Detection of Chinese Word Usage Errors for Non-Native Chinese Learners with Bidirectional LSTM

Yow-Ting Shiue, Hen-Hsen Huang and Hsin-Hsi Chen
Department of Computer Science and Information Engineering
National Taiwan University
No. 1, Sec. 4, Roosevelt Road, Taipei, 10617 Taiwan
orina1123@gmail.com,hhhuang@nlg.csie.ntu.edu.tw,hhchen@ntu.edu.tw

Abstract

Selecting appropriate words to compose a sentence is one common problem faced by non-native Chinese learners. In this paper, we propose (bidirectional) LSTM sequence labeling models and explore various features to detect word usage errors in Chinese sentences. By combining CWINDOW word embedding features and POS information, the best bidirectional LSTM model achieves accuracy 0.5138 and MRR 0.6789 on the HSK dataset. For 80.79% of the test data, the model ranks the ground-truth within the top two at position level.

1 Introduction

Recently, more and more people around the world choose Chinese as their second language. That results in an increasing need for automatic grammatical error detection and correction (GEC) tools. To measure the performance of GEC systems in a standardized manner, several shared tasks have been conducted for English (Dale and Kilgarriff, 2011; Dale et al., 2012; Ng et al., 2013, 2014) and Chinese (Yu et al., 2014; Lee et al., 2015, 2016).

In Chinese sentences, a word usage error (WUE) is a grammatically or semantically incorrect token which is written in a wrong form itself, or is an existent word but is improper for its context (refer to example (E1)). In fact, many Chinese WUEs result from subtle semantic unsuitability instead of violation of syntactic constraints. In example (E1), both 權力 (power) and 權利 (right) are nouns in Chinese, and both versions are grammatically correct. It is difficult to formulate an explicit rule for recognizing this kind of errors.

(E1) 人們有(*權力,權利)吃安全的食品。
(People have the (*power, right) to enjoy safe food.)

Shiue and Chen (2016) adopted the HSK corpus, a dynamic composition corpus built by Beijing Language and Culture University, to study the detection of WUEs. Instead of specific position information, their model only determines whether a sentence segment contains WUEs. Huang et al. (2016) used the HSK corpus to study the preposition selection problem. They proposed gated recurrent unit (GRU)-based models to select the most suitable one from a closed set of Chinese prepositions given the sentential context. Although their approach can be utilized to detect and correct preposition errors, it is still worth investigating how to recognize WUEs involving other types of words such as verbs and nouns.

In the past few years, distributed word representations derived from neural network models (Mikolov et al., 2013a; Pennington et al., 2014) have become popular among various studies in natural language processing. Beyond surface forms, these low-dimensional vector representations can encode syntactic and semantic information implicitly (Mikolov et al., 2013b). Because WUEs involve syntactic or semantic problems, vector representations could be promising for finding the erroneous tokens.

One challenging aspect of dealing with grammatical errors is that the errors usually do not stand on their own, but are dependent on the context (Chollampatt et al., 2016). Therefore, we need a model that considers the sequence of words in a sentence as a whole to determine which position needs correction. One possible model for this task is the Long Short-Term Memory (LSTM) model (Hochreiter and Schmidhuber, 1997), which processes sequential data and generates the output based not only on the information of the current time step, but also on the past information stored in the memory layer. Rei and Yannakoudakis (2016) adopted neural network models, including

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 404–410
Vancouver, Canada, July 30 - August 4, 2017. ©2017 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17-2064

LSTM, to detect errors in English learner writing. However, they mainly focused on comparing different composition architectures under the same word representation, so it remained unclear to what extent pre-trained word embeddings can help. Huang and Wang (2016) used LSTM for Chinese grammatical error diagnosis, but their models are trained only on learner data, without external well-formed text. That means the performance might be limited by the relatively small amount of annotated sentences written by foreign learners.

This paper utilizes LSTM and its extension (Bidirectional LSTM) along with the information derived from external resources to deal with Chinese WUE detection. Several types of pre-trained word embeddings and additional token-level features are considered. Each token in a sentence will be labeled correct or incorrect. Experimental results show that our models can rank the ground-truth error position toward the top of the candidate list.

2 WUE Detection Based on Bidirectional LSTM

We formulate the Chinese WUE detection task as a sequence labeling problem. Each token, the fundamental unit after word segmentation, is labeled either correct (0) or incorrect (1).

We utilize the LSTM model for labeling. LSTM models long sequences better than simple recurrent neural network (RNN) does, since it is equipped with input, output and forget gates to control how much information is used. The ability of LSTM to capture longer dependencies among time steps makes it suitable for modeling the complex dependencies of the erroneous token on the other parts of the sentence.

We train the LSTM model with the Adam optimizer (Kingma and Ba, 2014) implemented in Keras (Chollet, 2015). The loss function is binary cross entropy. The batch size and the initial learning rate is set to 32 and 0.001 respectively. The training process is stopped when the validation accuracy does not increase for two consecutive epochs. The model with the highest validation accuracy is selected as the final model.

We apply a sigmoid activation function before the output layer, so the output score of each token, which is between 0 and 1, can be interpreted as the predicted level of incorrectness. With these scores, our system can output a ranked list of candidate error positions. The positions with the highest incorrectness scores will be marked as incorrect. In (E2) we show an example labeling result of our system. The tokens 差 (bad) and 知識 (knowledge), with the highest scores, are most likely to be incorrect.

(E2)	學習	的	知識	也	很	差
	0.056	0.035	**0.153**	0.039	0.030	**0.429**

(The **knowledge** learned is also very **bad**.)

Bidirectional LSTM (Schuster and Paliwal, 1997) is an extension of LSTM which includes a backward LSTM layer. Both information before and after the current time step are taken into consideration. We need the "future" information to detect the error in example (E3). The incorrectness of the token 留在 (left at) cannot be determined without considering its object 我們 (us).

(E3) 店 是 爸爸 (*留在,留給) 我們 的 。

(The store is our father left (*at,to) us.)

3 Sequence Embedding Features

We consider the word sequence in a sentence and the corresponding POS tag sequence. They are mapped to sequences of real-valued vectors through an embedding layer. These vectors are also updated during the training process.

3.1 Word Embeddings

We set the word embedding size to 400. Besides randomly initialized embedding, we also tried several types of pre-trained word vectors. To train the word embeddings, we utilize the Chinese part of the ClueWeb09 dataset[1]. The Chinese part was extracted and segmented by Yu et al. (2012).

CBOW/Skip-gram Word Embeddings

We trained word vectors with the two architectures included in the word2vec software (Mikolov et al., 2013a). The continuous bag-of-words model (CBOW) uses the words in a context window to predict the target word, while the skip-gram model (SG) uses the target word to predict every word in the context window.

CWINDOW/Structured Skip-gram Word Embeddings

Taking the order of the context words into consideration, we also employ the continuous window model (CWIN) and the structured skip-gram

[1] http://lemurproject.org/clueweb09.php

model (Struct-SG) (Ling et al., 2015). The former replaces the summation of context word vectors in CBOW with a concatenation operation, and the latter applies different projection matrices for predicting context words in different relative position with the target word.

3.2 POS Embeddings

The POS embeddings are randomly initialized. We set the embedding size to 20, which is slightly smaller than the number of different POS tags (30) in our dataset.

4 Token Features

In addition to representing each token as a real-valued vector, we also incorporate some abstract features. These features are derived from the Google Chinese Web 5-gram corpus (Liu et al., 2010) and will be referred to as "n-gram features".

4.1 Out-of-Vocabulary Indicator

This feature is simply a bit indicating whether a word is an out-of-vocabulary word or not. If a token never appears in the Web 5-gram corpus, the bit is set to 1; otherwise it is set to 0.

4.2 N-gram Probability Features

We compute the n-gram probability of each token using the occurrence count in the Web 5-gram corpus. We consider only up to trigrams since the probabilities are mostly zero when $n > 3$. Given the limited amount of available learner data, these probabilities may serve as useful features indicating how likely an expression is valid in Chinese.

5 Experiments

5.1 Dataset

We obtain the "wrong" part of the HSK dataset used in (Shiue and Chen, 2016). Each sentence segment has exactly one token-level position that is erroneous. Word segmentation and POS tagging are performed with the Stanford CoreNLP toolkit (Manning et al., 2014). We filter out any sentence segment whose corrected version differs from it by more than one token due to segmentation issue. That is, we only focus on the cases in which the error can be corrected by replacing one single token. After filtering, we end up with 10,510 sentence segments. We use 10% data for validation and testing respectively, and the remaining 80% data as the training set.

5.2 Evaluation

Accuracy

We use the detection accuracy as our main evaluation metric. A test instance is regarded as correct only if our system gives the highest score of incorrectness for the ground-truth position. This metric is relatively strict as the average length of the sentence segments in our dataset is 9.24. The McNemar's test is adopted to perform statistical significance test.

Mean Reciprocal Rank (MRR)

The mean reciprocal rank rewards the test instances for which the model ranks the ground-truth near the top of the candidate list. MRR is defined as $\frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank(i)}$, where N is the total number of test instances and $rank(i)$ is the rank of the ground-truth position of test instance i.

Hit@k Rate

The Hit@k rate regards a test instance as correct if the answer is ranked within the top k places. In the experiments, k is set to 2. We report this metric since one of the most common types of WUEs is collocation error. In example (E2), the problem involves a pair of words, i.e., the adjective 差 (bad) is not a suitable modifier of the noun 知識 (knowledge). (E4) and (E5) are both acceptable.

(E4) 學習 的 知識 也 很 不足

(The knowledge learned is also **insufficient**.)

(E5) 學習 的 態度 也 很 差

(The **attitude** of learning is also very bad.)

Which correction is better highly depends on the context or even the intended meaning in the writer's mind. If the model proposes two potentially erroneous tokens which are closely related to each other, it can be useful for Chinese learners.

Hit@r% Rate

Finding the exact position of the error could be more challenging in a longer sentence segment. We propose another hit rate measure which takes the segment length (len) into account. Specifically, we regard one test instance as correct if the answer is ranked within the top $\max(1, \lfloor len * r\% \rfloor)$ candidates. We report hit@20%. That is, for segments shorter than 10 tokens, the system is allowed to propose one candidate; for those whose length is between 10 and 14, the system is allowed to propose two, and so on. Equivalently, this measure judges whether our system can rank the ground-truth error position within the top 20%

Model	Features	Accuracy	MRR	Hit@2	Hit@20%
Random Baseline	-	0.1239	0.3312	0.2478	0.1611
LSTM	Rand. Init. Word Embedding	0.4186	0.6010	0.7222	0.6565
	CBOW	0.4072	0.5923	0.7155	0.6432
	CBOW + POS	0.4263	0.6150	0.7564	0.6908
	CBOW + POS + n-gram	0.4386	0.6204	0.7526	0.6755
	SG	0.4072	0.5910	0.7146	0.6365
	SG + POS	0.4301	0.6170	0.7593	0.6965
	SG + POS + n-gram	0.4386	0.6205	0.7507	0.6755
	CWIN	0.4853	0.6537	0.7774	0.7031
	CWIN + POS	0.4681	0.6435	0.7783	0.7022
	CWIN + POS + n-gram	0.4700	0.6502	0.7945	0.7269
	Struct-SG	0.4710	0.6412	0.7650	0.6889
	Struct-SG + POS	0.4757	0.6441	0.7593	0.6822
	Struct-SG + POS + n-gram	0.4881	0.6577	0.7840	0.7184
Bi-LSTM	CWIN	0.4795	0.6547	0.7840	0.7174
	CWIN + POS	**0.5138**	**0.6789**	0.8097	0.7479
	CWIN + POS + n-gram	0.4948	0.6719	**0.8173**	**0.7507**
	Struct-SG	0.4710	0.6412	0.7650	0.6889
	Struct-SG + POS	0.4757	0.6441	0.7593	0.6822
	Struct-SG + POS + n-gram	0.4948	0.6658	0.8040	0.7374

Table 1: Performance of the LSTM/Bi-LSTM sequence labeling models with different sets of features.

of the candidate list. This metric compromises Accuracy and Hit@k.

6 Results and Analysis

Table 1 shows the performance of our WUE detection models with different input features. The random baseline is a system randomly choosing one token as the incorrect position. The LSTM model using only randomly initialized word embeddings largely outperforms the random baseline. The pre-trained CBOW/SG word embeddings seem not very useful, leading to detection performance slightly lower than the model with random initial word embeddings. For both CBOW and SG, introducing the POS sequence improves the detection accuracy by about 2% and also improves all other measurements. The n-gram features further increase the accuracy by about 1%.

On the other hand, the CWIN and Struct-SG embeddings themselves are very powerful. Incorporating the POS and n-gram features leads to only slight improvements in terms of accuracy. Despite the small impact on accuracy, the n-gram features bring obvious improvements on hit@2 and hit@20% rates, indicating that they do facilitate the model in promoting the rank of the ground-truth position. Under the same set of features, all models with CWIN/Struct-SG significantly outperform their CBOW/SG counterparts ($p < 0.05$).

Bidirectional LSTM (Bi-LSTM) further enhance the performance of LSTM. The Bi-LSTM with CWIN+POS features achieves the best accuracy and MRR, and significantly outperforms its LSTM counterpart ($p < 0.005$). Bi-LSTM with CWIN+POS+n-gram features achieves the best Hit@2 and Hit@20%. To take a closer look, we analyze the performance of the two types of models on different length of segments in Table 2. We use the versions with all set of features and report hit@20% rates. Using Bi-LSTM leads to some improvement on short ($\leq$ 9 tokens) segments, and larger improvement on mid-length (10~14 tokens) ones. Even longer ($\geq$ 15 tokens) segments are relatively rare since foreign learners seldom construct complex sentences.

In Section 5.2 we justify the use of the hit@2 metric by pointing out that a WUE usually involves a pair of words dependent on each other. We can verify whether the top two candidates proposed by our model are closely related by examining the dependency distance. We take the output of the Bi-LSTM model with CWIN+POS+n-gram features and analyze the error cases where the model ranks the ground-truth error position second. We use the dependency parsing output of CoreNLP to construct an undirected graph, where

Length (# tests)	# proposed	LSTM	Bi-LSTM
$\leq$ 9 (645)	1	0.7426	**0.7659**
10~14 (317)	2	0.6908	**0.7319**
$\geq$ 15 (89)	$\geq$ 3	**0.7416**	0.7079

Table 2: Hit@20% rates of LSTM and Bi-LSTM on segments with different lengths.

# correct ($c_1 = a$)	520 (49.48%)
# tests where $c_2 = a$	339 (32.25%)
Average $dis(c_1, c_2)$ when $c_2 = a$	2.07
# tests where $c_2 = a$ and $dis(c_1, c_2) = 1$	129 (12.27%)

Table 3: Summary of the analysis of the dependency between the top two candidates proposed by the CWIN+POS+n-gram Bi-LSTM model. a denotes the ground-truth error position. c_1 and c_2 denote the first and the second candidate positions proposed by the model. $dis(c_1, c_2)$ is the distance between c_1 and c_2 on the dependency graph.

POS (# tests)	CWIN	CWIN+POS
VV (325)	0.8123	**0.8185**
NN (282)	0.6879	**0.7447**
AD (134)	0.6194	**0.7015**

Table 4: Hit@20% rates of Bi-LSTM models with or without POS features on three most frequent POS tags of the erroneous token.

each dependency corresponds to an edge, and calculate the shortest distance between the top two candidates in these cases. The results are summarized in Table 3. The average distance (2.07) is small compared to the average length of the segments (9.24), indicating that our model can consider the dependencies among words when ranking the candidate positions.

A factor that might limit the effectiveness of POS features is that the POS tagger trained on well-formed text may not perform well on noisy learner data. In fact, for 26.7% of the test data, the POS tag of the original erroneous token differs from that of its corrected version. We compare the performance of the model with or without POS features on three most frequent POS tags in Table 4. As can be seen, the POS information of the erroneous segment, which potentially contains errors, can still be helpful for detecting anomaly of the segment. In example (E6) we show the scores of incorrectness predicted by models with or without POS features. The ”DEC + AD” construction is invalid in Chinese, so in this case the error can be detected more easily if POS information is available.

(E6)	應該	有	別人	的	*盡力
POS tag	VV	VE	NN	DEC	**AD**
w/o POS	0.048	**0.226**	0.030	0.016	0.042
w/ POS	0.010	0.066	0.031	0.071	**0.077**

(There should be someone else’s ***utmost**.)

Word	Error rate	Precision	Recall
產生 (generate)	0.571 (8/14)	0.700 (7/10)	0.875 (7/8)
經驗 (experience)	0.500 (5/10)	0.667 (4/6)	0.800 (4/5)
發生 (happen)	0.455 (5/11)	0.571 (4/7)	0.800 (4/5)
而 (so)	0.417 (20/48)	0.550 (11/20)	0.550 (11/20)

Table 5: Precision/recall of Bi-LSTM models with CWIN+POS features on four most commonly misused ($err\_rate(w) > 0.4$) words.

In Table 5 we show the precision/recall of the Bi-LSTM model with CWIN+POS features on four most commonly misused words. The error rate of a word w is calculated on the test set by $err\_rate(w) = \frac{\text{\# segments in which } w \text{ is misused}}{\text{\# segments containing } w}$. We exclude words that occur in less than 10 segments regardless of their error rates. In general, our model achieves high recall and fair precision. Discriminating correct and wrong usage of the conjunction 而 (so), which often connects more than one segment, seems to be the most difficult. For example, in (E7) the inappropriateness of 而 cannot be recognized unless we consider the wider context of this segment.

(E7) (*而,並) 當成 此生 做人 的 道理
(..., (*so,and) take it as a lifelong way to
behave around others.)

7 Conclusion

In this paper we propose an LSTM-based sequence labeling model for detecting WUEs in sentences written by non-native Chinese learners. The experimental results suggest that the CWIN/Struct-SG embeddings, which consider word orders, are better word features for Chinese WUE detection. Moreover, Bi-LSTM is more preferred than LSTM. While a wrong usage often involves more than one token, making it difficult to determine which one should be corrected, the best model can rank the ground-truth error position within the top two in 80.97% of the cases.

One possible future direction is to exploit more sophisticated structural information such as dependency paths. Moreover, it is also worth studying how to extend our system to cope with the correction task.

Acknowledgments

This research was partially supported by Ministry of Science and Technology, Taiwan, under grants MOST-104-2221-E-002-061-MY3 and MOST-105-2221-E-002-154-MY3.

References

Shamil Chollampatt, Kaveh Taghipour, and Hwee Tou Ng. 2016. Neural network translation models for grammatical error correction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. pages 2768–2774.

François Chollet. 2015. Keras. https://github.com/fchollet/keras.

Robert Dale, Ilya Anisimoff, and George Narroway. 2012. Hoo 2012: A report on the preposition and determiner error correction shared task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics, pages 54–62. http://aclweb.org/anthology/W12-2006.

Robert Dale and Adam Kilgarriff. 2011. Helping our own: The hoo 2011 pilot shared task. In *Proceedings of the 13th European Workshop on Natural Language Generation*. Association for Computational Linguistics, pages 242–249. http://aclweb.org/anthology/W11-2838.

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

Hen-Hsen Huang, Yen-Chi Shao, and Hsin-Hsi Chen. 2016. Chinese preposition selection for grammatical error diagnosis. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, pages 888–899. http://aclweb.org/anthology/C16-1085.

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 (NLPTEA 2016)*. The COLING 2016 Organizing Committee, pages 148–154. http://aclweb.org/anthology/W16-4919.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

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 (NLPTEA 2016)*. The COLING 2016 Organizing Committee, pages 40–48. http://aclweb.org/anthology/W16-4906.

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 *Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA 2015)*. Association for Computational Linguistics, pages 1–6. https://doi.org/10.18653/v1/W15-4401.

Wang Ling, Chris Dyer, W. Alan Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)*. Association for Computational Linguistics, pages 1299–1304. https://doi.org/10.3115/v1/N15-1142.

Fang Liu, Meng Yang, and Dekang Lin. 2010. Chinese web 5-gram version 1. *Linguistic Data Consortium, Philadelphia* .

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, pages 55–60. https://doi.org/10.3115/v1/P14-5010.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)*. Association for Computational Linguistics, pages 746–751. http://aclweb.org/anthology/N13-1090.

Tou Hwee Ng, Mei Siew Wu, Ted Briscoe, Christian Hadiwinoto, Hendy Raymond Susanto, and Christopher Bryant. 2014. The conll-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. Association for Computational Linguistics, pages 1–14. https://doi.org/10.3115/v1/W14-1701.

Tou Hwee Ng, Mei Siew Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The conll-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*. Association for Computational Linguistics, pages 1–12. http://aclweb.org/anthology/W13-3601.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, pages 1532–1543. https://doi.org/10.3115/v1/D14-1162.

Marek Rei and Helen Yannakoudakis. 2016. Compositional sequence labeling models for error detection in learner writing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 1181–1191. https://doi.org/10.18653/v1/P16-1112.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45(11):2673–2681.

Yow-Ting Shiue and Hsin-Hsi Chen. 2016. Detecting word usage errors in chinese sentences for learning chinese as a foreign language. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA), pages 220–224.

Chi-Hsin Yu, Yi jie Tang, and Hsin-Hsi Chen. 2012. Development of a web-scale chinese word n-gram corpus with parts of speech information. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*. European Language Resources Association (ELRA), pages 320–324.

Liang-Chih Yu, Lung-Hao Lee, and Li-Ping Chang. 2014. Overview of grammatical error diagnosis for learning chinese as a foreign language. In *Proceedings of the 1st Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA 2014)*. pages 42–47.