Improving Grammatical Error Correction with Machine Translation Pairs

Wangchunshu Zhou[1]* **Tao Ge**[2] **Chang Mu**[3] **Ke Xu**[1] **Furu Wei**[2] **Ming Zhou**[2]
[1]Beihang University, Beijing, China
[2]Microsoft Research Asia, Beijing, China
[3]Peking University, Beijing, China
zhouwangchunshu@buaa.edu.cn, kexu@nlsde.buaa.edu.cn
{tage, fuwei, mingzhou}@microsoft.com
1801210867@pku.edu.cn

Abstract

We propose a novel data synthesis method to generate diverse error-corrected sentence pairs for improving grammatical error correction, which is based on a pair of machine translation models (e.g., Chinese→English) of different qualities (i.e., poor and good). The poor translation model can resemble the ESL (English as a second language) learner and tends to generate translations of low quality in terms of fluency and grammaticality, while the good translation model generally generates fluent and grammatically correct translations. With the pair of translation models, we can generate unlimited numbers of *poor→good* English sentence pairs from text in the source language (e.g., Chinese) of the translators. Our approach can generate various error-corrected patterns and nicely complement the other data synthesis approaches for GEC. Experimental results demonstrate the data generated by our approach can effectively help a GEC model to improve the performance and approaching the state-of-the-art single-model performance in BEA-19 and CoNLL-14 benchmark datasets.

1 Introduction

Recent work on grammatical error correction (GEC) has proved that synthetic error-corrected data is helpful for improving GEC models (Ge et al., 2018; Zhao et al., 2019; Lichtarge et al., 2019; Zhang et al., 2019). However, the error patterns generated by the existing data synthesis approaches tend to be limited by either pre-defined rule sets or the seed error-corrected training data (e.g., for back-translation). To generate more diverse error patterns to further improve GEC training, we propose a novel data synthesis approach for GEC, which employs two machine translation (MT) models of different qualities.

Source(Chinese)	你有困难就去找警察
Beginner Translator	You have difficulty to go to the police.
Advanced Translator	Go to the police if you have trouble.
Reference	You can turn to the police when having trouble.

Figure 1: Examples of translations generated by the beginner and advanced translator. The beginner translator is implemented with a phrase-based SMT model with the decreased language model weight; while the advanced translator is a state-of-the-art NMT model. The beginner translator tends to literally translate its source language to English, which resembles the way an English learner writes English sentences; while the advanced translator is capable of generating fluent and grammatically correct sentences. By pairing the results of beginner and advanced translators, we can harvest unlimited grammatically improved sentence pairs, as the red dashed arrow shows.

The main idea of our approach is demonstrated in Figure 1: we use a beginner and an advanced MT model to translate the same sentence in the source language (e.g., Chinese) into English, and pair the poor and good sentence generated by the beginner and advanced translator as an error-corrected sentence pair. This idea is motivated by the studies in English language learning theory (Watcharapunyawong and Usaha, 2013; Bhela, 1999; Derakhshan and Karimi, 2015) which find that ESL (English as a second language) learners tend to compose an English sentence by literally translating from their native language with little consideration of the grammar and the expression custom in English.

In our approach, we develop a phrase-based statistical machine translation (SMT) model but decrease its language model weight to make it act as the beginner translator. With the decreased language model weight, the SMT model becomes less aware of the grammar and the expression custom in English, which simulates the behaviors of ESL learners to produce less fluent translations that may contain grammatical errors. On the other hand, we

*This work was done during the first author's internship at Microsoft Research Asia.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 318–328
November 16 - 20, 2020. ©2020 Association for Computational Linguistics

employ the state-of-the-art neural machine translation (NMT) model as the advanced translator which tends to produce fluent and grammatically correct translations. In this way, we can generate diverse error patterns without being limited by the predefined rule set and the seed error-corrected data.

We conduct experiments in both the BEA-19 (Bryant et al., 2019) and the CoNLL-14 (Ng et al., 2014) datasets to evaluate our approach. Experiments show the *poor→good* sentence pairs generated by our approach can effectively help a GEC model to improve its performance and achieve the state-of-the-art results in the benchmarks.

Our contributions can be summarized as follows:

- We propose a novel data synthesis method to generate diverse error-corrected data for pre-training GEC models based on a pair of machine translation models.
- We conduct an empirical study of the commonly used data synthesis approaches for GEC and find their shortcomings in terms of limited error-corrected patterns which can be well addressed by our proposed method.
- Our proposed approach can effectively help a GEC model improve its performance and approach the state-of-the-art results in both the CoNLL-14 and the BEA-19 benchmarks.

2 Background: SMT vs NMT

In this section, we briefly introduce both SMT and NMT models and discuss some of their characteristics that motivate the proposed approach.

The phrase-based SMT model is based on the noisy channel model. It formulates the translation probability for translating a foreign sentence f into English e as:

$$\text{argmax}_{\mathbf{e}} P(\mathbf{e}|\mathbf{f}) = \text{argmax}_{\mathbf{e}} P(\mathbf{f}|\mathbf{e})P(\mathbf{e}) \quad (1)$$

where $P(\mathbf{e})$ corresponds to an English language model and $P(\mathbf{f}|\mathbf{e})$ is a separate phrase-based translation model. In practice, an SMT model combines the translation model with a language model with weights tuned through minimum error rate training (MERT) (Och, 2003) on a validation set. The role of the language model in SMT models is to avoid literal (i.e., phrase-by-phrase) translation and make generated translation more natural and grammatically correct. Without the language model, its produced translation will become less fluent and more likely to contain grammatical errors.

Translation Model	BLEU	Perplexity
SMT	20.3	23.1
NMT	27.2	15.7

Table 1: The performance (i.e. BLEU score) and the fluency of the output sentences (i.e. Perplexity) of the beginner translator (i.e. SMT model) and the advanced translator (i.e. NMT model) used in our experiments on newstest17 Chinese-English translation test set.

In contrast, a neural machine translation (NMT) model based on sequence-to-sequence architecture is optimized by directly maximizing the likelihood of the target sentences given source sentences $P(\mathbf{e}|\mathbf{f})$. It proves effective to generate adequate and fluent translations, and substantially outperforms SMT models in most cases.

Table 1 gives a comparison of SMT and NMT in newstest17 Chinese-English news translation dataset. It can be observed that the SMT model is inferior to the NMT model in terms of both the translation quality (reflected by **BLEU**) and the fluency (reflected by **Perplexity**[1]).

3 *Poor→Good* Sentence Pair Generation

As discussed above, an SMT model is generally inferior to an NMT model in terms of both fluency and translation quality. Motivated by the fact, we propose to employ an SMT model as a **beginner translator**, and an NMT model as an **advanced translator**. We use both the translators to translate the same sentences in the source language of the MT models into English, obtaining *poor→good* English sentence pairs. These fluency-improving sentence pairs prove helpful in improving the performance of GEC models, according to the previous work (Ge et al., 2018; Zhang et al., 2019); thus they can be used as augmented data for pre-training a GEC model. The overview of our approach is illustrated in Figure 2.

3.1 Poor Sentence Generation

To generate poor sentences that contain grammatical errors, we employ a beginner translator, which is implemented through a phrase-based SMT model, to translate sentences from monolingual corpora in the source language (e.g. Chinese) to En-

[1] The perplexity of output sentences is measured by GPT-2 (Radford et al., 2019).

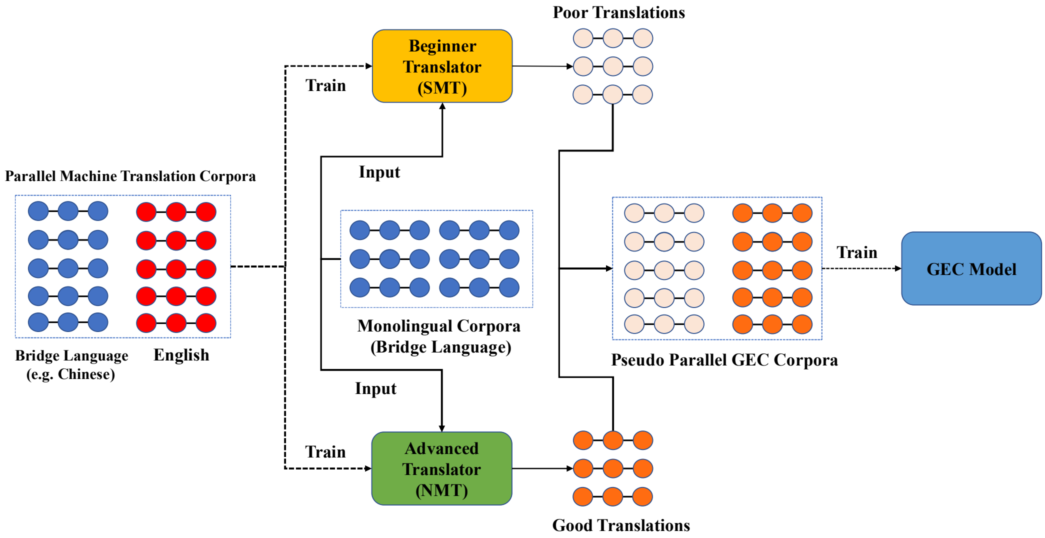
Poor Translations
Beginner Translator (SMT)
Train
Input
Parallel Machine Translation Corpora
Bridge Language (e.g. Chinese)
English
Monolingual Corpora (Bridge Language)
Pseudo Parallel GEC Corpora
Train
GEC Model
Input
Train
Advanced Translator (NMT)
Good Translations

Figure 2: The overview of our approach. Our approach consists of two machine translation models with different qualities that are trained with bi-lingual parallel corpora. The beginner translator and the advanced translator generate poor and good translations respectively. We can thus establish error-corrected data using the *poor→good* translation pairs of sentences in the source language. (Best viewed in color)

glish. To make the generated sentences less fluent and poor enough, we propose to decrease the tuned language model weight of the SMT model. The resulting beginner translator tends to generate translations that resemble the sentences composed by ESL learners: they translate phrase by phrase from their native (i.e., source) language into English but combine the phrase translations in an unnatural way with little awareness of grammar in English.

We present some samples generated by a automatically tuned SMT model and its counterpart with decreased language model weights respectively in Table 2. We can see the translation generated by the SMT model with the decreased language model weight contains more grammatical errors than the automatically tuned SMT model. Such poor sentences can provide more diverse error-corrected learning signals that benefit training a GEC model.

3.2 Good Sentence Generation

To generate good counterparts for the poor sentences, we use an advanced translator, which is implemented with a state-of-the-art NMT model, to generate good translations from the source sentences. Since both the poor translations and good translations are translated from the same source sentences, they tend to express the same meaning but in different ways. As observed in Table 2, compared with the output sentences from the SMT models, the NMT model's outputs are generally more fluent and native-sounding. Therefore, we can pair the poor sentences with the good ones to establish *poor→good* sentence pairs as potentially useful training instances for GEC models.

The aforementioned method can be used to generate poor and good sentences from monolingual corpora in the source language (e.g., Chinese). It can be even more easily used when MT parallel corpora (e.g., Chinese-English) are available. For MT parallel datasets whose source sentences' ground truth English translations are available, we can directly use their ground truth English sentences as the good sentences without the need to use the NMT model to get the good translations from the sources. Since bi-lingual parallel data for MT is much more than that for GEC, it is also a feasible solution to collect the *poor→good* sentence pairs in this way.

4 Models

4.1 MT Models

We train both the beginner translator and the advanced translator with the Chinese-English parallel corpus – UN Corpus (Ziemski et al., 2016), which contains approximately 15M Chinese-English parallel sentence pairs with around 400M tokens.

Source(Chinese)	两国复交符合两国人民的共同利益.
SMT$_{original}$	Diplomatic rapprochement between **the** two countries is the common interest of the peoples of the two countries.
SMT$_{low}$	Diplomatic rapprochement between two countries **in** the common interest of the **two** peoples of the two countries .
NMT	The resumption of diplomatic relations between the two countries is in the common interest of the two peoples.
Source(Chinese)	以后的生活肯定还会一年比一年好.
SMT$_{original}$	Life will **be** for **the** rest of their lives better over **the** years .
SMT$_{low}$	Life will for the **their** rest of lives better over years .
NMT	Life will definitely be better every year.

Table 2: Examples of translation results generated by SMT models with different language model weights and that generated by the NMT model. The differences between the poor sentences generated by SMT$_{original}$ and SMT$_{low}$ are bolded. We can see that translations generated by SMT with decreased language model weight (i.e., SMT$_{low}$) contains more grammatical errors, while the NMT model produces fluent and grammatically correct sentences.

Beginner Translator We use Moses (Koehn et al., 2007) to implement the phrase-based SMT model for the beginner translator. Specifically, we use MGIZA++ (Gao and Vogel, 2008) for word-alignment, and KenLM (Heafield, 2011) for training a trigram language model on the target sentences of the Chinese-English parallel data. We tune the weights of each component in the Moses system (e.g. phrase table, language model, etc.) using MERT (Och, 2003) to optimize the system's BLEU score (Papineni et al., 2002) on a development set that is constructed by randomly sampling 5,000 sentence pairs from the parallel corpus. To make the SMT's outputs worse, as Section 3.1 discusses, we decrease the weight of the language model by 20%. To distinguish the automatically tuned SMT model and the one with the decreased language model weight, we call them SMT$_{original}$ and SMT$_{low}$ respectively.

Advanced Translator Model We use the Transformer-based NMT model as the advanced translator. Specifically, we use the "transformer-big" architecture (Vaswani et al., 2017). Chinese sentences are segmented into word-level. Afterward, both Chinese words and English words are split into subwords using the byte-pair encoding technique (Sennrich et al., 2015). The vocabulary size is 32K for both Chinese and English. We train the advanced translator with Adam optimizer (Kingma and Ba, 2014) with the learning rate 0.0003 and the dropout rate 0.3. We warm-up the learning rate during the first 4K updates and then decrease proportionally to the inverse square root of the number of steps. The resulting model yields a BLEU score of 27.2 on newstest17 Chinese-English translation test set, which is competitive to state-of-the-art results. We use beam search with beam size of 4 when using it to generate good translations.

4.2 GEC Model

As for the GEC model, we use the same "transformer-big" architecture as our GEC model with tied output layer, decoder embedding, and encoder embeddings. Both input and output sentences are tokenized with byte-pair encoding with shared codes consisting of 30,000 token types. Following the previous work (Zhang et al., 2019), we train the GEC models on 8 Nvidia V100 GPUs, using Adam optimizer (Kingma and Ba, 2014) with β_1=0.9, β_2=0.98. We allow each batch to have at most 5,120 tokens per GPU. During pre-training, we set the learning rate to 0.0005 with linear warm-up for the first 8k updates, and then decrease proportionally to the inverse square root of the number of steps. For fine-tuning, the learning rate is set to 0.0001 with warmup over the first 4,000 steps and inverse square root decay after warmup. The dropout ratio is set to 0.2 in both pre-training and fine-tuning stages. We pre-train the model for 200k steps and fine-tune it up to 50k steps.

We use the synthesized data generated with different data synthesis approaches for pre-training GEC models. Then, we use the GEC training data (see Section 5.1) to fine-tune the pre-trained models. We select the best model checkpoint according to the perplexity on the BEA-19 validation set for both the pre-training and fine-tuning. We use beam search to decode with the beam size of 12.

5 Experiments

5.1 Data

Following the previous work (Grundkiewicz et al., 2019; Choe et al., 2019; Kiyono et al., 2019) in GEC, the GEC training data we use is the public Lang-8 (Mizumoto et al., 2011),

Dataset	#sent(pairs)	Split
SMT-NMT pairs	15M	pre-train
SMT-gold pairs	15M	pre-train
corruption	60M	pre-train
back-translation	60M	pre-train
Lang-8	1.04M	fine-tune
NUCLE	57.1K	fine-tune
FCE	28.4K	fine-tune
W&I train	34.3K	fine-tune
W&I valid	4,384	valid
W&I test (BEA-19)	4,477	test
CoNLL-14	1,312	test

Table 3: Statistics of the datasets used for pre-training and fine-tuning.

Method	BEA-19	CoNLL-14
Junczys-Dowmunt et al. (2018)	-	53.0
Lichtarge et al. (2019)	-	56.8
Zhao et al. (2019)	-	59.8
Choe et al. (2019)	63.1	60.3
Kiyono et al. (2019)	64.2	61.3
Ours	**65.2**	**62.1**

Table 4: The comparison of our single model against the state-of-the-art single models in the previous work in the BEA-19 and CoNLL-2014 test set. "-" denotes that the previous work does not report its single model's performance in the test set. It is notable that among the previous work in this table, the first three do not use W&I+LOCNESS for training the model. We do not compare with the work that does not report its single model's performance in either of the test sets.

NUCLE (Dahlmeier et al., 2013), FCE (Yannakoudakis et al., 2011) and W&I+LOCNESS datasets (Bryant et al., 2019; Granger, 1998).

To generate the *poor→good* English sentence pairs that benefit GEC training, we collect monolingual Chinese news corpora – Chinese Gigaword (Graff and Chen, 2005) and news2016zh (Xu, 2019) – to generate poor and good English translations using the SMT and NMT model respectively. After filtering[2], we obtain 15M *poor→good* English sentence pairs. In addition, we generate 15M poor English sentences from the Chinese-English parallel corpus – UN Corpus – by translating the Chinese sentences with the beginner translator, and then pair them with their ground truth English translations as *poor→good* sentence pairs.

We additionally include 60M sentence pairs obtained by the corruption based approach (Zhao et al., 2019), 60M pairs from the round-trip translation approach (Lichtarge et al., 2019), and 60M pairs by the fluency boost back-translation approach (Ge et al., 2018) for GEC pre-training. Specifically, the corruption-based and round-trip translation data is obtained from the NewsCrawl dataset; while the back-translated data is harvested from English Wikipedia with a backward model trained on the public Lang-8 and NUCLE dataset.

We evaluate the performance of GEC models on the BEA-19 and the CoNLL-14 benchmark datasets. Following the latest work in GEC (Lichtarge et al., 2019; Zhao et al., 2019; Grundkiewicz et al., 2019; Kiyono et al., 2019; Choe et al., 2019; Zhou et al., 2020; Omelianchuk et al., 2020), we evaluate the performance of trained GEC models using $F_{0.5}$ on test sets using official scripts[3] in both datasets. The data sources used for pretraining, fine-tuning, and evaluating the GEC models are summarized in Table 3.

5.2 Results

Table 4 compares the performance of our model to the previous studies in the same test sets. It is notable that all the results in Table 4 are the single model's result with beam search decoding. We do not compare to the results obtained with additional inference methods like iterative decoding and reranking with an LM or a right-to-left GEC model, because they are not related to our contributions.

According to Table 4, our model outperforms the state-of-the-art single model results in both BEA-19 and CoNLL-14 test set. The main difference between our model and the previous state-of-the-art models is that we additionally use the *poor→good* English sentence pairs obtained from the pair of MT models, accounting for the improvement over the previous work.

To conduct an in-depth analysis of the improvement by the synthesized data, we compare the performance of GEC models pre-trained with different data sources. According to Table 5, the model pre-trained with the 30M synthesized sentence pairs from the beginner and advanced translator outperforms its counterparts that are pre-trained with the same amount of data synthesized from back-translation, round-trip translation, and corruption-based approaches, demonstrating that

[2] We discard the sentence pairs whose edit rate (i.e., the edit distance normalized by the source sentence's length) is larger than 0.6.

[3] M2scorer for CoNLL-14; Errant for BEA-19.

Method	**BEA-19**	**CoNLL-14**
Baseline	57.1	51.5
Pre-train with 30M synthesized data & fine-tune		
Corr(30M)	59.5	55.7
RT(30M)	58.9	55.2
BT(30M)	59.4	55.9
Ours(30M)	**60.4**	**56.6**
Pre-train with 60M synthesized data & fine-tune		
Corr(60M)	59.9	55.9
RT(60M)	59.7	55.8
BT(60M)	60.5	56.5
Corr(30M) + RT(30M)	61.2	57.1
Ours(30M)+Corr(30M)	61.9	57.7
Ours(30M)+BT(30M)	**63.1**	**58.5**

Table 5: The performance of GEC models pre-trained with various synthesized data and fine-tuned with the GEC training data. **Ours** denotes the synthesized data generated with our approach, **Corr**,**RT**, and **BT** denotes the synthesized data generated by random corruption, round-trip translation, and back-translation. **Baseline** denotes the GEC model directly trained with the GEC training data without pre-training.

Method	**BEA-19**	**CoNLL-14**
Pre-train with 30M synthesized data		
Corr(30M)	37.1	27.5
Round-trip(30M)	39.7	29.2
BT(30M)	**45.1**	**33.2**
Ours(30M)	43.5	31.1
Pre-train with 60M synthesized data		
Corr(60M)	38.7	28.1
RT(60M)	40.2	29.8
BT(60M)	47.4	35.5
Corr(30M) + RT(30M)	47.2	34.7
Ours(30M)+Corr(30M)	45.6	33.9
Ours(30M)+BT(30M)	**49.5**	**36.2**

Table 6: The performance of GEC models pretrained with different synthesized data without fine-tuning on GEC training data.

Method	**BEA-19**	**CoNLL-14**
Pre-training Only		
Ours(30M)	**43.5**	**31.1**
- w/o SMT-NMT (15M)	40.1	28.8
- w/o SMT-gold (15M)	39.5	28.4
Pre-training + Fine-tuning		
Ours(30M)	**60.4**	**56.6**
- w/o SMT-NMT (15M)	58.2	55.2
- w/o SMT-gold (15M)	57.8	54.8

Table 7: The ablation study for comparing the contribution of the SMT-NMT and SMT-gold pairs to the final results.

Method	**BEA-19**	**CoNLL-14**
Pre-training Only		
Ours(30M)	**43.5**	**31.1**
- w/o decreased LM score	42.7	30.3
Pre-training + Fine-tuning		
Ours(30M)	**60.4**	**56.6**
- w/o decreased LM score	59.1	55.8

Table 8: The ablation study to test the effect of decreasing the language model weight of the SMT model in the final results.

our approach provides more valuable and diverse error-corrected learning signals for GEC models. When we increase the amount of the synthesized data for pre-training to 60M, we observe that the models pre-trained with a single data source (i.e., **Corr**, **BT**,and **RT**) improve only a little over their 30M counterparts, indicating that their generated error-corrected patterns are limited, which is consistent with the observation of the previous studies (Edunov et al., 2018). In contrast, if combining multiple data sources for pre-training, the performance will be significantly improved. Among them, the GEC models pre-trained with the *poor→good* sentence pairs yield the best results, which demonstrates that the error-corrected patterns provided by the *poor→good* sentence pairs generated through our approach are different from those by back-translation and corruption-based approaches and they can nicely complement each other to achieve a better result.

Moreover, we test the performance of the pretrained models without fine-tuning to see the quality of the synthesized data. According to Table 6, among the 30M single data sources, the data generated by back translation yields the best performance in both test sets, because back translation introduces informative error-corrected patterns with much less undesirable noise than the corruption-based approach and our approach. When we double the data size for pre-training, we observe the similar results to those in Table 5: the combinations of different sources of the synthesized data lead to the best results, verifying our assumption that the error-corrected patterns of different data sources are different and they are complementary.

Source Sentence	“我们应该保持身体健康”
Ground-truth Translation	“We should stay healthy.”
Translation from NMT	“We should stay healthy.”
Translation from SMT	“We should keep a body healthy.”
Rule-based Corruption	“We _ stays stay healthy.”
Back Translation	“We should _ healthy.”
Round-trip Translation	“We should stay healthy.”
Source Sentence	“无论如何, 我对大家的表现都很满意”
Ground-truth Translation	“Anyway , I am very satisfied with everyone’s performance .”
Translation from NMT	“Anyway , I am satisfied with everyone’s performance .”
Translation from SMT	“Regardless of whether such to what , I am very satisfied with both the performance of together.”
Rule-based Corruption	“Anyway , I _ very satisfy with with everyone’s _ .”
Back Translation	“Anyway , I was satisfied with everyone performance .”
Round-trip Translation	“Anyway , I am very satisfied with everyone’s performance .”
Source Sentence	“在大众并不了解AI 技术时更是如此.”
Ground-truth Translation	“This is especially true when the public does not understand AI technology .”
Translation from NMT	“This is particularly true when the public does not understand AI technology .”
Translation from SMT	“Popular understanding of AI technology not at the time is even more true .”
Rule-based Corruption	“This are especially _ when _ public not do understand AI AI technology .”
Back Translation	“This is especially true when _ public do not understand the AI technology .”
Round-trip Translation	“This is particularly true when the public does not understand AI technology .”

Table 9: Examples of translations generated by the beginner translator and the advanced translator, together with synthetic erroneous sentences generated by existing approaches.

5.3 Analysis of the synthesized *poor→good* sentence pairs

As mentioned in Section 5.1, among the 30M synthesized *poor→good* sentence pairs, 15M are SMT→NMT pairs, while the others are SMT→ground-truth translation pairs. We perform an ablation study to analyze how much they separately contribute to GEC models. According to Table 7, the sentence pairs with ground truth translation as their good sentences yield better results. The reason is easy to understand: the quality of the ground truth translations is generally better than that of the advanced translator. However, given the fact that bi-lingual parallel data is much less than the monolingual text data, it is more practical to use the beginner and advanced translator to generate the *poor→good* sentence pairs that will not be limited by bi-lingual parallel corpora. Moreover, the results in Table 8 show that decreasing the language model weight leads to more than 1.0 absolute improvement in $F_{0.5}$ score in both test sets, because it can help the SMT model to act more like a beginner translator, which can also be illustrated by the examples in Table 2.

We also conduct a qualitative analysis of different data synthesis approaches in Table 9. We can see that the beginner translator (i.e., SMT) generates less fluent English sentences by literally (phrase by phrase) translating the source sentence, which resembles how the ESL learners write an English sentence. The advanced translator (i.e., NMT) generates high-quality English sentences that can be comparable to the ground-truth translations. Such diverse *poor→good* sentence pairs are helpful to teacher a GEC model how to rewrite a poor sentence into a good one, accounting for the improved results we achieved. In contrast, existing data synthesis approaches such as rule-based corruption and back-translation tends to generate similar error patterns such as verb/noun forms and word deletions, while the round-trip translation method generates limited modifications which are often paraphrase-like and grammatically correct.

Method	**Error Rate**	**% Error in Rules**
Real Data	21.3	87.2
Ours	45.3	68.6
Rule-based Corruption	40.3	100
Round-trip Translation	6.2	61.7
Back-translation	25.7	99.2

Table 10: Error-type analysis of different data sources. Error in rules represent the ratio of errors that are **not** noted as ”other” or ”unknown” errors by ERRANT.

We then conduct an in-depth analysis of the error-type contrained in the synthetic data generated by our approach and other data synthesis methods using ERRANT (Bryant et al., 2017). The result is in 10. We find that almost all error generated by rule-based corruption and back-translation are included in the pre-defined rules in ERRANT system, while many errors in the real datasets are beyond these rules. This mismatch in the distribution of error types can often severely impact the performance of data synthesis techniques for grammar correction (Felice et al., 2014). In contrast, our method can generate much more diverse error patterns that are not limited by the pre-defined error types, which may account for the performance gain.

6 Related Work

Grammatical error correction (GEC) is a well-established natural language processing (NLP) task that aims to build systems for automatically correcting errors in written text, particularly in non-native written text. While recently sequence tagging (Ribeiro et al., 2018; Omelianchuk et al., 2020) or word substitution with pre-trained language model (Zhou et al., 2019; Li et al., 2020) based GEC models have shown promising results and improved efficiency, seq2seq-based GEC models still remain to be the mainstream method for automated gramamtical error correction. However, as shown in Table 3, the combination of parallel GEC corpora only yields less than 1.5M sentence pairs, which makes it hard to train large neural models (e.g. transformers) to achieve better results. Prior studies (Rei et al., 2017; Xie et al., 2018; Ge et al., 2018; Zhao et al., 2019; Kiyono et al., 2019) have investigated various approaches to alleviate the data scarcity problem for training GEC models by synthesizing pseudo-parallel GEC data for pretraining GEC models. We introduce the most commonly used data synthesis approaches for pre-training GEC models and discuss their pros and cons in this section.

Rule-based Monolingual Corpora Corruption A straightforward data synthesis method is to corrupt monolingual corpora with either pre-defined rules or errors extracted from the seed parallel GEC data (Foster and Andersen, 2009; Zhao et al., 2019; Wang et al., 2019). The advantage of this approach is that it is very simple and efficient for generating pseudo-parallel GEC data from monolingual corpora. However, manually designed rules are limited and only cover a small portion of grammatical error types written by ESL learners. This makes the improvement yielded by pretraining exclusively with synthetic data generated by this approach very limited, which is demonstrated in our experiments.

Back-translation based Error Generation This approach trains an error generation model by using the existing error-corrected corpora in the opposite direction and introduces noise into a clean corpus (Rei et al., 2017; Xie et al., 2018; Ge et al., 2018). Concretely, an error generation model is trained to take a correct sentence as input and outputs an erroneous version of the original sentence. It is used to synthesize error-corrected data by taking monolingual corpora as input. This approach is able to cover more diverse error types compared with rule-based corruption method. However, it requires a large amount of annotated error-corrected data, which is not always available, to train the error generation model. In addition, the error patterns generated by this method are generally limited to that contained in the GEC parallel data, which makes the effect of synthetic data generated by back-translation quickly saturates as the amount of synthetic data grows, as demonstrated in our experiments.

Data Generation from Wikipedia Revision This approach is based on revision histories from Wikipedia (Cahill et al., 2013; Lichtarge et al., 2019). Specifically, it extracts source-target pairs from Wikipedia edit histories by taking two consecutive snapshots as a single revision to the page to form the error-corrected sentence pairs. This method is able to collect human-made revisions that may better resemble real error-corrected data. However, the vast majority of extracted revisions are not grammatical error corrections, which makes the synthesized data noisy and requires sophisticated filtering before used for pre-training. In addition, the domain of available revision history is limited, which makes this method less general compared to other approaches that can generate synthetic data using any monolingual corpora.

Data Generation from Round-trip Translations Round-trip translation (Désilets and Hermet, 2009; Madnani et al., 2012; Lichtarge et al., 2019) is an alternative approach to synthesis pseudo-parallel data for GEC pre-training with monolingual corpora. This approach attempts to introduce noise via bridge translations. Specifically, it uses two state-of-the-art NMT models, one from English to a bridge language and the other from the bridge language to English. With the MT models, it takes the original sentences from monolingual corpora as the target sentences, and takes the outputs of the round-trip translation as the corresponding source sentences. However, when good translation models are employed, as in the case of (Lichtarge et al., 2019), the resulting source sentences are very likely to be clean and without grammatical errors; on the other hand, when poor machine translation models, such as the SMT models with decreased language model weight, are employed, it may result in a semantic drift from target sentences because two consecutive low quality translation are made, which is undesirable for training GEC models. In contrast, the difference between the two machine translation models employed in our approach ensures that the source sentences are of low fluency and contain many grammatical errors.

7 Conclusion and Future Work

We propose a novel method to synthesize *poor→good* sentence pairs for pre-training GEC models based on a pair of MT models of different quality. The generated sentence pairs contain diverse error-corrected patterns that can nicely complement other data augmentation approaches, leading to a performance approaching the state-of-the-art single model results in GEC benchmarks. For future work, we plan to investigate the influence of different source languages of the MT models in the performance of GEC, which might be helpful in building a customized English GEC model for the people speaking a specific foreign language.

Acknowledgments

We thank the anonymous reviewers for their valuable comments.

References

Baljit Bhela. 1999. Native language interference in learning a second language: Exploratory case studies of native language interference with target language usage.

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.

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 (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.

Aoife Cahill, Nitin Madnani, Joel Tetreault, and Diane Napolitano. 2013. Robust systems for preposition error correction using wikipedia revisions. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 507–517.

Yo Joong Choe, Jiyeon Ham, Kyubyong Park, and Yeoil Yoon. 2019. A neural grammatical error correction system built on better pre-training and sequential transfer learning. *arXiv preprint arXiv:1907.01256*.

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.

Ali Derakhshan and Elham Karimi. 2015. The interference of first language and second language acquisition. *Theory and Practice in language studies*, 5(10):2112–2117.

Alain Désilets and Matthieu Hermet. 2009. Using automatic roundtrip translation to repair general errors in second language writing. *MT Summit XII*, pages 198–206.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Mariano Felice, Zheng Yuan, Øistein E Andersen, Helen Yannakoudakis, and Ekaterina Kochmar. 2014. Grammatical error correction using hybrid systems and type filtering. Association for Computational Linguistics.

Jennifer Foster and Oistein Andersen. 2009. Generrate: generating errors for use in grammatical error detection. In *Proceedings of the fourth workshop on innovative use of nlp for building educational applications*, pages 82–90.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software engineering, testing, and quality assurance for natural language processing*, pages 49–57.

Tao Ge, Furu Wei, and Ming Zhou. 2018. Fluency boost learning and inference for neural grammatical error correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1055–1065.

David Graff and Ke Chen. 2005. Chinese gigaword. *LDC Catalog No.: LDC2003T09, ISBN*, 1:58563–58230.

Sylviane Granger. 1998. *The computer learner corpus: a versatile new source of data for SLA research*. na.

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.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197. 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. *arXiv preprint arXiv:1804.05940*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. An empirical study of incorporating pseudo data into grammatical error correction. *arXiv preprint arXiv:1909.00502*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Yiyuan Li, Antonios Anastasopoulos, and Alan W Black. 2020. Towards minimal supervision bert-based grammar error correction (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13859–13860.

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. *arXiv preprint arXiv:1904.05780*.

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Exploring grammatical error correction with not-so-crummy machine translation. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 44–53, Montréal, Canada. 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.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The conll-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. Gector–grammatical error correction: Tag, not rewrite. *arXiv preprint arXiv:2005.12592*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Marek Rei, Mariano Felice, Zheng Yuan, and Ted Briscoe. 2017. Artificial error generation with machine translation and syntactic patterns. *arXiv preprint arXiv:1707.05236*.

Joana Ribeiro, Shashi Narayan, Shay B Cohen, and Xavier Carreras. 2018. Local string transduction as sequence labeling. In *Proceedings of the 27th international conference on computational linguistics*, pages 1360–1371.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

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*, pages 5998–6008.

Chencheng Wang, Liner Yang, Yun Chen, Yongping Du, and Erhong Yang. 2019. Controllable data synthesis method for grammatical error correction. *arXiv preprint arXiv:1909.13302*.

Somchai Watcharapunyawong and Siriluck Usaha. 2013. Thai efl students' writing errors in different text types: The interference of the first language. *English Language Teaching*, 6(1):67–78.

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, Volume 1 (Long Papers)*, pages 619–628, New Orleans, Louisiana. Association for Computational Linguistics.

Liang Xu. 2019. Large-scale chinese datasets for nlp. In *Proceedings of NLPCC2019*.

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-Volume 1*, pages 180–189. 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:1909.06002*.

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. *arXiv preprint arXiv:1903.00138*.

Wangchunshu Zhou, Tao Ge, and Ke Xu. 2020. Pseudo-bidirectional decoding for local sequence transduction. *arXiv preprint arXiv:2001.11694*.

Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. 2019. Bert-based lexical substitution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3368–3373.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1. 0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 3530–3534.