Chinese Spelling Check based on Sequence Labeling

Zijia Han
School of Computer Science and Technology,
Heilongjiang University
Harbin, China
e-mail: hanzijianlp@hotmail.com

Chengguo Lv
School of Computer Science and Technology,
Heilongjiang University
Harbin, China
e-mail: 13936160042@163.com

Qiansheng Wang
School of Computer Science and Technology,
Heilongjiang University
Harbin, China
e-mail: chncwang@gmail.com

Guohong Fu
School of Computer Science and Technology,
Soochow University
Suzhou, China
e-mail: ghfu@hotmail.com

***Abstract*—Chinese has been widely used by people from all over the world. Various Chinese spelling errors may occur inevitably from Chinese as Foreign Language (CFL) beginners. This paper presents a method for Chinese spelling check to detect and correct spelling errors in a sentence. Our approach is based on the sequence labeling model using the bi-direction LSTM network. We also utilize the Pinyin feature and Chinese strokes feature to improve performance. The evaluation on the SIGHAN-8 shows that our approach gets good performance on both detection and correction tasks.**

Keywords—Chinese spelling check, Bi-LSTM, sequence labeling, Pinyin, Chinese strokes

I. INTRODUCTION

With the development of China's economy and the popularity of Chinese, booming people are learning Chinese as a Foreign Language (CFL). The number of CFL learners is expected to become larger for the years to come. First CFL learners tend to find Chinese is a difficult language for its confusing tones and complicated characters, and they are more prone to misspelling Chinese words or characters than native users. In this situation, a spelling check system is necessary for these people.

In this paper, we present a Chinese spelling check method based on a sequence labeling model. A sentence with potential typos is corrected by the model and a new sentence without typo is re-generated character by character. In a sentence, most characters are correct and origin-correction sentence pairs are equal in length, which means the origin-correction sentence pairs have a strong correlation. However, the neural machine translation (NMT) model is designed to translate one language into another language even if they are totally different. Intuitively, compared to the NMT model, the sequence labeling model is more suitable to CSC task. Besides, the sequence labeling model has the strength of lightweight, so it can save the computing resource and makes the model trained faster.

There are two main types of spelling errors: pronunciation similarity errors and shape similarity errors[1]. For example, a sentence with a typo like "我覺得你們會好好的碗。"(I think you will bowl yourselves.), where the typo "碗" (bowl) should be corrected to "玩" (enjoy). That is a pronunciation similarity error. For another sentence "我覺得你們會好好的元。"(I think you will meta yourselves.), where the typo "元"(meta) should be corrected to "玩"(enjoy). That is a shape similarity error.

To address both of these errors, we incorporate the Pinyin feature and the strokes feature into our sequence labeling model separately. Pinyin is widely used as a phonetic system. Strokes, also call CJK strokes, are the smallest unit of Chinese characters. The characters with similar pronunciations have similar Pinyin while the characters with similar shapes have similar strokes.

Experiments show that the performance of the sequence labeling model is better than the NMT model. Our model performs even better utilizing the Pinyin feature and the strokes feature.

The rest of the paper is organized as follows. We review the related work in Section II. Then we introduce our method based on sequence labeling with utilizing the Pinyin feature and strokes feature to improve the performance of the model in Section III. In section IV, we detail the evaluation metrics and experimental performance of our models. Finally, we conclude the paper and explore future work in Section V.

II. RALATED WORK

Chinese Spelling Check (CSC)[2] has been active research in recent years. It can be divided into two sub-tasks: spelling error detection and spelling error correction. There other similar tasks like Chinese Grammatical Error Diagnosis (CGED)[3] and Spelling Error Correction (SEC)[4]. In our work, we address the task of correcting Chinese spelling errors for CFL learners and raise an effective model with linguistic features to improve the performance.

Early works on spelling correction tasks are mainly two kinds of methods: rule-based methods and statistical-based methods. Mangu and Brill[5] proposed a rule-based approach for automatically acquiring linguistic knowledge to help correct spelling errors. Mays et al.[6] proposed a statistical-based model on spelling error correction tasks,

where context information is incorporated into the detection and correct spelling errors.

Gao et al.[6] introduced statistical machine translation to search query spelling correction. Lopez et al.[7] improved the statistical machine translation model to build a data-driven system. Recently, neural machine translation (NMT) has been applied to spelling check task. Yuan and Briscoe[8] first present NMT model for English grammatical error correction.

Rei and Yannakoudakis[9] first applied the sequence labeling model on error detection task. Their models are based on bi-direction LSTM and are able to outperform other participants on detecting errors in learner writing.

As to the Chinese spelling check task, Xie et al.[10] utilized an N-gram language model on CSC. Their models are based on the bi-gram and tri-gram language model as well as Chinese word segmentation. Wang and Liao[11] proposed an approach based on word vector and conditional random field.

Jin et al.[12] utilized a hybrid approach to Chinese spelling correction. They integrated three models including the n-gram language model, Pinyin-based language model and tone-based language model to improve the performance of Chinese checking spelling error system

Generating artificial misspelled sentences is a practical way to help train models, which can address the problem of corpus lack. Li et al.[13] added artificial error data to expand the dataset and applied the NMT model on the Chinese spelling check task. The result shows their approach is able to significantly improves the performance.

A united framework for Chinese spelling check called HANSpeller was proposed by Zhang et al [14]. The framework is based on extended HMM and ranker-based models, together with a rule-based model for further polishing. They afterward proposed HANspeller++[15] based on their previous work. The improvements including candidates generating, candidates re-ranking and final global decision and the result show the state-of-the-art performance on CSC tasks.

Compared to previous research on Chinese spelling check, we apply the sequence labeling model on Chinese spelling check and it shows that our approach performs better than the NMT model. Furthermore, we utilize the Pinyin feature and the strokes feature to enhance performance on typo detection and correction.

III. METHOD

We consider the Chinese spelling check as a sequence labeling task. A sentence with potential misspelling characters will be re-generated as an error-free sentence character by character through a sequence labeling neural network.

In our approach, we prepare training data from SIGHAN successive CSC tasks (Section A). We then introduce the architecture of our sequence labeling based neural network model (Section B) and incorporate the Pinyin feature 0(Section C) along with the strokes feature (Section D).

A. Data Processing

We use SIGHAN-2013 CSC Datasets, CLP-2014 CSC datasets, and SIGHAN-2015 CSC training data as our training data. The source texts are a set of the sentence with mistakes and corrections. We extract the typos in the source texts and replace them with correction characters. For example, the sentence is in SGML format

<ESSAY title="上學遲到">
<TEXT>
<PASSAGE id="A2-3006-1">他快進座下，下課以後跟老師說「不好意思！昨晚很晚才睡覺。」</PASSAGE>
</TEXT>
<MISTAKE id="A2-3006-1" location="4">
<WRONG>座下</WRONG>
<CORRECTION>坐下</CORRECTION>
</MISTAKE>
</ESSAY>

is converted to

“他快進座下，下課以後跟老師說「不好意思！昨晚很晚才睡覺。”

“他快進坐下，下課以後跟老師說「不好意思！昨晚很晚才睡覺。”

The upper one is the original sentence with potential errors and the one below is the error-free sentence after correction. We replace the character “座 ” (seat) with the right character “坐 ”(sit).

B. Sequence Labeling Model

We train a character-based sequence labeling model, to translate a sentence with potential errors to an error-free sentence. The source sentence $X = [x_1, x_2, \dots, x_J]$ and the target sentence $Y = [y_1, y_2, \dots, y_K]$. We use origin-and-correction sentence pairs as source-and-target training sentence pairs.

It has been shown that word embedding plays an important role to improve sequence labeling performance [16]. We use Chinese word2vec to convert $X = [x_1, x_2, \dots, x_J]$ into vectors $E = [e_{x1}, e_{x2}, \dots, e_{xJ}]$, which is fed into a bi-direction long-short memory (LSTM) [17] model. Each character will be predicted to a new character after LSTM hidden layer via a softmax classifier.

Recurrent neural networks (RNN)[18] are a promising model to capture long-distance dependencies while in practice they fail due to gradient vanishing or exploding problems.[19] LSTMs are designed to solve these problems. It is proved that LSTMs have better performance than RNNs dealing with long sequences. An LSTM unit is composed of three gates including input gate, forget gate and output gate, which control the cell to forget or pass on the information to the next time step.

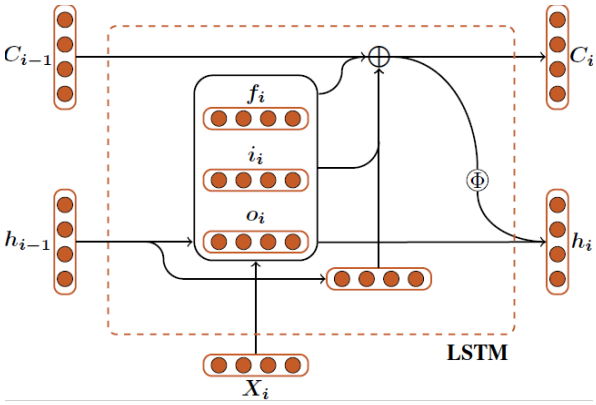

Figure 1. The architecture of LSTM

Figure 1 shows the architecture of an LSTM unit. The formulas of an LSTM unit are below:

$$\mathbf{f_t} = \boldsymbol{\sigma}(\mathbf{W_f}[\mathbf{h_{t-1}}, \mathbf{x_t}] + \mathbf{b_f}) \quad (1)$$

$$\mathbf{i_t} = \boldsymbol{\sigma}(\mathbf{W_i}[\mathbf{h_{t-1}}, \mathbf{x_t}] + \mathbf{b_i}) \quad (2)$$

$$\tilde{\mathbf{C}}_\mathbf{t} = \mathbf{tanh}(\mathbf{W_C}[\mathbf{h_{t-1}}, \mathbf{x_t}] + \mathbf{b_C}) \quad (3)$$

$$\mathbf{C_t} = \mathbf{f_t} \odot \mathbf{C_{t-1}} + \mathbf{i_t} \odot \tilde{\mathbf{C}}_\mathbf{t} \quad (4)$$

$$\mathbf{o_t} = \boldsymbol{\sigma}(\mathbf{W_o}[\mathbf{h_{t-1}}, \mathbf{x_t}] + \mathbf{b_o}) \quad (5)$$

$$\mathbf{h_t} = \mathbf{o_t} \odot \mathbf{tanh}(\mathbf{c_t}) \quad (6)$$

where $\boldsymbol{x_t}$ is the input vector with word embedding representation and $\boldsymbol{h_t}$ is the hidden state at time t. $\boldsymbol{W_f}, \boldsymbol{W_i}, \boldsymbol{W_C}, \boldsymbol{W_o}$ is the weight matrices for hidden state $\boldsymbol{h_t}$ and $\boldsymbol{b_f}, \boldsymbol{b_i}, \boldsymbol{b_C}, \boldsymbol{b_o}$ is the bias vectors. $\boldsymbol{\sigma}$ denotes sigmoid function.

In order to get both past and future contexts, we use bi-direction LSTM (Bi-LSTM)[20] to improve our model. The input vectors are fed to two LSTMs in opposite directions. Then two hidden states of LSTM are concatenated to form the final outputs of the hidden layers.

C. *Incorporating Pinyin Feature*

Pinyin is a scheme of using Latin letters as the phonetic symbol of Mandarin Chinese. It includes four diacritics denoting tones. We use Pinyin as the additional feature to improve the performance of the spelling check task.

First, we obtain the Pinyin of every character. Both with-tone and without-tone Pinyin are incorporated into our model. Pinyin input vectors are concatenated with Chinese character input vectors, and we feed them into Bi-LSTM network. Figure 2 shows the network we use to extract Pinyin feature together with Chinese character.

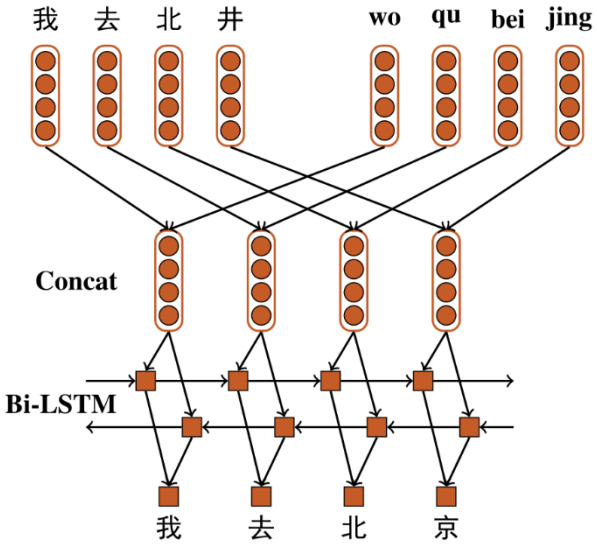

Figure 2. The architecture of our network using the Pinyin feature. The character representation is converted to word embedding. Then the character vector is concatenated with the Pinyin vector before feeding into the Bi-LSTM network. The outputs are the correction character after a softmax layer.

D. *Incorporating Strokes Feature*

Chinese characters have various strokes like Dot (“丶”,“点”)、Horizontal (“一”, “横”)、Vertical (“丨”, “竖”) and so on. These strokes make up all kinds of complex Chinese characters. First, we break up characters and extract strokes from all the characters in training data. Then we gather and number them, obtain the stroke indexes of each character. For example, the stroke indexes of character “才” (ability; talent; just) are [2,6,4], where ‘2’ denotes Horizontal (“一”), ‘0’ denotes J hook (“亅”), ‘4’ denotes Throw (“丿”).

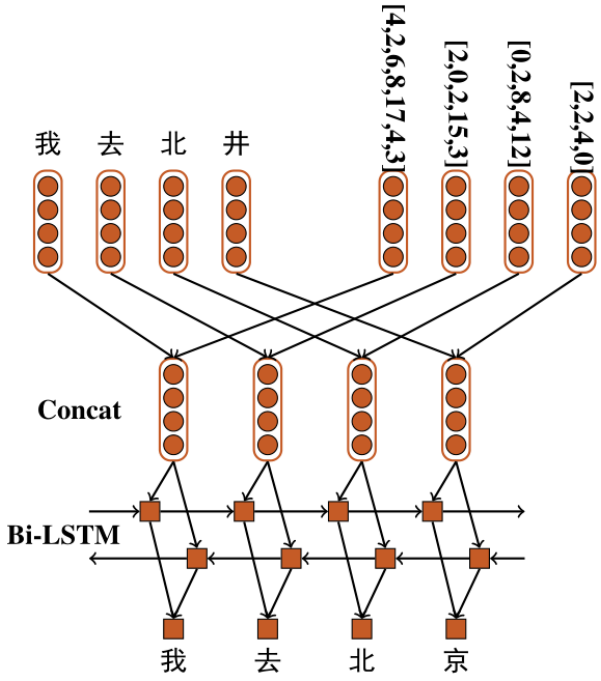

Figure 3. The architecture of our model with strokes feature. The character representation is converted to word embedding. The character vector is concatenated with the strokes indexes vector before feeding

into the Bi-LSTM network. The outputs is the correction character after a softmax layer.

After vectorization, the stroke feature is concatenated with character input vectors. We feed them into Bi-LSTM and get the prediction outputs. Figure 3 shows the model of Bi-LSTM with strokes feature.

IV. EXPERIMENT

In this section, we introduce the dataset used in Section A, and the evaluations of our task in Section B. Then we detail the hyper-parameters of sequence labeling model and NMT model in Section C. Finally, in Section D, we present the performance of these models.

A. Dataset

Training Data: We use CLP-2014 CSC Datasets[21] and SIGHAN-2013 CSC Datasets[2] as well as SIGHAN-2015 CSC training data as our training data[22]. There are 897 sentences without error and there are 7641 sentences with one or more errors. The total number of sentences in training data is 8538.

Test Data: We use SIGHAN-2015 CSC test data[22] as our test data. There are 897 sentences without error and there are 7641 sentences with one or more errors. The total number of sentences in training data is 8538.

B. Evaluation

The criteria we use for judging correctness is divided into two subtasks. One is the detection level and the other is the correction level. For the detection level, all locations of incorrect characters in a given passage should be completely identical with the gold standard. For detection level, locations and corresponding corrections of incorrect characters should be completely identical with the gold standard.

Table I shows the Confusion matrix. The following metrics are measured at both levels with the help of the confusion matrix.

TABLE I. CONFUSION MATRIX FOR EVALUATION

Confusion Matrix		System Results	
		Positive	Negative
		(Error)	(No Error)
Gold Standard	Positive	TP	FN
	Negative	FP	TN

As is shown in the following, the Evaluation metric consists of false positive rate, accurate rate, precision rate, recall rate and F1-score.

$$False\ Positive\ Rate\ (FPR) = \frac{FR}{FP + TN} \tag{7}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

$$F1 - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{11}$$

C. Hyper-parameters of our Model

The hyper-parameters of our sequence labeling models are the same among these models. We use traditional Chinese word2vec with d = 200 dimensions as our word embedding, which is trained with CTB-60 dataset.

We set our input vector dimension as 200. The dimension of Pinyin input vectors and stroke input vectors are 200 as well. Our model contains a 1-layer Bi-LSTM network with 400 hidden nodes. We use Adam Algorithm[23] as the optimizer to train the model with learning rate 0.001. The Learning rate decay method is applied in our model with 0.05 decay each epoch. The batch size is set as 10 and the dropout is 0.5.

We train an NMT model as our baseline model. We use a sequence-to-sequence model and both encoder and decoder are a bi-direction gated recurrent unit (GRU)[24]. We set the input vector dimension as 512 and the hidden layer dimension is 512. We use Adam with learning rate 0.0001.

D. Result

Table II shows the evaluation on the detection level, table III shows the evaluation on the correction level. Seq2Seq denotes the model based on the sequence-to-sequence approach, which is the NMT model. SL denotes the model based on the sequence labeling approach, following with which is the sequence labeling model incorporating the Pinyin feature and the strokes feature.

As we can see, models based on sequence labeling, which is Bi-LSTM, get much better performances than the sequence-to-sequence model on both evaluations. When we using the Pinyin feature or strokes feature, the performances are better than Bi-LSTM only models. The combined models with the Pinyin feature and the strokes feature performs are superior to Bi-LSTM only models but inferior to using the Pinyin feature only models.

We consider that the excessive features are not favorable to our model. In addition, most of the Chinese typos are pronunciation similar error. That's the reason of why the Individual Pinyin feature model performed better than the strokes feature and the combine feature.

Experiments show that the model trained with the Pinyin feature gets the best recall and F1-score on the detection level and correction level. While the model trained with the strokes feature gets the best precision and accuracy on both levels.

Moreover, the combined model trained with the Pinyin feature and the strokes feature can be regarded as a compromise model between the above two models.

V. CONCLUSION

This paper proposes a spelling check approach based on the sequence labeling model. We also incorporate the Pinyin feature and strokes feature to improve the

evaluation performance. Our approach achieved a considerable result at SIGHAN-2015 Chinese spelling check task.

Future works on Chinese spelling check (CSC) are as follows: (1) The critical problem for CSC task is the lack of corpus source. Sentences with potential errors are difficult to collect or generate. Ample corpus is also a necessary condition to train a fancy model. Finding a way to solve the corpus deficiency problem is key to the CSC task. (2) Transfer learning ought to happen used in CSC. We can use transfer learning to enhance the proximity of origin corpus and artificial corpus

ACKNOWLEDGMENT

The authors would like to thank the organizers and the reviewers of IALP-2019 for their dedicated and professional works. This research was supported by the National Natural Science Foundation of China No. 61672211).

TABLE II. PERFORMANCE OF OUR MODEL AND SEQ2SEQ MODEL ON CORRECTION LEVEL

	Seq2Seq	SL	SL+Pinyin	SL+strokes	SL+Pinyin+strokes
FPR	0.1309	0.1064	0.1119	**0.0881**	0.1064
Accuracy	0.5645	0.5976	0.6076	**0.6104**	0.6085
Precision	0.6651	0.7467	0.752	**0.7848**	0.7593
Recall	0.26	0.3076	**0.3327**	0.3147	0.3291
F1-score	0.3739	0.4357	**0.4613**	0.4493	0.4592

TABLE III. PERFORMANCE OF OUR MODEL AND SEQ2SEQ MODEL ON DETECTION LEVEL

	Seq2Seq	SL	SL+Pinyin	SL+strokes	SL+Pinyin+strokes
FPR	0.1611	0.1014	0.1119	**0.0807**	0.1064
Accuracy	0.5845	0.6240	0.6231	**0.6267**	0.6231
Precision	0.6500	0.7615	0.7681	**0.8112**	0.7743
Recall	0.3159	0.3377	**0.3633**	0.3399	0.3579
F1-score	0.4252	0.4679	**0.4933**	0.4791	0.4895

REFERENCES

[1] C.-L. Liu, M.-H. Lai, K.-W. Tien, Y.-H. Chuang, S.-H. Wu, and C.-Y. Lee, "Visually and phonologically similar characters in incorrect Chinese words: Analyses, identification, and applications," ACM Transactions on Asian Language Information Processing (TALIP), vol. 10, no. 2, pp. 10, 2011.

[2] S.-H. Wu, C.-L. Liu, and L.-H. Lee, "Chinese spelling check evaluation at SIGHAN Bake-off 2013." pp. 35-42, 2013.

[3] Lee L H, Gaoqi R A O, Yu L C, et al. Overview of nlp-tea 2016 shared task for chinese grammatical error diagnosis[C]//Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016). 2016: 40-48.

[4] K. Oflazer, "Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction," Computational Linguistics, vol. 22, no. 1, pp. 73-89, 1996.

[5] L. Mangu, and E. Brill, "Automatic rule acquisition for spelling correction." pp. 187-194, 1997.

[6] Gao J, Hwang M, Huang X D, et al. Statistical Machine Translation Based Search Query Spelling Correction: U.S. Patent Application 13/296,640[P]. 2013-5-16.

[7] Lopez Ludeña V, San Segundo Hernández R, Montero Martínez J M, et al. Architecture for text normalization using statistical machine translation techniques[J]. 2012.

[8] Z. Yuan, and T. Briscoe, "Grammatical error correction using neural machine translation." pp. 380-386, 2016.

[9] M. Rei, and H. Yannakoudakis, "Compositional sequence labeling models for error detection in learner writing," arXiv preprint arXiv:1607.06153, 2016.

[10] W. Xie, P. Huang, X. Zhang, K. Hong, Q. Huang, B. Chen, and L. Huang, "Chinese spelling check system based on n-gram model." pp. 128-136, 2016.

[11] Y.-R. Wang, and Y.-F. Liao, "Word Vector/Conditional Random Field-based Chinese Spelling Error Detection for SIGHAN-2015 Evaluation," Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing. pp. 46-49, 2015.

[12] P. Jin, X. Chen, Z. Guo, and P. Liu, "Integrating pinyin to improve spelling errors detection for Chinese language." pp. 455-458, 2014.

[13] Li C W, Chen J J, Chang J S. Chinese spelling check based on neural machine translation[C]//Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation. 2018.

[14] Zhang Q, Zhang S, Hou J, et al. HANSpeller: A Unified Framework for Chinese Spelling Correction[C]//International Journal of Computational Linguistics & Chinese Language Processing, Volume 20, Number 1, June 2015-Special Issue on Chinese as a Foreign Language. 2015.

[15] S. Zhang, J. Xiong, J. Hou, Q. Zhang, and X. Cheng, "HANSpeller++: A Unified Framework for Chinese Spelling Correction," Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing. pp. 38-45, 2015.

[16] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," Journal of machine learning research, vol. 12, no. Aug, pp. 2493-2537, 2011.

[17] S. Hochreiter, and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[18] C. Goller, and A. Kuchler, "Learning task-dependent distributed representations by backpropagation through structure." pp. 347-352, 1996.

[19] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” IEEE transactions on neural networks, vol. 5, no. 2, pp. 157-166, 1994.

[20] C. Dyer, M. Ballesteros, W. Ling, A. Matthews, and N. A. Smith, “Transition-based dependency parsing with stack long short-term memory,” arXiv preprint arXiv:1505.08075, 2015.

[21] Yu L C, Lee L H, Tseng Y H, et al. Overview of SIGHAN 2014 bake-off for Chinese spelling check. 126-132, 2014.

[22] Y.-H. Tseng, L.-H. Lee, L.-P. Chang, and H.-H. Chen, "Introduction to sighan 2015 bake-off for chinese spelling check." pp. 32-37, 2015.

[23] D. P. Kingma, and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

[24] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” arXiv preprint arXiv:1409.1259, 2014.