

# Comparison of Grammatical Error Correction Using Back-Translation Models

Aomi Koyama Kengo Hotate\* Masahiro Kaneko† Mamoru Komachi

Tokyo Metropolitan University

koyama-aomi@ed.tmu.ac.jp, kengo\_hotate@r.recruit.co.jp  
masahiro.kaneko@nlp.c.titech.ac.jp, komachi@tmu.ac.jp

## Abstract

Grammatical error correction (GEC) suffers from a lack of sufficient parallel data. Therefore, GEC studies have developed various methods to generate pseudo data, which comprise pairs of grammatical and artificially produced ungrammatical sentences. Currently, a mainstream approach to generate pseudo data is back-translation (BT). Most previous GEC studies using BT have employed the same architecture for both GEC and BT models. However, GEC models have different correction tendencies depending on their architectures. Thus, in this study, we compare the correction tendencies of the GEC models trained on pseudo data generated by different BT models, namely, Transformer, CNN, and LSTM. The results confirm that the correction tendencies for each error type are different for every BT model. Additionally, we examine the correction tendencies when using a combination of pseudo data generated by different BT models. As a result, we find that the combination of different BT models improves or interpolates the  $F_{0.5}$  scores of each error type compared with that of single BT models with different seeds.

## 1 Introduction

Grammatical error correction (GEC) aims to automatically correct errors in text written by language learners. It is generally considered as a translation from ungrammatical sentences to grammatical sentences, and GEC studies use machine translation (MT) models as GEC models. After Yuan and Briscoe (2016) applied an encoder–decoder (EncDec) model (Sutskever et al., 2014; Bahdanau et al., 2015) to GEC, various EncDec-based GEC models have been proposed (Ji et al., 2017; Chollampatt and Ng, 2018; Junczys-Dowmunt et al., 2018; Zhao et al., 2019; Kaneko et al., 2020).

GEC models have different correction tendencies in each architecture. For example, a GEC

model based on CNN (Gehring et al., 2017) tends to correct errors effectively using the local context (Chollampatt and Ng, 2018). Furthermore, some studies have combined multiple GEC models to exploit the difference in correction tendencies, thereby improving performance (Grundkiewicz and Junczys-Dowmunt, 2018; Kantor et al., 2019).

Despite their success, EncDec-based models require considerable amounts of parallel data for training (Koehn and Knowles, 2017). However, GEC suffers from a lack of sufficient parallel data. Accordingly, GEC studies have developed various pseudo data generation methods (Xie et al., 2018; Ge et al., 2018a; Zhao et al., 2019; Lichtarge et al., 2019; Xu et al., 2019; Choe et al., 2019; Qiu et al., 2019; Grundkiewicz et al., 2019; Kiyono et al., 2019; Grundkiewicz and Junczys-Dowmunt, 2019; Wang et al., 2020; Takahashi et al., 2020; Wang and Zheng, 2020; Zhou et al., 2020a; Wan et al., 2020). Moreover, Wan et al. (2020) showed that the correction tendencies of the GEC model are different when using (1) a pseudo data generation method by adding noise to latent representations and (2) a rule-based pseudo data generation method. Furthermore, they improved the GEC model by combining pseudo data generated by these methods. Therefore, the combination of pseudo data generated by multiple methods with different tendencies allows us to improve the GEC model further.

One of the most common methods to generate pseudo data is back-translation (BT) (Sennrich et al., 2016a). In BT, we train a BT model (i.e., the reverse model of the GEC model), which generates an ungrammatical sentence from a given grammatical sentence. Subsequently, a grammatical sentence is provided as an input to the BT model, generating a sentence containing pseudo errors. Finally, pairs of erroneous sentences and their input sentences are used as pseudo data to train a GEC model.

Kiyono et al. (2019) reported that a GEC model using BT achieved the best performance among

\*Current affiliation: Recruit Co., Ltd.

†Current affiliation: Tokyo Institute of Technology

other pseudo data generation methods. However, most previous GEC studies using BT have used the BT model with the same architecture as the GEC model (Xie et al., 2018; Ge et al., 2018a,b; Zhang et al., 2019; Kiyono et al., 2019, 2020). Thus, it is unclear whether the correction tendencies differ when using BT models with different architectures.

We investigated correction tendencies of the GEC model using pseudo data generated by different BT models. Specifically, we used three BT models: Transformer (Vaswani et al., 2017), CNN (Gehring et al., 2017), and LSTM (Luong et al., 2015). The results showed that correction tendencies of each error type are different for each BT model. In addition, we examined correction tendencies of the GEC model when using a combination of pseudo data generated by different BT models. As a result, we found that the combination of different BT models improves or interpolates the  $F_{0.5}$  scores of each error type compared with that of single BT models with different seeds.

The main contributions of this study are as follows:

- We confirmed that correction tendencies of the GEC model are different for each BT model.
- We found that the combination of different BT models improves or interpolates the  $F_{0.5}$  scores compared with that of single BT models with different seeds.

## 2 Related Works

### 2.1 Back-Translation in Grammatical Error Correction

Sennrich et al. (2016a) showed that BT can effectively improve neural machine translation. Therefore, many MT studies focused on BT (Poncelas et al., 2018; Fadaee and Monz, 2018; Edunov et al., 2018; Graça et al., 2019; Caswell et al., 2019; Edunov et al., 2020; Soto et al., 2020; Dou et al., 2020). Subsequently, BT was applied to GEC. For example, Xie et al. (2018) proposed noising beam search methods, and Ge et al. (2018a) proposed back-boost learning. Moreover, Rei et al. (2017) and Kasewa et al. (2018) applied BT to a grammatical error detection task.

Kiyono et al. (2019) compared pseudo data generation methods, including BT. They reported that (1) the GEC model using BT achieved the best performance and (2) using pseudo data for pre-training improves the GEC model more effectively than

using a combination of pseudo data and genuine parallel data. This is because the amount of pseudo data is much larger than that of genuine parallel data. This usage of pseudo data in GEC contrasts with the usage of a combination of pseudo data and genuine parallel data in MT (Sennrich et al., 2016a; Edunov et al., 2018; Caswell et al., 2019).

Htut and Tetreault (2019) compared four GEC models—Transformer, CNN, PRPN (Shen et al., 2018), and ON-LSTM (Shen et al., 2019)—using pseudo data generated by different BT models. Specifically, they used Transformer and CNN as BT models. It was reported that the Transformer using pseudo data generated by CNN achieved the best  $F_{0.5}$  score. However, the correction tendencies for each BT model were not reported. Moreover, although using pseudo data for pre-training is common in GEC (Zhao et al., 2019; Lichtarge et al., 2019; Grundkiewicz et al., 2019; Zhou et al., 2020a; Hotate et al., 2020), they used a less common method of utilizing pseudo data for re-training after training with genuine parallel data. Therefore, we used Transformer as the GEC model and investigated correction tendencies when using Transformer, CNN, and LSTM as BT models. Further, we used pseudo data to pre-train the GEC model.

### 2.2 Correction Tendencies When Using Each Pseudo Data Generation Method

White and Rozovskaya (2020) conducted a comparative study of two rule/probability-based pseudo data generation methods. The first method (Grundkiewicz et al., 2019) generates pseudo data using a confusion set based on a spell checker. The second method (Choe et al., 2019) generates pseudo data using human edits extracted from annotated GEC corpora or replacing prepositions/nouns/verbs with predefined rules. Based on the comparison results of these methods, it was reported that the former has better performance in correcting spelling errors, whereas the latter has better performance in correcting noun number and tense errors. In addition, Lichtarge et al. (2019) compared pseudo data extracted from Wikipedia edit histories with that generated by round-trip translation. They reported that the former enables better performance in correcting morphology and orthography errors, whereas the latter enables better performance in correcting preposition and pronoun errors. Similarly, we reported correction tendencies of the GEC model when using pseudo data generated by three

Dataset	Sents.	Refs.	Split
BEA-train	564,684	1	train
BEA-valid	4,384	1	valid
CoNLL-2014	1,312	2	test
JFLEG	747	4	test
BEA-test	4,477	5	test
Wikipedia	9,000,000	-	-

Table 1: Dataset used in the experiments.

BT models with different architectures.

Some studies have used a combination of pseudo data generated by different methods for training the GEC model (Lichtarge et al., 2019; Zhou et al., 2020a,b; Wan et al., 2020). For example, Zhou et al. (2020a) proposed a pseudo data generation method that pairs sentences translated by statistical machine translation and neural machine translation. Then, they combined pseudo data generated by it with pseudo data generated by BT to pre-train the GEC model. However, they did not report the correction tendencies of the GEC model when using combined pseudo data. Conversely, we reported correction tendencies when using a combination of pseudo data generated by different BT models.

### 3 Experimental Setup

#### 3.1 Dataset

Table 1 shows the details of the dataset used in the experiments. We used the BEA-2019 workshop official shared task dataset (Bryant et al., 2019) as the training and validation data. This dataset consists of FCE (Yannakoudakis et al., 2011), Lang-8 (Mizumoto et al., 2011; Tajiri et al., 2012), NUCLE (Dahlmeier et al., 2013), and W&I+LOCNESS (Granger, 1998; Yannakoudakis et al., 2018). Following Chollampatt and Ng (2018), we removed sentence pairs with identical source and target sentences from the training data. Next, we applied byte pair encoding (Sennrich et al., 2016b) to both source and target sentences. Here, we acquired subwords from the target sentences in the training data and set the vocabulary size to 8,000. Hereinafter, we refer to the training and validation data as BEA-train and BEA-valid, respectively.

We used Wikipedia<sup>1</sup> as a seed corpus to generate pseudo data and removed possibly inappropriate sentences, such as URLs. In total, we extracted 9M sentences randomly.

<sup>1</sup>We used the 2020-07-06 dump file at <https://dumps.wikimedia.org/other/cirrussearch/>.

#### 3.2 Evaluation

We evaluated the CoNLL-2014 test set (CoNLL-2014) (Ng et al., 2014), the JFLEG test set (JFLEG) (Heilman et al., 2014; Napoles et al., 2017), and the official test set of the BEA-2019 shared task (BEA-test). We reported  $M^2$  (Dahlmeier and Ng, 2012) for the CoNLL-2014 and GLEU (Napoles et al., 2015, 2016) for the JFLEG. We also reported the scores measured by ERRANT (Felice et al., 2016; Bryant et al., 2017) for the BEA-valid and BEA-test. All the reported results, except for the ensemble model, are the average of three distinct trials using three different random seeds<sup>2</sup>. In the ensemble model, we reported the ensemble results of the three GEC models.

#### 3.3 Grammatical Error Correction Model

Following Kiyono et al. (2019), we adopted Transformer, which is a representative EncDec-based model, using the fairseq toolkit (Ott et al., 2019). We used the “Transformer (base)” settings of Vaswani et al. (2017)<sup>3</sup>, which has a 6-layer encoder and decoder with a dimensionality of 512 for both input and output and 2,048 for inner-layers, and 8 self-attention heads. We pre-trained GEC models on each 9M pseudo data generated by each BT model<sup>4</sup> and then fine-tuned them on BEA-train. We optimized the model by using Adam (Kingma and Ba, 2015) in pre-training and with Adafactor (Shazeer and Stern, 2018) in fine-tuning. Most of the hyperparameter settings were the same as those described in Kiyono et al. (2019). Additionally, we trained a GEC model using only the BEA-train without pre-training as a baseline model.

We investigated correction tendencies when using a combination of pseudo data generated by different BT models. Therefore, we pre-trained a GEC model on combined pseudo data and then fine-tuned it on the BEA-train. Notably, in this experiment, we combined pseudo data generated by the Transformer and CNN because they improved the GEC models compared with LSTM in most cases (Section 4.1). Specifically, we obtained 9M pseudo data from the Transformer and CNN and then created 18M pseudo data by combining them.

<sup>2</sup>To reduce the influence of the BT model’s seed, we prepared BT models trained with the corresponding seed of each GEC model. Then, we pre-trained each GEC model using pseudo data generated by the corresponding BT models.

<sup>3</sup>Considering the limitation of computing resources, we used “Transformer (base)” instead of “Transformer (big)”.

<sup>4</sup>See Section 3.4 for details of the BT models.

Back-translation model	Pseudo data	CoNLL-2014			JFLEG	BEA-test		
		Prec.	Rec.	F <sub>0.5</sub>	GLEU	Prec.	Rec.	F <sub>0.5</sub>
None (Baseline)	-	58.5/65.8	31.3/31.5	49.8/54.0	53.0/53.7	52.6/61.4	42.8/42.8	50.2/56.5
Transformer	9M	<b>65.0/68.6</b>	<b>37.6/37.7</b>	<b>56.7/59.0</b>	57.7/58.3	61.1/66.5	49.8/50.7	58.4/62.6
CNN	9M	64.0/68.1	37.4/37.4	56.0/58.5	<b>57.8/58.4</b>	<b>61.9/67.5</b>	<b>50.7/51.0</b>	<b>59.3/63.4</b>
LSTM	9M	64.7/ <b>68.8</b>	36.2/36.4	55.9/58.4	57.0/57.4	61.3/67.1	49.5/49.9	58.5/62.8
Transformer & CNN	18M	65.2/ <b>69.1</b>	<b>38.7/39.1</b>	<b>57.3/59.9</b>	<b>57.9/58.5</b>	<b>63.1/67.6</b>	51.1/51.1	<b>60.2/63.5</b>
Transformer & Transformer	18M	65.5/68.3	37.9/38.0	57.2/58.9	57.5/58.0	63.0/67.0	51.0/50.7	<b>60.2/63.0</b>
CNN & CNN	18M	<b>65.6/69.1</b>	38.2/38.7	<b>57.3/59.8</b>	<b>57.9/58.6</b>	61.9/67.1	<b>51.4/51.6</b>	59.5/63.3

Table 2: Results of each GEC model. The left and right scores represent single and ensemble model results, respectively. The top group delineates the performance of the GEC model using each BT model, and the bottom group delineates the performance of the GEC model when using combined pseudo data.

To eliminate the effect of increasing the pseudo data amount, we prepared GEC models that used a combination of pseudo data generated by single BT models with different seeds. We provided all BT models with the same target sentences to focus on the difference in the pseudo source sentences. Hence, in the combined pseudo data, the number of source sentence types increases; however, the number of target sentence types does not increase.

### 3.4 Back-Translation Model

Based on the GEC studies that used BT, we selected the Transformer (Vaswani et al., 2017), CNN (Gehring et al., 2017), and LSTM (Luong et al., 2015). For all BT models, we used implementations of the fairseq toolkit and its default settings, except for common settings<sup>5</sup>.

**Common settings.** We used the Adam optimizer with  $\beta_1 = 0.9$  and  $\beta_2 = 0.98$ . We used label smoothed cross-entropy (Szegedy et al., 2016) as a loss function and selected the model that achieved the smallest loss on the BEA-valid. We set the maximum number of epochs to 40. The learning rate schedule is the same as that described in Vaswani et al. (2017). We applied dropout (Srivastava et al., 2014) with a rate of 0.3. We set the beam size to 5 with length normalization. Moreover, to generate various errors, we used the noising beam search method proposed by Xie et al. (2018). In this method, we add  $r\beta_{\text{random}}$  to the score of each hypothesis in the beam search. Here,  $r$  is randomly sampled from a uniform distribution of interval  $[0, 1]$ , and  $\beta_{\text{random}} \in \mathbb{R}_{\geq 0}$  is a hyperparameter that adjusts the noise scale. In this experiment,  $\beta_{\text{random}}$  was set to 8, 10, and 12 for the Transformer, CNN,

and LSTM, respectively<sup>6</sup>.

**Transformer.** Our Transformer model was based on Vaswani et al. (2017), which is a 6-layer encoder and decoder with 512-dimensional embeddings, 2,048 for inner-layers, and 8 self-attention heads.

**CNN.** Our CNN model was based on Gehring et al. (2017), which is a 20-layer encoder and decoder with 512-dimensional embeddings, both using kernels of width 3 and hidden size 512.

**LSTM.** Our LSTM model was based on Luong et al. (2015), which is a 1-layer encoder and decoder with 512-dimensional embeddings and hidden size 512.

## 4 Results

### 4.1 Overall Results

**Separate pseudo data.** The top group in Table 2 depicts the results of the GEC model using each BT model; the best BT model was different for each test set. The GEC model using the Transformer achieved the best scores in the CoNLL-2014. In contrast, in the JFLEG and BEA-test, the GEC model using CNN achieved the best scores. Moreover, the GEC model using LSTM achieved a higher  $F_{0.5}$  than that using the Transformer in the BEA-test. These results suggest that the Transformer, which is robust as the GEC model (Kiyono et al., 2019), is not necessarily a good BT model.

**Combined pseudo data.** The bottom group of Table 2 shows the results of the GEC model using combined pseudo data. As shown in Table 2, a combination of pseudo data generated by different BT models consistently improved the performance

<sup>5</sup>When training each BT model, the argument `-arch` in the fairseq toolkit was set to `transformer`, `fconv`, and `lstm` for the Transformer, CNN, and LSTM, respectively.

<sup>6</sup>Each  $\beta_{\text{random}}$  achieved the best  $F_{0.5}$  score on the BEA-valid in the preliminary experiments.

Error type	Freq.	Back-translation model						
		Baseline	Transformer	CNN	LSTM	Transformer & CNN	Transformer & Transformer	CNN & CNN
OTHER	697	22.2±1.77	<b>31.8±0.71</b>	31.7±0.77	30.6±0.16	<b>34.2±1.03</b>	31.8±1.01	31.6±0.74
PUNCT	613	65.6±2.02	64.6±0.42	<b>67.8±0.83</b>	67.3±1.83	65.9±1.51	66.0±0.73	<b>67.8±0.93</b>
DET	607	53.8±0.71	64.8±1.62	65.0±0.41	<b>65.2±0.83</b>	64.8±0.64	<b>66.7±1.15</b>	64.7±0.75
PREP	417	48.2±0.55	58.1±0.76	<b>59.3±0.54</b>	55.2±1.74	<b>61.1±0.43</b>	60.3±0.76	60.3±1.06
ORTH	381	72.7±2.47	77.2±0.50	<b>78.7±1.50</b>	78.0±1.95	<b>79.2±1.25</b>	78.4±1.28	78.8±0.74
SPELL	315	58.3±3.49	71.0±1.71	71.1±1.45	<b>71.6±0.50</b>	<b>73.3±1.03</b>	72.5±0.40	71.1±0.49
NOUN:NUM	263	57.8±2.23	<b>64.4±1.09</b>	63.7±0.90	63.9±1.35	66.2±0.43	<b>66.3±0.61</b>	64.6±1.41
VERB:TENSE	256	43.9±2.35	52.1±1.58	<b>54.6±0.94</b>	52.6±0.50	53.7±1.71	54.6±0.64	<b>54.8±1.27</b>
VERB:FORM	213	62.0±2.26	66.7±2.63	<b>67.1±0.46</b>	66.0±1.60	66.3±0.34	<b>66.9±1.54</b>	66.6±1.01
VERB	196	32.5±3.41	36.0±1.18	36.3±0.91	<b>39.7±3.05</b>	<b>42.7±3.83</b>	39.0±0.76	38.2±0.98
VERB:SVA	157	66.1±1.38	73.7±3.00	<b>75.6±0.86</b>	73.8±2.51	75.1±1.04	<b>76.3±1.20</b>	74.3±0.44
MORPH	155	54.0±2.03	61.9±1.97	<b>63.8±1.23</b>	<b>63.8±0.53</b>	64.5±0.62	<b>66.3±1.26</b>	63.8±2.84
PRON	139	43.8±2.00	<b>53.0±2.79</b>	51.8±0.14	49.6±1.93	<b>53.3±1.10</b>	52.7±2.75	<b>53.3±0.46</b>
NOUN	129	19.7±2.04	<b>31.4±0.62</b>	30.2±2.39	30.5±2.17	<b>35.9±2.90</b>	34.5±1.48	32.8±2.80

Table 3: Each error type’s  $F_{0.5}$  of the single models on the BEA-test. We extracted error types with a frequency of 100 or more. The total frequency of all error types was 4,882. For details of error types, see Bryant et al. (2017).

compared with pseudo data from a single source (Transformer & CNN > Transformer, CNN). In contrast, in some of the items in Table 2, the performances of the GEC models using the single BT models with different seeds were lower than that using only a single BT model. For example, when using the Transformer as the BT model, the  $F_{0.5}$  score of the ensemble model using a single BT model was 59.0 on the CoNLL-2014, whereas that using two homogeneous BT models was 58.9 (Transformer & Transformer: 58.9 < Transformer: 59.0). Similarly, for CNN, the  $F_{0.5}$  score of the ensemble model using only a single BT model was 63.4 on the BEA-test, whereas that using two homogeneous BT models was 63.3 (CNN & CNN: 63.3 < CNN: 63.4). Hence, the combination of different BT models enables the construction of a more robust GEC model than the combination of single BT models with different seeds.

## 4.2 Results of Each Error Type

**Separate pseudo data.** The left side of Table 3 illustrates the  $F_{0.5}$  scores of the single models on the BEA-test across various error types. When using the Transformer as the BT model, the performance of PRON was high. In contrast, the performance of PREP, VERB:TENSE, and VERB:SVA was high when using CNN, and the performance of VERB was high when using LSTM, to name a few. Therefore, it is considered that correction tendencies of each error type are different depending on the BT model.

In PUNCT, the performance of the GEC model

using the Transformer was lower than that of the baseline model (Transformer: 64.6 < Baseline: 65.6). Moreover, when using CNN and LSTM as the BT model, the performance of PUNCT improved by only approximately 2 points in  $F_{0.5}$  from the baseline model (CNN: 67.8, LSTM: 67.3 > Baseline: 65.6). It can be seen that this improvement of PUNCT is small compared with that of other error types. Therefore, when using pseudo data generated by BT, PUNCT is considered an error type that is difficult to improve.

**Combined pseudo data.** The right side of Table 3 shows the  $F_{0.5}$  scores of the single models using combined pseudo data on the BEA-test across various error types. Except for 3 of the 14 error types shown in Table 3, the GEC model using Transformer & CNN yielded the higher  $F_{0.5}$  scores than using at least either Transformer & Transformer or CNN & CNN. Therefore, it is considered that the combination of different BT models improves or interpolates performance compared with that of single BT models with different seeds.

In OTHER, the combination of single BT models with different seeds did not improve the performance of OTHER compared with a single BT model (Transformer & Transformer: 31.8 = Transformer: 31.8 and CNN & CNN: 31.6 < CNN: 31.7). Conversely, the combination of different BT models improved the performance of OTHER compared with a single BT model (Transformer & CNN: 34.2 > Transformer: 31.8, CNN: 31.7). Thus, by using different BT models, the GEC model is expected to correct more diverse error types.

Error type	Transformer				CNN				LSTM			
	Token	Type	F <sub>0.5</sub> w/ FT	F <sub>0.5</sub> w/o FT	Token	Type	F <sub>0.5</sub> w/ FT	F <sub>0.5</sub> w/o FT	Token	Type	F <sub>0.5</sub> w/ FT	F <sub>0.5</sub> w/o FT
Overall	64,733,183	12,364,575	58.4	<b>32.7</b>	77,784,638	17,711,223	<b>59.3</b>	31.4	<b>90,205,852</b>	<b>25,502,133</b>	58.5	25.2
OTHER	16,463,382	6,084,184	<b>31.8</b>	<b>10.0</b>	20,237,776	8,453,119	31.7	9.6	<b>29,286,403</b>	<b>13,844,773</b>	30.6	6.3
PUNCT	3,716,117	37,360	64.6	<b>47.1</b>	3,814,449	46,724	<b>67.8</b>	46.1	<b>4,082,594</b>	<b>53,739</b>	67.3	43.1
DET	8,074,615	<b>39,606</b>	64.8	<b>41.5</b>	<b>8,491,264</b>	39,402	65.0	39.2	8,217,106	33,389	<b>65.2</b>	33.0
PREP	6,832,627	19,521	58.1	<b>36.7</b>	7,935,894	23,564	<b>59.3</b>	35.8	<b>8,091,043</b>	<b>25,923</b>	55.2	30.3
ORTH	3,378,022	521,032	77.2	<b>62.7</b>	<b>3,973,439</b>	<b>646,475</b>	<b>78.7</b>	61.4	3,587,805	513,787	78.0	60.0
SPELL	6,620,395	2,795,425	71.0	<b>57.0</b>	11,224,522	4,737,493	71.1	56.3	<b>11,342,223</b>	<b>5,643,091</b>	<b>71.6</b>	50.8
NOUN:NUM	<b>2,241,413</b>	<b>31,939</b>	<b>64.4</b>	<b>45.2</b>	2,149,748	30,205	63.7	43.9	2,177,546	28,226	63.9	41.3
VERB:TENSE	2,585,017	58,935	52.1	<b>27.2</b>	<b>2,599,663</b>	<b>60,266</b>	<b>54.6</b>	26.6	2,418,040	59,207	52.6	22.6
VERB:FORM	1,287,912	47,071	66.7	45.5	1,421,381	<b>48,776</b>	<b>67.1</b>	<b>46.2</b>	<b>1,517,365</b>	48,117	66.0	41.1
VERB	1,821,117	328,147	36.0	<b>18.5</b>	2,201,360	453,181	36.3	17.2	<b>2,704,117</b>	<b>647,785</b>	<b>39.7</b>	12.9
VERB:SVA	761,768	<b>6,564</b>	73.7	52.5	784,762	6,136	<b>75.6</b>	<b>52.8</b>	<b>824,241</b>	6,019	73.8	45.5
MORPH	2,306,204	148,506	61.9	32.5	2,308,793	147,657	<b>63.8</b>	<b>32.6</b>	<b>2,613,870</b>	<b>167,440</b>	<b>63.8</b>	29.2
PRON	810,875	3,642	<b>53.0</b>	<b>14.7</b>	995,686	4,013	51.8	12.7	<b>1,248,554</b>	<b>5,267</b>	49.6	10.9
NOUN	4,402,909	1,888,994	<b>31.4</b>	<b>14.8</b>	6,155,680	2,697,991	30.2	14.4	<b>8,196,758</b>	<b>4,032,482</b>	30.5	9.8

Table 4: Number of edit pair tokens and types in pseudo data generated by each BT model and each error type’s F<sub>0.5</sub> of the single models with and without fine-tuning on the BEA-test. As with Table 3, we extracted error types with a frequency of 100 or more in the BEA-test. FT denotes fine-tuning.

**Effects of different seeds.** Here, we consider the effect of different seeds in the BT model. In some error types in Table 3, the GEC model using single BT models with different seeds has the higher F<sub>0.5</sub> score than that using different BT models. One of the reasons for this is that there exists some variation (i.e., high standard deviation) in the F<sub>0.5</sub> score of each error type, even when changing merely the seed of the BT model. For example, in the GEC model using the Transformer, the standard deviation of DET was 1.62, which is relatively high. Then, the F<sub>0.5</sub> score of DET using Transformer & Transformer was higher than that using Transformer & CNN. Thus, in error types with some variation, using single BT models with different seeds may improve performance compared with using different BT models.

## 5 Discussion

We examined the number of edit pairs in pseudo data generated by each BT model. We annotated pseudo data using ERRANT and extracted edit pairs from the pseudo source sentences and target sentences. Table 4 shows the number of edit pair tokens and types in the pseudo data generated by each BT model. We expected that the higher the number of errors in each error type, the better the F<sub>0.5</sub> score of the GEC model for each error type. However, the results did not show such a tendency. Specifically, when the number of edit pair tokens and types was the highest in each error type, only 6 of the 14 error types had the highest F<sub>0.5</sub> score (ORTH, SPELL, NOUN:NUM, VERB:TENSE, VERB, and MORPH). This fact implies that simply increasing

the number of tokens or types in each error type may not improve each error type’s performance in the GEC model.

Moreover, we investigated the performance of the GEC model with and without fine-tuning. As shown in Table 4, when fine-tuning was not carried out (i.e., pre-training only), the GEC model using the Transformer had the highest F<sub>0.5</sub> score, and there was a 7.5 point difference in F<sub>0.5</sub> between the Transformer and the LSTM (Transformer: 32.7 > LSTM: 25.2). However, interestingly, when fine-tuning was performed, the GEC model using LSTM achieved a better F<sub>0.5</sub> score than that using the Transformer (Transformer: 58.4 < LSTM: 58.5). This result suggests that even if the performance of the GEC model is low in pre-training, it may become high after fine-tuning.

## 6 Conclusions

In this study, we investigated correction tendencies based on each BT model. The results showed that the correction tendencies of each error type varied depending on the BT models. In addition, we found that the combination of different BT models improves or interpolates the F<sub>0.5</sub> score compared with that of single BT models with different seeds.

## Acknowledgments

We would like to thank Lang-8, Inc. for providing the text data. We would also like to thank the anonymous reviewers for their valuable comments. This work was partly supported by JSPS KAKENHI Grant Number 19KK0286.

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. [Neural Machine Translation by Jointly Learning to Align and Translate](#). In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, California.
- Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. [The BEA-2019 Shared Task on Grammatical Error Correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.
- Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. [Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.
- Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. [Tagged Back-Translation](#). In *Proceedings of the Fourth Conference on Machine Translation*, pages 53–63, Florence, Italy. Association for Computational Linguistics.
- Yo Joong Choe, Jiyeon Ham, Kyubyong Park, and Yeol Yoon. 2019. [A Neural Grammatical Error Correction System Built On Better Pre-training and Sequential Transfer Learning](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 213–227, Florence, Italy. Association for Computational Linguistics.
- Shamil Chollampatt and Hwee Tou Ng. 2018. [A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction](#). In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5755–5762, New Orleans, Louisiana. Association for the Advancement of Artificial Intelligence.
- Daniel Dahlmeier and Hwee Tou Ng. 2012. [Better Evaluation for Grammatical Error Correction](#). In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.
- 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 *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.
- Zi-Yi Dou, Antonios Anastasopoulos, and Graham Neubig. 2020. [Dynamic Data Selection and Weighting for Iterative Back-Translation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 5894–5904, Online. Association for Computational Linguistics.
- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. [Understanding Back-Translation at Scale](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.
- Sergey Edunov, Myle Ott, Marc’Aurelio Ranzato, and Michael Auli. 2020. [On The Evaluation of Machine Translation Systems Trained With Back-Translation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2836–2846, Online. Association for Computational Linguistics.
- Marzieh Fadaee and Christof Monz. 2018. [Back-Translation Sampling by Targeting Difficult Words in Neural Machine Translation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 436–446, Brussels, Belgium. Association for Computational Linguistics.
- Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. [Automatic Extraction of Learner Errors in ESL Sentences Using Linguistically Enhanced Alignments](#). In *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, pages 825–835, Osaka, Japan. The COLING 2016 Organizing Committee.
- Tao Ge, Furu Wei, and Ming Zhou. 2018a. [Fluency Boost Learning and Inference for Neural Grammatical Error Correction](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1055–1065, Melbourne, Australia. Association for Computational Linguistics.
- Tao Ge, Furu Wei, and Ming Zhou. 2018b. [Reaching Human-level Performance in Automatic Grammatical Error Correction: An Empirical Study](#). *arXiv preprint arXiv:1807.01270v5 [cs.CL]*.
- Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. [Convolutional Sequence to Sequence Learning](#). In *Proceedings of the 34th International Conference on Machine Learning*, pages 1243–1252, Sydney, Australia. PMLR.
- Miguel Graça, Yunsu Kim, Julian Schamper, Shahram Khadivi, and Hermann Ney. 2019. [Generalizing Back-Translation in Neural Machine Translation](#). In *Proceedings of the Fourth Conference on Machine Translation*, pages 45–52, Florence, Italy. Association for Computational Linguistics.
- Sylviane Granger. 1998. [The computerized learner corpus: a versatile new source of data for SLA research](#). In *Learner English on Computer*, pages 3–18. Addison Wesley Longman, London and New York.

- Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2018. [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*, pages 284–290, New Orleans, Louisiana. Association for Computational Linguistics.
- Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2019. [Minimally-Augmented Grammatical Error Correction](#). In *Proceedings of the 5th Workshop on Noisy User-generated Text*, pages 357–363, Hong Kong, China. Association for Computational Linguistics.
- Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. [Neural Grammatical Error Correction Systems with Unsupervised Pre-training on Synthetic Data](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy. Association for Computational Linguistics.
- Michael Heilman, Aoife Cahill, Nitin Madnani, Melissa Lopez, Matthew Mulholland, and Joel Tetreault. 2014. [Predicting Grammaticality on an Ordinal Scale](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 174–180, Baltimore, Maryland. Association for Computational Linguistics.
- Kengo Hotate, Masahiro Kaneko, and Mamoru Komachi. 2020. [Generating Diverse Corrections with Local Beam Search for Grammatical Error Correction](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2132–2137, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Phu Mon Htut and Joel Tetreault. 2019. [The Unbearable Weight of Generating Artificial Errors for Grammatical Error Correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 478–483, Florence, Italy. Association for Computational Linguistics.
- Jianshu Ji, Qinlong Wang, Kristina Toutanova, Yongen Gong, Steven Truong, and Jianfeng Gao. 2017. [A Nested Attention Neural Hybrid Model for Grammatical Error Correction](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 753–762, Vancouver, Canada. Association for Computational Linguistics.
- Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. [Approaching Neural Grammatical Error Correction as a Low-Resource Machine Translation Task](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 595–606, New Orleans, Louisiana. Association for Computational Linguistics.
- Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. [Encoder-Decoder Models Can Benefit from Pre-trained Masked Language Models in Grammatical Error Correction](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4248–4254, Online. Association for Computational Linguistics.
- Yoav Kantor, Yoav Katz, Leshem Choshen, Edo Cohen-Karlik, Naftali Liberman, Assaf Toledo, Amir Menczel, and Noam Slonim. 2019. [Learning to combine Grammatical Error Corrections](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 139–148, Florence, Italy. Association for Computational Linguistics.
- Sudhanshu Kasewa, Pontus Stenetorp, and Sebastian Riedel. 2018. [Wronging a Right: Generating Better Errors to Improve Grammatical Error Detection](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4977–4983, Brussels, Belgium. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A Method for Stochastic Optimization](#). In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, California.
- Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. [An Empirical Study of Incorporating Pseudo Data into Grammatical Error Correction](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 1236–1242, Hong Kong, China. Association for Computational Linguistics.
- Shun Kiyono, Jun Suzuki, Tomoya Mizumoto, and Kentaro Inui. 2020. [Massive Exploration of Pseudo Data for Grammatical Error Correction](#). *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2134–2145.
- Philipp Koehn and Rebecca Knowles. 2017. [Six Challenges for Neural Machine Translation](#). In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver, Canada. Association for Computational Linguistics.
- Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. [Corpora Generation for Grammatical Error Correction](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3291–3301, Minneapolis, Minnesota. Association for Computational Linguistics.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. [Effective Approaches to Attention-based Neural Machine Translation](#). In *Proceed-*



- ings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.
- Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. [Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners](#). In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 147–155, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.
- Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. [Ground Truth for Grammatical Error Correction Metrics](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 588–593, Beijing, China. Association for Computational Linguistics.
- Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2016. [GLEU Without Tuning](#). *arXiv preprint arXiv:1605.02592v1 [cs.CL]*.
- Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. [JFLEG: A Fluency Corpus and Benchmark for Grammatical Error Correction](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.
- Heewon Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. [The CoNLL-2014 Shared Task on Grammatical Error Correction](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. [fairseq: A Fast, Extensible Toolkit for Sequence Modeling](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.
- Alberto Poncelas, Dimitar Shterionov, Andy Way, Gideon Maillette De Buy Wenniger, and Peyman Passban. 2018. [Investigating Backtranslation in Neural Machine Translation](#). In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 249–258, Alacant, Spain. European Association for Machine Translation.
- Mengyang Qiu, Xuejiao Chen, Maggie Liu, Krishna Parvathala, Apurva Patil, and Jungyeul Park. 2019. [Improving Precision of Grammatical Error Correction with a Cheat Sheet](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 240–245, Florence, Italy. Association for Computational Linguistics.
- Marek Rei, Mariano Felice, Zheng Yuan, and Ted Briscoe. 2017. [Artificial Error Generation with Machine Translation and Syntactic Patterns](#). In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 287–292, Copenhagen, Denmark. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. [Improving Neural Machine Translation Models with Monolingual Data](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. [Neural Machine Translation of Rare Words with Subword Units](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- Noam Shazeer and Mitchell Stern. 2018. [Adafactor: Adaptive Learning Rates with Sublinear Memory Cost](#). In *Proceedings of the 35th International Conference on Machine Learning*, pages 4596–4604, Stockholm, Sweden. PMLR.
- Yikang Shen, Zhouhan Lin, Chin-Wei Huang, and Aaron C. Courville. 2018. [Neural Language Modeling by Jointly Learning Syntax and Lexicon](#). In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, Canada. OpenReview.net.
- Yikang Shen, Shawn Tan, Alessandro Sordani, and Aaron C. Courville. 2019. [Ordered Neurons: Integrating Tree Structures into Recurrent Neural Networks](#). In *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, Louisiana. OpenReview.net.
- Xabier Soto, Dimitar Shterionov, Alberto Poncelas, and Andy Way. 2020. [Selecting Backtranslated Data from Multiple Sources for Improved Neural Machine Translation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3898–3908, Online. Association for Computational Linguistics.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. [Dropout: A Simple Way to Prevent Neural Networks from Overfitting](#). *Journal of Machine Learning Research*, 15(56):1929–1958.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. [Sequence to Sequence Learning with Neural Net-](#)

- works. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112, Montreal, Canada. Curran Associates, Inc.
- Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. [Rethinking the Inception Architecture for Computer Vision](#). In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, Las Vegas, Nevada. Institute of Electrical and Electronics Engineers.
- Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. [Tense and Aspect Error Correction for ESL Learners Using Global Context](#). In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 198–202, Jeju Island, Korea. Association for Computational Linguistics.
- Yujin Takahashi, Satoru Katsumata, and Mamoru Komachi. 2020. [Grammatical Error Correction Using Pseudo Learner Corpus Considering Learner’s Error Tendency](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 27–32, Online. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is All you Need](#). In *Advances in Neural Information Processing Systems 30*, pages 5998–6008, Long Beach, California. Curran Associates, Inc.
- Zhaohong Wan, Xiaojun Wan, and Wenguang Wang. 2020. [Improving Grammatical Error Correction with Data Augmentation by Editing Latent Representation](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2202–2212, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Chencheng Wang, Liner Yang, Yun Chen, Yongping Du, and Erhong Yang. 2020. [Controllable Data Synthesis Method for Grammatical Error Correction](#). *arXiv preprint arXiv:1909.13302v3 [cs.CL]*.
- Lihao Wang and Xiaoqing Zheng. 2020. [Improving Grammatical Error Correction Models with Purpose-Built Adversarial Examples](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 2858–2869, Online. Association for Computational Linguistics.
- Max White and Alla Rozovskaya. 2020. [A Comparative Study of Synthetic Data Generation Methods for Grammatical Error Correction](#). In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 198–208, Seattle, Washington (Online). Association for Computational Linguistics.
- Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. 2018. [Noising and Denoising Natural Language: Diverse Backtranslation for Grammar Correction](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 619–628, New Orleans, Louisiana. Association for Computational Linguistics.
- Shuyao Xu, Jiehao Zhang, Jin Chen, and Long Qin. 2019. [Erroneous data generation for Grammatical Error Correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 149–158, Florence, Italy. Association for Computational Linguistics.
- Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. [A New Dataset and Method for Automatically Grading ESOL Texts](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon. Association for Computational Linguistics.
- Helen Yannakoudakis, Øistein E Andersen, Ardeshir Geranpayeh, Ted Briscoe, and Diane Nicholls. 2018. [Developing an automated writing placement system for ESL learners](#). *Applied Measurement in Education*, 31(3):251–267.
- Zheng Yuan and Ted Briscoe. 2016. [Grammatical error correction using neural machine translation](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386, San Diego, California. Association for Computational Linguistics.
- Yi Zhang, Tao Ge, Furu Wei, Ming Zhou, and Xu Sun. 2019. [Sequence-to-sequence Pre-training with Data Augmentation for Sentence Rewriting](#). *arXiv preprint arXiv:1909.06002v2 [cs.CL]*.
- Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. [Improving Grammatical Error Correction via Pre-Training a Copy-Augmented Architecture with Unlabeled Data](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 156–165, Minneapolis, Minnesota. Association for Computational Linguistics.
- Wangchunshu Zhou, Tao Ge, Chang Mu, Ke Xu, Furu Wei, and Ming Zhou. 2020a. [Improving Grammatical Error Correction with Machine Translation Pairs](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 318–328, Online. Association for Computational Linguistics.
- Wangchunshu Zhou, Tao Ge, and Ke Xu. 2020b. [Pseudo-Bidirectional Decoding for Local Sequence Transduction](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1506–1511, Online. Association for Computational Linguistics.