Chinese Grammatical Error Diagnosis Using Single Word Embedding

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

Abstract

Automatic grammatical error detection for Chinese has been a big challenge for NLP researchers. Due to the formal and strict grammar rules in Chinese, it is hard for foreign students to master Chinese. A computer-assisted learning tool which can automatically detect and correct Chinese grammatical errors is necessary for those foreign students. Some of the previous works have sought to identify Chinese grammatical errors using template- and learning-based methods. In contrast, this study introduced convolutional neural network (CNN) and long-short term memory (LSTM) for the shared task of Chinese Grammatical Error Diagnosis (CGED). Different from traditional word-based embedding, single word embedding was used as input of CNN and LSTM. The proposed single word embedding can capture both semantic and syntactic information to detect those four type grammatical error. In experimental evaluation, the recall and f1-score of our submitted results Run1 of the TOCFL testing data ranked the fourth place in all submissions in detection-level.

1 Introduction

The growing global influence of China has prompted a surge of interest in learning Chinese as a foreign language (CFL) (Yu et al., 2014). The number of commonly used Chinese characters are about 2000, but there are a large number of corresponding vocabulary. In this way, some same words may have different meanings because of different contexts and moods. This has caused difficulties for foreigners to learn Chinese. However, while many learning tools of computer-assisted have been developed for students of English as a Foreign Language (EFL), there is relatively little support for CFL learners. Especially, these tools cannot automatically detect and correct Chinese grammatical errors. For example, although Microsoft Word has been integrated with robust English spelling and grammar checking for many years, the tools for Chinese are still primitive (Yu et al. 2014). The aim of Chinese Grammatical Error Diagnosis (CGED) shared task is to develop computer-assisted tools to help detect four types of grammatical errors in the written Chinese, including missing word (**M**), redundant word (**R**), word ordering error (**W**) and word selection error (**S**).

The shared task is divided into three levels, including detection-, identification- and position-level. Detection-level task can be considered as a binary classification of a given sentence, i.e., correct or incorrect should be exactly as same as the gold standard. All error types will be treated as incorrect. Identification-level task could be considered as a multi-label classification task. In addition to the correct instance, all error types should be clearly identified. This level identified the error types for the wrong sentence. Besides identifying the error types, the position-level also judges the positions of erroneous range. Some of the previous works have sought to identify Chinese grammatical errors using template- and learning-based methods.

Wu et al. (2010) proposed a combination of relative position and analytic template language model to detect Chinese errors written by American learners. Yu and Chen et al. (2012) used simplified Chinese corpus to study word ordering errors (W) in Chinese and proposed syntactic features, external corpus features and perturbation features for W detection. Cheng et al. (2014) detected and corrected word

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http://creativecommons.org/licenses/by/4.0/

ordering errors by using conditional random field (CRF) (Lafferty, 2010) and support vector machine (SVM) together with frequency learning from a large *n*-gram corpus. Zampieri et al. (2014) used frequent *n*-grams and news corpus as a reference corpus to detect errors in the written by CFL learners. Chen et al. (2015) used conditional random fields based on word attributes and grammar rules to detect Chinese syntax errors. However, there are several limitations of the existing methods, these methods on the one hand only consider part of the grammar rules, and the other hand only consider the order of the word or relationship. They didn’t consider the semantic relationship between words and the flexible expression and irregular grammar in Chinese.

In this paper, we introduced convolutional neural network (CNN) and long-short term memory (LSTM) for the task of Chinese Grammatical Error Diagnosis. In contrast of traditional word-based embedding (Mikolov et al., 2013), single word embedding was used as input of CNN and LSTM, which is similar to character-level embedding in English. The proposed single word embedding can capture both semantic and syntactic information to detect those four type grammatical error. Then, the single word vectors were used to establish the sentence representation for detection-level and identification-level tasks. In position-level, this paper also used single word embedding as input feature to train a multi-class support vector machine (SVM) to identify the error type of each word. The recall and f1-score of the submitted results Run1 of the TOCFL testing data ranked the fourth place in all submissions in detection-level. In identification-level, the recall score also ranked in the fourth place.

The remainder of this paper is organized as follows. Section 2 describes the learning method that used for Chinese grammatical error diagnosis. Section 3 shows the experimental results. Conclusions are drawn in section 4.

2 Feature Selection and Error Detection

The procedure for using single word embedding for each level grammatical error detection is described as follow. Given a large Chinese corpus, single word embedding are first trained through word2vec and fastText tools. Then, the obtained single word representations were input to CNN, LSTM and SVM for the mentioned three level diagnosis tasks. The following sub-sections explain the details of single word embedding and the CNN, LSTM and SVM models implementation.

2.1 Single Word Embedding

We use fastText (Bojanowski et al. 2016) and word2vec (Mikolov et al. 2013) toolkits to train single word embedding on Chinese Wikipedia corpus.

Word2vec is a set of related models used to generate word embedding. These models are two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec toolkit takes large corpus text as its input and produces a high-dimensional null and each unique word in the corpus is assigned the corresponding vector in that space. In Chinese, a sentence may be generated to ambiguities segmentation, leading to different embedding results.

- Example 1 “乒乓球拍卖完了”
  1. “乒乓球/拍卖/完了” (The auction of table tennis is over.)
  2. “乒乓球拍/卖/完了” (The paddles of table tennis are sold out.)
- Example 2 “在这种环境下工作是太可怕了”
  1. “在/这种/环境/下工/是/太/可怕/了”
  2. “在/这种/环境/下/工作/是/太/可怕/了”(Working in such environment is horrible)

For Example 1, these two forms of segmentation are both syntactically and semantically. Even if the manual division of the sentences in Example 1 will be ambiguous. In this case, we can get correct sentence segmentation when taking the context into consideration. However, for example 2, only the second sentence segmentation is correct. Therefore, the segmented for Chinese word may produce ambiguities segmentation.

In addition, there are 696,326 words in the Chinese corpus of Chinese Wikipedia corpus. There are some uncertainties in the Chinese word segmentation, and it cannot completely cover the vocabulary of the training set. Such as “开一个” (open a), "上海"(Shanghai) did not appear directly in the corpus, but

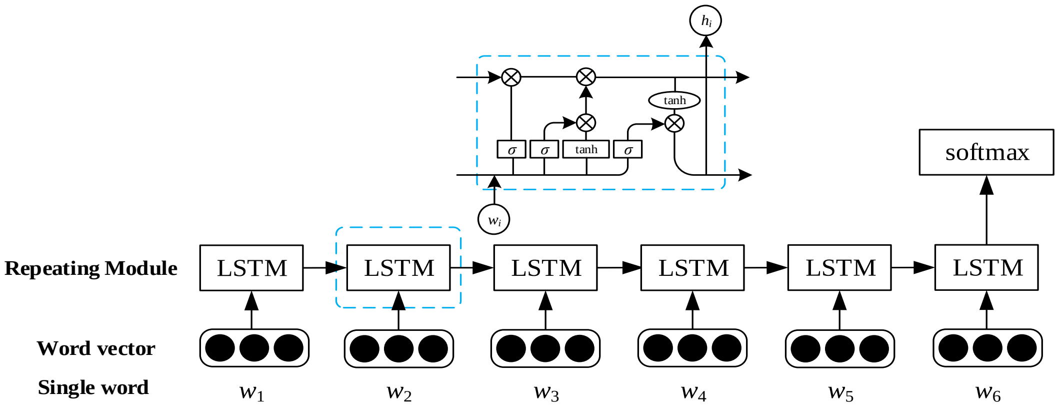

Fig. 1. The sequential LSTM model

these was split or integrated with other words, such as “开” (open), “一个”(one), "上海医院"(Shanghai Hospital).

The grammatical rules of Chinese are different from English. The spaces between English words can be used to segment a sentence. In English, the method of word embedding acquires the characteristics of the sentence, but this method is not suitable for the Chinese sentences. Based on the characteristics of Chinese, a Chinese sentence without considering the context of the case can only use single word to segment the sentence. By using the single word embedding method, all Chinese sentences in corpus can be segmented into single words. Meanwhile, the single word embedding can be obtained by using either word2vec or fastText. Finally, we can get the text features of Chinese sentences.

All words in the document are converted into vectors by lookup table, and the results are classified by linear classifier. As same as word2vec, fastText used *n*-grams to train embedding. The word vector will be derived from the *n*-grams. This improvement enhances the effect of the model on morphology, which means that the distance of similar words will be smaller. Thus, two different single word embedding are obtained by these two models.

2.2 Grammatical Error Diagnosis Models

Taking the single word embedding as input, the convolutional neural network (CNN) (LeCun, et al., 1990), the recursive neural network (RNN) (Ronald, et al.,1989), and the long-short term memory (LSTM) (Hochreiter et al., 1997) were introduced to classify the sentences. The sequential LSTM model was shown in Fig. 1.

The obtained single word vectors (word2vec and fastText) were fed to deep neural network models, such as CNN, RNN and LSTM. To tune the best performance, 5-folder cross-validation was applied. For unbalanced problem of positive and negative training samples in identification- and position-level, e.g. the number of train samples within **R** label is smaller *L* times than other classes, we divided the more abundant class into *L* distinct clusters. Then *L* classifiers were trained, where each classifier is trained on only one of the distinct clusters, but on all of the data from the rare class. That is, the data from rare class was used in the training of all *L* classifier. Finally, the averaging output of *L* models was considered as the final classification result. The cross-validation results of different models are shown in Table 1. Then, the support vector machine model is applied to find the error location in each error sentence.

This paper completes the Shared Task requirement in the following three steps.

- **Determining the correctness of a sentence (detection-level).** The method adopted in this paper is to segment each sentence in the training set by every single word, e.g. “你/开/一/个/庆/祝/会/的/时/候/我/不/能/会/参/加/是/因/为/我/在/外/国/做/工/作”, so that each single word in a sentence corresponds to a single word vector. By using word2vec and fastText toolkit, we train single word embedding in the Chinese Wikipedia corpus. Then, we trained neural network models, such as CNN, RNN, and LSTM, to distinguish correct sentences from wrong sentences in training set. The trained models were then used to categorize each sentence in testing set into correct or wrong class.
- **Judge the four types of errors (identification-level).** The method in this level can be considered

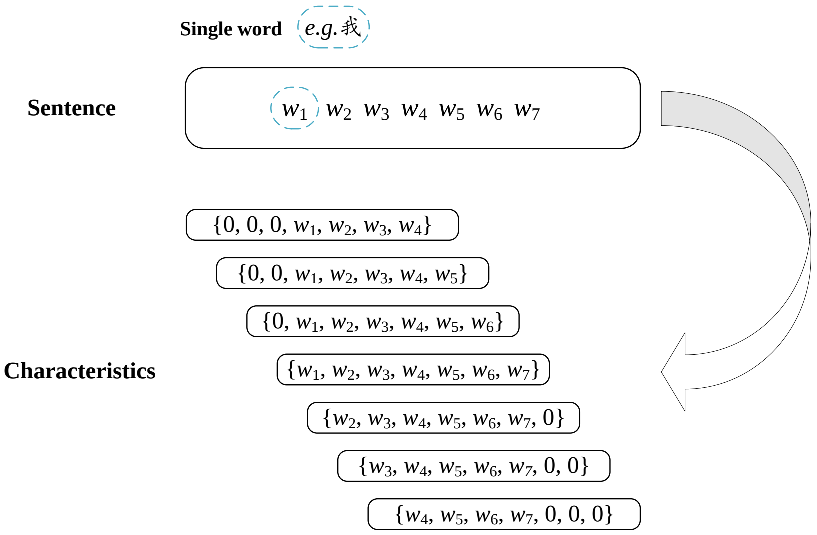
Single word
e.g.我
Sentence
w_1 w_2 w_3 w_4 w_5 w_6 w_7
Characteristics
{0, 0, 0, w_1, w_2, w_3, w_4}
{0, 0, w_1, w_2, w_3, w_4, w_5}
{0, w_1, w_2, w_3, w_4, w_5, w_6}
{w_1, w_2, w_3, w_4, w_5, w_6, w_7}
{w_2, w_3, w_4, w_5, w_6, w_7, 0}
{w_3, w_4, w_5, w_6, w_7, 0, 0}
{w_4, w_5, w_6, w_7, 0, 0, 0}

Fig. 2. Single word embedding feature extraction for SVM

as a multi-label problem, which is something different with the detect-level. Each samples in this level contain only one or more error labels. Therefore, *one-vs-all* strategy were applied. It consists in fitting one classifier per class. For each classifier, the class (**R**, **M**, **S** and **W**) is fitted against all the other classes. One advantage of this approach is its interpretability. Since each class is represented by one classifier only, it is possible to categorize each sentence with more than one labels. For each error type, we trained a neural network model, such as CNN, RNN and LSTM to determine the error type in testing dataset.

- **Locate the wrong position (position-level).** In this level, support vector machine (SVM) (Christopher, 1998) was used as classifier to locate the wrong position. As shown in Fig. 2, a single word with its context can be considered as a training sample, so that the word appeared in the middle can be judged whether the location is wrong or not. The size of slide window was set to 7 in our experiment. Then, single word embedding of 7 words were concatenated as a 1-D vector, and then fed into SVM. Since each position is assigned to one and only one label. That is, position can be either **R**, **M**, **S**, **W** or correct, but not both at the same time. Therefore, we trained a multi-class SVM to determine the error type of each word in testing sentence.

3 Experimental Results

3.1 Dataset

Two Chinese corpus are given in this shared task: TOCFL and HSK. TOCFL is the traditional Chinese training set, and HSK is the simplified Chinese training set. Apart from the difference between traditional and simplified, there is almost no difference of grammar and expression. In the training set, each id corresponds to two sentences, including a wrong sentence, and a corrected formation of this sentence. The error type, as well as the location of the error range are also provided. Each wrong sentence may have one error type, or more. These data sets were preprocessed to extract the single words, the error types and the error positions. The TOCFL corpus consists of 10693 training texts and 3528 testing texts. Similarly, the HSK corpus consists of 10072 training texts and 3011 testing texts.

3.2 Implement Details

As previously mentioned, the proposed method includes neural network and single word embedding. The two parts may have their own parameters for optimization. We use fastText and word2vec toolkits to pre-train the word vector on Chinese Wikipedia corpus.

The Chinese Wikipedia corpus is segmented by single words, we set the embedding dimension of each single word to 300. In this way, we can get 300-dimensional feature vectors for all single words in the corpus. There are 2563 single words in TOCFL training set and 2583 single words in HSK training set.

Method	Detection Level			
	Accuracy	Precision	Recall	F1
LSTM	0.3769	0.3813	0.4088	0.3923
RNN	0.4081	0.4053	0.4017	0.4013
CNN	0.480	0.486	0.660	0.560

Table 1: The cross-validation results of different methods using word2vec embedding

Results	False Positive Rate	
	TOCFL	HSK
Run1	0.6289	0.5608
Run2	0.5931	0.7122
Run3	0.3382	0.271
Average	0.5201	0.5147

Table 2: The false positive rate results of different methods.

The two training sets have already contained the most commonly used 2000 Chinese characters. Therefore, this method can obtain the text feature for each sentence in the training set.

We submitted three results for both TOCFL and HSK testing sets, the first submission (Run1) used the word representation trained by word2vec and classified by LSTM. The second submission (Run2) also used the LSTM to do the classification with word representation trained by fastText. Besides, the word representation of the third submission (Run3) was trained by word2vec and classified by CNN. The results can be obtained in three steps in Section 2.2. The sharing task has five evaluation indicators, they are false positive rate, accuracy (**Acc**), precision **(Pr**), recall (**Re**) and f1-score (**F1**).

3.3 Experimental Results

A total of 15 teams participated in the sharing of tasks, nine teams submitted the results of the operation in the final. For TOCFL training set, only 5 teams submitted the results of the operation. For the HSK test set, 9 teams have submitted the results of the operation. We have submitted three runs of results for both test sets. Table 2 shows the false positive rate. Table 3, Table 4 and Table 5 show the formal run results in detection-level, identification-level, and position-level respectively.

As shown in Table 2, the accuracy of the following two levels is reduced due to the high false positives. The results of Run1 and Run2 shows that the performance of word vectors trained by word2vec are better than that by fastText, since the fastText model makes the distance between similar words smaller. For example, the meaning of "trading" (贸易) is close to the "transaction" (交易) in Chinese, and word2vec can reflect this relationship. However, in the fastText, "trading" (贸易) is even more closer to "trade laws" (贸易法), which makes word vector by fastText cannot accurately reflect the sentence characteristics. Similarly, by comparing the results of Run1 and Run2, we can find that the classification performance of LSTM is better than CNN. Although CNN considers the local characteristics of the sentence, which makes it easy to high degree of similarity between the two sentences, LSTM can consider the relationship between the contexts of the sentence, which is particularly important in Chinese. Therefore, LSTM can capture the logical relationship between the sentences, e.g. *cause* and *contrast* relationship, etc.

Since the number of sentence in different label (correct and incorrect) is unbalanced, which will impact the result in all detection-, identification- and position-level. Hence, the wrong sentences are the majority in testing date. If all the sentences in the testing set are classified as wrong, the learning model will get high accuracy, precision, recall and f1-score, and even a higher false positive rate.

4 Conclusion and Future work

Since the grammar rules in Chinese are formal and strict, it is hard for foreign students to master Chinese. A computer-assisted learning tool which can automatically detect and correct Chinese grammatical errors is necessary for those foreign students. In this paper, neural network models, such as convolutional neural network (CNN) and long-short term memory (LSTM), were introduced for the task of Chinese

Results	Detection-Level							
	TOCFL				HSK			
	Acc	Pr	Re	F1	Acc	Pr	Re	F1
Run1	0.5420	0.5444	0.7014	0.6130	0.5191	0.5069	0.6026	0.5506
Run2	0.5026	0.5167	0.5918	0.5517	0.4949	0.4886	0.7113	0.5793
Run3	0.4847	0.503	0.3195	0.3908	0.5058	0.4902	0.2724	0.3502
Average	0.5098	0.5214	0.5376	0.5185	0.5066	0.4952	0.5288	0.4934

Table 3: Performance evaluation in detection-level.

Results	Identification-Level							
	TOCFL				HSK			
	Acc	Pr	Re	F1	Acc	Pr	Re	F1
Run1	0.2211	0.1588	0.3196	0.2824	0.3485	0.2800	0.3879	0.3252
Run2	0.2322	0.1675	0.3136	0.2122	0.3092	0.2681	0.4565	0.3378
Run3	0.4023	0.2810	0.1359	0.2184	0.4306	0.2886	0.1448	0.1928
Average	0.2852	0.2024	0.2564	0.2377	0.3628	0.2789	0.3297	0.2853

Table 4: Performance evaluation in identification-level

Results	Position-Level							
	TOCFL				HSK			
	Acc	Pr	Re	F1	Acc	Pr	Re	F1
Run1	0.0886	0.0002	0.0002	0.0002	0.0654	0.0024	0.0062	0.0035
Run2	0.0991	0	0	null	0.0373	0.0022	0.007	0.0034
Run3	0.2797	0.0012	0.0005	0.0007	0.2701	0.001	0.0005	0.0007
Average	0.1558	0.0005	0.0005	0.0005	0.1243	0.0019	0.0046	0.0025

Table 5: Performance evaluation in position-level

Grammatical Error Diagnosis. For capturing both semantic and syntactic information, we proposed the use of single word embedding as input of CNN and LSTM, which is similar to character-level embedding in English. In system evaluation, the recall and f1-score of our submitted results Run1 of the TOCFL testing data ranked the fourth place in all submissions in detection-level.

By participating in this shared task for CGED, we have made a preliminary study in this area. The future work will focus on improving the accuracy of our models.

Acknowledgements

This work is supported by The Natural Science Foundation of Yunnan Province (Nos. 2013FB010).

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *arXiv*: 1607.04606v1 [cs.CL].

Christopher J. C. Burges. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121-167

Po-Lin Chen, Shih-Hung Wu, Liang-Pu Chen, and Ping-Che Yang. 2015. Chinese Grammatical Error Diagnosis by Conditional Random Fields. In *Proceedings of The 2nd Workshop on Natural Language Processing Techniques for Educational Applications*, pp: 7–14.

Shuk-Man Cheng, Chi-Hsin Yu, and Hsin-Hsi Chen. 2014. Chinese Word Ordering Errors Detection and Correction for Non-Native Chinese Language Learners. In *Proceedings of The 25th International Conference on Computational Linguistics: Technical Papers*, pp: 279-289

Sepp Hochreiter, and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2010. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pp: 282-289

Bernhard E. LeCun, John Stewart Denker, Donnie Henderson, Richard E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. 1990. Handwritten digit recognition with a backpropagation network. *Advances in Neural Information Processing Systems*, pp: 396-404

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, pp: 3111-3119.

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.

Ronald J. Williams, and David Zipser. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. 1989. *Neural Computation*,1(2):270-280

Shih-Hung Wu, and Hsien-You Hsieh. 2012. Sentence Parsing with Double Sequential Labeling in Traditional Chinese Parsing Task. In *Proceedings of the Second CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pp: 222–230.

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*, pp:42–47.

Chi-Hsin Yu, and Hsin-Hsi Chen. 2012. Detecting Word Usage Errors in Chinese Sentences f-or Learning Chinese as a Foreign Language. In *Proceedings of The 24th International Conference on Computational Linguistics: Technical Papers*, pp:3003–3018.

Marcos Zampieri, and Liling Tan. 2014. Grammatical Error Detection with Limited Training Data: The Case of Chinese. In *Proceedings of the 1st Workshop on Natural Language Processing Techniques for Educational Applications*, pp: 69-74.