A Unified Strategy for Multilingual Grammatical Error Correction with Pre-trained Cross-Lingual Language Model

Xin Sun[1*], **Tao Ge**[2], **Shuming Ma**[2], **Jingjing Li**[3], **Furu Wei**[2], **Houfeng Wang**[1]
[1]MOE Key Lab of Computational Linguistics, School of Computer Science, Peking University
[2]Microsoft Research Asia
[3]The Chinese University of Hong Kong
{sunx5, wanghf}@pku.edu.com {tage,shumma,fuwei}@microsoft.com lijj@cse.cuhk.edu.hk

Abstract

Synthetic data construction of Grammatical Error Correction (GEC) for non-English languages relies heavily on human-designed and language-specific rules, which produce limited error-corrected patterns. In this paper, we propose a generic and language-independent strategy for multilingual GEC, which can train a GEC system effectively for a new non-English language with only two easy-to-access resources: 1) a pre-trained cross-lingual language model (PXLM) and 2) parallel translation data between English and the language. Our approach creates diverse parallel GEC data without any language-specific operations by taking the non-autoregressive translation generated by PXLM and the gold translation as error-corrected sentence pairs. Then, we reuse PXLM to initialize the GEC model and pre-train it with the synthetic data generated by itself, which yields further improvement. We evaluate our approach on three public benchmarks of GEC in different languages. It achieves the state-of-the-art results on the NLPCC 2018 Task 2 dataset (Chinese) and obtains competitive performance on Falko-Merlin (German) and RULEC-GEC (Russian). Further analysis demonstrates that our data construction method is complementary to rule-based approaches.

1 Introduction

Grammatical Error Correction (GEC) is a monolingual text-to-text rewriting task where given a sentence containing grammatical errors, one needs to modify it to the corresponding error-free sentence. In recent years, pre-training on synthetic erroneous data and then fine-tuning on annotated sentence pairs has become a prevalent paradigm [Grundkiewicz and Junczys-Dowmunt, 2019; Lichtarge *et al.*, 2019; Zhang *et al.*, 2019; Zhou *et al.*, 2021] in English GEC, advancing the state-of-the-art results [Ge *et al.*, 2018b; Sun *et al.*, 2021; Rothe *et al.*, 2021] with various novel data synthesis approaches [Ge *et al.*, 2018a; Grundkiewicz and Junczys-Dowmunt, 2019; Kiyono *et al.*, 2019].

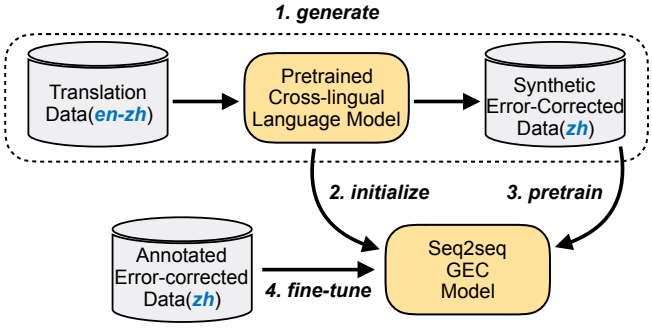


Figure 1: The overall framework of our approach. We use PXLM and a large-scale translation corpus to produce synthetic error-corrected sentence pairs. The seq2seq GEC model is initialized by PXLM and pre-trained by the synthetic data. Then we fine-tune it with language-specific annotated GEC data. **En** and **Zh** denote English and Chinese, respectively.

As GEC in other languages has drawn increasing attention [Flachs *et al.*, 2021; Rothe *et al.*, 2021], synthetic erroneous data construction has been borrowed to non-English languages for improving the results given a lack of annotated data. For instance, the rule-based approaches obtain promising results [Grundkiewicz and Junczys-Dowmunt, 2019; Náplava and Straka, 2019; Wang *et al.*, 2020a]. However, these approaches require language-specific rules and confusion sets for word replacement based on expertise to simulate diverse linguistic phenomena across multiple languages, e.g., homophones in Chinese characters and morphological variants in Russian. Moreover, rule-based approaches always produce erroneous data with limited error-corrected patterns [Zhou *et al.*, 2020].

To address the above limitations, we propose a generic strategy for training GEC systems in non-English languages. Our approach is easily adapted to new languages if only provided with two relatively easy-to-obtain resources: 1) a pre-trained cross-lingual language model (PXLM); 2) the parallel translation data between English and the language. In this paper, we choose InfoXLM [Chi *et al.*, 2020] as the PXLM in our implementation.

Our approach consists of synthetic data construction and model initialization. Since InfoXLM was pre-trained with translation language modeling objective, which requires the

*This work was done during the author's internship at MSR Asia. Contact: Tao Ge (tage@microsoft.com)

model to recover the masked tokens conditioned on the concatenation of a translation pair, it already possesses the capability of Non-Autoregressive Translation (NAT). That is, when presented with an English sentence, InfoXLM can provide a rough translation in dozens of non-English languages.

Compared with AT, NAT sacrifices translation quality due to the multimodality problem [Gu *et al.*, 2017; Ghazvininejad *et al.*, 2019]. When vanilla NAT performs independent predictions at every position, it tends to consider many possible translations of the sentence at the same time and output inconsistent results, such as token repetitions, missing or mismatch [Ran *et al.*, 2020; Du *et al.*, 2021]. Compared with predefined rules, such error-corrected patterns are more reasonable and diverse with large model capacity and dependency in sentence context. We regard the rough translation generated by InfoXLM as a source sentence and the gold translation as a corrected sentence for pre-training. To further improve the generalization ability of the seq2seq GEC model, we initialize the GEC model with InfoXLM and pre-train it with the synthetic data generated by itself.

We conduct experiments on Chinese, German and Russian GEC benchmarks. Our approach achieves the state-of-the-art results on the NLPCC 2018 Task 2 dataset (Chinese) and obtains competitive performance on Falko-Merlin (German) and RULEC-GEC (Russian). The results also demonstrate that our approach can effectively complement rule-based corruption methods.

The contributions of this paper are as follows:

- We propose a unified strategy for GEC in the non-English languages consisting of synthetic data construction and model initialization.
- We propose a novel NAT-based synthetic data construction approach, which generates diverse error-corrected data for pre-training. To the best of our knowledge, it is the first to utilize the non-autoregressive translation ability of a PXLM for GEC erroneous data construction. The generated sentence pairs perform promising results alone and also nicely complement rule-based corruption methods.
- Our approach achieves the state-of-the-art performance on the Chinese benchmark and very competitive results for German and Russian benchmarks as well.

2 Methodology

In this section, we present the unified strategy for non-English languages. At first, we briefly describe Translation Language Modeling (TLM) objective and Non-Autoregressive Translation (NAT) ability of InfoXLM. Then, we introduce two steps in our framework: NAT-based synthetic data construction and model initialization. Figure 2 shows the overview of our data construction approach.

2.1 Background: Translation Language Modeling

The basis of our data construction is the non-autoregressive translation ability of InfoXLM, owing to its Translation Language Modeling (TLM) objective during pre-training. Given a sentence $x = x_1 \cdots x_{|x|}$ in the source language (e.g., English) and the corresponding translation $y = y_1 \cdots y_{|y|}$ in another language (e.g., Chinese), the input sequence of TLM is the concatenation of these two parallel sentences $S = \langle s \rangle\ x\ \langle /s \rangle\ y\ \langle /s \rangle$ and some percentage of tokens are replaced with `[MASK]` at random. Formally, let $M = \{m_1, \cdots, m_{|M|}\}$ denote the positions of the masks:

$$m_i \sim \text{uniform}\{1, |x| + |y| + 3\} \quad \text{for } i = 1, \cdots, |M| \tag{1}$$

$$S_M = \text{replace}(S, M, \texttt{[MASK]}) \tag{2}$$

where the replace denotes the replacement operation at the certain positions. By leveraging bilingual context, the model is required to predict the original tokens with cross entropy loss. The TLM loss is computed as:

$$\mathcal{L}_{\text{TLM}} = -\sum_{S \in \mathcal{T}} \log \prod_{m \in M} p(S_m | S_{\setminus M}) \tag{3}$$

where $S_{\setminus M} = \{S_i\}_{i \notin M}$ means tokens that are not included in the M and $\mathcal{T}$ is the translation corpus.

To the extreme, we can use InfoXLM as a non-autoregressive translator. Specifically, we concatenate an English sentence x with enough placeholders (`[MASK]`) as the input. InfoXLM is capable of translating it to other languages by predicting tokens at all masked positions in parallel. Formally, $M = \{|x| + 3, \cdots, |x| + |y| + 2\}$ denotes all target tokens are replaced with `[MASK]` and the predicted translation y^* is derived by maximizing the following equation:

$$S' = \arg\max_{\boldsymbol{S_m}} \log \prod_{m \in M} p(S_m | S_{\setminus M}) \tag{4}$$

$$y^* = \text{replace}(S_M, M, S') \tag{5}$$

which infills the words with the highest probability.

In practice, following Mask-predict [Ghazvininejad *et al.*, 2019], we partially mask some percentage of target translation ($m \in [|x| + 3, |x| + |y| + 2]$) rather than all of them, which ensures the outputs are of appropriate quality.

2.2 NAT-based Data Construction

To generate diverse error-corrected sentences for GEC in a non-English language (e.g., Chinese), our approach utilizes sentence pairs of machine translation (e.g., English-Chinese parallel corpus). Our approach starts by adding noise to the target sentence and then masking sampled tokens. We feed the corrupted target sentence with the original English sentence as the input of InfoXLM. InfoXLM performs TLM predictions at every masked position. To obtain poor sentences containing grammatical errors, we randomly sample the word from the top predictions.

Specifically, given a sentence y in the target language, we select tokens for modification with a certain probability p_{noise} and perform the following operations:

Mask. Replace the token with `[MASK]` with a probability of p_{mask}.

Insert. Add a `[MASK]` after the token with a probability of p_{insert}.

Delete. Delete the token with a probability of p_{delete}.

Swap. Replace the token with its right token with a probability of p_{swap}.

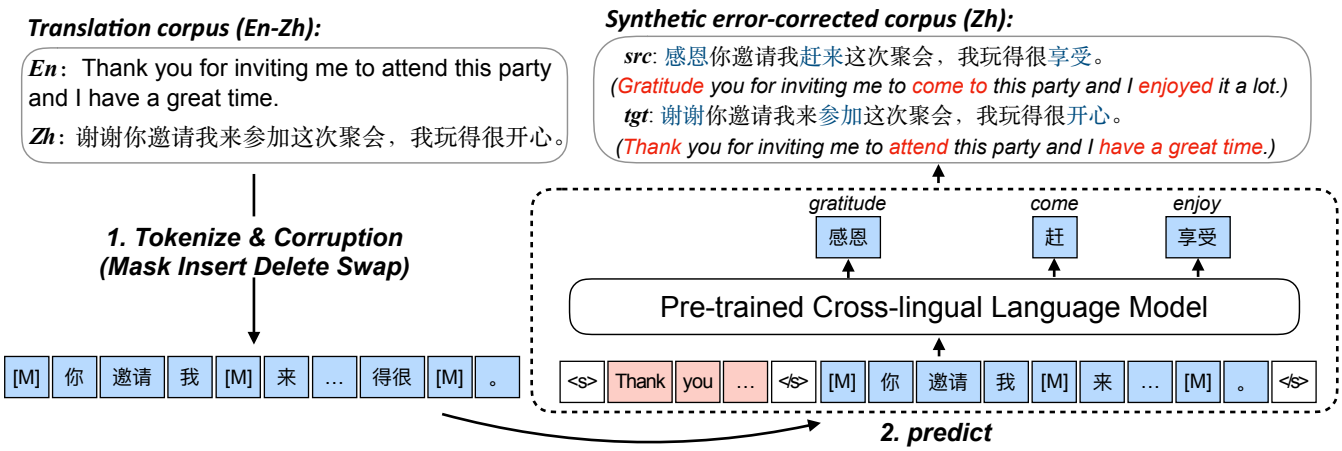


Figure 2: The overview of NAT-based data construction. Given a translation sentence pair (e.g., English-Chinese), our approach applies four operations (*Mask*, *Insert*, *Delete* and *Swap*) randomly to the non-English sentence. Then, PXLM predicts the possibility over the vocabulary at every masked position based on the concatenation of the English sentence and the corrupted sentence. Finally, we sample the predicted words and regard the recovered sequence as the source sentence containing grammatical errors and the gold non-English sentence as the corrected target sentence. `[M]` denotes `[MASK]`.

We get the noisy text $\widetilde{y} = \text{NOISE}(y)$ and the corresponding positions of the masks M. Then, we concatenate the English sentence x with the corrupted sequence $\widetilde{y}$ containing enough masked tokens as the input of InfoXLM. The predicted words are sampled for every `[MASK]` according to the output distribution:

$$y'_m \sim p(S_m|S_{\setminus M}) \quad \text{for } m \in \boldsymbol{M} \tag{6}$$

$$y^* = \text{replace}(\widetilde{y}, M, y') \tag{7}$$

where we produce erroneous sentence by replacing `[MASK]` with sampled tokens.

Our artificial corruption by four operations before TLM prediction improves the difficulty of translation. The independent assumption between target tokens brings in more errors and less fluency. The predicted words are sampled based on distribution rather than the best predictions to create more inconsistencies. It resembles the scenario where elementary language learners render a low-quality sentence when completing the cloze task. However, we only mask some percentage of target tokens and the English sentence restricts InfoXLM to recover original information, which ensures that the sampled tokens are plausible.

Since the recovered sentence contains diverse and reasonable word-level grammatical errors, we apply character-level corruption operations to add more spelling errors: 1) insert; 2) substitute; 3) delete; 4) swap-right; 5) change the casing. We call it **post edit**. Finally, we regard the gold translation as the corrected sentence and the corrupted prediction as the erroneous sentence.

2.3 Model Initialization

To further improve the generalization ability of the GEC model, we use InfoXLM to initialize the seq2seq model. We follow [Ma *et al.*, 2021] and use DeltaLM for multilingual GEC. DeltaLM is an InfoXLM-initialized encoder-decoder model trained in a self-supervised way. We continue pre-training it with synthetic data generated by our NAT-based approach.

Overall, our unified strategy exploits InfoXLM in two ways. We make use of its NAT ability to produce synthetic GEC data and its pre-trained weights to initialize our GEC model.

Dataset	Language	Train	Valid	Test
NLPCC 2018 Task 2	Chinese	1.2M	5000	2000
Falko-Merlin	German	19237	2503	2337
RULEC-GEC	Russian	4980	2500	5000

Table 1: Statistics of the benchmarks for evaluation. The numbers in the table indicate the count of sentence pairs.

3 Experiments

3.1 Data

We conduct our experiments on three GEC datasets: NLPCC 2018 Task 2 [Zhao *et al.*, 2018] in Chinese, Falko-Merlin [Boyd, 2018] in German and RULEC-GEC [Rozovskaya and Roth, 2019] in Russian. The statistics of the datasets are listed in Table 1. We use the official MaxMatch [Dahlmeier and Ng, 2012] scripts[1] to evaluate precision, recall and $F_{0.5}$.

For non-autoregressive translation generation, we use datasets of the WMT20 news translation task[2] – UN Parallel Corpus v1.0 for Chinese and Russian, the combination of Europarl v10, ParaCrawl v5.1 and Common Crawl corpus for German. We construct 10M synthetic sentence pairs in every language for pre-training and then fine-tune the GEC model on respective annotated datasets.

[1]https://github.com/nusnlp/m2scorer

[2]https://www.statmt.org/wmt20/translation-task.html

Model	NLPCC-18		
	P	R	$F_{0.5}$
YouDao	35.24	18.64	29.91
AliGM	41.00	13.75	29.36
BLCU	**47.23**	12.56	30.57
BERT-encoder	41.94	22.02	35.51
BERT-fuse	32.20	23.16	29.87
Dropout-Src	39.08	18.80	32.15
MaskGEC	44.36	22.18	36.97
- Our Implementation	41.66	25.81	37.10
[Wang *et al.*, 2020a]	39.43	22.80	34.41
Rule(10M)	44.66	26.54	39.30
Ours(10M)	44.27	26.76	39.15
- w/ DeltaLM	**45.95**	**27.94**	**40.70**
Ours(10M) + Confusion set	45.17	26.11	39.42
Ours(5M) + Rule(5M)	45.33	**27.61**	**40.17**

Table 2: Performance of systems on the NLPCC-2018 Task 2 dataset. The results of different model architectures are shown at the top group. Different training strategies are shown in the middle. The approaches with pre-training are shown at the bottom. **Rule** denotes the synthetic data generated by rule-based corruption. **Ours** denotes data generated by our approach.

3.2 Implementation Details

Unless explicitly stated, we use Transformer (base) model in fairseq[3] as our GEC model. For Chinese, we construct a character-level vocabulary consisting of 7K tokens. We apply Byte Pair Encoding to preprocess German and Russian sentences and obtain the vocabularies with size of 32K tokens, respectively. When using DeltaLM, we utilize its shared vocabulary of 250000 tokens. During pre-training for German and Russian, following [Náplava and Straka, 2019], we use source and target word dropouts and edit-weighted MLE [Junczys-Dowmunt *et al.*, 2018]. We leave the detailed hyperparameters in the supplementary notes.

3.3 Baselines

Most of the previous studies for Chinese GEC focus on model architecture or training strategy, which are orthogonal with our synthetic data construction method. For example, **YouDao** [Fu *et al.*, 2018] combines five hybrid correction models and a language model together. **AliGM** [Zhou *et al.*, 2018] combines NMT-based, SMT-based and rule-based models together. **BLCU** [Ren *et al.*, 2018] uses multi-layer convolutional seq2seq model. **BERT-encoder** [Wang *et al.*, 2020b] initializes the encoder of seq2seq model with BERT . **BERT-fuse** [Wang *et al.*, 2020b] incorporates BERT for additional features. As for training strategy, **Dropout-Src** [Junczys-Dowmunt *et al.*, 2018] sets the full embeddings of randomly selected source words to 0 during the training process. **MaskGEC** [Zhao and Wang, 2020] performs dynamic masking method by substituting the source word with a padding symbol or other word.

The most comparable approach is [Wang *et al.*, 2020a], which constructs pre-training data using the rule-based corruption method. For our approach, we implement MaskGEC during the fine-tuning stage. To make a fair comparison, we also construct synthetic data with **rule-based corruption** in the same setting as baseline. It incorporates a character-level confusion set [4] and uses *pypinyin*[5] to perform homophony replacement.

For German and Russian, the main data construction method is rule-based corruption. [Grundkiewicz and Junczys-Dowmunt, 2019] and [Náplava and Straka, 2019] build confusion sets with edit distance, word embedding or spell-checker (e.g., Aspell dictionary[6]). [Flachs *et al.*, 2021] uitlizes Unimorph which provides morphological variants of words for word replacement operations. They also incorporate WikiEdits and Lang8 as additional training resources. [Rothe *et al.*, 2021] only applies language-agnostic operations without any confusion set. They pre-train a unified seq2seq model for 101 languages and fine-tune for respective languages. [Katsumata and Komachi, 2020] proposes to directly use mBART without pre-training.

3.4 Main Results

Table 2 shows the performance of our approach and previous methods on the NLPCC-2018 Chinese benchmark. Our NAT-based synthetic data construction approach is comparable with the rule-based corruption approach. We assume that 0.15 $F_{0.5}$ descend comes from that rule-based corruption leverages many useful confusion sets. When combined with the confusion sets, our approach obtains 39.42 $F_{0.5}$ which outperforms the rule-based counterpart. If combining two data sources from the rule-based and NAT-based approaches, we obtain better performance which demonstrates two methods complement each other. Initializing the GEC model with DeltaLM achieves 40.70 $F_{0.5}$, which is the state-of-the-art result of the dataset.

Table 11 shows the performance for German and Russian datasets. In the same setting, our NAT-based synthetic approach outperforms rule-based corruption methods and most baselines with **two exceptions**. For instance, [Náplava and Straka, 2019] leverages more training strategies during fine-tuning phase such as mixing pre-training data and oversampled fine-tuning data, checkpoint averaging and so on. Although [Rothe *et al.*, 2021] obtains the best performance with the pre-trained T5-XXL (11B parameters), its base-size model lags behind ours significantly under the similar model capacity. Overall, the performance on the German and Russian datasets demonstrates the effectiveness of our unified strategy and NAT-based synthetic approach, which performs competitive results alone and also nicely complements rule-based corruption methods.

To make fair comparison with multiple synthetic construction approaches, we follow the experimental setting and model hyperparameters[7] in [Flachs *et al.*, 2021]. The results on the German dataset are shown in Table 4. Our approach significantly outperforms commonly used synthetic methods such as the rule-based approach with Unimorph, Aspell word replacement and Wikipedia edits extracted from the

[3]https://github.com/pytorch/fairseq

[4]http://nlp.ee.ncu.edu.tw/resource/csc.html

[5]https://github.com/mozillazg/python-pinyin

[6]http://aspell.net/

[7]We use the "transformer_clean_big_tpu" setting.

Model	Size	German			Russian		
	Layer, Hidden, FFN	P	R	$F_{0.5}$	P	R	$F_{0.5}$
[Grundkiewicz and Junczys-Dowmunt, 2019]	6, 512, 2048	73.0	61.0	70.24	36.3	**28.7**	34.46
[Náplava and Straka, 2019] ♣	6, 512, 2048	**78.11**	59.13	**73.40**	59.13	26.05	47.15
[Rothe *et al.*, 2021] ♠	12, 768, 2048	-	-	69.21	-	-	26.24
Rule(10M)	6, 512, 2048	73.71	59.28	70.29	49.38	23.49	40.46
Ours(10M)	6, 512, 2048	73.86	60.74	70.80	57.96	23.51	44.82
- w/ DeltaLM	12-6, 768, 2048	**75.59**	**65.19**	**73.25**	**59.31**	27.07	**47.90**
Ours(5M) + Rule(5M)	6, 512, 2048	74.31	**61.46**	71.33	**61.40**	**27.47**	**49.24**
[Flachs *et al.*, 2021]	6, 1024, 4096	-	-	69.24	-	-	44.72
[Náplava and Straka, 2019] ♣	6, 1024, 4096	**78.21**	**59.94**	**73.71**	**63.26**	**27.50**	**50.20**
[Katsumata and Komachi, 2020]	12, 1024, 4096	73.97	53.98	68.86	53.50	26.35	44.36
[Rothe *et al.*, 2021] ♠	24, 4096, 10240	-	-	**75.96**	-	-	**51.62**

Table 3: Performance of systems on German and Russian datasets. **Layer**, **Hidden** and **FFN** denote depth, embedding size and feed forward network size of Transformer. **12-6** denotes that DeltaLM-initialized model has a 12-layer encoder and a 6-layer decoder. The top and bottom group shows the results of base-scale models and large-scale models, respectively. ♣ Our re-implementation of this approach is **Rule(10M)**, whose results are inferior to **Ours(10M)**. We use synthetic data generated by their released codes and the same training strategy as ours. ♠ While [Rothe *et al.*, 2021] obtains the best performance with the pre-trained T5-XXL (11B parameters), its base-size model lags behind ours significantly under the similar model size.

Method	$F_{0.5}$
Rule(Unimorph)*	60.87
Rule(Aspell)*	63.49
Rule(Combine)*	62.55
WikiEdits*	58.00
Rule + WikiEdits*	**66.66**
Back-translate	61.37
Round-trip translation	62.91
Ours	**69.17**

Table 4: $F_{0.5}$ scores of different data construction approaches on the German dataset. For the approaches with $\star$, their results are from [Flachs *et al.*, 2021].

Method		None	Rule	Ours	Both
Error Type	Ratio	$F_{0.5}$	$F_{0.5}$	$F_{0.5}$	$F_{0.5}$
Punctuation	14.9	58.67	72.41	**73.98**	73.75
Spelling	14.0	43.89	76.73	77.64	**78.76**
Other	9.8	8.64	29.18	**35.66**	31.51
Determiner:FORM	9.7	58.43	80.62	81.39	**81.90**
Orthography	8.3	66.38	76.33	73.70	**77.72**
Adposition	5.6	28.15	52.29	**53.20**	51.71
Determiner	4.7	25.00	50.10	55.91	**57.14**
Adjective:FORM	4.0	57.14	81.44	82.29	**83.41**
Pronoun	3.9	19.44	47.87	45.81	**51.40**

Table 6: Performance of synthetic data construction approaches on top 9 error types of the German dataset.

revision history. Although back-translate is effective for English, it performs poorly with limited annotated sentence pairs to learn diverse error-corrected patterns. Round-trip translation utilizes the same translation corpus as us but achieves inferior performance since it usually produces sentences without grammatical errors.

3.5 Ablation Study

Method	P	R	$F_{0.5}$
Ours	**73.86**	60.74	**70.80**
- [MASK] replacement	71.17	55.07	67.24
- [MASK] insert	72.52	59.00	69.34
- post edit	73.00	**61.36**	70.34
- bilingual constraint	71.17	55.89	67.48
w/ an autoregressive translator	66.99	55.44	64.31

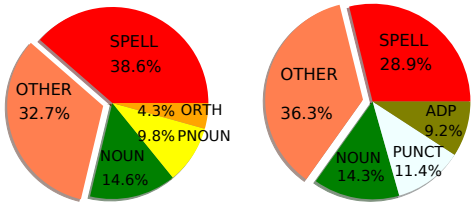
Table 5: Performance of our approach with different schemes on the German dataset. - denotes removing the component or replacing it with the rule-based operation.

We further conduct an ablation study as shown in Table 5. Overall, we find that all of these variants perform worse than the original strategy. From the last row, PXLM is much better than a regular translation model under the same setup (i.e., training data and sample strategy). Our approach can control the degree of overlap and errors, while the sentences generated by AT have few grammatical errors or little overlap with the original sentences. The removal of NAT-based replacement and bilingual constraint also results in a significant degradation, which indicates substitution with similar semantic meanings plays a crucial role in our strategy.

3.6 Error-type Analysis

We analyze the GEC performance of data construction approaches on different error types. We use the German extension [Boyd, 2018] of the automatic error annotation tool ERRANT [Bryant *et al.*, 2017] for evaluation. Table 6 shows the $F_{0.5}$ score of top 9 error types on the German dataset. We can observe that our approach improves the model in all error types significantly compared with that trained from the scratch and outperforms that with rule-based corruption in 7 out of 9 error types. For example, the largest improvement comes from the 'Other' type by 6.5 $F_{0.5}$ score, which is defined as the errors that do not fall into any other specific type, such as paraphrasing (e.g., *feel happy* $\rightarrow$ *be delighted*) [Bryant *et al.*, 2017]. Such error type is beyond pre-defined rules and hard to simulate [Zhou *et al.*, 2020].

Two exceptions are 'Orthography' and 'Pronoun'. 'Orthography' refers to the error related with case or whitespace errors (e.g., *Nexttime* $\rightarrow$ *next time*), which the specific rules



(a) Rule-based corruption. (b) Our approach.

Figure 3: Top 5 error types distribution of different synthetic data construction approaches.

are able to simulate very well. 'Pronoun' denotes the substitutes for the noun (e.g. *you* → *yourself*) which also fall into the strength of the rule-based approach with language-specific confusion set. We also observe the combination of these two sources of synthetic data yields better results, which demonstrates that they are helpful to complement each other to enrich error-corrected patterns.

To verify our explanation, we present the top 5 error types distribution of rule-based corruption and our approach in Figure 3. Our approach yields more 'Other' type errors compared with 'Spell' errors, which may account for the improvement in that category. The large ratio of 'Orthography' and 'Pronoun' errors generated by rule-based corruption is consistent with its better performance on these two types.

3.7 Case Study

Source	总之，她们的生活质量非常低。 (In short, their quality of life is very poor.)
BT	总之她们的生活质量非常不好$_1$。 (In short their quality of life is very bad$_1$.)
Rule	总，之她闷$_1$的生活效率$_2$非常低。 (In, short her boring$_1$ life's efficiency$_2$ is very poor.)
RT	简言$_1$之，他们的生活质量很$_2$低。 (In short, their quality of life is very poor.)
Ours	可以说$_1$，她们的生命$_2$素质$_3$非常弱$_4$。 (So to speak$_1$, their quality$_3$ of life$_2$ is very weak$_4$.)

Table 7: Examples of synthetic erroneous sentences. The rewritten tokens are highlighted in blue. **BT**, **Rule** and **RT** denote back-translation, rule-based corruption and round-trip translation.

To give a qualitative analysis of generated erroneous sentences, we present an example of our approach and existing synthetic methods in Table 7. We can see that back-translation tends to generate similar modifications such as token deletion and simple paraphrasing. The rule-based corruption approach is hard to simulate human writing since it directly swaps adjacent tokens and performs word replacement without consideration of the sentence context. Round-trip translation generates an error-free sentence. In contrast, our approach generates the less fluent sentence by paraphrasing the corrupted contents and maintaining the meaning of the corresponding English sentence.

4 Related Work

Pre-training a seq2seq model on synthetic data and then fine-tuning on annotated error-corrected sentence pairs is the common practice for GEC. Available datasets in non-English languages such as German [Boyd, 2018], Russian [Rozovskaya and Roth, 2019] and Czech [Náplava and Straka, 2019] only contain a lack of annotated data, which requires high-quality synthetic data construction.

Back-translation is the reverse of GEC, which takes corrected sentences as input and error sentences as output. It is popular and effective for English GEC [Kiyono *et al.*, 2019; Zhang *et al.*, 2019] but difficult to adapt to these low-resource scenarios, since it is hard to learn diverse error-corrected patterns with less annotated sentence pairs. Round-trip translation (e.g., translating German to English then back to German) [Lichtarge *et al.*, 2019] has been blamed for errors that it introduced are relatively clean.

The most effective construction method for non-English languages is rule-based corruption [Náplava and Straka, 2019; Grundkiewicz and Junczys-Dowmunt, 2019]. Most of them rely on word substitutions with ASpell or language-specific confusion sets. It requires well-designed rules to simulate diverse linguistic phenomena across different languages. [Rothe *et al.*, 2021] only performs language-agnostic operations without any confusion set to construct corrupted sentences but achieves inferior performance with moderate model size. Wikipedia edits extracted from the revision history of each page are also useful GEC pre-training resources [Boyd, 2018; Flachs *et al.*, 2021]. Most studies for Chinese GEC focus on model architecture [Fu *et al.*, 2018; Zhou *et al.*, 2018; Ren *et al.*, 2018] and training strategy [Zhao and Wang, 2020], which are orthogonal with our approach.

The most similar approach to ours is [Zhou *et al.*, 2020] which trains two autoregressive translation models with poor and good qualities, respectively. With the same sentence in the source language, they regard two translations of two models as error-corrected sentence pairs. In comparison, our approach directly utilizes the non-autoregressive translation ability of the PXLM without training translators additionally, which is easier to adapt to new languages. Utilizing a pre-trained language model to propose candidate words for replacement and insertion has also been applied to lexical substitution [Zhou *et al.*, 2019], text generation [Li *et al.*, 2020], etc. By contrast, we adopt bilingual constraints to avoid generating candidate words that are inconsistent with the original meaning.

Leveraging pre-trained language models in GEC seq2seq models has been extensively explored [Katsumata and Komachi, 2020; Wang *et al.*, 2020b; Kaneko *et al.*, 2020]. We initialize the model with DeltaLM [Ma *et al.*, 2021], which adjusts the InfoXLM-initialized encoder-decoder model to generation mode by self-supervised pre-training.

5 Conclusion and Future Work

We propose a unified and generic strategy for training GEC systems in non-English languages given a PXLM and the parallel translation data. Our approach obtains state-of-the-art results on the NLPCC 2018 Task 2 dataset (Chinese) and competitive results on German and Russian benchmarks. The synthetic sentence pairs also complement rule-based corrup-

tion to yield further improvement. Compared with a regular translator, NAT by the PXLM can control the degree of overlap between the generated sentence and the original sentence. The bilingual constraint also ensures that the sampled tokens will not deviate from the original meaning, which plays an important role in our strategy.

We plan to investigate whether utilizing back-translated English sentences rather than gold English sentences leads to similar performance, which could get rid of quantitative restriction by the size of the translation corpus to generate an unlimited number of error-corrected sentence pairs.

A Hyper-parameters

The parameters for NAT-based data construction and post edit are presented in Table 8 and Table 9. The hyper-parameters of training the Transformer on NLPCC 2018 Task 2 (Chinese) are listed in Table 10. The hyper-parameters for German and Russian are shown in Table 11. For Russian, we use rule-based corruption without any language-specific operation and confusion set with 50% probability to assist in synthetic data construction.

Language	p_{noise}	mask	insert	delete	swap
Chinese	0.5	0.7	0.1	0.1	0.1
German	0.3	0.65	0.15	0.15	0.05
Russian	0.15	0.65	0.15	0.15	0.05

Table 8: Parameters for NAT-based data construction.

Language	p_{noise}	substitute	insert	delete	swap	recase
Chinese	0.05	0.3	0.2	0.3	0.2	0
German	0.02	0.25	0.25	0.2	0.2	0.1
Russian	0.02	0.25	0.25	0.2	0.2	0.1

Table 9: Parameters for post edit.

Configurations	Values
Pre-training	
Model Architecture	Transformer (base)
Devices	8 Nvidia V100 GPU
Max tokens per GPU	5120
Update Frequency	8
Optimizer	Adam (β_1=0.9, β_2=0.98, ϵ=1×10^{-8})
Learning rate	7×10^{-4}
Learning rate scheduler	polynomial decay
Warmup	8000
Weight decay	0.0
Loss Function	label smoothed cross entropy (label-smoothing=0.1)
Dropout	0.3
Fine-tuning	
Devices	4 Nvidia V100 GPU
Training Strategy	MaskGEC [Zhao and Wang, 2020]
Update Frequency	[2, 4]
Learning rate	[5×10^{-4}, 7×10^{-4}]
Warmup	4000

Table 10: Hyper-parameters values of pre-training and fine-tuning on NLPCC 2018 Task 2 (Chinese).

Configurations	Values
Pre-training	
Model Architecture	Transformer (base)
Devices	8 Nvidia V100 GPU
Max tokens per GPU	5120
Update Frequency	8
Optimizer	Adam (β_1=0.9, β_2=0.98, ϵ=1×10^{-8})
Learning rate	[5×10^{-4}, 7×10^{-4}]
Learning rate scheduler	polynomial decay
Warmup	8000
Weight decay	0.0
Loss Function	label smoothed cross entropy (label-smoothing=0.1)
Dropout	[0.1, 0.2]
Source Dropout	0.2
Target Dropout	0.1
Edit-weighted MLE	3
Fine-tuning	
Devices	1 Nvidia V100 GPU
Update Frequency	[2, 4]
Learning rate	[3×10^{-4}, 5×10^{-4}, 7×10^{-4}]
Dropout	[0.1, 0.2, 0.3]
Warmup	2000

Table 11: Hyper-parameters values of pre-training and fine-tuning on Falko-Merlin (German) and RULEC-GEC (Russian).

Acknowledgments

We thank all the reviewers for their valuable comments to improve our paper. The work is supported by National Natural Science Foundation of China under Grant No.62036001 and PKU-Baidu Fund (No.2020BD021). The corresponding author of this paper is Houfeng Wang.

References

[Boyd, 2018] Adriane Boyd. 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*, 2018.

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

[Chi *et al.*, 2020] Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. *arXiv preprint arXiv:2007.07834*, 2020.

[Dahlmeier and Ng, 2012] Daniel Dahlmeier and Hwee Tou Ng. Better evaluation for grammatical error correction. In *Proc. of ACL*, 2012.

[Du *et al.*, 2021] Cunxiao Du, Zhaopeng Tu, and Jing Jiang. Order-agnostic cross entropy for non-autoregressive machine translation. *arXiv preprint arXiv:2106.05093*, 2021.

[Flachs *et al.*, 2021] Simon Flachs, Felix Stahlberg, and Shankar Kumar. Data strategies for low-resource grammatical error correction. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, 2021.

[Fu *et al.*, 2018] Kai Fu, Jin Huang, and Yitao Duan. Youdao's winning solution to the nlpcc-2018 task 2 challenge: a neural machine translation approach to chinese grammatical error correction. In *Proc. of NLPCC*, 2018.

[Ge *et al.*, 2018a] Tao Ge, Furu Wei, and Ming Zhou. Fluency boost learning and inference for neural grammatical error correction. In *Proc. of ACL*, 2018.

[Ge *et al.*, 2018b] Tao Ge, Furu Wei, and Ming Zhou. Reaching human-level performance in automatic grammatical error correction: An empirical study. *arXiv preprint arXiv:1807.01270*, 2018.

[Ghazvininejad *et al.*, 2019] Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In *Proc. of EMNLP*, 2019.

[Grundkiewicz and Junczys-Dowmunt, 2019] Roman Grundkiewicz and Marcin Junczys-Dowmunt. Minimally-augmented grammatical error correction. *W-NUT 2019*, 2019.

[Gu *et al.*, 2017] Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*, 2017.

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

[Kaneko *et al.*, 2020] Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *Proc. of ACL*, 2020.

[Katsumata and Komachi, 2020] Satoru Katsumata and Mamoru Komachi. Stronger baselines for grammatical error correction using a pretrained encoder-decoder model. In *Proc. of ACL*, 2020.

[Kiyono *et al.*, 2019] Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. An empirical study of incorporating pseudo data into grammatical error correction. In *Proc. of EMNLP*, 2019.

[Li *et al.*, 2020] Jingjing Li, Zichao Li, Lili Mou, Xin Jiang, Michael Lyu, and Irwin King. Unsupervised text generation by learning from search. In *Proc. of NeurIPS*, 2020.

[Lichtarge *et al.*, 2019] Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. Corpora generation for grammatical error correction. In *Proc. of ACL*, 2019.

[Ma *et al.*, 2021] Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, Alexandre Muzio, Saksham Singhal, Hany Hassan Awadalla, Xia Song, and Furu Wei. Deltalm: Encoder-decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders. *arXiv preprint arXiv:2106.13736*, 2021.

[Náplava and Straka, 2019] Jakub Náplava and Milan Straka. Grammatical error correction in low-resource scenarios. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, 2019.

[Ran *et al.*, 2020] Qiu Ran, Yankai Lin, Peng Li, and Jie Zhou. Learning to recover from multi-modality errors for non-autoregressive neural machine translation. In *Proc. of ACL*, 2020.

[Ren *et al.*, 2018] Hongkai Ren, Liner Yang, and Endong Xun. A sequence to sequence learning for chinese grammatical error correction. In *Proc. of NLPCC*, 2018.

[Rothe *et al.*, 2021] Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. A simple recipe for multilingual grammatical error correction. *arXiv preprint arXiv:2106.03830*, 2021.

[Rozovskaya and Roth, 2019] Alla Rozovskaya and Dan Roth. Grammar error correction in morphologically rich languages: The case of russian. *Transactions of the Association for Computational Linguistics*, 2019.

[Sun *et al.*, 2021] Xin Sun, Tao Ge, Furu Wei, and Houfeng Wang. Instantaneous grammatical error correction with shallow aggressive decoding. In *Proc. of ACL*, 2021.

[Wang *et al.*, 2020a] Chencheng Wang, Liner Yang, Yingying Wang, Yongping Du, and Erhong Yang. Chinese grammatical error correction method based on transformer enhanced architecture. *Journal of Chinese Information Processing*, 2020.

[Wang *et al.*, 2020b] Hongfei Wang, Michiki Kurosawa, Satoru Katsumata, and Mamoru Komachi. Chinese grammatical correction using bert-based pre-trained model. *arXiv preprint arXiv:2011.02093*, 2020.

[Zhang *et al.*, 2019] Yi Zhang, Tao Ge, Furu Wei, Ming Zhou, and Xu Sun. Sequence-to-sequence pre-training with data augmentation for sentence rewriting. *arXiv preprint arXiv:1909.06002*, 2019.

[Zhao and Wang, 2020] Zewei Zhao and Houfeng Wang. Maskgec: Improving neural grammatical error correction via dynamic masking. In *Proc. of AAAI*, 2020.

[Zhao *et al.*, 2018] Yuanyuan Zhao, Nan Jiang, Weiwei Sun, and Xiaojun Wan. Overview of the nlpcc 2018 shared task: Grammatical error correction. In *Proc. of NLPCC*, 2018.

[Zhou *et al.*, 2018] Junpei Zhou, Chen Li, Hengyou Liu, Zuyi Bao, Guangwei Xu, and Linlin Li. Chinese grammatical error correction using statistical and neural models. In *Proc. of NLPCC*, 2018.

[Zhou *et al.*, 2019] Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. Bert-based lexical substitution. In *Proc. of ACL*, 2019.

[Zhou *et al.*, 2020] Wangchunshu Zhou, Tao Ge, Chang Mu, Ke Xu, Furu Wei, and Ming Zhou. Improving grammatical error correction with machine translation pairs. In *Proc. of EMNLP*, 2020.

[Zhou *et al.*, 2021] Wangchunshu Zhou, Tao Ge, Canwen Xu, Ke Xu, and Furu Wei. Improving sequence-to-sequence pre-training via sequence span rewriting. In *Proc. of EMNLP*, 2021.