Chinese Grammatical Error Diagnosis with Long Short-Term Memory Networks

Bo Zheng, Wanxiang Che*, Jiang Guo, Ting Liu
†Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology, China
{bzheng, car, jguo, tliu}@ir.hit.edu.cn

Abstract

Grammatical error diagnosis is an important task in natural language processing. This paper introduces our Chinese Grammatical Error Diagnosis (CGED) system in the NLP-TEA-3 shared task for CGED. The CGED system can diagnose four types of grammatical errors which are redundant words (R), missing words (M), bad word selection (S) and disordered words (W). We treat the CGED task as a sequence labeling task and describe three models, including a CRF-based model, an LSTM-based model and an ensemble model using *stacking*. We also show in details how we build and train the models. Evaluation includes three levels, which are detection level, identification level and position level. On the CGED-HSK dataset of NLP-TEA-3 shared task, our system presents the best F1-scores in all the three levels and also the best recall in the last two levels.

1 Introduction

Chinese has been considered as one of the most difficult languages in the world. Unlike English, Chinese has no verb tenses and pluralities, and there usually exist various ways to express the same meaning in Chinese. Consequently, it is common for non-native speakers of Chinese to make grammatical errors of various types in their writings. The goal of Chinese Grammatical Error Diagnosis (CGED) is to build a system that can automatically diagnose errors in Chinese sentences. Evaluation is carried out in three levels, based on the detection of error occurrences in a sentence, as well as their types and positions.

In this work, we formalize the CGED task as a sequence labeling problem, which assigns each Chinese character in a target sentence with a tag indicating both the error type (R, M, S, W) and position (Beginning, Inside). Therefore, the CGED task can be readily solved with a typical conditional random fields (CRF) model (Lafferty et al., 2001).

However, the main challenge for CGED is that the detection of errors usually requires long-term dependencies. For example, in Table 1, the grammatical error at “表示(represent)” may not be detected until the last word “损害(damage)” shows up. Traditional models with features extracted from a limited context window may not be able to handle these situations.

Neural network-based models have been extensively used in natural language processing (NLP) during recent years, due to their strong capability of automatical feature learning. In particular, the long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) based recurrent neural networks (RNN) have been proved to be highly effective in various applications that involves sequence modeling, such as language modeling, named entity recognition (Lample et al., 2016) and parsing (Vinyals et al., 2015), etc. Therefore, in this paper, we propose to use LSTM-based RNNs to solve the CGED problem. In order to leverage both the merits of CRF models and LSTM models, we further present an ensemble model using *Stacking* (Nivre and McDonald, 2008). Evaluations on the NLP-TEA-3 shared task for CGED show that our models achieve the best F1-scores in all levels and the best recall in two levels.

The rest of the paper is organized as follows: Section 2 gives the definition of the CEGD task. Section 3 describes how LSTM network is used to predict errors and what other works we have done. Section 4

*Email correspondence.

Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications,
pages 49–56, Osaka, Japan, December 12 2016.

shows the evaluation results. Section 5 gives some related works. Section 6 gives conclusion and future work of this paper.

2 Task Definition

The shared task of CGED in NLP-TEA-3 is defined as follows: given a Chinese sentence, a CGED system is expected to diagnose four types of grammatical errors, including *redundant words* (R), *missing words* (M), *bad word selection* (S) and *disorder words* (W). Once an error is found, the system should be able to recognize its beginning and ending positions.

Table 1 and Table 2 show two examples in the dataset:

这$_{1}$ 种$_{2}$ 材$_{3}$ 料$_{4}$ **表**$_{5}$ **示**$_{6}$ 吸$_{7}$ 烟$_{8}$ 引$_{9}$ 起$_{10}$ 了$_{11}$ **人**$_{12}$ **们**$_{13}$ 多$_{14}$ 么$_{15}$ 大$_{16}$ 的$_{17}$ 损$_{18}$ 害$_{19}$ 。$_{20}$		
Error Interval	5, 6	12, 13
Error Type	S	R
Correction	这种材料**表明**吸烟引起了多么大的损害。 This material shows how much harm smoking causes.	

Table 1: Two errors are found in the sentence above, one is bad word selection (S) error from position 5 to 6, the other one is redundant words (R) error from position 12 to 13.

但$_{1}$ 是$_{2}$ 文$_{3}$ 章$_{4}$ 中$_{5}$ 的$_{6}$ 妻$_{7}$ 子$_{8}$ **是**$_{9}$ **还**$_{10}$ 有$_{11}$ 意$_{12}$ 识$_{13}$ 的$_{14}$ ，$_{15}$ 她$_{16}$ 还$_{17}$ 有$_{18}$ 活$_{19}$ **的**$_{20}$ 意$_{21}$ 义$_{22}$ 。$_{23}$		
Error Interval	9, 10	20, 20
Error Type	W	M
Correction	但是文章中的妻子**还是**有意识的。她还有活**着**的意义。 But the wife in the passage is still conscious, she still has a meaning to live.	

Table 2: Two errors are found in the sentence above, one is disordered words (W) error from position 9 to 10, the other one is missing words (M) error in position 20.

3 Methodology

In this work, we treat the CGED task as a sequence labeling problem. Specifically, given a sentence $\boldsymbol{x}$, our model generates a corresponding label sequence $\boldsymbol{y}$. Each label in $\boldsymbol{y}$ is a token from a specific tag set. Here we have tag 'O' indicating correct characters, 'B-X' indicating the beginning positions for errors of type 'X' and 'I-X' as middle and ending positions for errors of type 'X'.

We first examine the traditional CRF model and use symbolical represented features. Then we propose our LSTM-based model that use distributed feature representations. At last, we present an ensemble model that combines the two models using *Stacking*.

In this section, we will first introduce how we prepare the data, and then describe the three models we used in this task.

3.1 Data Preparation

Since the CGED task involves identifying the error boundaries, segmenting a sentence into words will bring a lot of misalignments between the words and the endpoints of a corresponding error interval. An example of misalignment is shown in Table 3. Therefore, we decided to solve the problem at character level. Other than the misaligned interval problem, there are many error intervals of different types which may overlap with others. One way to avoid this overlapping problem is to deal with the four types of errors separately. However, we think the four types of errors may have mutual effects on each other, so we pre-processed the training data so that we can keep as many errors as possible by deleting the least numbers of overlapped error intervals. We finally deleted a small part of error intervals which is acceptable. An example of overlapping problem is shown in Table 4.

如$_1$ 果$_2$ 你$_3$ 是$_4$ 青$_5$ 少$_6$ 年$_7$ 你$_8$ 多$_9$ 想$_{10}$ 自$_{11}$ 己$_{12}$ 的$_{13}$ 未$_{14}$ 来$_{15}$ ；$_{16}$ 那$_{17}$ 你$_{18}$ **可**$_{19}$ 以$_{20}$ **禁**烟$_{21-22}$ 了$_{23}$ 。$_{24}$		
Error Interval	19, 19	21, 21
Error Type	M	S
Correction	如果你是青少年你要多想想自己的未来；那你**就**可以**戒**烟了。 If you are a teenager you should think about your future so you can quit smoking.	

Table 3: An misalignment example. The two characters “禁(forbid)” and “烟(cigarette)” would be one word after segmenting the sentence into words, which would cause a misalignment problem because only the character “禁(forbid)”.

每$_1$ 年$_2$ 暑$_3$ 假$_4$ **是**$_5$ **我**$_6$ **们**$_7$ **运**$_8$ **动**$_9$ **员**$_{10}$ **来**$_{11}$ **说**$_{12}$ 很$_{13}$ 苦$_{14}$ 的$_{15}$ 时$_{16}$ 候$_{17}$ 。$_{18}$		
Error Interval	5, 12	6, 6
Error Type	W	M
Correction	每年暑假**对我们运动员来说是**很苦的时候。 Every summer is hard for us athletes.	

Table 4: An overlapping example. The two error intervals are overlapped. In this situation, we will delete the least number of intervals to eliminate the problem.

One kind of features that may be useful in this task is the Part-of-speech (POS) of words. Table 5 shows a snapshot of training data after the pre-processing. Note that our task is being solved at the character level, so we split the POS tag of a word to character level by attaching position indicators (‘B-’ and ‘I-’) to the POS of a word.

Character	POS	Label
像	B-p	O
我	B-r	B-W
对	B-p	I-W
不	B-d	B-M
吸	B-n	O
烟	I-n	O
者	I-n	O
来	B-u	O
说	I-u	O

Table 5: A snapshot of our training data after the pre-processing

In the training phase, a sentence is first segmented into terms. Each term is consisted with a character, a corresponding POS tag and an error type tag.

3.2 CRF-Based Model

CRF has been successfully used in various natural language processing applications, especially sequence labeling tasks. Formally, the model can be defined as:

$$P\left(\boldsymbol{y} \mid \boldsymbol{x}\right) = \frac{1}{Z\left(\boldsymbol{x}\right)} exp\left(\Sigma_k \lambda_k f_k\right) \tag{1}$$

where $Z(\boldsymbol{x})$ is the normalization factor, f_k is a set of features, λ_k is the corresponding weight. In this task, $\boldsymbol{x}$ is the input sentence, and $\boldsymbol{y}$ is the corresponding error type label. The feature templates are defined in Table 6. We use stochastic gradient descent (SGD) for training, with L2 regularization to prevent overfitting.

Feature templates
00: $ch_{i+k}, -2 \leq k \leq 2$
01: $ch_{i+k} \circ ch_{i+k+1}, -1 \leq k \leq 0$
02: $pos_{i+k}, -2 \leq k \leq 2$
03: $pos_{i+k} \circ pos_{i+k+1}, -2 \leq k \leq 1$
04: $pos_{i+k} \circ pos_{i+k+1} \circ pos_{i+k+2}, -2 \leq k \leq 0$
Unigram Features
$y_i \circ 00 - 04$
Bigram Features
$y_{i-1} \circ y_i$

Table 6: feature templates of CRF-based model. ch_i refers to the $i_t h$ character, pos_i refers to the POS of $i_t h$ character, y_i refers to the output tag of $i_t h$ character.

3.3 LSTM-Based Model

LSTM network is a variant of recurrent neural network (RNN) and have better ability to capture long-term dependencies. At each time step t, LSTM networks read a current input vector x_t and the hidden state of the previous time step h_{t-1}, and use them to compute a new hidden state h_t.

Vanilla RNNs (Pascanu et al., 2013) typically suffer from the gradient vanishing problem while LSTM networks solve it with an extra memory "cell" (c_t). Specifically, LSTM networks are controlled by three kinds of gates, each gate consists of a sigmoid neural net layer and a point-wise multiplication operation. The three gates are input gate, forget gate and output gate. The input gate controls what proportion of the current input to pass into the memory cell (i_t), and the forget gate controls what proportion of the previous memory cell to "forget" (f_t). When here comes the input x_t, the memory cell is updated as follows:

$$i_t = \sigma\left(W_{ix}x_t + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i\right) \tag{2}$$

$$f_t = \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + W_{fc}c_{t-1} + b_f\right) \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh\left(W_{cx}x_t + W_{ch}h_{t-1} + b_c\right) \tag{4}$$

where σ represents point-wise logistic sigmoid function, and $\odot$ is the point-wise Hadamard product.

The output gate (o_t) controls the hidden state h_t at each time step, and they are computed as follows:

$$o_t = \sigma\left(W_{ox}x_t + W_{oh}h_{t-1} + W_{oc}c_t + b_o\right) \tag{5}$$

$$h_t = o_t \odot tanh\left(c_t\right) \tag{6}$$

We use the hidden state h_t to calculate the output label at each time step at last. The architecture of our bidirectional LSTM-based model is illustrated in Figure 1. We used the concatenation of character embeddings and bigram embeddings as lexicalized input features at each position.

The character embeddings are initialized randomly. To obtain the bigram embeddings, we first convert the original character sequence to a bigram sequence. For example, the bigram sequence of sentence "我是中国人" will be ["我是", "是中", "中国", "国人"]. Then we can train bigram embeddings readily using word2vec (Mikolov et al., 2013) on the resulting bigram sequences. In addition, we also used the POS of words as a discrete feature to improve the performance of our model.

We give the comparison between LSTM-based model with unigram feature and LSTM-based model with bigram and also unigram feature in next section. We also adjusted the model by tuning the value of the input dimension of LSTM and the dimension of bigram embeddings.

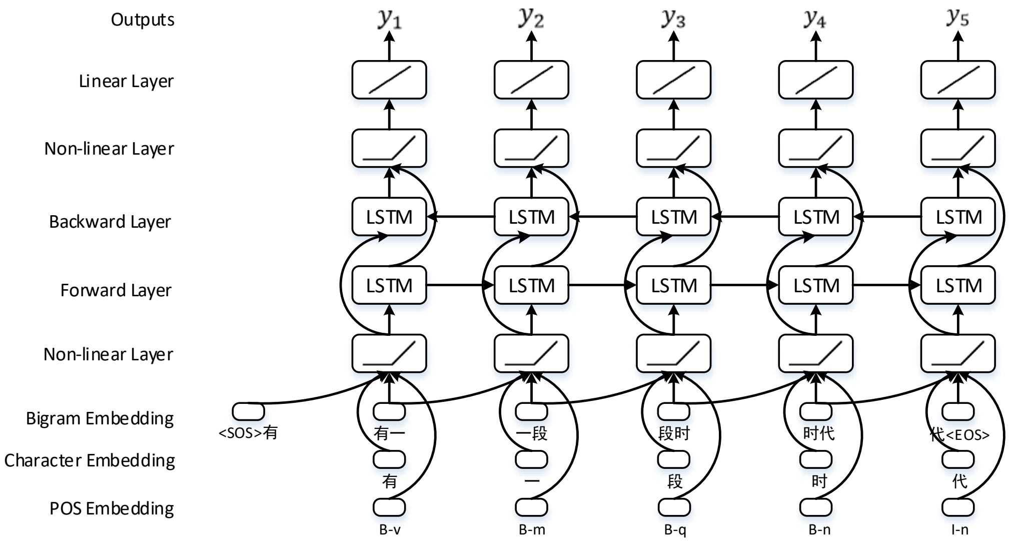

Figure 1: An illustration of the LSTM-based model. The concatenated character embedding, POS embedding and two bigram embeddings are used as the input of the neural network.

3.4 Stacking Model

After preliminary experiments using the two models above, we found that the LSTM-based model has a high recall rate and the CRF-based model has a high precision rate. To take advantage of both models, we further present an ensemble model using *stacking* (Nivre and McDonald, 2008).

We conducted 10-fold cross-validation on the training dataset and obtained the tagging results automatically. Then we put the result of the CRF-based model as a discrete feature to the LSTM layer by adding an additional feature to the input layer of LSTM. We expect that by combining the two models together, the LSTM-based model can achieve higher precision rate.

Results show that after combining the two models, the recall of the LSTM-based model increases, but unfortunately, the precision decreases. The reason could be the results of CRF-based model help LSTM find errors which LSTM-based model wasn't able to find. We will discuss it specifically in the next section.

4 Experiments

4.1 Data and Settings

We obtain the full dataset from the shared task CGED-HSK of NLP-TEA-3 for training and validation, of which 16,142 sentences are used for training and the rest 4000 sentences for validation. The ratio of training dataset size to validation dataset size is about 4:1. Table 7 shows the data distribution in the CGED-HSK training data. In addition, we use the Chinese Gigawords to get the pretrained bigram embeddings. For the CRF-based model, we adopt the CRFsuite toolkit (Okazaki, 2007).

The criterias for evaluation include:

(1) **Detection level**: this is a binary classification of a given sentence, i.e. correct or incorrect should be completely identical with the gold standard. All error types will be regarded as incorrect.

(2) **Identification level**: this could be considered as a multi-class categorization problem. In addition to correct instances, all error types should be clearly identified.

(3) **Position level**: besides identifying the error types, this level also judges the positions of erroneous range. That is, the system results should be perfectly identical with the quadruples of gold standard.

Type	Train	Validation
Redundant	4374	1074
Missing	5250	1203
Selection	8533	2177
Disorder	1196	291
Correct	8086	2002

Table 7: Data statistics.

4.2 Experiment Results

We first conduct experiments with the CRF-based and the LSTM-based model. After that, we examine the effect of *Stacking* by taking the output of the CRF model as features of the LSTM model.

4.2.1 Results on Validation Dataset

We use the validation dataset to select the best hyper-parameters in both the CRF-based model and the LSTM-based model. Table 8 shows the results. As we can see, the LSTM-based model (LSTM (U+B)) has better Recall and F1-score than the CRF-based model, but lower in precision. Besides, the bigram embeddings has a very significant impact on the LSTM-based model.

Model	Detection Level			Identification Level			Position Level		
	P	R	F1	P	R	F1	P	R	F1
CRF	**0.7500**	0.2282	0.3500	**0.7154**	0.1663	0.2699	**0.6507**	0.1296	0.2162
LSTM (U)	0.5188	0.2908	0.3727	0.4458	0.1925	0.2689	0.3329	0.1197	0.1761
LSTM (U+B)	0.6526	0.3629	0.4664	0.5625	0.2484	0.3446	0.4115	**0.1587**	**0.2290**
Stacking	0.6344	**0.3909**	**0.4837**	0.5401	**0.2565**	**0.3478**	0.3797	0.1513	0.2164

Table 8: Results on Validation Dataset. ‘U’ in the bracket after LSTM refers to using unigram of characters and ‘B’ refers to using bigram of characters.

4.2.2 Results on Evaluation Dataset

When testing on the final evaluation dataset, we merged our training dataset and validation dataset, and retrain our models. Table 9 shows the results of our three submissions.

Submission	Detection Level			Identification Level			Position Level		
	P	R	F1	P	R	F1	P	R	F1
HIT-Run1	**0.6111**	0.712	0.6577	0.5146	0.5219	0.5182	**0.4034**	0.3691	**0.3855**
HIT-Run2	0.6108	0.7099	0.6566	**0.5224**	0.5094	0.5185	0.397	0.3483	0.3711
HIT-Run3	0.6071	**0.7296**	**0.6628**	0.5002	**0.5447**	**0.5215**	0.3695	**0.3697**	0.3696

Table 9: Results on Evaluation Dataset.

The three models we submitted includes the LSTM-based model (HIT-Run1), Stacking model (HIT-Run2) and LSTM-based model with some post-process (HIT-Run3). The post-process mainly includes changing the ‘I-X’ errors without a ‘B-X’ error before it into a single ‘B-X’ error. This increases the recall rate on three levels but slightly decreases the precision.

The stacking model increases the precision on the identification level while it reduces overall performance. The reason could be that our CRF-based model doesn't have good feature templates or the inherent properties of the task.

Our system presents the best F1 scores in all three levels and also the best recall rates in the last two levels on evaluation dataset. However, the results of this task are not that credible because there are many ways to correct a wrong Chinese sentence. For example, deleting some redundant words may replace errors of missing words.

5 Related Works

In NLP-TEA-1 (Yu et al., 2014) shared task for CGED, there were four types of errors, which were the same as the task of this year. The evaluation was only based on detection of error occurrence, disregarding the recognization of boundaries. In NLP-TEA-2 (Lee et al., 2015) shared task for CGED, the participating systems are required to not only detect the errors but also locate them. Evaluations were focused on traditional Chinese texts rather than simplified Chinese, and one sentence includes one error at most in last two years.

There have been several studies focused on Chinese grammatical error detection. Wu et al. (2010) proposed a method using both Relative Position Language Model and Parse Template Language Model to detect Chinese errors written by US learner. Yu and Chen (2012) proposed a classifier to detect word-ordering errors in Chinese sentences from the HSK dynamic composition corpus. Lee et al. (2013) proposed linguistic rule based Chinese error detection for second language learning. Lee et al. (2014) developed a sentence judgment system using both rule-based and n-gram statistical methods to detect grammatical errors in Chinese sentences. However, all of these previous works used hand-crafted features which may be incomplete and cause the loss of some important information. Comparatively, our neural network approaches have strong capability of automatical feature learning and are completely data-driven.

6 Conclusion

This paper describes our system in the NLP-TEA-3 task for CGED-HSK. We explored the CRF-based model, the LSTM-based model and further used stacking to combine the two models. We achieved highest F1 scores in all three levels and highest Recall rates in identification level and position level.

In our future work, we plan to try more methods such as bagging or adding more features to the CRF-based model. Since Chinese grammar is flexible and irregular, it is difficult to judge the credibility of these results on testing data. In our future work, we will try more models and find better ways to judge the result if possible.

Acknowledgements

This work was supported by the National Key Basic Research Program of China via grant 2014CB340503 and the National Natural Science Foundation of China (NSFC) via grant 61632011 and 61370164.

References

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, volume 1, pages 282–289.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *NAACL*, pages 260–270, San Diego, California, June.

Lung-Hao Lee, Li-Ping Chang, Kuei-Ching Lee, Yuen-Hsien Tseng, and Hsin-Hsi Chen. 2013. Linguistic rules based chinese error detection for second language learning. In *Work-in-Progress Poster Proceedings of the 21st International Conference on Computers in Education (ICCE-13)*, pages 27–29.

Lung-Hao Lee, Liang-Chih Yu, Kuei-Ching Lee, Yuen-Hsien Tseng, Li-Ping Chang, and Hsin-Hsi Chen. 2014. A sentence judgment system for grammatical error detection. In *COLING (Demos)*, pages 67–70.

Lung-Hao Lee, Liang-Chih Yu, Li-Ping Chang, et al. 2015. Overview of the nlp-tea 2015 shared task for chinese grammatical error diagnosis. *ACL-IJCNLP 2015*, page 1.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *International Conference on Learning Representations (ICLR) Workshop*.

Joakim Nivre and Ryan T McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *ACL*, pages 950–958.

Naoaki Okazaki. 2007. Crfsuite: a fast implementation of conditional random fields (crfs).

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. *ICML (3)*, 28:1310–1318.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pages 2773–2781.

Chung-Hsien Wu, Chao-Hong Liu, Matthew Harris, and Liang-Chih Yu. 2010. Sentence correction incorporating relative position and parse template language models. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1170–1181.

Chi-Hsin Yu and Hsin-Hsi Chen. 2012. Detecting word ordering errors in chinese sentences for learning chinese as a foreign language. In *COLING*, pages 3003–3018.

Liang-Chih Yu, Lung-Hao Lee, and Li-Ping 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 (NLPTEA'14), Nara, Japan*, pages 42–47.