Data Augmentation of Incorporating Real Error Patterns and Linguistic Knowledge for Grammatical Error Correction

Xia Li and **Junyi He**
Guangzhou Key Laboratory of Multilingual Intelligent Processing,
School of Information Science and Technology,
Guangdong University of Foreign Studies, Guangzhou, China
xiali@gdufs.edu.cn

Abstract

Data augmentation aims at expanding training data with clean text using noising schemes to improve the performance of grammatical error correction (GEC). In practice, there are a great number of real error patterns in the manually annotated training data. We argue that these real error patterns can be introduced into clean text to effectively generate more real and high quality synthetic data, which is not fully explored by previous studies. Moreover, we also find that linguistic knowledge can be incorporated into data augmentation for generating more representative and more diverse synthetic data. In this paper, we propose a novel data augmentation method that fully considers the real error patterns and the linguistic knowledge for the GEC task. We conduct extensive experiments on public data sets and the experimental results show that our method outperforms several strong baselines with far less external unlabeled clean text data, highlighting its extraordinary effectiveness in the GEC task that lacks large-scale labeled training data.

1 Introduction

Grammatical Error Correction (GEC) is the task of automatically detecting and correcting grammatical errors of texts into natural and correct forms, which is an important topic in the field of education, especially in language learning.

In recent years, different methods have been proposed to improve the performance of the GEC models. Some early studies (Gamon et al., 2008; Tetreault et al., 2010; Dahlmeier and Ng, 2011; Berend et al., 2013; Rozovskaya and Roth, 2014) take GEC as a classification task and rely much on hand-crafted rules. More recently, the technique of neural machine translation is applied to GEC and has made remarkable performance (Zhao et al., 2019; Awasthi et al., 2019; Kiyono et al., 2019; Kaneko et al., 2020; Omelianchuk et al., 2020).

Due to insufficient parallel training data publicly available, the GEC task is often considered as a low-resource sequence generation task (Junczys-Dowmunt et al., 2018). In order to address this issue, many data augmentation methods have been proposed for generating synthetic pseudo training data to improve the performance of GEC. Some works focus on the edit-based approach. For example, Zhao et al. (2019) use four basic edit operations for error introduction, which are random deletion, random insertion, random substitution and random shuffling. Takahashi et al. (2020) use more fine-grained noising strategies by replacing the words in a sentence with those of the same type randomly chosen from the dictionary. Some works focus on the back-translation-based method, which attempt to train a sequence-to-sequence model to translate correct sentences into wrong ones. For example, Xie et al. (2018) generate synthetic parallel data by first training a noising model with the reversed GEC data, and then using the trained noising model to translate correct sentences into error-contained ones. Some other studies utilize a large number of edit records from Wikipedia in different periods of time for data augmentation. For example, Lichtarge et al. (2019) extract source and target sentence pairs from the Wikipedia edit histories and treate them as real-world errors.

Although previous data augmentation studies have achieved good results, there are still two

Original Sentence: 0 This 1 are 2 gramamtical 3 sentence 4 .

A1: 1 2 ||| R:VERB:SVA ||| is ||| REQUIRED ||| -NONE- ||| 0
A2: 2 2 ||| M:DET ||| a ||| REQUIRED ||| -NONE- ||| 0
A3: 2 3 ||| R:SPELL ||| grammatical ||| REQUIRED ||| -NONE- ||| 0

Correct Sentence: This is a grammatical sentence.

Extracted Real Error Patterns:
are → is ; null → a ; gramamtical → grammatical

Figure 1: An example of real error patterns extracted from existing training data with annotations in M2 format.

Proceedings of the 25th Conference on Computational Natural Language Learning (CoNLL), pages 223–233
November 10–11, 2021.

points that can be improved: (1) We find that most of the existing methods introduce errors through random strategies, such as random deletion of words and random replacement of words, similar to the work of Zhao et al. (2019) and Awasthi et al. (2019). This random noise introduction strategy will generate unreal and low-quality synthetic errors which could be further propagated into the GEC model and harm its performance. However, there are a large number of real error patterns in the manually annotated training data. As shown in Figure 1, there are three real error patterns existed in the annotated sentence *"This are gramamtical sentence."*, which are *"are"* → *"is"*, *"null"* → *"a"* and *"gramamtical"* → *"grammatical"*. We argue that these real error patterns can be incorporated into data augmentation to effectively generate more real and high-quality synthetic data, which is less explicitly explored by previous studies. (2) Few previous studies have fully considered the linguistic knowledge in data augmentation, which can also be incorporated to improve the representativeness and diversity of the generated synthetic data.

To this end, we propose a novel data augmentation method for the GEC task, which consists of four types of noising schemes, namely the real error pattern based noising scheme, the synonym noising scheme, the inflection noising scheme and the functional word noising scheme. The real error pattern based noising scheme is designed for introducing the real errors made by English learners into clean text to generate high-quality synthetic data. The other three schemes are designed for incorporating linguistic knowledge to generate more representative and more diverse synthetic data. These generated synthetic data can be used as pseudo training data for the GEC task. The main contributions of our work are as followed:

(1) We propose a real error pattern based noising scheme for introducing the real errors made by learners into clean text to effectively generate more real and high-quality synthetic data. To the best of our knowledge, this is the first work that introduces real error patterns extracted from the manually annotated training data into the GEC task.

(2) We also propose three novel noising schemes that fully incorporate linguistic knowledge into data augmentation. We will demonstrate the effectiveness of the linguistic noising schemes in the GEC task.

2 Our Method

In this section, we first give the task definition of data augmentation in section 2.1 and then describe the overview architecture of our method in section 2.2. Finally, we will describe our proposed noising schemes that incorporate real error patterns and linguistic knowledge in detail in section 2.3.

2.1 Task Definition

Given a correct sentence $X = (x_1, x_2, \cdots, x_m)$ where x_i is the i^{th} word in X and the set of noising schemes $F = \{f_1, f_2, \cdots, f_k\}$ where f_j is one of the noising schemes. For each word x_i randomly selected from X, we randomly select a noising scheme f_j from F with equal probability to introduce noise into the word x_i and produce the noised word $\tilde{x}_i$, as in Equation 1:

$$\tilde{x}_i = f_j(x_i) \tag{1}$$

After the noising operation, we can obtain a generated noised sentence $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_m)$ in which each word $\tilde{x}_i$ has been noised from the word x_i through different noising schemes randomly selected from F. Data augmentation is to combine $\tilde{X}$ and X to construct a parallel sentence pair $(\tilde{X}, X)$ as pseudo training data. Performing the described noising process on an unlabeled clean text corpus, we can obtain a considerable number of pseudo sentence pairs as supplement for the original GEC training data.

2.2 Overview Architecture of Our Method

As shown in Figure 2, our method consists of three components: data augmentation, model pre-training and model fine-tuning. For the component of data augmentation, given an unlabeled clean text corpus $D_{clean} = \{X_1, X_2, \cdots, X_n\}$ where X_i is a correct sentence, we first use our proposed noising schemes to introduce errors into D_{clean} and get a noised pseudo corpus $D_{noised} = \{\tilde{X}_1, \tilde{X}_2, \cdots, \tilde{X}_n\}$ where each sentence $\tilde{X}_i$ is noised from the sentence X_i through the data augmentation process described in section 2.1. We then combine D_{noised} and D_{clean} to construct the pseudo training data, which is denoted as $D_{pseudo} = \{(\tilde{X}_1, X_1), (\tilde{X}_2, X_2), \cdots, (\tilde{X}_n, X_n)\}$. For the component of pre-training, we use the generated pseudo training data D_{pseudo} to pre-train a coarse GEC model. After that, we finetune the pre-trained coarse model with the annotated GEC train-

ing data and obtain the final GEC model, which is finally used to generate results on testing data.

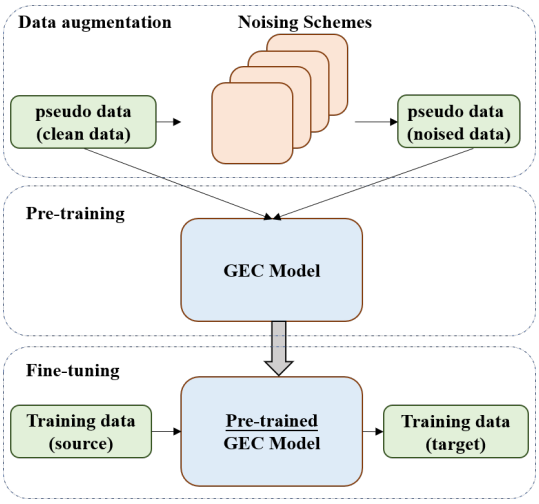

Figure 2: Overview architecture of our proposed method. Our method consists of three components, which are data augmentation using our proposed noising schemes, pre-training the model using the generated noised pseudo data and fine-tuning the final model using the original labeled training data.

2.3 Proposed Noising Schemes

As mentioned earlier, the real error patterns extracted from the existing annotated training data and some linguistic knowledge can be effectively incorporated into the data augmentation strategies. In this section, we describe our proposed noising schemes in detail.

2.3.1 Real Error Pattern based Scheme

In practice, the training data used in the task of GEC are usually real-world writings created by English learners, such as the FCE (First Certificate in English) [1] dataset and the NUCLE (NUS Corpus of Learner English) [2] dataset. The FCE dataset contains exam scripts written by candidates of the Cambridge ESOL First Certificate in English (FCE) examination and the NUCLE dataset consists of essays written by students at the National University of Singapore on a wide range of topics. In these datasets, experts annotate the grammatical errors in each sentence with detailed information, such as error position, error type and error's correct form. Most of the current GEC datasets provide M2 format files (Dahlmeier and Ng, 2012) to store these annotations.

As shown in Figure 1, the original incorrect sentence *"This are gramamtical sentence."* is annotated in M2 format. There are three grammatical errors in the sentence which are annotated as A1, A2 and A3. From the annotations, we can extract the real error patterns and reverse them as our data augmentation scheme, which is in the form of $p = \{\text{word}_{\text{correct}} : \text{word}_{\text{wrong}}\}$. Take A1 as an example, the word *"are"* has a `VERB:SVA` (subject-verb agreement) error and its correct form is the word *"is"*. Then we can construct a noising scheme {*"is"*: *"are"* }, which means that we substitute the correct word *"is"* with the wrong word *"are"* for noise introduction.

Using this method, we can extract all of the real error patterns in the GEC datasets from the M2 files and construct our real error pattern based noising scheme, which are denoted as $P = \{p_1, p_2, \cdots, p_n\}$. By using our proposed real error pattern based noising scheme, we can generate more real and high-quality synthetic data as pseudo training data. It should be noted that a correct word may correspond to different erroneous forms in different contexts. For example, the erroneous form of the word *"is"* could either be the word *"are"* in the sentence *"This **are** gramamtical sentence."* or be the word *"was"* in the sentence *"He **was** singing now."*. Hence, the same correct word may correspond to different wrong words in those noising schemes p_i of P.

2.3.2 Linguistic Knowledge based Scheme

It is generally believed that writing is a representation of language learning by learners. When learners fail to learn certain language knowledge well, they are likely to make grammatical errors related to the knowledge. That is, the linguistic knowledge is actually associated with the GEC data. To this end, we argue that the linguistic knowledge that the learners need to learn and master can be effectively integrated into the noising strategy.

A variety of linguistic knowledge can be incorporated into GEC data augmentation. Based on the error types of ERRANT [3] (Bryant et al., 2017) and the statistics on the GEC datasets, we propose the synonym noising scheme, the inflection noising scheme and the functional word noising scheme for incorporating linguistic knowledge into data augmentation.

[1] https://ilexir.co.uk/datasets/index.html
[2] https://www.comp.nus.edu.sg/%7Enlp/corpora.html
[3] A toolkit for automatically annotating parallel English sentences with error type information.

Specifically, we first use ERRANT to label the error type of each grammatical error in the GEC datasets. Then, we analyze the characteristics and linguistic patterns of each error type. We find that certain kinds of error types are similar and can be grouped into a category, which is shown in Table 1. For example, noun, verb, adverb, adjective errors are about synonym misuse. Form errors, inflection errors, noun number errors, subject-verb agreement errors and morphology errors are related to inflection. Prepositions, determiners errors and so on are functional word errors. And there remain some error types that are individual and is not easy to be grouped together (**Other Error Types** in Table 1). We select the synonym, inflection and functional word categories as our noising schemes, as they cover most of the error types (18 among 25).

Synonym
NOUN, VERB, ADV, ADJ
Inflection
VERB:FORM, ADJ:FORM, NOUN:INFL, VERB:INFL, NOUN:NUM, VERB:SVA, MORPH
Functional Word
PREP, DET, PRONOUN, CONJ, PART, CONTR
Other Error Types
OTHER, WO, ORTH, SPELL, NOUN:POSS, PUNCT, VERB:TENSE

Table 1: Error type analysis. We find that certain kinds of error types are similar and can be grouped into a category. Error types that are individual and is not easy to be grouped together are put in **Other Error Types** in the table.

Synonym Noising Scheme. We analyze various error patterns in the GEC datasets according to error types and find that certain types of grammatical errors are closely related to synonyms, such as noun errors, verb errors, adjective errors, and adverb errors. We list some examples related to the synonym error according to different error types in Table 2. As shown in Table 2, the misused word *"moment"*, *"told"*, *"big"* and *"sincerely"* are all synonyms of the correct word *"time"*, *"said"*, *"wide"* and *"faithfully"* respectively. We believe that these types of errors are caused by the learners' confusion over the use of synonyms.

Inspired by this, we propose the synonym noising scheme to mimic such type of errors. When using it to introduce noise into the word x_i, we first generate the synonym list of x_i. Then we randomly select a word from the synonym list as the noised version $\tilde{x}_i$.

Error Type	Misused Word	Correct Word
Noun	*"moment"*	*"time"*
Verb	*"told"*	*"said"*
Adjective	*"big"*	*"wide"*
Adverb	*"sincerely"*	*"faithfully"*

Table 2: Examples of synonym-related errors according to error types. Words in the "Misused Word" column are mistaken for the words in the "Correct Word" column.

Sometimes a word may be substituted with its synonym without changing the basic meanings of the sentence. At that time the noised sentence plays a role in diversifying the training corpus.

Inflection Noising Scheme. Inflection is one of the characteristics of English and other inflection languages such as French and German. Noun declension and verb conjugation are two types of inflections which are mainly manifested in the changes of word suffices in English such as *"-s"* in noun number changing and*"-ed"* in verb time changing. For example, the noun *"book"* may become the plural form *"books"*, which belongs to the noun number declension. The verb *"is"* may become *"are"*, which is the conjugation of verb number.

English contains a lot of inflections, which require learners to learn and master in the process of language learning. To this end, we use inflection as one of the linguistic knowledge to be introduced as noises into the correct text. For a word x_i, We first obtain its inflection word list and randomly select one of these possible inflection forms as the noise introduction result $\tilde{x}_i$.

Functional Word Noising Scheme. Functional words such as prepositions, pronouns, and determiners refer to words that achieve certain functions in a sentence. Statistics [4] show that about 26% of grammatical errors in the GEC datasets belong to functional word errors, which takes up a large proportion of the errors made by English learners. Therefore, we take functional word replacement as one of the noising schemes to generate functional word misuse errors, such as the misuse of word

[4]We made statistics on grammatical error types on FCE, NUCLE and Lang-8 datasets.

Type	Examples
Contraction	*"'s"*, *"'m"*, *"'t"*, *"'ve"*, *"'ll"*, *"'d"*
Determiner	*"a"*, *"all"*, *"another"*, *"both"*
Particle	*"away"*, *"at"*, *"back"*, *"by"*
Preposition	*"aboard"*, *"above"*, *"across"*
Pronoun	*"I"*, *"me"*, *"we"*, *"us"*, *"you"*
Conjunction	*"after"*, *"although"*, *"because"*

Table 3: Word lists for functional words. Words in the same list belong to the same type.

"at" as word *"in"*.

We create a word list for each type of the functional words. Each word list is a functional word collection of a specific type. For example, the preposition word list contains words such as *"in"*, *"at"*, *"on"*, *"from"*, *"to"*, etc. We created 6 word lists in total for prepositions, particles, pronouns, conjunctions, contractions, and determiners respectively. Examples of each word list is listed in Table 3.

For a word x_i that needs to be noised, we look for it in the 6 word lists. If it exists in a certain word list, a word is randomly selected from the list as the noised version $\tilde{x}_i$.

3 Experiments

3.1 Datasets

We use the FCE dataset (Yannakoudakis et al., 2011) [5], National University of Singapore Corpus of Learner English (NUCLE) (Dahlmeier et al., 2013) [6] and Lang-8 Corpus of Learner English (Lang-8) (Mizumoto et al., 2011) [7] as training data and use CoNLL-2013 (Ng et al., 2013) [8] as development data. We use the CoNLL-2014 (Ng et al., 2014) [9] dataset as test data, and report the precision, recall and $F_{0.5}$ with the MaxMatch (M^2) scorer (Dahlmeier and Ng, 2012). Following previous works, we exclude error-free sentences from the Lang-8 corpus. The training set contains roughly 1.2M sentence pairs, and the validation set consists of about 1.4K pairs.

We use the One Billion Word Benchmark (OBC) (Chelba et al., 2013) [10] as the seed corpus for data augmentation. It is a large sentence-level English corpus for language modeling produced from the WMT 2011 News Crawl data. The corpus contains roughly 30M sentence pairs. In our experiment, we utilize merely approximately 4.5M data of the OBC corpus. Since the error-free Lang-8 sentences are excluded from the training data, we also use them as part of the data for augmentation, which contains approximately 0.5M sentences. Therefore, the total number of sentences we use for data augmentation is merely 5M. The summary of each dataset used in our experiment is presented in Table 4.

Dataset	#Pairs	Split
FCE-train	32,073	Train
NUCLE	57,119	Train
Lang-8	1,097,274	Train
CoNLL-2013	1,381	Valid
CoNLL-2014	1,312	Test
OBC (partial)	4.5M	Pretrain

Table 4: Summary of each dataset. #Pairs indicates the number of sentence pairs in each dataset.

3.2 Training Strategy

We use a two-stage of training strategy in this paper, which is the pre-training stage and the fine-tuning stage.

In the pre-training stage, we first roughly pre-train a coarse GEC model with the pseudo data generated through our proposed data augmentation method. In the fine-tuning stage, we then finetune the pre-trained coarse GEC model with the annotated GEC training data (i.e., FCE, NUCLE and Lang-8) to obtain the final GEC model.

3.3 Settings

We use the Transformer-copy model (Zhao et al., 2019) [11] as our sequence-to-sequence architecture, which is a variant of the Transformer (Vaswani et al., 2017). The encoder of the model has 6 layers, with a self-attention module of 8 attention heads and a feed-forward module of 4,096 dimensions in each layer. Each module is followed by a residual connection and batch normalization. The decoder of the model also has 6 layers, and each layer is similar to the encoder layer, except that there is an additional encoder-decoder attention module with 8 attention heads between the self-attention module and the feed-forward module. The embedding and

[5] https://ilexir.co.uk/datasets/index.html
[6] https://www.comp.nus.edu.sg/ nlp/corpora.html
[7] https://sites.google.com/site/naistlang8corpora/
[8] https://www.comp.nus.edu.sg/ nlp/conll13st.html
[9] https://www.comp.nus.edu.sg/ nlp/conll14st.html
[10] https://www.statmt.org/lm-benchmark/
[11] https://github.com/yuantiku/fairseq-gec

hidden size of the model are 512. The number of copy attention heads is set to 1.

We construct a 50,000-word dictionary by extracting words from the spell-corrected Lang-8 dataset. Hyper-parameters of our method in the pre-training stage and the fine-tuning stage are shown in Table 5.

	Pre-Training	**Fine-Tuning**
max epoch	30	30
batch size	64	64
max tokens	3,000	3,000
clip norm	2	2
learning rate	2e-3	1e-3
min lr	1e-4	-
lr shrink	9.99e-3	9.5e-2
dropout	2e-1	2e-1
lr scheduler	-	triangular
max lr	-	4e-3

Table 5: Hyper-parameters of the GEC model. The second column are hyper-parameters for pretraining and the third column are for fine-tuning. "lr" indicates learning rate.

For the settings of our proposed noising schemes, we utilize WordNet (Dahlmeier and Ng, 2012) [12], a lexical database of English, to obtain the synonyms of a given word. We use `word_forms` [13], a Python package for generating all forms of an English word based on WordNet and the Xtag Project [14], to obtain the inflection list of a certain word. And we collect functional words from an English learning website [15].

3.4 Compared Models

In our experiments, we use the **Pre-training Decoder** proposed by Zhao et al. (2019), the **Denoising Auto-encoder** proposed by Zhao et al. (2019) and the **Error type w/ Data selection** proposed by Takahashi et al. (2020) as our compared models. They are all based on the Transformer-copy architecture and are pre-trained with the One Billion Word Benchmark corpus using the two-stage training strategy, which are the same with us. Therefore, we group them together in experiment results in Table 6.

In addition, we also compare our results with the recent models proposed by Lichtarge et al. (2019) and Kiyono et al. (2019). They use the vanilla Transformer without the copy mechanism as their backbone model. For pre-training, Lichtarge et al. (2019) utilize Wikipedia data, while Kiyono et al. (2019) generate pseudo data from the Gigaword corpus. Since their models and pre-training data are different from us, we put them in another group in experiment results.

3.5 Results and Analysis

The experimental results are shown in Table 6. We list three groups of results for different models. The first group is the result of the vanilla Transformer-copy model without being pre-trained with pseudo training data. The second group lists the results of the models utilized data augmentation, all of which are based on the Transformer-copy architecture and share the data settings. We denote our model as "Ours". The third group lists the results of recent works that are different from us in model architecture and pre-training data.

Firstly, we can see that the vanilla Transformer-copy model without pre-training produces 52.7 $F_{0.5}$ score, which is the worst compared with models in the second group. It shows that the pre-training stage is important for improving the performance of GEC task.

Secondly, we compare our model with previous state-of-the-art GEC models which also use the data augmentation strategy. As shown in Table 6, our model achieves 59.4 $F_{0.5}$ score which surpasses the Pre-training Decoder, the Denoising Auto-encoder and the Error type w/ Data selection model by 2.2, 0.6 and 1.8 $F_{0.5}$ score respectively, which demonstrates the effectiveness of our method. In addition, our model achieves the best recall (38.3), which further demonstrates the gains from the diversity of the error types and high quality synthetic data generate by our method.

Finally, we compare our model with two GEC models using the vanilla Transformer architecture. As shown in Table 6, our model (59.4 $F_{0.5}$) surpasses the result of Lichtarge et al. (2019) (56.8 $F_{0.5}$) by a large margin with using less amount of pseudo data. However, our result is not higher than that of Kiyono et al. (2019) which use back translation and edit operations for introducing grammatical errors. We consider that there are two possible reasons. First, they use 70M pseudo data for pre-training, which are 14 times much more than us (5M). Second, they use the big settings

[12] https://wordnet.princeton.edu/
[13] https://github.com/gutfeeling/word_forms
[14] https://www.cis.upenn.edu/%7Extag/
[15] https://7esl.com/

Model	#Pseudo Data	P	R	$F_{0.5}$
Vanilla Transformer-copy	0M (w/o pre-training)	65.8	29.3	52.7
Pre-training Decoder (Zhao et al., 2019)	30M	68.0	35.0	57.2
Denoising Auto-encoder (Zhao et al., 2019)	30M	69.0	37.0	58.8
Error type w/ Data selection (Takahashi et al., 2020)	10M	**69.1**	34.5	57.6
Real Error Patterns & Linguistic Knowledge (Ours)	**5M**	69.0	**38.3**	**59.4**
Lichtarge et al., 2019 (Lichtarge et al., 2019)	170M	65.5	37.1	56.8
Kiyono et al., 2019 (Kiyono et al., 2019)	70M	67.9	**44.1**	**61.3**

Table 6: Experimental results of the GEC models. The first section lists the result of the vanilla Transformer-copy model without pretraining. The second section lists results of our proposed method and several baselines for comparison, all of which are based on the Transformer-copy architecture. "Ours" denotes our method. We also collect results of some relevant works in the third section. #Pseudo Data indicates the amount of pretraining data used by each model. **Bold** indicates the highest score in each column.

of Transformer with 213M parameters, while our Transformer-copy is a variant of Transformer-base containing merely 97M parameters, which is much more smaller in size than theirs. We believe that further improvements of our proposed model can be observed when adapting the Transformer-copy to the big settings. In addition, it should be noted that the precision of our model (69.0) is much higher than that of Kiyono et al. (2019) (67.9), further demonstrating the high quality of data generated by our method.

As shown in Table 6, we use less augmentation data for pre-training (5M) than all other models in the second and the third group. For example, the models proposed by Zhao et al. (2019) use 30M augmentation data which is six times more than ours, and the model of Kiyono et al. (2019) uses 70M data which is fourteen times more than ours. We argue that the distribution of the augmented data generated by our proposed noising schemes are more close to that of the annotated data, which helps the model converge more quickly with less augmentation data.

4 Discussion

In order to prove the superiority of our proposed method over other noising strategies and explore different contributions of the proposed noising schemes, we conduct several ablation studies. We also give a case study to show the superiority of our method in generating high-quality and diverse pseudo data.

4.1 Ablation Studies

Our ablation studies contain two parts. In the first part (Section 4.1.1), we compare the effectiveness of our proposed noising strategy (real error patterns & linguistic knowledge) with the edit operations noising strategy as well as the mixed of our method with the edit operations. In the second part (Section 4.1.2), We investigate the contribution of each noising scheme to model performance. All results in the two ablation studies are produced by pre-training a Transformer-copy model with 200K error-free Lang-8 sentences.

4.1.1 Noising Strategies Comparison

In this part, we compare our proposed noising strategy (**Ours**) with the edit operations noising strategy (**Edits**) used by Zhao et al. (2019) and Awasthi et al. (2019). In addition, we also make a comparison between using only our method (**Ours**) and using the mixed of ours and the edit operation noising strategy (**Edits + Ours**).

Noising Strategy	Precision	Recall	$F_{0.5}$
Edits	19.6	5.0	12.4
Edits + Ours	24.7	6.3	15.6
Ours	28.9	7.1	18.0

Table 7: Noising strategies Comparison.

From Table 7, we observe that our method produces the highest $F_{0.5}$ score (18.0), beating those of the edit operations strategy (15.6) and the mixed (12.4). Specifically, our method produces both higher precision and recall, proving that the human-like errors generated by our method benefit the model performance. When our method is combined with edit operation noises, the performance drops to 15.6. It may be that in a small dataset (200K in the ablation experiments), real errors plays a more important role in performance boosting and noises

introduced by the edit operations noising strategy reduce the quality of the augmented data and result in worse model performance. If the scale of pre-training data increases, the mixed one (**Edits + Ours**) may produce better results as random noises can relief the model from overfitting.

4.1.2 Contribution of Each Noising Scheme

In this section, We investigate the contribution of each noising scheme. In each time, we remove one noising scheme while keeping the others. The results are shown in Table 8. **All** denotes the result of using all noising schemes, and the following lines are the results of removing one noising scheme while keeping the remaining.

Noising Scheme	Precision	Recall	$F_{0.5}$
All	**28.9**	**7.1**	**18.0**
- **Real Pattern**	24.1	6.6	15.7
- **Synonym**	28.1	6.9	17.4
- **Inflection**	25.0	6.6	16.1
- **Functional Word**	26.6	6.8	16.8

Table 8: Contribution of each noising scheme. The first row (**All**) is the result of using all noising schemes. And the remaining are the results of removing one of the noising schemes each.

From Table 8, we observe that using **All** noising schemes achieves the best $F_{0.5}$ of 18.0. When one of the noising schemes is removed each time, the performance drops, showing that every noising scheme is necessary for obtaining better performance. When the four noising schemes are combined together, the performance is boosted significantly, highlighting the importance of error diversity since different noising schemes generate different types of errors. We also notice that removing the real error pattern based noising scheme results in the largest performance drop with a 4.8, 0.5 and 2.3 decrease in precision, recall and $F_{0.5}$ respectively compared with **All**, proving that the real error pattern based noising scheme plays the most important role in improving the performance of the GEC model.

4.2 Case Study

In order to demonstrate the superiority of our proposed noising schemes, we show three sentences noised using our noising schemes and using the basic edit operations (random deletion, random insertion, random substitution and random shuffling) respectively (Figure 3).

As shown in Figure 3, the first line of each group is the original sentence. The second line is the sentence noised through the basic edit operations, where we use the red color to denote the noised words. The third line is the sentence noised using our noising schemes, where we use the blue color to denote the noised words. We can see that the sentences noised with our proposed noising schemes contain more errors that are similar to the real errors made by learners. However, the sentences noised by the basic edit operations contain some unreal and low-quality noised errors. For example, the second sentence augmented with the edit operations is noisy and not real, where the word *"I"* is replaced with the word *"Good"* and the word *"him"* is removed. However, the augmented sentence using our method makes more sense. For example, the contraction *"'m"* is replaced with another contraction *"'d"* and the word *"going"* is replaced with its inflection *"go"*.

We can also see that the grammatical errors generated by our methods are more diverse. Take the first sentence for example. The word *"luck"* is replaced by *"luckily"*, which mimics an inflection error. And the word *"start"* is replaced by *"begin"*, which is a synonym misuse error.

5 Related Work

Grammatical Error Correction (GEC) is the task of automatically detecting and correcting grammatical errors of texts into natural and correct forms. In recent years, different methods have been proposed to improve the performance of the GEC models. Early studies (Gamon et al., 2008; Tetreault et al., 2010; Dahlmeier and Ng, 2011; Berend et al., 2013; Rozovskaya and Roth, 2014) take GEC as a classification task and rely much on hand-crafted rules. More recently, the techniques of statistical machine translation and neural machine translation are applied to GEC and have made remarkable performance (Behera and Bhattacharyya, 2013; Junczys-Dowmunt and Grundkiewicz, 2016; Junczys-Dowmunt et al., 2018; Chollampatt and Ng, 2018; Zhao et al., 2019; Awasthi et al., 2019; Kiyono et al., 2019; Kaneko et al., 2020; Omelianchuk et al., 2020; Zhao and Wang, 2020).

Due to insufficient training data in the GEC task, many data augmentation methods have been proposed for generating synthetic pseudo training data to improve the performance of GEC. We review

Good luck on your new start! Edit operations: **.** Luck **your on** new start **like**! Ours: Good **luckily** on your new **begin**!
I'm looking forward to seeing him again through here. Edit operations: **Good** 'm looking forward to seeing again **going was words**. Ours: I**'d looked** forward to seeing him again through here.
I'm going to miss him but I really wish him the best of luck with his new life. Edit operations: I'm going to miss him but really **Kanji** wish him **job** of luck **Good** with life **new**. Ours: I'd go to miss **them** but I really wish him the best of luck with his new **liveliness**.

Figure 3: Noised sentences generated by several augmentation methods. In each section of the table, the first line is the clean sentence to noise. The second line is the sentence noised by the basic edit operations including random deletion, random insertion, random substitution and random shuffling. The third line is the sentence noised by our proposed noising schemes, which generate realistic errors.

the methods of data augmentation in detail in the next several subsections.

Edit-based approach. Some data augmentation methods focus on using edit operations for noise introduction. Zhao et al. (2019) use four basic edit operations for introducing errors. For each correct sentence, they randomly delete, insert or substitute some words with other randomly chosen words in the dictionary, or shuffle adjacent words. Takahashi et al. (2020) propose more fine-grained edit-based data augmentation method. They replace words with those of the same type, instead of substituting words with random words chosen from the dictionary.

Back translation approach. Some other method focus on back translation based augmentation method (Kasewa et al., 2018; Xie et al., 2018; Htut and Tetreault, 2019; Lichtarge et al., 2019), which attempt to train a sequence-to-sequence model to translate correct sentences into wrong ones. Kasewa et al. (2018) use a bidirectional LSTM model for converting correct sentences into wrong sentences. Xie et al. (2018) generate parallel data by first training a CNN noising model with the reversed GEC data, which turns error-free sentences into wrong ones. The trained noising model is then used for translating sentences in a clean corpus into error-contained sentences. Htut and Tetreault (2019) investigate and compare the effects of the back translation augmentation method on multiple neural models. Although the back translation augmentation method is intuitive, it requires a relative large annotated corpus for training a noising model, which is also expensive.

Wikipedia-based approach. Some studies utilize the large amount of edit records from Wikipedia in different periods of time for data augmentation (Boyd, 2018; Lichtarge et al., 2019). Boyd (2018) use Wikipedia edits to generate synthetic GEC data. They think that some Wikipedia edits contain grammatical corrections, which are similar to the corrections in the GEC dataset. Lichtarge et al. (2019) extract source and target sentence pairs from the Wikipedia edit histories and treat them as real-world errors.

6 Conclusion

In this paper, we propose a data augmentation method that incorporates real error patterns and linguistic knowledge into the GEC task, which effectively generates high-quality and diverse synthetic GEC data. Experimental results on public datasets demonstrate the effectiveness of our proposed method. Especially, our method can produce remarkable results with limited data, proving the superiority of our approach and its potential usage in low-resource scenarios.

In the future, we plan to fuse more linguistic knowledge into the GEC data augmentation to make training data more diverse. In addition, we are also willing to explore hybrid augmentation methods by combining our proposed noising schemes with other data augmentation methods such as back-translation.

Acknowledgements

This work is supported by National Natural Science Foundation of China (No. 61976062).

References

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4260–4270, Hong Kong, China. Association for Computational Linguistics.

Bibek Behera and Pushpak Bhattacharyya. 2013. Automated grammar correction using hierarchical phrase-based statistical machine translation. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 937–941, Nagoya, Japan. Asian Federation of Natural Language Processing.

Gábor Berend, Veronika Vincze, Sina Zarrieß, and Richárd Farkas. 2013. LFG-based features for noun number and article grammatical errors. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 62–67, Sofia, Bulgaria. Association for Computational Linguistics.

Adriane Boyd. 2018. Using Wikipedia edits in low resource grammatical error correction. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 79–84, Brussels, Belgium. 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 (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.

Ciprian Chelba, Tomás Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, and Phillipp Koehn. 2013. One billion word benchmark for measuring progress in statistical language modeling. *CoRR*, abs/1312.3005.

Shamil Chollampatt and Hwee Tou Ng. 2018. A multilayer convolutional encoder-decoder neural network for grammatical error correction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Daniel Dahlmeier and Hwee Tou Ng. 2011. Grammatical error correction with alternating structure optimization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 915–923, Portland, Oregon, USA. Association for Computational Linguistics.

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.

Michael Gamon, Jianfeng Gao, Chris Brockett, Alexandre Klementiev, William B. Dolan, Dmitriy Belenko, and Lucy Vanderwende. 2008. Using contextual speller techniques and language modeling for ESL error correction. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*.

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.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Phrase-based machine translation is state-of-the-art for automatic grammatical error correction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1546–1556, Austin, Texas. 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, Volume 1 (Long Papers)*, 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.

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.

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 (EMNLP-IJCNLP)*, pages 1236–1242, Hong Kong, China. 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, Volume 1 (Long and Short Papers)*, pages 3291–3301, Minneapolis, Minnesota. 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.

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, Baltimore, Maryland. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.

Alla Rozovskaya and Dan Roth. 2014. Building a state-of-the-art grammatical error correction system. *Transactions of the Association for Computational Linguistics*, 2(0):419–434.

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.

Joel Tetreault, Jennifer Foster, and Martin Chodorow. 2010. Using parse features for preposition selection and error detection. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 353–358, Uppsala, Sweden. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

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.

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, USA. Association for Computational Linguistics.

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, Volume 1 (Long and Short Papers)*, pages 156–165, Minneapolis, Minnesota. Association for Computational Linguistics.

Zewei Zhao and Houfeng Wang. 2020. Maskgec: Improving neural grammatical error correction via dynamic masking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):1226–1233.