Chinese Grammatical Error Correction Based on Hybrid Models with Data Augmentation

Yi Wang[1,3,*], **Ruibin Yuan**[2,*], **YanGen Luo**[1,3,*], **YuFang Qin**[1,3,*], **NianYong Zhu**[1,3], **Peng Cheng**[1,3], and **Lihuan Wang**[1,3]

[1]State Key Laboratory of Media Convergence Production Technology and Systems Xinhua News Agency,Beijng,100077,China.

[2]Stardust.ai ,Beijing, China.

[3]The Technical Bureau of Xinhua News Agency

[*]These authors contributed to the work equally and should be regarded as co-first authors.

Abstract

A better Chinese Grammatical Error Diagnosis (CGED) system for automatic Grammatical Error Correction (GEC) can benefit foreign Chinese learners and lower Chinese learning barriers. In this paper, we introduce our solution to the CGED2020 Shared Task Grammatical Error Correction in detail. The task aims to detect and correct grammatical errors that occur in essays written by foreign Chinese learners. Our solution combined data augmentation methods, spelling check methods, and generative grammatical correction methods, and achieved the best recall score in the Top 1 Correction track. Our final result ranked fourth among the participants.

1 Introduction

In recent years, a global upsurge of Chinese learning has been set off. However, due to the language environment and language structure differences between countries, foreign Chinese learners are more prone to grammatical errors. Traditional grammatical error correction mainly relies on rule-based methods and performs poorly. Therefore, a better Chinese grammar diagnosis system is needed. Thanks to NLPTEA, the Chinese Grammatical Error Diagnosis (CGED) shared task provides a free communication platform for computing technology researchers in natural language processing (NLP) to seek more advanced Chinese grammar diagnosis solutions.

Due to the deficiency of parallel corpora, Chinese GEC often used statistical methods and rule-based methods in the early stage. Until recently, with larger-scale parallel corpora developed, machine learning techniques were applied to the Chinese GEC task. Chen(Zheng et al., 2016) used an approach based on the conditional random fields (CRF) model. The model added a collocation feature in order to better identify grammatical errors in word choice. Zhengn(Zheng et al., 2016) used a CRF based model, along with an RNN based model and an ensemble model, reached high F1-scores and recall rates across the three assessment levels of the NLP-TEA-3 shared task. Yang(Yang et al., 2017) leveraged a bi-LSTM-CRF model. Spliced word vectors with manual features such as bi-gram, POS, and PMI were added during training. Fu(Fu et al., 2018) also used a bi-LSTM-CRF model, with ePMI values integrated. They obtained promising results in the CGED2018 shared

Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications, pages 78–86
Suzhou, China, December 4, 2020.

task.

In this paper, we introduce our solution to the CGED2020 shared tasks. By combining data augmentation methods, spelling check methods, and generative grammatical correction methods, we achieved the best recall score in the Top 1 Correction track. Our final result ranked fourth among the participants. The rest of this article is organized as follows: Section 2 describes the shared tasks of CGED2020. Section 3 describes the methods used in this paper, including data preprocessing, data augmentation, and various deep learning error correction models. Section 4 conducts experiments on the methods mentioned above. In Section 5, the conclusion and the planning for future works are given.

2 Task Definition

The CGED2020 shared task is the sixth Chinese grammar diagnosis error competition, held since 2014. This task provides participants a shared data set using the writing part of Hanyu Shuiping Kaoshi (HSK). The goal and direction of the task are to use modern NLP techniques to detect foreign Chinese learners' grammatical errors in Chinese writing and build an automatic Chinese grammatical error diagnosis system. It mainly distinguishes four different types of errors, including Redundant Words (R), Missing Words(M), Word Selection (S), and Word Order Error(W). On this basis, a comprehensive evaluation is carried out according to these dimensions of error judgment of the problem sentence, error type analysis, the error location, and sentence modification suggestions. The detailed sample is shown in Table 1.

The criteria for judging correctness are determined at three levels as follows.

(1) Detection-level: Binary classification of a given sentence, correct or incorrect, should be completely identical with the gold standard. All error types will be regarded as incorrect.

(2) Identification-level: This level could be considered as a multi-class categorization problem. All error types should be identified according to the gold standard.

(3) Position-level: In addition to identifying the error types, this level also judges the grammatical error's occurrence range.

3 Methodology

3.1 Data Preparation

We use the dataset from Ren (Ren et al., 2018), which contains 1.3 million sentence pairs collected from Lang-8 and HSK. Native Chinese speakers wrote news articles published by the Xinhua News Agency during 2017 and 2018, and compositions are collected for data augmentation. The former contains 6 million sentence pairs, and the latter contains 1 million sentence pairs. Texts are split into sentences, and all the non-Chinese, non-English, and non-punctuation characters in the sentences are removed. Also, sentences that are longer than 64 characters are discarded. Finally, we randomly choose 10000 pairs of sentences from non-augmented data and equally split them into validation and testing sets. The rest is used for training.

3.2 Data Augmentation

Obtaining adequate parallel data for deep learning-based GEC models is a challenging task, especially in the Chinese language. To mitigate the problem, a data augmentation scheme is applied. In this paper, we combine both rule-based and neural network-based

Error type	Error Sentence	Correct Sentence	Error location
M (missing word)	总之抽烟可以帮助所有的人了解到**对**环境的污染。	总之抽烟可以帮助所有的人了解到**它对**环境的污染。	16 - 16
R (redundant word)	现在必须得考虑怎样对待这种社会问题**的时期了**。|	现在必须得考虑怎样对待这种社会问题**了**。	18 - 20
S (word selection)	最重要的是**做**孩子想学，积极学习的环境。	最重要的是**创造**孩子想学，积极学习的环境。	6 - 6
W (word order)	**刚满 13 岁的对我**来说，流行歌曲跟我的生活非常密切。	**对刚满 13 岁的我**来说，流行歌曲跟我的生活非常密切。	1 - 7

Table 1: Error Examples

methods for generating noisy, ungrammatical texts from their clean counterparts. After the augmentation process, 6 million pairs of rule-based and 1 million pairs of neural network-based clean-corrupted sentences are obtained for training.

3.2.1 Rule Based Corruption

Inspired by previous work by Wang(Wang et al., 2019), we propose a rule-based corpora corruption method. Unlike Wang' s method, our method performs both word grain and character grain corruption and introduces sentence grain word ordering error to corpora. This method aims to obtain a large amount of parallel data with rich, diverse errors within a short time.

According to Wang, imbalanced error types will lead to low recall on the low-frequency error types. Therefore, the probability of each artificial error type are set to equal. As using low error rate data for training will cause the model to become too conservative, the corruption rate $P_{corrupt}$ is set to 0.4. With a sentence given, we obtain both character grain tokens t_c and word grain tokens t_w (using jieba). At each step, we corrupt a t_c or a t_w with a probability of $P_{corrupt}$. The corruption operations include, inserting a random character c_{rand} in the vocabulary V or a synonym syn (using synonyms) to the left of a token with a probability of p_r (redundant error type), replacing a token with a random character c_{rand} in V or a low similarity synonym syn_{low} with a probability of p_s (selection error type), deleting a token with a probability of p_m (missing error type), moving a token to a random position with a probability of p_w (word ordering error type). Algorithm 1 formalizes this method and Table 2 shows the corrupted results.

3.2.2 Neural Network Based Corruption

We train an attention-based sequence-to-sequence model with a bidirectional GRU encoder to generate noisy counterparts for clean sentences. This approach aims to generate realistic ungrammatical parallel corpora from clean corpora but is limited by the inference speed. Borrowing ideas from but being different from Xie(Xie et al., 2018) which used a noisy beam search scheme to introduce noise into the decoding stage, we define noisy score s_{noisy} as:

$$\mathrm{s_{noisy}} = \mathrm{s} - \tau\beta_{\mathrm{random}}$$

where s is the log-probability of a token, β_{random} is a scale factor and τ is a uniform random variable $\tau \sim U(0, 1)$. We use the beam size of 6 to balance between the decod-

Algorithm 1: Rule Based Corpora Corruption Method

Input: vocabulary V, clean sentences corpora C_{clean}, corruption rate $P_{corrupt}$, probability of redundant error type p_r , probability of selection error type p_s, probability of missing error type p_m, probability of word ordering error type p_w , synonym $syn()$ generator
Output: corrupted corpora C_{noisy}
Initialize $C_{noisy} = \{\}$
for each sentences s in C_{clean} ***do***
    $rand$ = Random(0,1)
    if $rand < P_{corrupt} \times p_w$ ***then***
        move a random word grain token t_w to a random position
    end
    $rand$ = Random(0,1)
    if rand $> P_{corrupt}$ ***then***
        continue
    end
    $rand$ = Random(0,1)
    if $rand < 0.5$ ***then***
        for each character grain token t_c in s ***do***
            $rand$ = Random(0,1)
            if rand $< p_r$ ***then***
            insert a token crand in V to the left of t_c
            else if $rand < p_r + p_s$ ***then***
            replace t_c with a token crand in V
            else if $rand < p_r + p_s + p_m$ ***then***
            delete t_c from s
            end if
        end
    else
        for each word grain token t_w in s ***do***
            $rand$ = Random(0,1)
            if $rand < p_r$ ***then***
            insert a synonym $syn = syn(t_w)$ to the left of t_w
            else if $rand < p_r + p_s$ ***then***
            replace t_w with a low similarity synonym $syn_{low} = syn(t_w)$
            else if $rand < p_r + p_s + p_m$ ***then***
            delete t_w from s
            end if
        end
    end if
    add s to C_{noisy}
end
return C_{noisy}

Input:	不过，特朗普无视礼仪，语出惊人，频戳痛点，不仅双边关系没能拉近，反而平添几分不和谐。
Corrupted:	不过，特朗普无视舆礼夷，语出惊人，频戳闽点，不仅双边关系没能拉近，反而平添几分炜不和谐 {。
Input:	本站比赛赛道以沙石路面为主，部分路段还设置了危险系数较高的巨石阵陡坡。
Corrupted:	本站比赛檄赛道沙石路面为主 < 部分路段还癫设叉置了步危系数较的巨石阵熘陡坡。
Input:	墨西哥总统培尼亚在首脑会议上说，今天美索美洲各国与会，正说明对话和开放是走向地区一体化的正确道路。
Corrupted:	墨西哥总统培在首脑会议上蕈, 今天鹅美洲衣与会, 渤正颢说对话妊开放是走向地区紧密结合一体化道路。
Input:	在穿着它跳舞、骑行、跑步、吃火锅时，也不会沾染汗渍或异味。
Corrupted:	在穿着跳舞、健行 =、跑步、吃火锅时“池也不会沾染狗或有毒气体]。
Input:	双方确认将敦促朝鲜遵守联合国安理会相关决议、放弃核武器和导弹计划。
Corrupted:	双方楚敦促朝鲜联合国安理会相关决议、放弃核武器和导弹计划。

Table 2: Rule based corrupted results.

ing speed and the performance. The best generation result is observed when $\beta_{random} = 3.6$. Also, coverage penalty $cp(X;Y)$ (Wu et al., 2016)is added to s_{noisy} after _EOS_ is predicted, which is computed by:

$$\mathrm{cp(X;Y)} = \beta \times \sum_{\mathrm{i=1}}^{\mathrm{|X|}} \log(\min(\sum_{\mathrm{j=1}}^{\mathrm{|Y|}} \mathrm{p_{i,j}}, 1.0))$$

where β is a scale factor, $|X|$ and $|Y|$ are the length of input sequence X and output sequence Y and $p_{i,j}$ is the attention probability of the j-th output word y_j on the i-th input word xi. Using coverage penalty can help avoid severe under-translation especially when the noisy score is used. We set $\beta = 0.3$. Examples of the neural network-based corruption results are shown in Table 3.

3.3 Generative Models

We employ two generative models, Lasertagger and Conv-Seq2Seq, for grammatical error correction.

Lasertagger proposed by Malmi (Malmi et al., 2019) employs a sequence tagging approach for GEC. It transforms the GEC problem into a text editing task, since the edit distance between an error sentence and its correction is always low. The method mainly combines a BERT encoder with an autoregressive Transformer decoder to predict three main edit operations: keeping a token, deleting a token, and adding a phrase before a token. We use $\mathrm{Lasertagger_{AR}}$ with RoBERTa(Cui et al., 2019) as backbone and set the vocab size to 1000 during the phrase extraction phase, other parameters are set to default.

Conv-Seq2Seq proposed by Gehring(Gehring et al., 2017) is a CNN based sequence to sequence (Seq2Seq) model. It treats the GEC problem as a machine translation task, and translates a noisy, ungrammatical sentence into a clean one. Different from RNN based Seq2Seq models, it is more efficient during the training stage and more sensitive to local n-gram features. Previous work by Ren (Ren et al., 2018) has shown its effectiveness. We use the same setting as Ren during train.

3.4 Spelling Check Models

We also employ spelling check models, SpellGCN and Electra, for better dealing with spelling errors. Note that when applying spell checkers to GEC, we use a post-process strategy to filter false positive samples, including modification of person name, or number or “他他它”, and modification which makes target sentence segment length larger than raw sentence segment length.

SpellGCN(Cheng et al., 2020) divides each similar character set into two similarity graphs, one for similar pronunciation and one

Input: <go> 歌曲使人感到快乐，<eos> Predicted: 歌曲被人很快乐，<eos>
Input: <go> 现在知道你讲中文很风趣了！<eos> Predicted: 今天我知道你的讲中文很风趣！<eos>
Input: <go> 你是哪里人？<eos> Predicted: 你是那里人？<eos>
Input: <go> 手下们都累得气喘吁吁的时候，那位老人的身份被公开了。<eos> Predicted: 下手都累得气喘吁吁的时候，那位老人被身分公开了。<eos>
Input: <go> 我们一起玩了一些时候，然后吃了晚饭。<eos> Predicted: 所以我们一起玩一些时间，然后就吃晚饭。<eos>
Input: <go> 关于中国政府奖学金申请，我有几个问题希望得到您的解答。<eos> Predicted: 有关于中国政府奖学金申请，我有几个问题希望得到您的答应。<eos>
Input: <go> 我害怕如果我们没有进展，那样我会很难过 <eos> Predicted: 我害怕如果我们不可以进展，不过我很难过 <eos>
Input: <go> 考虑到韩国的老年贫困率和老年自杀率位居世界第一这一点，情况就特别严重。<eos> Predicted: 考虑韩国的老年贫困率老年的杀害率位于世界第一这一点，情况特别严重。<eos>
Input: <go> 尽管我有了非常多空闲，但我还没写完我答应要发送给你的故事。<eos> Predicted: 无论我有非常多空，我还没写完我答应要送给你的故事。<eos>

Table 3: Neural network-based corruption results.

for similar shape. Then it takes the graphs as input and generates an embedding for each character after the interaction between similar characters. These embeddings are then constructed into a character classifier for the semantic representation extracted from another backbone module. With the Combination of graph representation and BERT, SpellGCN can leverage the similarity knowledge and generate the right corrections accordingly. We use the default setting as Cheng, and a fine tuned BERT by Xu(Ming, 2020) as the backbone.

Electra(Clark et al., 2020) is a pre-training language model with a new pre-training task and framework, which changed the generative masked language model (MLM) pre-training task into the discriminant replaced token detection (RTD) task to determine whether the current token has been replaced by the language model. Experiments of paper show that the context representation learned by Electra is much better than the context representation learned by Bert and XL-net under the same model size, data, and calculation conditions. We use the Chinese version Electra-base model released by iFLYTEK Joint Laboratory of Harbin Institute of technology for spelling check.

3.5 N-Best Reranker

With the mentioned spelling check models and generative models, we use a recursive method for Chinese GEC as shown in Figure 1. For a given input sentence, it will first go through two spelling check models in a parallel fashion. Then the post spelling check output with the original sentence will go through the generative models, also in parallel. The whole process can loop for K (K=3) rounds to obtain N (N=6) output sentences. A MERT reranker is deployed to rerank N sentences, and we select the top-1 sentence as the correction result. Note that we use a post processing script to transform the result to present the detection and position level result. Several features are introduced during reranking: 1. Normalized 4-gram language model score divided by sentence length, 2. Edit operations including character add/delete/swap count from

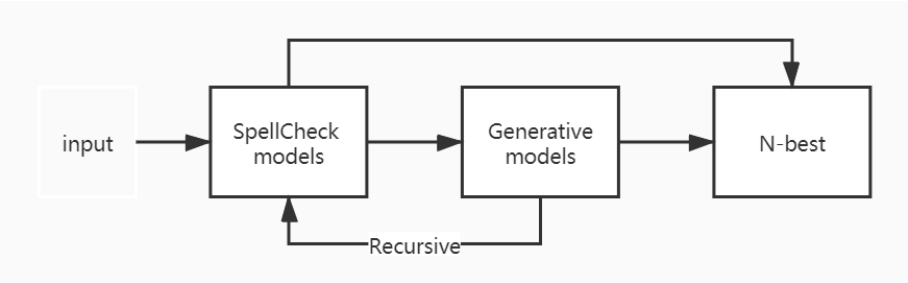

Figure 1: System diagram of our GEC system

source sentence to target sentence, 3. mask-predicted Bert probability score of a target sentence (Chollampatt et al., 2019), 4. Target sentence length penalty. The reranker is based on MERT from Moses (Koehn et al., 2007), with tuning metric of M2 F1-score.

Also, a more simple reranking approach is deployed to choose the best sentence according to the mask-predicted Bert score, which is computed by:

$$\mathrm{S_{BERT}(Y)} = \sum_{\mathrm{i=1}}^{\mathrm{[Y]}} \log \mathrm{P_{BERT}(y_i|Y_{i-masked})}$$

where Y is the target sentence, $Y_{i-msked}$ is the target sentence with i-th word y_i being masked, P_{Bert} is the Bert output probability.

4 Experiments

We first train Lasertagger on HSK+Lang8 dataset to obtain LASER_RAW. Then we add spelling checkers to obtain LASER_RAW+SPELL, which boosts each score by about 0.03. We also train a Lasertagger with augmented data only to obtain LASER_AUG. The promising result shows the effectiveness of our data augmentation methods. After adding spelling checkers, denoted as LASER_AUG+SPELL, we obtain the best correction score, which is 0.1993. Also, we try to pretrain a Lasertagger on augmented data for 10 epoch and fine tune on real data (HSK+Lang8), denoted by LASER_FINE. The model performs better than LASER_RAW and LASER_AUG on detection level and identification level, but worse in correction level. With spelling checkers adding in, we obtain LASER_FINE+SPELL, with a similar boosting effect as previous results. Finally, we add in Conv-Seq2Seq and the two rerankers and denoted as MERT_RERANK and SIMPLE_RERANK. We observe a boosting effect on each level except for a significant downgrade in correction level when using MERT. We submitted the MERT_RERANK as the final result. The testing results are shown in Table 4.

We first train Lasertagger on HSK+Lang8 dataset to obtain LASER_RAW. Then we add spelling checkers to obtain LASER_RAW+SPELL, which boosts each score by about 0.03. We also train a Lasertagger with augmented data only to obtain LASER_AUG. The promising result shows the effectiveness of our data augmentation methods. After adding spelling checkers, denoted as LASER_AUG+SPELL, we obtain the best correction score, which is 0.1993. Also, we try to pretrain a Lasertagger on augmented data for 10 epoch and fine tune on real data (HSK+Lang8), denoted by LASER_FINE. The model performs better than LASER_RAW and LASER_AUG on de-

EXPS	Detection	Identification	Position	Correction Top1
LASER_RAW	0.8258	0.5080	0.2375	0.1332
LASER_RAW+SPELL	0.8517	0.5410	0.2649	0.1668
LASER_AUG	0.8221	0.5319	0.2595	0.1826
LASER_AUG+SPELL	0.8475	0.5672	0.2799	**0.1993**
LASER_FINE	0.8597	0.5610	0.2531	0.1602
LASER_FINE+SPELL	0.8731	0.5837	0.2727	0.1857
MERT_RERANK	**0.8852**	**0.6203**	0.2812	0.1683
SIMPLE_RERANK	0.8846	0.5966	**0.3009**	0.1976

Table 4: Testing results on CGED2020 testing set.

tection level and identification level, but worse in correction level. With spelling checkers adding in, we obtain LASER_FINE+SPELL, with similar boosting effect as previous results. Finally, we add in Conv-Seq2Seq and the two rerankers and denoted as MERT_RERANK and SIMPLE_RERANK. We observe a boosting effect in each level except for a significant down grade in correction level when using MERT. We submitted the MERT_RERANK as final result. Testing results are shown in Table 2.

Since our pipeline does not work well in the FPR track (FPR$\geq$ 0.7068) and performance downgrade is observed, we look into the MERT reranker results. We find that the 4-gram model does not perform well. It tends to give shorter sentences higher scores, and it was trained on news domain data, so domain adaption can be an issue.

5 Conclusion and Future Works

This paper describes our system in the CGED2020 Shared Task Grammatical Error Correction. We explored a scheme by combining data augmentation methods, spelling check methods, and generative grammatical correction methods. We achieved the best recall score and our final result ranked fourth. However, there are still many efforts needed to solve this problem. A lot of improvements can be made to our current model. In the future, we will continue working on this problem. Possible future directions include improving data augmentation methods, finding a better reranking strategy, and finding better measurements for evaluation.

References

Xingyi Cheng, Weidi Xu, Kunlong Chen, Shaohua Jiang, Feng Wang, Taifeng Wang, Wei Chu, and Yuan Qi. 2020. Spellgcn: Incorporating phonological and visual similarities into language models for chinese spelling check.

Shamil Chollampatt, Weiqi Wang, and Hwee Ng. 2019. Cross-sentence grammatical error correction. pages 435–445.

Kevin Clark, Minh Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators.

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.

Ruiji Fu, Zhengqi Pei, Jiefu Gong, Wei Song, Dechuan Teng, W. Che, S. Wang, G. Hu, and T. Liu. 2018. Chinese grammatical error diagnosis using statistical and prior knowledge driven features with probabilistic ensemble enhancement. In *NLP-TEA@ACL*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer,

OndÅ™ej Bojar, Alex Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing.

Xu Ming. 2020. pycorrector: Text correction tool [software].

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*.

Chencheng Wang, Liner Yang, Yun Chen, Yongping Du, and Erhong Yang. 2019. Controllable data synthesis method for grammatical error correction.

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

Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. 2018. Noising and denoising natural language: Diverse back-translation for grammar correction. pages 619–628.

Yi Yang, Pengjun Xie, J. Tao, Guangwei Xu, L. Li, and S. Luo. 2017. Alibaba at ijcnlp-2017 task 1: Embedding grammatical features into lstms for chinese grammatical error diagnosis task. In *IJCNLP*.

Bo Zheng, W. Che, Jiang Guo, and T. Liu. 2016. Chinese grammatical error diagnosis with long short-term memory networks. In *NLP-TEA@COLING*.