Bi-LSTM Neural Networks for Chinese Grammatical Error Diagnosis

Shen Huang and **Houfeng Wang**[*]
Key Laboratory of Computational Linguistics, Ministry of Education
School of Electronics Engineering and Computer Science, Peking University
Beijing, P.R.China, 100871
{huangshenno1,wanghf}@pku.edu.cn

Abstract

Grammatical Error Diagnosis for Chinese has always been a challenge for both foreign learners and NLP researchers, for the variousity of grammar and the flexibility of expression. In this paper, we present a model based on Bidirectional Long Short-Term Memory(Bi-LSTM) neural networks, which treats the task as a sequence labeling problem, so as to detect Chinese grammatical errors, to identify the error types and to locate the error positions. In the corpora of this year's shared task, there can be multiple errors in a single offset of a sentence, to address which, we simutaneously train three Bi-LSTM models sharing word embeddings which label Missing, Redundant and Selection errors respectively. We regard word ordering error as a special kind of word selection error which is longer during training phase, and then separate them by length during testing phase. In NLP-TEA 3 shared task for Chinese Grammatical Error Diagnosis(CGED), Our system achieved relatively high F1 for all the three levels in the traditional Chinese track and for the detection level in the Simpified Chinese track.

1 Introduction

As China plays a more and more important role of the world, learning Chinese as a foreign language is becoming a growing trend, which brings opportunities as well as challenges. Due to the variousity of grammar and the flexibility of expression, Chinese Grammatical Error Dignosis(CGED) poses a serious challenge to both foreign learners and NLP researchers. Unlike inflectional languages such as English which follows grammatical rules strictly(i.e. subject-verb agreement, strict tenses and voices), Chinese, as an isolated language, has no morphological changes. Various characters are arranged in a sentence to represent meanings as well as the tense and the voice. These features make it easy for beginners to make mistakes in speaking or writing. Thus it is necessary to build an automatic grammatical error detection system to help them learn Chinese better and faster.

In NLP-TEA 3 shared task for Chinese Grammatical Error Diagnosis(CGED), four types of errors are defined: **'M'** for missing word error, **'R'** for redundant errors, **'S'** for word selection error and **'W'** for word ordering error. Some typical examples of the errors are shown in Table 1. Different from the two previous editions for the CGED shared task, each input sentence contains at least one of defined error types. What's more, there can be multiple errors in a single offset of a sentence, which means we can no longer treat it a simple multi-class classification problem. As a result of that, we cannot simply rely on some existing error detection systems but can only seek for a new solution.

[*] Corresponding author

Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications,
pages 148–154, Osaka, Japan, December 12 2016.

Error Type	**Error Sentence**	**Correct Sentence**
M(Missing word)	我一完了考試，就回家。	我一考完了考試，就回家。
R(Redundant word)	爸爸是軍人，也很想他的太太。	爸爸是軍人，很想他的太太。
S(Selection error)	一般人不可以隨意**出國外**。	一般人不可以隨意**出國**。
W(Word ordering error)	它是讓我哭其中之一**的電影**。	它是讓我哭**的電影**其中之一。

Table 1: Some typical examples for grammatical errors in Chinese

In order to address the problem, we regard it as a sequence multi-labeling problem and split it into multiple sequence labeling problems which only label 0 or 1. To avoid feature engineering, for each error type except **’W’**, we trained a Bi-LSTM based neural network model, sharing word embeddings and POS tag embeddings. We treat the word ordering error as a special kind of word selection error. They are trained together and separated during the testing phase. Experiments show that together training is better than separate training. More details are described in the rest of the paper.

This paper is organized as follows: Section 2 briefly introduces some previous work in this area. Section 3 describes the Bi-LSTM neural network model we proposed for this task. Section 4 demonstrates the data analysis and some interesting findings. Section 5 shows the data analysis and the evaluation results. Section 6 concludes this paper and illustrates the future work.

2 Related Work

Grammatical error detection and correction has been studied with considerable efforts in the NLP community. Compared to Chinese, the language of English attracted more attention from the researchers, especially during the CoNLL2013 and 2014 shared task (Ng et al., 2013; Ng et al., 2014). However, different from English which has various language materials and annotated corpura, the grammatical error correction related resource for Chinese is far from enough. We are glad to see the shared tasks on CGED (Yu et al., 2014; Lee et al., 2015) in last two years.

There were some previous related work for Chinese grammatical error detection or correction. Wu et al. (2010) proposed two types of language models to detect the error types of word order, omission and redundant, corresponding to three of the types in the shared task. Experimental results showed syntactic features, web corpus features and perturbation features are useful for word ordering error detection (Yu and Chen, 2012). A set of handcrafted linguistic rules with syntactic information are used to detect errors occurred in Chinese sentences (Lee et al., 2013), which are shown to achieve good results. Lee et al. (2014) introduced a sentence level judgment system which integrated several predefined rules and N-gram based statistical features.

Our submission was an exploration to a neural network model in CGED which didn’t need any feature selection efforts. As a model well known for its good maintainance of both preceding and succeeding information, Bi-LSTM came to be the first choice.

3 Bi-LSTM Neural Network based Model

We regard CGED task as a word-based sequence multi-labeling problem, by labeling each word zero or more tags from $\{\mathbf{M}, \mathbf{R}, \mathbf{S}, \mathbf{W}\}$. For some reason described in Section 4, we treat word ordering error as a special kind of word selection error, as a result of which, we need to deal with only three kinds of error types during the training phase.

As different error types are relatively independent from each other from a single word’s perspective, we can train three sequence labeling models to judge whether this kind of error ocurrs in a certain position for each error type respectively. As shown in Figure 1, the architecture of a Bi-LSTM neural network model for CGED for a single error type can be characterized by the following three specialized layers: (1) Embedding layer (2) Encoding layer (3) Decoding layer.

As the errors are judged on words instead of characters, we first segment the input sentence into individual words using the CKIP Chinese Segmentation System[1] provided by Taiwan Academia Sinica and

[1] http://ckipsvr.iis.sinica.edu.tw/

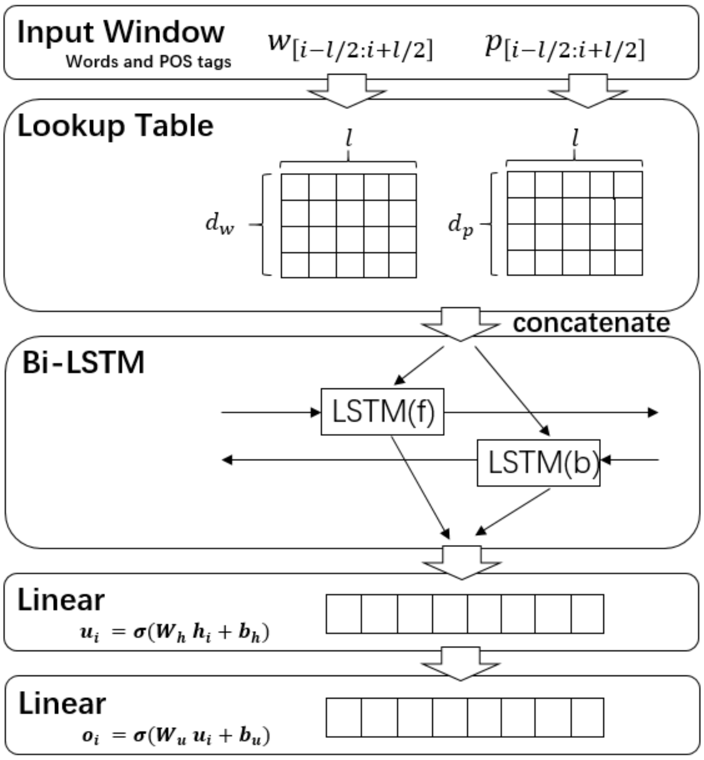

Figure 1: Architecture of Bi-LSTM neural network model for CGED for a single error type

get the simplified Part-of-Speech tag for each corresponding segmented word, which is our preprocess for the system. The words and POS tags are then embeded in the embedding layer.

The most common tagging approach is the window approach. The window approach assumes that the tag of a word largely depends on its neighboring words. For each word w_i in a given input sentence $w_{[1:n]}$, the context words $w_{[i-l/2:i+l/2]}$ and the context POS tags $p_{[i-l/2:i+l/2]}$ are chosen to be fed into the networks, where l is the context window size and usually $l = 5$ or $l = 7$. Here we set $l = 7$ in our experiments. The words and POS tags exceeding the sentence boundaries are mapped to two special symbols, "<BOS>" and "<EOS>", representing *Beginning of a Sentence* and *End of a Sentence* respectively. And the out-of-character-set words will be replaced with a symbol "<UNK>" which represents *Unknown*.

Given a word set V of size $|V|$, the embedding layer will map each word $w \in V$ into a d_w-dimensional embedding space as $Embed_w(w) \in \mathbb{R}^{d_w}$ by a lookup table $\mathbf{M_w} \in \mathbb{R}^{d_w \times |V|}$. In the same way, we can map each POS tag $p \in P$ into a d_p-dimensional embedding space as $Embed_p(p) \in \mathbb{R}^{d_p}$ by a lookup table $\mathbf{M_p} \in \mathbb{R}^{d_p \times |P|}$, where P is the POS tag set whose size if $|P|$. The embeddings of the context words $w_{[i-l/2:i+l/2]}$ and The embeddings of the context POS tags $p_{[i-l/2:i+l/2]}$ are then concatenated into a single vector $x_i \in \mathbb{R}^{H_1}$, where $H_1 = l \times (d_w + d_p)$. Then this vector x_i is fed into the encoding layer.

The encoding layer is a Bi-LSTM layer followed by a full-connection layer, which can be simply expressed by the following:

$$h_i = BiLSTM_\theta(x_i) \tag{1}$$

$$u_i = \sigma(W_h h_i + b_h) \tag{2}$$

where θ is the inner parameters of the Bi-LSTM layer and σ is the logistc sigmoid function.

The Long Short-Term Memory cell(Hochreiter and Schmidhuber, 1997) is a special kind of the RNN cell which replaces the hidden layer updates by purpose-built memory cells. As a result, they can utilize long range dependencies and realize the function just like memory. A single LSTM cell is illustrates in Figure 2.

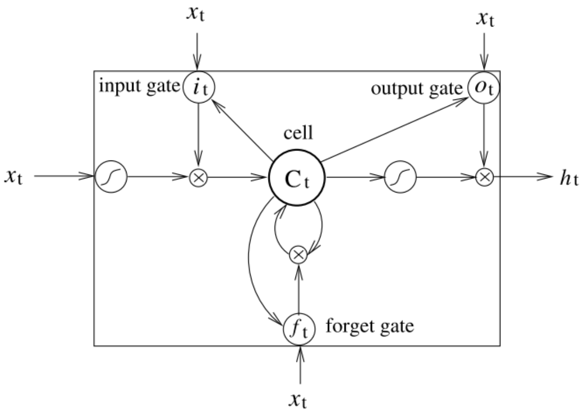

Figure 2: A LSTM cell

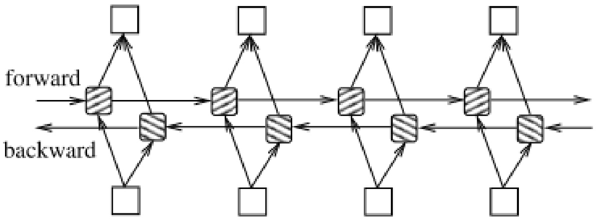

Figure 3: The structure of a Bi-LSTM layer

The LSTM cell is implemented as the following:

$$\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}C_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}C_{t-1} + b_f) \\
c_t &= f_t c_{t-1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}C_t + b_o) \\
h_t &= o_t tanh(c_t)
\end{aligned} \tag{3}$$

where σ is the logistic sigmoid function, and i, f, o and c are the input gate, forget gate, output gate and the cell, all of which are the same size as the hidden output h. The subscripts of the weight matrix describe the meaning as the name suggests. For instance, W_{xi} is the input gate weight matrix for input x.

A single LSTM forward layer can only utilize the previous information, which is not enough for grammatical error detection, where sometimes the error can only be inferred from the following words. Therefore, a bidirectional LSTM layer is proposed (Graves, 2013), which can be regarded as a simple stack of a forward LSTM layer and a backward LSTM layer. The structure of a Bi-LSTM layer is shown in Figure 3.

The output of the encoding layer is then fed into a decoding layer, which is another full-connection layer with 1 output size. The output layer is implemented as the following:

$$o_i = \sigma(W_u u_i + b_u) \tag{4}$$

$$f(w_i) = o_i > 0.5 \tag{5}$$

where σ is still the logistic sigmoid function and $f(w_i)$ indicates whether there is an error of this type on the word w_i or not.

As there are many more non-errors than errors in a sentence from a word perspective, the model always tends to label 0, which means correct for the word, if without any balance. Thus we assigned weights for the loss function, in order to rebalance the correct and incorrect labels. The loss function without regularization is calculated as follows:

$$loss = y_i * -log(f(w_i)) * W_{pos} + (1 - y_i) * -log(1 - f(w_i)) \tag{6}$$

where W_{pos} is the coefficient on the positive examples.

We can decide if there are errors from $\{$**M**, **R**, **S&W**$\}$ through the model described above. Then we separate the **'S'** and **'W'** tags according to the successive word length of the error during the testing phase. If the length is 1 the error is a word selection error, otherwise it is a word ordering error if the length is greater than 1.

4 Data Preparation and Analysis

4.1 Datasets

In the TOCFL track, the data we used for training includes training and testing data from NLP-TEA 1 (Yu et al., 2014), training data from NLP-TEA 2 (Lee et al., 2015), and training data from NLP-TEA 3. We used the testing data from NLP-TEA 2 for validation.

In the HSK track, despite of the training set provided by the organizers, we simplified the training data from TOCFL track as supplements. However, the simplified data from TOCFL track seem to be no use to the evaluation results.

Table 2 shows the statistics of our training sets.

	NLP-TEA 1	**NLP-TEA 2**	**NLP-TEA 3 TOCFL**	**NLP-TEA 3 HSK**
number of sentences	7389	2205	10693	10072
total errors	7389	2205	24831	24784
Missing words	2932	620	9078	6619
Redundant words	2399	430	4472	5532
Word selection errors	1087	849	9897	10942
Word ordering errors	971	306	1384	1691

Table 2: Statistics of training sets

Due to the limitation of time and resource, the word embeddings and POS tag embeddings we used are all random initialized.

4.2 Word selection error and word ordering error

Take NLP-TEA 3 TOCFL dataset as an example, as there are 1384 word ordering errors in the training set, which takes only 5.5% in all 24831 errors. It is difficult to train this kind of errors without rebalance or resampling. Thus we came up with a new method, by treating word ordering error as a special kind of word selection error. Suprisingly, in the training set, after word segmentation, most word selection errors are within one word and all word ordering errors are longer than one word, we can easily separate them by the successive error length.

5 Experiments

In the formal run of NLP-TEA 3 CGED shared task, there are 5 teams submitting 15 runs in total for the TOCFL dataset track and 8 teams submitting 21 runs in total for the HSK dataset track. Our system achieved relatively high F1 for all the three levels in the traditional Chinese track and for the detection level in the Simpified Chinese track. Since our evaluation results for HSK dataset are not good, here we only display the evaluation results compared with the average values for TOCFL dataset. The performance evaluations in detection level, identification level and position level are shown as follows:

	False Positive Rate	**Accuracy**	**Precision**	**Recall**	**F1**
PKU-Run1	0.2284	0.521	**0.5739**	0.2871	0.3828
PKU-Run2	**0.7205**	0.5258	0.5292	**0.7556**	**0.6224**
PKU-Run3	0.525	0.5349	0.5467	0.5907	0.5678
Average of all 15 runs	0.4812	**0.5442**	0.5701	0.5680	0.5456

Table 3: Performance evaluation in detection level

	Accuracy	**Precision**	**Recall**	**F1**
PKU-Run1	**0.4575**	**0.3418**	0.1173	0.1747
PKU-Run2	0.3242	0.2792	**0.3712**	**0.3187**
PKU-Run3	0.3705	0.2729	0.2192	0.2431
Average of all 15 runs	0.3912	0.3265	0.2732	0.2716

Table 4: Performance evaluation in identification level

6 Conclusion and Future work

In this paper, we present a Bi-LSTM neural network based model to predict the possible grammatical errors for Chinese, which needs no feature engineering and provides reasonable evaluation results in the NLP-TEA 3 CGED shared task. Different from most previous work, we didn't use any external corpus or rule-based inductions. Due to the limitation of time and resource, we didn't test our system under various experiment environments. More neural network architectures and more features can be tried. There is still space for further development.

Acknowledgements

We gratefully acknowledge the kind cooperation of Prof. Houfeng Wang in guidance of this work.

References

Ng, Hwee Tou and Wu, Siew Mei and Wu, Yuanbin and Hadiwinoto, Christian and Tetreault, Joel 2013. *The CoNLL-2013 Shared Task on Grammatical Error Correction*. Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task. Sofia, Bulgaria, 1-12.

Ng, Hwee Tou and Wu, Siew Mei and Briscoe, Ted and Hadiwinoto, Christian and Susanto, Raymond Hendy and Bryant, Christopher 2014. *The CoNLL-2014 Shared Task on Grammatical Error Correction*. In Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task. Baltimore, Maryland, 1-14.

Chung-Hsien Wu, Chao-Hong Liu, Harris Matthew 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 Proceedings of the 24th International Conference on Computational Linguistics (COLING'12), 3003-3017

	Accuracy	**Precision**	**Recall**	**F1**
PKU-Run1	**0.3844**	**0.0996**	0.0263	0.0416
PKU-Run2	0.1381	0.068	**0.0824**	**0.0745**
PKU-Run3	0.2331	0.0872	0.0651	**0.0745**
Average of all 15 runs	0.2402	0.0846	0.0460	0.0597

Table 5: Performance evaluation in position level

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 Proceedings of the 21st International Conference on Computers in Education (ICCE'13), 27-29,

Lung-Hao Lee, Liang-Chih Yu, Kuei-Ching Lee, Yuen-Hsien Tseng, Li-Ping Chang, Hsin-Hsi Chen 2014. *Sentence Judgment System For Grammatical Error Detection*. In Proceedings of the 25th International Conference on Computational Linguistics (COLING'14), 67-70

S. Hochreiter and J. Schmidhuber. 1997. *Long short-term memory*. Neural Computation, 9(8):1735-1780.

A. Graves, A. Mohamed, and G. Hinton. 2013. *Speech Recognition with Deep Recurrent Neural Networks*. arxiv.

Liang-Chih Yu, Lung-Hao Lee, and Li-Ping Chang 2014 *Overview of Grammatical Error Diagnosis for Learning Chinese as a Foreign Language*. Proceedings of the 1st Workshop on Natural Language Processing Techniques for Educational Applications, 42-47.

Lee, Lung-Hao and Yu, Liang-Chih and Chang, Li-Ping 2015 *Overview of the NLP-TEA 2015 Shared Task for Chinese Grammatical Error Diagnosis*. Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications, 1-6.