Confusionset-guided Pointer Networks for Chinese Spelling Check

Dingmin Wang[♣], **Yi Tay**[♠], **Li Zhong**[♣]
[♣]Tencent Cloud AI
[♠] Nanyang Technological University, Singapore
wangdimmy@gmail.com ytay017@e.ntu.edu.sg reggiezhong@tencent.com

Abstract

This paper proposes Confusionset-guided Pointer Networks for Chinese Spell Check (CSC) task. More concretely, our approach utilizes the off-the-shelf confusionset for guiding the character generation. To this end, our novel Seq2Seq model jointly learns to copy a correct character from an input sentence through a pointer network, or generate a character from the confusionset rather than the entire vocabulary. We conduct experiments on three human-annotated datasets, and results demonstrate that our proposed generative model outperforms all competitor models by a large margin of up to 20% F1 score, achieving state-of-the-art performance on three datasets.

1 Introduction

In our everyday writing, there exists different types of errors, one of which that frequently occurs is misspelling a character due to the characters' similarity in terms of sound, shape, and/or meaning. Spelling check is a task to detect and correct such problematic usage of language. Although these tools been useful, detecting and fixing errors in natural language, especially in Chinese, remains far from solved. Notably, Chinese is very different from other alphabetical languages (e.g., English). First, there are no word delimiters between the Chinese words. Second, the error detection task is difficult due to its context-sensitive nature, i.e., errors can be only often determined at phrase/sentence level and not at character-level.

In this paper, we propose a novel neural architecture for the Chinese Spelling Check (CSC) task. For the task at hand, it is intuitive that the generated sentence and the input sentence would usually share most characters, along with same sentence structure with a slight exception for several incorrect characters. This is unlike other generative tasks (e.g., neural machine translation or dialog translation) in which the output would differ greatly from the input.

To this end, this paper proposes a novel Confusionset-guided copy mechanism which achieves significant performance gain over competitor approaches. Copy mechanisms (Gulcehre et al., 2016), enable the copying of words directly from the input via pointing, providing an extremely appropriate inductive bias for the CSC task. More concretely, our model jointly learns the selection of appropriate characters to copy or to generate a correct character from the vocabulary when an incorrect character occurs. The clear novelty of our work, however, is the infusion of Confusionsets[1] with Pointer Networks, which help reduce the search space and vastly improve the probability of generating correct characters. Experimental results on three benchmark datasets demonstrate that our model outperforms all competitor models, obtaining performance gains of up to 20%.

2 Our Proposed Model

Given an input, we represent the input sentence as $X = \{c_1^s, c_2^s, \cdots, c_n^s\}$, where c_i is a Chinese character[2] and n is the number of characters. We map X to an output sentence $Y = \{c_1^t, c_2^t, \cdots, c_n^t\}$, namely maximizing the probability $P(Y|X)$. Our model consists of an encoder and a decoder similar to (Sutskever et al., 2014), as shown in Figure 1. The encoder maps X to a higher-level representation with a bidirectional BiLSTM architecture similar to that of (Hochreiter and Schmidhuber, 1997). The decoder is also a recurrent neural

[1]Confusionsets are a lexicon of commonly confused characters. Details are deferred to later sections.

[2]In Chinese, there is no explicit delimiter between words and one word usually consists of two or more characters, e.g., 中国 (China) as a word consists of two characters: 中 and 国. In this paper, we use c and w to denote Chinese word and Chinese character, respectively.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5780–5785
Florence, Italy, July 28 - August 2, 2019.

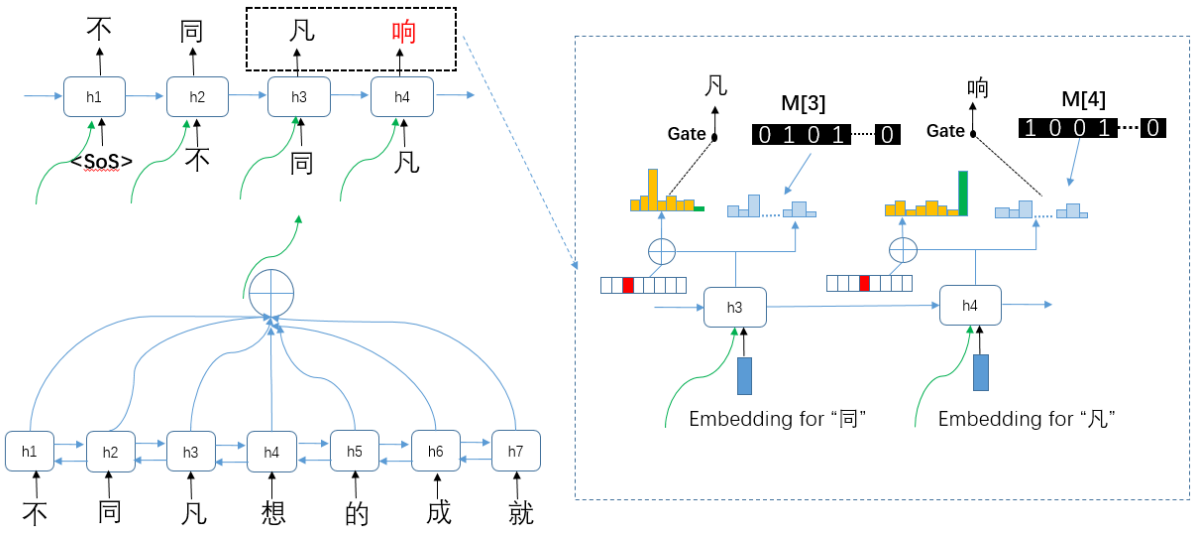

Figure 1: Structure of Confusionset-guided Pointer Network with for Chinese Spelling Check.

network with the attention mechanism (Bahdanau et al., 2014) to attend to the encoded representation and generate Y one character at a time. In our setting, the length of Y is limited to be equal to the length of X.

Confusionset M Confusionset, a prepared set which consists of commonly confused characters plays a key role in spelling error detection and correction. Most Chinese characters have similar characters in shape or pronunciation. According to the statistic result of incorrect Chinese characters collected from the Internet (Liu et al., 2010), 83% of these errors were related to phonological similarity, and 48% of them were related to visual similarity between the involved characters. To reduce the searching space while ensuring that the target characters are not excluded, we build a confusionset matrix $M \in \mathbb{R}^{n*w}$, where w is the size of the vocabulary, n corresponds to the number of characters in X, in which each element is 0 or 1. Take an input "这使我永生难望" as an example, the 7-th character "望" is a spelling error and its confusion set [3] is "汪圣忘晚往完万网···". In $M[7]$, the locations these confusion words occur in will be set to be 1 and the left are set to be 0.

2.1 Encoder

Before diving into the model, we first give a character-level reasoning. Consider the characteristic of Chinese characters, in which there is no explicit delimiter between words like some alphabetic-based languages, i.e., English, so our neural network model operates at the character level. One of reasons is that even for the state-of-the-art word segmenter, there exists some segmenting errors , and texts with spelling errors will exacerbate this phenomenon. Incorrectly segmented results might influence the capture of semantic representation in X for the encoder.

The encoder reads X and outputs a sequence of vectors, associated with each word in the sentence, which will be selectively accessed during decoding via a soft attentional mechanism. We use a bidirectional LSTM network to obtain the hidden states h_i^s for each time step i,

$$h_i^s = \text{BiLSTM}(h_{i-1}^s, e_i^s) \quad (1)$$

where h_i^s is the concatenation of the forward hidden state $\overleftarrow{h_i^s}$ and the backward hidden state $\overrightarrow{h_i^s}$, and e_i^s is the character embedding[4] for c_i^s in X.

2.2 Decoder

The decoder utilizes another LSTM that produces a distribution over the next target character given the source vectors $[h_1^s, h_2^s, \cdots, h_n^s]$, the previously generated target characters $\hat{Y}_{<j} = [\hat{c}_1^t, \hat{c}_2^t, \cdots, \hat{c}_j^t]$, and $M \in \mathbb{R}^{n*w}$, mathematically,

$$h_j^t = \text{LSTM}(h_{j-1}^t, e_{j-1}^t) \quad (2)$$

[3] Confusionset is downloaded from https://github.com/wdimmy/Automatic-Corpus-Generation, and this confusionset claims to cover most of spelling errors (Wang et al., 2018).

[4] We pretrain the Chinese character embedding based on the large quantities of online Chinese corpus via using the method proposed in (Sun et al., 2014).

where h_j^t is the summary of the target sentence up to the j-th word, where e_j^t is the word embedding for c_{j-1}^t. Note that during training the ground truth c_{j-1}^t is fed into the network to predict c_j^t, while at test time the most probable $\hat{c}_{j-1}^t$ is used.

We extend this decoder with an attention based model (Bahdanau et al., 2014; Luong et al., 2015), where, at every time step t, an attention score a_i^s is computed for each hidden state h_i^s of the encoder, using the attention mechanism of (Vinyals et al., 2015). Mathematically,

$$u_i = v^T \tanh(W_1 h_j^t + W_2 h_i^s) \tag{3}$$

$$a_i = \text{softmax}(u_i) \tag{4}$$

$$h_j^{t'} = \sum_{i=0}^{n} a_i h_i^s \tag{5}$$

The source vectors are multiplied with the respective attention weights, and summed to a new vector as the summary of the source vectors, $h_j^{t'}$. $h_j^{t'}$ is then interacted with the current decoder hidden state h_j^t to produce a context vector C_j:

$$C_j = tanh(W(h_j^t; h_j^{t'}) \tag{6}$$

where U, W_1, $W2$, and W are trainable parameters of the model. C_j is then used for generating two distributions: one is over the vocabulary, which is given by applying an affine transformation to C_j followed by a softmax,

$$P_{vocab} = \text{softmax}(W_{vocab} C_j) \tag{7}$$

and the other is over the input sentence, in which we use the copy mechanism. Additionally, we add the location information of the corresponding character c_j^s in X, Loc_j, and this allows the decoder to have knowledge of previous (soft) alignments at each time step. Loc_j is a vector of length n initialized by 0, and at the timestep j, the j-th element in Loc_j is set to be 1 and the other is kept to be 0. The hidden state for generating the distribution over the input sentence is as follows,

$$L_j = \text{softmax}(W_i[W_g C_j; Loc_j]) \tag{8}$$

where $\cdot;\cdot$ denotes the concatenation operation. To train the pointer networks, we define the position label at the decoding time step j as,

$$L_j^{loc} = \begin{cases} \max(z), \text{if} \exists z \quad s.t.\ c_j^t = X[z] \\ n+1, \qquad \text{otherwise} \end{cases} \tag{9}$$

The position $n+1$ is a sentinel token deliberately concatenated to the end of X that allows us to calculate loss function even if c_j^t does not exist in the input sentence. Then, the loss between L_t and L_t^{loc} is defined as,

$$Loss_l = \sum_{i}^{m} -\log L_j[L_j^{loc}] \tag{10}$$

During the inference time, $\hat{c}_j^t$ is defined as,

$$\hat{c}_j^t = \begin{cases} \arg\max(L_j), & \text{if } \arg\max(L_j) != n+1 \\ \arg\max(P_{vocab} \odot M[j]), & \text{otherwise} \end{cases} \tag{11}$$

where $\odot$ is the element-wise multiplication, and $M[j]$ is utilized to limit the scope of generated words based on the assumption that the correct character is contained in the corresponding confusionset of the erroneous character.

3 Experiments

Train data We use the large annotated corpus which contains spelling errors, either visually or phonologically resembled characters, by an automatic approach proposed in (Wang et al., 2018). In addition, a small fraction of three human-annotated training datasets provided in (Wu et al., 2013; Yu et al., 2014; Tseng et al., 2015) are also included in our training data.

Test data To evaluate the effectiveness of our proposed model, we test our trained model on benchmark datasets from three shared tasks of CSC (Wu et al., 2013; Yu et al., 2014; Tseng et al., 2015). Since these testing datasets are written in traditional Chinese, we convert them into simplified Chinese characters using OpenCC[5].

Details of experimental data statistics information, including the training datasets, the testing datasets and the Confusionsets used in our model, are shown in Table 1.

Evaluation metrics We adopt precision, recall and F1 scores as our evaluation metrics, which are widely used as evaluation metrics in CSC tasks.

Baseline models We compare our model with two baseline methods for CSC: one is N-gram language modeling with a pre-constructed confusionset (LMC), and for its simplicity and power, it is widely used in CSC (Liu et al., 2013; Yu

[5]https://github.com/BYVoid/

Name		Data Size(lines)	Avg. Sentence Length	# of Errors
Train Data	(Wang et al., 2018)	271,329	44.4	382,704
	SIGHAN 2013(**train**)	350	49.2	350
	SIGHAN 2014(**train**)	6,526	49.7	10,087
	SIGHAN 2015(**train**)	3,174	30.0	4,237
	Total	281,379	44.4	397,378
Test Data	SIGHAN 2013(**test**)	974	74.1	1,227
	SIGHAN 2014(**test**)	526	50.1	782
	SIGHAN 2015(**test**)	550	30.5	715

Name	# of Characters	Avg. # of confusionset
Confusionsets	4,922	7.8

Table 1: Experimental Data Statistics Information.

Methods	Detection-level									Correction-level								
	Test_{13}			Test_{14}			Test_{15}			Test_{13}			Test_{14}			Test_{15}		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
LMC	**79.8**	50.0	61.5	56.4	34.8	43.0	**83.8**	26.2	40.0	77.6	22.7	35.1	71.1	50.2	58.8	67.6	31.8	43.2
SL	54.0	69.3	60.7	51.9	66.2	58.2	56.6	69.4	62.3	\	\	\	\	\	\	\	\	\
Ours$^-$	40.7	84.3	54.8	51.1	72.3	59.9	58.7	61.7	60.2	67.1	31.9	43.2	51.6	64.7	57.4	46.7	43.9	45.3
Ours$^+$	56.8	**91.4**	**70.1**	**63.2**	**82.5**	**71.6**	66.8	**73.1**	**69.8**	**79.7**	**59.4**	**68.1**	**79.3**	**68.9**	**73.7**	**71.5**	**59.5**	**64.9**

Table 2: Experimental results of detection-level and correction-level performance on three testing datasets (%). + and - denote using Confusionsets and not using Confusionsets, respectively.

and Li, 2014; Xie et al., 2015). By utilizing the confusionset to replace characters in a sentence, the sentence probability is calculated after and before the replacement, which is then used to determine whether the sentence contains spelling errors. We re-implement the pipline proposed in (Xie et al., 2015); Another is the sequence labeling method (SL), which casts Chinese spelling error detection into a sequence tagging problem on characters, in which the correct and incorrect characters are tagged as 1 and 0, respectively. We follow the baseline model (Wang et al., 2018) that implements a LSTM based sequence tagging model.

Model Hyperparameters The training hyperparameters are selected based on the results of the validation set. The dimension of word embedding is set to 300 and the hidden vector is set to 512 in both the encoder and decoder. The dimension of the attention vector is also set to 512 and the dropout rate is set to 0.5 for regularization. The mini-batched Adam (Kingma and Ba, 2014) algorithm is used to optimize the objective function. The batch size and base learning rates are set to 64 and 0.001, respectively.

Results As shown in Table 2, we compare our confusionset-guided pointer networks with two baseline methods. Not to our surprise, except for two precision results lower than LMC, our model consistently improves performance over other models for both detection-level and correction-level evaluation. One reason might be that compared with SL, which considers the spelling check as a classification task at the character-level, and the information available for the current timpstep is somewhat constrained while our generative model can utilize both the location information and the whole input information by an attention mechanism, and the copy mechanism also make the decoding more effective. As for LMC, how to set a threshold probability for judging whether a given sentence is correct remain explored, and there exists great trade-off between the precision and the recall as reported in (Jia et al., 2013).

Utility of M Specifically, by comparing the experimental results of Ours$^-$ and Ours$^+$, we can observe that the latter achieves better performance,

which validates the effectiveness of utilizing Confusionsets that can help improve the probability of generating correct target characters.

4 Discussion and Future Work

In our everyday Chinese writing, there exist a variety of problematic usage of language, one of which is the spelling error referred in this paper. Such spelling errors are mainly generated due to the similarity of Chinese characters in terms of sound, shape, and/or meaning, and the task is to detect the misspelled words and then replace them with their corresponding correct ones. Besides the spelling errors mentioned above, grammar errors are also common in our Chinese writing, which requires us to correct the erroneous sentence by insertion, deletion and even re-ordering. Take as an example "我真不不明白，为啥他要自杀。" (Translation: I really don't understand why he committed suicide.), we need to delete the character in red in order to guarantee the correctness of the sentence. However, our model is unable to handle such errors in that we limit the length of the generated sentence to be same to that of the input sentence in order to incorporate Confusionsets into our model as a guiding resource.

For the future work, we hope to extend this idea proposed in this paper to train a model capable of handling different types of errors through the generative model since it can generate different lengths of results. One concern is that we need to reconsider how to incorporate Confusionsets into the encoder-decoder architecture.

5 Related Work

Most CSC related studies have emerged as a result of a series of shared tasks (Wu et al., 2013; Yu et al., 2014; Tseng et al., 2015; Fung et al., 2017; Gaoqi et al., 2018), which involve automatic detection and correction of spelling errors for a given sentence. Earlier work in CSC focus mainly on unsupervised methods such as language model with a pre-constructed confusionset (Liu et al., 2013; Yu and Li, 2014). Subsequently, some work cast CSC as a sequential labeling problem, in which conditional random fields (CRF) (Lafferty et al., 2001), gated recurrent networks (Hochreiter and Schmidhuber, 1997; Chung et al., 2014) have been employed to model the problem (Zheng et al., 2016; Xie et al., 2017; Wu et al., 2018). More recently, motivated by a serials of remarkable success achieved by neural network-based sequence-to-sequence learning (Seq2Seq) in various natural language processing (NLP) tasks (Sutskever et al., 2014; Cho et al., 2014), generative models have also been applied to the spelling check task by considering it as an encoder-decoder (Xie et al., 2016; Ge et al., 2018).

6 Conclusion and Future Work

We proposed a novel end-to-end confusionset-guided encoder-decoder model for the Chinese Spelling Check (CSC) task. By the infusion of Confusionsets with copy mechanism, our proposed approach achieves a huge performance gain over competitive baselines, demonstrating its effectiveness on the CSC task.

Acknowledgements

The authors want to express special thanks to all anonymous reviewers for their insightful and valuable comments and suggestions on various aspects of this work.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv preprint arXiv:1409.0473*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations Using RNN Encoder-decoder for Statistical Machine Translation. *arXiv preprint arXiv:1406.1078*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv preprint arXiv:1412.3555*.

Gabriel Fung, Maxime Debosschere, Dingmin Wang, Bo Li, Jia Zhu, and Kam-Fai Wong. 2017. Nlptea 2017 shared task–chinese spelling check. In *Proceedings of the 4th Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA 2017)*, pages 29–34.

RAO Gaoqi, Qi Gong, Baolin Zhang, and Endong Xun. 2018. Overview of nlptea-2018 share task chinese grammatical error diagnosis. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 42–51.

Tao Ge, Furu Wei, and Ming Zhou. 2018. Reaching Human-level Performance in Automatic Grammatical Error Correction: An Empirical Study. *arXiv preprint arXiv:1807.01270*.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the Unknown Words. *arXiv preprint arXiv:1603.08148*.

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

Zhongye Jia, Peilu Wang, and Hai Zhao. 2013. Graph Model for Chinese Spell Checking. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, pages 88–92.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.

Chao-Lin Liu, Min-Hua Lai, Yi-Hsuan Chuang, and Chia-Ying Lee. 2010. Visually and Phonologically Similar Characters in Incorrect Simplified Chinese Words. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 739–747. Association for Computational Linguistics.

Xiaodong Liu, Kevin Cheng, Yanyan Luo, Kevin Duh, and Yuji Matsumoto. 2013. A hybrid Chinese Spelling Correction Using Language Model and Statistical Machine Translation with Reranking. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, pages 54–58.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. *arXiv preprint arXiv:1508.04025*.

Yaming Sun, Lei Lin, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2014. Radical-enhanced Chinese Character Embedding. In *International Conference on Neural Information Processing*, pages 279–286. Springer.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in neural information processing systems*, pages 3104–3112.

Yuen-Hsien Tseng, Lung-Hao Lee, Li-Ping Chang, and Hsin-Hsi Chen. 2015. Introduction to Sighan 2015 Bake-off for Chinese Spelling Check. *ACL-IJCNLP 2015*, page 32.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar As a Foreign Language. In *Advances in neural information processing systems*, pages 2773–2781.

Dingmin Wang, Yan Song, Jing Li, Jialong Han, and Haisong Zhang. 2018. A hybrid approach to automatic corpus generation for chinese spelling check. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2517–2527.

Shih-Hung Wu, Chao-Lin Liu, and Lung-Hao Lee. 2013. Chinese Spelling Check Evaluation at SIGHAN Bake-off 2013.

Shih-Hung Wu, Jun-Wei Wang, Liang-Pu Chen, and Ping-Che Yang. 2018. CYUT-III Team Chinese Grammatical Error Diagnosis System Report in NLPTEA-2018 CGED Shared Task. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 199–202.

Pengjun Xie et al. 2017. Alibaba at ijcnlp-2017 task 1: Embedding grammatical features into lstms for chinese grammatical error diagnosis task. *Proceedings of the IJCNLP 2017, Shared Tasks*, pages 41–46.

Weijian Xie, Peijie Huang, Xinrui Zhang, Kaiduo Hong, Qiang Huang, Bingzhou Chen, and Lei Huang. 2015. Chinese Spelling Check System Based on N-gram Model. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, pages 128–136.

Ziang Xie, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y Ng. 2016. Neural language correction with character-based attention. *arXiv preprint arXiv:1603.09727*.

Junjie Yu and Zhenghua Li. 2014. Chinese Spelling Error Detection and Correction Based on Language Model, Pronunciation, and Shape. In *Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 220–223.

Liang-Chih Yu, Lung-Hao Lee, Yuen-Hsien Tseng, Hsin-Hsi Chen, et al. 2014. Overview of SIGHAN 2014 Bake-off for Chinese Spelling Check. In *Proceedings of the 3rd CIPSSIGHAN Joint Conference on Chinese Language Processing (CLP'14)*, pages 126–132.

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.