Controlled Generation with Prompt Insertion for Natural Language Explanations in Grammatical Error Correction

Masahiro Kaneko[1,2] **Naoaki Okazaki**[2]
[1]MBZUAI [2]Tokyo Institute of Technology
Masahiro.Kaneko@mbzuai.ac.ae okazaki@c.titech.ac.jp

Abstract

In Grammatical Error Correction (GEC), it is crucial to ensure the user's comprehension of a reason for correction. Existing studies have offered indirect explaination for corrections, such as tokens, examples, and clues, but do not explicitly explain the reasons. While a lot of researches have proposed to employ Large Language Models (LLMs) for generating direct natural language explanations across a range of tasks, GEC currently lacks such a method. Generating explanations for corrections necessitates the alignment of input and output tokens, identification of corrected spans, and then generation of explanations corresponding to those identified points. However, it is difficult for LLMs to achieve such a complex task to generate explanations. In this study, we introduce controlled generation with Prompt Insertion (PI), a method that enables LLMs to provide explanations for corrections in natural language. In PI, LLMs first correct the input text, and then we automatically extract the corrected spans based on the rules. Subsequently, we incorporate the spans into the prompt and insert it into the generation process, with an aim to generate explanations for the corrected spans. We also create an Explainable GEC (XGEC) dataset of correction reasons by annotating NUCLE, CoNLL2013, and CoNLL2014.Although generated texts from GPT-3.5 and ChatGPT using original prompts cannot cover all the corrected spans, our proposed method can steer LLMs to explicitly provide explanations for every corrected span. Moreover, this enhancement boosts the model's performance.

Keywords: Grammatical Error Correction, Explainability, Large Language Models

1. Introduction

Grammatical Error Correction (GEC) is the task of correcting grammatical errors in a text. In GEC, various methods have been proposed from a wide range of perspectives, including correction performance (Grundkiewicz and Junczys-Dowmunt, 2019; Chollampatt et al., 2019; Omelianchuk et al., 2020; Kaneko et al., 2020; Qorib et al., 2022), controlling (Hotate et al., 2019; Yang et al., 2022; Loem et al., 2023), diversity (Xie et al., 2018; Hotate et al., 2020; Han and Ng, 2021), and efficiency (Malmi et al., 2019; Chen et al., 2020). It is also important in GEC for the model to provide explanations that allow users to understand the reasons behind the corrections. Improving explainability leads to a better judgment of whether the correction reflects the intended result, learning of grammatical knowledge, and overall enhancement of GEC systems.

Kaneko et al. (2022b) introduced a method of presenting the retrieved examples as the basis for correction, in contrast to a method of retrieving data similar to the correction target from the training data set and using it for prediction. Fei et al. (2023) proposed a method that presents the token positions that are the basis of errors and error types, and showed that they are useful for learners. Nagata (2019) proposed the task of generating useful hints and feedback for language learning on essays written by language learners. This task does not necessarily generate a correction result or reason, because it is not intended for correction. Since these existing studies do not directly explain the reason for the correction, the user must infer the reason from the system output.

Large Language Models (LLMs) such as ChatGPT (OpenAI, 2023) and GPT-3.5 (Brown et al., 2020) have advanced language capabilities and can explain the inference reasons in natural language in various tasks (Wei et al., 2022; Wiegreffe et al., 2022; Kaneko et al., 2023b). With natural language, the model can directly explain the details of the inference reasons to the user. LLMs are also effective in GEC, achieving state-of-the-art in both unsupervised (Loem et al., 2023) and supervised settings (Kaneko and Okazaki, 2023). Explicability in GEC first requires the alignment of input and output tokens and identifies all error and correction pairs. Then, it is necessary to generate an explanation for each of the extracted pairs. However, it is hard to control the generation of to LLMs with prompts in a specified format for GEC. Fang et al. (2023) showed that ChatGPT improves performance by using natural language to generate step-by-step error detection and correction processes for each span. On the other hand, they found that it is difficult for ChatGPT to generate step-by-step according to the specified format with simple prompt instructions. Loem et al. (2023) showed that prompting did not contribute significantly to the control of correction style for GPT-3.5.

In this study, we introduce a method to explain

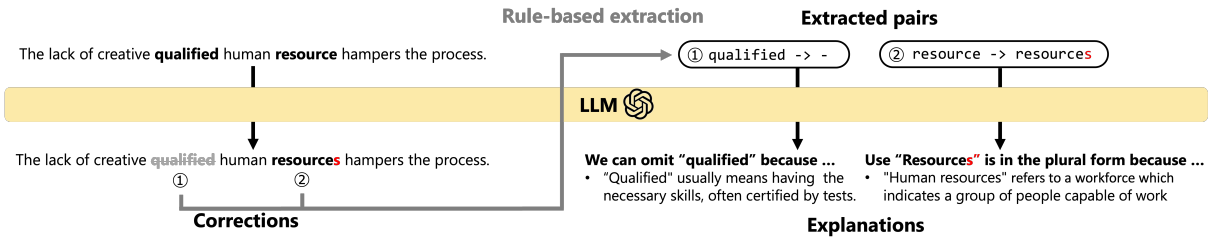

Figure 1: How to generate an explanation of the proposed method PI.

the reason for correction in natural language by a controlled generation with Prompt Insertion (PI). As shown in Figure 1, we guide LLMs to the desired format output by inserting prompts during inference. First, LLM corrects grammatical errors in the input text. Then, we automatically align the error and corrected spans from the input and output text using rules and extract error-correction pairs. By inserting these error-correction pairs as additional prompts, we explicitly control the LLM's explanation of the reasons for all pairs. Furthermore, we created an Explainable GEC (XGEC) dataset for explaining correction reasons in natural language by annotating NUCLE, CoNLL2013, and CoNLL2014 datasets (Dahlmeier et al., 2013; Ng et al., 2013, 2014).

In our experiments on GPT-3.5 and ChatGPT, we found that the original prompt-based generation resulted in pair omissions and ambiguity as to which pair the explanation was for. On the other hand, the control of generation by PI can explicitly control the LLM to generate explanations for all the corrections, which contributes to the performance improvement of the explanation of correction reasons.

2. Generate Natural Language Explanations with PI

For a GEC system to be valuable for learners, it is essential to have natural language explanations that are *precisely aligned* with *all* corrected spans. Existing methods generate explanations from the input in a single step (Wei et al., 2022; Wiegreffe et al., 2022; Chen et al., 2023; Kaneko et al., 2023b). However, addressing this complex task in such way may prove challenging. Our method tackle this challenge by inserting prompts of edits into generation process, thereby explicitly guiding the LLM to generate explanations for all corrections.

Specifically, the LLM receives the instruction to rewrite the input text (e.g. *"What is the difference between genetic disorder and other disorders ."*) into grammatically correct text (e.g. *"What is the difference between genetic disorders and other disorders ?"*) and provide explanations for the performed corrections. We compute the token alignment between the input and the corrected output text, which allows us to extract the edits, such as (*"disorder"* → *"disorders"*) or (*"."* → *"?"*). The extracted edits are consecutively provided to the LLM, prompting the LLM to generate an explanation for each edit. To make it easier to distinguish each edit, we assign numerical identifiers to edits, such as (*"1. disorder* → *disorders:"*) or (*"2. . → ?:"*).

3. Creating XGEC Dataset

The XGEC dataset includes incorrect texts, correct texts, and explanations for each edit. We annotated explanations for the original edits in existing GEC datasets. We examine the performance of LLMs with few-shot learning to create training, development, and test datasets.

We adopted NUCLE (Dahlmeier et al., 2013), CoNLL2013 (Ng et al., 2013), and CoNLL2014 (Ng et al., 2014) for XGEC dataset. NUCLE and CoNLL2013 contain only one correct text per incorrect text. We randomly selected 362 correct texts and annotated explanations for them. On the other hand, CoNLL2014 contains multiple correct texts per incorrect text because it is commonly used to evaluate GEC models. Specifically, CoNLL2014 consists of *a* and *b* datasets, which were created by different annotators. To reduce the number of test instances with low annotator agreement on the editing results, we selected edits that are regarded as appropriate by most humans. CoNNL2014 also includes additional 8 annotations (Bryant and Ng, 2015), bringing the total count to 10 annotations. We annotated explanations only for those corrected spans that exhibited an agreement of 7 or higher out of 10 within CoNLL2014 *a* and *b*, resulting in a total of 444 correct texts.

We assigned two native English speakers[1] for annotating explanations for the edits. Annotators were provided with incorrect texts, correct texts, and the corresponding edits. They were tasked with writing an explanation for each edit in a free-writing format. We provided 10 example explanations that were not included in the annotation dataset, serving as

[1]We compensated each annotator with a payment of $4 per explanation.

		XGECa			XGECb		
		Precision	Recall	F1	Precision	Recall	F1
ChatGPT	Post w/ PI	**83.2**	**85.5**	**84.3**	**83.9**	**84.5**	**84.2**
	Post w/o PI	62.1	79.6	70.0	62.6	78.2	69.6
	Pre w/o PI	60.9	75.2	68.1	61.1	74.4	67.7
GPT-3.5	Post w/ IP	**81.2**	**83.8**	**82.4**	**82.0**	**83.0**	**82.5**
	Post w/o IP	61.2	79.4	69.1	61.8	78.1	69.0
	Pre w/o IP	59.9	75.6	67.7	60.7	75.5	68.1

Table 1: The BERTScore of GPT-3.5 and ChatGPT in generating explanations with and without PI on the XGEC test datasets.

references for the annotators. For both NUCLE and CoNLL2013, we devided dataset into two parts and assigned one annotator for each. For CoNLL2014, two annotators write one explanation each. In total, we obtained 888 texts.

4. Experiment

4.1. Setting

We used the following text as the instruction: *"Correct the input text grammatically and explain the reason for each correction. If the input text is grammatically correct, only the input text should be generated as is."*. We used `text-davinci-003` for GPT-3.5 and `gpt-3.5-turbo-16k` for ChatGPT in OpenAI API[2]. The number of examples for few-shot is 16. The examples contain input texts, correct texts, edits, and explanations. We used the ERRANT (Felice et al., 2016; Bryant et al., 2017)[3] as the token alignment. We automatically evaluated the performance to generate explanations with the BERTScore (Zhang et al., 2019) of reference text and output text on CoNLL2014.

We compare our method, which generates the explanation text with PI after generating the corrected text (**Post w/ PI**), with two baselines that generate explanations without inserting edit prompts. The first baseline generates the explanation text without PI after generating the corrected text (**Post w/o PI**). We demonstrate the effectiveness of explicitly providing edits and generating explanation text through a comparison with Post w/o PI. The second baseline generates explanation text before generating corrected text (**Pre w/o PI**). We compare Pre w/o PI that generates edits and explanations step by step before generating the entire corrected text, like a chain of thought (Wei et al., 2022), with a model that generates explanations after the entire corrected text. This demonstrates the effectiveness of generating explanations after correction.[4]

		Validity	Coverage
ChatGPT	Post w/ PI	**1.5**	**2.0**
	Post w/o PI	1.2	1.4
	Pre w/o PI	1.1	0.9
GPT-3.5	Post w/ PI	**1.4**	**2.0**
	Post w/o PI	1.1	1.5
	Pre w/o PI	1.1	1.0

Table 2: Human evaluations of GPT-3.5 and ChatGPT with and without PI on the XGEC test dataset.

4.2. The Performance of Generating Explanations

Table 1 shows precision, recall, and F1 scores with BERTScore of GPT-3.5 and ChatGPT in generating explanations with and without PI on XGECa and XGECb datasets. The scores of the GPT-3.5 and ChatGPT with PI are better than the models without PI in all scores on both datasets. The performance improvement is believed to result from enhanced coverage of edits included in the explanations generated by the PI. Moreover, it can be seen from the results of Post w/o PI and Pre w/o PI that generating explanations after correction is more effective than generating them before correction.

5. Analysis

5.1. Human Evaluation

We examine the quality of LLM-generated explanations by human evaluation. We sample 200 explanations from CoNLL2013, and four human annotators evaluate those explanations from *validity* and *coverage* perspectives. The *validity* perspective refers to the accuracy and usefulness of grammatical information in LLM-generated explanations for

[2] https://platform.openai.com/docs/models/overview

[3] https://github.com/chrisjbryant/errant

[4] The proposed method cannot be applied to the process of generating explanations before correction, as it requires edits extracted from correction to generate explanations.

		CoNLL2014	W&I	JFLEG
ChatGPT	Pre Human	**55.2**	51.2	61.7
	Post Human	54.8	**51.5**	61.5
	Pre PI	54.9	51.7	61.5
	Post PI	54.7	49.7	**61.8**
	No explanation	52.3	40.1	55.3
GPT-3.5	Pre Human	54.0	**44.2**	**57.8**
	Post Human	**54.5**	44.0	57.3
	Pre PI	53.7	**44.2**	57.1
	Post PI	54.1	39.9	57.1
	No explanation	50.1	35.8	53.7

Table 3: The GEC performance of GPT-3.5 and ChatGPT when using explanation text as examples for few-shot methods.

language learners. It is scored on three levels: 0 if the explanation for more than half of corrections is incorrect and unuseful, 1 if the explanation for more than half of corrections is correct and useful but not perfect, 2 if the explanation for all corrections is perfect. The *coverage* perspective means that the LLM-generated explanation mentions all grammatical corrections. It is scored on three levels: 0 if the explanation does not cover more than half of the corrections, 1 if the explanation covers more than half of the corrections but not all corrections, 2 if the explanation covers all corrections. We evaluate the methods by averaging the annotated scores for *validity* and *coverage*, respectively.

Table 2 shows the results of validity and coverage scores from human annotators for GPT-3.5 and ChatGPT, both with and without PI. Both the validity and coverage scores for GPT-3.5 and ChatGPT using PI are better than those not using PI. The PI makes it clear to LLM the corrections that need to be explained, and allows for specific explanations tied to each correction, improving the quality of LLM's explanations. The coverage scores show that by explicitly instructing correction positions using the proposed method, LLM can generate explanations that completely cover the edits. Moreover, comparing the post-generating models and the pre-generation model demonstrates that generating an explanation before a correction has more negative effects in terms of the coverage of edits than generating an explanation after a correction.

5.2. Impact of Explanation on GEC performance

Providing explanations in addition to gold texts to the LLM as few-shot examples improves performance for tasks (Wei et al., 2022; Kaneko et al., 2023b). We evaluate a model's ability to generate explanations by assessing their impact on GEC performance using generated explanations as examples of few-shot. If the quality of the generated explanation is high, the GEC performance will improve to the same extent as with human-created explanations. Conversely, if the quality is poor, the performance will not be as good as with human-created explanations. We randomly sample 8 instances from the XGEC valid dataset to use as few-shot examples. To include more generated explanatory text for evaluation, we perform random sampling for each instance in the test data to select few-shot examples. These examples consist of human-written explanations and explanations generated by the PI, inserted both before and after the corrected text, allowing us to compare their effectiveness, respectively.

Table 3 displays the GEC performance of GPT-3.5 and ChatGPT using explanatory texts as examples for few-shot learning in the CoNLL2014, W&I, and JFLEG test datasets. Comparing the results without explanations to the results with explanations, it is evident that using explanations as examples for few-shot learning improves GEC performance. When comparing the results of human-authored explanatory text and text generated by the PI, both achieve nearly equivalent GEC performance. This suggests that the explanatory text generated by the PI is of the same quality as the explanatory text authored by humans. Furthermore, it can be observed that adding explanatory text before or after correction for few-shot learning has little influence.

6. Conclusion

In this study, we introduce a method for generating comprehensive and high-quality explanatory text in LLMs by explicitly instructing the edits. Additionally, we have created the XGEC dataset for explanatory text generation. The experimental results demonstrate that our approach, compared to methods that do not explicitly provide edits to LLMs for explanatory text generation, yields benefits in both human evaluation and automated evaluation. In future work, we plan to investigate the impact

of LLM-generated explanatory text on language learners.

7. Ethical Considerations

We paid each annotator $4 per explanation, totaling approximately $6,700 for the creation of the XGEC dataset. Therefore, we provided an adequate wage to the annotators.

While we do not foresee any ethical risks caused by our research, LLMs not only exhibit biased li kelihood based on surface-level information such as words and sentence structure but also on information like gender, religion, and race (Kaneko et al., 2022a; Zhou et al., 2022; Kaneko et al., 2023a; Anantaprayoon et al., 2023; Oba et al., 2024). For instance, LLMs might assign a higher likelihood to "She is a nurse" compared to "He is a nurse". Reducing likelihood bias could potentially address social bias in evaluators. However, it is worth noting that this study does not investigate such aspects, and this remains a task for future research.

8. Bibliographical References

Panatchakorn Anantaprayoon, Masahiro Kaneko, and Naoaki Okazaki. 2023. Evaluating gender bias of pre-trained language models in natural language inference by considering all labels. *ArXiv*, abs/2309.09697.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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.

Christopher Bryant and Hwee Tou Ng. 2015. How far are we from fully automatic high quality grammatical error correction? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 697–707, Beijing, China. Association for Computational Linguistics.

Mengyun Chen, Tao Ge, Xingxing Zhang, Furu Wei, and Ming Zhou. 2020. Improving the efficiency of grammatical error correction with erroneous span detection and correction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7162–7169, Online. Association for Computational Linguistics.

Yanda Chen, Ruiqi Zhong, Narutatsu Ri, Chen Zhao, He He, Jacob Steinhardt, Zhou Yu, and Kathleen McKeown. 2023. Do models explain themselves? counterfactual simulatability of natural language explanations. *arXiv preprint arXiv:2307.08678*.

Shamil Chollampatt, Weiqi Wang, and Hwee Tou Ng. 2019. Cross-sentence grammatical error correction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 435–445, Florence, Italy. Association for Computational Linguistics.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner English: The NUS corpus of learner English. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.

Tao Fang, Shu Yang, Kaixin Lan, Derek F Wong, Jinpeng Hu, Lidia S Chao, and Yue Zhang. 2023. Is chatgpt a highly fluent grammatical error correction system? a comprehensive evaluation. *arXiv preprint arXiv:2304.01746*.

Yuejiao Fei, Leyang Cui, Sen Yang, Wai Lam, Zhenzhong Lan, and Shuming Shi. 2023. Enhancing grammatical error correction systems with explanations. *arXiv preprint arXiv:2305.15676*.

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.

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

Wenjuan Han and Hwee Tou Ng. 2021. Diversity-driven combination for grammatical error correction. In *2021 IEEE 33rd International Conference*

on Tools with Artificial Intelligence (ICTAI), pages 972–979. IEEE.

Kengo Hotate, Masahiro Kaneko, Satoru Katsumata, and Mamoru Komachi. 2019. Controlling grammatical error correction using word edit rate. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 149–154, Florence, Italy. Association for Computational Linguistics.

Kengo Hotate, Masahiro Kaneko, and Mamoru Komachi. 2020. Generating diverse corrections with local beam search for grammatical error correction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2132–2137, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Masahiro Kaneko, Danushka Bollegala, and Naoaki Okazaki. 2023a. Comparing intrinsic gender bias evaluation measures without using human annotated examples. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2857–2863, Dubrovnik, Croatia. Association for Computational Linguistics.

Masahiro Kaneko, Aizhan Imankulova, Danushka Bollegala, and Naoaki Okazaki. 2022a. Gender bias in masked language models for multiple languages. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2740–2750, Seattle, United States. 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.

Masahiro Kaneko, Graham Neubig, and Naoaki Okazaki. 2023b. Solving nlp problems through human-system collaboration: A discussion-based approach. *arXiv preprint arXiv:2305.11789*.

Masahiro Kaneko and Naoaki Okazaki. 2023. Reducing sequence length by predicting edit operations with large language models. *arXiv preprint arXiv:2305.11862*.

Masahiro Kaneko, Sho Takase, Ayana Niwa, and Naoaki Okazaki. 2022b. 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.

Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, et al. 2022. Explanations from large language models make small reasoners better. *arXiv preprint arXiv:2210.06726*.

Mengsay Loem, Masahiro Kaneko, Sho Takase, and Naoaki Okazaki. 2023. Exploring effectiveness of gpt-3 in grammatical error correction: A study on performance and controllability in prompt-based methods. *arXiv preprint arXiv:2305.18156*.

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.

Ryo Nagata. 2019. Toward a task of feedback comment generation for writing learning. 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 3206–3215, Hong Kong, China. Association for Computational Linguistics.

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.

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.

Daisuke Oba, Masahiro Kaneko, and Danushka Bollegala. 2024. In-contextual gender bias suppression for large language models. In *Findings of the Association for Computational Linguis-*

tics: EACL 2024, pages 1722–1742, St. Julian's, Malta. 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.

OpenAI. 2023. Introducing ChatGPT. Accessed on 2023-05-10.

Muhammad Qorib, Seung-Hoon Na, and Hwee Tou Ng. 2022. Frustratingly easy system combination for grammatical error correction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1964–1974, Seattle, United States. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Sarah Wiegreffe, Jack Hessel, Swabha Swayamdipta, Mark Riedl, and Yejin Choi. 2022. Reframing human-AI collaboration for generating free-text explanations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 632–658, Seattle, United States. Association for Computational Linguistics.

Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. 2018. 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, New Orleans, Louisiana. Association for Computational Linguistics.

Liner Yang, Chengcheng Wang, Yun Chen, Yongping Du, and Erhong Yang. 2022. Controllable data synthesis method for grammatical error correction. *Frontiers of Computer Science*, 16:1–10.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Yi Zhou, Masahiro Kaneko, and Danushka Bollegala. 2022. Sense embeddings are also biased – evaluating social biases in static and contextualised sense embeddings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1924–1935, Dublin, Ireland. Association for Computational Linguistics.