

Contextualized Character Representation for Chinese Grammatical Error Diagnosis

Jianbo Zhao, Si Li, Zhiqing Lin

Beijing University of Posts and Telecommunications
No.10 Xitucheng Road, Haidian District, Beijing, China
{zhaojianbo, lisi, linzq}@bupt.edu.cn

Abstract

Nowadays, more and more people are learning Chinese as their second language. Establishing an automatic diagnosis system for Chinese grammatical error has become an important challenge. In this paper, we propose a Chinese grammatical error diagnosis (CGED) model with contextualized character representation. Compared to the traditional model using LSTM (Long-Short Term Memory), our model have better performance and there is no need to add too many artificial features.

1 Introduction

With the rapid development of China, more and more non-native Chinese speakers begin to learn Chinese. Writing is a very important part of Chinese learning. However, there are some differences between Chinese and English, such as no changes in tense in Chinese, which makes it difficult for many Chinese learners to find their own mistakes in writing. Traditional Chinese learning methods cost a lot of labor and time, so it is very important to establish an automatic diagnosis system for Chinese grammatical error. This is also the purpose of this shared task.

The task of CGED2018¹ is to automatically diagnose grammatical errors in Chinese sentences written by second language learners. The errors include four types, redundant words (denoted as a capital "R"), missing words ("M"), word selection errors ("S") and word ordering errors ("W"). Table 1 shows examples of errors. The CGED system needs to detect the location of errors and gives the type of each error. For error typed S and M, the model can give at most three correct candidates.

¹<https://sites.google.com/view/nlptea2018/shared-task>

| Error Type | Error Sentence | Correct Sentence |
|------------|----------------|---|
| R | 大家合作无间而救了那位病人。 | 大家合作无间救了那位病人。(We saved that patient cooperatively.) |
| M | 千万别人家添麻烦。 | 千万别给人家添麻烦。(Don't bother others.) |
| S | 怎样把孩子教养成人呢? | 怎样把孩子抚养成人呢?(How to raise the child?) |
| W | 我有强度高的思维能力。 | 我有高强度的思维能力。(I have high-intensity thinking skills.) |

Table 1: Examples of each error types

In this paper, we regard CGED task as a sequence labeling problem(Zheng et al., 2016) and propose a CGED model with contextualized character representation. This model have better considered the different semantics of words in Chinese texts. The experiment results show that our model have better result compared to the baseline without artificial features.

2 Contextualized Character Representation

2.1 Character Embedding

Words are the smallest unit of semantic expressions in Chinese texts. In different contexts, the same words may express different meanings. Also, the same situation exists for single characters. For example, the character "打" in word "一打" (a dozen) means dozen, in word "打鼓" (play the

drum) means play. Therefore, we use the same character vector to represent the same character in different contexts is inaccurate, and sometimes there may be a big semantic deviation. To address this issue, we propose to use the contextualized character representation for CGED to solve the ambiguity problem.

2.2 Building Contextualized Character Representation

(Choi et al., 2016) puts forward that each dimension of a word vector may represent some semantic information of the word. But in different texts, the semantic information we need to use is different, so we need to ignore the unneeded semantic information. That is to say, under the different context conditions, we need to mask out some dimensions of the word embedding vectors. We take advantage of this method proposed in (Choi et al., 2016) for our model.

$$T = \frac{1}{M} \sum_{t=1}^M NN_{\xi}(x_t) \quad (1)$$

where x_t in our work represents the character representation in each time step. T represents the text representation. M is the max sequence length for the sentence. $NN_{\xi}: \mathbf{R}^{C_E} \rightarrow \mathbf{R}^{T_E}$ is a feedforward neural network parametrized by ξ . C_E is the character embedding size and T_E is the text representation size.

Then we use T to calculate the contextualized character vectors as input of traditional sequence labeling model of LSTM instead of the traditional character vectors.

$$mask = \sigma(W_m T + b_m) \quad (2)$$

$$x_t \leftarrow x_t \odot mask \quad (3)$$

where σ is the sigmoid activation function to control the output between 0 to 1. W_m is the weight of calculating $mask$ and b_m is the bias. \odot is an element-wise multiplication.

We use the $mask$ to get the contextualized character representation which can better represent the meaning of characters and better obtain the information we need in the text.

| The number of sentences | The number of characters | The average number of characters |
|-------------------------|--------------------------|----------------------------------|
| 402 | 19382 | 48.21 |

Table 2: Information of training set: The average number of characters represent the average number of characters of one sentence.

| Error type | The number of error characters | The proportion of error characters in training set |
|------------|--------------------------------|--|
| R | 281 | 1.44% |
| M | 298 | 1.53% |
| S | 797 | 4.11% |
| W | 493 | 2.54% |
| all errors | 1869 | 9.64% |

Table 3: The number of errors in training set

3 Function of Save Model and Loss Function

3.1 Error Sparse Problem

In the given Chinese text, we find that a relatively long sentence may only contains one or two errors. Although one sentence may contain multiple errors but the number of errors is insufficient. In Table 2 and 3, we give the number of errors in CGED2018.

After dividing the errors into four categories, it can be seen that due to the small number of errors, it may not be conducive to the training of the model.

3.2 Function of Save Model

We use the traditional training method, accuracy, to train our model. However, when the development set has reached the greatest accuracy, the output of the model in test set is not good. Analyzing the result, we see that the model learns the correct part more, and learns the error information less. The model discriminates most of test sentences to be correct. Therefore, we propose to save the model no longer when the development set achieves the max accuracy, but when Eq. 4 is max in development set.

$$f_s = \frac{\sum_n \sum_i c_{ni}}{\sum_n \sum_i e_{ni}} \quad (4)$$

$$c_{ni} = \begin{cases} 1 & p_{ni} = y_{ni} \text{ and } (p_{ni} \neq 0 \text{ or } y_{ni} \neq 0) \\ 0 & \text{others} \end{cases} \quad (5)$$

$$e_{ni} = \begin{cases} 1 & \text{others} \\ 0 & p_{ni} = 0 \text{ and } y_{ni} = 0 \end{cases} \quad (6)$$

where p represents the output label of the model of a character and y represents the ground-truth label.

The significance of Eq. 4 is that when we save the model, we expect the model to detect more wrong information and ignore some correct information. The model can capture more error information when there are fewer errors in the sentence.

3.3 Loss Function

Although the model can detect more error information but it is not enough, when we use Eq. 4 to save the model. From the table 6, 7, 8, 9, it can also be seen that although the results have improved but the increase is limited.

In the traditional LSTM model of sequence labelling, the cross-entropy loss function, Eq. 7, is generally used as its loss function.

$$loss_1 = -\frac{1}{n} \sum_x [y \ln a + (1 - y) \ln(1 - a)] \quad (7)$$

However, the problem that the number of correct characters in the dataset is much larger than the number of incorrect characters still exists. Therefore, the training of the model may have some problem. To address this issue, we add a loss function Eq. 8 to $loss_1$.

$$loss_2 = -\frac{1}{n} \sum_x [y_m \ln a_m + (1 - y_m) \ln(1 - a_m)] \quad (8)$$

$$y_m = mask_r(y) \quad (9)$$

$$a_m = mask_r(a) \quad (10)$$

where we use $mask_r$ to keep the correct place in the training tag, forcing the model to capture more error information. The overall loss function is Eq. 11.

$$loss = (1 - \gamma)loss_1 + \gamma loss_2 \quad (11)$$

| Segmented sentence | 2-gram | 3-gram |
|-------------------------|-------------------|-------------------------|
| 我/自己/本身/偏好/于/男女/混合式/教育。 | <本身 偏好> <偏好 于> | <自己 本身 偏好> <偏好 于 男女> |

Table 4: Example of n-gram

where γ is a weight, indicates the importance of the error information that needs to be retained, and can be adjusted according to different tasks.

4 Correction System

Correct system we use in our model is the method proposed in (Chen et al., 2016). Since we mainly deal with the detection problem, we have simplified the method in (Chen et al., 2016) and only put forward one candidate correction.

(Chen et al., 2016) uses the method of calculating the n-gram score of each word to judge whether the word is correct or not and put forward correct candidates. If the original word has the highest score, the original word is considered to be correct. If the candidate word has a higher score than the original word, the original word is considered to be wrong. The candidate word with highest scoring is regarded as the correction.

$$SL(S) = \sum_n (n \times \sum_{u \in SubStr(S, n)} \log(gsf(u))) \quad (12)$$

Eq. 12 gives the equation of length-weighted string log-frequency score $SL(S)$. Where S represents the sentence after word segmentation or character segmentation. $SubStr(S, n)$ represent all substring of sentence S with n words or characters. $gsf(\cdot)$ is the frequency of u . Obviously, matching a higher gram is more welcome than a lower gram. To increase the accuracy of correction, (Chen et al., 2016) adds weights to the different n-gram by their length to favor higher gram.

We use this score for errors typed S. In order to reduce the amount of calculation, we only keep the calculation of 2-gram and 3-gram, the example of n-gram of words is shown in table 4.

For the error which is typed with S is a word, we will calculate the SL score of the word. We use the dictionaries of characters with similar pronunciation and similar shape in (Wu et al., 2013) and convert characters into simplified Chinese². We merged the two dictionaries to one dictionary of candidates for characters. When we choose the word to replace, we prefer to select the word that have only one character different from the original word. We replace each characters in the words and calculate the score separately. We select the candidate word with the highest score as the correct one.

For the error which is typed with S is a character, we calculate the SL score for the character. The candidate dictionary is directly used to replace the character and the score is calculated. The character with the highest score is considered to be correct.

For the error typed with M, we also use SL to calculate the score using 2-gram and 3-gram. We first search the words in the word dictionary which have the same character as the character labeled M. Then, calculate the candidates' score. We regard that the candidate with the highest score is the correct candidate.

5 Evaluation

5.1 Baseline

In this experiment, we build the Bi-LSTM model for sequence labelling as our baseline model. Unlike traditional sequence labeling, Chinese grammatical error diagnosis may result in inaccurate word segmentation due to existing errors, so we use character embeddings to replace word embeddings.

5.2 Hyper-parameter and Data

We use word2vec³ to pretrain our character embeddings by wiki corpus⁴. We also use wiki corpus to build our n-gram dictionaries. The character embedding size is 400, the hidden units of Bi-LSTM is 256. We set the batch size is 32. We use Adam optimizer to train our model and the learning rate is 0.001.

The training data we use comes from NLPTEA2016 and NLPTEA2018 and we di-

²<http://zh.wikipedia.org/wiki/Wikipedia:繁简处理>

³<https://code.google.com/archive/p/word2vec>

⁴<https://dumps.wikimedia.org/zhwiki/latest/zhwiki-latest-pages-articles.xml.bz2>

| | Train | Dev | Test 2016 | Test 2017 |
|--------------------|--------|--------|--------------|--------------|
| Num _{sen} | 7602 | 2402 | 3011 | 3154 |
| Num _c | 349230 | 112617 | 150826 | 141973 |
| Num _{ec} | 32117 | 10633 | 6680 | 8508 |

Table 5: Information of training data: Num_{sen} means the total number of sentences in each set. Num_c means the total number of characters. Num_{ec} represent the total number of error characters.

vide part of data from NLPTEA2016 to the development set. We use two test set from NLPTEA2016 and NLPTEA2017. Table 5 shows the data information in detail.

5.3 Evaluation Method

According to (Lee et al., 2016), the evaluation method includes three levels, detection level, identification level, position level. And this year add correction level.

Detection level: Determines whether a sentence is correct or not. If there is an error, the sentence is incorrect. In other words, the sentences are classified into two categories.

Identification level: The correct situation should be exactly the same as the gold standard for a given type of error. This can be considered as a multi-classification problem.

Position level: The system results should be perfectly identical with the quadruples of the gold standard.

Correction level: Characters marked as S and M need to give correct candidates. The model can recommend at most 3 correction at each error.

The following metrics are measured at detection, identification, position-level.

$$FalsePositiveRate = \frac{FP}{FP + TN} \quad (13)$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (14)$$

$$Precision = \frac{TP}{TP + FP} \quad (15)$$

$$Recall = \frac{TP}{TP + FN} \quad (16)$$

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (17)$$

| | False Positive Rate |
|----------------------|---------------------|
| Bi-LSTM | 0.0136 |
| Bi-LSTM+FS | 0.0884 |
| Bi-LSTM+loss_0.5 | 0.2073 |
| Bi-LSTM+FS+loss_0.5 | 0.9831 |
| Bi-LSTM+FS+loss_0.4 | 0.9519 |
| Bi-LSTM+FS+loss_0.3 | 0.8571 |
| Bi-LSTM+FS+loss_0.2 | 0.5491 |
| Bi-LSTM+FS+loss_0.1 | 0.3028 |
| Bi-LSTM+FS+loss_0.05 | 0.1832 |
| Bi-LSTM+mask | 0.7057 |
| Bi-LSTM+mask+POS | 0.7596 |

Table 6: Result on false positive rate: FS represents model that save model with new function and loss means the model with new loss function and the number after “_” is γ .

6 Result

In this part, we show our experiment results in the CGED2016 test set. Since the experiment results are similar on CGED2017 dataset, they are not given.

The first part of table 6, 7, 8, 9 shows the results of the comparison between the model using new function to save model, with the reconstruction loss function and the original model. The γ of the model with reconstructive loss is set to 0.5. It can be seen from the experiment that modifying the save function and rebuilding the loss function all have a good improvement on the error detection of the model. The results of mixing the above methods are also given. There is an improvement in error detection, but too many errors are detected and the correct information is ignored. So after that we modify the value of the weight γ in Eq. 11 to get more reasonable model.

The second part of table 6, 7, 8, 9 shows the different models with new function to save model and reconstructive loss for modifying the value of γ in Eq. 11. It can be seen that when the weight decreases, the false positive rate decreases significantly, which indicates that the model captures more correct information. When γ is 0.2 or 0.1 is more suitable for our task. When the weight is too large, false positive rate is too large indicates that the error is not detected, which is not consistent with the objectives of this task. At 0.05, the F1 values of all levels are too low, so we use 0.1 as the weight in the following experiments.

The third part of table 6, 7, 8, 9 shows the ex-

| | Acc | Pre | Re | F ₁ |
|------------|--------|--------|--------|----------------|
| Bi-LSTM | 0.5111 | 0.5 | 0.0143 | 0.0277 |
| Bi-LSTM+FS | 0.5064 | 0.4729 | 0.0829 | 0.141 |
| Bi-LSTM | 0.5065 | 0.4888 | 0.2072 | 0.291 |
| +loss_0.5 | | | | |
| Bi-LSTM+FS | 0.4902 | 0.4894 | 0.9851 | 0.6539 |
| +loss_0.5 | | | | |
| Bi-LSTM+FS | 0.4829 | 0.4851 | 0.9375 | 0.6393 |
| +loss_0.4 | | | | |
| Bi-LSTM+FS | 0.4839 | 0.484 | 0.8404 | 0.6142 |
| +loss_0.3 | | | | |
| Bi-LSTM+FS | 0.4909 | 0.4813 | 0.5326 | 0.5056 |
| +loss_0.2 | | | | |
| Bi-LSTM+FS | 0.5005 | 0.4822 | 0.2948 | 0.3659 |
| +loss_0.1 | | | | |
| Bi-LSTM+FS | 0.5148 | 0.5096 | 0.199 | 0.2863 |
| +loss_0.05 | | | | |
| Bi-LSTM | 0.4899 | 0.4848 | 0.6943 | 0.5709 |
| +mask | | | | |
| Bi-LSTM | 0.5015 | 0.4937 | 0.7745 | 0.603 |
| +mask+POS | | | | |

Table 7: Results on detection level: ACC represents accuracy. Pre means precision. Re is recall.

perimental results of our proposed model with new save function and reconstructive loss. γ is set to 0.1. The results from F1 show that the proposed model is improved compared to the baseline model. The model can also detect error information very well without artificial features. We also tried to add artificial information to the model to improve the experimental results, so we added POS (Part of Speech) information. Since we are dealing with characters, so we use POS for the character’s corresponding word as the character’s POS. It can be seen that POS is useful in Chinese error detection. For errors, POS may provide some information to help the model detect better.

Table 10, 11, 12, 13 shows the experiment results we submitted in CGED2018 in detection part. Table 14 show the results in CGED2018 in correction part. Since our model only proposes one candidate, the results on Correction and Top3 Correction are the same.

7 Related Work

Chinese grammatical error diagnosis task has been developed for a long time. From the initial statistical methods to the current machine learning, more and more attention has been paid to.

| | Acc | Pre | Re | F ₁ |
|------------|--------|--------|--------|----------------|
| Bi-LSTM | 0.5098 | 0.4048 | 0.0068 | 0.0134 |
| Bi-LSTM+FS | 0.4982 | 0.3788 | 0.0401 | 0.0725 |
| Bi-LSTM | 0.4646 | 0.3039 | 0.0842 | 0.1419 |
| +loss_0.5 | | | | |
| Bi-LSTM+FS | 0.2557 | 0.2536 | 0.6919 | 0.3712 |
| +loss_0.5 | | | | |
| Bi-LSTM+FS | 0.2772 | 0.2713 | 0.5239 | 0.3575 |
| +loss_0.4 | | | | |
| Bi-LSTM+FS | 0.3064 | 0.2848 | 0.4256 | 0.3412 |
| +loss_0.3 | | | | |
| Bi-LSTM+FS | 0.4122 | 0.3431 | 0.2479 | 0.2878 |
| +loss_0.2 | | | | |
| Bi-LSTM+FS | 0.4579 | 0.349 | 0.1368 | 0.1965 |
| +loss_0.1 | | | | |
| Bi-LSTM+FS | 0.488 | 0.3735 | 0.0895 | 0.1443 |
| +loss_0.05 | | | | |
| Bi-LSTM | 0.361 | 0.3167 | 0.365 | 0.3392 |
| +mask | | | | |
| Bi-LSTM | 0.3752 | 0.3451 | 0.4902 | 0.405 |
| +mask+pos | | | | |

Table 8: Results on identification level: ACC represents accuracy. Pre means precision. Re is recall.

(Zhang et al., 2000) searched the optimal string from all possible derivation of the input sentence using operations of character substitution, insertion, and deletion with a traditional word 3-gram language model. (Chen et al., 2013) still used n-gram as the main method, and added Web resources to improve detection results. (Lin and Chu, 2015) used n-gram to establish a scoring system to better give correction options. (Yeh et al., 2017) based on n-gram used the KMP algorithm to speed up the search for correct candidates.

Due to the continuous rise of machine learning in recent years, the field of natural language processing is increasingly turning to machine learning. In the past few years, the diagnosis of Chinese grammatical errors has also been developing in machine learning. Grammatical error detection is usually considered as the sequence labeling task (Zheng et al., 2016). (Huang and WANG, 2016) used Bi-LSTM to annotate the errors in the sentence. (Shiue et al., 2017) combined machine learning with traditional n-gram methods, using Bi-LSTM to detect the location of errors and adding additional linguistic information, POS, n-gram. (Li et al., 2017) used Bi-LSTM to generate

| | Acc | Pre | Re | F ₁ |
|------------|--------|--------|--------|----------------|
| Bi-LSTM | 0.5041 | 0.0227 | 0.0003 | 0.0005 |
| Bi-LSTM+FS | 0.4643 | 0.0528 | 0.0043 | 0.008 |
| Bi-LSTM | 0.3746 | 0.0101 | 0.0024 | 0.0039 |
| +loss_0.5 | | | | |
| Bi-LSTM+FS | 0.0239 | 0.0227 | 0.1256 | 0.0385 |
| +loss_0.5 | | | | |
| Bi-LSTM+FS | 0.0319 | 0.0261 | 0.0877 | 0.0402 |
| +loss_0.4 | | | | |
| Bi-LSTM+FS | 0.0553 | 0.0297 | 0.0612 | 0.04 |
| +loss_0.3 | | | | |
| Bi-LSTM+FS | 0.188 | 0.0346 | 0.0265 | 0.03 |
| +loss_0.2 | | | | |
| Bi-LSTM+FS | 0.3447 | 0.0648 | 0.0217 | 0.0325 |
| +loss_0.1 | | | | |
| Bi-LSTM+FS | 0.4107 | 0.0517 | 0.01 | 0.0168 |
| +loss_0.05 | | | | |
| Bi-LSTM | 0.1323 | 0.0582 | 0.0709 | 0.0639 |
| +mask | | | | |
| Bi-LSTM | 0.1965 | 0.1217 | 0.1729 | 0.1429 |
| +mask+pos | | | | |

Table 9: Results on position level: ACC represents accuracy. Pre means precision. Re is recall

| | False Positive Rate |
|------------------|---------------------|
| Bi-LSTM+mask | 0.5029 |
| Bi-LSTM+mask+POS | 0.5480 |

Table 10: Result on false positive rate in CGED2018

the probability of each characters, and used two s-strategies to decide whether a character is correct or not. (Liao et al., 2017) used the LSTM+CRF model to detect dependencies between outputs to better detect error messages. (yang et al., 2017) added more linguistic information on LSTM+CRF model, such as POS, n-gram, PMI score and dependency features.

8 Conclusion

As more and more people learn Chinese, the automatic diagnosis of Chinese grammatical error becomes more and more important. This paper proposes a contextualized character representation for CGED and related solutions for the error s-parse problem, which are improved compared to the baseline approach.

In the future, we will add this contextualized character representation to models that are better at Chinese grammatical error diagnosis such as Bi-

| | Acc | Pre | Re | F ₁ |
|-----------|--------|--------|--------|----------------|
| Bi-LSTM | 0.6005 | 0.6331 | 0.6809 | 0.6562 |
| +mask | | | | |
| Bi-LSTM | 0.6236 | 0.6377 | 0.7584 | 0.6929 |
| +mask+POS | | | | |

Table 11: Results on detection level in CGED2018: ACC represents accuracy. Pre means precision. Re is recall.

| | Pre | Re | F ₁ |
|-----------|--------|--------|----------------|
| Bi-LSTM | 0.4134 | 0.3519 | 0.3802 |
| +mask | | | |
| Bi-LSTM | 0.4084 | 0.4161 | 4122 |
| +mask+pos | | | |

Table 12: Results on identification level in CGED2018: Pre means precision. Re is recall.

LSTM+CRF and consider better correction methods.

Acknowledge

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

References

- Kuan-Yu Chen, Hung-Shin Lee, Chung-Han Lee, Hsin-Min Wang, and Hsin-Hsi Chen. 2013. [A study of language modeling for chinese spelling check](#). In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, pages 79–83. Asian Federation of Natural Language Processing.
- Shao-Heng Chen, Yu-Lin Tsai, and Chuan-Jie Lin. 2016. [Generating and scoring correction candidates in chinese grammatical error diagnosis](#). In *Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016)*, pages 131–139. The COLING 2016 Organizing Committee.
- Heeyoul Choi, Kyunghyun Cho, and Yoshua Bengio. 2016. [Context-dependent word representation for neural machine translation](#). *CoRR*, abs/1607.00578.
- Shen Huang and Houfeng WANG. 2016. [Bi-lstm neural networks for chinese grammatical error diagnosis](#). In *Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016)*, pages 148–154, Osaka, Japan. The COLING 2016 Organizing Committee.

| | Pre | Re | F ₁ |
|-----------|--------|--------|----------------|
| Bi-LSTM | 0.0608 | 0.0504 | 0.0551 |
| +mask | | | |
| Bi-LSTM | 0.0630 | 0.0609 | 0.0620 |
| +mask+pos | | | |

Table 13: Results on position level in CGED2018: Pre means precision. Re is recall.

| | Pre | Re | F ₁ |
|-----------|-------|-------|----------------|
| Bi-LSTM | 0.33% | 0.28% | 0.30% |
| +mask | | | |
| Bi-LSTM | 0.92% | 0.87% | 0.90% |
| +mask+pos | | | |

Table 14: Results on Correction Part in CGED2018: Pre means precision. Re is recall.

Lung-Hao Lee, Gaoqi RAO, Liang-Chih Yu, Endong XUN, Baolin Zhang, 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 (NLPTEA2016)*, pages 40–48. The COLING 2016 Organizing Committee.

Xian Li, Peng Wang, Suixue Wang, Guanyu Jiang, and Tianyuan You. 2017. [Cvte at ijcnlp-2017 task 1: Character checking system for chinese grammatical error diagnosis task](#). In *Proceedings of the IJCNLP 2017, Shared Tasks*, pages 78–83, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Quanlei Liao, Jin Wang, Jinnan Yang, and Xuejie Zhang. 2017. [Ynu-hpcc at ijcnlp-2017 task 1: Chinese grammatical error diagnosis using a bi-directional lstm-crf model](#). In *Proceedings of the IJCNLP 2017, Shared Tasks*, pages 73–77, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Chuan-Jie Lin and Wei-Cheng Chu. 2015. [A study on chinese spelling check using confusion sets and n-gram statistics](#). In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 20, Number 1, June 2015-Special Issue on Chinese as a Foreign Language*.

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 *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 404–410, Vancouver, Canada. Association for Computational Linguistics.

Shih-Hung Wu, Chao-Lin Liu, and Lung-Hao Lee. 2013. [Chinese spelling check evaluation at sighthan bake-off 2013](#). In *Proceedings of the Seventh*

SIGHAN Workshop on Chinese Language Processing, pages 35–42, Nagoya, Japan. Asian Federation of Natural Language Processing.

Yi yang, Pengjun Xie, Jun tao, Guangwei xu, Linlin li, and Si lu. 2017. [Alibaba at ijcnlp-2017 task 1: Embedding grammatical features into lstms for chinese grammatical error diagnosis task](#). In *Proceedings of the IJCNLP 2017, Shared Tasks*, pages 41–46, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Jui-Feng Yeh, Li-Ting Chang, Chan-Yi Liu, and Tsung-Wei Hsu. 2017. [Chinese spelling check based on n-gram and string matching algorithm](#). In *Proceedings of the 4th Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA 2017)*, pages 35–38, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Lei Zhang, Changning Huang, Ming Zhou, and Haihua Pan. 2000. Automatic detecting/correcting errors in chinese text by an approximate word-matching algorithm. In *Meeting on Association for Computational Linguistics*, pages 248–254.

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 (NLPTEA2016)*, pages 49–56, Osaka, Japan. The COLING 2016 Organizing Committee.