Multi-pass Decoding for Grammatical Error Correction

Xiaoying Wang[1] Lingling Mu[1] Jingyi Zhang[2] Hongfei Xu[1*]
[1]Zhengzhou University, Henan, China
[2]Hasso Plattner Institute, University of Potsdam, Germany
xywangnlp@gs.zzu.edu.cn, iellmu@zzu.edu.cn, Jingyi.Zhang@hpi.de, hfxunlp@foxmail.
[*]Corresponding author: Hongfei Xu

February 21, 2026

Abstract

Sequence-to-sequence (seq2seq) models achieve comparable or better grammatical error correction performance compared to sequence-to-edit (seq2edit) models. Seq2edit models normally iteratively refine the correction result, while seq2seq models decode only once without aware of subsequent tokens. Iteratively refining the correction results of seq2seq models via Multi-Pass Decoding (MPD) may lead to better performance. However, MPD increases the inference costs. Deleting or replacing corrections in previous rounds may lose useful information in the source input. We present an early-stop mechanism to alleviate the efficiency issue. To address the source information loss issue, we propose to merge the source input with the previous round correction result into one sequence. Experiments on the CoNLL-14 test set and BEA-19 test set show that our approach can lead to consistent and significant improvements over strong BART and T5 baselines (+1.80, +1.35, and +2.02 $F_{0.5}$ for BART 12-2, large and T5 large respectively on CoNLL-14 and +2.99, +1.82, and +2.79 correspondingly on BEA-19), obtaining $F_{0.5}$ scores of 68.41 and 75.36 on CoNLL-14 and BEA-19 respectively.

1 Introduction

Grammatical Error Correction (GEC) aims to correct grammatical errors in the given sentence [Ng et al.2013, Ng et al.2014]. Nowadays, there are two mainstream GEC approaches. Sequence-to-edit (seq2edit) methods regard GEC as a sequence tagging task, where the model predicts edit tags (e.g., keep, delete, insert, replace, etc.) for each token iteratively for multiple rounds until all tokens are assigned the keep tag [Malmi et al.2019, Stahlberg and Kumar2020, Omelianchuk et al.2020, Yuan et al.2021]. Seq2edit methods normally require to correct for a number of correction rounds to complete the correction. In contrast, Sequence-to-sequence (seq2seq) approaches consider the GEC task as Machine Translation (MT) from ungrammatical texts to grammatical texts [Zhao et al.2019, Kiyono et al.2019, Wang et al.2021, Li et al.2022, Fang et al.2023a]. The seq2seq model encodes the input sentence and auto-regressively decodes the corrected sentence. Current methods normally utilize the pre-trained models for better performance, such as BERT [Devlin et al.2019] and XLNet [Yang et al.2019] for seq2edit [Omelianchuk et al.2020], and BART [Lewis et al.2020] and T5 [Raffel et al.2020] for seq2seq [Kaneko et al.2020, Liu et al.2021].

Seq2seq models lead to comparable or better performance than seq2edit approaches without using language-specific edit operations. However, current seq2seq GEC studies typically decode only once without aware of subsequent tokens. Multi-Pass Decoding (MPD) may enhance the performance through iterative refinements [Ge et al.2018]. Training MPD models to generate the gold reference given its correction results may also benefit its learning via self-correction [Li et al.2021].

Multi-pass decoding leads to two problems: 1) iterative decoding increases the inference computational costs, and 2) deleting or replacing in previous correction rounds may incur information loss. We propose to introduce an early-stop mechanism to alleviate the efficiency issue. It takes the hidden representation of the end-of-sentence token (`<eos>`) as input, and stops MPD in cases: 1) the next round's correction result matches the current correction result, or 2) the next round's correction result has a larger edit distance to the reference.

As for the information loss issue, we present methods to merge the source sentence and the previous round's correction output into a single sequence, as pre-trained models normally do not have multiple encoders for more than one inputs. We evaluate our approach on the CoNLL 2014 and BEA 2019 GEC shared tasks, and obtain significant improvements over the strong BART and T5 baselines, showing the effectiveness of our method.

- To improve the efficiency of multi-pass decoding, we present an early-stop mechanism to terminate the multi-pass decoding when the next decoding round would not lead to better correction result.

- We propose source information fusion methods to address the information loss issue due to deleting or replacing edit operations in preceding correction rounds, and present comparison-based sequence merging approach to ensure the efficiency of source information fusion.
- Our method brings about +1.80, +1.35, and +2.02 $F_{0.5}$ improvements over the strong BART 12-2, large and T5 baselines respectively on CoNLL-14 test set, and +2.99, +1.82, and +2.79 correspondingly on the BEA-19 test set, showing the effectiveness of our approach.

2 Preliminaries: Sequence-to-sequence GEC

The seq2seq model M comprises an encoder and a decoder. It takes the input sequence x to correct, and generates the corrected sequence $\hat{x}$.

The encoder takes the input sequence x, and computes the contextual hidden state vectors h_e:

$$h_e = \text{encoder}(x) \tag{1}$$

The decoder generates the hidden state h^d_k based on the encoder hidden states h_e and the decoding history $\hat{x}_{<k}$:

$$h^d_k = \text{decoder}(h_e, \hat{x}_{<k}) \tag{2}$$

where $\hat{x}_k$ is the kth token in the sequence. $\hat{x}_0$ is the start-of-sentence token `<sos>`. $\hat{x}_{<k}$ means the token sequence from $\hat{x}_0$ to $\hat{x}_{k-1}$.

The decoder classifier conditions on the decoder hidden state h^d_k, and predicts the probability of each token in the vocabulary. The decoder selects the token with the highest probability as $\hat{x}_k$ for subsequent decoding steps:

$$\hat{x}_k = \text{classifier}(h^d_k) \tag{3}$$

The decoder repeats this process until the classifier produces the end-of-sentence token (`<eos>`) given the hidden state $h^d_{\text{¡eos¿}}$.

Pre-training by reconstructing the corrupted text can compress the knowledge of large-scale corpus into model parameters. And fine-tuning pre-trained models (such as BART and T5) for GEC can lead to better performance [Sun et al.2021, Rothe et al.2021].

3 Our Method

3.1 Multi-pass Decoding with Early-stop

In the GEC task, the seq2seq GEC model M takes the input sentence x that might be incorrect, and generates the corrected sentence $\hat{x}$. Instead of using $\hat{x}$ as the final result, multi-pass decoding iteratively repeats the correction process, by feeding the correction result of the previous round $\hat{x}^{t-1}$ into the model and asking the model to correct $\hat{x}^{t-1}$ into $\hat{x}^t$, until $\hat{x}^t = \hat{x}^{t-1}$. The termination condition involves decoding the same sequence twice. This increases the computational costs for inference while improving the performance. We train an early-stop mechanism together with the seq2seq model to address this issue.

Algorithm 1 Multi-pass decoding with early-stop.

```
Require: Input sentence to correct $x$, GEC model $M$, early-stop classifier $C_e$, maximum number of decoding rounds $n$, early-stop threshold $\tau$
Ensure: Corrected sentence $y$
$\hat{x}^0, h^0_{\text{¡eos¿}} = M(x)$
$p_e = C_e(h^0_{\text{¡eos¿}})$
if $p_e > \tau$ then
    $y = \hat{x}^0$
else
    for $t = 1$ to $n$ do
        $\hat{x}^t, h^t_{\text{¡eos¿}} = M(x, \hat{x}^{t-1})$
        $p_e = C_e(h^t_{\text{¡eos¿}})$
        $y = \hat{x}^t$
        if $\hat{x}^{t-1} == \hat{x}^t$ or $p_e > \tau$ then
            break
        end if
    end for
end if
return $y$
```

The early-stop mechanism introduces a lightweight logistic regression classifier C_e to predict the probability of stopping the multi-pass decoding. C_e consists of a weight vector w_e and a bias scalar b_e. During the decoding of $\hat{x}^{t-1}$, we take the decoder hidden representation $h^{t-1}_{\text{¡eos¿}}$ of the special end-of-sentence token (`<eos>`) to compute the early-stop probability:

$$p_e = \sigma(h^{t-1}_{\text{¡eos¿}} \cdot w_e + b_e) \tag{4}$$

where "$\cdot$" and "σ" are dot-product and sigmoid.

We optimize the Binary Cross Entropy (BCE) loss between p_e and the early-stop label y_e:

$$l_e = \text{BCE}(p_e, y_e) \tag{5}$$

In MPD training, we first decode $\hat{x}^t$, and label y_e of the previous decoding round based on $\hat{x}^{t-1}$, $\hat{x}^t$ and the gold GEC reference r. y_e is true if: 1) $\hat{x}^t$ equals to $\hat{x}^{t-1}$, or 2) the edit distance between r and $\hat{x}^t$ is larger than that with $\hat{x}^{t-1}$. The edit-distance condition aims to ensure that multi-pass decoding will not deteriorate the performance. To provide the training label of the current decoding round for the early-stop classifier C_e, the decoding result of the next round $\hat{x}^{t+1}$ is always generated during training, to compare the edit distances between the reference with the current round decoding result $\hat{x}^t$ and the next round decoding result $\hat{x}^{t+1}$.

The training loss is the weighted combination of the original seq2seq generation loss l_{seq2seq} and l_e:

$$l = l_{\text{seq2seq}} + \lambda * l_e \tag{6}$$

We use Algorithm 1 for inference. We use a maximum number of decoding rounds n of 3, and early-stop if $\hat{x}^t = \hat{x}^{t-1}$ or $p_e > \tau$. λ and τ are default to 1 and 0.5 respectively. λ of 1 treats the correction task and the early-stop classifier equally during training. A threshold of 0.5 indicates to early-stop if the probability is larger than 0.5, which is reasonable for the binary classification task. The number of decoding rounds is tested on the development set, and using a value larger than 3 does not lead to better performance. We did not carefully tune λ and τ despite this may lead to better performance.

3.2 Source Information Fusion during Iterative Correction

If the model deletes or replaces tokens in previous rounds, the original tokens are infeasible for thereafter correction rounds, even they might be valuable references for subsequent correction rounds. As shown in the example in Figure 1, the model requires to correct:

"We go to the orchard and brought apples, but forget pears." to: "We go to the orchard and buy apples, but forget pears."

The model only fixes the tense of the verb "brought" by replacing it with "bring" in the first round. When the model correcting the semantic meaning of the verb "bring" in the second round, choosing from "pick" and "buy" could be hard if it is not aware of the existence of the wrong verb "brought" in the source input. Despite "brought" is wrongly spelt, it encourages the model to select "buy" instead of "pick", as the past tense of "buy" ("bought") is closer to "brought" than the past tense of "pick" ("picked").

Thus, keeping all source tokens feasible in all correction rounds may benefit the performance. But pre-trained seq2seq models normally do not have multiple encoders for both the source sentence x and the decoding result of the previous correction round $\hat{x}^{t-1}$. Concatenating x and $\hat{x}^{t-1}$ as the input of the encoder results in long and redundant sequences. The unchanged tokens also have two distant positions in the concatenated sequence. To encode x and $\hat{x}^{t-1}$ efficiently with the single encoder, we propose to merge x and $\hat{x}^{t-1}$ into a single sequence, as shown in Figure 1. Specifically, we first compare x with $\hat{x}^{t-1}$, then extract the common and different segments, and finally merge the segments into a single sequence according to their orders in corresponding sequences. The merged sequence contains unchanged tokens, inserted tokens and deleted tokens with their original orders. Replacing can be regarded as an insertion plus a deletion.

IMAGE NOT PROVIDED
Source information fusion diagram showing edit
operations and position mappings

Figure 1: Source information fusion.

We use edit tags or separated position encodings to distinguish tokens in the merged sequence. For edit tags, we use "e" (equal), "d" (delete) and "i" (insert) to represent the tokens' roles in the merged sequence, standing for tokens in both x and $\hat{x}^{t-1}$, appearing only in x, and newly added to $\hat{x}^{t-1}$ respectively. We add an embedding layer for edit tags and add the edit embeddings to the word embeddings of the seq2seq model before encoder layers.

For position encoding, we use 2 position labels for the merged sequence: source position stands for the token's position in x and decode position for its position in $\hat{x}^{t-1}$. The position of the token is 0 if it does not appear in the sequence. To mitigate the gap between the new position embeddings and pre-trained models, the new position embeddings are initialized based on the pre-trained position embeddings. But we reduce the weights of position embeddings by half. This is because position embeddings are added twice when using the merged sequence as the input: once for the source position and another for the decode position.

4 Experiments

4.1 Settings

To test the effectiveness of our approach, we conducted experiments using the strong BART (12-2), BART (12-12) and T5 large baselines, and strictly followed the settings of yakovlev2023 for data processing and BART fine-tuning. We used the same data set as yakovlev2023, and the models were fine-tuned for 3 stages following omelianchuk2020. Our Multi-Pass Decoding (MPD) method was only applied in the last stage. As this is more efficient than applying to all stages, and the model may produce more reasonable correction results ($\hat{x}^0$ is normally no worse than x compared to r) after the second stage. The original GEC training loss ($M(x) \rightarrow r$) was still kept. We implemented our approaches based on the Neutron implementation [Xu and Liu2019] of the Transformer.

We evaluated on the CoNLL 2014 test set [Ng et al.2014] with M2 scorer [Dahlmeier and Ng2012] and the BEA 2019 test set, and validated on the BEA 2019 (W&I+L) development set, and reported precision (P), recall (R) and $F_{0.5}$ scores following common practices.

Despite all these datasets are in English, they are widely used by the community, and we suggest that our approaches are language-agnostic and can be easily adapted to the other languages, as verified in Section 4.5.

4.2 Main Results

Based on the ablation studies, the MPD training only used single-pass decoding results, and the inference was multi-pass with early-stop (Section 4.3). We used both edit tags and position encoding for source information fusion (Section 4.4). Results on the CoNLL 2014 test set and BEA 2019 test set are shown in Tables 1 and 2 respectively.

Table 1 shows that: 1) the performance of the powerful LLaMa 2-7B Large Language Model (LLM) is far behind fine-tuned seq2edit and seq2seq methods even after fine-tuning, and 2) MPD can significantly and consistently improve the performance of all our baselines with different model sizes and settings (+1.80, +1.35 and +2.02 $F_{0.5}$ over BART 12-2, BART 12-12 and T5 large respectively). Results in Table 2 on the BEA-19 development set are also consistent. Although we only applied our methods to the widely used BART and T5 baselines, we suggest that our method is likely to bring about further improvements with more advanced baseline models.

4.3 Ablation Study for MPD Training and Inference

In addition to training the model to generate the gold reference r given the input x, the MPD training also takes the output of the previous decoding round $\hat{x}^{i-1}$ as the input. The output of the previous decoding round may be either the result of a single decoding round like omelianchuk2020, or the result of several decoding rounds until the inference termination condition. We study the effects of single-round and multi-round decoding for MPD training while using multi-pass decoding with early-stop for inference.

For single-round decoding in MPD training, we use the model to decode x into $\hat{x}^0$, and train the model to generate r given x and $\hat{x}^0$:

$$M(x, \hat{x}^0) \rightarrow r \tag{7}$$

For multi-round decoding in MPD training, we start from x as $\hat{x}^{-1}$ and iteratively decode $\hat{x}^{i-1}$ to $\hat{x}^i$ for several rounds until meeting the termination condition, and train the model to generate r given x and $\hat{x}^i$:

$$M(x, \hat{x}^i) \rightarrow r \tag{8}$$

We also study the effects of the maximum number of decoding rounds with/without early-stop for MPD inference while using single-round decoding in MPD training. Additionally, we compare our simple early-stop mechanism with the policy network proposed by geng2018. geng2018 employ reinforcement learning method to decide the number of decoding rounds based on the differences between the two consecutive decoding passes, and optimize the BLEU-based reward for machine translation. While in our experiment for the GEC task, we used the $F_{0.5}$ score as the reward instead of BLEU.

To analyze the inference efficiency of our approach, we compare our method with the BART (12-4) baseline with vanilla fine-tuning and the ensemble of 2 vanilla BART (12-2) models initialized with different random seeds [Tarnavskyi et al.2022]. Both the BART (12-4) setting with 4 decoder layers and the ensemble can lead to better performance but slower inference speed compared to the BART (12-2) baseline.

Results in Table 3 show that: 1) for MPD training, both settings obtain similar performance, but the single-round decoding setting achieves slightly higher $F_{0.5}$ scores while being more computationally efficient, 2) the performances of different numbers of maximum decoding rounds are also similar, larger n leads to slower inference, but the early-stop mechanism can mitigate this and bring about the best performance, 3) multi-pass decoding based on the policy network can also lead to consistent $F_{0.5}$ improvements on the two shared tasks, but our simple early-stop method is more efficient than the policy network [Geng et al.2018] and leads to higher $F_{0.5}$ scores, and 4) the performance of our MPD method with the BART (12-2) setting achieves better performance than both the BART (12-4) baseline with vanilla fine-tuning and the ensemble of 2 vanilla BART (12-2) models, and it is also faster than the BART (12-4) and the ensemble baselines for inference. This shows that our method can achieve better performance more efficiently.

Previous state-of-the-art multi-pass decoding study for NMT [Geng et al.2018] uses very complex reinforcement learning method to decide the required number of decoding rounds. The reinforcement learning training might be unstable and lead to unstable performances. Our supervised method directly trains the simple binary classifier based on the representation of the decoded sequence. We suggest that our early-stop method is easy to implement and very effective in practice.

4.4 Effects of Source Information Fusion

We test the effects of different source information fusion methods with the BART (12-2) setting, including: 1) using only $\hat{x}^{t-1}$ instead of both $\hat{x}^{t-1}$ and x for MPD inference ("None"), 2) sequence concatenation ("Concat"), 3) edit tags ("Edit"), 4) position encoding ("Pos"), and 5) both edit tags and position encoding ("Pos+Edit"). Results are shown in Table 4.

Table 4 shows that: 1) vanilla MPD without source information fusion ("None") can already lead to +0.80 and +1.09 $F_{0.5}$ improvements on the BEA-19 development set and the CoNLL-14 test set respectively, showing the effectiveness of multi-pass decoding, 2) source information fusion through sequence concatenation ("Concat") can lead to +0.46 and +0.12 $F_{0.5}$ score improvements on the BEA 2019 development set and the CoNLL-14 test set respectively than without source information fusion ("None"), showing the positive effects of source information fusion, 3) both position encoding ("Pos") and edit tags ("Edit") bring about higher $F_{0.5}$ scores than sequence concatenation ("Concat") while being more efficient, empirically showing the advantages of our sequence merging approach, and position encoding consistently brings about slightly better performance than edit tags, probably because of the pre-trained position embedding initialization, and 4) the combination of position encoding and edit tags ("Pos+Edit") leads to the best performance, but the difference is small compared to using only position encoding, probably because position encoding and edit tags provide similar information in denoting the roles of tokens in the two sequences despite in different forms and are complementary to some extent.

4.5 Verification on the Other Language

We suggest that our approach is language-agnostic. To test its effectiveness on the other languages, we also conducted experiments on Chinese GEC datasets exactly following the experiment settings of yang2024. Specifically, we used the combination of the Lang-8 corpus provided by NLPCC 2018 [Zhao et al.2018b], the HSK dataset and FCGEC training set [Xu et al.2022] as the training set, MuCGEC development set [Zhang et al.2022a] for validation, and tested on the NLPCC 2018 test set, FCGEC development set and NaCGEC test set [Ma et al.2022].

For evaluation metrics, we follow previous work and report word-level precision (P)/ recall (R)/ F-measure ($F_{0.5}$) performance on NLPCC18-Test using the official MaxMatch scorer [Ng et al.2014] and PKUNLP word segmentation tool. For the FCGEC development set and the NaCGEC test set, we report the character-level P/ R/ $F_{0.5}$ scores using the ChERRANT scorer [Zhang et al.2022a].

We use a large Transformer and the pre-trained BART model as the baselines. The batch size is 1024 and the maximum sentence length of training data is 128. The maximum number of training epochs is 20 and 10, respectively, and the beam size is 10. Results are shown in Tables 5 and 6.

Tables 5 and 6 show similar phenomena as Tables 1 and 2. Our method also leads to consistent and significant improvements on all Chinese test sets (+2.06, +2.30, and +3.45 $F_{0.5}$ score improvements on the NLPCC 2018 test set, FCGEC development set and the NaCGEC test set respectively over the strong BART baseline).

5 Related Work

Seq2edit GEC. Seq2edit GEC methods [Malmi et al.2019, Awasthi et al.2019, Stahlberg and Kumar2020] iteratively assign edit operations to tokens, such as insertion, deletion, replacement, or language-specific transformations [Omelianchuk et al.2020], etc., and improve the performance with self-correction [Parnow et al.2021], type-based multi-turn training [Lai et al.2022], decoupled error detection [Tan et al.2023], etc. Due to the limited correction ability of pre-defined edit operations, seq2edit models normally require to iteratively correct the sentence for multiple rounds and naturally benefit from multi-round correction.

Seq2seq GEC. Seq2seq GEC methods [Fang et al.2023a, Li et al.2022, Liu et al.2021, Wang et al.2021] transform the input sentence using seq2seq models. Recent studies mainly focus on: 1) unsupervised pre-training [Grundkiewicz et al.2019], 2) shallow aggressive [Sun et al.2021] or non-autoregressive decoding [Yakovlev et al.2023] to accelerate the inference, 3) leveraging language-specific knowledge [Mita and Yanaka2021, Fei et al.2023, Kaneko et al.2022] or syntax [Zhang et al.2022b], 4) decoding methods on fluency boost [Ge et al.2018], SMT and NMT integration [Grundkiewicz and Junczys-Dowmunt2018], precision-recall trade-off [Sun and Wang2022], reranking [Zhang et al.2023] or decoding interventions [Zhou et al.2023], and 5) optimized multi-task training schedule [Bout et al.2023]. As most seq2seq methods only decode once, we suggest that our work is complementary and can be easily adapted to these methods for further improvements.

MPD in NMT. MPD has been investigated to improve Neural Machine Translation (NMT) [Xia et al.2017, Mahmood et al.2017, Zhang et al.2018, Geng et al.2018, Liu et al.2019]. Automatic Post-Editing (APE) can also be regarded as a special case of MPD [Correia and Martins2019, Pal et al.2020, Bhattacharyya et al.2022, Jung et al.2023]. These studies also underline the importance of source information fusion, but they employ dual-encoder structures for the source input and the decoded sequence as they are in different languages and quite different in spelling. While we are the first: 1) addressing the efficiency issue of MPD with an early-stop mechanism, and 2) deriving source information fusion methods to benefit from pre-trained seq2seq models that have only a single encoder, given that the two sequences in GEC are normally close.

6 Conclusion

We utilize multi-pass decoding to improve the performance of seq2seq grammatical error correction. We present an early-stop mechanism to alleviate the inference efficiency issue, and derive source information fusion approaches to address the source information loss issue.

Our experiments on the CoNLL-14 test set and the BEA-19 test set show that our approach can lead to significant improvements (+1.80, +1.35, +2.02 $F_{0.5}$ scores for BART 12-2, large and T5 large respectively on CoNLL-14 and +2.99,

+1.82, and +2.79 correspondingly on BEA-19) over strong baselines, showing the effectiveness of our method.

Limitations

We only applied our methods on the widely used BART and T5 baselines, without applying it to the state-of-the-art sequence-to-sequence grammatical error correction framework.

Acknowledgements

We thank anonymous reviewers for their insightful comments. Xiaoying Wang, Lingling Mu and Hongfei Xu acknowledge the support of the National Natural Science Foundation of China (Grant No. 62306284), China Postdoctoral Science Foundation (Grant No. 2023M743189), and the Natural Science Foundation of Henan Province (Grant No. 232300421386). Jingyi Zhang is supported by the German Federal Ministry for Education and Research (BMBF) within the project "KI-Servicezentrum Berlin Brandenburg" 01IS22092.

References

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

[Bhattacharyya et al.2022] Pushpak Bhattacharyya, Rajen Chatterjee, Markus Freitag, Diptesh Kanojia, Matteo Negri, and Marco Turchi. 2022. Findings of the wmt 2022 shared task on automatic post-editing. In *Proceedings of the Seventh Conference on Machine Translation*, pages 109–117, Abu Dhabi. Association for Computational Linguistics.

[Bout et al.2023] Andrey Bout, Alexander Podolskiy, Sergey Nikolenko, and Irina Piontkovskaya. 2023. Efficient grammatical error correction via multi-task training and optimized training schedule. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5800–5816, Singapore. Association for Computational Linguistics.

[Cao and Zhao2023] Hejing Cao and Dongyan Zhao. 2023. Leveraging denoised Abstract Meaning Representation for grammatical error correction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7180–7188, Toronto, Canada. Association for Computational Linguistics.

[Cao et al.2023a] Hang Cao, Zhiquan Cao, Chi Hu, Baoyu Hou, Tong Xiao, and Jingbo Zhu. 2023a. Improving autoregressive grammatical error correction with non-autoregressive models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12014–12027, Toronto, Canada. Association for Computational Linguistics.

[Cao et al.2023b] Hannan Cao, Liping Yuan, Yuchen Zhang, and Hwee Tou Ng. 2023b. Unsupervised grammatical error correction rivaling supervised methods. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3072–3088, Singapore. Association for Computational Linguistics.

[Correia and Martins2019] Gonçalo M. Correia and André F. T. Martins. 2019. A simple and effective approach to automatic post-editing with transfer learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3050–3056, Florence, Italy. Association for Computational Linguistics.

[Dahlmeier and Ng2012] Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.

[Devlin et al.2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[Du2019] Zeyao Du. 2019. Gpt2-chinese: Tools for training gpt2 model in chinese language. `https://github.com/Morizeyao/GPT2-Chinese`.

[Fang et al.2023a] Tao Fang, Jinpeng Hu, Derek F. Wong, Xiang Wan, Lidia S. Chao, and Tsung-Hui Chang. 2023a. Improving grammatical error correction with multimodal feature integration. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9328–9344, Toronto, Canada. Association for Computational Linguistics.

[Fang et al.2023b] Tao Fang, Xuebo Liu, Derek F. Wong, Runzhe Zhan, Liang Ding, Lidia S. Chao, Dacheng Tao, and Min Zhang. 2023b. TransGEC: Improving grammatical error correction with translationese. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3614–3633, Toronto, Canada. Association for Computational Linguistics.

[Fei et al.2023] Yuejiao Fei, Leyang Cui, Sen Yang, Wai Lam, Zhenzhong Lan, and Shuming Shi. 2023. Enhancing grammatical error correction systems with explanations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7489–7501, Toronto, Canada. Association for Computational Linguistics.

[Fu et al.2018] Kai Fu, Jin Huang, and Yitao Duan. 2018. Youdao's winning solution to the nlpcc-2018 task 2 challenge: A neural machine translation approach to chinese grammatical error correction. In *Natural Language Processing and Chinese Computing*.

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

[Geng et al.2018] Xinwei Geng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2018. Adaptive multi-pass decoder for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 523–532, Brussels, Belgium. Association for Computational Linguistics.

[Grundkiewicz and Junczys-Dowmunt2018] Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2018. Near human-level performance in grammatical error correction with hybrid machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 284–290, New Orleans, Louisiana. Association for Computational Linguistics.

[Grundkiewicz et al.2019] Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy. Association for Computational Linguistics.

[Hinson et al.2020] Charles Hinson, Hen-Hsen Huang, and Hsin-Hsi Chen. 2020. Heterogeneous recycle generation for Chinese grammatical error correction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2191–2201, Barcelona, Spain (Online). International Committee on Computational Linguistics.

[Jung et al.2023] Baikjin Jung, Myungji Lee, Jong-Hyeok Lee, and Yunsu Kim. 2023. Bring more attention to syntactic symmetry for automatic postediting of high-quality machine translations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1433–1441, Toronto, Canada. Association for Computational Linguistics.

[Kaneko et al.2020] Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4248–4254, Online. Association for Computational Linguistics.

[Kaneko et al.2022] Masahiro Kaneko, Sho Takase, Ayana Niwa, and Naoaki Okazaki. 2022. Interpretability for language learners using example-based grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7176–7187, Dublin, Ireland. Association for Computational Linguistics.

[Kiyono et al.2019] Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. An empirical study of incorporating pseudo data into grammatical error correction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1236–1242, Hong Kong, China. Association for Computational Linguistics.

[Lai et al.2022] Shaopeng Lai, Qingyu Zhou, Jiali Zeng, Zhongli Li, Chao Li, Yunbo Cao, and Jinsong Su. 2022. Type-driven multi-turn corrections for grammatical error correction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3225–3236, Dublin, Ireland. Association for Computational Linguistics.

[Lewis et al.2020] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

[Li et al.2022] Bei Li, Quan Du, Tao Zhou, Yi Jing, Shuhan Zhou, Xin Zeng, Tong Xiao, JingBo Zhu, Xuebo Liu, and Min Zhang. 2022. ODE transformer: An ordinary differential equation-inspired model for sequence generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8335–8351, Dublin, Ireland. Association for Computational Linguistics.

[Li et al.2021] Chong Li, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. 2021. Exploration and exploitation: Two ways to improve Chinese spelling correction models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 441–446, Online. Association for Computational Linguistics.

[Li et al.2019] Si Li, Jianbo Zhao, Guirong Shi, Yuanpeng Tan, Huifang Xu, Guang Chen, Haibo Lan, and Zhiqing Lin. 2019. Chinese grammatical error correction based on convolutional sequence to sequence model. volume 7, pages 72905–72913.

[Li et al.2023] Yinghao Li, Xuebo Liu, Shuo Wang, Peiyuan Gong, Derek F. Wong, Yang Gao, Heyan Huang, and Min Zhang. 2023. TemplateGEC: Improving grammatical error correction with detection template. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6878–6892, Toronto, Canada. Association for Computational Linguistics.

[Lichtarge et al.2019] Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3291–3301, Minneapolis, Minnesota. Association for Computational Linguistics.

[Liu et al.2019] Cao Liu, Shizhu He, Kang Liu, and Jun Zhao. 2019. Vocabulary pyramid network: Multi-pass encoding and decoding with multi-level vocabularies for response generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3774–3783, Florence, Italy. Association for Computational Linguistics.

[Liu et al.2021] Xuebo Liu, Longyue Wang, Derek F. Wong, Liang Ding, Lidia S. Chao, and Zhaopeng Tu. 2021. Understanding and improving encoder layer fusion in sequence-to-sequence learning.

[Ma et al.2022] Shirong Ma, Yinghui Li, Rongyi Sun, Qingyu Zhou, Shulin Huang, Ding Zhang, Li Yangning, Ruiyang Liu, Zhongli Li, Yunbo Cao, Haitao Zheng, and Ying Shen. 2022. Linguistic rules-based corpus generation for native Chinese grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 576–589, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

[Mahmood et al.2017] Rehan Mahmood, Zulin Wang, and Qin Huang. 2017. Multi-pass decoding for the robust transmission of deep-space images. In *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, pages 1–5.

[Malmi et al.2019] Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. 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 5054–5065, Hong Kong, China. Association for Computational Linguistics.

[Mita and Yanaka2021] Masato Mita and Hitomi Yanaka. 2021. Do grammatical error correction models realize grammatical generalization? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4554–4561, Online. Association for Computational Linguistics.

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

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

[Omelianchuk et al.2020] Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr

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

[Pal et al.2020] Santanu Pal, Hongfei Xu, Nico Herbig, Sudip Kumar Naskar, Antonio Krüger, and Josef van Genabith. 2020. The transference architecture for automatic post-editing. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5963–5974, Barcelona, Spain (Online). International Committee on Computational Linguistics.

[Parnow et al.2021] Kevin Parnow, Zuchao Li, and Hai Zhao. 2021. Grammatical error correction as GAN-like sequence labeling. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3284–3290, Online. Association for Computational Linguistics.

[Qiu and Qu2019] Zhaoquan Qiu and Youli Qu. 2019. A two-stage model for chinese grammatical error correction. *IEEE Access*, 7:146772–146777.

[Raffel et al.2020] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

[Rothe et al.2021] Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.

[Stahlberg and Kumar2020] Felix Stahlberg and Shankar Kumar. 2020. Seq2Edits: Sequence transduction using span-level edit operations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5147–5159, Online. Association for Computational Linguistics.

[Sun et al.2021] Xin Sun, Tao Ge, Furu Wei, and Houfeng Wang. 2021. Instantaneous grammatical error correction with shallow aggressive decoding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5937–5947, Online. Association for Computational Linguistics.

[Sun and Wang2022] Xin Sun and Houfeng Wang. 2022. Adjusting the precision-recall trade-off with align-and-predict decoding for grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 686–693, Dublin, Ireland. Association for Computational Linguistics.

[Tan et al.2023] Minghuan Tan, Min Yang, and Ruifeng Xu. 2023. Focal training and tagger decouple for grammatical error correction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5978–5985, Toronto, Canada. Association for Computational Linguistics.

[Tang et al.2021] Zecheng Tang, Yixin Ji, Yibo Zhao, and Junhui Li. 2021. Chinese grammatical error correction enhanced by data augmentation from word and character levels. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 813–824, Huhhot, China. Chinese Information Processing Society of China.

[Tarnavskyi et al.2022] Maksym Tarnavskyi, Artem Chernodub, and Kostiantyn Omelianchuk. 2022. Ensembling and knowledge distilling of large sequence taggers for grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3842–3852, Dublin, Ireland. Association for Computational Linguistics.

[Touvron et al.2023] Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

[Wang et al.2021] Yu Wang, Yuelin Wang, Kai Dang, Jie Liu, and Zhuo Liu. 2021. A comprehensive survey of grammatical error correction. *ACM Trans. Intell. Syst. Technol.*, 12(5).

[Wu and Wu2022] Xiuyu Wu and Yunfang Wu. 2022. From spelling to grammar: A new framework for Chinese grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 889–902, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

[Xia et al.2017] Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2017. Deliberation networks: Sequence generation beyond one-pass decoding. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[Xu and Liu2019] Hongfei Xu and Qiuhui Liu. 2019. Neutron: An Implementation of the Transformer Translation Model and its Variants. *arXiv preprint* arXiv:1903.07402.

[Xu et al.2022] Lvxiaowei Xu, Jianwang Wu, Jiawei Peng, Jiayu Fu, and Ming Cai. 2022. FCGEC: Fine-grained corpus for Chinese grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1900–1918, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

[Yakovlev et al.2023] Konstantin Yakovlev, Alexander Podolskiy, Andrey Bout, Sergey Nikolenko, and Irina Piontkovskaya. 2023. GEC-DePenD: Non-autoregressive grammatical error correction with decoupled permutation and decoding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1546–1558, Toronto, Canada. Association for Computational Linguistics.

[Yang et al.2023a] Ai Ming Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Hai Zhao, Hang Xu, Hao-Lun Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, Juntao Dai, Kuncheng Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Pei Guo, Ruiyang Sun, Zhang Tao, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yan-Bin Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. 2023a. Baichuan 2: Open large-scale language models. *ArXiv*, abs/2309.10305.

[Yang and Quan2024] Haihui Yang and Xiaojun Quan. 2024. Alirector: Alignment-enhanced Chinese grammatical error corrector. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 2531–2546, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

[Yang et al.2023b] Lingyu Yang, Hongjia Li, Lei Li, Chengyin Xu, Shutao Xia, and Chun Yuan. 2023b. LET: Leveraging error type information for grammatical error correction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5986–5998, Toronto, Canada. Association for Computational Linguistics.

[Yang et al.2019] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

[Yasunaga et al.2021] Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2021. LM-critic: Language models for unsupervised grammatical error correction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7752–7763, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

[Yuan et al.2021] Zheng Yuan, Shiva Taslimipoor, Christopher Davis, and Christopher Bryant. 2021. Multi-class grammatical error detection for correction: A tale of two systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8722–8736, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

[Zhang et al.2018] Xiangwen Zhang, Jinsong Su, Yue Qin, Yang Liu, Rongrong Ji, and Hongji Wang. 2018. Asynchronous bidirectional decoding for neural machine translation. *AAAI'18/IAAI'18/EAAI'18*. AAAI Press.

[Zhang et al.2023] Ying Zhang, Hidetaka Kamigaito, and Manabu Okumura. 2023. Bidirectional transformer reranker for grammatical error correction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3801–3825, Toronto, Canada. Association for Computational Linguistics.

[Zhang et al.2022a] Yue Zhang, Zhenghua Li, Zuyi Bao, Jiacheng Li, Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022a. MuCGEC: a multi-reference multi-source evaluation dataset for Chinese grammatical error correction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3118–3130, Seattle, United States. Association for Computational Linguistics.

[Zhang et al.2022b] Yue Zhang, Bo Zhang, Zhenghua Li, Zuyi Bao, Chen Li, and Min Zhang. 2022b. SynGEC: Syntax-enhanced grammatical error correction with a tailored GEC-oriented parser. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2518–2531, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

[Zhao et al.2019] Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 156–165, Minneapolis, Minnesota. Association for Computational Linguistics.

[Zhao et al.2018a] Yuanyuan Zhao, Nan Jiang, Weiwei Sun, and Xiaojun Wan. 2018a. Overview of the NLPCC 2018 shared task: Grammatical error correction. In *Natural Language Processing and Chinese Computing - 7th CCF International Conference, NLPCC 2018, Hohhot, China, August 26-30, 2018, Proceedings, Part II*, volume 11109 of Lecture Notes in Computer Science, pages 439–445. Springer.

[Zhao et al.2018b] Yuanyuan Zhao, Nan Jiang, Weiwei Sun, and Xiaojun Wan. 2018b. Overview of the NLPCC 2018 Shared Task: Grammatical Error Correction: 7th CCF International Conference, NLPCC 2018, Hohhot, China, August 26–30, 2018, Proceedings, Part II, pages 439–445.

[Zhao and Wang2020] Zewei Zhao and Houfeng Wang. 2020. MaskGEC: Improving neural grammatical error correction via dynamic masking. In *AAAI Conference on Artificial Intelligence*.

[Zhou et al.2023] Houquan Zhou, Yumeng Liu, Zhenghua Li, Min Zhang, Bo Zhang, Chen Li, Ji Zhang, and Fei Huang. 2023. Improving Seq2Seq grammatical error correction via decoding interventions. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7393–7405, Singapore. Association for Computational Linguistics.

Table 1: Main results. “*” and “†” denote our replication and using additional datasets respectively. BART (12-2) means the BART model with 12/2 encoder/decoder layers.

Method	CoNLL 2014 (test)			BEA 2019 (test)		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$
LLaMa 2-7B (zero-shot) [Touvron et al.2023]	27.41	42.24	29.48	45.85	53.58	47.21
LLaMa 2-7B (fine-tune) [Touvron et al.2023]	65.64	41.81	58.92	66.28	49.15	61.96
Seq2edit						
PIE [Awasthi et al.2019]	66.1	43.0	59.7	–	–	–
Lichtarge et al. [Lichtarge et al.2019]	66.7	40.6	59.8	–	–	–
Kiyono et al. [Kiyono et al.2019]	72.4	46.1	65.0	65.5	59.4	64.2
Kaneko et al. [Kaneko et al.2020]	72.6	46.4	65.2	72.3	61.4	69.8
ERRANT tags [Stahlberg and Kumar2020]	63.0	45.6	58.6	68.8	63.4	67.7
GECToR [Omelianchuk et al.2020]	77.5	40.1	65.3	79.2	53.9	72.4
Yuan et al. [Yuan et al.2021]	60.4	39.0	54.4	60.8	50.8	58.5
GST [Parnow et al.2021]	78.4	39.9	65.7	79.4	54.5	72.8
Tarnavskyi et al. [Tarnavskyi et al.2022]	76.1	41.6	65.3	80.70	53.39	73.21
Lai et al. [Lai et al.2022]	70.73	43.88	63.01	81.33	51.55	72.91
LET [Yang et al.2023b]†	61.2	40.9	55.6	61.8	52.1	59.5
Seq2seq						
Zhao et al. [Zhao et al.2019]	71.6	38.7	61.2	–	–	–
T5 large [Rothe et al.2021]	–	–	66.1	–	–	72.06
BIFI [Yasunaga et al.2021]†	78.0	40.6	65.8	79.4	55.0	72.9
SynGEC [Zhang et al.2022b]	74.7	49.0	67.6	75.1	65.5	72.9
BART (12-2) [Yakovlev et al.2023]	69.2	49.8	64.2	68.3	57.1	65.6
AMR-GEC [Cao and Zhao2023]	70.3	48.2	64.4	73.5	55.9	69.1
BTR [Zhang et al.2023]	71.62	48.74	65.47	74.68	60.27	71.27
Cao et al. [Cao et al.2023a]†	65.10	32.29	54.11	65.10	32.29	54.11
GEC-DePenD [Yakovlev et al.2023]	73.2	37.8	61.6	72.9	53.2	67.9
TemplateGEC [Li et al.2023]	74.8	50.0	68.1	76.8	64.8	74.1
TransGEC [Fang et al.2023b]†	74.7	51.6	68.6	–	–	–
Multimodal-GEC [Fang et al.2023a]†	75.0	53.2	69.3	77.1	66.7	74.8
unsupervised GEC [Cao et al.2023b]†	75.0	53.8	69.6	78.8	68.5	76.5
BART (12-2)*	72.56	44.73	64.53	69.62	63.56	68.32
+ MPD	73.70	47.39	66.33	72.98	65.35	71.31
BART (12-12)*	72.04	52.55	67.06	73.14	64.65	71.27
+ MPD	74.78	51.08	68.41	75.28	65.46	73.09
T5 large*	71.73	50.44	66.14	74.25	66.54	72.57
+ MPD	74.77	50.34	68.16	77.81	66.95	75.36

Table 2: Results on the BEA-19 development set.

Method	P	R	$F_{0.5}$
BART (12-2)*	69.69	50.27	64.69
+ MPD	72.11	50.54	66.44
BART (12-12)*	71.62	49.73	65.82
+ MPD	71.86	54.20	67.46
T5 large*	71.75	51.85	65.63
+ MPD	71.69	54.33	67.38

Table 3: Results of various MPD training and inference settings. Speed is the inference speed on the BEA 2019 dev set (relative to BART 12-2 baseline).

Setting	BEA 2019 dev		CoNLL 2014 test	
	$F_{0.5}$	Speed	$F_{0.5}$	Speed
BART (12-2)	64.69	1.00x	64.53	1.00x
BART (12-4)	65.11	0.61x	65.46	0.61x
BART (12-2)* 2 (Ensemble)	65.16	0.49x	65.50	0.49x
Training				
Single-round	66.44	0.83x	66.33	0.83x
Multi-round	66.21	0.79x	66.12	0.79x
Inference				
Policy network [Geng et al.2018]	65.84	0.27x	65.71	0.27x
without C_e				
$n = 1$	66.09	0.46x	65.98	0.46x
$n = 2$	65.88	0.41x	65.72	0.41x
$n = 3$	65.98	0.38x	65.82	0.38x
with $C_e, n = 3$	66.44	0.83x	66.33	0.83x

Table 4: Results of source information fusion methods (BART 12-2 setting).

Method	BEA 2019 dev		CoNLL 2014 test	
	P	R	$F_{0.5}$	$F_{0.5}$
BART (12-2)	69.69	50.27	64.69	64.53
None	69.66	52.82	65.49	65.62
Concat	70.34	52.79	65.95	65.74
Edit	70.73	52.65	66.18	65.92
Pos	71.06	52.45	66.36	66.05
Pos+Edit	72.11	50.54	66.44	66.33

Table 5: Results on the NLPCC 2018 test set.

Method	P	R	$F_{0.5}$
LLMs (zero-shot)			
LLaMa2-7B [Touvron et al.2023]	11.79	11.46	11.72
BaiChuan-7B [Yang et al.2023a]	20.87	23.28	21.31
LLMs (fine-tune)			
LLaMa2-7B [Touvron et al.2023]	45.85	27.44	40.43
BaiChuan-7B [Yang et al.2023a]	51.69	27.92	44.17
Seq2edit			
BERT-base-Chinese [Devlin et al.2019]	41.38	24.55	36.39
HRG [Hinson et al.2020]	36.79	27.82	34.56
SG-GEC [Wu and Wu2022]	50.56	25.24	42.11
Seq2seq			
AliGM [Zhao et al.2018a]	41.00	13.75	29.36
YouDao [Fu et al.2018]	35.24	18.64	29.91
BLCU [Li et al.2019]	47.63	12.56	30.57
[Qiu and Qu2019]	36.88	18.94	31.01
MaskGEC [Zhao and Wang2020]	44.36	22.18	36.97
GPT2-Chinese [Du2019]	41.94	36.13	40.63
WCDA [Tang et al.2021]	47.29	23.89	39.49
Copy [Zhao et al.2019]	51.25	32.55	45.97
SynGEC [Zhang et al.2022b]	49.96	33.04	45.32
TemplateGEC [Li et al.2023]	54.5	27.4	45.5
unsupervised GEC [Cao et al.2023b]	57.1	28.9	47.8
Alirector [Yang and Quan2024]	51.76	33.49	46.67
Ours			
Transformer	42.37	23.49	36.50
+ MPD	46.64	24.08	39.28
BART	50.63	31.83	45.28
+ MPD	52.56	33.89	47.34

Table 6: Results on the FCGEC development set and NaCGEC test set.

Method	FCGEC dev			NaCGEC test
	P	R	$F_{0.5}$	$F_{0.5}$
Transformer	47.83	22.99	39.33	49.07
+ MPD	58.67	24.76	46.06	54.06
BART	56.26	40.71	52.27	58.64
+ MPD	59.21	41.57	54.58	62.09