Integrating BERT and Score-based Feature Gates for Chinese Grammatical Error Diagnosis

Yongchang Cao[a] **Liang He**[a,b] **Robert Ridley**[a] **Xinyu Dai**[a]
[a]National Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing, 210023, China
[b]Linguistic Intelligence and Knowledge Engineering Research, Nanjing, China
{caoyc, heliang, robertr}@smail.nju.edu.cn
daixinyu@nju.edu.cn

Abstract

This paper describes our proposed model for the Chinese Grammatical Error Diagnosis (CGED) task in NLPTEA2020. The goal of CGED is to use natural language processing techniques to automatically diagnose Chinese grammatical errors in sentences. To this end, we design and implement a CGED model named BERT with Score-feature Gates Error Diagnoser (BSGED), which is based on the BERT model, Bidirectional Long Short-Term Memory (BiLSTM) and conditional random field (CRF). In order to address the problem of losing partial-order relationships when embedding continuous feature items as with previous works, we propose a gating mechanism for integrating continuous feature items, which effectively retains the partial-order relationships between feature items. We perform LSTM processing on the encoding result of the BERT model, and further extract the sequence features. In the final test-set evaluation, we obtained the highest F1 score at the detection level and are among the top 3 F1 scores at the identification level.

1 Introduction

Recently, with the continuous development of China, more and more people have begun to learn Chinese as their second language. Due to the many complexities of Chinese, such as the differences in how tenses are formed in Chinese and English, many learners mistakenly write many Chinese sentences with grammatical errors when they first learn Chinese. Therefore, it is necessary to develop a CGED system, which can not only improve the learning efficiency of Chinese learners, but also serve many downstream tasks based on Chinese corpora.

Compared with English grammatical error diagnosis, Chinese grammatical error correction has received limited interest in the research community. English grammar error detection models began being developed as early as the 1980s, such as the early Writer's Workbench system (Macdonald NH, 1983) for detecting punctuation errors and style errors. Later, a series of tasks for English grammatical error detection and correction were proposed, such as CoNLL-2013 (Ng et al., 2013) and CoNLL-2014 (Ng et al., 2014). With the release of the CGED task in the NLPTEA workshop in recent years, grammar diagnosis models for Chinese have also begun to be developed.

The goal of the CGED task is to use natural language processing techniques to diagnose grammatical errors in Chinese sentences written by learners who use Chinese as a second language. The CGED task allows researchers to exchange experiences and ultimately promote the development of this shared task. It defines four types of Chinese grammatical errors, which are: redundant words (denoted as a capital "R"), missing words ("M"), word selection errors ("S"), and word ordering errors ("W"). The system developed for this task needs to identify the type and location of the errors in the input sentence.

Most recent solutions to the CGED shared task convert the problem into a sequence labeling problem and use a BiLSTM-CRF-based architecture as a basic framework to train the model. However, in previous work, feature engineering for the input sequence has become more and more complex. In addition, for some score-based features which exhibit partial-order relationships, such as the commonly used PMI Score features, previous works usually learn their embedding matrix after discretizing the scores. Through this process, the partial-order relationships between items will be lost, and the dimensionality of the feature embedding matrix will gradually increase as the granularity of the score

Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications, pages 49–56
Suzhou, China, December 4, 2020.

discretization becomes finer, increasing the number of parameters needed to be trained. In response to the above problems, we design and implement BERT with Score-feature Gates Error Diagnoser (BSGED), and integrate score-based features through the use of a gating mechanism, which not only greatly reduces the workload of feature engineering, but also retains the original partial-order relationships for score-based features. Experiments verify that the BSGED model achieves excellent results with less feature engineering.

In summary, our contributions are as follows:

- We propose a novel model BSGED for the CGED task, which achieves better results with fewer prior features and greatly reduces the workload of feature engineering.
- We propose a gating mechanism for integrating score-based features, which not only preserves the partial-order relationships between feature items, but also greatly reduces the amount of model training parameters.
- Through ablation experiments, we verify the effectiveness of adding a BiLSTM layer to further improve the model's ability to capture long-term dependencies of input sequences.

2 Related Work

Grammatical error diagnosis models appeared as early as the 1980s. Early grammatical error diagnosis models used rule-based methods to check and correct grammatical errors (Naber D, 2003). However, because the design of matching rules requires rich linguistic knowledge, it has become more and more difficult as well as time-consuming to design rules for such models.

In order to deal with more complex error types, a series of grammatical error detection and correction models based on machine translation technology have been proposed. Brockett et al. (2006) proposed a model that uses Statistical Machine Tranalation (SMT) techniques to detect and correct grammatical errors, which deal with mass/count noun confusions by translating the incorrect phrases as a whole. Felice et al. (2014) proposed a model for grammatical error diagnosis which combines rule-based and SMT systems in a pipeline. The model first uses rules to detect errors and generate candidates. After the candidates are roughly screened by the n-gram language model, they are sent to the SMT model for further screening. In the end, candidates will be further selected through language models and filtering rules. In order to solve the CGED2018 shared task, Hu et al. (2018) proposed a sequence-to-sequence network to model the problem, and used a semi-supervised method to generate pseudo-grammatical error data for training the model.

Models based on machine translation require a large-scale training corpus to train the model. Inspired by the powerful capabilities of Neural Machine Translation (NMT) in grammatical error diagnosis, Zheng et al. (2016) regarded CGED as a sequence labeling problem, and used the powerful feature learning ability of an LSTM network to model the input sequence, and achieved better results. Yang et al. (2017) incorporated more grammatical features into the model based on the BiLSTM-CRF framework. Based on the LSTM-CRF error detection model, Li et al. (2018) combined three error correction models: a rule-based model, an NMT GEC model, and an SMT GEC model. The three GEC models aid the BiLSTM-CRF model in marking possible error locations during the detection phase. Fu et al. (2018) designed a model that incorporates richer features and added a template matcher and probability fusion mechanism.

3 Methodology

3.1 Baseline Model

Similar to most previous models for CGED shared tasks, we treat the CGED problem as a sequence labeling problem, and use BiLSTM-CRF as the basic framework of BSGED. Specifically, for a given input sequence s_i, which consists of a character sequence $[c_1, c_2, \ldots, c_n]$, BSGED will output an equal-length sequence Y_i, which is composed of a label sequence $[y_1, y_2, \ldots, y_n]$ composition. We adopt the BIO marking strategy, that is, for characters without grammatical errors, we mark them as 'O', and for a subsequence of grammatical errors, such as word selection errors, the initial characters will be marked as 'B-S', The remaining single characters will be marked as 'I-S'.

Inspired by previous work, we use a BiLSTM network as the RNN unit to obtain the input character encoding sequence. The BiLSTM network has a strong ability to capture long-term dependencies of the input sequence. CRFs are widely used in a large number of natural language processing tasks, especially sequence-annotation tasks. With the addition of a CRF, the BiLSTM-CRF model can predict the input sequence more accurately. For example, the BiLSTM-CRF model can avoid incorrect sequence predictions beginning with "I-X". In terms of feature selection, we select some simple features, such as the POS tag sequence, POS Score, and PMI Score. Different from previous work, BSGED adopts the BERT model as the character encoder of the input sequence, and uses a novel fusion mechanism to incorporate score-based features. The model details are introduced in the next section. The framework of the base model adopted by BSGED is shown in Figure 1.

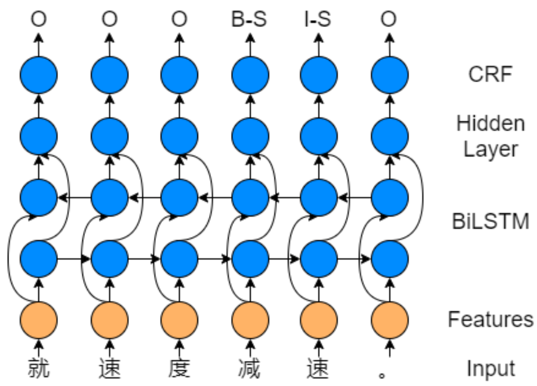

Figure 1: The base model of the BiLSTM-CRF framework used by BSGED

3.2 BERT-Encoder and Gating mechanism

Unlike previous models based on the BiLSTM-CRF architecture, BSGED does not utilize overly complex feature engineering, but uses the novel BERT model to obtain a token embedding representation of the input sequence. As a pre-trained language model, BERT has been successfully applied to many natural language understanding tasks, such as Chinese spelling error correction (Zhang et al. 2020). Due to its powerful semantic extraction capabilities, we utilize BERT as a semantic feature extractor, converting characters into vector representations. In order to preserve the long-term dependencies on the input sequence better, BSGED takes the final layer output of the BERT model as part of the BiLSTM input, instead of concatenating it with the output results of the other features through the BiLSTM network. Experiments verify that this operation can further improve the overall performance of BSGED.

We use prior knowledge to calculate the POS features of the input sequence and the PMI features between adjacent words. Specifically, we first use the LTP word segmentation tool[1] to perform word segmentation processing on the input sequence, and then perform part-of-speech tagging on the segmented sequence. This step also makes use of the LTP library. We also integrate location information into the POS tags. For example, for a Chinese sequence $A_1A_2A_3B_1B_2C_1$, the segmented sequence should be $A_1A_2A_3$ - B_1B_2 - C_1. Assuming the POS information of word A, word B, and word C are c, p, r respectively, then the result of POS labeling should be $B_cI_cI_c$ - B_pI_p - B_r.

For the score-based features, we use the news corpus provided by SogouCS[2] as a large corpus to obtain prior-knowledge statistics. Similar to the approach of Yang et al. (2017), for the POS Score feature, we first count the discrete probability distribution of the POS feature of each word, and use the probability value as its POS Score. Similarly, we count the co-occurrence frequency between every two words on the same large corpus, and use the normalized co-occurrence frequency score as the PMI Score of two adjacent words. It should be noted that we also merge the character position information in the vocabulary into these feature items.

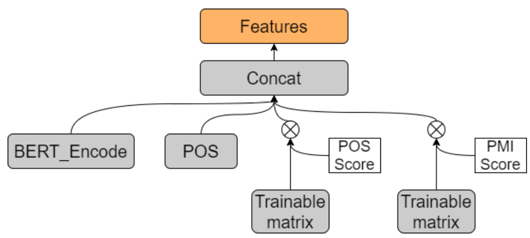

Figure 2: Schematic diagram of the features used in BSGED

We propose a novel fusion mechanism for score-based features. For continuous score features, traditional models usually discretize them first, and then embed the discretized score into a low-dimensional space. However, this embedding method will lose the partial-order relationships between the scores. In addition, the size of the feature space will change with the discretization granularity and the original value range of the score, and the model will have more parameters to be trained. Our approach differs in that we retain the continuity of

[1] https://github.com/HIT-SCIR/ltp

[2] https://www.sogou.com/labs/resource/cs.php

score features and train a matrix $\boldsymbol{M}_f \in \mathbb{R}^{2*D}$ for each score-based feature, where D is the preset embedding matrix dimension. For the i -th character, the final score embedding vector is as follows:

$$emb_i = \boldsymbol{M}_i[pos_i] * score_i \tag{1}$$

Where emb_i is the final embedding representation and pos_i is the position information of the character, $pos_i = 0$ for a "B-Word", and $pos_i = 1$ for an "I-Word". At this point, the role of score-based features is similar to an input gate (Hochreiter and Schmidhuber, 1997). This strategy not only preserves the partial-order relationship of score features, but also greatly reduces the size of the parameter matrix. The composition structure of the features for the input sequence is shown in Figure 2.

3.3 Ensemble mechanism

Following our experiments, we find that for different initialization parameters, the prediction results of the model are highly variable. This observation is consistent with that of Yang et al. (2017). In order to further improve the performance, we train multiple single models and use an ensemble mechanism to fuse them together. We adopt a simple and effective voting mechanism as our ensemble method, which improves the precision of the model while preserving the recall value.

In our final version, we use a total of four parameter groups, and we select 4 random factors for each group, so we finally merge 16 single models.

3.4 Post-Processing

The ensemble mechanism may produce conflicting prediction results. To solve this problem, we perform post-processing operations on the results of the ensemble model. We adopt some rule-base schemes, which integrate prior knowledge simply and effectively. The main processing methods are as follows:

First, in cases when some single models predict a sentence to be correct and other single models predict it to be incorrect, the conflict is resolved by retaining the prediction 'incorrect'. The 'correct' label is only output when *all* models predict the sentence as 'correct'.

Second, we resolve 'incorrect' predictions with overlaps, such as when the following two predictions are output for sentence s_i:

$$\begin{cases} < b_1, e_1, type_1 > \\ < b_2, e_2, type_2 > \end{cases} \tag{2}$$

Where b is the starting position of the prediction, e is the ending position, and $type$ is the predicted error type. When Equation 3 is established, BSGED believes that the two prediction results overlap.

$$\begin{cases} type_1 = type_2 \\ b_1 \in (b_2,\ e_2) \vee b_2 \in (b_1,\ e_1) \end{cases} \tag{3}$$

When overlapping occurs, the model uses the segmentation boundary of the original sentence to filter. Suppose that the word segmentation boundary of sentence s_i is $D = [d_1, d_2, \dots, d_j, \dots d_n]$, that is, $s_i[d_{j-1}: d_j]$ represents a word of the sentence. The model will retain the prediction result of $< b_i, e_i, type_i >$ which is more suitable for D.

4 Experiment

4.1 Data Preparation

We use all the data from the CGED2015-CGED2018 training and test sets, as well as the training data from CGED2020. More specifically, our training data consists of the following parts: all data from the CGED2015-2016 training set and test set, all data from the CGED2017-2018 training set, 50% of the CGED2017-2018 test set, and 20% of the CGED2020 training set. The validation set consists of 50% of the CGED2017-2018 test set and 80% of the CGED2020 training set.

Since the training set of CGED2020 has the same data as the test set from CGED2017-2018, in order to prevent data leakage, we de-duplicate the training set. Following de-duplication, the training set contains 43925 samples, and the validation set contains 3843 samples.

4.2 BERT Selection

Since richer model initialization parameters result in more diverse predictions, thereby further improving the recall rate of the model after ensembling, we therefore choose two different BERT pre-training parameters to initialize our model.

In addition to using the BERT-Base-Chinese version released by Google (Devlin et al., 2018), we also use another version of Chinese BERT. In order to further promote the research and development of Chinese information processing, the HFL team released the Chinese pre-training model BERT-wwm (Cui et al., 2019), which uses a Whole Word Masking technique, as well as models closely related to this technology: BERT-wwm-ext.

BERT-wwm is trained on Chinese Wikipedia (including simplified and traditional characters) and LTP is used to perform word segmentation before masking is carried out on all Chinese characters that make up the same word. Similar to other BERT-based models, it has 12 layers, 768 hidden size, and 12 self-attention heads.

Filter threshold	Detection Level			Identification Level			Position Level		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
1	0.7013	**0.9633**	0.8117	0.4683	**0.851**	0.6041	0.2091	**0.5419**	0.3017
4	0.8115	0.8254	**0.8184**	0.6347	0.608	**0.6211**	0.4255	0.3751	**0.3987**
10	**0.8881**	0.5408	0.6722	**0.7733**	0.3511	0.4829	**0.622**	0.2247	0.3301

Table 1: Performance of the BSGED on the validation set with different filtering thresholds

	Detection Level			Identification Level			Position Level		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
Run #1	0.8565	**0.9757**	**0.9122**	0.5571	**0.8432**	0.6709	0.2097	**0.4648**	0.2890
Run #1	0.9303	0.8478	0.8872	0.7018	0.5779	0.6339	0.4008	0.288	0.3351
Run #1	0.9739	0.5513	0.7041	**0.7939**	0.2975	0.4328	**0.5757**	0.1519	0.2404
Best Team	0.9875	0.9757	0.9122	0.7939	0.8432	0.6736	0.5757	0.4648	0.4041

Table 2: The performance of the three submissions on the official evaluation test data set. The scores in bold represent the best scores we obtained among all the participating teams. The "Best Team" row records the best scores among all participating teams for each task-specific evaluating metric.

Filter threshold	Pre	Rec	F1
1	0.2091	**0.5419**	0.3017
2	0.307	0.4603	0.3683
3	0.3754	0.4149	0.3942
4	0.4255	0.3751	**0.3987**
5	0.4678	0.3439	0.3964
6	0.5071	0.3214	0.3934
7	0.5356	0.2932	0.3789
8	0.569	0.2692	0.3655
9	0.5988	0.2475	0.3502
10	**0.622**	0.2247	0.3301

Table 3: The influence of filtering threshold on the performance of the ensemble model

4.3 Validation Results

We select the model parameters through the validation set results, which mainly include the selection of the filtering threshold during model integration. Since BSGED uses a total of 16 single models for integration, we first simply set the max filtering threshold to 10, and explore the performance of the model after integration within this range. The performance of the model on the validation set is shown in Table 3 and Figure 3. It should be noted that when selecting parameters, we only paid attention to the performance of the model at the position level.

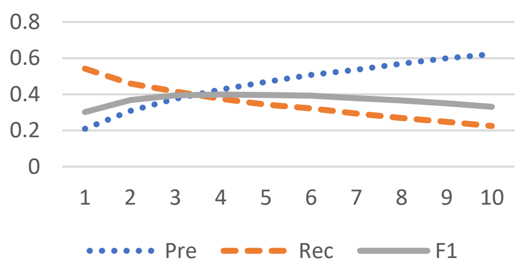

Figure 3: The influence of filtering threshold on precision, recall and F1 value.

It can be seen that as the filtering threshold increases, as does the precision, and the resulting predictions are more reliable; and as the filtering threshold decreases, the recall rate of the results will increase, enabling the model to be able to cover more actual errors. A low threshold will encourage retention of a large number of over-detection errors, while a high threshold will filter out partially correct results during post-processing. When the filter threshold is in the middle of the range, the model can achieve a higher F1 value.

Finally, we select three fusion models by selecting the parameter group with the highest precision rate, the parameter group with the highest recall rate, and the parameter group with the highest F1 value. The results on the validation set are shown in Table 1.

4.4 Testing Results

The final version of BSGED obtained the top F1 score at the detection level and was among the top 3 F1 scores at the identification level on the test set released by CGED2020. In addition, BSGED obtained the highest precision rate and recall rate among all error diagnosis evaluation levels except the precision rate at the Detection Level. The specific results are shown in Table 2.

Single Models Num	**Type**	**Avg. Detection Level**			**Avg. Identification Level**			**Avg. Position Level**		
		Pre	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**
7	embed	0.8416	0.694	0.7599	0.6609	0.4344	0.5238	0.4204	0.2249	0.2927
	gating	0.8264	0.7342	0.7772	0.6468	0.4958	0.5609	0.4164	0.2703	0.3275

Table 4: The influence of the gating mechanism on the model's results on the validation set. The value is the average of 7 models.

Single Models Num	**Type**	**Avg. Detection Level**			**Avg. Identification Level**			**Avg. Position Level**		
		Pre	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**
4	BERT -CRF	0.8298	0.7171	0.7691	0.6487	0.4575	0.5361	0.4093	0.2380	0.3006
	BSGED	0.8174	0.7556	0.7850	0.6344	0.5141	0.5677	0.4010	0.2765	0.3271

Table 5: The influence of the BiLSTM layer in the BSGED on the model's results on the validation set. The value is the average of 4 models

4.5 Ablation Experiment

In order to evaluate the novel components of our approach, we conduct two sets of ablation experiments.

The first set of ablation experiments focuses on the gating mechanism. We use 7 parameter groups from the 16 parameter groups from our original experiments. 7 single models use the traditional discretized embedding method for score-based features, and 7 single models used the novel gating approach we propose. The final comparison results are shown in Table 4. The results show that the control group that uses the gating mechanism achieves higher F1 values at each level of error detection; the performance improvements at the detection level, identification level, and position level are 0.0173, 0.0371 and 0.0348 respectively, demonstrating the effectiveness of the gating mechanism.

The second set of ablation experiments shows the performance improvement brought about by the addition of the BiLSTM layer compared to the BERT-only model. Through connecting the encoded output of the BERT model to the BiLSTM layer, the model can further improve its ability to capture the long-term dependencies of the input sequence. We conduct an experimental comparison of the model with and without the connected BiLSTM layer. For this experiment, 4 single models use a BERT-CRF architecture, and 4 single models connect the BERT output to a BiLSTM (BSGED). The two single model groups use the same parameter settings. The comparison result is shown in Table 5. As can be seen, the control group with the addition of the BiLSTM achieves F1 value improvements of 0.0159, 0.0316, and 0.0265 at the detection level, identification level, and position level, demonstrating the effectiveness of the BiLSTM layer.

4.6 Case Study

We found that different optimizations enable BSGED to solve different types of errors better. Among them, the gating mechanism directly retains the partial-order relationships of the original score-based features, so it has an improved ability for recognizing errors at character- or word-level. Some examples are shown in Table 6. For example, in the first sentence in Table 6, "多爱" (meaning "much love") should be identified as being incorrect, with the correct phrase being "最爱" (meaning "favorite"). Similarly, "沿" (meaning "along") and "没" (meaning "no") are words formed with similar strokes. In the second example sentence, "沿" should be replaced with "没", because the PMI score of "沿有" is extremely low. In the third sentence, "速度减速" is a word-level error, and the correct expression should be "速度减慢".

The addition of the BiLSTM layer enables the model to better capture the long-term dependencies of the input sequence so that the model has stronger processing capabilities for error samples that rely on semantic understanding and long-term dependencies. Some examples are shown in Table 7. For example, in the first sentence, "在…去" should be identified as an incorrect expression in Chinese, with the correct structure being "到…去". Identifying this error that requires judging long-term dependencies of the text. Finally, the phrase "首歌" in the second example is a common collocation, but in the example, through the semantic understanding of the last clause, "首" should be identified as an R type error.

Original Sentence	**Detect Result**
我的多爱的画家也画抽象的画儿。	3, 3, S (多)
在前面，故事沿有什么特别，就是在音乐学校一个男生和一个女生交朋友。	7, 7, S (沿)
12 回合结束后，就速度减速。	12, 13, S (减速)
世界里有很多挑选新能源。最主要的生态学的能源有是：太阳能，风能，潮汐能，还有地热能。	3, 3, S (里)
首开先我们应该自问什么是成熟。对我来说，成熟就是成为负责的人，对生活的情况和问题发展自己的思考。	2, 2, R (开)

Table 6: Some examples of errors that the gating mechanism can identify but the baseline model cannot

Original Sentence	**Detect Result**
去年我们决定在挪威去。我们已经乘船去过一次挪威了。很喜欢这次航行的起点是阿姆斯特丹。	7, 7, S (在)
再说，我认为愚公当英雄，因为我们对他很尊重。香港的音乐组，他叫张华，写了一个愚公首歌。	41, 41, R (首)
星期二上午我去在大学。我学习、和我上课。下午我休息和学习在家里。星期三早上我上汉语果。	8, 8, R (在)

Table 7: Some examples of errors that the model with BiLSTM layer can identify but the baseline model cannot

5 Conclusion and Future Work

This paper describes our novel BSGED model for the CGED2020 shared task, which uses only a few and simple features, greatly reducing the workload of feature engineering for CGED; a gating mechanism is also proposed to retain the original partial-order relationships between score-based features and at the same time reduce the amount of model training parameters. In addition, we connect the sequence encoding result of the BERT model to the BiLSTM layer, which improves the BSGED model's ability to capture long-term dependencies of the input sequence. BSGED achieves the best F1 score at the detection level and the third highest F1 score at the identification level.

In the future, we intend to use the MLM model to build a model that includes grammatical error correction, and apply the natural language generation capabilities of the pre-trained language model to the task of correcting Chinese grammatical errors. In addition, we will also integrate more explicit grammatical rules, which will also greatly help the improvement of model performance.

Acknowledgments

Special thanks to the NLP-TEA workshop for sharing work, which allows us to discuss technologies and jointly promote the development of solutions.

References

Chris Brockett, William B Dolan, and Michael Gamon. 2006. Correcting ESL errors using phrasal SMT techniques. In *Proceedings of the 21stInternational Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 249–256 https://www.aclweb.org/anthology/P06-1032/

Cui Y, Che W, Liu T, et al. *Pre-training with whole word masking for chinese bert*[J]. arXiv preprint arXiv:1906.08101, 2019.

Devlin J, Chang M W, Lee K, et al. *Bert: Pre-training of deep bidirectional transformers for language understanding*[J]. arXiv preprint arXiv:1810.04805, 2018. http://dx.doi.org/10.18653/v1/N19-1423

Felice M, Yuan Z, Andersen Ø E, et al. *Grammatical error correction using hybrid systems and type filtering*[C]. Association for Computational Linguistics, 2014. http://dx.doi.org/10.3115/v1/W14-1702

Fu R, Pei Z, Gong J, et al. Chinese grammatical error diagnosis using statistical and prior knowledge driven features with probabilistic ensemble enhancement[C]//*Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*. 2018: 52-59. http://dx.doi.org/10.18653/v1/W18-3707

Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

Hu Q, Zhang Y, Liu F, et al. Ling@ CASS Solution to the NLP-TEA CGED Shared Task 2018[C]//*Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*. 2018: 70-76. http://dx.doi.org/10.18653/v1/W18-3709

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The conll2013 shared task on grammatical error correction. https://www.aclweb.org/anthology/W13-3601/

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. http://dx.doi.org/10.3115/v1/W14-1701

Li C, Zhou J, Bao Z, et al. A hybrid system for Chinese grammatical error diagnosis and correction [C]//*Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*. 2018: 60-69. http://dx.doi.org/10.18653/v1/W18-3708

Macdonald N H. Human factors and behavioral science: The UNIX™ Writer's Workbench software: Rationale and design[J]. Bell System Technical Journal, 1983, 62(6): 1891-1908.

Naber D. *A rule-based style and grammar checker*[J]. 2003.

Yang Y, Xie P, Tao J, et al. Alibaba at IJCNLP-2017 task 1: Embedding grammatical features into LSTMs for Chinese grammatical error diagnosis task[C]//*Proceedings of the IJCNLP 2017, Shared Tasks*. 2017: 41-46.

Zhang S, Huang H, Liu J, et al. *Spelling Error Correction with Soft-Masked BERT*[J]. arXiv preprint arXiv:2005.07421, 2020. http://dx.doi.org/10.18653/v1/2020.acl-main.82

Zheng B, Che W, Guo J, et al. Chinese grammatical error diagnosis with long short-term memory networks[C]//*Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016)*. 2016: 49-56. https://www.aclweb.org/anthology/W16-4907/