Grammatical Error Detection with Self Attention by Pairwise Training

Quanbin Wang and Ying Tan
Key Laboratory of Machine Perception (MOE)
Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University
Beijing, 100871, China
Email: {qbwang362, ytan}@pku.edu.cn

***Abstract*—Automatic grammatical error detection system is useful for language learners to identify whether the texts written by themselves have errors. Researches have paid more attention on different models to deal with this task, various approaches have been proposed and better results have been achieved compare with rules base methods. It is known that artificially generated incorrect texts can further improve the performance of grammatical error correction and pairwise training is necessary for many recommendation algorithms. We incorporating these two techniques together to solve the error detection task with pre-trained words embeddings from BERT in this paper. It is the first work that adopt pairwise training with pairs of samples to detect grammatical errors since all previous work were training models with batches of samples piontwisely. Pairwise training is useful for models to capture the differences within the pair of samples, which are intuitive useful for model to distinguish errors. Extensive experiments have been carried out to prove the effectiveness of pairwise training mechanism. The experimental results shown that the proposed method can achieve the state of the art performance on four different standard benchmarks. With the help of data augmentation and filtering, the value of $F_{0.5}$ can be further improved. The overall improvements among the four test set are around 2.5% which demonstrate the generality of pairwise training for datasets from differen domains.**

***Index Terms*—pairwise training, grammatical error detection, BERT, data augmentation**

I. Introduction

Grammatical error detection (GED) is one of the key component in grammatical error correction (GEC) community. Once upon a time, it was the first step of GEC, since many GEC approaches are based on hand crafted rules. Those rules are different for various grammatical error types, detecting errors in the given text is the basis of the correction system. There are still some kinds of neural networks based GEC methods are designed for specific error types [1], it is essential to build a useful GEC system for those kinds of model. For language learners, an effective and automatic GED system is useful for them to find whether the texts written by themselves have errors or not. Especially for second language learners, the reliable tool would be a powerful auxiliary for them to master foreign languages.

The mainstream approaches for error detection are all based on deep neural networks, including recurrent neural networks and Transformer [2], in supervised learning, semi-supervised learning and multi-task learning settings, like classical methods used in traditional text classification [3]. Researchers coped with this problem like a sequence labelling task [4], predict the correct or incorrect label for each word in the given text. Pre-trained model have attracted much more attentions because of its ability in image classification tasks, many pre-trained word embeddings are adopted in a variety of natural language processing (NLP) tasks. The most representative model is BERT [5], which had achieved superior performance on 11 different task. On account of this, pre-trained word embedding like BERT and semi-supervised with large unlabeled corpus are popular used in many NLP problems. The current state of the art models for text error detection were trained sequence labeler incorporating pre-trained BERT with Transformer and large scale unlabeled data, like works in [6] and [7].

Like common used for error correction, using synthetic error texts to augment training data can further improve the performance of some GEC system. Some attempts had been investigated for error detection, methods proposed in [8] and [9] obtained better results with the help of large amount of artificially generated data.

In this paper, we further exploit the pre-trained word embedding and large scale synthetic error texts for grammatical error detection task with pairwise training mechanism. Different from the work in [6] and [7], [6] study the effectiveness of contextual embeddings conducted by different pre-train methods adequately, including BERT, ELMo [10] and flair embeddings [11]. [7] investigate how to take the full advantage of BERT by utilize information not only from the final layer but also from intermediate layers. Our model just use BERT as a weight initialization and fine-tuned with the augmented corpora. Other than [8] and [9], we not only use all error texts but also the corresponding right texts to conduct pairwise training. The large amount of fake data is generated by two ways in this work, the first one like back translation which translate right text to wrong with sequence to sequence framework and another one is rule based approach. To the best of our knowledge, this is the first attempt which encouraging the model to distinguish error tokens from right ones by training the model with right and error pairs explicitly. The experimental results demonstrate that our pairwise training mechanism can achieve the state of the art performance when incorporating character level embedding and word level pre-

Ying Tan is the corresponding author.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

trained embedding together with synthetic data.

This paper is organized as follows. We describe some related works of GED and GEC in section II. In section III, we will represent the corpora used in our work, including true data like First Certificate in English (FCE) [12] and synthetic data generated by ourselves. And then, a brief introduction of BERT and the details of our method will be given in section IV. The experimental results are discussed in section V. As a result, we make conclusion in section VI.

II. Related Work

The earliest work in GEC and GED can back in the early 90s, many well designed rules were widely used to detect and correct errors which appeared in texts frequently. In [13], Macdonald et al. proposed an automatic grammar checker which will report error if any rule is matched. There are still a few rules-based method like [14] while the mainstream approaches are based on machine learning algorithms nowadays. Many learning based approaches have two steps, solid feature engineering is the most important part before to construct traditional machine learning model. A variety of algorithms were adopted to detect errors, like maximum entropy based classifiers in [15] and LFG-based features used in [16]. Some Classic works were put forwarded to deal with specific error types, such as Tetreault et al. detected preposition errors by max maximum entropy classifier [17], [18] used this kind of model to detect articles usage error while [1] adopted Convolution Neural Networks (CNN) [19] to detect the same kind of error. [20] improved the basic approach of template matching with the aim of detecting verb form errors.

Similar with many attempts in GEC, there are some approaches had investigated the effectiveness of synthetic ungrammatical data for GED and the usage of large scale unlabeled right texts. Foster et al. proposed to generate errors by some rules and hoped to improve the performance of error detection by data augmentation [21]. Besides, [22] constructed features by a language model which was trained on a large and general domain corpus.

In recent years, neural networks based error detection methods are widely researched like works in GEC task [23]. The first meaningful work was carried out in [24], Rei et al. used bidirectional LSTM [25] to construct a neural sequence labelling model for incorrect token annotation based on the entire text representation. They make several improvements to enhance the ability of the sequence labelling model by incorporating character embedding to learn better word embedding [26] and adding another language model task to encourage the model to learn more accurate and generic representation of each word and the whole sentence [4].

The current best methods for GED are training neural sequence labelling model with data augmentation, rule and learning based approaches are utilized to generate ungrammatical texts artificially. [9] used neural machine translation model to back translate error free text to its corresponding error one and further improved the performance with the model in [24]. Bell et al. fully investigated the usage of pre-trained model like BERT and ELMo, and figured out that accurate contextualized embeddings can lead to much better result on several standard test set [7].

III. Data

In this section, we will introduce the datasets used in this paper. As commonly used in related work, we use both the publicly available real training data and generate some ungrammatical texts by rules and back translation.

A. Training Data

We collected three different datasets to fine-tuned the base BERT model in a pairwise style. The first one is the FCE which was released in 2011. The FCE corpora consists of nearly 30K sentences from 1141 essays, which were written by non-native English learners when they are taking language assessment exam. Some professional annotators had labeled whether each word is correct or incorrect for every sentence, in addition, suggested corrections are given by those experts with the corresponding error types.

The second dataset is the NUS Corpus of Learner English (NUCLE), which collected 1414 essays written by non-native students, too. Two native and professional instructors made corrections for each sentences carefully and resulted in a publicly available dataset with more than 57K pairwise samples. NUCLE is the official training data in the CoNLL 2013 and 2014 shared task for GEC.

The last corpus we adopted is commonly used in GEC, which was larger and less professional since it was collected from a website. The Lang-8.com is a social platform for various language learners to write essays in any language and some volunteers will put forward some corrections to them. Lang8 has over 1M pairs of texts released publicly.

What is more, artificially generated error texts have been widely used in GEC, we also constructed a large amount of texts with errors from right ones in the three publicly available training data described above. The first way we exploited is to design some rules, like replace, delete or add few words at random with a low probability of 15% for all sentences and 5% for each word, and change the words order of a short span of text or phrase with the probability of 10%. We also designed special rule for some particular error types. For example, we replace one of the article or preposition in a sentence by another article or preposition randomly or delete it directly, if the sentence have article or preposition.

Another way we utilized is a sequence to sequence model, back translation is a powerful tool for error texts generation but with large range of quality.

To avoid the noise brought by poor quality texts, we filter the data generated by rules and back translation with three metrics, language model based fluency evaluation like in [27], sentence embedding based semantic similarity measurement and edit distance based syntax similarity. We found that texts with fluency between 80% to 95% of the corresponding original correct ones are more like human-made and in lower quality. With the constraints of semantic similarity larger than 0.9 and

TABLE I
CORPORA STATISTICAL INFORAMTION

Corpora	Class	Max-Len	Min-Len	Avg-Len	Words-Num	Chars-Num
NUCLE	source	222	3	20.89	33805	115
	target	222	3	20.68	33258	114
Lang-8	source	448	3	12.35	126667	94
	target	494	3	12.6	109537	94
FCE	source	131	1	15.76	14532	93
	target	142	1	16.67	15076	94
Synthetic	source	423	1	18.21	153362	123
	target	422	1	18.94	156076	123
CoNLL-2014 test set		227	1	22.96	3143	75
FCE test set		92	1	15.25	3871	81
JFLEG test set		77	3	18.87	2787	77

TABLE II
SOME EXAMPLES OF WHICH THE ERRORS ARE GENERATED BY OURSELVES

methods	original text	error text
Rules	All Lang - 8 users are welcome .	All Lang - 8 users is welcome .
	This is a Japanese pop song .	This is an Japanese pop song .
	On the other hand , only a few people on Car2 got off at Suidobashi .	On the other hand , only a few people on Car2 get off at Suidobashi .
	One is English , and the others are professional courses .	One is English , and others the are professional courses .
	It makes me feel optimistic .	It make me feel optimistic .
	Can I go with a friend ?	Can I go with a riend ?
Back-Translation	All Lang - 8 users are welcome .	Every Lang - 8 users is welcomed .
	My father is the same age .	My father is in the same age .
	This may worsen the situation .	This may worse situations .
	In addition , there are huge differences between Asian culture and Western culture .	In addition , there have many differences between Asian culture and Western .
	I 'm good at listening and passing the vocabulary test .	I am good at listen the vocabulary test .
	So I want anyone who knows English , please correct it .	I want anyone who konw English correct my English .

levenshtein lower than 4, we filter out all the synthetic data and resulted in more than 3M pairs of samples.

Some texts with generated errors are shown in table II, errors like typo, preposition, article and word orders are more likely to be written by second language learners.

The statistics of these 3 training datasets and the generated dataset are shown in table I.

B. Test Data

To fully measure the effectiveness of the method proposed in this paper, we utilize four different test set. The FCE test set has 2270 sentences from 97 scripts which is similar with the FCE training corpus. The CoNLL-2014 test set has 1312 samples which are widely used in GEC and has two different annotators, result in two different test set. The last one is JHU FLuency-Extended GUG corpus (JFLEG), which contains 747 samples written by English learner from different background and was annotated by 4 experts considering not only grammar edits but also fluency. The information of these three test set are shown in table I, too.

IV. METHOD

In this section, we will give a brief introduction of the model used in this paper and the details of pairwise training mechanise utilized in our methods.

A. BERT

Recurrent neural networks represented by LSTM and GRU [28] have always dominated sequence modeling tasks such as machine translation and question answering. The proposal of the Transformer model based on self-attention mechanism in 2017 [2] broke this monopoly. At present, the best results for many language processing tasks are achieved by BERT which are based on encoder of Transformer. Transformer no longer relies on cyclic or convolutional network structures, and completely relies on attention mechanisms to model sequence data, and can well capture the dependencies between long-distance texts. In addition, this structure has natural parallelism and can avoid serial shortcomings caused by sequence dependence of recurrent neural network structure. Here are some of the components of Transformer.

The key component of Transformer is scaled dot-product attention calculated by equation 1, $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ represent the vector of query, key and value, in the GEC setting, they are all the input texts with embedding vectors. The equation 1 calculates the similarity of each word with all other words in the same sentence as the value of attention, and the similarity with itself is also included. All calculation can be paralleled with the matrix of word embeddings. Then the representation of each word can be obtained by the weighted sum of all the tokens in the corresponding sentence. Experimental results in [2] shown that calculate the attention multi-times by different affine transformations can further improve the performance of self-attention. Equation 2 indicates the process of this kind of multi-head attention.

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}})\boldsymbol{V} \tag{1}$$

$$\begin{aligned} &\text{MultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = Concat(head_1, ..., head_h)\boldsymbol{W}^O \\ &\text{head}_i = \text{Attention}(\boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}\boldsymbol{W}_i^V) \end{aligned} \tag{2}$$

Since Transformer does not modeling any order or position information, the explicit positional encoding is necessary for the model to capture the relationship of order between words. We use the positional encoding as shown in equation 3.

$$\begin{aligned} PE(pos, 2i) &= sin(pos/10000^{2i/d}) \\ PE(pos, 2i+1) &= cos(pos/10000^{2i/d}) \end{aligned} \tag{3}$$

In addition, in order to capture morphological information and avoid unknown words affection, we add a LSTM layer to model character embeddings of each word, and concatenate the last state of the LSTM to the word embeddings which initialized by BERT to form the word representation. The word representation and positional encoding are the input to the BERT, which will be fine tuned and trained on the whole training set we described above. After 6 layers' transformation, we classify whether each word is correct or not based on the output states.

B. Pairwise Training

The key component of our error detection method is the special training mechanism, and this particularity is manifested in two aspects. The first one is how we construct the batches of training samples and another is the calculation of training loss.

1) Training Batch: Considering that with each correct text in our datasets, there are several different possible wrong texts. These ungrammatical texts may come from the real data set, or they are artificially. So our entire training data is composed of a series of clusters corresponding to the same real and correct text. When constructing training data for each batch of each iteration step, we first randomly select a sample from each cluster, each batch selects half the sample of the batch size, and then randomly selects another sample from the clusters corresponding to the first one to form a pair sample. A batch of training data is finally obtained, The iteration will repeat this selecting process until there are no samples that can form a pair.

Through this construction method, the samples of each batch are composed of the same true and correct sample itself or its wrong version, which allows the model to learn the correct form and different incorrect styles.

2) Training Loss: In the training process, A normal binary classification training loss for each word of each sample is calculated, we call it the list-wise loss function. What is more, in order to force the model to learn the differences and commonalities between pairs of samples, we also construct a pair-wise training loss function, that is, for each pair of training samples in the same batch, we need to perform a four class classification task, corresponding to each word in the pair. The four cases are the two words are both correct or both wrong, or the first is right, the second is wrong, or vice verse. The total loss is calculated as equation 4, CrossEntropy(BinaryClassification) indicates the list-wise loss function and CrossEntropy(4ClassClassification) is the pair-wise training loss, a and b represent the weights for different type of loss.

$$\begin{aligned} \text{Loss} = a * &CrossEntropy(BinaryClassification) + \\ b * &CrossEntropy(4ClassClassification) \end{aligned} \tag{4}$$

The training framework and the two pairs of training samples are shown in figure 1.

V. Experiments and Analysis

In this section, we will describe the details of our experiments and represent the results of our method on three different test sets. Besides, some discussions will be made based on the performance of different model settings.

A. Experimental Details

1) Model settings: In this paper, we use 768-dimension words embeddings of the base and cased version BERT as the initialization of words embeddings used in our model, and the size of vocabulary with characters and words are 123 and 150K, we also concatenate word embedding by 1 layer LSTM based learnable character embedding with 256 dimensions to capture the information of stems and affixes. We only use 6 layers of the encoder from Transformer on account of the memory limitation of GPU, the heads number and hidden size is 8 and 4096 respectively. The max size of position and sentence length is 256, samples of which the length larger than 256 are truncated but little than this value are padded by PAD symbol, the label of PAD is set as correct. We classify each token and the pair of tokens by the last layer output.

2) Training details: Since there are two different kinds of loss in our error detection method, but we only care about the performance of single sentence detection when applying the model to real application. With the aim of meeting the requirements in inference, we adopt three different strategy in the early, mid and late states in training process. In the early stage (20 epoches), we set a=0.2 and b=0.8, to encourage the

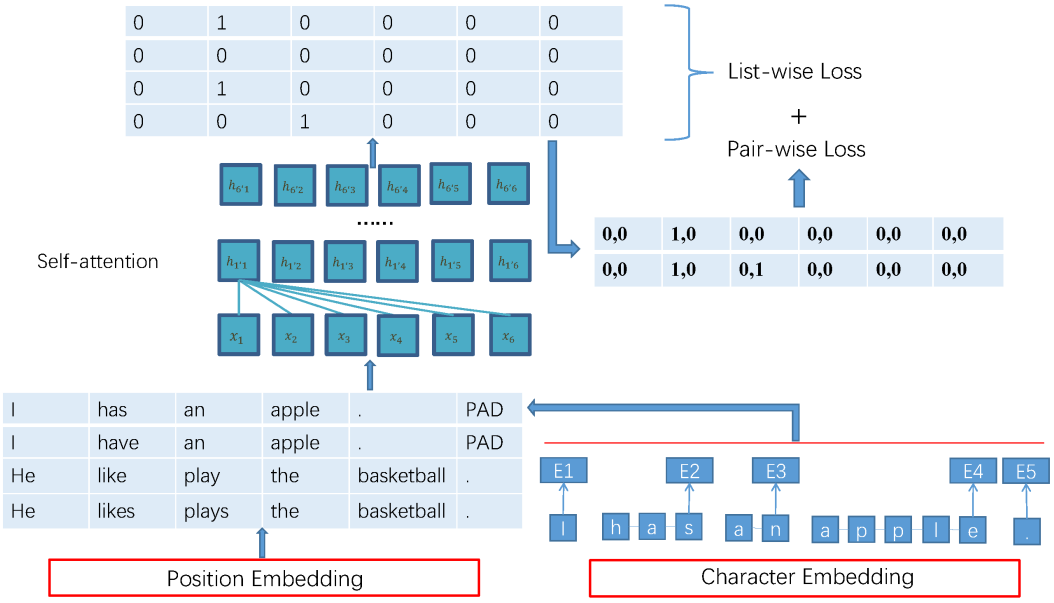

Fig. 1. The architecture of our model and pairwise training mechanism

model to pay more emphasis on the pair-wise loss, leading the model to capture the information from the difference of the pair which is useful to detect errors and learn more accurate representation of each word. The value of a and b are changed smoothly to 0.8 and 0.2 in 20 to 40 epoch with linear increase and decrease. In the last 20 epoches, we set a=0.8 and b=0.2, forcing the model to learn how to classify whether the word is incorrect or not based on the single sentence itself which is more similar to the scenario in inference.

3) Evaluation: As common used in other classification task, precision, recall and F-value is used to evaluate the performance of different approaches for GED. It is worthy to notice, since ignoring a error has little influence than misclassify a correct token to incorrect, $F_{0.5}$ is more meaningful to measure whether the performance is better or not since it pay 2 times of weight on precision than recall. $F_{0.5}$ is calculated as equation 5.

$$F_{0.5} = \frac{5PR}{P + 4R} \tag{5}$$

B. Results

With the model of 6 layers BERT and three stages of training with two kinds of loss in 60 epoches. Real data and synthetic data are used simultaneously, our model converge to a lower value of loss. We test the performances of our method on three four different test set, the results of baseline algorithms and our method are shown in table III. The Bert-base means that we use pre-trained word embedding from base version of Bert and fine tuned on all real data and artificially generated data, we conduct this experiment because of the improvements obtained by incorporating Bert in other works and adopt base version due to the computation limitation. DF represents Data Filtering which indicates whether we filter out the synthetic samples with poor quality. The PT denotes Pairwise Training that demonstrates whether the pair-wise loss is adopted or not. It is obviously that our pair-wise training mechanism with data augmentation and filter can help the self attention model achieves the best results on all four different test datasets. What is more, from the example in figure 2, we can figured out that the self attention mechanism can actually pay attention to other words with different weights to represents each word.

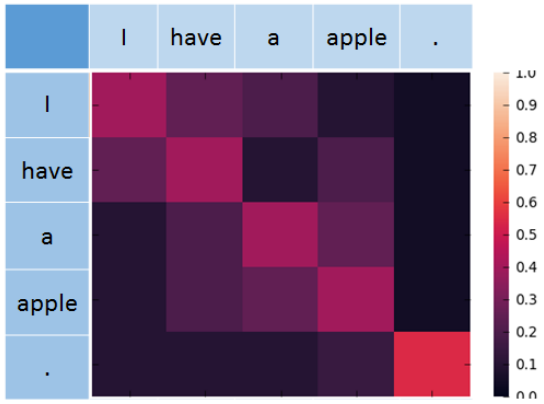

Fig. 2. The example of attention visualization

C. Discussion

It can be concluded that our method with words embeddings from pure base version BERT and fine tuned with all real training data and rule based generated data obtains performance on pair with other baselines, except the lower $F_{0.5}$ value on CoNLL-2014 test set annotated by second expert. Result on JFLEG test set is even better than the large

TABLE III
THE DETECTION RESULTS ON VARIOUS STANDARD TEST SETS BY DIFFERENT MODELS

	method	CoNLL-2014 test1			CoNLL-2014 test2			FCE test			JFLEG test		
		P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$
Baseline	Rei	17.68	19.07	17.86	27.6	21.18	25.88	58.88	28.92	48.48	72.84	22.83	50.65
	Rei et al	23.28	18.01	21.87	35.28	19.42	30.13	60.67	28.08	49.11	-	-	-
	Kasewa et al	-	-	28.3	-	-	35.5	-	-	55.6	-	-	-
	Bell et al BERT-base	37.62	29.65	35.7	53.52	30.05	46.29	64.96	38.89	57.28	79.51	32.94	61.98
	Bell et al BERT-large	38.04	33.12	36.94	51.4	31.89	45.8	64.51	38.79	56.96	76.47	34.52	61.52
ours	BERT-base	36.62	**36.87**	36.67	49.28	**34.54**	45.40	63.01	**43.02**	57.65	75.25	**37.12**	62.43
	+DF	37.81	35.03	37.22	51.38	32.25	45.93	64.81	40.77	57.97	76.97	36.64	63.08
	+PT	38.27	34.12	37.36	50.95	33.47	46.13	**66.72**	40.13	58.91	77.46	35.95	62.93
	+DF+PT	**40.19**	36.72	**39.44**	**52.26**	34.09	**47.23**	66.53	42.37	**59.72**	**78.84**	35.58	**63.42**

version BERT based baseline. Incorrect data generated by back translation can improve the performance to some extent, but the promotion is mainly brought by recall that not really useful in error detection.

Further more, the results shown that pairwise training mechanism can achieves better results on all test set except JFLEG compared with data augmented by back translation, as a conclusion, pairwise training is meaningful for error detection method to capture the differences between pairs of samples, which are valuable to distinguish errors. After incorporation pairwise training with data augmentation, our approach get the state of the art results on four test set. Nearly 2.5% improvement is obtained on CoNLL 2014 shared task for both annotators, the value of $F_{0.5}$ for JFLEG test set is increased little less than 2%. We achieve the highest improvement on FCE test set with nearly 3%.

The metrics of precision and recall are both improved by pairwise training and generated data using back translation, except the precision of FCE and recall of JFLEG. It can be concluded that our method is effective for various datasets from different domains, the overall performance can be improved significantly.

VI. CONCLUSION

In this paper, we propose pairwise training mechanism with words embeddings pre-trained with base version BERT for grammatical error detection, incorporating rules and back translation based data augmentation, our method achieves the state of the art results on four different standard test sets. To the best of our knowledge, this is the first attempts to adopt the pairwise training to copy with error detection task, despite that pairwise learning is commonly used in recommendation system. Extensive experiments with different model settings and training mechanism have shown the improvements on all test sets are mainly on account of the pairwise loss, which demonstrate that pairwise training is a valuable technique for classification tasks of natural language processing.

ACKNOWLEDGMENT

This work is supported by National Key R&D Program of China (Grant Nos.: 2018AAA0100300, 2018AAA0102301), and the National Natural Science Foundation of China (Grant Nos.: 61673025, 61375119), and partially supported by National Key Basic Research Development Plan (973 Plan) Project of China (Grant No. 2015CB352302).

REFERENCES

[1] C. Sun, X. Jin, L. Lin, Y. Zhao, and X. Wang, "Convolutional neural networks for correcting english article errors," in *Natural Language Processing and Chinese Computing - 4th CCF Conference, NLPCC 2015, Nanchang, China, October 9-13, 2015, Proceedings*, ser. Lecture Notes in Computer Science, J. Li, H. Ji, D. Zhao, and Y. Feng, Eds., vol. 9362. Springer, 2015, pp. 102–110. [Online]. Available: https://doi.org/10.1007/978-3-319-25207-0_9

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 5998–6008. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need

[3] W. Xu, H. Sun, C. Deng, and Y. Tan, "Variational autoencoder for semi-supervised text classification," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, S. P. Singh and S. Markovitch, Eds. AAAI Press, 2017, pp. 3358–3364. [Online]. Available: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14299

[4] M. Rei, "Semi-supervised multitask learning for sequence labeling," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, R. Barzilay and M. Kan, Eds. Association for Computational Linguistics, 2017, pp. 2121–2130. [Online]. Available: https://doi.org/10.18653/v1/P17-1194

[5] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 4171–4186. [Online]. Available: https://doi.org/10.18653/v1/n19-1423

[6] M. Kaneko and M. Komachi, "Multi-head multi-layer attention to deep language representations for grammatical error detection," *Computación y Sistemas*, vol. 23, no. 3, 2019. [Online]. Available: https://www.cys.cic.ipn.mx/ojs/index.php/CyS/article/view/3271

[7] S. Bell, H. Yannakoudakis, and M. Rei, "Context is key: Grammatical error detection with contextual word representations," in *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@ACL 2019, Florence, Italy, August 2, 2019*, H. Yannakoudakis, E. Kochmar, C. Leacock, N. Madnani, I. Pilán, and T. Zesch, Eds. Association for Computational Linguistics, 2019, pp. 103–115. [Online]. Available: https://doi.org/10.18653/v1/w19-4410

[8] M. Rei, M. Felice, Z. Yuan, and T. Briscoe, "Artificial error generation with machine translation and syntactic patterns," in *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational*

Applications, BEA@EMNLP 2017, Copenhagen, Denmark, September 8, 2017, J. R. Tetreault, J. Burstein, C. Leacock, and H. Yannakoudakis, Eds. Association for Computational Linguistics, 2017, pp. 287–292. [Online]. Available: https://doi.org/10.18653/v1/w17-5032

[9] S. Kasewa, P. Stenetorp, and S. Riedel, "Wronging a right: Generating better errors to improve grammatical error detection," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds. Association for Computational Linguistics, 2018, pp. 4977–4983. [Online]. Available: https://www.aclweb.org/anthology/D18-1541/

[10] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, M. A. Walker, H. Ji, and A. Stent, Eds. Association for Computational Linguistics, 2018, pp. 2227–2237. [Online]. Available: https://doi.org/10.18653/v1/n18-1202

[11] A. Akbik, D. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling," in *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, E. M. Bender, L. Derczynski, and P. Isabelle, Eds. Association for Computational Linguistics, 2018, pp. 1638–1649. [Online]. Available: https://www.aclweb.org/anthology/C18-1139/

[12] H. Yannakoudakis, T. Briscoe, and B. Medlock, "A new dataset and method for automatically grading ESOL texts," in *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, D. Lin, Y. Matsumoto, and R. Mihalcea, Eds. The Association for Computer Linguistics, 2011, pp. 180–189. [Online]. Available: https://www.aclweb.org/anthology/P11-1019/

[13] N. Macdonald, L. Frase, P. Gingrich, and S. Keenan, "The writer's workbench: Computer aids for text analysis," *IEEE Transactions on Communications*, vol. 30, no. 1, pp. 105–110, 1982.

[14] J. Foster and C. Vogel, "Parsing ill-formed text using an error grammar," *Artif. Intell. Rev.*, vol. 21, no. 3-4, pp. 269–291, 2004. [Online]. Available: https://doi.org/10.1023/B:AIRE.0000036259.68818.1e

[15] T. J. R. Chodorow M and H. N. R., "Detection of grammatical errors involving prepositions," *Proceedings of the fourth ACL-SIGSEM workshop on prepositions. Association for Computational Linguistics*, pp. 25–30, 2007.

[16] G. Berend, V. Vincze, S. Zarrieß, and R. Farkas, "Lfg-based features for noun number and article grammatical errors," in *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013*, H. T. Ng, J. R. Tetreault, S. M. Wu, Y. Wu, and C. Hadiwinoto, Eds. ACL, 2013, pp. 62–67. [Online]. Available: https://www.aclweb.org/anthology/W13-3608/

[17] J. R. Tetreault and M. Chodorow, "The ups and downs of preposition error detection in ESL writing," in *COLING 2008, 22nd International Conference on Computational Linguistics, Proceedings of the Conference, 18-22 August 2008, Manchester, UK*, D. Scott and H. Uszkoreit, Eds., 2008, pp. 865–872. [Online]. Available: https://www.aclweb.org/anthology/C08-1109/

[18] N. Han, M. Chodorow, and C. Leacock, "Detecting errors in english article usage by non-native speakers," *Nat. Lang. Eng.*, vol. 12, no. 2, pp. 115–129, 2006. [Online]. Available: https://doi.org/10.1017/S1351324906004190

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., 2012, pp. 1106–1114. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks

[20] J. Lee and S. Seneff, "Correcting misuse of verb forms," in *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, K. R. McKeown, J. D. Moore, S. Teufel, J. Allan, and S. Furui, Eds. The Association for Computer Linguistics, 2008, pp. 174–182. [Online]. Available: https://www.aclweb.org/anthology/P08-1021/

[21] J. Foster and Ø. E. Andersen, "Generrate: Generating errors for use in grammatical error detection," in *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@NAACL-HLT 2009, Boulder, CO, USA, June 5, 2009*, J. R. Tetreault, J. Burstein, and C. Leacock, Eds. Association for Computational Linguistics, 2009, pp. 82–90. [Online]. Available: https://www.aclweb.org/anthology/W09-2112/

[22] M. Gamon, "Using mostly native data to correct errors in learners' writing," in *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*. The Association for Computational Linguistics, 2010, pp. 163–171. [Online]. Available: https://www.aclweb.org/anthology/N10-1019/

[23] Q. Wang and Y. Tan, "Automatic grammatical error correction based on edit operations information," in *Neural Information Processing - 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13-16, 2018, Proceedings, Part V*, ser. Lecture Notes in Computer Science, L. Cheng, A. C. Leung, and S. Ozawa, Eds., vol. 11305. Springer, 2018, pp. 494–505. [Online]. Available: https://doi.org/10.1007/978-3-030-04221-9_44

[24] M. Rei and H. Yannakoudakis, "Compositional sequence labeling models for error detection in learner writing," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. [Online]. Available: https://doi.org/10.18653/v1/p16-1112

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[26] M. Rei, G. K. O. Crichton, and S. Pyysalo, "Attending to characters in neural sequence labeling models," in *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, N. Calzolari, Y. Matsumoto, and R. Prasad, Eds. ACL, 2016, pp. 309–318. [Online]. Available: https://www.aclweb.org/anthology/C16-1030/

[27] T. Ge, F. Wei, and M. Zhou, "Fluency boost learning and inference for neural grammatical error correction," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, I. Gurevych and Y. Miyao, Eds. Association for Computational Linguistics, 2018, pp. 1055–1065. [Online]. Available: https://www.aclweb.org/anthology/P18-1097/

[28] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555