Combining ResNet and Transformer for Chinese Grammatical Error Diagnosis

†‡Shaolei Wang, †Baoxin Wang, †Jiefu Gong, ‡Zhongyuan Wang, †Xiao Hu, †Xingyi Duan, †Zizhuo Shen, †Gang Yue, †Ruiji Fu, †Dayong Wu, ‡Wanxiang Che, †Shijin Wang, †Guoping Hu, ‡Ting Liu

†Joint Laboratory of HIT and iFLYTEK Research (HFL), iFLYTEK Research, Beijing, China

†State Key Laboratory of Cognitive Intelligence, iFLYTEK Research, China

‡Research Center for Social Computing and Information Retrieval (SCIR),
Harbin Institute of Technology, Harbin, China

{slwang9, bxwang2, jfgong, xiaohu2, xyduan, zzshen, gangyue, rjfu, dywu2, sjwang3, gphu}@iflytek.com, {slwang, zywang, car, tliu}@ir.hit.edu.cn

Abstract

This paper introduces our system at NLPTEA-2020 Task: Chinese Grammatical Error Diagnosis (CGED). CGED aims to diagnose four types of grammatical errors which are missing words (M), redundant words (R), bad word selection (S) and disordered words (W). The automatic CGED system contains two parts including error detection and error correction. For error detection, our system is built on the model of multi-layer bidirectional transformer encoder and ResNet is integrated into the encoder to improve the performance. We also explore stepwise ensemble selection from libraries of models to improve the performance of the single model. For error correction, we design two models to recommend corrections for S-type and M-type errors separately. In official evaluation, our system obtains the highest F1 scores at identification level and position level for error detection, and the second-highest F1 score at correction level.

1 Introduction

Chinese language is commonly regarded as one of the most complicated languages. Compared to English, Chinese has neither singular/plural change, nor the tense changes of the verb. In addition, word segmentation usually has to be processed before deeper analysis, since word boundaries are not explicitly given in Chinese. All these problems make Chinese learning challenging to new learners. In recent years, more and more people with different language and knowledge background have become interested in learning Chinese as a second language. It is necessary to develop an automatic Chinese Grammatical Error Diagnosis (CGED) tool to help to identify and correct grammatical errors written by these people.

In order to promote the development of automatic grammatical error diagnosis in Chinese learning, the Natural Language Processing Techniques for Educational Applications (NLP-TEA) have taken CGED as one of the shared tasks since 2014. Many methods have been proposed to solve CGED task.

In this work, we introduce our system at NLPTEA-2020 CGED task. For error detection, our system is built on the model of multi-layer bidirectional transformer encoder and ResNet is integrated into the encoder to improve the performance. We also explore stepwise ensemble selection from libraries of models to improve the performance of the single model. For error correction, we design two models to recommend corrections for S-type and M-type errors separately. More specifically, we use the RoBERTa (Liu et al., 2019) and the n-gram language model for the S-type correction, and utilize a combination of pretrained masked language model and a statistical language model to generate possible correction results for M-type correction. In official evaluation, our system obtains the highest F1 scores at identification level and position level for error detection, and the second-highest F1 score at correction level.

The paper is organized as follows: Section 2 briefly introduces the CGED shared task. Section 3 talks about our methodology. Section 4 shows the experiment result. Section 5 shows the related work. Finally, the conclusion and future work are drawn in Section 6.

2 Chinese Grammatical Error Diagnosis

The goal of NLPTEA CGED task is to indicate errors in the sentences written by Chinese Foreign Language learners. The sentences contain

Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications, pages 36–43

Error Type	Original Sentence	Correct Sentence
M	每个城市的超市能看到这些食品。	每个城市的超市**都**能看到这些食品。
R	我和妈妈**是**不像别的母女。	我和妈妈不像别的母女。
S	最重要的是**做**孩子想学的环境。	最重要的是**创造**孩子想学的环境。
W	“静音环境”**是对人体应该**有危害的。	“静音环境”**应该是对人体**有危害的。

Table 1: Typical Error Examples, where “M” means type of missing word, “R” means type of redundant word, “S” means type of word selection and “W” means type of disordered words.

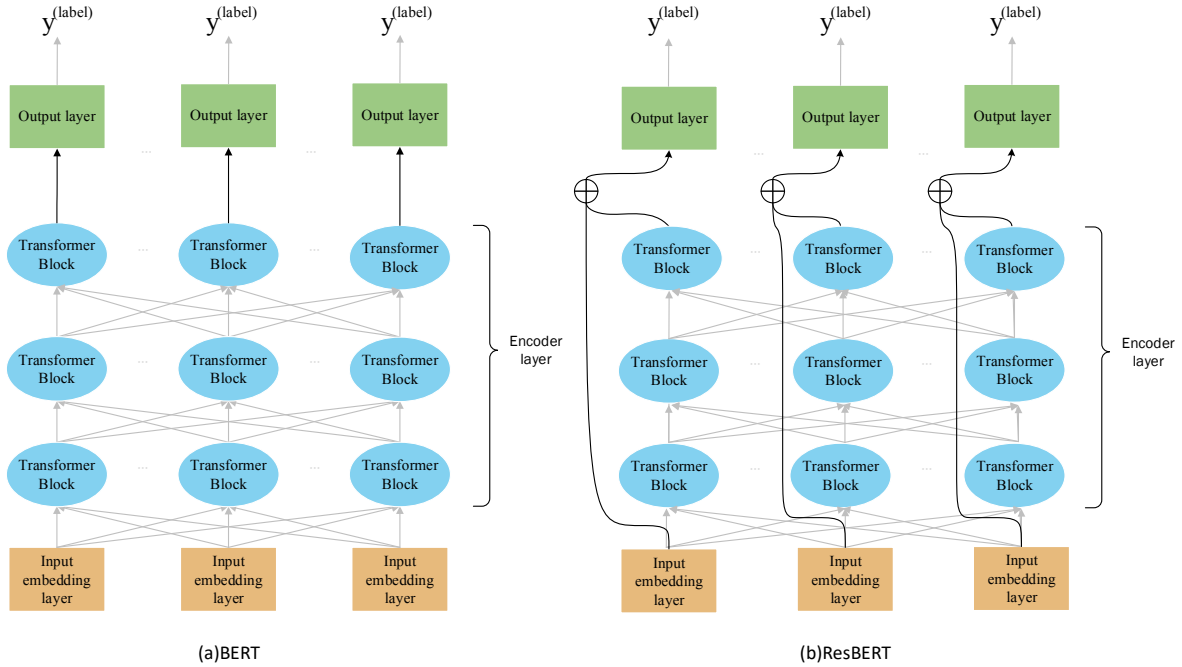

Figure 1: Architectures of BERT and ResBERT for grammatical error detection, where “BERT” means the multi-layer bidirectional transformer encoder.

four types of grammatical errors, including missing words (M), redundant words (R), word selection errors (S) and word ordering errors (W). The input sentence may contain one or more such errors. Given a sentence, the system needs to indicate: (1) If the sentence is correct or not; (2) What kind of errors the sentence contains; (3) The exact error position; (4) Possible corrections for S-type and M-type errors. Some typical examples are shown in Table 1.

3 Methodology

3.1 Error Detection

We treat the error detection problem as a sequence tagging problem. Specifically, given a sentence x, we generate a corresponding label sequence y using the BIO encoding (Kim et al., 2004). We then combine ResNet and transformer encoder to solve the tagging problem.

Transformer Encoder

We use the multi-layer bidirectional transformer encoder (BERT) described in Vaswani et al. (2017) to encode the input sentence. As shown in Figure 1(a), the model consists of three parts: an input embedding layer I, an encoder layer E and an output layer O. Given a sequence S = $w_0, ..., w_N$ as input, the encoder is formulated as follows:

$$h_i^0 = W_e w_i + W_p \quad (1)$$

$$h_i^l = \text{transformer\_block}(h_i^{l-1}) \quad (2)$$

$$y_i^{BERT} = \text{softmax}(W_o h_i^L + b_o) \quad (3)$$

where w_i is a current token, and N denotes the sequence length. Equation 1 thus creates an input embedding. Here, transformer_block includes self-attention and fully connected layers, and outputs

h_i^l. l is the number of the current layer, $l \geq 1$. L is the total number of layers of BERT. Equation 3 denotes the output layer. W_o is an output weight matrix, b_o is a bias for the output layer, and y_i^{BERT} is a grammatical error detection prediction.

Integrating ResNet

Deep neural networks learn different representations for each layer. For example, Belinkov et al. (2017) demonstrated that in a machine translation task, the low layers of the network learn to represent the word structure, while higher layers are more focused on word meaning. For tasks that emphasize the grammatical nature such as Chinese grammatical error detection, information from the lower layers is considered to be important. In this work, we use the residual learning framework (He et al., 2016) to combine the information from word embedding with the information from deep layer. Given a sequence S = $w_0, ..., w_N$ as input, ResBERT is formulated as follows:

$$h_i^0 = W_e w_i + W_p \tag{4}$$

$$h_i^l = \text{transformer\_block}(h_i^{l-1}) \tag{5}$$

$$R_i = h_i^L - w_i \tag{6}$$

$$H_i^L = \text{concat}(h_i^L, R_i) \tag{7}$$

$$y_n^{ResBERT} = \text{softmax}(W_o H_i^L + b_o) \tag{8}$$

Equation 6 denotes the residual learning framework, where the hidden output of h_i^L and the input embedding is used to approximate the residual functions. We then send the concatenation of h_i^L and R_i to the output layer.

Stepwise Ensemble Selection from Libraries of Models

We found that different random seeds and dropout values may result in different performances at the end of each training. It is straightforward to merge different model results to increase the performance. Rather than combine all the single models by weighted averaging, we use forward stepwise selection from the library of models (Caruana et al., 2004) to find a subset of models that yield excellent performance when averaged together. Library of models is generated using different random seeds and dropout values. The basic ensemble selection procedure is very simple:

1. Start with the empty ensemble.
2. Add to the ensemble the model in the library that maximizes the ensemble's performance to the Chinese grammatical error detection metric on validation set.
3. Repeat Step 2 for a fixed number of iterations or until all the models have been used.
4. Return the ensemble from the nested set of ensembles that has maximum performance on the validation set.

The voting system when selecting the best model to add at each step is span-level and it works as follow:

1. Each single model that tags a span of error text counts as a vote for that span of error text (e.g., if the word "是" in a given position, is tagged as an R-type by one single model, then it receives one vote). Note that only the spans of text that have been recognized as an error type by any of the single model are considered as candidates.
2. Each candidate span of error text is tagged as a true error if it collected a minimum number of votes, like 30% * number_of_subset_models.

The simple forward model selection procedure presented is effective, but sometimes overfits to the validation set, reducing ensemble performance on test set. To reduce the overfitting on the validation set, we make three additions to this selection procedure as described by Caruana et al. (2004):

Selection with Replacement. With model selection without replacement, performance improves as the best models are added to the ensemble, peaks, and then quickly declines. Selecting models with replacement greatly reduces this problem. Selection with replacement allows the models to be added to the ensemble multiple times. This allows selection to fine-tune ensembles by weighting models: models added to the ensemble multiple times receive more weight.

Sorted Ensemble Initialization. The simple forward model selection procedure starts with the empty ensemble. Forward selection sometimes overfits early in selection when ensembles are

small. To prevent overfitting, we sort the models in the library by their performance, and put the N best model in the ensemble before the procedure. We use N = 5.

Bagged Ensemble Selection. As the number of models in a library increases, the chances of finding combinations of models that overfit the validation set increases. Bagging can minimize this problem. We reduce the number of models by drawing a random sample of models from the library and selecting from that sample. If a particular combination of M models overfits, the probability of those M models being in a random bag of models is less than $(1-p)^M$ for p the fraction of models in the bag. We use p = 0.5, and bag ensemble selection 20 times to insure that the best models will have many opportunities to be selected. The final ensemble is the average of the 20 ensembles.

3.2 Error Correction

The systems are also required to recommend corrections for S-type and M-type errors. In this work, we design two different models to recommend corrections for S-type and M-type errors separately. We will describe them separately.

S-type Correction

For the S-type correction, we mainly use the RoBERTa (Liu et al., 2019) and the n-gram language model. Firstly, we perform domain adaptation on the language model. We use CGED training sets from previous competitions to fine-tune RoBERTa-wwm, and combine the CGED data with news corpora to train a 5-gram language model.

S-type correction includes single-character correction and multi-character correction. For the single-character correction, we consider the top 20 generated results of RoBERTa and 3,500 most frequent characters on L2 learner corpus as candidates. We score the candidates according to the prediction probability of RoBERTa and n-gram, visual similarity, and phonological similarity (Hong et al., 2019). Afterward, we select the character with the highest score as the correction result. For the multi-character correction, we also select the top 20 characters generated by RoBERTa at each position. We put these characters together to form words and reserved those in the vocabulary as candidates. In addition to the four kinds of features at the single-character correction, we also consider Levenshtein distance between the error words and candidate words.

	Error	R	M	S	W
Train	52,312	11,548	13,931	23,014	3,769
Validation	4,871	1,060	1,269	2,156	386

Table 2: Data statistics

M-type Correction

Specially, we consider the correction of M-type errors as a cloze task and utilize a combination of pretrained masked language model and a statistical language model to generate possible correction results. Given suspected missing positions, we divide the correction process of M-type errors into two steps, firstly offering possible corrections, then evaluating and picking the most reasonable ones.

When using pretrained masked language model, We first predict the number of missing characters at the suspected M-type error position through a BERT-based sequence labeling model. Then we add the same number of [MASK] symbols as predicted to the sentence before the position. Afterward, we use BERT to predict the most likely character of each [MASK] symbol, which is considered as correction candidates. When using statistical language models, we prepared a Chinese high-frequency vocabulary of L2 learners, and supplement all possible Chinese words from this vocabulary to the suspected M-type error position, generating a series of correction candidates. To evaluate the probability of each candidate, we use them to construct modified sentences and calculate the perplexity of the original sentence and all modified sentences using a statistical language model pretrained on L2 learner corpus. If the perplexity of modified sentence is significantly lower than the perplexity of the original sentence, which is controlled by a manual threshold, we consider the candidate as a predicted correction result.

4 Experiment

4.1 Dataset

Following the work of Fu et al. (2018), We trained our single models using training units that contain both the erroneous and the corrected sentences from 2016 (HSK Track), 2017 and 2018 training data sets. CGED 2016 HSK track training set consists of 10,071 training units with a total of 24,797 grammatical errors, categorized as redundant (5,538 instances), missing (6,623), word selection (10,949) and word ordering (1,687). CGED 2017 training set consists of 10,449 training units

model	FPR	Detection level			Identification			Position		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
BERT	0.6333	0.6974	0.8626	0.7713	0.5406	0.5721	0.5559	0.3362	0.3178	0.3267
BERT-WWM	0.6966	0.6826	0.8854	0.7709	0.5306	0.5894	0.5585	0.3324	0.3302	0.3313
ELECTRA	0.8530	0.6519	**0.9439**	0.7712	0.5185	0.6489	0.5764	0.3288	0.372	0.3491
ResELECTRA	0.7709	0.6680	0.9167	0.7728	0.5304	0.6520	0.5849	0.3503	0.396	0.3722
WA Ensemble	0.5675	0.7216	0.8962	0.7885	0.6175	0.5799	0.5981	**0.4871**	0.3841	0.4295
S Ensemble	**0.4333**	**0.7719**	0.8667	**0.8166**	**0.6411**	**0.6562**	**0.6486**	0.4805	**0.4693**	**0.4748**

Table 3: Validation Results using single models and ensemble methods. "S Ensemble" denotes for Stepwise ensemble model.

with a total of 26,448 grammatical errors, categorized as redundant (5,852 instances), missing (7,010), word selection (11,591) and word ordering (1,995). CGED 2018 training set consists of 1,067 grammatical errors, categorized as redundant (208 instances), missing (298), word selection (87) and word ordering (474). Table 2 shows the overall data distribution in the training data.

The sentences from 2017 testing data set are used for validation. It consists of 4,871 grammatical errors, categorized as redundant (1,060 instances), missing (1,269), word selection (2,156) and word ordering (386).

4.2 Metric

The evaluation method includes four levels:

Detection level. Determine whether a sentence is correct or not. If there is an error, the sentence is incorrect. All error types will be regarded as incorrect.

Identification level. This level could be considered as a multi-class categorization problem. The correction situation should be exactly the same as the gold standard for a given type of error.

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 recommends at most 3 correction at each error.

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

$$\text{FalsePositiveRate} = \frac{FP}{FP + TN} \tag{9}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{10}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{11}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{12}$$

$$\text{F1} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{13}$$

Since each team is allowed to submit three results, we run the stepwise ensemble selection for three times, according to the performance on detection level, identification level, position level separately.

4.3 Training Details

We try different pre-trained model parameters as the transformer's initialization such as BERT (Devlin et al., 2018), ELECTRA discriminator (Clark et al., 2020) and BERT-WWM (Cui et al., 2019). We find that the models initialized with ELECTRA discriminator always achieve better performance. So we select ELECTRA discriminator as the transformer's initialization. More concretely, we use Chinese ELECTRA-Large discriminator model[1] with 1024 hidden units, 16 heads, 24 hidden layers, 324M parameters.

For other parameters, we use streams of 128 tokens, a mini-batch of size 64, learning rate of 2e-5 and epoch of 120. We use 16 different random seeds and 5 different dropout values for each random seed to train 80 single models for the stepwise ensemble selection.

4.4 Validation Results

As shown in Table 3, we build five baseline systems including: (1) **BERT** means single model initialized with BERT (Devlin et al., 2018); (2)

[1]https://github.com/ymcui/Chinese-ELECTRA

runs	FPR	Detection level			Identification			Position		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
1	**0.1010**	0.9649	0.7409	0.8382	0.7769	0.4738	0.5886	0.4970	0.2529	0.3352
2	0.2573	0.9273	0.6213	0.6736	0.7356	0.6213	**0.6736**	0.4320	0.3514	0.3876
3	0.3257	0.9101	0.8800	0.8948	0.7320	0.6011	0.6601	0.4715	0.3536	**0.4041**
Best Team	0.0163	-	-	**0.9122**	-	-	0.6736	-	-	0.4041

Table 4: Error detection performances of Submitted Runs on Official Evaluation Testing data sets. "Best Team" row records the best scores among all participant teams at each task-specific evaluating metric.

runs	Correction Top1			Correction Top3		
	Precision	Recall	F1	Precision	Recall	F1
1	0.246	0.1149	0.1567	0.246	0.1149	0.1567
2	0.2105	0.1540	0.1779	0.2105	0.1540	0.1779
3	0.2290	0.1575	0.1867	0.2290	0.1575	0.1867
Best Team	-	-	**0.1891**	-	-	**0.1885**

Table 5: Error correction performances of Submitted Runs on Official Evaluation Testing data sets. "Best Team" row records the best scores among all participant teams at each task-specific evaluating metric.

BERT-WWM means single model initialized with BERT-WWM (Cui et al., 2019); (3) **ELECTRA** means single model initialized with ELECTRA discriminator (Clark et al., 2020); (4) **ResELECTRA** means single model with ResNet unit added; (5) **WA Ensemble** means simple weighed averaging ensemble model.

Table 3 shows the overall performances of our model on the 2017 test data. The ELECTRA single model achieves much better performance than both the BERT single model and the BERT-WWM single model. We conjecture that ELECTRA discriminator is trained without masked tokens, and this makes it more suitable for CGED task which is very sensitive to surrounding words. The ResELECTRA single model achieves more than 2 point improvements on position level over the baseline ELECTRA single model, which proves the effectiveness of integrating ResNet unit. The stepwise selection ensemble model achieves almost 10 point improvements on position level over the best ResELECTRA single model. Even compared with WA ensemble model, the stepwise selection ensemble model also achieves more than 4 point improvements.

4.5 Testing Results

Table 4 shows the performances on error detection. Our system achieves the best F1 scores at the identification level and position level. Although we achieve the highest position-level F1 score of 0.4041 among all teams, there still has a wide gap for our system to solve the Chinese grammatical error diagnosis.

Table 5 shows the performances on error correction. We achieve the second-highest correction top1 score. Since we only provide zero or one candidate word, our correction top1 score is the same as our correction top3 score.

5 Related Work

The researchers used many different methods to study the English Grammatical Error Correction task and achieved good results (Ng et al., 2014). Compared with English, the research time of Chinese grammatical error diagnosis system is short, the data sets and effective methods are lacking. Chen et al. (2013) still used n-gram as the main method, and added Web resources to improve detection performance. Lin and Chu (2015) established a scoring system using n-gram, and get better correction options. In recent years, Chinese grammatical error diagnosis has been cited as a shared task of NLPTEA CGED. Many methods are proposed to solve this task (Yu et al., 2014; Lee et al., 2015, 2016). Zheng et al. (2016) proposed a BiLSTM-CRF model based on character embedding on bigram embedding. 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

the probability of each characters, and used two 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. Their system achieved the best F1-scores in identification level and position level on CGED2017 task. Fu et al. (2018) added richer features on BiLSTM-CRF model such as word segmentation, Gaussian ePMI, combination of POS and PMI. They also adopted a probabilistic ensemble approach to improve system performance. Their system achieved the best F1-score in identification level and position level on CGED2018 task.

6 Conclusion and Future Work

The paper describes our system on NLPTEA-2020 CGED task, which combines ResNet and BERT for Chinese Grammatical Error Diagnosis. We also design two different ensemble strategies to maximize the model's capability. At all six evaluating levels, we have the best F1 scores in identification level and position level, the second-highest F1 score in correction top1 level, the third-highest F1 score in detection level. In the future, we are planning to build a more powerful grammatical error diagnosis system with more training data and try to improve the system's ability by using the different cross-domain corpus.

Acknowledgments

We thank the organizers of CGED 2020 for their great job. We also thank the anonymous reviewers for insightful comments and suggestions. This work was supported by the National Key R&D Program of China via grant 2018YFB1005100, and the National Natural Science Foundation of China (NSFC) via grant 61976072, 61632011 and 61772153.

References

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? *arXiv preprint arXiv:1704.03471*.

Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. 2004. Ensemble selection from libraries of models. In *Proceedings of the twenty-first international conference on Machine learning*, page 18.

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, Nagoya, Japan. Asian Federation of Natural Language Processing.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ruiji Fu, Zhengqi Pei, Jiefu Gong, Wei Song, Dechuan Teng, Wanxiang Che, Shijin Wang, Guoping Hu, and Ting Liu. 2018. Chinese grammatical error diagnosis using statistical and prior knowledge driven features with probabilistic ensemble enhancement. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 52–59.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Yuzhong Hong, Xianguo Yu, Neng He, Nan Liu, and Junhui Liu. 2019. Faspell: A fast, adaptable, simple, powerful chinese spell checker based on daedecoder paradigm. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 160–169.

Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. 2004. Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*, pages 70–75. Citeseer.

Lung Hao Lee, Gaoqi Rao, Liang Chih Yu, Endong Xun, and Li Ping Chang. 2016. Overview of the 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'16)*.

Lung-Hao Lee, Liang-Chih Yu, and Li-Ping Chang. 2015. Guest editoral: Special issue on chinese as a

foreign language. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 20, Number 1, June 2015-Special Issue on Chinese as a Foreign Language*.

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*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy 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*, pages 1–14.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yi Yang, Pengjun Xie, Jun Tao, Guangwei Xu, Linlin Li, and Luo Si. 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.

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*, pages 42–47.

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.