LLM におけるスコアリングバイアスの緩和

安里 優真　　小田 康平　　白井 清昭　　Natthawut Kertkeidkachorn
北陸先端科学技術大学院大学
{yuma-a,s2420017,kshirai,natt}@jaist.ac.jp

概要

大規模言語モデル (LLM) は，テキストの品質評価や文間類似度の推定など，スコアとして数字を出力させるために使われることも多い．本研究は，LLM が生成する数字トークンの偏り (バイアス) を緩和する 2 つの手法を提案する．ひとつは LLM にランダムに数字を生成させることで LLM が持つ潜在的なバイアスを測定し，これを基にデバイアスを行う手法，もうひとつは入力と類似した事例に対するバイアスを測定し，デバイアスを行う手法である．3 つのタスク，5 つの LLM を用いた実験の結果，提案手法は数字トークンのバイアス緩和に貢献し，既存のキャリブレーション手法に匹敵する性能を示した．

1　はじめに

大規模言語モデル (Large Language Model; LLM) は機械翻訳や要約といった様々な自然言語処理タスクで利用され，優れた性能を発揮している．LLM は流暢なテキストを生成する能力に長けているが，スコアリングのために数字を生成することを目的として使われることも多い．近年では LLM を評価器として利用する LLM as a Judge と呼ばれる活用も盛んに行われている．従来のテキスト生成の自動評価は，生成されたテキストと参照テキストの類似度を測る手法が主流であった．これに対し，LLM の評価器としての利用は，大量のサンプルに対して参照テキストを用意するための人的コストを軽減するだけでなく，参照テキストを用いた自動評価よりも適切な場合もあることが知られている [1, 2]．この際，LLM 評価器は生成テキストの品質をスコア (数字) として出力することが多い．

しかしながら，LLM が生成するテキストには様々なバイアスがあることが知られている．LLM で数字を生成させるときにもバイアスの影響は無視できない．例えば，LLM 評価器において，評価対象テキストの内容に依らず特定の数値に出力が偏ることが知られている [3, 4]．本研究では，LLM が数字をスコアとして生成する際のバイアスを「スコアリングバイアス」と呼ぶ．

スコアリングバイアスに関する研究は近年注目を集めている．Li らは，プロンプトに与える採点基準の順序をシャッフルしたり，参照解答をプロンプトに加えたりすることで，意図的にスコアリングバイアスを誘発し，既存モデルのスコアリングバイアスに対する頑健性を評価した [3]．Fujinuma は，スコア範囲を変動させると LLM 評価器の性能が変化する「スコア範囲バイアス」を指摘し，メインモデルとアシスタントモデルと呼ばれる 2 つの異なる LLM を用いた対照デコーディング [5] によってスコア範囲バイアスを相殺する手法を提案した [6]．Zhao らは，N/A などの意味を持たない入力に対するトークンの生成確率分布から分類ラベルのバイアスを測定し，それを基に推論時のラベルの生成確率を補正する Contextual Calibration と呼ばれる手法を提案した [7]．

本研究はスコアリングバイアスを緩和する 2 つのアプローチを提案する．1 つ目は数字トークンの潜在的バイアスの緩和である．LLM が潜在的に高い・低い確率で生成する数字を観察し，これを LLM によるバイアスとみなして是正する．2 つ目は類似事例に基づく数字トークンのバイアス緩和である．入力と類似した事例を LLM によって評価したときの数字トークンの生成確率を観察し，これを用いてスコアリングバイアスを緩和する．提案手法の有効性を検証するために，LLM が生成した応答を評価するタスクと文間の類似度を評価するタスクについて，バイアス緩和を行わない手法や従来のデバイアス手法をベースラインとし，提案手法と比較する．

2　提案手法

本研究で想定しているのは，LLM に 0 から N までの整数をスコアとして出力させるタスクである．以下，これを「数字生成タスク」と呼ぶ．LLM 評価

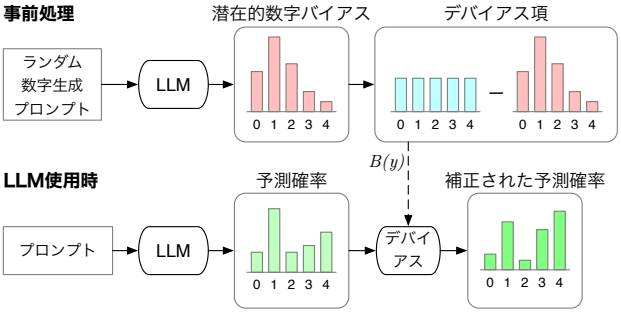

図 1: 潜在的数字デバイアス手法

器がテキストの品質を表すスコアを生成するタスクや，文間の類似度を表すスコアを生成させるタスクなどが該当する．

2.1 潜在的数字デバイアス手法

LLM には，入力に関わらず，潜在的に生成されやすい数字と生成されにくい数字があると仮定する．以下，このような数字トークンの生成確率の偏りを「潜在的数字バイアス」と定義し，これを緩和する手法を提案する．本手法の処理の流れを図 1 に示す．まず，LLM にランダムに数字を生成するプロンプトを与え，数字トークンの生成確率の分布を観察することで潜在的数字バイアスを測定する．次に，一様分布と観察した数字トークンの生成確率との差によってデバイアス項を求める．最後に，LLM を数字生成タスクに適用する際，LLM の出力 logit にデバイアス項を加えることで数字トークンの生成確率を補正する．

潜在的数字バイアスの測定　LLM の潜在的数字バイアスを測定するために，LLM にランダムに数字を生成させるプロンプトを与える．まず，GPT-4o[8] を用いてランダムな数字の生成を指示するプロンプトを 100 個生成する．生成に用いたプロンプトを付録 A の図 4 に示す．次に，数字生成タスクに用いる LLM(GPT-4o とは異なる LLM) を用いてプロンプトの perplexity を測り，それが小さい 10 個のプロンプトを選別する．

次に，10 個のプロンプトを用いて数字トークンをランダムに生成させる．数字トークンを y，プロンプト x_i を用いたときの LLM による y の生成確率を $p(y|x_i)$ としたとき，数字トークンの平均生成確率を式 (1) で算出する．こうして得られた $P_{\text{random}}(y)$ を LLM の潜在的数字バイアスとする．

$$P_{\text{random}}(y) = \frac{1}{10}\sum_{i=1}^{10}\frac{p(y|x_i)}{\sum_{y\in\{0,\cdots,N\}} p(y|x_i)} \tag{1}$$

デバイアス項の推定　ランダムに数字を生成させたとき，その生成確率分布は一様分布になるはずだが，実際はそうではない．ここでは一様分布との差をバイアスの強さとし，数字トークンの潜在的バイアスを是正するデバイアス項 $B(y)$ を式 (2) のように定義する．

$$B(y) = \log U(y) - \log P_{\text{random}}(y) \tag{2}$$

ここで $U(y)$ は一様分布 $(\forall y : U(y) = \frac{1}{N+1})$ を表す．$B(y)$ は潜在的に生成されやすい数字トークンに対して低く，生成されにくい数字トークンに対して高くなるよう設定される．

数字トークン生成時のデバイアス　LLM を数字生成タスクに適用したとき，数字トークンの生成確率を式 (3) のように補正する．

$$\tilde{P}_{\text{random}}(y) = \text{softmax}(l(y) + \lambda \cdot B(y)) \tag{3}$$

$l(y)$ は数字トークン y に対応する logit，λ は補正の強さを調整するハイパーパラメタである．後述の実験では，λ は開発データを用いて最適化する．

2.2 類似事例デバイアス手法

潜在的数字デバイアス手法では LLM が持つ潜在的なスコアリングバイアスに着目したが，バイアスがどの程度強く現れるかは入力に依存すると考えられる．すなわち，数字トークンの生成確率がバイアスの影響を強く受ける入力とそうでない入力がある．ここでは，文脈から次のトークンを予測するという LLM の特性から，意味的に類似する文脈では同じようなスコアリングバイアスが生じるという仮説に基づき，バイアスを緩和する．

図 2 に本手法の概要を示す．まず，事例データベースから入力の文脈ならびに LLM が数字トークンに対して出力する logit の分布が似ている事例を検索する．次に，類似事例に対する logit 分布の平均を求め，それを基に予測確率のデバイアスを行う．

事例データベースの構築　LLM を数字生成タスクに適用するとき，テストデータ以外の同タスクのサンプルが存在すると仮定し，これから事例データベース $E = \{(c_i, L_i)\}$ を構築する．各事例 (c_i, L_i) は文脈 c_i と logit 分布 L_i から構成される．ここで，c_i は LLM に与える入力であり，L_i はそれに対して LLM が出力する数字トークン y に対する logit $(l_i(y))$ の分布である．

類似事例の検索　新たな入力が LLM に与えられたとき，その文脈と logit 分布の組を (c, L) とし，それ

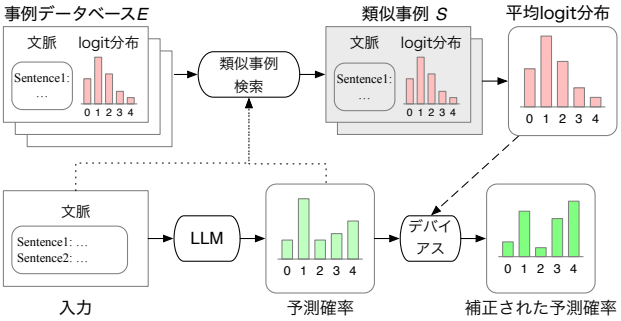

図 2: 類似事例デバイアス手法

と類似した事例を事例データベース E から検索する．LLM は類似事例に対して似たようなバイアスを持つという考えに基づき，スコアリングバイアスの緩和を試みる．ただし，入力が意味的に類似した事例でも，それに対する正解スコアが異なる場合には，LLM が予測する数字トークンの確率分布が大きく異なり，入力と同じようなバイアスが出現していない可能性がある．そのため，正解スコアが同じ事例を類似事例として取得する．具体的には，文脈に関する条件と logit 分布に関する条件の両方を満たす事例を検索し，類似事例の集合 S とする．

文脈に関する条件　文脈 c の埋め込みを $\mathbf{v}$，c_i の埋め込みを $\mathbf{v}_i$ とするとき，両者のコサイン類似度が閾値 α 以上である (式 (4))．

$$\cos(\mathbf{v}, \mathbf{v}_i) \geq \alpha \tag{4}$$

logit 分布に関する条件　類似事例の条件のひとつは正解スコアが同じであることだが，入力に対する正解スコアは不明である．そのため，logit 分布が似ているときには正解スコアも同じである可能性が高いとみなす．具体的には，L と L_i の Wasserstein 距離 W_1 を基に類似度を式 (5) のように定義し，それが閾値 β 以上であることを類似事例の条件とする．

$$sim(L, L_i) \stackrel{\text{def}}{=} 1 - \frac{W_1(L, L_i)}{Y_{max} - Y_{min}} \geq \beta \tag{5}$$

$$W_1(L, L_i) \stackrel{\text{def}}{=} \sum_{t=Y_{min}}^{Y_{max}} \left| \sum_{y=Y_{min}}^{t} l(y) - \sum_{y=Y_{min}}^{t} l_i(y) \right| \tag{6}$$

Y_{min} と Y_{max} はスコアの最小値と最大値を表す ($Y_{min} = 0$，$Y_{max} = N$)．

なお，後述の実験では，閾値 α と β は開発データを用いて最適化する．

数字トークン生成時のデバイアス　ある入力 (c, L) に対する数字トークンの生成確率は類似事例と類似するという仮定の下，数字トークン y の生成確率を式 (7) のように補正する．

$$\tilde{P}_{\text{example}}(y) = \text{softmax}(l(y) + \Delta(y)) \tag{7}$$

$$\Delta(y) = \frac{1}{|S|} \sum_{i \in S} l_i(y) \tag{8}$$

$\Delta(y)$ は類似事例集合における数字トークン y の logit の平均値であり，これを加算することで y の生成確率が類似事例のそれに近くなる．直観的には，LLM は個々の事例に対して強いバイアスを持つ可能性があるが，類似事例の平均的な logit を加算することで個別の事例に対するバイアスを緩和している．

3　評価実験

3.1　評価タスク

提案手法を 2 つのタスクで評価する．第 1 のタスクは LLM 生成テキストの評価である．テストデータとして HelpSteer2 [9] を用いる．HelpSteer2 は LLM のアライメントのためのデータセットであり，様々なプロンプトに対する LLM 生成テキストの品質が helpfulness，correctness，coherence，complexity，verbosity の 5 つの観点から 5 段階 (0〜4 の整数) で人手評価されている．本実験では LLM に helpfulness のスコアを予測させる．第 2 のタスクは文間の類似度推定である．LLM に 2 つの文の類似度を 0〜5 の整数で予測するよう指示する．テストデータとして STS-B [10] と SemRel2024 [11] の 2 つを用いる．これらのデータセットでは文のペアとそれに対する類似度が範囲 $[0, 1]$ の実数で与えられているが，本実験ではそれを範囲 $[0, 5]$ の実数に線形変換する．

3 つのデータセットの詳細を付録 B に示す．

3.2　実験設定

ベースライン　提案手法と比較するためのベースラインとして Vanilla と Contextual Calibration (CC)[7] を用いる．Vanilla はデバイアスを行わない手法である．

LLM　評価に用いた LLM と略号を以下に示す．

- Llama-3.1-8B-Instruct [12] (L3.1-8B)
- Qwen2.5-14B-Instruct [13] (Q2.5-14B)
- Prometheus-7B-v2.0 [2, 14] (P2-7B)
- Gemma-2-9B-Instruct [15] (G2-9B)
- Mistral-7B-Instruct-v0.3 [16] (M0.3-7B)

実装の詳細　類似事例デバイアス手法における

表 1: 数字生成タスクの Spearman の順位相関係数 (%)

LLM	HelpSteer2				STS-B				SemRel2024			
	Vanilla	CC	Deb-R	Deb-E	Vanilla	CC	Deb-R	Deb-E	Vanilla	CC	Deb-R	Deb-E
L3.1-8B	8.0	13.3	**16.8**	15.8	54.3	70.6†	60.0†	**60.1**†	53.3	**58.5**	51.6	51.7
Q2.5-14B	**42.9**	42.2	42.7	41.8	87.6	88.0	**88.1**	87.9	79.6	**79.9**	79.8	78.8
G2-9B	26.2	**34.8**	32.7	30.1	74.9	75.9	**76.5**	75.4	63.9	65.5†	**67.0**†	62.2
P2-7B	−2.6	7.1†	**9.15**	−4.2	70.7	**78.2**†	75.9	74.7	71.5	**73.4**	69.8	70.2
M0.3-7B	15.3	15.5	15.5	**18.2**	78.6	78.7	**79.1**	78.9	71.6	71.2	**72.5**	72.1

Vanilla はデバイアスなし，**CC** は Contextual Calibration，**Deb-R** は潜在的数字デバイアス手法，**Deb-E** は類似事例デバイアス手法を表す．太字は各タスク・各モデルにおける最良の結果を示す．† は Vanilla と有意差があることを示す ($p < 0.05$).

文脈は，LLM 生成テキスト評価タスクでは評価対象テキストを埋めた入力プロンプトそのものとし，文間の類似度推定タスクでは 2 つの文を連結したテキストとする．文脈の埋め込みを得るために `gte-base-en-v1.5` [17, 18] を使用する．

全てのタスクにおいて，zero-shot プロンプトを用いて LLM に数字を生成させる．プロンプトの例を付録 A の図 5 に示す．温度パラメタは 0.2 に設定する．最終的な出力は，確率が最大の数字トークンではなく，生成確率 $\tilde{p}(y)$ を重みとする数値の重み付き和によって算出されたスコアとする (式 (9)).

$$\text{score} = \sum_{y \in \{0,\cdots,N\}} \tilde{p}(y) \times y \tag{9}$$

評価基準　テストデータのサンプルを参照スコア (正解スコア) および予測スコアで並べたときの両者の Spearman の順位相関係数を評価基準とする．

3.3　実験結果と考察

実験結果を表 1 に示す．潜在的数字デバイアス手法 (Deb-R) は，多くの場合で Vanilla よりも高い順位相関係数が得られている．一方，類似事例デバイアス手法 (Deb-E) は，HelpSteer2 および SST-B については Vanilla よりも良い結果が得られたものの，SemRel2024 についてはその効果は限定的であった．このことは，数字生成タスクにおける LLM のバイアスは，入力の違いよりも，LLM そのものが潜在的に持つ数字の生成しやすさに起因することを示唆する．

各デバイアス手法と Vanilla の予測結果に有意差があるかを調べるため，試行回数 10,000 回の片側の並べ替え検定 (permutation test) を実施した．その結果，Deb-R では 2 つのケース，Deb-E では 1 つのケースについて有意水準 0.05 で有意差があることを確認した．

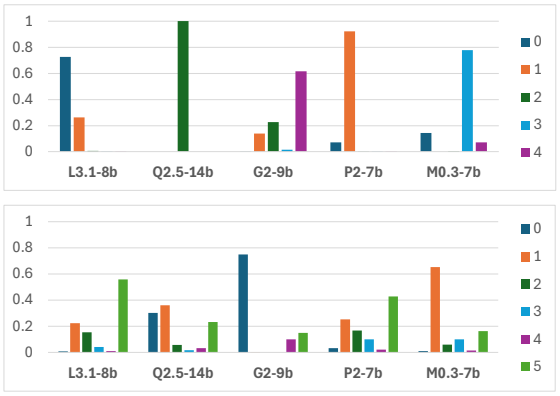

図 3: 潜在的数字バイアス (上段は数字の範囲 $[0, 4]$, 下段は $[0, 5]$)

Deb-R と CC を比較すると，タスクやモデルによって優劣は異なるが，おおむね同等のバイアス緩和効果が得られていると言える．

3.4　潜在的数字バイアスの考察

2.1 項で述べた潜在的数字デバイアス手法において，測定した潜在的数字バイアス $P_{\text{random}}(y)$ を図 3 に示す．ランダムに数字を生成するプロンプトを与えても，数字の生成確率に強い偏りが見られ，バイアスの存在が確認できる．また，LLM によってよく生成される数字は異なること，同じ LLM でも生成する数字の範囲を変えると潜在的数字バイアスが変化することも確認できる．

4　おわりに

本論文は LLM のスコアリングバイアスを緩和する 2 つの手法を提案した．今後は，few-shot プロンプトを用いる設定での実験や，パラメタ最適化のための開発データとしてラベル付きデータを必要としないデバイアス手法の探究に取り組みたい．

参考文献

[1] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2511–2522, Singapore, December 2023. Association for Computational Linguistics.

[2] Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4334–4353, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[3] Qingquan Li, Shaoyu Dou, Kailai Shao, Chao Chen, and Haixiang Hu. Evaluating scoring bias in llm-as-a-judge, 2025.

[4] 佐藤郁子, 金輝燦, 陳宙斯, 三田雅人, 小町守. アライメントが大規模言語モデルの数値バイアスに与える影響. 言語処理学会第 31 回年次大会発表論文集, pp. 2809–2814, 3 2025.

[5] Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12286–12312, Toronto, Canada, July 2023. Association for Computational Linguistics.

[6] Yoshinari Fujinuma. Contrastive decoding mitigates score range bias in llm-as-a-judge, 2025.

[7] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pp. 12697–12706. PMLR, 2021.

[8] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[9] Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. Helpsteer 2: Open-source dataset for training top-performing reward models. *Advances in Neural Information Processing Systems*, Vol. 37, pp. 1474–1501, 2024.

[10] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pp. 353–355, 2018.

[11] Nedjma Ousidhoum, Shamsuddeen Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Ahmad, Sanchit Ahuja, Alham Aji, Vladimir Araujo, Abinew Ayele, Pavan Baswani, et al. Semrel2024: A collection of semantic textual relatedness datasets for 13 languages. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2512–2530, 2024.

[12] Llama-3.1-8b-instruct. https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct (accessed 2026-01-06).

[13] Qwen Team. Qwen2.5: A party of foundation models. https://qwenlm.github.io/blog/qwen2.5/ (accessed 2026-01-06), September 2024.

[14] Prometheus-7b-v2.0. https://huggingface.co/prometheus-eval/prometheus-7b-v2.0 (accessed 2026-01-06).

[15] Gemma Team. Gemma-2-9b-instruct. https://huggingface.co/google/gemma-2-9b-it (accessed 2026-01-06).

[16] Mistral-7b-instruct-v0.3. https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3 (accessed 2026-01-06).

[17] Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. mGTE: Generalized long-context text representation and reranking models for multilingual text retrieval. In Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1393–1412, Miami, Florida, US, November 2024. Association for Computational Linguistics.

[18] gte-base-en-v1.5. https://huggingface.co/Alibaba-NLP/gte-base-en-v1.5.

A　プロンプト

ランダムに数値を生成させるプロンプトを生成させるプロンプトを図 4 に示す．{{N}}は数字の範囲 (N= 4 or 5) を示す．

```
Generate 100 prompts that have the same meaning as below the one.
Please generate a random integer between 0 and {{N}}.
```

図 4: ランダム数値生成プロンプトを生成させるメタプロンプト

データセット HelpSteer2 のサンプルに対して与えるプロンプトを図 5 に示す．STS-B，SemRel2024 のサンプルに対して与えるプロンプトは省略する．

```
Here is a task description.
Please provide a response that completes the request appropriately.

### instruction:
Evaluation Task: Assess Helpfulness

Definition:
"Overall helpfulness of the response to the prompt."

Please evaluate the response below based on this definition.

Scoring Guide:
- 4 (Extremely helpful): The response perfectly and comprehensively addresses the user's prompt.
- 3 (Helpful): The response successfully addresses the prompt but could be slightly more comprehensive.
- 2 (Moderately helpful): The response partially answers the prompt but has significant room for improvement.
- 1 (Slightly helpful): The response makes an attempt but is largely unhelpful or misses the main point.
- 0 (Not helpful): The response is completely irrelevant or fails to address the prompt.

### input:
Prompt:
{{prompt}}

Response：
{{response}}

Helpfulness (0-4):

### response:
```

図 5: HelpSteer2 のためのプロンプト

B　データセットの詳細

3 つのデータセットの統計を表 2 に示す．テストデータはデバイアス手法の評価に，開発データはハイパーパラメタの最適化に用いる．訓練データは，類似事例デバイアス手法では事例データベースとして用いるが，潜在的数字デバイアス手法では使用しない．

表 2: データセットの統計

	訓練	開発	テスト
HelpSteer2	23,324	1,038	208
SST-B	5,703	1,463	1,378
SemRel2024	5,500	250	2,600

C　アブレーション分析

2.2 項で述べた類似事例デバイアス手法において，類似事例を検索する際の条件として，文脈に関する条件ならびに logit 分布に関する条件の効果を検証するアブレーション分析を行う．フルモデルならびに各条件を除いたときの順位相関係数を表 3 に示す．Mistral (M0.3-7B) を除いて，logit 分布に関する条件を除くと順位相関係数は下がる．一方，文脈に関する条件を除くと順位相関係数が逆に向上するケースもある．両者を比較すると logit 分布に関する条件の方がより重要である．

表 3: アブレーション分析の結果

Model	**HelpSteer2**	**STS-B**	**SemRel2024**
L3.1-8B	**15.8**	**60.1**	51.7
w/o 文脈	8.16	56.7	**52.8**
w/o logit	11.9	59.5	51.6
Q2.5-14B	41.8	87.9	78.8
w/o 文脈	**42.5**	**88.1**	**79.9**
w/o logit	41.3	87.8	79.4
G2-9B	**30.1**	75.4	62.2
w/o 文脈	26.6	**75.5**	**65.0**
w/o logit	**30.1**	75.2	61.6
P2-7B	-4.2	74.7	70.2
w/o 文脈	**-1.7**	**76.8**	**72.1**
w/o logit	-4.2	74.6	70.2
M0.3-7B	**18.2**	**78.9**	**72.1**
w/o 文脈	15.6	78.6	71.5
w/o logit	**18.2**	78.8	**72.1**