Leveraging Denoised Abstract Meaning Representation for Grammatical Error Correction

Hejing Cao[1,2], **Dongyan Zhao**[1,2]*
[1] Wangxuan Institute of Computer Technology, Peking University
[2] Center for Data Science, Peking University
{caohejing,zhaody}@pku.edu.cn

Abstract

Grammatical Error Correction (GEC) is the task of correcting errorful sentences into grammatically correct, semantically consistent, and coherent sentences. Popular GEC models either use large-scale synthetic corpora or use a large number of human-designed rules. The former is costly to train, while the latter requires quite a lot of human expertise. In recent years, AMR, a semantic representation framework, has been widely used by many natural language tasks due to its completeness and flexibility. A non-negligible concern is that AMRs of grammatically incorrect sentences may not be exactly reliable. In this paper, we propose the AMR-GEC, a seq-to-seq model that incorporates denoised AMR as additional knowledge. Specifically, We design a semantic aggregated GEC model and explore denoising methods to get AMRs more reliable. Experiments on the BEA-2019 shared task and the CoNLL-2014 shared task have shown that AMR-GEC performs comparably to a set of strong baselines with a large number of synthetic data. Compared with the T5 model with synthetic data, AMR-GEC can reduce the training time by 32% while inference time is comparable. To the best of our knowledge, we are the first to incorporate AMR for grammatical error correction.

1 Introduction

Nowadays, high performance of grammatical error correction model mainly depends on data augmentation (Kiyono et al., 2019; Grundkiewicz et al., 2019; Raffel et al., 2020; Wan and Wan, 2021; Wu and Wu, 2022; Zhang et al., 2022). According to the type of additional information, grammatical error correction models can be divided into data-enhanced models and knowledge-enhanced models. Data-enhanced models require millions of synthetic data, which is obtained by back-translation or directly adding noise. Training on these synthetic datasets is very time-consuming, which is unacceptable in some application scenarios. Knowledge-enhanced model is to artificially design a large number of grammatical rule templates, and add the templates as external knowledge to GEC model. This external knowledge is language-dependent and it requires the intervention of human grammar experts.

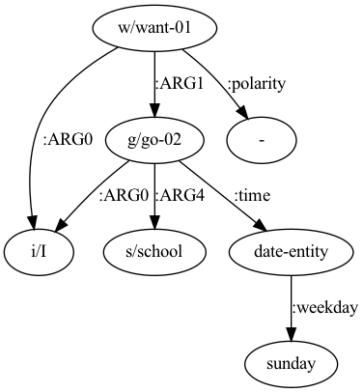

Figure 1: AMR of sentence "*I don't want to go to school on Sunday.*"

Abstract Meaning Representation (AMR) is a type of rooted, labeled graph which contains semantic structures with fine-grained node and edge types. AMR breaks through the limitations of the traditional syntax tree structure and supports reentrancy. Figure 1 is a graph of sentence "*I don't want to go to school on Sunday.*". In AMR, *:arg0* is typically the agent, *:arg1* is typically the patient, and other arguments do not have standard definitions and may vary with the verb being annotated. Negative meaning is denoted as "-". Special keywords such as entity types, quantities and logical conjunctions are supported by AMR. AMR obtains a simple representation from natural language sentence and it is suitable for GEC as extra knowledge.

A non-negligible concern is that AMRs of errorful sentences may not be exactly reliable. If these AMRs with errors are directly introduced

* Corresponding author: Dongyan Zhao.

Findings of the Association for Computational Linguistics: ACL 2023, pages 7180–7188

into the GEC model as additional information, it may confuse the model. We use a pre-trained AMR parser to predict AMR of erroneous sentences and corrected sentences separately on the BEA-19 development set. If two AMRs are completely consistent, we assume that the AMR of errorful sentences is reliable. After statistical analysis, we found that about half of the graphs are reliable.

We designed a denoising semantic aggregated grammatical error correction model. Specifically, we added a graph aggregation encoder based on a sequence-to-sequence model. The graph encoder aims to update the representation of the sequence encoder by AMR semantic structure. Besides, we designed two mask strategies to reduce the dependence on the model graph information. We designed these mask strategies by granularity: node/edge level mask and subgraph level mask. Experiments have proved that the denoising semantic aggregated grammatical error correction model significantly improved the error correction accuracy.

2 Related works

Data-enhanced GEC models. Lots of works have found their way to incorporating additional data into GEC model. Kaneko et al. (2020) uses a pre-trained mask language model in grammatical error correction by using the output of BERT as additional features in the GEC model. Kiyono et al. (2019) and Grundkiewicz et al. (2019) explore methods of how to generate and use the synthetic data and make use of Gigaword to construct hundreds of millions of parallel sentence pairs. Some works (Katsumata and Komachi, 2020, Pajak and Gonczarek, 2021, Rothe et al., 2021) give a strong baseline by finetuning BART (Lewis et al., 2020), T5 (Raffel et al., 2020) on a GEC corpus. Malmi et al. (2019) casts GEC as a text editing task. Zhao et al. (2019) and Panthaplackel et al. (2021) propose a copy-augmented architecture for the GEC task by copying the unchanged words and spans.

Knowledge-enhanced GEC models. Wan and Wan (2021) use dependency tree as syntactic knowledge to guide the GEC model. Wu and Wu (2022) adds part-of-speech features and semantic class features to enhance the GEC model. Omelianchuk et al. (2020) design thousands of custom token-level transformations to map input tokens to target corrections. Lai et al. (2022) proposes a multi-stage error correction model based on the previous model.

Applications of AMR. Song et al. (2019) and Li and Flanigan (2022) incorporate AMR in neural machine translation. Bonial et al. (2020) makes use of AMR by abstracting the propositional content of an utterance in dialogue. Xu et al. (2021) constructs a dynamic semantic graph employing AMR to cope with Multi-hop QA problems.

3 Model

We add a graph encoder based on Transformer to aggregate denoised semantic information. The architecture of AMR-GEC is shown on Figure 2.

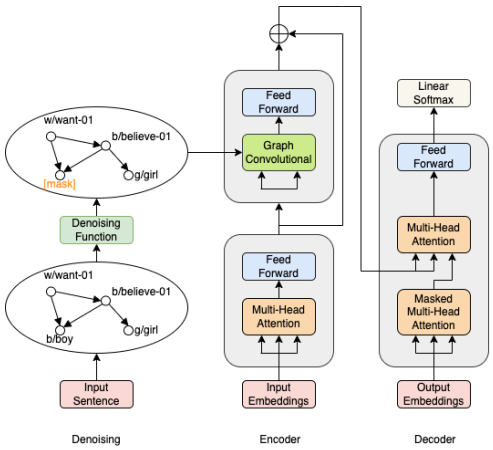

Figure 2: Denoising Semantic Aggregated GEC Model

3.1 Semantic Aggregated Encoder

Transformer is an attention-based encoder-decoder model, where the encoder encodes the input sentence into a context vector, and the decoder converts the context vector into an output sentence. Formally, we denote the tokens of the sentence is $T_n = \{t_1, t_2, ..., t_n\}$. Vinilla encoder-decoder model works as follows:

$$h_1, h_2, ..., h_n = \text{Enc}(t_1, t_2, ..., t_n) \quad (1)$$

$$y_1, y_2, ..., y_m = \text{Dec}(h_1, h_2, ..., h_n) \quad (2)$$

We then designed a semantic graph encoder based on a graph attention network to incorporate semantic graph information. To preserve the information of the sequence encoder, we use a residual connection to combine the outputs of two encoders.

$$\hat{y}_1, \hat{y}_2, ..., \hat{y}_m = \text{GNN}(h_1, h_2, ..., h_n) \quad (3)$$

$$y'_i = y_i \oplus \hat{y}_i, \quad i = 1, 2, ..., m \quad (4)$$

3.2 Denoising Function

Masked Language Modeling (MLM) is a classic pre-trained model modeling method. The task of

MLM is to mask some tokens with a special token `mask` and train the model to recover them. This allows the model to handle both the left and right context of the masked token. MLM can divided into five types: single word masking, phrase making, random span masking, entity masking, whole word masking.

Referring to Bai et al. (2022), we use the mask strategy on AMR. We used two ways to add masks: node/edge level mask and sub-graph level mask. Node/edge level mask refers to mapping the nodes/edges in the AMR graph using a noise function to generate a graph with noise. Sub-graph level mask means randomly removing subgraphs and replacing them with a mask label.

3.3 Sequence-AMR Graph Construction

In this section, we will show details about the graph encoder module. To preserve sequence information, we design a graph that fuses sequence and AMR. We first use the alignment tool JAMR to get the mapping from AMR node to sequence token. First connect the sequences through the special labels forward-label and backward-label respectively, and then map the edges of AMR to the sequence-AMR graph.

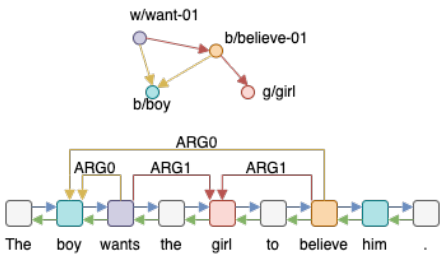

Figure 3: sequence-AMR graph

Algorithm 1 Graph Construction

Require: AMR, sequence (x_1,x_2,...,x_n), Aligner
Ensure: sequence-AMR graph

```
amr2seq = Aligner(sequence, AMR)
graph= new Graph()
for i=1 to n-1 do
    AddEdge(x_i, x_{i+1}, label-forward)
    AddEdge(x_{i+1}, x_i, label-backward)
end for
for edge in AMR.edges() do
    AddEdge(amr2seq[s], amr2seq[t], label)
end for
return graph
```

4 Experiments

4.1 Dataset

CoNLL-2014. The CoNLL-2014 shared task test set contains 1,312 English sentences with error annotations by 2 expert annotators. Models are evaluated with M2 scorer (Dahlmeier and Ng, 2012) which computes a span-based $F_{0.5}$-score.
BEA-2019. The BEA-2019 test set consists of 4477 sentences and the outputs are scored via ERRANT toolkit (Felice et al., 2016, Bryant et al., 2017). The released data are collected from Write & Improve and LOCNESS dataset.

4.2 Baseline Model

Following Rothe et al. (2021), we use T5 as the baseline model for GEC.

4.3 AMR Parsing and Alignment

We adopt SPRING (Bevilacqua et al., 2021) as our AMR parsing model. SPRING performs nearly state-of-the-art AMR parsing by linearizing AMR to sequence and converting text-to-amr task to seq-to-seq task. It obtained 84.5 Smatch F1 points on AMR 2.0 dataset.We use JAMR (Flanigan et al., 2014) to align the AMRs to sentences. JAMR is an alignment-based AMR parsing model that finds a maximum spanning, connected subgraph as an optimization problem. We use the alignment for graph information aggregation.

4.4 Others

Our models were trained on a single GPU (GeForce GTX 1080), and our implementation was based on publicly available code[1]. we set the batch_size to 6 and the learning_rate to 2e-5. We use pytorch_geometric[2] to implement the semantic aggregated encoder.

5 Results and Analysis

5.1 Results

Table 1 shows the results of the BEA-test and CoNLL-2014 dataset. 1) Compared with the model without synthetic data, the single model of AMR-GEC is 2.8 points and 1.8 points higher in BEA-19 and CoNLL-14, respectively. Ensemble models give similar results. 2) Compared with models using synthetic data, AMR-GEC gives com-

[1] https://github.com/huggingface/transformers
[2] https://github.com/pyg-team/pytorch_geometric

Models	Synthetic data	BEA-test			CoNLL-14		
		P	R	$F_{0.5}$	P	R	$F_{0.5}$
Katsumata and Komachi (2020)	-	68.3	57.1	65.6	69.3	45.0	62.6
Kiyono et al. (2019)	✓	69.5	59.4	64.2	67.9	44.1	61.3
Kaneko et al. (2020)	✓	67.1	**61.0**	65.6	69.2	45.6	62.6
Rothe et al. (2021)	✓	-	-	67.1	-	-	65.1
Omelianchuk et al. (2020)	✓	**79.2**	53.9	**72.4**	**77.5**	40.1	**65.3**
AMR-GEC	-	71.5	58.3	68.4	70.2	**48.3**	64.4
Katsumata and Komachi (2020)	-	68.8	57.1	66.1	69.9	45.1	63.0
Kiyono et al. (2019)	✓	74.7	56.7	70.2	67.3	44.0	**67.9**
Omelianchuk et al. (2020)	✓	**79.4**	**57.2**	**73.7**	**78.2**	41.5	66.5
AMR-GEC	-	73.5	55.9	69.1	70.3	**48.2**	64.4

Table 1: Results of AMR-GEC. The first group shows the results of single models. The second group shows the results of ensemble models. The ERRANT for BEA-test and the M^2 score for CoNLL-14 (test) are reported. we simply rerank outputs by generation probabilities of single models.

parable or even higher F-score, except for GECToR (Omelianchuk et al., 2020), which uses both synthetic data and human knowledge. For example, our single model achieves 68.4 on BEA-19, higher than the models by Kiyono et al. (2019), Kaneko et al. (2020), and Rothe et al. (2021). This shows that semantic graphs, as additional knowledge for GEC, have a comparative advantage over synthetic data. Our ensemble model does not show significant improvements over the single model, probably because more optimal ensemble strategies are needed: averaging generation probabilities (Omelianchuk et al., 2020), ensemble editings (Pajak and Gonczarek, 2021), etc.

5.2 Advantages of AMR

Error Type	T5-GEC			AMR-GEC		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$
PUNCT	79.8	49.4	71.0	78.7	72.9	**77.4**
DET	78.6	64.8	75.4	78.6	65.8	**75.7**
PREP	72.9	48.0	66.0	73.1	61.5	**70.4**
ORTH	84.6	55.7	**76.7**	69.5	62.9	68.1
SPELL	83.0	58.3	**76.5**	80.9	61.9	76.2

Table 2: BEA-test scores for the top five error types, except for OTHER

We compared the most common error types in BEA-test (except for OTHER) between T5-GEC and AMR-GEC. As shown in Table 2, the F-scores of PUNCT and PREP in AMR-GEC is 4-6 points higher than T5-GEC. AMR dropped prepositions, tense, and punctuation to obtain simple and base meanings, and exactly these error types are the most common errors in GEC scenarios. With such error ignored in AMR, sentences generated from AMR are more likely to get correct results.

Besides, graphs are good at solving the long sentence dependency problem. The pain point of the sequence model is that it is difficult to pay attention to long-distance dependent information. In AMR, associative concept nodes are explicitly connected with edges, making it easier for the model to focus on long-distance information.

6 Ablation Study

6.1 Graph Neural Networks Ablation Results

Graph neural networks have been proven effective in dealing with unstructured data problems. However, few studies have analyzed the effect of different GNN-encoded AMRs for natural language generation tasks. To study the differences of graph neural networks of encoding AMR, we carry on a set of experiments. We select different graph encoders of GCN, GAT, and DeepGCN as variables, and conduct experiments on BEA-2019 dataset while ensuring the same amount of model parameters. We do not use the denoising method in this ablation study.

Model	P	R	$F_{0.5}$
T5-GEC	71.47	53.46	66.96
AMR-GCN	**72.95**	52.17	**67.57**
AMR-GAT	68.26	**63.41**	67.23
AMR-DeepGCN	66.34	62.57	65.55

Table 3: Results on BEA-test with GCN, GAT, DeepGCN as AMR encoders

Table 3 shows the results of BEA-test with different graph encoders. We can draw these conclusions:

1) Even if the AMRs of the errorful sentences are not reliable, they still benefit GEC. Compared with T5-GEC, AMR-GCN and AMR-GAT are about 0.2 and 0.4 points higher respectively. This shows that the model makes use of the semantic information and connection relationship of reliable AMR. 2) AMR-GCN gives the best performance among the three models. When picking a graph encoder, the GCN model is sufficient to encode the semantic structure information of AMR. It is worth noting that GAT and DeepGCN have high recall value and low precision. In the grammatical error correction task, precision measures the error correction result. Generally speaking, precision is more important than recall. In the grammatical error correction task, most of the errors are local errors, and the semantic information required for grammatical error correction in AMR can be captured without a deeper graph convolution model.

6.2 Denoise method ablation study

Model	P	R	$F_{0.5}$
T5-GEC	71.47	53.46	66.96
AMR-GCN	72.95	52.17	67.57
AMR-GCN (node/edge)	**73.52**	55.91	**69.14**
AMR-GCN (subgraph)	72.12	**57.60**	68.60

Table 4: Results on BEA-test with node/edge and subgraph denoising methods

Table 4 shows the results of BEA-test with node/edge and subgraph denoising methods. The node/edge level denoising strategy and the subgraph level denoising strategy increased by 1.57 and 1.03 points, respectively. Node level mask strategy performs better because the subgraph may mask too much information.

7 Conclusion

In this paper, We propose a denoising semantic aggregated grammatical error correction model, AMR-GEC, leveraging AMR as external knowledge to the GEC. We believe it gives a strong baseline for incorporating AMR in GEC.

Limitations

In this paper, we leverage AMR to the GEC model as external knowledge, and achieve a high F-score on single model. However, we do not use R2L reranking, model ensemble and other methods to ensemble single model and compare them with state-of-the-art ensemble models. Our aim is to provide a strong baseline for incorporating AMR in GEC, so it is easy to generalize AMR-GEC to ensemble models.

Ethics Statement

The training corpora including the Lang-8, NUCLE and the BEA-2019 test data and CoNLL-2014 test data used for evaluating our framework are publicly available and don't pose privacy issues. The algorithm that we propose does not introduce ethical or social bias.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments. We would like to express appreciation to Yansong Feng for his insightful suggestions on the algorithm framework. This work was supported by the National Key Research and Development Program of China (No. 2020AAA0106600).

References

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for AMR parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6001–6015, Dublin, Ireland. Association for Computational Linguistics.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One SPRING to rule them both: Symmetric AMR semantic parsing and generation without a complex pipeline. In *Proceedings of AAAI*.

Claire Bonial, Lucia Donatelli, Mitchell Abrams, Stephanie M. Lukin, Stephen Tratz, Matthew Marge, Ron Artstein, David Traum, and Clare Voss. 2020. Dialogue-AMR: Abstract Meaning Representation for dialogue. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 684–695, Marseille, France. European Language Resources Association.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.

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.

Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 825–835, Osaka, Japan. The COLING 2016 Organizing Committee.

Jeffrey Flanigan, Sam Thomson, Jaime G Carbonell, Chris Dyer, and Noah A Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436.

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.

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.

Satoru Katsumata and Mamoru Komachi. 2020. 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, Suzhou, China. Association for Computational Linguistics.

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

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.

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.

Changmao Li and Jeffrey Flanigan. 2022. Improving neural machine translation with the Abstract Meaning Representation by combining graph and sequence transformers. In *Proceedings of the 2nd Workshop on Deep Learning on Graphs for Natural Language Processing (DLG4NLP 2022)*, pages 12–21, Seattle, Washington. Association for Computational Linguistics.

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.

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.

Krzysztof Pajak and Adam Gonczarek. 2021. Grammatical error correction with denoising autoencoder. *International Journal of Advanced Computer Science and Applications*, 12(8).

Sheena Panthaplackel, Miltiadis Allamanis, and Marc Brockschmidt. 2021. Copy that! editing sequences by copying spans. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13622–13630.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

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.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019. Semantic neural machine translation using AMR. *Transactions of the Association for Computational Linguistics*, 7:19–31.

Zhaohong Wan and Xiaojun Wan. 2021. A syntax-guided grammatical error correction model with dependency tree correction. *arXiv preprint arXiv:2111.03294*.

Xiuyu Wu and Yunfang Wu. 2022. From spelling to grammar: A new framework for chinese grammatical error correction. *arXiv preprint arXiv:2211.01625*.

Weiwen Xu, Huihui Zhang, Deng Cai, and Wai Lam. 2021. Dynamic semantic graph construction and reasoning for explainable multi-hop science question answering. *arXiv preprint arXiv:2105.11776*.

Yue Zhang, Bo Zhang, Zhenghua Li, Zuyi Bao, Chen Li, and Min Zhang. 2022. 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.

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.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
"Limitations".

☑ A2. Did you discuss any potential risks of your work?
"1 Introduction", "2 Related works".

☑ A3. Do the abstract and introduction summarize the paper's main claims?
"Abstract", "1 Introduction".

☑ A4. Have you used AI writing assistants when working on this paper?
We used Grammarly to correct the grammar of the full paper.

B ☑ Did you use or create scientific artifacts?

"4 Experiments".

☑ B1. Did you cite the creators of artifacts you used?
"4 Experiments".

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
"4 Experiments".

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
"4 Experiments", "5 Results and Analysis".

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
"4 Experiments".

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
"4 Experiments".

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
"4 Experiments".

C ☑ Did you run computational experiments?

"4 Experiments".

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
"4 Experiments".

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
"4 Experiments".

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
"5 Results and Analysis".

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
"4 Experiments".

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.