Likelihood-based Mitigation of Evaluation Bias in Large Language Models

Anonymous ACL submission

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 high-biased instances as few-shot examples for in-context learning. Our experiments in evaluating the Data2Text 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, enabled both by the large training data they have access to (Chowdhery et al., 2022; Brown et al., 2020) and by the use of instruction tuning (Wei et al., 2022; Ouyang et al., 2022). LLMs can also model the likelihood of a given sentence, as evidenced by their good natural language generation (NLG) performance. Relying on this ability, recent studies (Liu et al., 2023; Fu et al., 2023; Kocmi and Federmann, 2023; Chiang and Lee, 2023) have employed LLMs as evaluators for NLG 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, the LLMs either produce evaluation scores (Liu et al., 2023) or estimate the likelihood

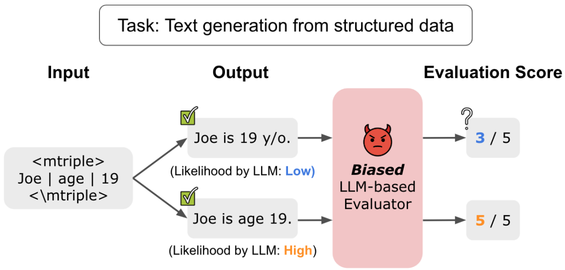


Figure 1: An example of likelihood bias. Correct, but low-likelihood output (top) is scored low while high-likelihood output (bottom) is scored high.

of generated sentences and interpret it directly as the evaluation score (Fu et al., 2023).

Consequently, the likelihood calculated by LLMs is closely linked to their role as evaluators in NLG tasks. It is intuitively possible that these likelihood estimations should somehow influence the evaluation results, even within those frameworks where LLM-based evaluators do not explicitly use likelihood as the primary metric for evaluation. However, it is known that the likelihood calculated by the LLM can fluctuate due to superficial differences in sentences, such as word order and sentence structure, even for sentences with identical meaning (Kuribayashi et al., 2020).

We hypothesize that such an inconsistency between the essential meaning of the sentence and the likelihood produced by the LLM causes a harmful bias for evaluation. We define that evaluation bias as **likelihood bias**, where LLM-based evaluators overrate the sentences with higher likelihoods (i.e., assign scores that are higher than those by humans) while underrating those sentences with lower likelihoods (i.e., assign scores that are lower than those by humans). 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).

Addressing this issue, we propose the first method that a) quantifies and b) mitigates likelihood bias. We quantify the bias by correlating the likelihood of a target text with the disparity between LLM-generated evaluation scores and those provided by human evaluators. In extensive experiments using two tasks (Data2Text and GEC, i.e., grammatical error correction), we show that both LLMs tested by us (GPT-3.5, llama2-13B (Touvron et al., 2023)) indeed suffer from likelihood bias. Our bias reduction method harvests highly-biased instances and uses them as few-shot examples for in-context learning. Our results show that apart from reducing bias, our method also improves evaluation performance in many cases: significantly so for Data2Text, and in trend also for GEC.

2 Method

We calculate the LLM's evaluation score Score_m based on the models' response to a prompt. This is a common methodology in LLM-based evaluation (Liu et al., 2023; Chiang and Lee, 2023). Our prompt includes a task description and the evaluation criteria, and several few-shot example instances for in-context learning. 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 Quantifying 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 a instance t with input t_i, output t_o, task description d, and model parameters θ, LS is defined as follows:

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

We next calculate US, **Unfairness Score**, which represents the difference between scores by LLM (Score_m) and scores by humans (Score_h). To account for different scoring ranges between models and humans, Score_m and Score_h are normalized to the same range.

$$\text{US}(t) = \text{Score}_\text{m}(t;\theta) - \text{Score}_\text{h}(t) \quad (2)$$

The Score_m is measured as the expected value over scores following the setting of Liu et al. (2023). Also, few-shot example instances are chosen at random when measuring the bias. The actual prompts

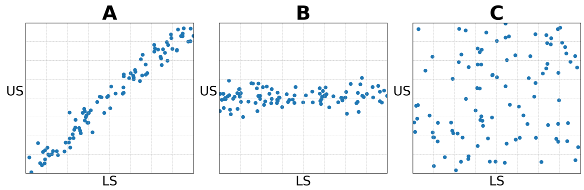


Figure 2: Likelihood bias of hypothetical evaluators. A: biased, B: unbiased with high performance, and C: unbiased with low performance.

and exact equation we use to calculate the Score_m are provided in Appendix A.

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:

$$\text{LS}_\text{D} = [\text{LS}(t_1), \text{LS}(t_2), \ldots, \text{LS}(t_n)] \quad (3)$$

$$\text{US}_\text{D} = [\text{US}(t_1), \text{US}(t_2), \ldots, \text{US}(t_n)] \quad (4)$$

$$\text{BiasScore} = \rho(\text{LS}_\text{D}, \text{US}_\text{D}) \quad (5)$$

BiasScore ranges between -1 and 1, with 1 indicating strong likelihood bias, and 0 suggesting no bias.

2.2 Mitigating Likelihood Bias

Figure 2 plots LS (Equation 1) against US (Equation 2) in order to show the likelihood bias of multiple hypothetical evaluators. Each point represents a pair of scores for a 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 Datasets

We conduct our experiments on two tasks: a) Data2Text, the task of converting RDF format data into English sentences and b) GEC. For Data2Text, 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[1]). We split each dataset into training and evaluation data at a ratio of 8:2.

3.2 Models

The LLMs used in our experiments are GPT-3.5 provided via API by OpenAI [2] and Llama2-13B (L-13B) (Touvron et al., 2023). 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 and 2 shows these results. The ballpark figures are that GPT-3.5 is the superior system for Data2Text, while for GEC, it roughly performs on a par with Llama2-13B.

3.3 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[3].

Results for Data2Text The "Before" column of BiasScores in Table 1 reveals a bias for both models and evaluation criteria, 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 BiasScores in Table 2 shows bias in both models and evaluation criteria also for the GEC task: all BiasScores exceed 0.16. As with Data2Text, 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 "Before" column 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. For instance, we cannot assess whether a piece of information is relevant by only looking at the output. The quality definition for those criteria is affected by the process that transforms the input into the output. Without looking at the input, we would miss information about the start state of the process. 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: Although LLM-based evaluators rely on

[1]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).

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

[3]Please note that when micro-averaging, the total BiasScore reported in Table 1 and 2 is not an average of the BiasScore 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.

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

Table 1: Data2Text: 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 ($p < 0.06$).

	BiasScore				Evaluation Performance ρ			
	Before		After		Before		After	
Criterion	L-13B	GPT-3.5	L-13B	GPT-3.5	L-13B	GPT-3.5	L-13B	GPT-3.5
grammar	.24	.46	.24	**.37** †	.45	.48	**.46**	**.54**
fluency	.16	.36	**.09**	**.29**	.49	.40	.48	**.47**
total (micro)	.21	.43	**.18**	**.37**	.48	.45	**.52**	**.52**

Table 2: GEC: BiasScore and Evaluation performance before and after mitigating likelihood bias. We use the notation in the same manner as Table 1.

likelihood when they score any criterion, the likelihood is a better estimator for intrinsic criteria than they are for non-intrinsic ones. This might be because, for intrinsic criteria, lots of output text is all that is required to learn it, and that is exactly what likelihood is all about.

3.4 Mitigating Likelihood Bias

We now use the method described in Section 2.2, with eight highly-biased examples for mitigation. In the "After" columns of Table 1 and 2, 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 $\alpha = 0.05$. 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 Data2Text The "After" column of BiasScores and Evaluation performance of 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 Data2Text, the "After" column of BiasScores and Evaluation performance of Table 2 shows our method brings the BiasScore closer to zero in many cases, and that evaluation performance is overall improved. 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 Data2Text and GEC tasks imply that our mitigation strategy can decrease the likelihood bias of LLMs and improve the evaluation performance simultaneously [4].

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 high-biased instances as few-shot examples for in-context learning. Experiments with two tasks (Data2Text and GEC) show that LLMs exhibit strong likelihood bias, and that our method successfully mitigates it, improving evaluation performance.

[4] We conduct further experiments on visualization and case study about the mitigation of bias in Appendix E

Limitations

Our work has several limitations. (i) Since we use in-context learning to mitigate likelihood bias, the number of tokens that can be used is limited by the method. 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. One solution 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.

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*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire.

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*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

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.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

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.

- Data2Text: *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 Data2Text

> 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

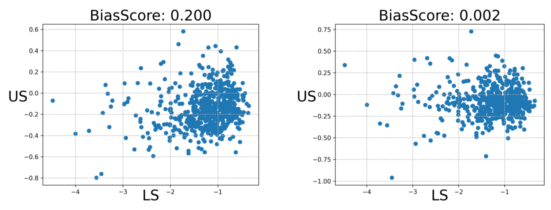
Data2Text We use WebNLG+ Dataset å(CC BY-NC-SA 4.0) (Castro Ferreira et al., 2020). Specifically, we collect instances that have human evaluation scores from their dataset. The total number of instances we use is 2846. We use them following their license. There are five criteria in the original dataset:

- 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
- 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



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

Figure 3: Visualization of the bias mitigation in Llama2-13B with Data2Text fluency

- 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 Visualization and Case Study

Figures 3a and 3b show the likelihood bias before and after mitigation in Llama2 13B for Data2Text 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 present an example of an instance where bias was mitigated and its evaluation results.

Input (excerpt):

<mtriple>MotorSport_Vision | city | Fawkham</mtriple>

Output:

The Motor sport of Vision is in Fawkham.

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

In the above example, apart from the space between `Motor` and `sport`, there are no issues, but the model rated it low before bias mitigation due to its low likelihood. However, the model rated it higher after bias mitigation, bringing it closer to the score by humans.