Youdao's Winning Solution to the NLPCC-2018 Task 2 Challenge: A Neural Machine Translation Approach to Chinese Grammatical Error Correction

Kai Fu(✉), Jin Huang, and Yitao Duan

NetEase Youdao Information Technology (Beijing) Co., LTD, Beijing, China
{fukai,huangjin,duan}@rd.netease.com

Abstract. The NLPCC 2018 Chinese Grammatical Error Correction (CGEC) shared task seeks the best solution to detecting and correcting grammatical errors in Chinese essays written by non-native Chinese speakers. This paper describes Youdao NLP team's approach to this challenge, which won the 1st place in the contest. Overall, we cast the problem as a machine translation task. We use a staged approach and design specific modules targeting at particular errors, including spelling, grammatical, etc. The task uses M^2 Scorer [5] to evaluate every system's performance, and our final solution achieves the highest recall and $F_{0.5}$.

Keywords: Grammatical error correction · Machine translation

1 Introduction

Chinese is the most spoken language in the world. With the growing trend in economic globalization, more and more non-native Chinese speakers are learning Chinese. However, Chinese is also one of the most ancient and complex languages in the world. It is very different from other languages in both spelling and syntactic structure. For example, unlike English or other western languages, there is no different forms of plurality and verb tenses in Chinese. Also, reiterative locution is much more common in Chinese than it is in e.g., English. Because of these differences, it is very common for non-native Chinese speakers to make grammatical errors when using Chinese. Effective Chinese Grammatical Error Correction (CGEC) systems can provide instant feedback to the learners and are of great value during the learning process.

However, there are much fewer studies on Chinese grammatical error correction compared with the study of English grammatical error correction. Relevant resources are also scarce. The NLPCC 2018 CGEC shared task provides researchers with both platforms and data to investigate the problem more thoroughly. The goal is to detect and correct grammatical errors present in Chinese essays written by non-native speakers of Mandarin Chinese. Performance is evaluated by computing the overlap between a system's output sequence and the gold standard.

M. Zhang et al. (Eds.): NLPCC 2018, LNAI 11108, pp. 341–350, 2018.
https://doi.org/10.1007/978-3-319-99495-6_29

Youdao's NLP team has been actively studying language learning technologies as part of the Company's greater endeavour to advance online education with AI. Through careful analysis of the problem, we tackle it using a three-stage approach: first we remove the so called "surface errors" (e.g., spelling errors, to be elaborated later) from the input. We then cast the grammatical error correction problem as a machine translation task and apply a sequence-to-sequence model. We build several models for the second stage using different configurations. Finally, those models are combined to produce the final output. With careful tuning, our system achieves the highest recall and $F_{0.5}$, ranking first in the task.

This paper describes our solution. It is organized as follows. Section 2 describes the task, as well as the corresponding data format. Section 3 describes the related research work on grammatical error correction. Section 4 illustrates how our whole system works. Section 5 presents evaluation results. We summarize in Sect. 6.

2 Chinese Grammatical Error Correction

Although Chinese Grammatical Error Diagnosis (CGED) task has been held for a few years, this is the first time correction is introduced into the challenge. The CGEC task aims at detecting and *correcting* grammatical errors in Chinese essays written by non-native Chinese speakers. The task provides annotated training data and unlabeled test data. Each participant is required to submit the corrected text on the test data.

The training data consists of sentences written by Chinese learners and corrected sentences revised by native Chinese speakers. It should be noted that there may be $0 \sim N$ corrected results for the sentences. Specifically, the distribution of original sentences and corrected sentences in the training data is shown in Table 1, and typical examples of the data are shown in Table 2.

Table 1. Overview of the training data.

Corrected Sentence	Sentences Number
0	123,500
1	299,789
2	170,261
3+	123,691
Total	717,241

The task uses M^2 Scorer [5] to evaluate every system's performance. It evaluates correction system at the phrase level in terms of correct edits, gold edits, and use these edits to calculate $F_{0.5}$ for each participant.

Table 2. Samples from training data.

Original Sentence	Corrected Sentences
我从去年3月开始学汉语	
请把我修改一下！	请帮我修改一下
他们是离婚了，所以不一起住	他们今年离婚了，所以不一起住 他们已经离婚了，所以不住在一起。 他们离婚了，所以不一起住

3 Related Work

Grammar Error Correction (GEC) task has been attracting more and more attention since the CoNLL 2013–2014 Shared task. Most earlier GEC systems build specific classifiers for different errors and combine these classifiers to form a hybrid system [11]. Later, some researchers begin to treat GEC as a translation problem and propose solutions based on Statistical Machine Translation (SMT) models [2]. Some achieve fairly good results with improved SMT [3]. Recently, with the development of deep learning, Neural Machine Translation (NMT) has emerged as a new paradigm in machine translation, outperforming SMT systems with great margin in terms of translation quality. Yuan and Briscoe [16] apply NMT to the GEC task. Specifically, they use a classical translation model framework: a bidirectional RNN encoder and an RNN decoder with attention. To address the issue of out of vocabulary (OOV) words, Ji [8] presents a GEC model based on hybrid NMT, combining both word and character level information. Chollampatt et al. [4] proposes using convolution neural network to better capture the local context via attention.

Until this year, studies on Chinese grammatical error problem have been focused on diagnosis, spearheaded by Chinese Grammatical Error Diagnosis shared task. Both Zheng [17] and Xie [15] treat CGED as a sequence labeling problem. Their solutions combine the traditional method of conditional random fields (CRF) and long short term memory (LSTM) network.

4 Methodology

In this paper, we regard the CGEC task as a translation problem. Specifically, we aim at letting the neural network learn the corresponding relation between wrong and corrected sentences, and translate the wrong sentence into the correct one. However, unlike in conventional machine translation task, the source sentences in GEC contain numerous types of errors. This is the nature of the GEC problem (otherwise there is no need to perform corrections). As a result, the apparent patterns in the GEC parallel corpus are far more sparse and difficult to learn. On the other hand, grammar is the higher level of abstraction of a language and there are only a few grammatical mistakes language learners tend to make. The traditional Chinese Grammatical Error Diagnosis (CGED) task deals

with only four types of grammatical errors: redundant words (R), missing words (M), bad word selection (S) and disordered words (W) [17]. Therefore once the surface errors (e.g., spelling errors) are removed, it becomes relatively easier for the model to learn to identify them. We thus use a three-stage approach: a preprocessing stage aimed to remove most of the surface errors (e.g., spelling and punctuation errors), transformation stage that identifies and corrects grammatical errors and ensemble stage that combines the above two stages to generate the final output. Separating the stages allows us to use different modules targeting at their specific goals and tuned individually. This results in better overall performance.

4.1 Data Preparation

During this task, in addition to the training data NLPCC provides, we make use of two public datasets:

Language Model. Language model is commonly used in the field of grammar correction since it's able to measure the probability of a word sequence. Specifically, a grammatically correct sentence will get a higher probability in language model while a grammatically incorrect or uncommon word sequence will reduce the probability of the sentence. We use a language model as an assistant model to provide features to score the results. The model we use is a character-based 5-gram Chinese language model trained on 25 million Chinese sentences crawled from the Internet.

Similar Character Set. Since Chinese is logographic, the causes of spelling errors are quite different from languages that are alphabetical such as English. For example, Chinese characters with similar shapes or pronunciations are often confused, even for native speakers. Also, since Chinese words are typically shorter (2 to 4 characters), the usual dictionary and edit-distance based spell correction method does not perform well. To this end, we design a specific algorithm for Chinese spell correction. Specifically, we obtain Similar Shape and Similar Pronunciation Chinese character set (generally referred to as the Similar Character Set (SCS)) from the SIGHAN 2013 CSC Datasets [9,14]. The following are some sample entries in the data:

- Similar Shape: 可， 何呵坷奇河柯苛阿倚寄崎荷椅畸啊婀蚵犄琦轲
- Similar Pronunciation: 隔， 革格咯骼膈葛鬲蛤

We use SCS to generate candidate spell corrections and the language model to pick the most probable one.

NLPCC Data Processing. Training a machine translation model requires parallel corpus in the form of a collection of (`srcSent, tgtSent`) pairs where `srcSent` is the source sentence and `tgtSent` the target sentence. The NLPCC 2018 CGEC shared task provides training corpus where each sentence is accompanied with 0 or more corrected sentence(s). The original data contains about 0.71 million sentences. We process the data and generate 1.22 million (`srcSent, tgtSent`) pairs

where `srcSent` is the sentence probably containing grammatical mistakes and `tgtSent` the corrected result. If there is no error in `srcSent`, `tgtSent` remains the same as `srcSent`. If there are multiple corrections for an incorrect sentence, multiple pairs are generated. Next, we use the character based 5-gram language model to filter out sentence pairs where the score of `srcSent` is significantly lower than that of `tgtSent`. After the data cleaning step, the data size is reduced to 0.76 million.

4.2 Spelling Error Correction

The main component in the preprocessing stage that removes most of the surface errors is a spelling correction model. For this we use a simple 5-gram language model. The probability of a character sequence W of length n is given by:

$$P(w_1, w_2, ..., w_n) = p(w_1)p(w_2|w_1) \cdots p(w_n|w_1, w_2, ..., w_{n-1}) \tag{1}$$

The perplexity of the sequence is defined as the geometric average of the inverse of the probability of the sequences:

$$PP(W) = P(w_1, w_2, ..., w_n)^{-\frac{1}{n}} \tag{2}$$

We will use $PP(W)$ as the language model score. Higher $PP(W)$ indicates less likely sentence.

To perform spelling error correction, we first divide the sentence x into characters. For each character c in x, we generate candidate substitution character set S_c using SCS. We then try to replace c in the sentence by every $c' \in S_c$. Among the sentences (including the original one) with the lowest perplexity will be selected.

4.3 Grammatical Error Correction Model

After removing the spelling errors, we treat the grammatical error correction task as a translation problem and use a Neural Machine Translation (NMT) model to correct the errors, i.e., "translating" an incorrect sentence into grammatically correct one. Recently, neural networks have shown superior performance in various NLP tasks and they have done especially well in sequence modeling such as machine translation. Generally, most neural translation models are based on the encoder-decoder paradigm, in which the encoder (a neural network) encodes the input sequence $(x_1, x_2, ..., x_n)$ into a sequence of hidden states $(h_1, h_2, ..., h_n)$ and the decoder (also a neural network) generates the output sequence $(y_1, y_2, ...y_m)$ based on hidden state. An obvious advantage of this framework is that it does not need to explicitly extract linguistic features.

There are several variants of NMT models. They can be based on Recurrent Neural Network (RNN) such as Long Short Term Memory (LSTM) or Gated Recurrent Unit (GRU) [1,10], or Convolutional Neural Network (CNN) [6,7]. The recent Transformer model is a new encoder-decoder framework based on

the self-attention mechanism, proposed by Google in 2017 [13]. Transformer has shown excellent capability and achieved state of the art performance in machine translation. Thus we adopt it for our task. However, our framework is general and once a new, more advanced MT model emerges, we can easily upgrade the system with the new model.

Specifically, when the transformer reads in a sequence, it encodes it by several self-attentional iterations. Decoding is done in a similar manner. The Attention mechanism (scaled dot-product Attention) is defined as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V \tag{3}$$

where Q represents the query matrix packed from individual queries, K the keys used for processing the query, and V the values to be retrieved. The transformer adopt multiple attention heads:

$$MultiHead(Q, K, V) = [head_1, ..., head_h]W^O \tag{4}$$

where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$. Besides, a residual connection and hierarchical normalization are added for each attentional layer.

We used the open source tensorflow-based implementation, tensor2tensor [1], to train the transformer model. The hidden size parameter is set to 800. All the other parameters are in the default configuration.

4.4 Models Ensemble and Reranking

NMT models can be configured in different ways to suit different situations. They can be character-based or word-based. To handle rare and out-of-vocabulary words, sub-words can also be used [12]. The general understanding in the machine translation community is that sub-word models perform the best. In the case of CGEC, however, we have to deal with various errors and each may be handled using different tools or configurations. For example, spelling and character level syntax errors in Chinese are not handled well by (sub-)word level models which do a good job at correcting word level grammatical errors. Therefore we take a hybrid approach and build several models using different configurations. We then use a reranking mechanism to select among the model results the best one for each error.

We build the following 5 models, which all take spelling error correction as the first step:

M1: Spelling Checker alone
M2: Spelling Checker + Character NMT
M3: Spelling Checker + Character + Sub-word NMTs
M4: Spelling Checker + Sub-word NMT
M5: Spelling Checker + Sub-word + Character NMTs

[1] https://github.com/tensorflow/tensor2tensor

M3 and M5 use the same models but in different order. They may produce different results since the input to a model can be altered by the models preceding it.

For an input sentence x, each of the five models above will output a corrected result. The reranking is simply scoring them using the 5-gram language model. The pipeline of this process is shown in Fig. 1.

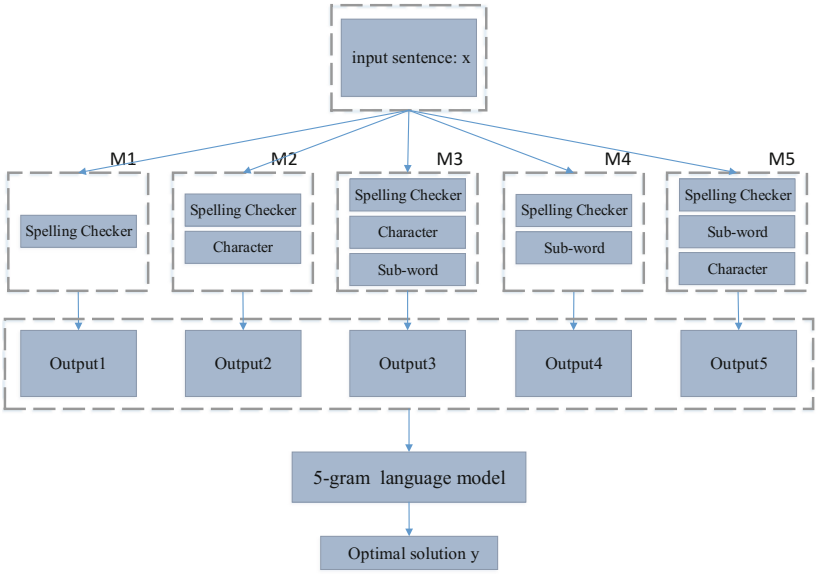

Fig. 1. Ensemble models.

5 Experiment Results

Among the 0.76 million sentence pairs generated according to the method described in Sect. 4.1, we take 3,000 pairs as the valid data set. The remaining are used for training. The validation set is mainly for parameter tuning and model selection.

The NLPCC 2018 CGEC shared task allows 3 submissions. Table 3 shows the performance of our models on the evaluation dataset for the 3 submissions. The differences among the models are mainly due to different selection strategies during the ensemble stage. Specifically, in all the cases, we use the 5-gram language model to score the outputs from the five individual models. S1 selects the result with the lowest perplexity. S2 behaves like S1 when the difference between the two lowest perplexity results is greater than a certain threshold (we set the threshold at a small value), otherwise it chooses the sentence with the *second* lowest perplexity. S3 behaves like S2 except that it makes a random selection

Table 3. Results on evaluation dataset.

Method	Annotator0			Annotator1			Annotator0,1		
	Precision	Recall	$F_{0.5}$	Precision	Recall	$F_{0.5}$	Precision	Recall	$F_{0.5}$
S1	34.17	**17.94**	**28.94**	34.30	**17.79**	**28.93**	35.24	**18.64**	**29.91**
S2	**34.18**	17.78	28.86	**34.40**	17.68	**28.93**	**35.34**	18.52	**29.91**
S3	34.16	17.73	28.82	34.33	17.60	28.85	35.28	18.45	29.83

between the outputs with the two lowest scores when the perplexity difference is less than the certain threshold. The purpose of this perturbation is to test the language model's selection capability.

Gold Standard is annotated by two annotators, denoted Annotator0 and 1 respectively. The union of the two is denoted Annotator0,1. S1 performs best, with the highest recalls and $F_{0.5}$ scores against all three annotations. S2 performs very closely. Both S1 and S2 are better than S3, showing that the language model is indeed capable of selecting correct sentences.

To evaluate contributions of each component, we test them individually. Table 4 shows the results. They all show significant performance drop if run individually. For example, the spelling checking model performs the worst and its $F_{0.5}$ score drops more than 15 % points compared with our overall system. This clearly shows that our staged approach is effective.

Table 4. Results of each component on evaluation dataset.

Method	Annotator0			Annotator1			Annotator0,1		
	Precision	Recall	$F_{0.5}$	Precision	Recall	$F_{0.5}$	Precision	Recall	$F_{0.5}$
Best model	34.17	**17.94**	**28.94**	34.30	**17.79**	**28.93**	35.24	**18.64**	**29.91**
Sub-word level NMT	30.26	10.82	22.26	30.85	10.89	22.57	31.66	11.40	23.36
Char level NMT	32.08	11.00	23.19	32.43	10.96	23.31	33.31	11.52	24.16
Spelling checker	**39.11**	4.17	14.61	**39.21**	4.11	14.49	**39.36**	4.24	14.83

6 Conclusion

This paper describes our solution to the NLPCC 2018 shared task 2. Ours is a staged approach. We first use a spelling error correction model to remove the spelling mistake. This reduces perturbation to later models and allows them to perform better. We then cast the problem into a translation task and use neural machine translation models to correct the grammatical errors. Experiments demonstrate that each stage plays a significant role. Our solution achieves the highest $F_{0.5}$ score and recall rates in all the three annotation files.

There is still plenty of room for improvement and future investigation. Due to the time constraint, we only used a simple 5-gram model for correcting spelling errors. A more sophisticated model such as neural network would perform better.

There are also techniques that we would like to try to improve the effectiveness of the 2nd stage (e.g., data augmentation). Finally, grammatical error correction is only a small initial step into advancing language learning through AI. The current solutions do not handle semantic issues well. This certainly is a challenging research direction that has great potential to change many aspects of language learning. Our goal is to build comprehensive products that could make learning more effective.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Brockett, C., Dolan, W.B., Gamon, M.: Correcting ESL errors using phrasal SMT techniques. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pp. 249–256. Association for Computational Linguistics (2006)
3. Chollampatt, S., Ng, H.T.: Connecting the dots: Towards human-level grammatical error correction. In: Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, pp. 327–333 (2017)
4. Chollampatt, S., Ng, H.T.: A multilayer convolutional encoder-decoder neural network for grammatical error correction. arXiv preprint arXiv:1801.08831 (2018)
5. Dahlmeier, D., Ng, H.T.: Better evaluation for grammatical error correction. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2012, pp. 568–572. Association for Computational Linguistics, Stroudsburg (2012). http://dl.acm.org/citation.cfm?id=2382029.2382118
6. Gehring, J., Auli, M., Grangier, D., Dauphin, Y.N.: A convolutional encoder model for neural machine translation. arXiv preprint arXiv:1611.02344 (2016)
7. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. arXiv preprint arXiv:1705.03122 (2017)
8. Ji, J., Wang, Q., Toutanova, K., Gong, Y., Truong, S., Gao, J.: A nested attention neural hybrid model for grammatical error correction. arXiv preprint arXiv:1707.02026 (2017)
9. Liu, C.L., Lai, M.H., Tien, K.W., Chuang, Y.H., Wu, S.H., Lee, C.Y.: Visually and phonologically similar characters in incorrect chinese words: analyses, identification, and applications. ACM Trans. Asian Lang. Inf. Process. (TALIP) **10**(2), 10 (2011)
10. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025 (2015)
11. Rozovskaya, A., Chang, K.W., Sammons, M., Roth, D., Habash, N.: The Illinois-Columbia system in the CoNLL-2014 shared task. In: Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task, pp. 34–42 (2014)
12. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909 (2015)
13. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 6000–6010 (2017)

14. Wu, S.H., Liu, C.L., Lee, L.H.: Chinese spelling check evaluation at SIGHAN bake-off 2013. In: Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing, pp. 35–42 (2013)
15. Xie, P., et al.: 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, pp. 41–46 (2017)
16. Yuan, Z., Briscoe, T.: Grammatical error correction using neural machine translation. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 380–386 (2016)
17. Zheng, B., Che, W., Guo, J., Liu, T.: 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), pp. 49–56 (2016)

NLP Applications