Controlling Grammatical Error Correction Using Word Edit Rate

Kengo Hotate, Masahiro Kaneko, Satoru Katsumata and **Mamoru Komachi**
Tokyo Metropolitan University
{hotate-kengo, kaneko-masahiro, satoru-katsumata}@ed.tmu.ac.jp
komachi@tmu.ac.jp

Abstract

When professional English teachers correct grammatically erroneous sentences written by English learners, they use various methods. The correction method depends on how much corrections a learner requires. In this paper, we propose a method for neural grammar error correction (GEC) that can control the degree of correction. We show that it is possible to actually control the degree of GEC by using new training data annotated with word edit rate. Thereby, diverse corrected sentences is obtained from a single erroneous sentence. Moreover, compared to a GEC model that does not use information on the degree of correction, the proposed method improves correction accuracy.

1 Introduction

The number and types of corrections in a sentence containing grammatical errors written by an English learner vary from annotator to annotator (Bryant and Ng, 2015). For example, it is known that the JFLEG dataset (Napoles et al., 2017) has a higher *degree of correction* in terms of the amount of corrections per sentence than that in the CoNLL-2014 dataset (Ng et al., 2014). This is because CoNLL-2014 contains only minimal edits, whereas JFLEG contains corrections with fluency edits (Napoles et al., 2017). Similarly, the degree of correction depends on the learners because it should be personalized to the level of learners. In this study, we used *word edit rate (WER)* as an index of the degree of correction. As WER is an index that shows the number of rewritten words in sentences, the WER between an erroneous sentence and a corrected sentence can represent the degree of correction of the sentence. Figure 1 shows that the WER of the JFLEG test set is higher than that of the CoNLL-2014 test set; thus, the WER shows the degree of correction.

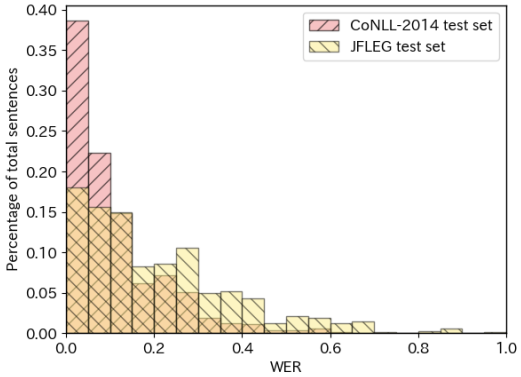

Figure 1: Histogram of the WER in one sentence.

However, existing GEC models consider only the single degree of correction suited for training corpus. Recently, neural network-based models have been actively studied for use in grammatical error correction (GEC) tasks (Chollampatt and Ng, 2018). These models outperform conventional models using phrase-based statistical machine translation (SMT) (Junczys-Dowmunt and Grundkiewicz, 2016). Nonetheless, controlling the amount of correction required to obtain an error-free sentence is not possible.

Therefore, we propose a method for neural GEC that can control the degree of correction. In the training data, in which grammatical errors are corrected, we add information about the degree of correction to erroneous sentences as WER tokens to create new training data. Then, we train the neural network model using the new training data annotated with the degree of correction. At the time of inference, this model can control the degree of correction by adding a WER token to the input. In addition, we propose a method to select and estimate the degree of correction required for each input sequence.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 149–154
Florence, Italy, July 28 - August 2, 2019.

Corpus	Sent.
Lang-8	1.3M
NUCLE	16K
Extra Data (NYT 2007)	0.4M

Table 1: Summary of training data.[1]

In the experiments, we controlled the degree of correction of the model for the CoNLL and JFLEG. As a result, we confirmed that the degree of correction of the model can actually be controlled, and consequently diverse corrected sentences can be generated. Moreover, we calculated the correction accuracies of both the CoNLL-2014 test set and JFLEG test set and demonstrated that the proposed method improved the scores of both $F_{0.5}$ using the softmax score and GLEU using the language model (LM) score more than the baseline model.

The main contributions of this work are summarized as follows:

- The degree of correction of the neural GEC model can be controlled using the WER.
- The proposed method increases correction accuracy and produces diverse corrected sentences to further improve GEC.

2 Controlling the degree of correction by using WER

We propose a method to control the degree of correction of the GEC model by adding tokens based on the WER, which is calculated for all sentences in the training data. The method of calculating WER and adding WER tokens is described as follows.

First, the Levenshtein distance is calculated from the erroneous sentence and the corresponding corrected sentence in the training data. Then, WER is calculated by normalizing this distance with respect to the source length.

Second, appropriate cutoffs are selected to divide the sentences into five equal-sized subsets. Different WER tokens are defined for each subset and added to the beginning of the source sentences.

Finally, the following parallel corpus is obtained: error-containing sentences annotated with the WER token representing the correction degree at the beginning of sentences and the corresponding sentences in which errors are corrected. The GEC model is trained using this newly created training data.

[1]Only sentences with corrections are used, and the sentence length limit is 80 words.

WER Token	Min	Max	Sent.
⟨1⟩	0.01	0.12	350K
⟨2⟩	0.12	0.20	350K
⟨3⟩	0.20	0.31	350K
⟨4⟩	0.31	0.53	350K
⟨5⟩	0.53	38.00 [2]	350K

Table 2: Thresholds of WER and number of sentences corresponding to WER tokens in the training data.

At the time of inference, five kinds of output sentences are obtained for each input sentence through the WER token. Therefore, we propose two simple ranking methods to automatically decide the optimal degree of correction for each input sentence.

Softmax. Ranking the 5 single best candidates $\mathbf{Y}$ using the sum of log probabilities of softmax score normalized by the hypothesis sentence length $|\boldsymbol{y}|$. The softmax score shows whether the hypothesis sentence $\boldsymbol{y}$ is appropriate for source sentence $\boldsymbol{x}$.

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y} \in \mathbf{Y}}{\arg\max} \frac{1}{|\boldsymbol{y}|} \sum_{i=1}^{|\boldsymbol{y}|} \log P(y_i | y_1, \cdots, y_{i-1}, \boldsymbol{x})$$

Language model (LM). Ranking the 5 single best candidates $\mathbf{Y}$ using the score of an n-gram LM. This score is normalized by the sentence length of the GEC model, and shows the fluency of hypothesis sentence $\boldsymbol{y}$.

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y} \in \mathbf{Y}}{\arg\max} \frac{1}{|\boldsymbol{y}|} \sum_{i=1}^{|\boldsymbol{y}|} \log P(y_i | y_{i-(n-1)}, \cdots, y_{i-1})$$

3 Experiments

3.1 Datasets

Table 1 summarizes the training data. We used Lang-8 (Mizumoto et al., 2012) and NUCLE (Dahlmeier et al., 2013) as the training data. The accuracy of the GEC task is known to be improved by increasing the amount of the training data (Xie et al., 2018). Therefore, we added more

[2]WER may exceed the one in which the Levenshtein distance is larger than the number of words in the target sentence.

Model	CoNLL-2013 (Dev)			CoNLL-2014 (Test)			JFLEG (Dev)	JFLEG (Test)	
	P	R	$F_{0.5}$	P	R	$F_{0.5}$	GLEU		WER
Baseline	42.19	15.28	31.20	53.20	25.18	43.52	47.92	51.77	0.10
WER Token									
⟨1⟩	**52.45**	13.60	33.39	**60.07**	23.52	*45.83	44.85	*48.45	0.06
⟨2⟩	47.55	17.94	**35.75**	54.64	28.41	***46.12**	47.96	*52.01	0.09
⟨3⟩	43.38	20.05	35.19	50.48	31.45	*45.03	**49.45**	***53.59**	0.12
⟨4⟩	40.91	**21.32**	34.56	47.43	**32.68**	43.50	49.16	*53.47	0.17
⟨5⟩	29.48	13.98	24.13	33.77	22.95	*30.86	37.52	*42.21	0.43

Table 3: Results of GEC experiments with controlled degree of correction.

Method	CoNLL-2014 (Test)			JFLEG (Test)
	P	R	$F_{0.5}$	GLEU
Softmax	60.15	24.03	***46.25**	49.07
LM	44.34	20.20	35.79	***53.87**
Oracle WER	72.57	34.40	59.39	58.49
Gold WER	55.25	28.38	46.45	54.48

Table 4: Results of GEC experiments with ranking of the 5 single best candidates. The oracle WER shows the scores when selecting a corrected sentence for each erroneous sentence that maximizes the $F_{0.5}$ on CoNLL-2014 test set and GLEU on JFLEG test set. The gold WER shows the scores when using the WER token calculated from the reference in evaluation datasets.

data by introducing synthetic grammatical errors to the 2007 New York Times Annotated Corpus (LDC2008T19)[3] to the original training data in the same manner as the random noising method done by Xie et al. (2018). We used the CoNLL-2014 test set and JFLEG test set as the test sets and CoNLL-2013 dataset (Ng et al., 2013) and JFLEG dev set as the development sets, respectively.

3.2 Model

We used a multilayer convolutional encoder-decoder neural network without pre-trained word embeddings and re-scoring using the edit operation and language model features (Chollampatt and Ng, 2018) as the GEC model with the same hyperparameters. We conducted the following two experiments. First, we trained the GEC model (baseline) by using the training data as is. Second, we created new training data by adding WER tokens defined by WER to the beginning of sentences in the original training data, and used it to train a GEC model. We added five types of WER tokens to the training data, as shown in Table 2, defined according to the WER score: ⟨1⟩ (the sentence set with the highest WER), ⟨2⟩, ⟨3⟩, ⟨4⟩, and ⟨5⟩ (the sentence set with the lowest WER).

In the ranking experiment, we used a 5-gram KenLM (Heafield, 2011) with Kneser-Ney smoothing trained on the web-scale Common Crawl corpus (Junczys-Dowmunt and Grundkiewicz, 2016).

As an evaluation method, we computed the $F_{0.5}$ score by using the MaxMatch (M^2) scorer (Dahlmeier and Ng, 2012) for the CoNLL-2013 dataset and CoNLL-2014 test set and computed the GLEU score for the JFLEG dev and test sets. In addition, we calculated the average WER of the JFLEG test set.

3.3 Controlling experiment

Table 3 shows the experimental result of controlling the degree of correction using WER. The "WER Token" models are all the same model except for each WER token added to the beginning of the all of input sentences at the time of inference.

The WER in Table 3 show that the average WER is proportional to the WER tokens added to the input sentences. Hence, the WER of the GEC model can be controlled by the WER tokens defined by WER.

The precision is the highest for the WER token ⟨1⟩ and the recall is low. In contrast, the precision is the lowest for the WER token ⟨4⟩, while the recall is the highest. Therefore, the recall is in proportional to the WER, while the precision is inversely proportion to the WER.

However, even with the WER of model ⟨5⟩ being the highest, both its precision and recall are low. In addition, the GLEU and $F_{0.5}$ scores of

[3] https://catalog.ldc.upenn.edu/LDC2008T19

*A statistically significant difference can be observed from the baseline ($p < 0.05$).

Source	Disadvantage is parking their car is very difficult .	WER
Reference	**The disadvantage** is **that** parking their car is very difficult .	0.33
Baseline	Disadvantage is parking their car is very difficult .	0.00
WER Token		
$\langle 1 \rangle$	Disadvantage is parking **;** their car is very difficult .	0.11
$\langle 2 \rangle$	**Disadvantages are** parking their car is very difficult .	0.22
$\langle 3 \rangle$	**The disadvantage** is parking their car is very difficult .	0.22
$\langle 4 \rangle$	**The disadvantage** is **that** parking their car is very difficult .	0.33
$\langle 5 \rangle$	**The disadvantage** is **that** their car **parking lot** is very difficult .	0.56

Table 5: Example of outputs on the JFLEG test set.

model $\langle 5 \rangle$ are the lowest. Table 2 shows the WER of the training data with WER token $\langle 5 \rangle$ is more than 0.5. The manual inspection of this training data revealed that it includes noisy data, for example, very short source sentences or very long target sentences with inserted comments not related to corrections. Consequently, the score is considered to decrease because the training fails.

The degree of correction differs between the CoNLL and JFLEG sets, as described in Section 1. In this result, the WER token with the highest score differs in CoNLL and JFLEG. Moreover, these scores are higher than the baseline scores.

The correction accuracies of both the CoNLL and JFLEG differ for each WER token. Hence, the proposed model can generate diverse corrected sentences by using the WER token.

3.4 Ranking experiment

In the controlling experiment, we obtained the 5 single best candidates with different degrees of correction. Table 4 shows the experimental results of GEC with the ranking of the 5 single best candidates. As shown, these simple ranking methods can decide the best WER token.

The row of softmax in Table 4 shows the result of the ranking of the 5 single best using the softmax score for each sentence. The result shows that the $\mathrm{F}_{0.5}$ score of CoNLL-2014 test set is higher than the scores of the baseline. In contrast, the GLEU score of JFLEG test set is low. The WER in Table 3 shows that the GEC model does not correct much. Hence, the softmax score of the GEC model tends to be high when there are few corrections.

The result of ranking the 5 single best sentences using the LM score is shown in the LM row of Table 4. The GLEU score of JFLEG containing fluency corrections is higher than the scores of the baseline model; however, the $\mathrm{F}_{0.5}$ score of CoNLL-2014 test set containing minimal corrections is low. This outcome is plausible because LM prefers fluency in a sentence regardless of the input.

Table 4 shows the scores of "Oracle WER" when selecting the corrected sentence, which has a higher evaluation score than any other corrected sentences for each input sentence. As a result, $\mathrm{F}_{0.5}$ achieves a score of 59.39 on the CoNLL-2014 test set and GLEU achieves a score of 58.49 on the JFLEG test set. These scores significantly outperform the baseline scores. This could be because the proposed model can generate diverse sentences by controlling the degree of correction. These results imply that the proposed model can be improved by selecting the best corrected sentences.

3.5 Example

Table 5 illustrates outputs of the GEC model with the addition of different WER tokens to the input sentences. This example is obtained from the outputs on the JFLEG test set for each WER token. The bold words represent the parts changed from the source sentence.

This example shows several gold edits to correct grammatical errors in the source sentence. Model $\langle 3 \rangle$ corrects only two of these errors, whereas model $\langle 4 \rangle$ covers all the parts to be corrected. Model $\langle 5 \rangle$ makes further changes although these edits are termed as erroneous corrections. This example confirms that the proposed method corrects errors with different degrees of correction. Although the output of the baseline is not corrected, the proposed method could be used to correct all the errors by performing substantial corrections by using the WER token.

3.6 Analysis

Effect of the WER token. We confirmed how accurately the WER token could control the de-

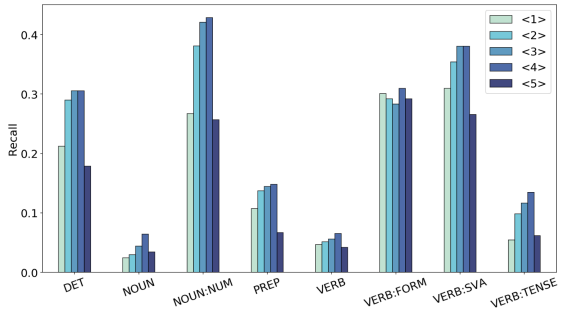

Figure 2: Comparison of the recall of each WER token per error type breakdown , which occurs more than 100 times in the CoNLL-2013 dataset.

gree of correction of model. Therefore, we determined the gold WER tokens for each sentence from the WERs calculated from erroneous and corrected sentences in the CoNLL-2014 test set and JFLEG test set, as shown in Table 2. Then, we calculated the average of the M^2 score, GLEU, and the controlling accuracy because the CoNLL-2014 test set and JFLEG test set have multiple references. The controlling accuracy is the concordance rate of the gold and system WER tokens determined from system output sentences using the gold WER token and erroneous sentences of the CoNLL-2014 test set and JFLEG test set.

The scores of $F_{0.5}$ and GLEU shown in the "Gold WER" row in Table 4 are higher than the baseline scores. However, the scores of $F_{0.5}$ and GLEU are not higher than the oracle WER. Moreover, the controlling accuracy is 62.16 for the CoNLL-2014 test set and 53.18 for the JFLEG test set. This could be because the proposed model corrects less than the degree of correction corresponding to the gold WER token. Specifically, the average number of output sentences below the degree of the correction of the gold WER token is 459.5 within 1,312 sentences in the CoNLL-2014 test set and 64 within 747 sentences in the JFLEG test set. This result shows that it is difficult to estimate of the WER from erroneous sentences. In other words, to improve the correction accuracy, considering GEC methods without relying on WER is necessary.

Error types. We calculated recall to analyze whether the degree of correction can be controlled in more detail for each error type by using ERRANT[4] (Bryant et al., 2017) on the CoNLL-2013 dataset. Figure 2 shows the result of comparison of each WER token and each error type. As the WER increases, the recall increases for almost all error types except for model $\langle 5 \rangle$. Among them, the recall of DET and NOUN:NUM especially increases compared to the recall of VERB and VERB:FORM. This result also shows that the degree of correction can be controlled by using the WER.

4 Related work

Junczys-Dowmunt and Grundkiewicz (2016) used an SMT model with task-specific features, which outperformed previously published results. However, the SMT model can only correct few words or phrases based on a local context, resulting in unnatural sentences. Therefore, several methods using a neural network were proposed to ensure fluent corrections, considering the context and meaning between words. Among them, the method by Chollampatt and Ng (2018) uses a multilayer convolutional encoder-decoder neural network (Gehring et al., 2017). This model is one of the state-of-the-art models in GEC, and its implementation is currently being published[5]. However, these models cannot be controlled in terms of the degree of correction.

Kikuchi et al. (2016) proposed to control the output length by hinting about the output length to the encoder-decoder model in the text summarization task. Sennrich et al. (2016) controlled the politeness of output sentences by adding politeness information to the training data as WER tokens in machine translation. In this research, similar to Sennrich et al. (2016), we added WER indicating the degree of correction as WER tokens to the training data to control the degree of correction for the input sentences.

Similar to our method, Junczys-Dowmunt et al. (2018) and Schmaltz et al. (2017) trained a GEC model with corrective edits information to control the tendency of generating corrections.

5 Conclusion

This study showed that it is possible to control the degree of correction of a neural GEC model by creating training data with WER tokens based on the WER to train a GEC model. Therefore, diverse corrected sentences can be generated from one erroneous sentence. We also showed that the proposed method improved correction accuracy.

[4] https://github.com/chrisjbryant/errant

[5] https://github.com/nusnlp/mlconvgec2018

In the future, we would like to work on selecting the best sentence from a wide variety of corrected sentences generated by a model varying the degree of correction.

Acknowledgments

We thank Yangyang Xi of Lang-8, Inc. for kindly allowing us to use the Lang-8 learner corpus.

References

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proc. of ACL*.

Christopher Bryant and Hwee Tou Ng. 2015. How far are we from fully automatic high quality grammatical error correction? In *Proc. of ACL*.

Shamil Chollampatt and Hwee Tou Ng. 2018. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proc. of AAAI*.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proc. of NAACL-HLT*.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner English: The NUS corpus of learner English. In *Proc. of BEA*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proc. of ICML*.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proc. of WMT*.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Phrase-based machine translation is state-of-the-art for automatic grammatical error correction. In *Proc. of EMNLP*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. In *Proc. of NAACL-HLT*.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proc. of EMNLP*.

Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2012. The effect of learner corpus size in grammatical error correction of ESL writings. In *Proc. of COLING*.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: A fluency corpus and benchmark for grammatical error correction. In *Proc. of EACL*.

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 *Proc. of CoNLL*.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *Proc. of CoNLL*.

Allen Schmaltz, Yoon Kim, Alexander Rush, and Stuart Shieber. 2017. Adapting sequence models for sentence correction. In *Proc. of EMNLP*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine translation via side constraints. In *Proc. of NAACL-HLT*.

Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. 2018. Noising and denoising natural language: Diverse backtranslation for grammar correction. In *Proc. of NAACL-HLT*.