YNU-HPCC at IJCNLP-2017 Task 1: Chinese Grammatical Error Diagnosis Using a Bi-directional LSTM-CRF Model

Quanlei Liao, Jin Wang, Jinnan Yang and **Xuejie Zhang**
School of Information Science and Engineering
Yunnan University
Kunming, P.R. China
Contact:xjzhang@ynu.edu.cn

Abstract

Building a system to detect Chinese grammatical errors is a challenge for natural-language processing researchers. As Chinese learners are increasing, developing such a system can help them study Chinese more easily. This paper introduces a bi-directional long short-term memory (BiLSTM) - conditional random field (CRF) model to produce the sequences that indicate an error type for every position of a sentence, since we regard Chinese grammatical error diagnosis (CGED) as a sequence-labeling problem. Among the participants this year of CGED shard task, our model ranked third in the detection-level and identification-level results. In the position-level, our results ranked second among the participants.

1 Introduction

With China's rapid development, more and more foreign people have begun to learn Chinese. Writing is an important part of language learning, and grammar is the basis of writing. Traditional learning methods rely on artificial work to point out grammatical errors in an article. This requires more time and labor costs. Thus, it is quite practical to develop a system that can automatically correct the grammatical errors in an article. This is the aim of the CGED shared task.

In the shared task, Chinese-grammar errors are divided into four types: redundant words, word-selection errors, missing words, and incorrect word order (Lee et al., 2016). They are represented as uppercase letters "R", "S", "M", and "W", respectively. For each sentence, the task should first determine whether the sentence is correct. If the sentence is incorrect, it should indicate the specific error types and their locations.

In this paper, the CGED task is treated as a sequence labeling problem, which is a classic natural language processing problem. The traditional solutions are a series of statistical learning methods, including the hidden Markov model (HMM), the maximum-entropy Markov model (MEMM), and the conditional random field.

The HMM model makes two assumptions. First, that the current implicit state is only related to the last implied state; second, that the current output state is only related to the current implied state (Dugad and Desai, 1996). However, the reality is not so simple. A CRF uses the entire output sequence and two adjacent implicit states to find a conditional probability (Lafferty et al., 2001). It can fit more complex situations. Practice has proven that the CRF works better than other models.

Recently, artificial neural networks have been used to do natural language processing tasks. For the sequence labeling problem, because of its equal length output, a recurrent neural network (RNN) is an appropriate model. It is more capable of understanding the information context; however, this is not a good method for learning state transfer laws. To improve the problems of exploding and vanishing gradients, new RNN units, e.g., long short-term memory (LSTM) and gated recurrent units (GRUs) (Chung et al., 2014), have been proposed.

In this study, we propose a BiLSTM-CRF model. Our model combines statistical learning with neural networks. The BiLSTM is used to obtain information about long or short distances in two directions (Huang et al., 2015). It then feeds the information to the CRF. Thus, the CRF can better use conditional probabilities to fit the data without handmade features. The CRF and LSTM models

Proceedings of the 8th International Joint Conference on Natural Language Processing, Shared Tasks, pages 73–77,
Taipei, Taiwan, November 27 – December 1, 2017. ©2017 AFNLP

are also used as part of the experiment to compare the models' performance.

The rest of this paper is organized as follows. Section 2 describes our model in detail. Section 3 presents our experiment, including the data preprocessing and results. Conclusions are drawn in Section 4.

2 Proposed Model

The proposed model consists of three major parts: the word-embedding layer, the bi-directional LSTM layer, and the CRF layer. CRF is a traditional statistical learning method for sequence labeling. It has two limits. First, it heavily depends on handcrafted features. Second, it cannot capture long distance context information. The context information for a CGED task is very important. For example, here are two correct sentences.

- "只有努力，才能过的更好。" (Only you work hard, you can be better.)
- "只要努力，就能过的更好。" (As long as you work hard, you will be able to be better.)

It is impossible to determine whether the sentence, "才能过的更好。" is correct without the previous context information. On the other hand, handcrafted features greatly increase the workload. Thus, our model combines an RNN with a CRF to improve the above problems. To capture information from two directions (Ma and Hovy, 2016), we use a bi-directional RNN instead of a unidirectional RNN. In addition, an LSTM cell is selected to avoid vanishing and exploding gradients (Sundermeyer et al., 2012).

We trained four models for four error types because there may be two or more errors in one position. Each model is given the original text index as input, its label sequence outputs 0 for correct and 1 for error.

The model includes three layers: a word-embedding layer to transfer the word index into word embedding, a BiLSTM layer to extract the information and features for each position, and a CRF layer to decode and produce labels. The three layers are introduced in the following sections.

2.1 Embedding Layer

A diagram of the embedding layer's structure is shown in Figure 1. It shows a pre-trained word-embedding lookup table. Every line of this table stands for one Chinese word. Therefore, the original text of the training data should be turned into a sequence of indexes for every word. This layer takes a sequence that contains the word indices, e.g., w_1, w_2,..., w_i,..., w_n where w_i, is an index number indicating the position of the original word in the table. Then, the layer finds the word vector for every index and outputs them in a new sequence, e.g., v_1, v_2,..., v_i,..., v_n where v_i is a word vector.

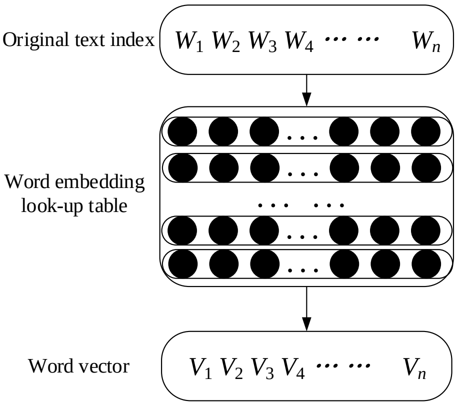
Original text index
W_1 W_2 W_3 W_4 ⋯ ⋯ W_n
Word embedding look-up table
Word vector
V_1 V_2 V_3 V_4 ⋯ ⋯ V_n

Figure 1: Embedding Layer

If the dimensionality of the original text index is N and the dimensionality of the word vector is M, the dimensionality of the output sequence should be $M * N$.

2.2 Bi-directional LSTM Layer

An RNN can effectively extract features from the entire sentence because of its ability to capture context information. For the reasons mentioned above, a bi-directional LSTM network was chosen.

An LSTM is a special type of RNN unit that can learn long-term dependency information. It is designed to avoid the long-term dependence problem. An LSTM can remove or increase the information to a cell state using a well-designed structure called a "gate". The gate determines whether information should pass. The LSTM uses the following formulas (Hochreiter and Schmidhuber, 1997):

$$i_t = \sigma(W_{vi}v_t + W_{hi}h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{vf}v_t + W_{hf}h_{t-1} + b_f) \quad (2)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{vc}v_t + W_{hc}h_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma(W_{vo}v_t + W_{ho}h_{t-1} + b_o) \quad (4)$$

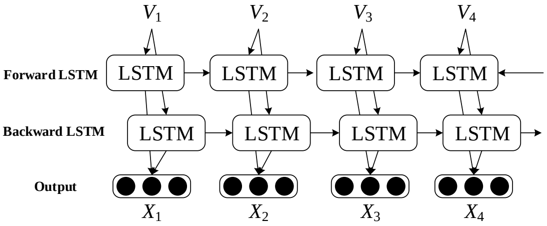

Figure 2: Bi-directional LSTM Layer

$$h_t = o_t \tanh(c_t) \quad (5)$$

where v is the input vector from the embedding layer. σ is the sigmoid activation function. i is the input gate, f is the forget gate, and o is the output gate. Parameters $\{W_{vi}, W_{hi}, W_{vf}, W_{hf}, W_{vc}, W_{hc}, W_{vo}, W_{ho}, b_i, b_f, b_c, b_o\}$ are the weights and biases of an LSTM cell. c is the cell vector and h is the hidden cell. K represents the output dimensionality of the LSTM unit. N is the dimensionality of the word vector. The size of the last four bias vectors are K and the others are $N * K$. Figure 2 shows a bi-directional LSTM network. The embedding output is fed into two LSTM layers with forward and backward directions. Two outputs from two layers at one position are linked into a new vector as the layer's output.

The dimensionality of the input is $M * N$. M is the length of the original text. Each word vector of input is also called an RNN time step. For instance, v_1, v_2,..., v_i,..., v_n is input, each v_i is a time step. For each time step, v_i is fed into two LSTMs; the forward LSTM layer produces an output vector $o_{i-forward}$ with dimensionality K, and the backward LSTM layer produces $o_{i-backward}$. Therefore, the result of a time step is $x_i(o_{i-forward}, o_{i-backward})$ and the dimensionality is $2 * K$, which is the result of combining the two layers' output. Thus, the dimensionality of the final output is $M * (2 * K)$.

2.3 CRF Layer

Because of the importance of the relationships between neighboring tags in CGED, a CRF is selected to capture the relationship. The CRF is a type of undirected discriminative graph model. In general, a CRF is a Markov random field with an observation set. A general CRF is defined as a Markov random field with random variable Y under the condition of random variable X. Y constitutes a Markov random field represented by an undirected graph, as in the formula below (Sutton and Mccallum, 2010):

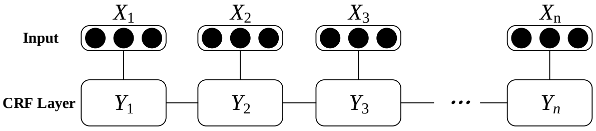

Figure 3: CRF Layer

$$P(Y_v|X, Y_w, w \neq v) = P(Y_v|X, Y_w, w \sim v) \quad (6)$$

where operator $\sim$ means that w and v have a public edge. X is the input variable or state sequence, and Y is the output variable or tag sequence. In our problem, we assume that X and Y have the same linear structure, as shown in Figure 3.

The conditional probability of random variable y with value x is given as follows:

$$P(y|x) = \frac{\prod_{i=1}^{n} M_i(y_{i-1}, y_i, x)}{Z(x)} \quad (7)$$

The denominator is a normalization item:

$$Z(x) = \sum_y \prod_{i=1}^{n} M_i(y_{i-1}, y_i, x) \quad (8)$$

where $M_i(y', y'', x)$ is a potential function, x is the input vector produced by the BiLSTM layer and y is the labeling of the input sentence.

3 Experiment

This section describes the contents of the experiment, including the training data processing, choice of parameters, experimental results, etc.

3.1 Dataset

The word embedding was trained using the word2vec toolkit with the Chinese Wikipedia corpus. According to the experimental results, the word-embedding results from word2vec are better than GloVe (Yang et al., 2016). In addition to the CGED17 training data, the HSK (i.e., Chinese Proficiency Test) training data from CGED16 was used. The number of training sets is 20,048, with 10,447 from CGED17 and 9,601 from CGED16.

For the reasons mentioned above, four models for every error type were selected. Thus, we preprocessed training sets for four error type models. For each error type, the position's label is 0 if correct, or 1 if erroneous. The training-data text was transferred into the word-index sequence, according to the pre-trained word-embedding table.

Model	Acc	Pre	Rec	F1
CRF	0.1805	0.1136	0.1483	0.1287
LSTM	0.1696	0.1824	0.0816	0.1128
BiLSTM-CRF	0.3325	0.2769	0.3502	0.3093

Table 1: Comparative results of three models.

3.2 Implementation Details

The experiment contains three models for comparison: CRF, LSTM, and BiLSTM-CRF. CRF represents the statistical learning method, which is the best simple statistical learning model in a variety of sequence labeling tasks. LSTM is a typical neural network model for sequence labeling. The above two models were used as the baseline for the experiment. The last model, proposed in this paper, combines both neural network and statistical learning models.

The CRF model was implemented using the CRF++ toolkit. CRF++ is an open-source, easy-to-use implementation of CRF, written in C++. The LSTM model and the BiLSTM-CRF model were implemented using the Keras framework with a Tensorflow backend. The CRF layer implementation in the BiLSTM-CRF model used Keras_contrib.

The training data for the three models comes from CGED16 or CGED17. Hence, the training data is regarded as a hyper-parameter. It will be CGED16 or CGED17 or a combination of both. In addition, there is a public hyper-parameter for the three models. The hyper-parameter for the CRF model is c, which controls the over-fitting of the training data. The hyper-parameters for LSTM and BiLSTM-CRF include the LSTM cell number and the training epoch.

Some empirical parameters are given as candidate values for the model. A grid search algorithm was used to find the best hyper-parameter combinations.

To evaluate the model's performance, we used four metrics such as accuracy (Acc), precision (Pre), recall (Rec) and F1-score (F1) for all three models on CGED16 HSK test data. Table 1 shows the best results for each model on position-level. The results show that the BiLSTM-CRF model has the best results in Table 1.

3.3 Experimental Results

Five teams submitted 13 results. We submitted three running results. The three results were produced by the three BiLSTM-CRF models that had the best three results on the CGED16 HSK test data. The three results have different hyperparameters, as shown in Table 2.

Parameter	Dataset	LSTM cell	Epoch
Run1	CGED16,17	120	22
Run2	CGED16,17	100	18
Run3	CGED16	100	18

Table 2: Parameter selection for BiLSTM-CRF models.

Results	Detection-Level			
	Acc	Pre	Rec	F1
Run1	0.5796	**0.65**	0.7163	0.6816
Run2	**0.5891**	0.6417	**0.7829**	**0.7053**
Run3	0.5311	0.6298	0.6148	0.6222

Table 3: Comparative results on detection-level.

The next three tables show the final test results for the three BiLSTM-CRF models. Table 3 shows the detection-level results. Table 4 shows the identification-level results. Table 5 shows the position-level results.

The false positive (FP) rates of the three results of BiLSTM-CRF models are 0.5796, 0.7383, and 0.614 shown in Table 6. The highest one is over 70%. This is because the model uses four submodels to generate the sentence label. If only one model misjudges the label of one position, from 0 to 1, it will produce a false negative (FN) sample. Thus, the model produces a high false positive rate. In addition, the experimental results show that the recall of the first two results is better than the previous in the detection level. This model is more likely to produce positive examples.

4 Conclusion

Compared with most previous models (Lee et al., 2016), the F1-score of the position level greatly increased with the CGED16 HSK test data. It was observed that neural network and statistical-learning methods could be combined to obtain better results. Among the participants this year, our model ranked third in the detection-level and identification-level results. In the position-level,

Results	Identification-Level			
	Acc	Pre	Rec	F1
Run1	**0.4218**	**0.4219**	0.4217	**0.4218**
Run2	0.3819	0.3825	**0.4575**	0.4167
Run3	0.3979	0.4086	0.3298	0.365

Table 4: Comparative results on identification-level.

Results	Position-Level			
	Acc	Pre	Rec	F1
Run1	**0.1778**	**0.1262**	**0.1191**	**0.1225**
Run2	0.1426	0.1056	**0.1191**	0.112
Run3	0.1702	0.0981	0.0698	0.0816

Table 5: Comparative results on position-Level.

Result	False Positive Rate
Run1	0.5796
Run2	0.7383
Run3	0.614

Table 6: False positive rates of three results.

our results ranked second among the participants.

Our model is a combination of BiLSTM and CRF. It combines the extraction capabilities of the LSTM context information and the conditional probability of CRF's local features. More complex models contain more parameters that need to be trained. Thus, more training data can improve the model; too little training data may cause overfitting.

The shared task provided us with more in-depth understanding about CGED. Our next step is to obtain more training data to enhance the model.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No.61702443 and No.61762091, and in part by Educational Commission of Yunnan Province of China under Grant No.2017ZZX030.The authors would like to thank the anonymous reviewers and the area chairs for their constructive comments.

References

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *arXiv preprint arXiv:1412.3555*.

Rakesh Dugad and U B Desai. 1996. A tutorial on hidden markov models. In *Proceedings of the IEEE*, pages 257–286.

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

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. In *arXiv preprint arXiv:1508.01991*.

John Lafferty, Andrew Mccallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning (ICML-01)*, pages 282–289.

Lung-Hao Lee, Liang-Chih Yu, 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 (NLP-TEA-16)*, pages 40–48.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL-16)*, pages 1064–1074.

Martin Sundermeyer, Ralf Schluter, and Hermann Ney. 2012. Lstm neural networks for language modeling. In *Proceedings of INTERSPEECH 2013*, pages 601–608.

Charles Sutton and Andrew Mccallum. 2010. *An Introduction to Conditional Random Fields.* Now Publishers Inc.

Jinnan Yang, Bo Peng, Jin Wang, Jixian Zhang, and Xuejie Zhang. 2016. Chinese grammatical error diagnosis using single word embedding. In *Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLP-TEA-16)*, pages 155–161.