arXiv:2402.15987v2 [cs.CL] 1 Mar 2024

Likelihood-based Mitigation of Evaluation Bias in Large Language Models

Masanari Ohi [1] **Masahiro Kaneko** [2,1] **Ryuto Koike** [1] **Mengsay Loem** [1] **Naoaki Okazaki** [1]
[1] Tokyo Institute of Technology [2] MBZUAI
{masanari.ohi@nlp., ryuto.koike@nlp., mengsay.loem@nlp., okazaki@}c.titech.ac.jp
masahiro.kaneko@mbzuai.ac.ae

Abstract

Large Language Models (LLMs) are widely used to evaluate natural language generation tasks as automated metrics. However, the likelihood, a measure of LLM's plausibility for a sentence, can vary due to superficial differences in sentences, such as word order and sentence structure. It is therefore possible that there might be a **likelihood bias** if LLMs are used for evaluation: they might overrate sentences with higher likelihoods while underrating those with lower likelihoods. In this paper, we investigate the presence and impact of likelihood bias in LLM-based evaluators. We also propose a method to mitigate the likelihood bias. Our method utilizes highly biased instances as few-shot examples for in-context learning. Our experiments in evaluating the data-to-text and grammatical error correction tasks reveal that several LLMs we test display a likelihood bias. Furthermore, our proposed method successfully mitigates this bias, also improving evaluation performance (in terms of correlation of models with human scores) significantly.

1 Introduction

Large Language Models (LLMs) exhibit robust language comprehension and text generation capabilities (Anil et al., 2023; OpenAI, 2023). Relying on this ability, recent studies (Liu et al., 2023; Kocmi and Federmann, 2023; Chiang and Lee, 2023) have employed LLMs as evaluators for natural language generation tasks, surpassing the performance of existing automatic evaluation methods such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004). To assess the quality of a text, LLMs output a comprehensive evaluation score based on criteria such as fluency and meaning.

LLMs generate a text based on the likelihood estimations derived from the training process that aims to maximize the likelihood of their large-scale training data. Consequently, it is intuitively possible that the likelihood of a text influences the generation of its evaluation score. However, the likelihood estimations by LLMs may not necessarily align with the quality of the text. For instance, the likelihood calculated by the LLM fluctuates due to superficial differences in sentences, such as word order and sentence structure, even for sentences with identical meaning (Kuribayashi et al., 2020). Such fluctuation of likelihood could negatively impact the evaluation score based on meaning criteria.

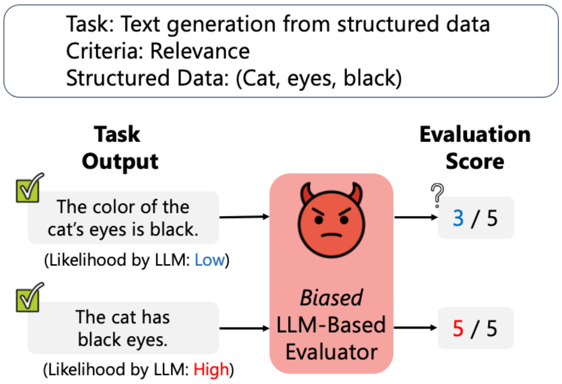


Figure 1: An example of likelihood bias. Correct, but low-likelihood output (top) is scored low while high-likelihood output (bottom) is scored high.

In this paper, we introduce **likelihood bias**, where LLM-based evaluators overrate high-likelihood sentences and underrate low-likelihood ones compared to human scores. Figure 1 shows one example of likelihood bias. Here, a biased evaluator gives a lower score of 3/5 to a correct but low-likelihood output (top) while giving a higher score of 5/5 to a high-likelihood output (bottom), based on the criteria of relevance. Addressing this issue, we propose a method that a) quantifies and b) mitigates likelihood bias. We quantify the bias by using the correlation between the disparity in evaluation scores generated by LLMs and those provided by human evaluators, and the likelihood of a target text. Our bias reduction method identifies and utilizes highly biased instances as few-shot examples for in-context learning.

The extent of likelihood bias may vary with the evaluation criteria used. For instance, likelihood bias is anticipated to be more pronounced in criteria like relevance, which is less directly related to likelihood. Conversely, the bias is expected to be less significant in criteria like fluency, which is closely related to likelihood. To verify this characteristic of likelihood bias, we adopt two tasks: data-to-text and Grammatical Error Correction (GEC). We use these tasks because, unlike most existing data (Freitag et al., 2021; Guan et al., 2021; Kamalloo et al., 2023), the evaluation data for these tasks include multiple criteria such as fluency and relevance.

Our experimental results show that both evaluators based on GPT-3.5 and Llama2-13B (Touvron et al., 2023) indeed suffer from likelihood bias. Moreover, our bias reduction method mitigates likelihood bias, and improves evaluation performance in many cases.

2 Method

Following a common methodology in LLM-based evaluation (Liu et al., 2023; Chiang and Lee, 2023), we calculate the LLM's evaluation score $\mathrm{Score_m}$ based on the models' response to a prompt. Specifically, we calculate $\mathrm{Score_m}$ as the expected value over candidate scores (e.g. $\{1, 2, 3, 4, 5\}$) based on the probability that models output these scores, following the setting of Liu et al. (2023). Our prompt includes a task description and the evaluation criteria, and several few-shot example instances for in-context learning [1]. The reason we use in-context learning is that it is known to stabilize the model. This puts us in a position to quantify the strength of likelihood bias.

2.1 Measuring Likelihood Bias

We define **likelihood bias** in LLM-based evaluators as the tendency to overrate high-likelihood sentences and underrate low-likelihood ones, compared to human ratings. First, we calculate LS, the **Likelihood Score**, representing the likelihood P calculated by LLM. Given an instance t with input t_i, output t_o, task description d, and model parameters θ, LS is defined as follows:

$$\mathrm{LS}(t) = \log P(t_o \mid t_i, d; \theta) \tag{1}$$

[1]The actual prompts and exact equation we use to calculate the $\mathrm{Score_m}$ are provided in Appendix A.

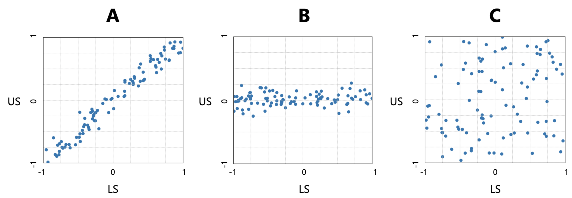


Figure 2: Likelihood bias of hypothetical evaluators. A: biased, B: unbiased with high performance, and C: unbiased with low performance.

We next calculate US, **Unfairness Score**, which represents the difference between scores by LLM ($\mathrm{Score_m}$) and scores by humans ($\mathrm{Score_h}$). To account for different scoring ranges between models and humans, $\mathrm{Score_m}$ and $\mathrm{Score_h}$ are normalized so that they have the same mean and range.

$$\mathrm{US}(t) = \mathrm{Score_m}(t;\theta) - \mathrm{Score_h}(t) \tag{2}$$

When measuring the bias, we choose few-shot example instances at random.

BiasScore is then our metric that measures likelihood bias, which is calculated as the correlation in terms of Spearman's rank correlation coefficient ρ between Likelihood Score and Unfairness Score across a Dataset $D = \{t^{(1)}, t^{(2)}, \ldots, t^{(n)}\}$, using each instance $t^{(i)}$:

$$\mathrm{LS_D} = [\mathrm{LS}(t^{(1)}), \mathrm{LS}(t^{(2)}), \ldots, \mathrm{LS}(t^{(n)})] \tag{3}$$

$$\mathrm{US_D} = [\mathrm{US}(t^{(1)}), \mathrm{US}(t^{(2)}), \ldots, \mathrm{US}(t^{(n)})] \tag{4}$$

$$\mathrm{BiasScore} = \rho(\mathrm{LS_D}, \mathrm{US_D}) \tag{5}$$

BiasScore ranges from -1 to 1, where 1 indicates strong likelihood bias, -1 implies the opposite bias from what we assume, and 0 suggests no bias.

2.2 Mitigating Likelihood Bias

Figure 2 plots LS against US in order to show the likelihood bias of multiple hypothetical evaluators. Each point represents a pair of scores for an instance. The BiasScore corresponds to the slope of the main cluster of instances.

Figure 2 (A) shows a middle-performing and biased evaluator. It unfairly gives high ratings to texts with high likelihood (points in the upper right) and low ratings to texts with low likelihood (points in the lower left). We assume that LLM-based evaluators are in this state before bias mitigation. Figure 2 (B) shows the ideal outcome of mitigation: the BiasScore is zero (i.e., there is no bias), and the performance remains high. There is also no bias in Figure 2 (C) (and thus BiasScore = 0), but this evaluator is of no use as the output is random (low-performance).

The target of our bias mitigation strategy is to change situation (A) into (B), while avoiding low evaluation performance as in (C). We concentrate on highly biased instances (top-right and bottom-left points in (A)) in our training data. For this, we require an instance-based measure of bias, which is provided by $\mathrm{RS}(t)$ as follows:

$$\mathrm{RS}(t) = |\mathrm{LS}^*(t) + \mathrm{US}^*(t)| \tag{6}$$

Here, LS^* and US^* are normalized so that they both have an average of 0 and a range from -1 to 1 across a dataset D. $\mathrm{RS}(t)$ is high for instances t that are closer to the top-right or bottom-left of the scatter plot. For our mitigation strategy, we choose instances with the highest RS(t) from the training data, and use these instances as few-shot examples for in-context learning, after replacing the LLM scores with the human gold-standard scores.

3 Experiments

3.1 Settings

Datasets From a limited set of tasks with available datasets assessed by humans on multiple criteria, we selected two for our experiments: a) data-to-text, converting RDF data into English sentences, and b) GEC. For data-to-text, we use WebNLG+ (Castro Ferreira et al., 2020), which contains 2846 instances. $\mathrm{Score}_\mathrm{h}$ is provided by human judges, who rated each instance on five criteria (text structure, relevance, fluency, correctness and data coverage). For GEC, we use the TMU-GFM-Dataset (Yoshimura et al., 2020), which contains 4221 instances. $\mathrm{Score}_\mathrm{h}$ is provided by human judges, who rated each instance on two criteria (grammar and fluency[2]). We split each dataset into training and test data at a ratio of 4:1.

Models The LLMs used in our experiments are GPT-3.5 provided via API by OpenAI [3] and Llama2-13B (L-13B). For GPT-3.5, since it does not support the output of token generation likelihood, we use Llama2-13B's likelihood as an approximation. We first measure how well the LLMs work as evaluators, using Spearman's rank correlation coefficient ρ between human and model scores. The "Before" column of Evaluation Performance in Table 1 shows these results. The ballpark figures are that GPT-3.5 is the superior system for data-to-text, while for GEC, it roughly performs on a par with Llama2-13B.

3.2 Measuring Likelihood Bias

We use the method described in Section 2.1 for likelihood bias measurement. We introduce a new criterion representing the overall result, total, by micro-averaging over the criteria[4].

Results for data-to-text The "Before" column of the "D2T" row of BiasScores in Table 1 reveals a bias for both models, with BiasScore for most evaluation criteria exceeding 0.17. Across all criteria (total), GPT-3.5 has the strongest bias (0.38), followed by Llama2-13B (0.17). Relevance is the criterion with the strongest bias in both models, GPT-3.5 (0.43) and Llama2-13B (0.28).

Results for GEC The "Before" column of the "GEC" row of BiasScores in Table 1 also shows bias in both models and evaluation criteria: all BiasScores exceed 0.16. As with data-to-text, GPT-3.5 overall displays a stronger bias across all criteria (0.43) than Llama2-13B (0.21).

Intrinsic vs non-intrinsic evaluation criteria Looking at the "Before" column of the "D2T" row of BiasScores in Table 1, there are two evaluation criteria which display relatively small likelihood biases across both models, namely fluency and text structure. These criteria are concerned with text quality alone and they are intrinsic to the output text. The criteria are true of the output text to a higher or lesser degree, but this is independent of what the input looked like. In contrast, relevance and data coverage are dependent on external factors in the input. The quality definition for those criteria is affected by the process that transforms the input into the output. Therefore, such criteria are not intrinsic. From our results, we see that there is a marked difference in BiasScore between non-intrinsic and intrinsic criteria: non-intrinsic criteria are much more prone to bias. These results suggest an intuitive interpretation: The effect of the

[2]All criteria and their definitions are given in Appendix B. The original GEC dataset contains a third criterion, meaning. However, we exclude this criterion because it does not contribute to the overall evaluation (Yoshimura et al., 2020).

[3]We use `gpt-3.5-turbo-instruct` as the model in API call.

[4]Please note that when micro-averaging, the BiasScores reported in Table 1 is not an average of the BiasScores of the individual evaluation criteria, since to calculate the total BiasScore we first average over the human and LLM evaluation scores and then apply Equation 5.

Task	Criterion	BiasScore				Evaluation Performance ρ			
		Before		After		Before		After	
		GPT-3.5	L-13B	GPT-3.5	L-13B	GPT-3.5	L-13B	GPT-3.5	L-13B
D2T	text structure	.36	.17	**.23** *	**.02** *	.46	.34	**.53** *	**.36**
	relevance	.43	.28	**.31** *	**.15** †	.35	.25	**.38**	.23
	fluency	.26	.20	.29	**.00** *	.41	.33	**.55** *	**.52** †
	correctness	.36	.21	**.32**	**-.01** *	.44	.37	**.47**	**.43**
	data coverage	.40	.24	**.32** *	.16	.20	.24	**.30** †	**.25**
	total (micro)	.38	.17	**.32** †	**.02** †	.48	.40	**.58** *	**.46**
GEC	grammar	.46	.24	**.37** †	.24	.48	.45	**.54**	**.46**
	fluency	.36	.16	**.29**	**.09**	.40	.49	**.47**	.48
	total (micro)	.43	.21	**.37**	**.18**	.45	.48	**.52**	**.52**

Table 1: BiasScore and Evaluation performance before and after mitigating likelihood bias. Values affected positively by our mitigation method appear boldfaced. * represents significant difference ($p < 0.05$) between before and after mitigation. † represents marginal significant difference ($p < 0.06$).

likelihood on the evaluation does not necessarily cause a harmful bias on intrinsic criteria as much as on non-intrinsic ones. This might be because likelihood is intrinsic to the output text, and thus, likelihood is strongly related to intrinsic criteria[5].

3.3 Mitigating Likelihood Bias

We now use the method described in Section 2.2, with eight highly biased examples for mitigation. Notably, as stated in Section 2.1, we also employ eight randomly picked examples for in-context learning when measuring bias, meaning the difference between before and after mitigation is only how we choose few-shot example instances. In the "After" columns of Table 1, we boldface the value if our method brings a BiasScore close to zero or if it improves evaluation performance. We test for the significance of differences using the two-sided randomized pair-wise permutation test with R=100000 and α=0.05. If a difference between unmitigated and mitigated conditions is significant, we indicate this with an asterisk (*); marginal significance ($p < 0.06$) is indicated using a dagger (†).

Results in data-to-text The "After" column of the "D2T" row of BiasScores and Evaluation performance in Table 1 shows that our method brings the BiasScore closer to zero and increases evaluation performance across the board. With our method, the BiasScores decrease significantly for Llama2-13B for text structure (-0.15), fluency (-0.20), and correctness (-0.20). For GPT-3.5, results are significantly decreased for text structure (-0.13), relevance (-0.12), and data coverage (-0.08). At the same time, the evaluation performance improves significantly for GPT-3.5 by +0.10 for total, by +0.14 for fluency, with marginally significant differences for GPT-3.5 in text structure, data coverage. For Llama2-13B, the only criterion with a marginally significant improvement is fluency. We consider this an overall successful mitigation.

Results for GEC As with data-to-text, the "After" column of the "GEC" row of BiasScores and Evaluation performance in Table 1 shows our method brings the BiasScore closer to zero and improves evaluation performance in many cases. Although few criteria achieve significant differences either in BiasScore or evaluation performance, our method at least shows changes in the right direction.

In summary, the results for the data-to-text and GEC tasks imply that our mitigation strategy can decrease the likelihood bias of LLMs and improve the evaluation performance simultaneously[6].

4 Conclusion

This paper identifies likelihood bias in LLMs as the phenomenon of LLMs overrating high-likelihood texts and underrating low-likelihood ones. We introduce a method for quantifying bias and propose a solution to the bias problem: using highly biased instances as few-shot examples for in-context learning. Experiments with two tasks (data-to-text and GEC) show that LLMs exhibit strong likelihood bias, and that our method successfully mitigates it, improving evaluation performance.

[5] We provide the reason we don't focus on the criteria of GEC and discuss the criterion of correctness in intrinsic/non-intrinsic paradigm in Appendix E.

[6] We conduct further experiments on visualization and case study about the mitigation of bias in Appendix F.

Limitations

Our work has several limitations. (i) Since our method uses in-context learning, the number of tokens that can be used is limited. Therefore, our method may not be suitable for tasks with long input or output lengths, such as summarization, as the amount of space that can be used is even more limited. (ii) In-context learning also brings another limitation. Since it increases the prompt length, the computational (or API call) costs also go up compared to a zero-shot setting. Again, please note that these limitations are derived from in-context learning, and our method doesn't increase prompt length and degrade efficiency compared to the settings that employ in-context learning. One solution to them is fine-tuning the model instead of in-context learning. It is therefore necessary to explore whether fine-tuning works better than in-context learning and how much data we need.

Ethics Statement

While we do not foresee any ethical risks caused by our research, LLMs not only exhibit biased likelihood based on surface-level information such as words and sentence structure but also on information like gender, religion, and race (Kaneko et al., 2023; Oba et al., 2023; Anantaprayoon et al., 2023). 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.

Acknowledgements

These research results were obtained from the commissioned research (No.22501) by National Institute of Information and Communications Technology (NICT) , Japan. Also, this work was partially supported by JSPS KAKENHI Grant Number 19H01118. Finally, We are grateful to Simone Teufel, a professor at the University of Cambridge, for her valuable advice and feedback.

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 preprint arXiv:2309.09697*.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report.

Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 2020. The 2020 bilingual, bi-directional WebNLG+ shared task: Overview and evaluation results (WebNLG+ 2020). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 55–76, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Jian Guan, Zhexin Zhang, Zhuoer Feng, Zitao Liu, Wenbiao Ding, Xiaoxi Mao, Changjie Fan, and Minlie Huang. 2021. OpenMEVA: A benchmark for evaluating open-ended story generation metrics. 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 6394–6407, Online. Association for Computational Linguistics.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5591–5606, Toronto, Canada. Association for Computational Linguistics.

Masahiro Kaneko, Danushka Bollegala, and Naoaki Okazaki. 2023. The impact of debiasing on the performance of language models in downstream tasks is underestimated. *arXiv preprint arXiv:2309.09092*.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203, Tampere, Finland. European Association for Machine Translation.

Tatsuki Kuribayashi, Takumi Ito, Jun Suzuki, and Kentaro Inui. 2020. Language models as an alternative evaluator of word order hypotheses: A case study in Japanese. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 488–504, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Daisuke Oba, Masahiro Kaneko, and Danushka Bollegala. 2023. In-contextual bias suppression for large language models. *arXiv preprint arXiv:2309.07251*.

OpenAI. 2023. Gpt-4 technical report.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem 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, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng 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.

Ryoma Yoshimura, Masahiro Kaneko, Tomoyuki Kajiwara, and Mamoru Komachi. 2020. SOME: Reference-less sub-metrics optimized for manual evaluations of grammatical error correction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6516–6522, Barcelona, Spain (Online). International Committee on Computational Linguistics.

A LLM evaluation method

Calculation of likelihood As shown in Equation 1, we calculate the likelihood of task output t_o based on task description d and task input t_i. This approach aims to obtain a more contextually relevant likelihood, factoring in both the specifics of the task and the input, rather than simply calculating $\log P(t_o; \theta)$. Specific examples of task description d are indicated below.

- data-to-text: *Please generate a description of the following xml data*
- GEC: *Please modify the following English text to make it grammatically correct*

Calculation of Score$_\mathrm{m}$ As is common in LLM-based evaluation (Liu et al., 2023; Chiang and Lee, 2023), the model is given a prompt I, which includes a task description, the evaluation criteria, and an instance t, and then predicts score Score$_m$. We also use in-context learning, with the intention of stabilizing the model. Examples are chosen at random when measuring the bias, and are chosen according to the method described in Section 2.2 when mitigating the bias. Finally, we calculate Score$_\mathrm{m}$ as the expected score over scores. We follow the setting of Liu et al. (2023), who have observed that using the expected score, considering the model's distribution over scores for each instance, rather than always taking the most likely score, leads to a more robust evaluation. Given score candidates $\{1, 2, ..., n\}$, the probability of each score $Q(i \mid t, F, I; \theta)$, Score$_\mathrm{m}$ is formulated as follows:

$$\mathrm{Score_m}(t; \theta) = \frac{\sum_{i=1}^{n} i \times Q(i \mid t, F, I; \theta)}{\sum_{j=1}^{n} Q(j \mid t, F, I; \theta)} \quad (7)$$

Example Prompts Here, we provide two examples of the prompts used for LLM-based evaluators. Our prompts are inspired by the prompts Liu et al. (2023) used.

Evaluate Correctness in data-to-text

You will be given an xml data and an English sentence that represents xml data. Your task is to rate the sentence that represents xml data on one metric. Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed. Evaluation Criteria: Correctness: (1-5) - does the text describe predicates with correct objects and does it introduce the subject correctly? 1 is the lowest score, 5 is the highest.

Evaluate Fluency in GEC

You will be given an English sentence that may have grammatical errors and a sentence that is the corrected version of the sentence. Your task is to rate the corrected sentence on one metric. Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed. Evaluation Criteria: Fluency: (0-4) - How natural the sentence sounds for native speakers; 4: Extremely natural, 3: Somewhat natural, 2: Somewhat unnatural, and 1: Extremely unnatural, and 0: Other.

B Dataset

data-to-text We use WebNLG+ Dataset (CC BY-NC-SA 4.0) (Castro Ferreira et al., 2020). Specifically, we collect instances that have human evaluation scores from their dataset. The total number of instances we use is 2846. We use them following their license. There are five criteria in the original dataset:

- text structure: whether the output is grammatically correct and well-structured
- relevance: whether the output is based on the input information
- fluency: whether the output is natural
- correctness: whether the output explains the input data correctly and reasonably
- data coverage: whether the output includes all the input data

Human annotators rate each instance on these criteria using a 100-point scale from 0 to 100.

GEC We use the TMU-GFM-Dataset (CC BY 4.0) (Yoshimura et al., 2020), which contains 4221 instances. We use them following their license. There are three criteria in the original dataset:

- grammar: whether the output is grammatically correct

- fluency: whether the output is natural
- meaning: whether the output has the same meaning as the input

Human annotators rate each instance on these criteria using a 5-point scale from 0 to 4. As mentioned in the footnote, we exclude meaning because, according to the original paper (Yoshimura et al., 2020), it does not contribute to the overall evaluation.

C Hyperparameters

To guarantee reproducibility as much as possible, we set the hyperparameters on API calls to make GPT-3.5 deterministic. We use `temperature` of 0, `top_p` of 0.

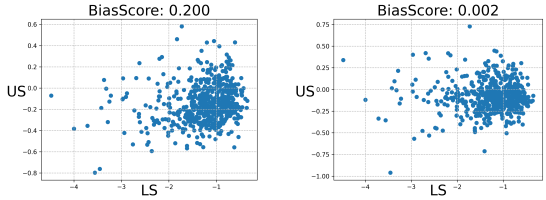
As for the number of few-shot examples for in-context learning, we use eight examples. This is the reasonable value that models can learn several pieces of information without violating the limit on the number of input tokens.

D Computational Budget

We run all the experiments on ABCI (`https://abci.ai/`), Compute Node(A), whose CPUs are two Intel Xeon Platinum 8360Y, and GPUs are eight NVIDIA A100 SXM4. The approximate total processing time is 30 hours.

E Additional Discussion on intrinsic/non-intrinsic evaluation criteria

We do not focus on the criteria of GEC within the intrinsic/non-intrinsic paradigm in Section 3.2, as its criteria, fluency and grammar, are both intrinsic. We also do not include a discussion of correctness within the paradigm because it exhibits a relatively medium level of bias, and our discussion aims to explain why certain evaluation criteria have higher or lower biases in relation to this paradigm. However, we can also explain the bias of correctness by the paradigm. Correctness has the feature of extrinsic criteria since it assesses if the output sentence explains the input correctly. At the same time, it has the feature of intrinsic criteria since it also assesses if the output sentence provides a reasonable explanation. Thus, we can explain why correctness has medium bias because it is the middle of intrinsic and extrinsic criteria.



(a) Before bias mitigation (b) After bias mitigation

Figure 3: Visualization of the bias mitigation in Llama2-13B with data-to-text, fluency

F Visualization and Case Study

Figures 3a and 3b show the visualization of likelihood bias before and after mitigation in Llama2 13B for data-to-text and fluency, respectively. We can see that our method brings BiasScore closer to zero (0.20 to 0.00), and points are gathered to the line of US = 0, similar to (B) in Figure 2. This indicates that our method successfully mitigates likelihood bias as expected.

Below, we show an instance selected from the scatterplot where the bias has been mitigated.

- Input (excerpt): (MotorSport_Vision, city, Fawkham)
- Output: The Motor sport of Vision is in Fawkham.
- The likelihood of the output: 541st out of 568 instances in our test data.
- Score by humans (Score_h): 85 / 100
- Score by LLM (Score_m) before bias mitigation: 2.46 / 5
- Score by LLM (Score_m) after bias mitigation: 4.32 / 5

Its evaluation score by humans (Score_h) is 85 out of 100, probably caused by a minor problem: the space between *Motor* and *sport*. However, LLM before bias mitigation scores the instance 2.46 out of 5, which is far from Score_h. Considering this underestimation and its low likelihood calculated by LLM (541st out of 568 instances in our test data), the score has been likely affected by likelihood bias. After bias mitigation, LLM increased the evaluation score to 4.32, which is closer to Score_h. These results indicate that our method successfully mitigates the bias in this instance, thus bringing the score by LLM closer to that of humans.