Automatic Grammatical Error Correction Based on Edit Operations Information

Quanbin Wang and Ying Tan(✉)

Key Laboratory of Machine Perception (Ministry of Education)
and Department of Machine Intelligence,
School of Electronics Engineering and Computer Science, Peking University,
Beijing 100871, People's Republic of China
{qbwang362,ytan}@pku.edu.cn

Abstract. For second language learners, a reliable and effective Grammatical Error Correction (GEC) system is imperative, since it can be used as an auxiliary assistant for errors correction and helps learners improve their writing ability. Researchers have paid more emphasis on this task with deep learning methods. Better results were achieved on the standard benchmark datasets compared to traditional rule based approaches. We treat GEC as a special translation problem which translates wrong sentences into correct ones like other former works. In this paper, we propose a novel correction system based on sequence to sequence (Seq2Seq) architecture with residual connection and semantically conditioned LSTM (SC-LSTM), incorporating edit operations as special semantic information. Our model further improves the performance of neural machine translation model for GEC and achieves state-of-the-art $F_{0.5}$-score on standard test data named CoNLL-2014 compared with other methods that without any re-rank approach.

Keywords: Grammatical error correction · Edit operations
Natural language processing · Semantically conditioned LSTM
Sequence to sequence

1 Introduction

With the development of globalization, the number of second language learners is growing rapidly. Errors, including grammar, misspelling and collocation (for simplicity, we call all these types of errors grammatical errors) are inevitable for freshman who just begin to learn a new language. In order to help those learners to avoid making errors in their learning process and improve their skills both for writing and speaking, an automatic grammatical error correction (GEC) system is necessary.

Specifically, GEC for English has attracted much attention as an important natural language processing (NLP) task since 1980s. Macdonald et al. developed a GEC tool named Writer's Workbench based on some rules in 1982 [22], this

L. Cheng et al. (Eds.): ICONIP 2018, LNCS 11305, pp. 494–505, 2018.
https://doi.org/10.1007/978-3-030-04221-9_44

work leads the research in this field. Rule based error correction methods can achieve high precision but with lower recall because the lack of generalization. Some learning based approaches had been adopted to alleviate this drawback, such as learning correction rules with corpora and machine learning algorithms with N-grams features. Mangu et al. proposed a method which learned rules for misspelling correction from a data set called Brown [13]. In addition, [29] used N-grams and language model (LM) to cope with GEC problem.

From a common accepted perspective, researchers always treat GEC as a special translation task which translates text with errors to correct one. On account of this, many machine translation methods are utilized to rectify errors. Statistical machine translation (SMT) as one of the most effective approach, was first adopted to GEC in 2006 [3], they used SMT based model to correct 14 kinds of noun number (Nn) errors and achieved much better performance than rules based systems. Compared to traditional rules based and learning based methods, machine translation based approaches only need corpora with pairwise sentences. What is more, they have no limit to specific error types and can construct general correction model for all kinds of errors. Whereas the main drawback of SMT based GEC is that it handles each word or phrase independently, which results in ignoring global context information and relationship between each entity. With the aim of making up this deficiency, researchers attempted to take the advantages of some neural encoder-decoder architectures such as sequence to sequence (Seq2Seq) [27] with recurrent neural networks (RNN), since these models considered the global source text and all the preceding words when decoding. Xie et al. proposed a neural machine translation (NMT) model based GEC system in 2014 [31], which was the first attempt to combine encoder-decoder architecture with attention mechanism like the work in NMT [1]. They used character level embedding and gated recurrent unit (GRU) [5] to correct all kinds of errors and obtained a result on pair with state-of-the-art at that time.

In this paper, we further exploit neural encoder-decoder architecture with RNN and attention mechanism which is similar as commonly used in NMT. In addition, we utilize residual connection as in ResNet but with RNN [16] between every two layers to make training process stable and effective. Different from [31], we adopt long short term memory (LSTM) [17] in both encoder and decoder steps with special semantic information called edit operations. We conclude 3 kinds of different edit operations in correction process as "Delete, Insert and Substitute", which can also be considered as 3 kinds of simple error types as "Unnecessary, Missing and Replacement" as defined in [4,11]. For the purpose of using these edit operations information, semantically conditioned LSTM (SC-LSTM) [30] is applied to our RNN based Seq2Seq model. In view of only a small part of the whole text need to be corrected, we use a gate for those edit operations. Through the results of our experiments, the gate is very useful to improve the performance of SC-LSTN in GEC task. Since whether opening the gate or not in a decoding step is mainly depends on all the words had been generated until now, and there exists a clear distinction between training process and inference, the model may give error gate information because of some mistakes made in former steps. With the aim of alleviating this drawback, we take the advantage of the scheduled sampling technique [2]. With all of these methods, our automatic GEC system with edit operations information achieves 48.67% $F_{0.5}$-score on the benchmark CoNLL-2014 test set [23]. It is state-of-the-art performance compared to other approaches without the help of large language model and other tricks to re-rank candidate corrections.

2 Related Work

Researchers in the field of NLP had paid much emphasis on GEC task since 2013, with the organization of CoNLL-2013 and 2014 shared tasks [23,24], of which were competitions to cope with grammatical error correction problem of essays written by second language learners. The test set in 2014 shared task had been used as a standard benchmark since then and many works were made to perform well on it.

The most commonly used methods in recent years are all related to machine translation including statistical and neural models. All the top-ranking teams in CoNLL shared tasks are used SMT based approaches to correct grammatical errors, such as CAMB [12] and AMU [19]. Susanto et al. proposed a system which combined SMT based method with a classification model and got a better result [26]. The most effective technique which purely based on SMT was put forward by Chollampatt et al. [6], they designed some sparse and dense features manually and incorporated some tricks, such as LM, spelling checker and neural network joint model (NNJMs) [8], to further improve their model's performance which was similar to [20].

In spite of the success of SMT based model for GEC task, those kinds of methods suffer from ignoring global context information and lacking of smooth representation which resulted in lower generalization and unnatural correction. To address these issues, several correction systems which adopted neural encoder-decoder framework have been presented. *RNNSearch* [1]was the first NMT model be utilized to correct grammatical errors by Yuan et al. [32]. They additionally applied an unsupervised word alignment technique and a word level SMT for unknown words replacing. However, their work were conducted with Cambridge Learner Corpus (CLC) which is non-public. Xie et al. [31] used a model with similar architecture, but they chose character level granularity to avoid unknown words problem effectively. They trained their model on two publicly available corpora called NUCLE [10] and Lang-8 [28]. For supplementary, they synthesized examples with frequent errors by some rules. A N-gram LM and edit classifier were incorporated to choose solutions. Ji et al. also proposed a RNN based Seq2Seq model with hybrid word and character level embedding and attention for known and unknown words respectively [18]. Except NUCLE and Lang-8, they employed non-public CLC dataset like [32] for training. What is more, they further improved the performance of their correction system by a candidates rescoring LM based on a very large scale corpora. Researchers have investigated the effectiveness of convolutional neural networks (CNN) for encoder-decoder architecture to cope with GEC task. Chollampatt et al. proposed a Seq2Seq model fully based on multi-layer CNN [7], they adopted the famous model in [14] with BPE-based sub-word units embedding. In order to select the best correction, they trained a resoring model with edit operations and LM as features explicitly.

The most valid correction system until now was put forward by Grundkiewicz et al. [15], they combined NMT and SMT model together and used corrections from the best SMT as the inputs of NMT model, incorporating with SMT based spelling checker and RNN based LM, they achieved state-of-the-art performance on CoNLL-2014 test set. Moreover, a most related work was proposed by Schmaltz et al. in 2017 [25]. Different from [7] and our work, they used edit operations as special tags in the target sentences and predicted those tags as atomic tokens in decoding.

3 GEC Based on Edit Operations

In the following sections, we will describe our work in details, including the corpora we used, our model architecture, experimental settings and results. At last, a results' analysis was presented.

3.1 Datasets

As general, we collected two publicly available corpora as talked above, NUCLE [10] and Lang-8 [28]. The details of this two data sets are shown in Table 1.

Table 1. Corpora statistical information

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
CoNLL-2014 test set		227	1	22.96	3143	75

Since NUCLE corpora is homologous with CoNLL-2014 test set but in a small amount compared with Lang-8, we adopt a simple up-sampling technique that using these samples twice for training. In data preprocessing step, we discard samples with more than 200 characters despite in source or target, in addition, we only use parallel samples that the difference of length between source text and target one are less than 50. Moreover, some samples' correct target texts are with all words been removed, we throw away all these kinds of data directly. After those processing steps, we split the whole corpora into training and validation sets randomly and results in over 0.9M training samples and nearly 10 K for validation. For model's performance comparison, we choose CoNLL-2104 test set [23] which has 1312 samples as commonly used in this task.

3.2 Model Architecture

The main architecture of our GEC system is the commonly used Seq2Seq [27] framework but with a soft attention mechanism in decoder which is similar as [1]. The simplified version of our model architecture with 3 layers is shown in Fig. 1. Our model is constituted by 4 layers encoder and 4 layers decoder with residual connection between each 2 layers and attention mechanism is adopted in the last decoder layer. The bottom-left corner represents the encoder of our model which encodes source text in character level including space symbol. The bottom layer is a bi-directional RNN with half layer size and traditional LSTM cell compared to upper layers, and process embedding data forward and backward respectively to make sure the encoder can obtain contextual information of the source text.

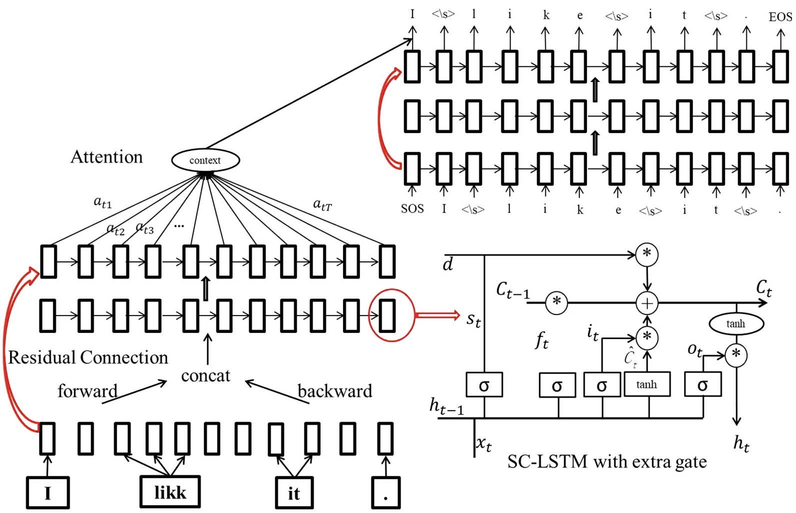

Fig. 1. The architecture of our GEC system with residual connection, attention mechanism and SC-LSTM with extra gate.

Upper layers are all in forward style and with SC-LSTM [30] which is very similar with traditional LSTM but with a semantical vector $\mathbf{d}$ that represents the semantical information of the text, in our model, it represents the edit operations needed for this error text. Since not all tokens need to be changed, we add a semantical gate to control the information flow of this vector. The SC-LSTM which illustrated in the bottom-right corner of Fig. 1 is defined by the following equations with main difference in Eq. 6.

$$\mathbf{i}_t = \sigma(\mathbf{W}_{wi}\mathbf{w}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1}) \tag{1}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{wf}\mathbf{w}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1}) \tag{2}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{wo}\mathbf{w}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1}) \tag{3}$$
$$\mathbf{s}_t = \sigma(\mathbf{W}_{ws}\mathbf{w}_t + \mathbf{W}_{hs}\mathbf{h}_{t-1}) \tag{4}$$
$$\hat{\mathbf{c}}_t = \tanh(\mathbf{W}_{ws}\mathbf{w}_t + \mathbf{W}_{hs}\mathbf{h}_{t-1}) \tag{5}$$
$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \hat{\mathbf{c}}_t + \mathbf{s}_t \odot \tanh(\mathbf{W}_{dc}\mathbf{d}) \tag{6}$$
$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \tag{7}$$

To avoid gradient vanishing and make training process stable, we adopt residual connection both in encoder and decoder which is represented by red-curved arrow. It changes the inputs of middle layers, of which can be defined by following equations. $\mathbf{I}_t$ indicates the inputs of time t and i means ith layer, $\mathbf{x}$ represents the source or target text with word embedding and $\mathbf{h}$ is the hidden states of RNN cells.

$$\mathbf{I}_t = \begin{cases} \mathbf{x}_t & i = 0 \\ \mathbf{h}_t^{i-1} & i = 1 \\ \mathbf{h}_t^{i-1} + \mathbf{h}_t^{i-2} & i > 1 \end{cases} \tag{8}$$

Another important component of our model is attention mechanism as used in [1] which is shown in the top-left corner of Fig. 1. We use weighted sum of encoder outputs as context vector in the last decoder layer for generates characters. The weight a_{tk} is computed as defined in Eqs. 9–11 where t indicates the decoding step that from 1 to T_t, and e_k represents the kth encoder output. k and j both range from 1 to T_s. ϕ_1 and ϕ_2 are two feedforward affine transforms, T_s and T_t represent the length of source error text and target right one respectively. $\mathbf{h}_t^L$ is the tth hidden state of the last decoder layer and C_t means of context vector computed by weights and encoder outputs for decoding at step t.

$$u_{tk} = \phi_1(\mathbf{h}_t^L)^T \phi_2(\mathbf{e}_k) \tag{9}$$
$$a_{tk} = \frac{u_{tk}}{\sum_{j=1}^{T_s} u_{tj}} \tag{10}$$
$$\mathbf{C}_t = \sum_{j=1}^{T_s} a_{tj}\mathbf{e}_j \tag{11}$$

3.3 Experiments

For experiments, we use the model described above with character level operations. In view of the correction of misspelling, we represent each sample in character style with a vocabulary constituted by 99 unique characters. The embedding dimension of each character is 256 and maximum sentence length is limited to 200.

The most important part of our method is the edit operations information $\mathbf{d}$ used in SC-LSTM which are extracted by a ERRor ANnotation Toolkit (ERRANT) [4,11]. The toolkit is designed to automatically annotate parallel English sentences with rule based error type information, all errors are grouped into 3 kinds of edit operations named "Unnecessary, Missing and Replacement". They are determined by whether tokens are deleted, inserted or substituted respectively. We use this toolkit to extract all edit operations, and represent them with a 3 dimensional one-hot vector $\mathbf{d}$ to indicate whether the operations are needed or not for a specific error sentence.

Training. The model is trained using negative log-likelihood loss function as defined in Eq. 12, where N is the number of pairwise samples in a batch and T_t^i is the number of characters in the ith target right sentence, $\mathbf{x}$ and $\mathbf{d}$ indicate the source error text and edit operations vector respectively $y_{i,j}$ represents the jth token in the correction for the ith instance.

$$Loss = -\frac{1}{N}\sum_{i}^{N}\frac{1}{T_t^i}\sum_{j=1}^{T_t^i}\log(p(y_{i,j}|y_{i,1},...,y_{i,j-1},\mathbf{x},\mathbf{d})) \tag{12}$$

The parameters are optimized by Adaptive Moment Estimation (Adam) [21] with learning rate set as 0.0003.

Another useful technique we adopt in our experiments is scheduled sampling [2]. On account of the computation of gate for edit operations information relies heavily on the preceding tokens. The different usage of target sentence between training and inference affects the accuracy of computing semantical gates greatly. In order to alleviate the influence of this distinction, we utilize scheduled sampling with linear decay on some randomly chosen samples to bridge the gap between training and inference.

Inference. For inference and testing, the edit operations we use in training are unavailable since we do not know the corrections of samples in test set. We take a simple traversal approach which means we consider all possible combinations of edit operations. This method results in 8 kinds of different cases. We do correction for each of them using beam search technique with same beam size. 24 candidates are obtained and sorted by the cumulative probability of each token, the top one is regarded as the best correction.

3.4 Results and Analysis

Experimental Results. We compare the loss on validation set for 3 different conditions as shown in Fig. 2, the green-triangle one represents experiments without edit operations information with traditional LSTM and orange-star one shows the loss without scheduled sampling technique in training. The blue-dot one is the performance of our final model.

More concretely, the MaxMatch (M^2) [9] scores computed by standard evaluation metric on CoNLL-2014 test set for those three different experimental settings are shown in Table 2.

In Table 2, the top 5 lines are some baselines of previous works by other researchers. The bottom 3 lines show the results of our model in which EOI

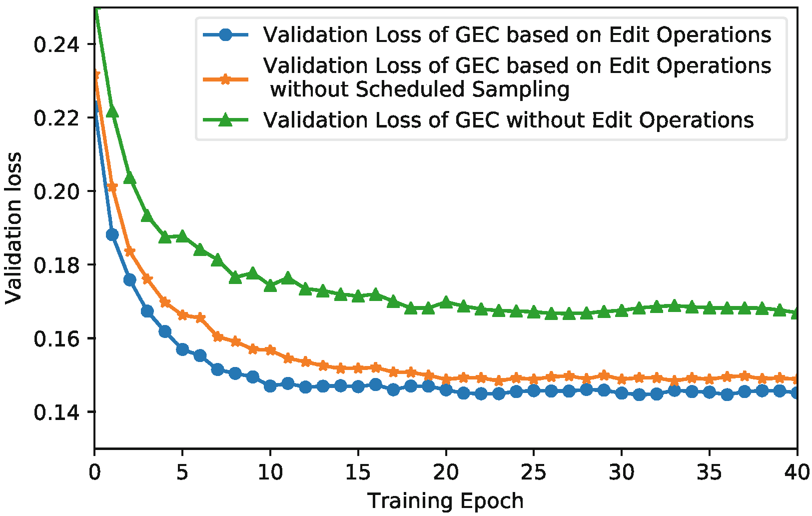

Fig. 2. The loss comparison of three different conditions on test set

Table 2. M^2 score comparison on CoNLL-2014 test set among our model and other previous work without the help of re-rank technique

Model	Parallel train data	P	R	$F_{0.5}$
Baseline				
SMT of [6]	Lang-8,NUCLE	58.24	24.84	45.90
SMT of [20]	Lang-8,NUCLE	57.99	25.11	45.95
NMT of [18]	Lang-8,NUCLE,CLC	-	-	41.53
NMT of [31]	Lang-8,NUCLE	45.86	26.40	39.97
MLConv [7]	Lang-8,NUCLE	59.68	23.25	45.36
MLConv(4 ens.) [7]	Lang-8,NUCLE	**67.06**	22.52	48.05
Ours				
GEC w/o EOI	Lang-8,NUCLE	60.43	20.61	43.58
GEC_EOI w/o SS	Lang-8,NUCLE	54.55	30.16	46.95
Best GEC w/ EOI	Lang-8,NUCLE	55.34	**32.83**	**48.67**

is the representation of Edit Operations Information and SS means Scheduled Sampling. In addition, some correction examples are show in Table 3.

Analysis. To be fair, all the baselines are without the help of re-rank or rescoring methods such as large scale LM since all of our experiments are conducted without any those kinds of techniques. From the results, we can conclude that our method obtain the best overall performance and edit operations are very effective for grammatical error correction. Of which some previous work also

Table 3. Some examples corrected by our EO_GEC

Source error sentence	Target right correction
It's **heavy rain** today	It **rained heavily** today
Everyone wants to be **success**	Everyone wants to be **successful**
I **likk** it	I **like** it
I **has a** apple	I **have an** apple
I start to learning English again	**I'm starting** to learn English again
I am very **interes on** the book	I am very **interested in** the book
The poor man needs a house to live	The poor man needs a house to live **in**
We must return **back** to school this afternoon	We must return to school this afternoon

had proved in other aspects, for example, [7] used edit operations information to train rescoring model and further improved their system's performance. In detail, compared with other approaches, our model achieves much higher recall but with lower precision, the main reason is that edit operations bring more information to correct errors. In addition, our straightforward traversal skill in inference is more likely to do more corrections which further results in higher recall but may lose precision.

4 Conclusion

In conclusion, we propose a neural sequence to sequence grammatical error correction system which utilizes edit operations information in encoder and decoder directly, the model with SC-LSTM achieves state-of-the-art performance on standard benchmark compared to other former effective approaches with fair conditions. To our knowledge, it is the first attempt to exploit edit operations as semantic information to control the correction process. The usage of character level representation, residual connection and scheduled sampling further improve our method's robustness and effectiveness. The traversal technique for edit operations in inference is intuitive but very valid. We can further enhance its capacity by some kinds of selection tricks to avoid unnecessary modification and result in promotion of precision. We will explore further in this direction in the future. What's more, direct utilization of error types information may be more effective but with many difficulties since there are more categories of errors, but it is a valuable research work.

Acknowledgments. This work was supported by the Natural Science Foundation of China (NSFC) under grant no. 61673025 and 61375119 and Supported by Beijing Natural Science Foundation (4162029), and partially supported by National Key Basic Research Development Plan (973 Plan) Project of China under grant no. 2015CB352302.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N.: Scheduled sampling for sequence prediction with recurrent neural networks. In: Advances in Neural Information Processing Systems 28, Annual Conference on Neural Information Processing Systems 2015, 7–12 December 2015, Montreal, Quebec, Canada, pp. 1171–1179 (2015)
3. Brockett, C., Dolan, W.B., Gamon, M.: Correcting ESL errors using phrasal SMT techniques. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia. Association for Computational Linguistics, pp. 249–256 (2006)
4. Bryant, C., Felice, M., Briscoe, T.: Automatic annotation and evaluation of error types for grammatical error correction. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, 30 July–4 August, Volume 1: Long Papers, pp. 793–805 (2017). https://doi.org/10.18653/v1/P17-1074
5. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
6. Chollampatt, S., Ng, H.T.: Connecting the dots: towards human-level grammatical error correction. In: Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, BEA@EMNLP 2017, Copenhagen, Denmark, 8 September 2017, pp. 327–333 (2017). https://aclanthology.info/papers/W17-5037/w17-5037
7. Chollampatt, S., Ng, H.T.: A multilayer convolutional encoder-decoder neural network for grammatical error correction. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, 2–7 February 2018 (2018). https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17308
8. Chollampatt, S., Taghipour, K., Ng, H.T.: Neural network translation models for grammatical error correction. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9–15 July 2016, pp. 2768–2774 (2016). http://www.ijcai.org/Abstract/16/393
9. Dahlmeier, D., Ng, H.T.: Better evaluation for grammatical error correction. In: Human Language Technologies, Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, 3–8 June 2012, Montréal, Canada, pp. 568–572 (2012). http://www.aclweb.org/anthology/N12-1067
10. Dahlmeier, D., Ng, H.T., Wu, S.M.: Building a large annotated corpus of learner English: the NUS corpus of learner English. In: Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@NAACL-HLT 2013, 13 June 2013, Atlanta, Georgia, USA, pp. 22–31 (2013). http://aclweb.org/anthology/W/W13/W13-1703.pdf
11. Felice, M., Bryant, C., Briscoe, T.: Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments. In: COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 11–16 December 2016, Osaka, Japan, pp. 825–835 (2016). http://aclweb.org/anthology/C/C16/C16-1079.pdf

12. Felice, M., Yuan, Z., Andersen, Ø.E., Yannakoudakis, H., Kochmar, E.: Grammatical error correction using hybrid systems and type filtering. In: Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task, CoNLL 2014, Baltimore, Maryland, USA, 26–27 June 2014, pp. 15–24 (2014). http://aclweb.org/anthology/W/W14/W14-1702.pdf
13. Francis, W.N., Kucera, H.: The brown corpus: a standard corpus of present-day edited American English. Department of Linguistics, Brown University [producer and distributor], Providence, RI (1979)
14. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017, pp. 1243–1252 (2017). http://proceedings.mlr.press/v70/gehring17a.html
15. Grundkiewicz, R., Junczys-Dowmunt, M.: Near human-level performance in grammatical error correction with hybrid machine translation. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, 1–6 June 2018, Volume 2 (Short Papers), pp. 284–290 (2018). https://aclanthology.info/papers/N18-2046/n18-2046
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR abs/1512.03385 (2015). http://arxiv.org/abs/1512.03385
17. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
18. Ji, J., Wang, Q., Toutanova, K., Gong, Y., Truong, S., Gao, J.: A nested attention neural hybrid model for grammatical error correction. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, 30 July 30–4 August, Volume 1: Long Papers, pp. 753–762 (2017). https://doi.org/10.18653/v1/P17-1070
19. Junczys-Dowmunt, M., Grundkiewicz, R.: The AMU system in the CoNLL-2014 shared task: grammatical error correction by data-intensive and feature-rich statistical machine translation. In: Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task, CoNLL 2014, Baltimore, Maryland, USA, 26–27 June 2014, pp. 25–33 (2014). http://aclweb.org/anthology/W/W14/W14-1703.pdf
20. Junczys-Dowmunt, M., Grundkiewicz, R.: Phrase-based machine translation is state-of-the-art for automatic grammatical error correction. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, 1–4 November 2016, pp. 1546–1556 (2016). http://aclweb.org/anthology/D/D16/D16-1161.pdf
21. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. CoRR abs/1412.6980 (2014). http://arxiv.org/abs/1412.6980
22. Macdonald, N., Frase, L., Gingrich, P., Keenan, S.: The writer's workbench: computer aids for text analysis. IEEE Trans. Commun. **30**(1), 105–110 (1982)
23. Ng, H.T., Wu, S.M., Briscoe, T., Hadiwinoto, C., Susanto, R.H., Bryant, C.: The CoNLL-2014 shared task on grammatical error correction. In: Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task, CoNLL 2014, Baltimore, Maryland, USA, 26–27 June 2014, pp. 1–14 (2014). http://aclweb.org/anthology/W/W14/W14-1701.pdf

24. Ng, H.T., Wu, S.M., Wu, Y., Hadiwinoto, C., Tetreault, J.R.: The CoNLL-2013 shared task on grammatical error correction. In: Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, CoNLL 2013, Sofia, Bulgaria, 8–9 August 2013, pp. 1–12 (2013). http://aclweb.org/anthology/W/W13/W13-3601.pdf
25. Schmaltz, A., Kim, Y., Rush, A.M., Shieber, S.M.: Adapting sequence models for sentence correction. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, 9–11 September 2017, pp. 2807–2813 (2017). https://aclanthology.info/papers/D17-1298/d17-1298
26. Susanto, R.H., Phandi, P., Ng, H.T.: System combination for grammatical error correction. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, 25–29 October 2014, Doha, Qatar. A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 951–962 (2014). http://aclweb.org/anthology/D/D14/D14-1102.pdf
27. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)
28. Tajiri, T., Komachi, M., Matsumoto, Y.: Tense and aspect error correction for ESL learners using global context. In: The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 8–14 July 2012, Jeju Island, Korea - Volume 2: Short Papers, pp. 198–202 (2012). http://www.aclweb.org/anthology/P12-2039
29. Zhang, K.L., Wang, H.F.: A unified framework for grammar error correction. In: CoNLL-2014, pp. 96–102 (2014)
30. Wen, T., Gasic, M., Mrksic, N., Su, P., Vandyke, D., Young, S.J.: Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, 17–21 September 2015, pp. 1711–1721 (2015). http://aclweb.org/anthology/D/D15/D15-1199.pdf
31. Xie, Z., Avati, A., Arivazhagan, N., Jurafsky, D., Ng, A.Y.: Neural language correction with character-based attention. arXiv preprint arXiv:1603.09727 (2016)
32. Yuan, Z., Briscoe, T.: Grammatical error correction using neural machine translation. In: NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, 12–17 June 2016, pp. 380–386 (2016). http://aclweb.org/anthology/N/N16/N16-1042.pdf