N-gram Model for Chinese Grammatical Error Diagnosis

Jianbo Zhao, Hao Liu, Zuyi Bao, Xiaopeng Bai[1], Si Li, Zhiqing Lin
Beijing University of Posts and Telecommunications, Beijing, China
[1]East China Normal University, Shanghai, China
{zhaojianbo, xiaohao_0033, baozuyi, lisi, linzq}@bupt.edu.cn,
xpbai@zhwx.ecnu.edu.cn

Abstract

Detection and correction of Chinese grammatical errors have been two of major challenges for Chinese automatic grammatical error diagnosis. This paper presents an N-gram model for automatic detection and correction of Chinese grammatical errors in NLPTEA 2017 task. The experiment results show that the proposed method is good at correction of Chinese grammatical errors.

1 Introduction

The goal of the NLPTEA 2017 shared task[1] for Chinese spelling check is to develop a computer-assisted system to automatically diagnose typing errors in Traditional Chinese sentences written by native Hong Kong primary students. Two kinds of errors are defined in the Chinese grammatical error diagnosis of NLPTEA 2017: typo and Cantonese usage. Typical error examples are shown in Table 1. In this NLPTEA task, the given sentences may not be wrong or not less than one error.

Spelling check is a common task for every language. It is an automatic mechanism to detect and correct spelling errors. An automatic spelling check system should have abilities about error detection and error correction. Error detection is to find the various types of spelling errors in the text. And error correction is to replace some inappropriate words and characters by some reasonable ones.

Error Type	Error Sentence
No error	我很喜歡吃媽媽做的涼瓜炒蛋飯。 (I like to eat my mother's rice with balsam pear scrambled eggs)
Typo only	我很喜歡吃媽媽做的梁瓜炒蛋飯。
Cantonese usage only	我很鍾意吃媽媽做的涼瓜炒蛋飯。
Typo and Cantonese usage	我很鍾意吃媽媽做的梁瓜炒蛋飯。
Multiple typos and multiple Cantonese usages	我很鍾意食媽媽做的梁瓜炒旦飯。

Table 1: Typical error examples

With the close connection of mainland China and Hong Kong, it is essential for native Hong Kong primary students, who use Cantonese in their daily life, to learn some grammar and semantics of Mandarin Chinese. Additionally, as primary students, they can not avoid making some spelling mistakes like typos. Therefore, Chinese spelling check is becoming a significant task nowadays.

The same Chinese words express different meanings in different contexts. These are very difficult for beginners to master and challenge the establishment of automatic Chinese detection and correction system. Language modeling

[1] https://www.labviso.com/nlptea2017/

Proceedings of the 4th Workshop on Natural Language Processing Techniques for Educational Applications, pages 39–44,
Taipei, Taiwan, December 1, 2017 ©2017 AFNLP

(LM) (Goodman, 2001) is widely used in Chinese spelling check. The most widely-used and well-practiced language model, by far, is the N-gram LM (Wu et al., 2015), because of its simplicity and fair predictive power. Ensemble Learning (Xiang et al., 2015), CRF (Wu et al., 2015) and LSTM network (Shiue et al., 2017) have also been used in recent years to diagnose Chinese error.

Our work in this paper uses an N-gram LM to detect and correct possible spelling errors. And we also do word segmentation in a pre-processing stage which can improve the system performance. In our model, we first make word and character segmentation of the text. Second, the processed text is used as an input of the N-gram model, then the output K value is used to determine whether the word and character are wrong. If the word and character are wrong, the detection model will output the location and the length of the wrong word and character. Finally, output information of the detection model is used as an input of the correction model. The correction model outputs the correction information by matching and searching in the dictionaries.

This paper is organized as follows：Section 2 describes the N-gram model of the detection system we proposed for this task. Section 3 describes the error correction model we put forward for this task. Section 4 shows the data analysis and the evaluation results. Section 5 concludes this paper and illustrates the future work.

2 A Chinese Error Detection Model Based on N-gram

2.1 Introduction of the N-gram basic model

The N-gram model (Wu et al., 2001) is one of the most common mathematical models in natural language processing. It is defined as: the assumption sequence $W_1W_2 \cdots W_n$ is a Markov chain, and then the probability of an element W_i is only related to the preceding N-1 elements:

$$P(W_i|W_1...W_{i-1}) = P(W_i|W_{i-n+1}...W_{i-1}) \quad (1)$$

Therefore, the N-gram model can be regarded as an N-1 Markov chain. According to Markov stochastic process, the probability of symbol string $S = W_1W_2 \cdots W_n$ can be calculated by the initial probability distribution and the transfer probability as follows:

$$P(S) = P(W_1) \cdot \prod (P(W_k|W_{k-n+1}^{k-1})) \quad (2)$$

where $P(W_1)$ can be considered as an initial probability distribution and $P(W_k|W_{k-n+1}^{k-1})$ can be regarded as a state transition probability. W_{k-n+1}^{k-1} indicates $W_{k-n+1}W_{k-n+2}...W_{k-1}$.

It can be seen that the bigger the N is, the closer the word order is to the real word, which produces better results. However, in practical application, the growth of the N not only causes the number of parameters increases sharply, but also brings some evaluation errors. So in actual use, we only consider the situation when N=1,N=2,N=3, namely Unigram, Bigram and Trigram (Liu et al., 2011).

2.2 A model of word continuous relation judgment

This model is used to determine whether characters or words continue to occur incorrectly, such as a sentence $S = Z_1Z_2 \cdots Z_iZ_{i+1} \cdots Z_m$. Z_iZ_{i+1} are two consecutive characters or words. By using the probability model of two characters or words, we check the target character or word, so as to determine whether the character or word is correct. In other words, if the character or word is correct, it can only be judged by its continuous relationship with the character or word.

Take Bigram as an example, in order to check whether Z_i is error, we just need to check the adjacent relations of Z_{i-1} and Z_i, if Z_{i-1} to Z_i transfer probability $P(Z_i|Z_{i-1})$ meets a certain threshold condition, then we consider Z_{i-1} and Z_i are continuous, otherwise we think Z_{i-1} and Z_i are not continuous, then we consider Z_i is error.

$$P(Z_{i-1}) = \frac{R(Z_{i-1})}{N} \tag{3}$$

$$P(Z_i) = \frac{R(Z_i)}{N} \tag{4}$$

$P(Z_{i-1})$ is the probability of Z_{i-1} in training corpus, and $P(Z_i)$ is the probability of Z_i in training corpus. $R(Z_{i-1})$ represents the number of occurrences of Z_{i-1} in the entire training corpus. $R(Z_i)$ represents the number of occurrences of Z_i in the entire training corpus. N represents the total number of strings in the training corpus.

$$P(Z_{i-1}, Z_i) = \frac{R(Z_{i-1}, Z_i)}{N} \tag{5}$$

$P(Z_{i-1}, Z_i)$ represents the probability of continuity of Z_{i-1} and Z_i. $R(Z_{i-1}, Z_i)$ indicates the total number of consecutive occurrences of Z_{i-1} and Z_i in the training corpus.

So the condition of judging whether Z_{i-1} and Z_i is continuous is $R(Z_{i-1}, Z_i) \geq \tau_0$. If $R(Z_{i-1}, Z_i) \geq \tau_0$ is established, then we consider Z_{i-1} and Z_i are continuous, otherwise we consider Z_i is wrong.

2.3 A model of error detection based on different N-gram models

In this NLPTEA task, we use different N-gram models to determine whether the text is wrong or not. From the above mentioned, we know that model based on N-gram needs to have the frequency of characters and words. Through large corpus, we can construct the Bigram model, the Trigram model of characters and words and characters and words frequency table.

The corpus we use is middle and primary school texts organized by East China Normal University that has been made Chinese word segmentation.

For the Unigram model, we need to count the number of occurrences of each character and word in the corpus. For example, the number of occurrences of W_i is C_i, the probability of W_i is

$$P(W_i) = \frac{C_i}{N} \tag{6}$$

For the Bigram model, we need to count the number of continuous occurrences of two characters and words in the corpus. For example, the number of continuous occurrences of W_i and W_{i-1} is $C_{i-1,i}$.

$$P(W_i|W_{i-1}) = \frac{C_{i-1,i}}{C_{i-1}} \tag{7}$$

For the Trigram model, we need to count the number of continuous occurrences of three characters and words in the corpus. For example, the number of continuous occurrences of W_{i-2}, W_{i-1} and W_i is $C_{i-2,i-1,i}$.

$$P(W_i|W_{i-1}W_{i-2}) = \frac{C_{i-2,i-1,i}}{C_{i-2,i-1}} \tag{8}$$

$$P(W_i|W_{i-1}W_{i-2}) = \frac{C_{i-2,i-1,i}}{C_{i-2,i-1}} \tag{9}$$

So we can get the detection model, including the character model, the word model, the Bigram model of characters and words, the Trigram model of characters and words.

The model of errors detection is shown in Figure 1.

2.4 Examples and parameters

Take the sentence “表演完了，空中的濃煙散開了，回復原來的消晰。” as an example, to check whether there is an error with “回復”. After making Chinese word segmentation, the sentence is “表演/完/了/a/空中/的/濃煙/散/開/了/a/回 復/原 來/的/消 晰/a”. “a” presents a space. Examples of inputs of each model are shown in Table 2.

We use LTP model (Wanxiang Che, 2010) to make Chinese word segmentation. Since LTP

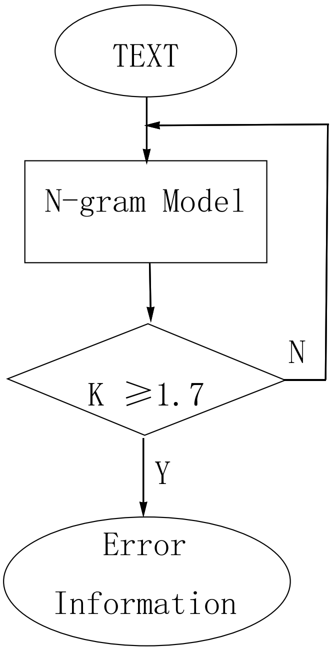

Figure 1: The Model of Errors Detection

is a model to segment simplified Chinese words, we first turn the traditional Chinese into simplified Chinese and then make word segmentation.

The character model: we take the text of characters as inputs and check whether the character exists in the dictionary of characters. If the character does not exist, $K = K + 2$, otherwise, the K value remains unchanged.

The word model: we take the text of words as inputs and check whether the word exists in the dictionary of words. If the word does not exist, $K = K + 2.7$, otherwise, the K value remains unchanged.

The Bigram model of words: we take the text of two consecutive words as inputs and check whether the two consecutive words exist in the dictionary of two consecutive words. If the two words do not exist, $K = K + 0.9$, otherwise, if the number of appearance is less than 3 times, $K = K + 0.2$, otherwise, $K = K - 1.2$.

The Trigram model of words: we take the text of three consecutive words as inputs and check whether the three consecutive words exist in the dictionary of three consecutive words. If the three words do not exist, $K = K + 0.4$, otherwise, $K = K - 1.2$.

The Model	**Input Strings**
The character Model	< 回>
The word Model	< 回復>
The Bigram model of words	<a, 回復> <回復,原來>
The Trigram model of words	< 了, a, 回復> < a,回復,原來> <回復,原來,的>
The Bigram model of characters	< a, 回> <回,復>
The Trigram model of characters	< 了, a,回> < a, 回,復> < 回,復, 原>

Table 2: Inputs of each model

The Bigram model of characters: we take the text of two consecutive characters as inputs and check whether the two consecutive characters exist in the dictionary of two consecutive characters. If the two characters do not exist, $K = K + 1$, otherwise, if the number of appearance is less than 3 times, otherwise, $K = K - 1.5$.

The Trigram model of characters: we take the text of three consecutive characters as inputs and check whether the three consecutive characters exist in the dictionary of three consecutive characters. If the three characters do not exist, $K = K + 0.3$, otherwise, $K = K - 1$.

After the above calculation, we get the K value, length and position of each character. The K value is used to determine whether it is wrong, and $length$ is used to indicate the length of wrong string, and $position$ refers to the start position of the wrong character in the sentence. If the K value is greater than 1.7, we consider the character and the word are wrong. The threshold value is determined by the combined effect of the above model.

Metric	Input Value
TP	88
FP	571
FN	664
Precision	13.3536%
Recall	11.7021%
Performance	12.4734%

Table 3: Detection Performance

Type	Input Value
Performance	90.9102%

Table 4: Correction Performance

3 Chinese Error Correction Model

$Similarprounciation$ and $Similarshape$ (Lee et al., 2015) are two dictionaries which are used to find similar characters and words in pronunciation and shape.

Corresponding to the two dictionaries, $Similarprounciation$ and $Similarshape$, we get the candidate sets SP_w and SS_w of the wrong character and word h_w respectively. SP_w refers to the elements of the set that has the similar pronunciation of h_w and SS_w refers to the elements of the set that has the similar shape of h_w. Then we concatenate SP_w and SS_w into a set called S_w. For $\forall s \in S$, we replace h_w by s into the original sentence, then we use 2-gram, 3-gram and 4-gram around the specific character, and we can collect 9 items for each character of specific position, including 2 items of 2-gram, 3 items of 3-gram and 4 items of 4-gram. We compare and sort the frequency of the 9 items in the word frequency dictionary W. We assume that after replacing, if some items are in dictionaries, the item which has more characters will have a higher probability to be the target choice. For example, the frequency of the item “和蔼” is 5, the frequency of the item “和蔼可亲” is 2. But if the second one appears in your candidate sets, it will have a higher probability than the first one as we can imagine, so we can distribute different proportions to different types in dictionaries. Finally the most probable character is selected for eventual replacement.

Type	Input Value
Performance	21.9370%

Table 5: Overall Performance

When $length$ is higher than 1, we should replace the character from the start position to end position. End position is the plus of $position$ and $length$. Therefore, there should be multiple characters to be replaced at the same time, such as when $length$ is equal to 3, we replace the character in the $position$, the second and the third character that begin with $position$, all these three characters need to be replaced at the same time successively. Then we compare the frequency of all items. The comparison method is as above.

4 Result Analysis

The system results we obtained are shown in Table 3, Table 4 and Table 5. We can see from the results that the detection performance is not very well. The reason may be that the parameters of the complex N-gram model are not easy to control and to adjust. The results also show that the method we proposed is good at correction of Chinese grammatical errors and achieves a high accuracy.

5 Conclusion and Future Work

In this paper, we present an N-gram model for automatic detection and correction of Chinese grammatical errors. As we can see from the results, the performance of correction of Chinese grammatical errors is pretty good. In the future, we plan to employ neural network to Chinese grammatical error diagnosis.

6 Acknowledgements

This work was supported by Beijing Natural Science Foundation (4174098), National Natural Science Foundation of China (61702047) and the Fundamental Research Funds for the Central Universities (2017RC02).

References

Joshua T. Goodman. 2001. A bit of progress in language modeling. *Computer Speech & Language* 15(4):403–434.

Lung Hao Lee, Liang Chih Yu, and Li Ping Chang. 2015. Overview of the nlp-tea 2015 shared task for chinese grammatical error diagnosis. In *Overview of the NLP-TEA 2015 Shared Task for Chinese Grammatical Error Diagnosis*.

C. L. Liu, M. H. Lai, K. W. Tien, Y. H. Chuang, S. H. Wu, and C. Y. Lee. 2011. Visually and phonologically similar characters in incorrect chinese words:analyses, identification, and applications. *Acm Transactions on Asian Language Information Processing* 10(2):1–39.

Yow Ting Shiue, Hen Hsen Huang, and Hsin Hsi Chen. 2017. Detection of chinese word usage errors for non-native chinese learners with bidirectional lstm. In *Meeting of the Association for Computational Linguistics*. pages 404–410.

Zhenghua Li Ting Liu Wanxiang Che. 2010. Ltp: A chinese language technology platform. In *Coling 2010:Demonstrations*. pages 13–16.

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

Y. Wu, G. Wei, and H. Li. 2001. Word segmentation algorithm for chinese language based on n-gram models and machine learning. *Journal of Electronics & Information Technology* .

Yang Xiang, Xiaolong Wang, Wenying Han, and Qinghua Hong. 2015. Chinese grammatical error diagnosis using ensemble learning. In *The Workshop on Natural Language Processing Techniques for Educational Applications*. pages 99–104.