages in the world to learn.

Under this circumstance, the workshop on Natural Language Processing Techniques for Educational Applications (NLP-TEA) has been organizing shared tasks for CGED (Yu et al., 2014; Lee et al., 2015, 2016; Rao et al., 2017, 2018) to help learners of Chinese as a Foreign Language (CFL) since 2014. The shared tasks provide common test conditions for researchers from both industry and academia. We believe they are very beneficial to advancing CGED technology.

This paper introduces our work on this year's CGED shared task. The task requires both error detection and correction, and we use a hybrid system to handle both. It uses as building blocks models designed for various NLP tasks, including BERT-based sequence labeling models, Seq2Seq, and GECToR. We tune our models to lean towards optimizing precision, which we believe is more crucial in practice. The performance is further improved by using synthetic data generated for individual

[*]Equal contribution.

Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications, pages 67–77
Suzhou, China, December 4, 2020.

tasks. Our system performs well in the correction tracks: measured in F1 score, we rank first, with the highest precision, in the TOP3 correction track and third in the TOP1 correction track, also with the highest precision. Ours are among the top 4 to 6 in other tracks, except for FPR where we rank 12. And our system achieves the highest precisions among the top 10 submissions at IDENTIFICATION and POSITION tracks.

The rest of this paper is organized as follows: A brief description of the CGED shared task is given in Section 2, followed by an overview of prior work in Section 3. Section 4 introduces our system in detail, and Section 5 demonstrates the experimental results. Finally, Section 6 concludes this paper.

2 Task Description

Generally, the CGED shared task classifies grammatical errors found in Chinese writings into four different classes, *i.e.*, redundant words (R), missing words (M), word selection errors (S), word ordering errors (W). Table 1 gives some examples of the errors, which are sampled from CGED 2020 training data. It should be noted that various error types may occur more than once in one sentence.

System performance is measured at the following levels:

- Detection-level. At this level, developed systems are required to distinguish whether a sentence contains the above-mentioned errors.
- Identification-level. At this level, developed systems need to identify the exact error types embedded in input sentences.
- Position-level. At this level, in addition to the error types, developed systems are asked to provide the positional information, indicating where the specific error occurs. For example, triples (5, 5, R) and (2, 3, W) are expected for S and W errors shown in Table 1.
- Correction-level. At this level, developed systems are required to provide up to 3 potential correction candidates for S or M errors.

3 Related Work

Grammatical Error Diagnosis. GED tasks are usually treated as a kind of sequential labeling problem. The common solution to this problem is utilizing the Long Short-Term Memory (LSTM) - Conditional Random Fields (CRF) model (Yang et al., 2017; Liao et al., 2017; Fu et al., 2018b; Zhang et al., 2018; Li et al., 2018). Performance of these approaches are usually highly dependent on the handcrafted features fed into the LSTM layer. Yang et al. (2017) extracted features including characters, character-level bi-gram, Part-of-Speech (POS), POS scores, adjacent and dependent word collocations. Later in 2018, the feature sets were further enlarged by incorporating new features like word segmentation and Gaussian exact Point-wise Mutual Information (ePMI, Fu et al., 2018b).

Grammatical Error Correction. Unlike the GED tasks, GEC tasks has been mostly treated as the machine translation problem. To the best of our knowledge, the multi-layer convolutional neural network accompanied by a large language model (Chollampatt and Ng, 2018) is considered as the first Neural Machine Translation (NMT)-like approach to handle GEC tasks in English. Then Ge et al. (2018) and Grundkiewicz and Junczys-Dowmunt (2018); Fu et al. (2018b) proposed to use recurrent neural networks, while recent work (Junczys-Dowmunt et al., 2018; Grundkiewicz et al., 2019; Lichtarge et al., 2019; Fu et al., 2018a) made use of the Transformer (Vaswani et al., 2017). Specially, GECToR (Omelianchuk et al., 2020), which considered the English GEC task as a sequential labeling problem, has obtained competitive results to previous GEC systems.

4 Methodology

The overall architecture of the developed system is depicted in Fig. 1. The proposed system can be functionally divided into three parts: GED, GEC, and post-processing. The GED framework is responsible for error diagnosis at detection, identification and position levels, while the GEC framework provides possible candidates for detected S and M errors. Finally, the post-processing module takes results from the GED and GEC frameworks and fuse them into the final form of the system outputs.

4.1 Synthetic Data Generation.

Pre-training on synthetic data is crucial for the present GEC and GED tasks since the parallel training data are still extremely scarce. It is found that the proposed basic GED models, Seq2Seq GEC models and GECToR models also benefit from synthetic data. Following previous work on English GEC tasks (Zhao et al., 2019; Grundkiewicz et al.,

Error type	Erroneous sentence	Correct sentence
R	我和妈妈是不像别的母女。 (Wǒ hé mā ma shì bù xiàng bié de mǔ nǚ.)	我和妈妈不像别的母女。 (Wǒ hé mā ma bù xiàng bié de mǔ nǚ.)
M	我同意后者主张。 (Wǒ tóng yì hòu zhě zhǔ zhāng.)	我同意后者的主张。 (Wǒ tóng yì hòu zhě de zhǔ zhāng.)
S	上周我的车刮疼啊。 (Shàng zhōu wǒ de chē guā téng a.)	上周我的车被刮了。 (Shàng zhōu wǒ de chē bèi guā le.)
W	我是还在学校上班。 (Wǒ shì hái zài xué xiào shàng bān.)	我还是在学校上班。 (Wǒ hái shì zài xué xiào shàng bān.)

Table 1: Example sentences with corresponding errors. Sequences in the bracket are the corresponding transliterations.

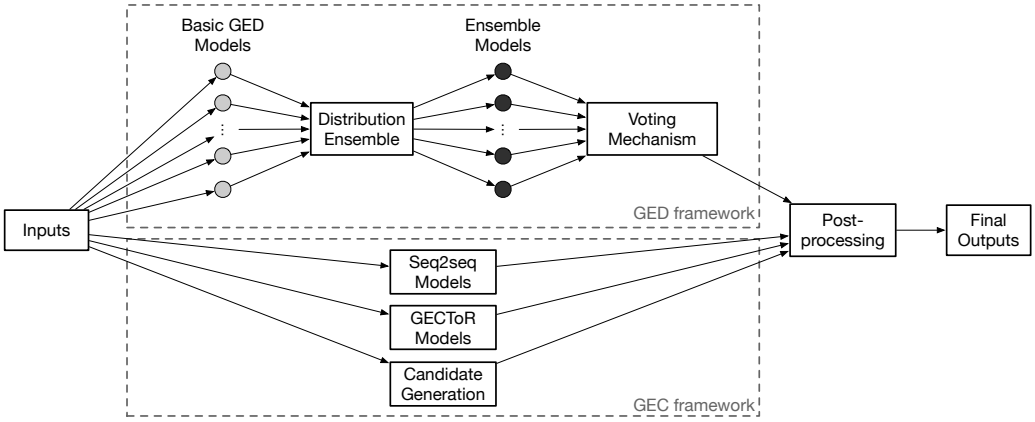

Figure 1: The overall architecture of the developed system.

2019; Xu et al., 2019), the synthetic data generation process in this work operates on two different levels, *i.e.*, word-level and character-level.

Word-level. At this level, error-free sentences are firstly segmented into words using the self-developed tokenizer. Then the following word-level errors are randomly added to the error-free sentences.

- Transposition: change the positions of words, where new positions are obtained by adding rounded bias to the original position values. The bias is sampled from a normal distribution with mean 0.0 and standard deviation 0.5.
- Deletion: delete a word.
- Insertion: add a word.
- Substitution: replace the current word with a random word in Chinese dictionary with a probability of 50%; replace the word with one of the synonyms generated by Chinese Synonyms toolkit [1] with a probability of 40%; replace the word with a word from its confusion set[2] with a probability of 10%.

The error probabilities of deletion and insertion are sampled from a normal distribution with mean 0.015 and standard deviation 0.2, while the error probability of substitution is sampled from a normal distribution with mean 0.075 and standard deviation 0.2.

Character-level. On top of the word-level errors, we also add the following character-level errors to 20% of the words, simulating spelling errors that occur in the real-world.

- Transposition: flip two consecutive characters existing in the current word with a probability of 10%.
- Deletion: delete a character in the word with a probability of 10%.
- Insertion: add a random Chinese character to the word with a probability of 10%.
- Substitution: substitute a character in the word with a probability of 30%, among which 70% of the characters are replaced by characters from their confusion sets [3], and the other 30%

[1] https://github.com/chatopera/Synonyms
[2] extracted from common mistakes made by students.
[3] http://nlp.ee.ncu.edu.tw/resource/csc.html

are replaced by random characters sampled from Chinese dictionary.

4.2 Grammatical Error Diagnosis

Basic GED Models. Recently, masked language models such as Bidirectional Encoder Representations from Transformers (BERT, Devlin et al., 2018), XLNet (Yang et al., 2019), and Generative Pre-Training 3 (GPT-3, Brown et al., 2020) have achieved superior performance on down-stream Natural Language Processing (NLP) tasks including question answering, language inference, sentence classification, *etc*.

To benefit from those efforts, we propose to use the BERT based sequential labeling model as our basic GED model rather than using the LSTM-CRF model. In general, BERT stacks 12 ($\text{BERT}_{\text{BASE}}$) or 24 ($\text{BERT}_{\text{LARGE}}$) identical Transformer blocks, which either takes a single sentence or a pair of sentences as input and outputs a hidden vector for each input token as well as a special [CLS] token for the whole input sentence (pair). Here, we denote the input sequence of Chinese characters as $\boldsymbol{X} = (x_1, x_2, ..., x_n)$, the final hidden vector generated by BERT as $\boldsymbol{H} = (h_1, h_2, ..., h_n)$, and the output BIO tags as $\boldsymbol{Y} = (y_1, y_2, ..., y_n)$. For better comprehension, we give some examples of BIO tags in Table 2. Then for an input token x_i and a specific BIO tag t, the conditional probability of x_i being labeled as t is derived using:

$$P\left(y_i = t|\boldsymbol{X}\right) = softmax\left(Wh_i + b\right). \quad (1)$$

Here, $\boldsymbol{X}$ denotes the input sequence, h_i is the final hidden state of BERT, W and b are model parameters. The tag with the largest conditional probability will be chosen as the final output corresponding to the input token x_i.

Distribution Ensemble. Top results are usually achieved by ensemble techniques (Zheng et al., 2016; Fu et al., 2018b), and this work also benefits from model ensemble approaches. Specifically, we assume that there are M different basic GED models $\{m_1, m_2, ..., m_M\}$. Then for each input sequence $\boldsymbol{X} = (x_1, x_2, ..., x_n)$, we have M output sequences $\{\boldsymbol{Y_1}, \boldsymbol{Y_2}, ..., \boldsymbol{Y_M}\}$. The distribution ensemble based on M different models can be written by:

$$P\left(y|\boldsymbol{X}\right) = \frac{1}{M}\sum_{k=1}^{M} P_k\left(y|\boldsymbol{X}; \boldsymbol{\theta_k}\right). \quad (2)$$

Here, $P\left(y|\boldsymbol{X}\right)$ denotes the conditional probability of final prediction, $\boldsymbol{\theta_k}$ indicates the trainable model parameters of kth model (m_k), and $P_k\left(y|\boldsymbol{X}; \boldsymbol{\theta_k}\right)$ is the conditional probability generated by model m_k.

Voting Mechanisms. Voting mechanisms are proposed for further improvement on overall performance, especially for model precisions. In this work, we explore the following two different voting mechanisms:

- Majority voting. In this mechanism, each output of the ensemble model is assigned the same weight, and the system selects the tag with the largest weight as the final output.
- Using F1-Score as weight. In this mechanism, we first evaluate the ensemble models using the development set and obtain corresponding F1 scores. Then the overall F1 scores serve as the weight for the ensemble models during the inference step.

4.3 Grammatical Error Correction

As shown in Figure 1, the GEC framework consists of Seq2Seq GEC models, GECToR models, and a candidates generation module.

Seq2Seq GEC Models. This work explores two kinds of Seq2Seq GEC models: one is the regular Transformer model (Vaswani et al., 2017), and the other is the copy augmented Transformer model (Zhao et al., 2019).

The attention-based Transformer is the most widely used sequence transduction model in Natural Language Processing (NLP) area that are capable of a broad spectrum of tasks (Vaswani et al., 2017; Lample et al., 2018; Yang et al., 2019; Devlin et al., 2018; Dai et al., 2019), including machine translation, text style transfer, reading comprehension, etc. Transformers employ Seq2Seq structures that are usually built up by stacking encoder and decoder layers. Encoder layers consist of a multi-head self-attention layer followed by a position-wise feed-forward layer, while decoder layers consist of a multi-head self-attention layer, a multi-head cross-attention layer and a position-wise feed-forward layer. Residual connections and layer normalizations are used to improve the performance of deep Transformers.

The copy-augmented Transformer was originally proposed for text summarization tasks (Gu et al.,

Input	因	为	,	雾	烟	刺	激	就	对	人	体	会	有	危	害	。
	(Yīn	wèi	,	wù	yān	cì	jī	jiù	duì	rén	tǐ	huì	yǒu	wēi	hài	.)
Output	O	O	O	B-S	I-S	O	O	O	B-W	I-W	I-W	I-W	O	O	O	O
Input	我	不	可	以	找	到	了	在	哪	里	我	会	买	菜	。	
	(Wǒ	bù	kě	yǐ	zhǎo	dào	le	zài	nǎ	lǐ	wǒ	huì	mǎi	cài	.)	
Output	O	B-S	I-S	I-S	O	O	B-M	B-S	O	O	B-R	I-R	O	O	O	

Table 2: Examples of BIO tags used in basic GED models. Sequences in the bracket are the corresponding transliterations.

2016; See et al., 2017) and subsequently revamped to handle GEC tasks (Zhao et al., 2019; Choe et al., 2019). Unlike the normal Transformers, copy-augmented Transformers are able to copy units (*e.g.* characters, sub-words, or words) from the source sentence, since the final probability distribution of a unit is the combination of a generative distribution and a copy distribution, balanced by a factor $\alpha^{copy} \in [0, 1]$. With a larger copy factor, the output units tend to copy from the source rather than generating their own, and vice versa.

GECToR Models. Similar to the Parallel Iterative Edit (PIE) model (Awasthi et al., 2019), GECToR (Omelianchuk et al., 2020) treats the GEC task as a sequential labeling problem. The core of the approach is the design of special output tags, which indicate the differences between source sentences and target sentences. In order to obtain the tags, minimal edits of the characters are firstly extracted based on the modified Levenshtein distance. Then the edits are converted to the following tags:

- $KEEP, indicates that the character is unchanged.
- $APPEND_X, indicates that there is a character X missing after the current character.
- $REPLACE_X, indicates that the current character should be replaced by character X.
- $REORDER, indicates that the current character is a part of the chunk where the reorder error occurs.
- $DELETE, indicates that the current character should be removed.

Identical to the basic GED models, GECToR model also stacks the fully connected layer and the softmax layer over the Transformer encoder.

Candidate Generation. During the experiment, we found that the set of correction candidates shares a large overlap across each year's training data. It is also consistent with intuition since there exist commonly confused words or characters in Chinese. To make use of this observation, we propose a candidate generation module based on a Chinese language model. Firstly, a Chinese character-level 5-gram language model (denoted by L in the following) is trained based on 30 million Chinese sentences. Then L is used to select the k most appropriate candidate words from a large set of candidates, which is extracted from the CGED training data, to replace the words in the original sentences according to the error type and position in the detection phase. Finally, the candidates along with those generated by Seq2Seq models and GECToR models are all sent to the post-processing module to obtain the final output.

4.4 Post-processing

Post-processing Outputs of GED Models. Considering that one input token is allowed to be labeled as multiple error types depending on the actual situation, we propose to apply the following heuristics to the outputs of the GED framework in the post-processing stage.

1. If current tag O is followed by a tag I-X and the last tag is B-X, where X indicates a specific error type, then the current tag will be replaced by I-X.
2. If one tag set is nested into another one, they will be decomposed based on their starting and ending points. For example, when the following case happens, (1, 4, X_1) and (2, 3, X_2) are extracted as the final outputs instead of (1, 1, X_1) and (2, 3, X_2).

Tags:	B-X_1	B-X_2	I-X_2	I-X_1
Position:	1	2	3	4

Re-ranking the Correction Candidates. To re-rank and select the elite candidates from those proposed by the three GEC models, this work proposes

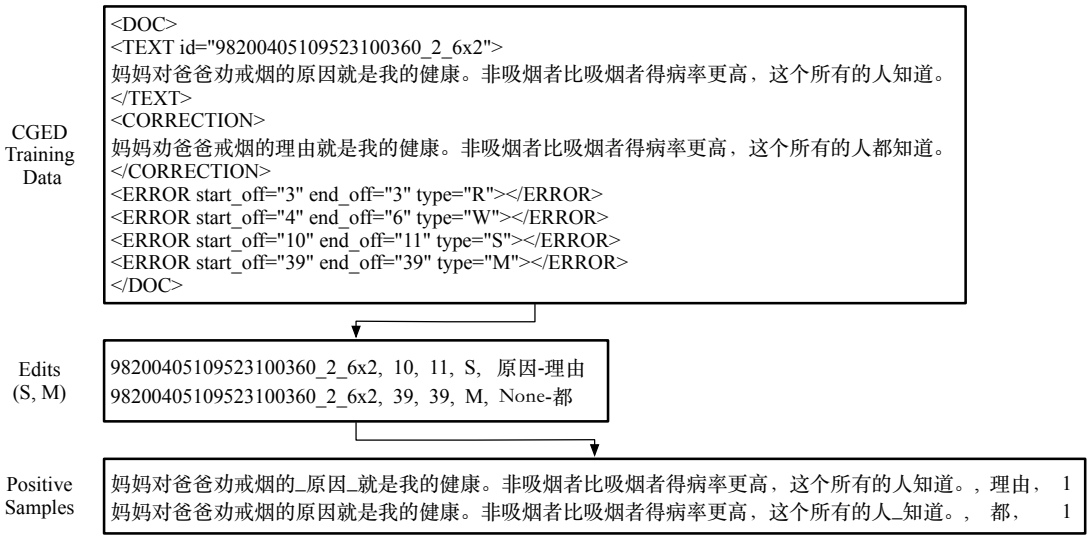

Figure 2: Example of positive data generation process.

a Chinese BERT-based [4] scoring model. The model takes a sentence and the corresponding correction candidate as input and returns the candidate's score, which lies between 0.0 and 1.0.

Since there is no ready-made data for training this kind of scoring model, it leads us to the data generation process. There are basically two kinds of data needed by the model, including positive samples and negative samples.

Positive samples can be directly generated from the CGED training data based on the process depicted in Fig. 2. We firstly design a rule-based system to extract word-level edits from the training data. Obviously, extracted edits will include all kinds of errors (R, S, W and M). However, we only keep the edits related to S and M errors, since R and W errors are not taken into consideration in the correction task. Each edit can then be converted to a training sample, which can be denoted as a triple (s, w, t), where s indicates the input sentence, w is the correction candidate, and t is the fitness score of the candidate. Specifically, for the input sentence s, we insert "_" to the left and right of the chunk where S error occurs, and we add the "_" symbol to the position where the M error occurs, as shown in Fig. 2. Considering that all training data provided by CGED shared tasks are manually annotated data, we assign higher scores (1.0 in this work) to these candidates.

The model cannot be trained only using positive samples. Hence we propose a negative data generation algorithm, as shown in Algorithm 1. Here we define D_p as the collection of all positive training data, W_p as the collection of all the candidate words in D_p, S_{pe} as the collection of all the sentences in D_p. For each input sentence s in S_{pe}, we score every word in the candidate set to find out unsuitable candidates for s. More specifically, a new sentence s_{sub}, which is reconstructed by substituting the corresponding word in s with the candidate word or inserting the candidate word to s, is scored by L. Then, we select k candidates which have the lowest scores. Finally, we randomly choose one candidate (*i.e.*, W_{cand}) from the k candidates, and form a negative sample (s, W_{cand}, 0.0) for the proposed scoring model.

Algorithm 1 Negative Training Data Generation

Input: S_{pe}, W_p, L, k
Output: negative training data D_n
$D_n \leftarrow \{\}$
$PP(t) \leftarrow$ score of sentence t calculated by L
for i in range($len\,(S_{pe})$) **do**
    $s \leftarrow S_{pe}[i]$
    for w $\in W_p$ **do**
        $s_{sub} \leftarrow$ replace the word in s with w
        $p_{s_{sub}} \leftarrow PP(s_{sub})$
    end for
    $S_{top_k} \leftarrow$ top k from W_p based on $p_{s_{sub}}$
    $w_{cand} \leftarrow random.sample(S_{top_k}, 1)$
    $D_n \leftarrow D_n + (s, w_{cand}, 0)$
end for

Inspired by the idea of "next sentence prediction" task (Devlin et al., 2018), we concatenate the input sentence s and the correction candidate w as a pair S_{pair}, and then feed it into our scoring model. Fig.3 demonstrates the architecture of the proposed scoring model as well as an example of S_{pair}.

[4] https://github.com/google-research/bert/blob/master/multilingual.md

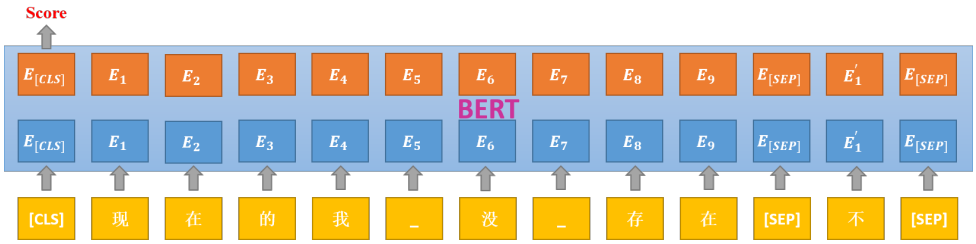

Figure 3: The architecture of the proposed scoring model.

Input	吸烟不但对自己的健康好处，而且给非吸烟者带来不好的影响。 (Xī yān bù dàn duì zì jǐ de jiàn kāng hǎo chù , ér qiě gěi fēi xī yān zhě dài lái bu hǎo de yǐng xiǎng.)
Output (Seq2Seq)	吸烟不但对自己的健康不好，而且会给非吸烟者带来不好的影响。 (Xī yān bù dàn duì zì jǐ de jiàn kāng bù hǎo, ér qiě huì gěi fēi xī yān zhě dài lái bu hǎo de yǐng xiǎng.)
Edits (Seq2Seq)	(11, 12, S, 好处-不好) (16, 16, M, None-会)
Output (GECToR)	吸烟不但对自己的健康不好，而且给不吸烟者带来不好的影响。 (Xī yān bù dàn duì zì jǐ de jiàn kāng bù hǎo, ér qiě gěi bù xī yān zhě dài lái bu hǎo de yǐng xiǎng.)
Edits (GECToR)	(11, 12, S, 好处-不好) (17, 17, M, 非-不)
Candidate Generation	(11, 12, S, 好处-不好,有害,不利)
Output (GED)	(11, 12, S)
Final Output	(11, 12, S, 好处-不好)

Table 3: Example of fusion of results. Sequences in the bracket are the corresponding transliterations.

Seq2Seq models, GECToR models and the candidate generation module tend to produce different candidates. Hence in the re-ranking stage, the correction candidate and its corresponding input sentence are fed into the scoring model one by one. We then select the top three candidates with the highest score for each input sentence.

Fusion of Results. It should be noted that the training data and vocabulary of the GEC models are different. Therefore, directly applying the ensemble techniques is infeasible. Instead, we propose to obtain the final edits by the following three steps. First, the corrected sentences produced by multiple GEC models are aligned with the original ones and the edits are extracted automatically by our rule-based extraction system. We also generate several edits with the candidate generation module based on the results of the detection phase. Second, we fuse these edits based on their error positions and types. In other words, a series of candidate words are generated for each error position. Third, we discard the edits that are not consistent with the detection results. This step is vital since the training processes of Seq2Seq models and GECToR models are completely independent and may produce conflict edits.

To improve the accuracy of correction candidates, we set a threshold to filter the candidates with less confidence. Finally, we obtain the final fusion result after all the processes described above. Table 3 shows an example of the fusion process.

Training Data	#Error	#R	#M	#S	#W
2016	48,010	9,742	14,941	20,323	3,004
2017	26,449	5,852	7,010	11,592	1,995
2018	1,067	208	298	474	87
2020	2,909	678	801	1,228	201
Total	78,435	16,480	23,050	33,617	5,287
Test Data	**#Error**	**#R**	**#M**	**#S**	**#W**
2016	7,795	1,584	2,471	3,232	508
2017	4,871	1,060	1,269	2,156	386
2018	5,040	1,119	1,381	2,167	373
2020	3,654	769	864	1,694	327

Table 4: Statistics information of the CGED data.

5 Experiments

5.1 Datasets

The proposed system utilizes training data provided by the CGED-2016, CGED-2017, CGED-2018 and CGED-2020 shared tasks. Table 4 shows the statistics of training and test data. In this work, the CGED-2016 test set is used as the training data, while CGED-2017 and CGED-2018 test sets are used as the development set. Besides the data pro-

		Detection-level			Identification-level			Position-level		
Runs	FPR	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
1	0.2052	0.9387	0.8383	0.8857	0.7788	0.5503	0.6449	0.5145	0.2965	0.3762
2	0.2345	0.9319	0.8565	0.8926	0.7623	0.5678	0.6508	0.4822	0.3011	0.3707
3	0.2182	0.9357	0.8478	0.8896	0.7711	0.5577	0.6473	0.5011	0.2995	0.3749
Mean	0.2193	0.9354	0.8475	0.8893	0.7707	0.5586	0.6477	0.4993	0.2990	0.3739

	Correction-level (Top 1)			Correction-level (Top 3)		
Runs	Prec.	Rec.	F1	Prec.	Rec.	F1
1	0.3238	0.1290	0.1845	0.2982	0.1372	0.1879
2	0.3293	0.1263	0.1826	0.3132	0.1337	0.1874
3	0.3386	0.1259	0.1836	0.3217	0.1333	0.1885
Mean	0.3306	0.1271	0.1835	0.3110	0.1347	0.1879

Table 5: Overall performance of the developed system on CGED 2020 shared task.

vided by CGED shared task, we also utilize the data provided by the NLPCC-2018 shared task (Zhao et al., 2018)[5] to train our GEC models. Moreover, NetEase News Corpus is used to generate synthetic data.

5.2 Evaluation Metrics

As previously stated in Section 2, submitted results are evaluated at four different levels, *i.e.* detection-level, identification-level, position-level and correction-level. At each level, precision (Pre.), recall (Rec.) and F1 score are calculated based on the gold standard and the system outputs. Specially at the detection-level, false positive rate (FPR) as well as accuracy is calculated in addition to the above evaluation metrics.

5.3 Training Details

In this work, we utilize the Chinese pre-trained language models with large configuration (24 layers) including Robustly optimized BERT pre-training approach (RoBERTa, Liu et al., 2019)[6] and pre-training with whole word masking for Chinese BERT (Cui et al., 2019)[7] as the starting point of the fine-tuning process. We also tested Chinese BERT [8], but it resulted in poorer performance on the GED task than the above mentioned two models. We trained 30 basic GED models based on various pre-trained models along with different initialization seeds. Then we averaged the last several checkpoints of models and apply distribution ensemble on every 4 or 5 models. GEC models also follow similar steps to obtain final models.

5.4 Results

The overall performance of our developed system is given in Table 5. It can be seen that the system achieves F1 scores up to 0.8926, 0.6508 and 0.3762 at the detection-level, identification-level and position-level, respectively. As for the correction task, we achieve F1 scores of 0.1845 and 0.1879 at TOP1 and TOP3 correction track.

Among the 43 submissions for the detection task, our system rank 4 to 6 at detection, identification and position tracks, but rank 12 at FPR track. It is remarkable that this system achieves the highest precisions among the top 10 submissions at identification and position tracks. This system performs even better at the correction tracks. It achieves the highest F1 score also with the highest precision at TOP3 correction track. Besides, the system gets the highest precision with third-highest F1 score at TOP1 correction track, however, the gap is only 0.0046.

6 Conclusion

This paper describes our system on NLPTEA-2020 CGED shared task. To make the system more robust against data sparseness and lack of data, we adopt the synthetic data generation process during model training. Besides utilizing up-to-date model architectures, we also carefully optimized the system performance by employing ensemble techniques, voting mechanisms and rule-based post-processing. We plan to integrate more grammatical features into the GED and GEC models and optimize the post-processing algorithm to further improve the system performance.

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

[6] https://github.com/brightmart/roberta_zh

[7] https://github.com/ymcui/Chinese-BERT-wwm

[8] https://github.com/google-research/bert/blob/master/multilingual.md

References

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4260–4270, Hong Kong, China. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Computing Research Repository*, arXiv:2005.14165. Version 4.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.

Yo Joong Choe, Jiyeon Ham, Kyubyong Park, and Yeoil Yoon. 2019. A neural grammatical error correction system built on better pre-training and sequential transfer learning. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 213–227, Florence, Italy. Association for Computational Linguistics.

Shamil Chollampatt and Hwee Tou Ng. 2018. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, USA.

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. *Computing Research Repository*, arXiv:1906.08101.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019. Style transformer: Unpaired text style transfer without disentangled latent representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5997–6007, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *Computing Research Repository*, arXiv:1810.04805. Version 2.

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 *Natural Language Processing and Chinese Computing*, pages 341–350, Cham. Springer International Publishing.

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, Melbourne, Australia. Association for Computational Linguistics.

Tao Ge, Furu Wei, and Ming Zhou. 2018. 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)*, pages 1055–1065, Melbourne, Australia. Association for Computational Linguistics.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2018. Near human-level performance in grammatical error correction with hybrid machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 284–290, New Orleans, Louisiana. Association for Computational Linguistics.

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, Florence, Italy. Association for Computational Linguistics.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

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)*, pages 595–606, New Orleans, Louisiana. Association for Computational Linguistics.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, Brussels, Belgium. Association for Computational Linguistics.

Lung-Hao Lee, Gaoqi Rao, Liang-Chih Yu, Endong Xun, Baolin Zhang, and Li-Ping Chang. 2016. Overview of NLP-TEA 2016 shared task for Chinese grammatical error diagnosis. In *Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016)*, pages 40–48, Osaka, Japan. The COLING 2016 Organizing Committee.

Lung-Hao Lee, Liang-Chih Yu, and Li-Ping Chang. 2015. Overview of the NLP-TEA 2015 shared task for Chinese grammatical error diagnosis. In *Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications*, pages 1–6, Beijing, China. Association for Computational Linguistics.

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, Melbourne, Australia. Association for Computational Linguistics.

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, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3291–3301, Minneapolis, Minnesota. Association for Computational Linguistics.

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. *Computing Research Repository*, arXiv:1907.11692.

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 *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem 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, Seattle, WA, USA â†' Online. Association for Computational Linguistics.

Gaoqi Rao, Qi Gong, Baolin Zhang, and Endong Xun. 2018. Overview of NLPTEA-2018 share task Chinese grammatical error diagnosis. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 42–51, Melbourne, Australia. Association for Computational Linguistics.

Gaoqi Rao, Baolin Zhang, Endong Xun, and Lung-Hao Lee. 2017. IJCNLP-2017 task 1: Chinese grammatical error diagnosis. In *Proceedings of the IJCNLP 2017, Shared Tasks*, pages 1–8, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Shuyao Xu, Jiehao Zhang, Jin Chen, and Long Qin. 2019. Erroneous data generation for grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 149–158, Florence, Italy. Association for Computational Linguistics.

Yi Yang, Pengjun Xie, Jun Tao, Guangwei Xu, Linlin Li, and Luo Si. 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, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for

language understanding. In *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc.

Liang-Chih Yu, Lung-Hao Lee, and Liping Chang. 2014. Overview of grammatical error diagnosis for learning Chinese as a foreign language. In *Proceedings of the 1st Workshop on Natural Language Processing Techniques for Educational Applications*, pages 42–47, Nara, Japan.

Yongwei Zhang, Qinan Hu, Fang Liu, and Yueguo Gu. 2018. CMMC-BDRC solution to the NLP-TEA-2018 Chinese grammatical error diagnosis task. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 180–187, Melbourne, Australia. Association for Computational Linguistics.

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. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 156–165, Minneapolis, Minnesota. Association for Computational Linguistics.

Yuanyuan Zhao, Nan Jiang, Weiwei Sun, and Xiaojun Wan. 2018. Overview of the nlpcc 2018 shared task: Grammatical error correction. In *Natural Language Processing and Chinese Computing*, pages 439–445, Cham. Springer International Publishing.

Bo Zheng, Wanxiang Che, Jiang Guo, and Ting Liu. 2016. 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)*, pages 49–56, Osaka, Japan. The COLING 2016 Organizing Committee.