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

Xin Sun¹*, Tao Ge², Shuming Ma², Jingjing Li³, Furu Wei², Houfeng Wang¹
¹MOE Key Lab of Computational Linguistics, School of Computer Science, Peking University
²Microsoft Research Asia
³The 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].

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...
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 NLPC 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 \( S = \langle s \rangle x \langle /s \rangle y \langle /s \rangle \) 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:

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

\[
S_M = \text{replace}(S, M, [\text{MASK}])
\]

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:

\[
L_{\text{TLM}} = - \sum_{S \in \mathcal{T}} \log \prod_{m \in M} p(S_m | S_{\setminus M})
\]

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 marked positions in parallel. Formally, \( M = \{\mid |x| + 3, \cdots, |x| + |y| + 2\} \) denotes all target tokens are replaced with [MASK] and the predicted translation \( y^* \) is derived by maximizing the following equation:

\[
S' = \arg \max_{S_m} \log \prod_{m \in M} p(S_m | S_{\setminus M})
\]

\[
y^* = \text{replace}(S_M, M, S')
\]

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 (\( m \in \{\mid |x| + 3, |x| + |y| + 2\} \) 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 \( p_{\text{noise}} \) and perform the following operations:

- **Mask.** Replace the token with [MASK] with a probability of \( p_{\text{mask}} \).
- **Insert.** Add a [MASK] after the token with a probability of \( p_{\text{insert}} \).
- **Delete.** Delete the token with a probability of \( p_{\text{delete}} \).
- **Swap.** Replace the token with its right token with a probability of \( p_{\text{swap}} \).
We get the noisy text \( \tilde{y} = \text{NOISE}(y) \) and the corresponding positions of the masks \( M \). Then, we concatenate the English sentence \( x \) with the corrupted sequence \( \tilde{y} \) containing enough masked tokens as the input of InfoXLM. The predicted words are sampled for every \([\text{MASK}]\) according to the output distribution:

\[
y^*_m \sim p(S_m | S, M) \quad \text{for } m \in M \\
y^* = \text{replace}(\tilde{y}, M, y')
\]

where we produce erroneous sentence by replacing \([\text{MASK}]\) 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 masked 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.

### 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
3.2 Implementation Details

Unless explicitly stated, we use Transformer (base) model in fairseq\(^1\) 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.

Table 2 shows the performance for German and Russian datasets. 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.

3.3 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 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 pinyin\(^5\) 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] utilizes 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 NLPPC-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

---

\(^1\)https://github.com/pytorch/fairseq

\(^2\)http://aspell.net/

\(^3\)https://github.com/mozillazg/python-pinyin

\(^4\)http://nlp.ee.ncu.edu.tw/resource/csc.html

\(^5\)http://pypinyin

\(^6\)We use the “transformer_clean_big_tpu” setting.
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.

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

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

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...
Our approach.

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 to 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 PXLW 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 PXLW and the parallel translation data. Our approach obtains state-of-the-art results on the NLPC 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.

<table>
<thead>
<tr>
<th>Language</th>
<th>p_{noise}</th>
<th>mask</th>
<th>insert</th>
<th>delete</th>
<th>swap</th>
<th>recase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chinese</td>
<td>0.3</td>
<td>0.7</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>German</td>
<td>0.3</td>
<td>0.65</td>
<td>0.15</td>
<td>0.15</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>Russian</td>
<td>0.15</td>
<td>0.65</td>
<td>0.15</td>
<td>0.15</td>
<td>0.05</td>
<td></td>
</tr>
</tbody>
</table>

Table 8: Parameters for NAT-based data construction.

<table>
<thead>
<tr>
<th>Configurations</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Architecture</td>
<td>Transformer (base)</td>
</tr>
<tr>
<td>Devices</td>
<td>8 Nvidia V100 GPU</td>
</tr>
<tr>
<td>Max tokens per GPU</td>
<td>5120</td>
</tr>
<tr>
<td>Update Frequency</td>
<td>8</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adam (β₁=0.9, β₂=0.98, ε=1×10⁻⁴)</td>
</tr>
<tr>
<td>Learning rate</td>
<td>7×10⁻⁴</td>
</tr>
<tr>
<td>Learning rate scheduler</td>
<td>polynomial decay</td>
</tr>
<tr>
<td>Warmup</td>
<td>8000</td>
</tr>
<tr>
<td>Weight decay</td>
<td>0.0</td>
</tr>
<tr>
<td>Loss Function</td>
<td>label smoothed cross entropy</td>
</tr>
<tr>
<td>Dropout</td>
<td>[0.0, 0.2, 0.3]</td>
</tr>
</tbody>
</table>

Table 10: 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


