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 EECS, 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 pretrained 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 pretrain 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, pretraining 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] in English GEC, advancing the state-of-the-art results with various novel data synthesis approaches [Ge et al., 2018; 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 [Lample and Conneau, 2019], 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 mispredictions at every position, it tends to consider many positions. By leveraging bilingual context, the model is required to predict the original tokens with cross entropy loss. The TLM loss is computed as:

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

where $S_{\setminus M} = \{S_i \mid i \notin M\}$ 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, $M = \{x[1] + 3, \ldots, x[|x| + |y| + 2]\}$ denotes all target tokens are replaced with [MASK] and the predicted translation $y^*$ is derived by maximizing the following equation:

$$
S' = \text{arg max}_{S_m} \log \prod_{m \in M} p(S_m | S_M \setminus m)
$$

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 ($m \in \{|x[1] + 3, \ldots, x[|x| + |y| + 2]\}$ 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 $p_{\text{delete}}$.
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.

**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:

\begin{align}
    y'_{m} & \sim p(S_{m} | S_{M}) \quad \text{for } m \in M \\
    y^* &= \text{replace}(\tilde{y}, M, y')
\end{align}

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 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-initialised 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 pretrained weights to initialize our GEC model.

### 3 Experiments

#### 3.1 Data

To validate the effectiveness of our approach in non-English languages, we conduct our experiments on three GEC datasets: NLPC 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 Max-Match [Dahlmeier and Ng, 2012] scriptsootnote{https://github.com/nusnlp/m2scorer} to evaluate precision, recall and $F_{0.5}$. For non-autoregressive translation generation, we use datasets of the WMT20 news translation task [Barrault et al., 2020] – UN Parallel Corpus v1.0 [Ziemski et al., 2016] for Chinese and Russian, the combination of Europarl v10ootnote{http://www.statmt.org/europarl/v10/}, ParaCrawl v5.1 [Bañón et al., 2020] and Common Crawl corpus for German. We construct 10M synthetic sentence pairs in every language for pretraining and then fine-tune the GEC model on respective annotated datasets.

| Dataset            | Language | Train | Valid | Test   |
|--------------------|----------|-------|-------|--------|
| NLPCC 2018 Task 2  | Chinese  | 1.2M  | 3000  | 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.
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 pretraining are shown at the bottom. Rule denotes the synthetic data generated by rule-based corruption. Ours denotes data generated by our approach.

| Model | NLPCC-18 |
|-------|----------|
|       | P       | R       | F0.5    |
| YouDao [Fu et al., 2018] | 35.24 | 18.64 | 29.91 |
| AliGM [Zhou et al., 2018] | 41.00 | 13.75 | 29.36 |
| BLCU [Ren et al., 2018] | 47.23 | 12.56 | 30.57 |
| BERT-encoder [Wang et al., 2020b] | 41.94 | 22.02 | 35.51 |
| BERT-fuse [Wang et al., 2020b] | 32.20 | 23.16 | 29.87 |
| Dropout-Src [Junczys-Dowmunt et al., 2018] | 39.08 | 18.80 | 32.15 |
| MaskGEC [Zhao and Wang, 2020] | 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 |

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 [Sennrich et al., 2016] 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 based on the SentencePiece model [Kudo and Richardson, 2018]. During pretraining for German and Russian, following Nápálda 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 [Gehring et al., 2017]. BERT-encoder [Wang et al., 2020b] initializes the encoder of seq2seq model with BERT [Kenton and Toutanova, 2019]. 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 pretrained 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 Nápálda and Straka [2019] build confusion sets with edit distance, word embedding or spell-checker (e.g., Aspell dictionary\(^6\)). Flachs et al. [2021] utilizes Unmorph [Kirov et al., 2018] which provides morphological variants of words for word replacement operations. They also incorporate WikiEdits [Lichtarge et al., 2019] and Lang8 [Mizumoto et al., 2011] as additional training resources. Rothe et al. [2021] only applies language-agnostic operations without any confusion set. They pretrain a unified seq2seq model for 101 languages and fine-tune for respective languages. Katsumata and Komachi [2020] proposes to directly use mBART [Liu et al., 2020] without pretraining.

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 F0.5 descend comes from that rule-based corruption leverages many useful confusion sets. When combined with one of the character-level confusion sets, our approach obtains 39.42 F0.5 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 F0.5, which is the state-of-the-art result of the dataset.

\(^3\)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/
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. ⚫ It is good-performing with large-scale model size (up to 11B parameters) and our approach is better than its variant with similar model size.

Table 10 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 pretraining data and oversampled fine-tuning data, checkpoint averaging and so on. Rothe et al. [2021] obtains the best performance with large-scale model capacity (up to 11B parameters) and more sentence pairs for pretraining. Our approach significantly outperforms its base-scale variant. 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.

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.

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].

To make fair comparison with multiple synthetic construction approaches, we follow the experimental setting and model hyperparameters by 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 revision history. Although back-translate is effective for English [Xie et al., 2018; Kiyono et al., 2019], 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 error while AT generates clean sentences without grammatical errors or have minimal overlap with the original sentence. 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

We use the “transformer_clean_big_tpu” setting.

\footnote{We use the “transformer_clean_big_tpu” setting.}
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 → be delighted) [Bryant et al., 2017]. Such error type is beyond pre-defined rules and hard to simulate [Zhou et al., 2020]. The second-most improvement of our approach comes from ‘Determiner’ (e.g., few → some), since the NAT-based method can replace tokens with similar semantics based on bilingual context. This reason also applies to the improvement of ‘Adjective:FORM’ (e.g., goostest → best). Besides, our approach is even good at ‘Adposition’ (e.g., at → in), ‘Punctuation’ and ‘Spelling’, which are intuitively more suitable for rule-based methods.

Two exceptions are ‘Orthography’ and ‘Pronoun’. ‘Orthography’ refers to the error related with case or whitespace errors (e.g., Nexttime → next time), which the specific rules 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 the help of a 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 ‘Spelling’ errors, which may account for the significant 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. So are ‘Adposition’ (5-th) and ‘Determiner’ (6-th) generated by our approach.

### 3.7 Case Study

To give a qualitative analysis of generated erroneous sentences, we present examples 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. Roundtrip translation generates error-free sentences. In contrast, our approach generates less fluent sentences by paraphrasing the corrupted contents and maintaining the meaning preserved in the corresponding English sentences.

### 4 Related Work

Pretraining 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], Czech [Náplava and Straka, 2019], and Spanish [Davidson et al., 2020] only contain less than 50000 pairs, which results in a high requirement on the quality of synthetic data construction.
Back-translation (BT) is the reverse of GEC, which takes corrected sentences as input and error sentences as output. It is popular and effective for English GEC [Ge et al., 2018; Xie et al., 2018; Kiyono et al., 2019; Wan et al., 2020; Stahlberg and Kumar, 2021] 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 and achieves inferior performance in the non-English language as shown in our experiment.

The most effective construction method on 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 is another useful GEC pretraining resource [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] or 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 pretrained 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 constraint to avoid inconsistency of candidate words with the original meaning, which plays an important role for data construction of GEC as shown in our experiment.

How to leverage pretrained language models in GEC seq2seq models has been extensively explored [Katsumata and Komachi, 2020; Wang et al., 2020b; Kaneko et al., 2020]. In this paper, we initialize the model with DeltaLM [Ma et al., 2021], which adjusts the InfoXLM-initialised 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 corruption 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 translation corpus to generate the unlimited number of error-corrected sentence pairs.

References

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pia Sempere, Gema Ramírez-Sánchez, Elsa Sarrias, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. ParaCrawl: Web-scale acquisition of parallel corpora. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4555–4567, Online, July 2020. Association for Computational Linguistics.

Loïc Barrault, Magdalena Biesielska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joannis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. Findings of the 2020 conference on machine translation (WMT20). In Proceedings of the Fifth Conference on Machine Translation, pages 1–55, Online, November 2020. Association for Computational Linguistics.

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, pages 79–84, 2018.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 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, 2017.

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

Daniel Dahlmeier and Hwee Tou Ng. 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, 2012.

Sam Davidson, Aaron Yamada, Paloma Fernandez Mira, Agustina Carando, Claudia H Sanchez Gutierrez, and Kenji Sagae. Developing nlp tools with a new corpus of learner spanish. In Proceedings of the 12th Language Resources and Evaluation Conference, pages 7238–7243, 2020.
Cunxiao Du, Zhaopeng Tu, and Jing Jiang. Order-agnostic cross entropy for non-autoregressive machine translation. arXiv preprint arXiv:2106.05093, 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, pages 117–122, 2021.

Kai Fu, Jin Huang, and Yitao Duan. Youdao’s winning solution to the nlpc-2018 task 2 challenge: a neural machine translation approach to Chinese grammatical error correction. In CCF International Conference on Natural Language Processing and Chinese Computing, pages 341–350. Springer, 2018.

Tao Ge, Furu Wei, and Ming Zhou. Fluency boost learning and inference for neural grammatical error correction. In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1055–1065, 2018.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In International Conference on Machine Learning, pages 1243–1252. PMLR, 2017.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. 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 6112–6121, 2019.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. Minimally-augmented grammatical error correction. In Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019), pages 357–363, 2019.

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

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 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, 2018.

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 Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4248–4254, 2020.

Satoru Katsumata and Mamoru Komachi. Stronger baselines for grammatical error correction using a pretrained encoder-decoder model. In Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 827–832, 2020.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of NAACL-HLT, pages 4171–4186, 2019.

Christo Kirov, Ryan Cotterell, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sabrina J Mielke, Arya D McCarthy, Sandra Kübler, et al. Unimorph 2.0: universal morphology. arXiv preprint arXiv:1810.11101, 2018.

Shun Kiyono, Jun Suzuki, Masato Mita, Tomoyo Mizumoto, and Kentaro Inui. 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, 2019.

Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 66–71, 2018.

Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. arXiv preprint arXiv:1901.07291, 2019.

Jingjing Li, Zichao Li, Lili Mou, Xin Jiang, Michael Lyu, and Irwin King. Unsupervised text generation by learning from search. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 10820–10831. Curran Associates, Inc., 2020.

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 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, 2019.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. Transactions of the Association for Computational Linguistics, 8:726–742, 2020.

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.

Tomoyo Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. Mining revision log of language learning srs for automated japanese error correction of second language learners. In Proceedings of 5th International Joint
Conference on Natural Language Processing, pages 147–155, 2011.

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), pages 346–356, 2019.

Qiu Ran, Yankai Lin, Peng Li, and Jie Zhou. Learning to recover from multi-modality errors for non-autoregressive neural machine translation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 3059–3069, 2020.

Hongkai Ren, Liner Yang, and Endong Xun. A sequence to sequence learning for chinese grammatical error correction. In CCF International Conference on Natural Language Processing and Chinese Computing, pages 401–410. Springer, 2018.

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.

Alla Rozovskaya and Dan Roth. Grammar error correction in morphologically rich languages: The case of russian. Transactions of the Association for Computational Linguistics, 7:1–17, 2019.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1715–1725, 2016.

Felix Stahlberg and Shankar Kumar. Synthetic data generation for grammatical error correction with tagged corruption models. In Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications, pages 37–47, 2021.

Zhaohong Wan, Xiaojun Wan, and Wenguang Wang. Improving grammatical error correction with data augmentation by editing latent representation. In Proceedings of the 28th International Conference on Computational Linguistics, pages 2202–2212, 2020.

Chengcheng 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, 34(6):106, 2020.

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

Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Y Ng, and Dan Jurafsky. 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, 2018.

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.

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

Yuanyuan Zhao, Nan Jiang, Weiwei Sun, and Xiaojun Wan. Overview of the nlpc 2018 shared task: Grammatical error correction. In CCF International Conference on Natural Language Processing and Chinese Computing, pages 439–445. Springer, 2018.

Junpei Zhou, Chen Li, Hengyou Liu, Zuyi Bao, Guangwei Xu, and Linlin Li. Chinese grammatical error correction using statistical and neural models. In CCF International Conference on Natural Language Processing and Chinese Computing, pages 117–128. Springer, 2018.

Michał Ziemska, Marcin Junczys-Dowmunt, and Bruno Pouliquen. The united nations parallel corpus v1.0. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16), pages 3530–3534, 2016.

A Hyper-parameters

The parameters for NAT-based data construction and post edit are presented in Table 8. The hyper-parameters of training the Transformer on NLPC 2018 Task 2 (Chinese) are listed in Table 9. The hyper-parameters for German and Russian are shown in Table 10. 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 8: Parameters for NAT-based data construction and post edit.

| Language | $p_{noise}$ | mask | insert | delete | swap | $p_{noise}$ | substitute | insert | delete | swap | recase |
|----------|-------------|------|--------|--------|------|-------------|------------|--------|--------|------|--------|
| Chinese  | 0.5         | 0.7  | 0.1    | 0.1    | 0.1  | 0.05        | 0.3        | 0.2    | 0.3    | 0.2  | 0     |
| German   | 0.3         | 0.65 | 0.15   | 0.15   | 0.05 | 0.02        | 0.25       | 0.25   | 0.2    | 0.2  | 0.1    |
| Russian  | 0.15        | 0.65 | 0.15   | 0.15   | 0.05 | 0.02        | 0.25       | 0.25   | 0.2    | 0.2  | 0.1    |

### Table 9: Hyper-parameters values of pretraining and fine-tuning on NLPCC 2018 Task 2 (Chinese).

| Configurations | Values |
|----------------|--------|
| **Pretraining** |        |
| Model Architecture | Transformer (base) |
| Devices | 8 Nvidia V100 GPU |
| Max tokens per GPU | 5120 |
| Update Frequency | 8 |
| Optimizer | Adam  |
| ($\beta_1=0.9$, $\beta_2=0.98$, $\epsilon=1 \times 10^{-8}$) |
| Learning rate | $7 \times 10^{-4}$ |
| 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 |
| Update Frequency | [2, 4] |
| Learning rate | $[5 \times 10^{-4}, 7 \times 10^{-4}]$ |
| Warmup | 4000 |

### Table 10: Hyper-parameters values of pretraining and fine-tuning on Falko-Merlin (German) and RULEC-GEC (Russian).

| Configurations | Values |
|----------------|--------|
| **Pretraining** |        |
| Model Architecture | Transformer (base) |
| Devices | 8 Nvidia V100 GPU |
| Max tokens per GPU | 5120 |
| Update Frequency | 8 |
| Optimizer | Adam  |
| ($\beta_1=0.9$, $\beta_2=0.98$, $\epsilon=1 \times 10^{-8}$) |
| Learning rate | $[5 \times 10^{-4}, 7 \times 10^{-4}]$ |
| 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 | $[3 \times 10^{-4}, 5 \times 10^{-4}, 7 \times 10^{-4}]$ |
| Dropout | [0.1, 0.2, 0.3] |
| Warmup | 2000 |