Chinese Grammatical Error Diagnosis

Based on CRF and LSTM-CRF model

Yujie Zhou[1], Yinan Shao[2], Yong Zhou[3,*]
[1]Department of Education Information Technology, East China Normal University,
[2] Harbin Institute of Technology Shenzhen Graduate School,
[3] Department of Education Information Technology, East China Normal University
*Corresponding author
Contact:yzhou@ied.ecnu.edu.cn

Abstract

When learning Chinese as a foreign language, the learners may have some grammatical errors due to negative migration of their native languages. However, few grammar checking applications have been developed to support the learners. The goal of this paper is to develop a tool to automatically diagnose four types of grammatical errors which are redundant words (R), missing words (M), bad word selection (S) and disordered words (W) in Chinese sentences written by those foreign learners. In this paper, a conventional linear CRF model with specific feature engineering and a LSTM-CRF model are used to solve the CGED (Chinese Grammatical Error Diagnosis) task. We make some improvement on both models and the submitted results have better performance on false positive rate and accuracy than the average of all runs from CGED2018 for all three evaluation levels.

1 Introduction

Nowadays, more and more foreigners take Chinese as their second language. Unlike English, Chinese has no verb tenses or pluralities, and meanwhile there are various ways to express the same meaning in Chinese, so Chinese has been considered as one of the most difficult languages in the world(Bo Zheng et al., 2016). Chinese as a Foreign Language(CFL) learners often make grammatical errors such as redundant words (R), missing words (M), word selection errors (S), and word ordering errors (W), due to language negative migration, over-generalization, teaching methods, learning strategies and other reasons. Natural Language Processing System(NLPS) which can detect and correct grammatical errors are important and invaluable to language learners. (Leacock et al., 2010). However, few grammar checking applications have been developed to support CFL learners. The goal of the CGED (Chinese Grammatical Error Diagnosis) task is to develop NLP (Natural Language Processing) techniques to automatically diagnose grammatical errors in Chinese sentences written by CFL learners.

In this paper, we use both a conventional linear CRF model (Lafferty et al., 2001) with specific feature engineering and a LSTM-CRF model to solve CGED task. Many researchers have already used these two models in the past few years, but our team make some improvement on both models. For CRF model, we integrate the syntactic feature into the CRF model. Character itself, POS feature and syntactic feature are used to generate 50 combinatorial features by template technology. As for LSTM-CRF model, most researchers use tag transition features only in CRF layer. The major improvement of our work is that more conventional sparse CRF features are incorporated into the CRF layer such as bag of POS n-grams features, words features, tag transition features, etc.

The rest of the paper is organized as follows: Section 2 gives the definition of the CEGD task. Section 3 introduces two methods we use to solve the CGED task. Section 4 describes the dataset we use, the evaluation results on the validation set and the test set. Section 5 discusses conclusion and future work.

2 Task Definition

The task of CGED is defined as follows: given a Chinese sentence, the goal of CGED tool is to diagnose four types of grammatical errors, including redundant words (R), missing words (M), words selection errors (S) and word ordering errors (W).

Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications, pages 165–171
Melbourne, Australia, July 19, 2018.

他[1]们[2]是[3]不[4]但[5]我[6]父[7]母[8]，[9]而[10]且[11]是[12]人[13]生[14]的[15]先[16]辈[17]。[18]		
Error Type	W	S
Error position-Start	**3**	**16**
Error position-End	**5**	**17**
Correction	他们不但是我父母，而且是人生的导师。	

Table 1: Two errors are found in the sentence above, one is word ordering error (W) from position 3 to 5, the other is word selection error (R) from position 16 to 17..

虽[1]然[2]吃[3]绿[4]色[5]的[6]食[7]品[8]是[9]对[10]身[11]体[12]健[13]康[14]很[15]有[16]好[17]处[18]。[19]		
Error Type	**R**	**M**
Error position-Start	**6**	**19**
Error position-End	**6**	**19**
Correction	虽然吃绿色食品是对身体健康很有好处的。	

Table 2: Two errors are found in the sentence above, one is redundant word (R) error at position 6, the other is missing word (M) error at position 19.

The input sentence may contain one or more such errors. The developed tool should indicate each error type and its position in the given sentence. To be specific, if an input sentence contains the grammatical errors, the output of each error should include four items: the id of the sentence, the positions of starting and ending character at which the grammatical error occurs, and the error type which should be one of the defined errors: "R", "M", "S", and "W". Example sentences and corresponding notes are shown in Table 1 and Table 2.

3 Methodology

We use two different models to solve the CGED task. One is the traditional model based on Conditional Random Field (CRF) with specific feature engineering. Many researchers have chosen CRF based models to solve CGED2016 and CGED2017 task. From previous research, we know that the CRF model with carefully designed feature templates could maintain the performance with neural networks at the same level (Lung-Hao Lee et al., 2016), especially when the training data is not big enough. Another is LSTM-CRF model with conventional sparse CRF features. The LSTM-CRF model is also used by some researchers before (Bo Zheng et al., 2016). The research proved that LSTM is effective in various applications that involves sequence modeling. This time, we make some improvements on both CRF model and LSTM-CRF model.

3.1 CRF model with feature engineering

Conditional random fields (CRF), an extension of both Maximum Entropy Model (MEMS) and Hidden Markov Models (HMMs), has been used to solve some natural language processing problems such as word segmentation, information extraction and parsing. The CGED task can be considered as a sequence labeling problem which assigns each Chinese character in a sentence with a tag including the error types (R, M, S, W). CRF is a sequence labelling model with flexible feature space. Therefore, with given feature set and labeled training data, the CRF model can be used to solve the CGED task. The model can be defined as:

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

where Z(x) is the normalization factor, f_k is the feature sets and λ_k is the corresponding weight of the features. x is the sequence of the training sentences (the first column of Table 3), and y is the error type label (the forth column of Table 3) which includes O(Correct), R(Redundant words), M(Missing Words), S(Selection errors) and W(Word ordering errors). Tag 'O' indicates correct characters, 'B-X' indicates the beginning positions for errors of type 'X' and 'I-X' shows the middle or ending positions for errors of type 'X'.

For example, the label 'B-S' indicates this character is the beginning of a words selection error. The CRF model can generate the corresponding label sequence y according to the sequence data x. The second column of Table 3 is the POS(Part-of-speech) feature. The task is being solved at the character level. The POS tag was split of a word to character level by attaching position indicators ('B-' and 'I-') to the POS of a word. We use LTP Segmenter and Postagger which is a Chinese Language Technology Platform (Wanxiang Che et al., 2010) to tag the training sentences.

Character	POS	Parsing	Error
他	B-r	2	0
们	I-r	2	0
是	B-v	0	B-W
不	B-c	5	I-W
但	I-c	5	I-W
我	B-r	5	0
父	B-n	2	0
母	I-n	2	0

Table 3: A snapshot of a sample sentence

The third column of Table3 is syntactic feature of the character. Syntactic feature is the dependency parsing results of a sentence. Dependency parsing provides a representation of grammatical relations between words in a sentence. To be specific, dependency parsing can be used to identify the grammatical components of the subject in the sentence and analyze the relationship between the components. Figure 1 and Figure 2 shows the example of the dependency parsing. LTP is also used to parse the sentence. The output of the parsing of the sample sentence is "2:SBV 0:HED 5:ADV 5:ATT 2:VOB". Table 4 describe the meaning of these tags. The number means which word in the sentence is related to the current word. For example, 2:SBV means the 2th word "是" and the current word "他们" are the subject-predicate relationships . We can find out the grammatical relations of the sentence more clearly from the figures below. Figure 1 is the sentence with grammatical errors and Figure 2 is the correction. The number of the output is used as the syntactic feature.

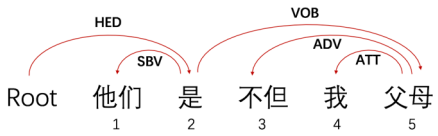

Figure 1: Dependency parsing of the sentence with grammatical errors

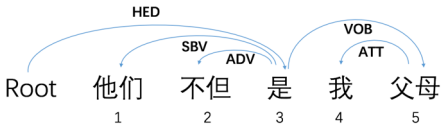

Figure 2: Dependency parsing of the correct sentence

Tag	Description
SBV	subject-verb
VOB	verb-object
IOB	indirect-object
FOB	fronting-object
DBL	double
ATT	attribute
ADV	adverbial
CMP	complement
COO	coordinate
POB	preposition-object
LAD	left adjunct
RAD	right adjunct
IS	independent structure
HED	head

Table 4: Description of syntactic features tag

Feature Templates
00-04: $Character_{i+k}$ (k=-2,-1,0,1,2)
05-09: POS_{i+k} (k=-2,-1,0,1,2)
10-14: $Parsing_{i+k}$ (k=-2,-1,0,1,2)
15-18: $Character_i/Character_{i+k}$ (k=-2,-1,1,2)
19-23: $Character_i/POS_{i+k}$ (k=-2,-1,0,1,2)
24-28: $Character_i/Parsing_{i+k}$ (k=-2,-1,0,1,2)
29-32: POS_i/POS_{i+k} (k=-2,-1,1,2)
33-37: $POS_i/Parsing_{i+k}$ (k=-2,-1,0,1,2)
38-41: $POS_i/Character_{i+k}$ (k=-2,-1,1,2)
42-45: $Character_i/Character_{i+k}/POS_{i+k}$ (k=-2,-1,1,2)
46-49: $POS_i/Character_{i+k}/POS_{i+k}$ (k=-2,-1,1,2)

Table 5: Feature templates

CRF++ (Kudo et al.,2007), a linear-chain CRF model software tool, is used to built the CRF model. To train a model with CRF++, we need to build some templates first. We use 50 templates to generate 50 combinatorial features which is listed in Table 5. The format of each template is %X[row, col], in which row is the number of row in a sentence and column is the number of column. The template %x[0,0]/%x[0,1] means the feature combining the current character and the next POS tag. Take the character "是" in sample sentence in Table 3 as an example, %x[0,0]/%x[0,1] represents "是/B-v".

3.2 LSTM-CRF model

LSTM-CRF model is currently a strong baseline in the task of sequence labeling. Compared with the conventional Bi-LSTM neural network, LSTM-CRF model can directly model probability distribution of the the label sequence by a CRF layer, and achieve better performance on several datasets (Z.Huang et al., 2015; X.Ma et al., 2016). An illustrative graph is shown in Figure 3. Under this framework, neural network (i.e. LSTM) is used to compute the features score in CRF, which are called neural features. These neural features are similar to the conventional sparse CRF features, which are directly used to compute the score of a given label sequence.

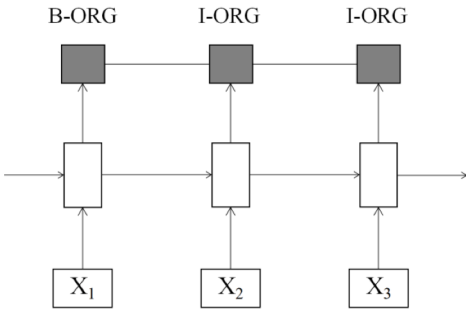

Figure 3: LSTM-CRF model

A LSTM-CRF model can efficiently capture past input features via a LSTM layer and other user specified sparse features (e.g. transition feature, n-gram feature.) via a CRF layer. In our case, plenty of features are considered, here we only take tag transition feature as an example for simplicity. Denoting a tag transition matrix $[A]$, where each $[A]_{i,j}$ models the transition score from i^{th} tag to j^{th} tag for a pair of consecutive time step. Note that this transition matrix is position independent. De-noting the matrix of scores $f_\theta[x]_i^T$ are output by the network. The element $[f_\theta]_{[i]_t}$ of the matrix is the score output by the network with parameters θ, for the sentence $[x]_i^T$ and for the i^{th} tag, at the t^{th} word. The score of a sentence $[x]_i^T$ along with a path of tags $[i]_i^T$ is then given by the sum of transition scores and net-work scores:

$$s([x]_i^T, [i]_i^T, \theta) = \sum_{t=1}^{T} (w_1[A]_{[i]_{t-1},[i]_t} + w_2[f_\theta]_{[i]_t,t})$$

Here we modified the objective function to attend differentially to neural features and conventional CRF sparse features. It is worth noting that the dynamic programming can be used efficiently to compute $[A]_{i,j}$ and optimal tag sequences for inference. Then, the modified CRF layer models the conditional probability of possible output sequence s over input sequence x as:

$$p(s|x) = \frac{1}{Z(x)} \exp\{s([x]_i^T, [i]_i^T, \theta)\}$$

$s([x]_i^T, [i]_i^T, \theta)$ is the score of a sentence $[x]_i^T$ along with a path of tags $[i]_i^T$. $Z(x)$ is the normalization factor of all the possible paths of tags $[i]$ over input sequence x. For our LSTM CRF training, we use the maximum conditional likelihood estimation. For a training set$\{(x_i, i_i)\}$, the log-likelihood is given as:

$$\mathcal{L}_{\mathcal{D}}(W) = \sum_{i \epsilon \mathcal{D}} \log p(i|x)$$

Maximum likelihood training chooses parameters W such that the log-likelihood $\mathcal{L}_{\mathcal{D}}(W)$ is maximized.

The training algorithm is giving as follows:

Algorithm 1 LSTM CRF training procedure

for each epoch do
for each batch do
1) neural network forward pass
forward pass for LSTM state
2) CRF layer forward and backward pass
3) neural network backward pass: backward pass for LSTM
4) update parameters
end
end

Table 6: the LSTM-CRF training procedure

In most LSTM-CRF based models (Z.Huang et al., 2015; X.Ma et al., 2016; M.Rei et al., 2016;

L.Kong et al., 2016; G. Lample et al., 2016), only tag transition features are considered in CRF layer. In our case, more conventional sparse CRF features are incorporated into the CRF layer. Specifically, we consider the following features defined over the inputs:

• Words features. Words that appear around the current position with a window of size 3.

• POS tags features. POS tags that appear around the current position with a window of size 3.

• Word n-grams features. Word n-grams that contain the current position, for n = 2, 3, 4.

• POS n-grams features. POS tags that contain the current position, for n = 2, 3, 4.

• Bag of words features. Bag of words that contains the current word, with a window of size 5.

• Tag transition features. Tag n-grams that contain the current position, for n = 2.

4 Experiments

4.1 Dataset

We collect datasets from CGED-HSK-2016, CGED-2017 and CGED-2018 as our training set and validation set. Table 7 shows the distributions of error types in both the training set and validation set. The ratio of training set size to validation set size is about 8:1. Besides the sentences with grammatical errors, 1539 correct sentences are added into the validation set.

	Training Set	Validation Set
Error	52313	6773
R	11598(22.17%)	3880(57.29%)
M	13931(26.63%)	991(14.63%)
S	23014(43.99%)	1620(23.82%)
W	3769(7.20%)	282(4.16%)

Table 7: The distributions of error types

4.2 Validation

We use the validation set to evaluate the results of the CRF models with and without syntactic feature. CRF-1 refers to the model with syntactic feature and CRF-2 refers to the model without syntactic feature. According to the results in Table 8, we can find out that syntactic feature does help to improve the performance of the CRF model. Therefore, CRF model with both Part-Of-Speech(POS) feature and syntactic feature is used in our final run.

	CRF-1	CRF-2
Accuracy	96.98%	96.34%
Precision	35.32%	31.53%
Recall	13.46%	12.28%
F1	19.49%	17.68%

Table 8: Evaluation results of CRF model on validation set for position level

We also thoroughly study the effectiveness of the handcraft features in our LSTM-CRF model. Experiment results are shown in Table 9. LSTM-CRF-1 refers to the LSTM-CRF model with handcraft features defined in section 3.2. LSTM-CRF2 refers to the LSTM-CRF model with no handcraft features (i.e. only tag transition feature is considered). As the experiment results shown that the feature engineering in CRF part can improve the performance (i.e. F1 value) about 2%, thus we use the LSTM-CRF1 model as our final model.

	LSTM-CRF-1	LSTM-CRF-2
Accuracy	97.28%	96.63%
Precision	33.10%	29.60%
Recall	15.76%	14.22%
F1	21.35%	19.21%

Table 9: Evaluation results of LSTM-CRF model on validation set for position level

4.3 Evaluation Results

In the CGED2018 shared task, there are 12 teams submitted the results, totally 32 runs. Among them, our team submitted three runs. Run1 and Run2 are based on the CRF model with different size of training set while Run3 is based on the LSTM-CRF model. The average of all runs is calculated from 32 runs of the 12 teams.

Table 10 shows the false positive rate of the 3 runs of our team and the average of all runs. FP (False Positive) is the number of sentences in which non-existent grammatical errors are identified as errors, so the lower the better. The best false positive rate of our team is 0.1255 (Run3) which is much lower than the average rate of all runs.

Table 11 Table 12 and Table 13 shows the evaluation result for detection level, identification level and position level. The submitted results of our

Submission	False Positive Rate
Run1	0.3470
Run2	0.3873
Run3	**0.1255**
Average of all runs	0.46685

Table 10: The False Positive Rate (The lower the better)

	Detection Level		
	Accuracy	**Recall**	**F1**
Run1	0.5923	0.5445	0.5993
Run2	0.5796	0.5536	0.5959
Run3	0.5762	0.3417	0.4745
Average of all runs	0.58701	0.63484	0.61310

Table 11: Evaluation Results for Detection Level

	Identification Level		
	Accuracy	**Recall**	**F1**
Run1	0.4767	0.2836	0.3556
Run2	0.4452	0.2740	0.3392
Run3	0.6139	0.1818	0.2805
Average of all runs	0.46223	0.41422	0.37791

Table 12: Evaluation Results for Identification Level

	Position Level		
	Accuracy	**Recall**	**F1**
Run1	0.1238	0.0667	0.0867
Run2	0.0901	0.0506	0.0648
Run3	**0.3745**	0.0858	**0.1397**
Average of all runs	0.17532	0.11386	0.12473

Table 13: Evaluation Results for Position Level

team have better performance on accuracy than the average of all runs from CGED2018 for all three evaluation levels, but all three runs do not perform well on recall rate. Table 13 indicates that Run 3 achieved the accuracy of 0.3745 for position level which is the most difficult level and it leads to the final F1 score of 0.1397 although the recall rate is still not above the average.

5 Conclusion and Future Work

In this paper, we thoroughly study the task of Chinese grammatical error diagnosis and propose two models to handle this issue. We use a conventional linear CRF with specific feature engineering and a LSTM-CRF model to solve this task. We make some improvements on these two models based on the previous research and get better performance on False Positive Rate and Accuracy than the average of all runs from CGED2018 for all three evaluation levels including detection level, identification level and position level, but all three runs do not perform well on recall rate which should be improved in the future . Future work includes explorations of semi-CRFs and neural semi-CRFs for the CGED shared task and exploring more task specific features such as phonology feature and grapheme feature.

Acknowledgments

This study was funded by Special Foundation for Graduate Students Attending International Conferences of East China Normal University.

References

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

Lunghao Lee, Rao Gaoqi, Liangchih Yu, Xun Endong, Baolin Zhang, and Liping 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, pages 40–48.

Bo Zheng, Wanxiang Che, Jiang Guo, and Ting Liu. 2016. *Chinese grammatical error diagnosis with long short-term memory networks.* In Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications, pages 49–56.

Po-Lin Chen, Wu Shih-Hung, Liang-Pu Chen, Ping-Che Yang, and Ren-Dar Yang. 2015. *Chinese grammatical error diagnosis by conditional random fields,* In Proceedings of The 2nd Workshop on Natural Language Processing Techniques for Educational Applications, pages 7–14.

Po-Lin Chen, Wu Shih-Hung, Liang-Pu Chen, and Ping-Che Yang. 2016. *Improving the selection error recognition in a Chinese grammar error detection system*. International Conference on Information Reuse and Integration, pages 525-530.

Jui-Feng Yeh, Chan-Kun Yeh, Kai-Hsiang Yu, Ya-Ting Li, and Wan-Ling Tsai. 2015. *Condition random fields-based grammatical error detection for Chinese as second language*. In Proceedings of The 2nd Workshop on Natural Language Processing Techniques for Educational Applications, pages 105–110.

Yajun Liu, Yingjie Han, Liyan Zhuo, and Hongying Zan. 2016. *Automatic grammatical error detection for Chinese based on conditional random field*. In Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications, pages 57–62.

Wanxiang Che, Zhenghua Li, and Ting Liu. 2010. *LTP: A Chinese language technology platform*. In Proceedings of the Coling 2010: Demonstrations, pages 13-16.

Taku Kudo. 2007. *"CRF++: Yet Another CRF toolkit"*, https://taku910.github.io/crfpp/.

Zhiheng Huang, Wei Xu and Kai Yu. 2015. *Bidirectional LSTM-CRF models for sequence tagging*. Computer Science.

Xuezhe Ma and Eduard Hovy. 2016. *End-to-end sequence labeling via Bi-directional LSTM-CNNs-CRF*. In Proceedings of ACL, pages 1064-1074.

Lingpeng Kong, Chris Dyer and Noah A. Smith. 2016. In Proceedings of ICML.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami and Chris Dyer. 2016. *Neural architectures for named entity recognition*. pages 260-270.