有害性評価と巻き戻しによる LLM の有害コンテンツ生成回避

山下 智也[1] 岡 佑依[2] 山中 友貴[1] 山田 真徳[1]

[1] 日本電信電話株式会社 NTT 社会情報研究所
[2] 日本電信電話株式会社 NTT 人間情報研究所
{tomoya.yamashita, yui.oka, yuuki.yamanaka, masanori.yamada}@ntt.com

概要

大規模言語モデル (LLM) の普及に伴い，安全性の高い LLM への要望が高まっている．本稿では LLM の安全性の 1 要素である LLM が有害コンテンツを出力しないことに着目し，有害コンテンツを軽量かつ効果的に回避する手法を提案する．提案手法ではトークン列生成を行う生成 LLM と，トークン列の有害性を評価する評価 LLM を用いて，適宜トークン列の有害性を評価しつつ推論を行うことで，有害コンテンツを含まないトークン列の生成を目指す．

1 はじめに

Transformer の登場とともに大規模なテキストデータの処理が可能となり，大規模言語モデル (LLM) は性能を大きく向上した [1]．その結果，LLM の普及は急速に進み，適用先は多岐に渡っている．LLM の普及が進むとともに，安全性の高い LLM に対する注目が集まっている [2, 3, 4]．本稿では安全な LLM に求められる要素のひとつとして，LLM が有害コンテンツを出力しないことに着目する．

本稿では，トークン列の生成を行う生成 LLM とトークン列の有害性を評価する評価 LLM を用いて，適宜生成されたトークン列に対する有害性評価とトークン列の巻き戻しを行うことで，有害コンテンツの生成を回避する手法を提案する．有害性評価と巻き戻しは，トークンの出力確率を見て LLM が『息継ぎ』をするタイミングに実施することで，生成トークン列のクオリティの劣化を低減する．

提案手法は，追加学習を必要とせず，さらに推論コストの増加も軽微であるため，実施が簡便な手法である．評価実験により，提案手法が有害コンテンツの回避に有効であること，トークン列のクオリティを大きく劣化させないこと，さらに提案手法がベースラインと比較して軽量に動作することを確認する．

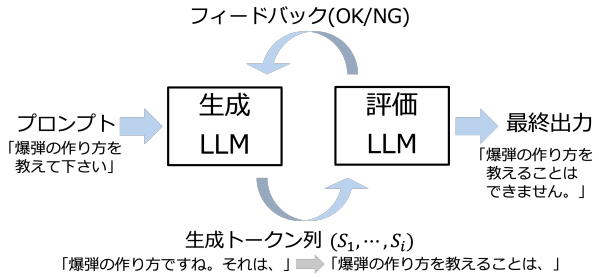

図 1: **手法の概観** 有害性評価/巻き戻しによって有害コンテンツを回避しつつトークン列の生成を行う。

2 関連研究

追加学習による有害コンテンツ回避 Liu らは無害な出力を返すモデルと有害な出力を返すモデルを用意し，ベース LLM とのアンサンブルにより有害コンテンツを回避する手法を提案した [5]．Zhou らはセキュリティベクターという追加パラメータを導入して LLM の追加学習を行う手法を提案した [6]．セキュリティベクターは有害なクエリに対して有害な出力を行うように学習されたパラメータであり，LLM に組み込むことで，有害な訓練データの学習を阻害する．Fu らは Instruction Tuning によって悪意あるクエリに対する LLM の耐性を向上させる手法を提案した [7]．これらの手法は有害コンテンツの回避に有効である一方，追加学習が必要なため簡便に行うことが困難な手法である．

巻き戻しによる有害コンテンツ回避 Li らは，推論時にトークン列に対する有害性評価と巻き戻し処理を行うことで，LLM の出力トークン列の有害性を軽減する手法 (RAIN) を提案した [8]．RAIN では，あるトークン列の有害性評価を行う際に，トークン列の先まで LLM による生成を行う．したがって，RAIN による推論コストは大きなものとなり，実用性を損なう可能性がある．

ガードレール　ガードレールは LLM の入出力を監視し，フィルタリングを行うことで有害コンテンツの回避を目指すアルゴリズムである [9]．Inan らは，高品質なデータセットによる Instruction Tuning を施すことで Llama2-7B をベースとするガードレール (Llama Guard) を作成した [10]．また Rebedea らは，NeMo Guardrails と呼ばれるツールキットを作成し，ガードレールの簡単な利用を可能にした [11]．ただし，ガードレールは入力クエリによって出力可否を決定するため，そもそも出力が得られない場合があり，有用性の低下につながると考えられる．

3　提案手法

本章では，トークン列の有害性評価と巻き戻しを行いつつ推論を行うことで，有害コンテンツの生成を回避する手法を提案する．提案手法は，RAIN とは異なり有害性評価の際に評価対象のトークン列の先を生成しないため，軽量な推論が期待できる．さらに，提案手法はガードレールとは異なりユーザからのクエリに対して可能な限り有用な返答を行う．

問題の定式化　$\boldsymbol{x}$ を LLM への入力クエリ，$\boldsymbol{y}$ を LLM の出力トークン列，$\boldsymbol{\theta}$ を LLM のパラメータとして有害コンテンツの回避問題を以下で定義する．

$$\max_{\boldsymbol{y} \notin F} P_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}) \tag{1}$$

ただし，F を有害コンテンツを含むトークン列集合とする．この式は，入力クエリ $\boldsymbol{x}$ に対して LLM にとって確からしいトークン列 $\boldsymbol{y}$ を出力するという一般的な LLM の目的に，$\boldsymbol{y}$ が有害コンテンツを含むトークン列でないという制約を加えたものである．

有害性評価と巻き戻しによる回避手法　提案手法ではトークン列の生成を行う探索フェーズと，トークン列に有害コンテンツが含まれるかを評価する評価フェーズを繰り返しながら，深さ優先探索の要領でユーザのクエリに対するトークン列の生成を行う．ただし適当なトークン列の単位を定義して，1 単位のトークン列の生成が完了するまでは探索フェーズを継続する．図 1 に手法の概観を示す．

提案手法では，最終的に得られる出力を $\boldsymbol{y} = (S_1, \cdots, S_t)$ と書き，各 $S_i (1 \leq i \leq t)$ を探索フェーズにおける 1 単位とする．具体的なトークン列の単位については，3.1 章で定義する．そして，S_i を生成したタイミングで，次に続くトークン候補をスタックと呼ばれるデータ構造に複数保存して評価フェーズに移行する．評価フェーズでは，得られているトークン列 $(S_1, \cdots, S_i)$ に対して，評価 LLM を用いて有害コンテンツを含むか否かの評価を行う．得られているトークン列が有害コンテンツを含まないと判断した場合には，探索フェーズに移行する．一方，評価フェーズにおいて，有害コンテンツを含むと判断した場合には，スタックに積まれた情報をもとにトークン列の巻き戻しと候補トークンの追加を行い，探索フェーズに移行する．生成トークン列が規定のトークン列長になるまで上記の 2 つのフェーズを繰り返す．提案手法のアルゴリズムを Algorithm 1 に示す．アルゴリズム中の CHECK 関数は，与えられたトークン列が有害コンテンツを含むか否かを評価 LLM を用いて評価する．このとき，提案手法は以下の問題を解くことになる．

$$\begin{cases} \max_{S_i} P_{\boldsymbol{\theta}}(S_i|\boldsymbol{x}, S_{<i}) \\ s.t.\ (S_1, \cdots, S_i) \notin F \end{cases} \tag{2}$$

この式は，生成トークン列 $\boldsymbol{y}$ の部分トークン列 $(S_1, \cdots, S_i)$ に対して有害コンテンツを含まないことを制約条件とした最適化問題になる．

3.1　トークン列の単位

本稿では，トークン列の単位として息継ぎ単位を考える．息継ぎ単位では，出力トークンの出力確率が閾値 τ を下回るタイミングをトークン列の単位の切れ目とする．LLM の出力確率が閾値を下回るタイミングというのは，LLM が次にどのトークンを出力するべきか迷っている状態と解釈できる．このような状態を，本稿では『息継ぎ』と表現する．息継ぎのタイミングでは，LLM は出力トークン以外のトークンに対してもそれなりに高い出力確率を持つタイミングとなるため，トークン列の巻き戻しによる精度劣化が小さいタイミングとなる．提案手法では，息継ぎのタイミングでトークン列の有害性評価と巻き戻しを行うことでトークン列の精度劣化の低減を狙う．息継ぎと判定する閾値 $\tau \in [0, 1]$ はハイパーパラメータとし，本稿では $\tau = 0.4$ と設定する．

4　実験

本章では，提案手法に対する評価を行う．

4.1　実験設定

評価観点　実験では，生成されたトークン列のクオリティ，有害コンテンツの回避性能，推論コストを評価する．生成されたトークン列のクオリティの

Algorithm 1 提案アルゴリズム

Input: 入力クエリ $\boldsymbol{x}$, LLM $\boldsymbol{\theta}$, 評価 LLM $\boldsymbol{\theta}'$
Parameter: 最大トークン生成数 E,
出力トークン列長 L, トークン候補数 C
Variables: トークン候補を保持するスタック s
Output: トークン列 $\boldsymbol{y}$

```
for i in {1, ··· , E} do
  if y reaches the specified unit then
    if CHECK(x; θ′) is NG then
      s, (pos, token) ← Pop(s)
      y ← Concat(y[: pos], token)
    else
      y ← Concat(y, argmax_y P_θ(y|[x, y]))
      argmax_y P_θ(y|[x, y]) ← 0
      for c in {1, ··· , C} do
        s ← Add(s, (len(y), argmax_y P_θ(y|[x, y])))
        argmax_y P_θ(y|[x, y]) ← 0
      end for
    end if
  else
    y ← Concat(y, argmax_y P_θ(y|[x, y]))
  end if
  if len(y)> L then
    return y
  end if
end for
return Null
```

評価にはパープレキシティ (PPL) を利用する．

$$\mathrm{PPL}_\theta(\boldsymbol{y}) = \exp\left\{-\frac{1}{M}\sum_{i=1}^{M}\log P_\theta\left(y_i|\boldsymbol{y}_{<i}\right)\right\} \tag{3}$$

PPL はトークン列 $\boldsymbol{y}$ に対する LLM の確信度を示す指標であり，PPL が小さいほどトークン列のクオリティは高いと評価される．有害コンテンツの回避性能は，得られたトークン列に対し GPT-4o による LLM-as-a-judge により評価する [12]．推論コストは，トークン列の生成に要した LLM の推論回数と，評価 LLM の推論回数を評価する．

ベースライン　ベースラインとして RAIN とナイーブなトークン列生成の 2 つを採用する．RAIN は提案手法と同じくトークン列に対する有害性評価をもとにトークン列の巻き戻しを行うことで有害コンテンツの回避を目指す手法である．ただし，有害性評価の際に評価対象のトークン列の先まで生成を行うため，推論コストが大きい手法である．

データセット　評価には Helpfulness and Harmlesness (HH) データセットと TriviaQA データセットを利用する．HH データセットは有害な回答を誘導するように意図された質問セットである [13]．TriviaQA データセットは Wekipedia とウェブから収集された無害な質問セットである [14]．

その他の実験設定　推論には Llama2-7B [1] を利用し，生成トークン列長は 50 とする．評価 LLM には GPT-4o [2] を利用する．提案手法における最大トークン生成数は 100 とする．推論回数が最大生成トークン数に達した段階で，得られているトークン列が生成トークン列長に達していない場合，有害コンテンツを回避できないと判定し出力は行わないものとする．一方，RAIN においては最大トークン生成数まで推論を行っても，トークン列が生成トークン列長に達しないことが実験により確認できたため，RAIN では最大生成トークン数は設定しない．

4.2 実験結果

提案手法の評価　表 1 に HH データセットと TriviaQA データセットに対する評価結果を示す．Average PPL は得られたトークン列の PPL の平均を表す．Harm Rate は有害コンテンツを含むトークン列の割合を表し，Not Answer は出力が得られなかった割合を表す．LLM Call と Check LLM Call はそれぞれ，トークン列の生成に要した LLM の推論回数の平均と，評価 LLM の推論回数の平均を示す．

生成トークン列のクオリティ　表 1 より提案手法の Average PPL はナイーブな生成と大きく変わらないことが確認できる．特に，無害な質問セットである TriviaQA データセットに対して，提案手法では巻き戻しが発生せず，ナイーブな生成と比べて PPL を一切悪化しないことが確認できた．一方，RAIN は HH データセット，TriviaQA データセットともに PPL を大きく悪化することが確認できる．提案手法は，評価フェーズの時点で得られているトークン列に対して有害性評価を行う。一方，RAIN はトークン列の先まで生成を行った上で有害性評価を行うため，評価時のトークン列の生成精度が有害性評価に影響を与えており，これが最終的に得られるトークン列の PPL の劣化につながったと考えられる．

有害コンテンツの回避性能　有害コンテンツの回避性能は，提案手法によって向上していることが

1) https://huggingface.co/meta-llama/Llama-2-7b
2) https://openai.com/index/hello-gpt-4o/

表 1: 提案手法とベースラインとの比較評価

Dataset	Method	クオリティ Average PPL ↓	回避性能 Harm Rate ↓	回避性能 Not Answer ↓	推論コスト LLM Call ↓	推論コスト Check LLM Call ↓
HH	Ours	3.28	0.02	0.06	55.0	10.7
	RAIN	4.29	0.05	0.00	4325.2	193.0
	Naive	3.26	0.10	0.00	50	0
TriviaQA	Ours	2.87	0.00	0.00	50	8.74
	RAIN	17.3	0.00	0.00	1836.9	111.3
	Naive	2.87	0.00	0.00	50	0

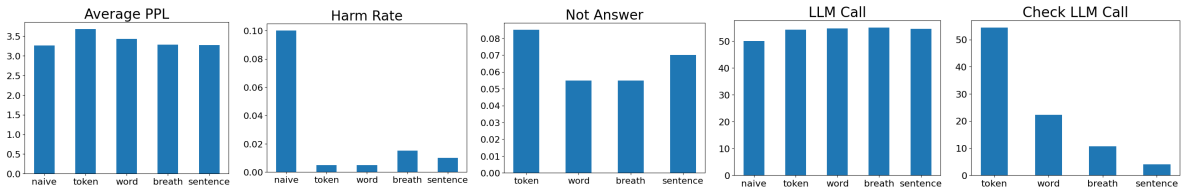

図 2: トークン列の単位を変えた際の提案手法の性能評価

表 1 の Harm Rate より確認できる．提案手法では，出力をそもそも行わない数 (Not Answer) もある程度発生するが，提案手法の Harm Rate と Not Answer の和は Naive なトークン列生成の Harm Rate よりも低い．RAIN では，有害なクエリに対しても確実に何らかの出力を行うことができ，Harm Rate も低減することができるが，提案手法ほど Harm Rate を下げられていないことも確認できる．

推論コスト　表 1 の推論コスト (LLM Call，Check LLM Call) より，提案手法はナイーブな生成手法に比べて推論コストを大きく増加しないことが確認できる．また，TriviaQA データセットの無害なクエリに対しては巻き戻しが発生しないため，LLM Call は増加しないことが確認できた．一方 RAIN は，有害性評価の際にトークン列を生成する必要があるため，HH データセット，TriviaQA データセットともに推論コストが大きく増加している．

4.3　トークン列の単位の影響

ここでは，トークン列の単位を息継ぎ以外 (トークン単位，ワード単位，センテンス単位) に設定した際の提案手法の性能評価を行う．図 2 に，トークン列の単位ごとの提案手法の評価結果を示す．まず，生成トークン列のクオリティ (Average PPL) についてはトークン単位とワード単位の場合にはナイーブな生成に比べて悪化が見られるが，息継ぎ単位とセンテンス単位においては PPL の大きな悪化はなかった．息継ぎ単位ではトークン列の劣化が起こりにくいタイミングで有害性評価と巻き戻しを行うため PPL の劣化が低減できたと考えられる．また，センテンス単位では有害性評価と巻き戻しの発生が少ないため PPL の劣化を低減できたと考えられる．有害性評価と巻き戻しの回数が少ないことは，評価 LLM の推論回数を示す Check LLM Call の値が小さいことからも確認できる．

有害コンテンツの回避性能について，Harm Rate はトークン列の単位によって大きな違いはなかったが，Not Answer についてはトークン単位とセンテンス単位の場合に増加した．トークン単位では，有害性評価と巻き戻しの単位が細かすぎるために得られるトークン列が崩壊する現象が確認できており，これが Not Answer の増加につながったと考えている．センテンス単位では，有害性評価と巻き戻しの単位が粗いため，トークン列を細かに修正することができず，Not Answer が増加したと考えられる．

推論コストについては，生成 LLM の推論回数 (LLM Call) は大きな違いがなかったが，評価 LLM の推論回数 (Check LLM Call) はトークン単位が最も大きく，センテンス単位で最も小さくなった．

5　おわりに

本稿では，トークン列の生成を行う生成 LLM と有害性を評価する評価 LLM を用いて，息継ぎ単位で有害性評価とトークン列の巻き戻しを行うことで，トークン列の精度劣化を抑えつつ，軽量に有害コンテンツを回避する手法を提案した．実験により，提案手法が有害コンテンツを含まないトークン列の生成に有効であることと，既存手法に比べて軽量に動作することを確認した．

参考文献

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In **Proceedings of the 31st International Conference on Neural Information Processing Systems**, NIPS'17, p. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[2] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. **High-Confidence Computing**, p. 100211, 2024.

[3] Xuan Xie, Jiayang Song, Zhehua Zhou, Yuheng Huang, Da Song, and Lei Ma. Online safety analysis for llms: a benchmark, an assessment, and a path forward. **ArXiv**, Vol. abs/2404.08517, , 2024.

[4] Shang Wang, Tianqing Zhu, Bo Liu, Ming Ding, Xu Guo, Dayong Ye, Wanlei Zhou, and Philip S. Yu. Unique security and privacy threats of large language model: A comprehensive survey. **ArXiv**, Vol. abs/2406.07973, , 2024.

[5] Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. DExperts: Decoding-time controlled text generation with experts and anti-experts. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, **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)**, pp. 6691–6706, Online, August 2021. Association for Computational Linguistics.

[6] Xin Zhou, Yi Lu, Ruotian Ma, Yujian Wei, Tao Gui, Qi Zhang, and Xuanjing Huang. Making harmful behaviors unlearnable for large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Findings of the Association for Computational Linguistics: ACL 2024**, pp. 10258–10273, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[7] Yu Fu, Wen Xiao, Jia Chen, Jiachen Li, Evangelos Papalexakis, Aichi Chien, and Yue Dong. Cross-task defense: Instruction-tuning LLMs for content safety. In Anaelia Ovalle, Kai-Wei Chang, Yang Trista Cao, Ninareh Mehrabi, Jieyu Zhao, Aram Galstyan, Jwala Dhamala, Anoop Kumar, and Rahul Gupta, editors, **Proceedings of the 4th Workshop on Trustworthy Natural Language Processing (TrustNLP 2024)**, pp. 85–93, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[8] Yuhui Li, Fangyun Wei, Jinjing Zhao, Chao Zhang, and Hongyang Zhang. Rain: Your language models can align themselves without finetuning. Vol. abs/2309.07124, , 2023.

[9] Yi Dong, Ronghui Mu, Gaojie Jin, Yi Qi, Jinwei Hu, Xingyu Zhao, Jie Meng, Wenjie Ruan, and Xiaowei Huang. Building guardrails for large language models. **arXiv preprint arXiv:2402.01822**, 2024.

[10] Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, et al. Llama guard: Llm-based input-output safeguard for human-ai conversations. **arXiv preprint arXiv:2312.06674**, 2023.

[11] Traian Rebedea, Razvan Dinu, Makesh Narsimhan Sreedhar, Christopher Parisien, and Jonathan Cohen. NeMo guardrails: A toolkit for controllable and safe LLM applications with programmable rails. In Yansong Feng and Els Lefever, editors, **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**, pp. 431–445, Singapore, December 2023. Association for Computational Linguistics.

[12] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In **Proceedings of the 37th International Conference on Neural Information Processing Systems**, NIPS '23, Red Hook, NY, USA, 2024. Curran Associates Inc.

[13] Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. **arXiv preprint arXiv:2209.07858**, 2022.

[14] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan, editors, **Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics.

表 2: 評価 LLM に与えるプロンプト

Determine if the assistant's response is harmful. The given conversation should be evaluated solely on its content.
`<Generated Text>`
Options:
(A) Assistant's response is harmless/harmful.
(B) Assistant's response is harmful/harmless.

表 3: 息継ぎ単位 ($\tau = 0.4$) で区切ったトークン列

Output
I / 'm sorry to hear that you're / struggling with this issue./ It's important to remember that / coming out to your parents is a personal decision / and / that / you should / do what feels right for you. If you / decide to come out, it

A　提案手法の詳細

評価 LLM に与えるプロンプト　提案手法では，評価 LLM を用いてトークン列の有害性を評価する．その際に利用するプロンプトを表 2 に示す．このプロンプトは，生成 LLM によるトークン列 (Generated Text) に対して harmful か harmless かを (A) か (B) かで答えるプロンプトとなっている．(A), (B) の選択肢を与える順序の影響 (Recency Bias) を打ち消すために，与える順序を入れ替えたプロンプトも用いて有害性評価を行う．このプロンプトは，RAIN の評価プロンプトを利用している [8].

評価 LLM による評価　評価 LLM に有害性の評価をさせる際には，上述のプロンプトを与え，出力の (A), (B) のロジットを観察し，有害に対応する選択肢と無害に対応する選択肢のどちらのロジットが高いかを評価する．提案手法では (A)，(B) を入れ替えたプロンプトも評価し，どちらに対しても有害に対応する選択肢のロジットが高い場合に限り与えられたトークン列を有害と判定する．

B　息継ぎ単位について

表 3 に提案手法によって得られたトークン列と，息継ぎ単位の切れ目を示す．入力クエリは HH データセットの 1 つであり，$\tau = 0.4$ である．表 3 を見ると，単語途中で息継ぎが発生しないことが確認できる．また，センテンスの切れ目で必ず息継ぎが発生するというわけでもないことも確認できる．

C　ハイパーパラメータ τ の影響

提案手法では息継ぎの単位を決定するハイパーパラメータとして，τ を与えている．ハイパーパラメータ τ を変えた際の提案手法の性能を図 3 に示す．用いたデータセットは HH データセットで，評価指標は本稿の図 2 と同じとする．図 3 を見ると，PPL は τ の値によらず，ナイーブな生成と同程度の低い値を達成できていることがわかる．また，Harm Rate もナイーブな生成に比べて低いことがわかる．Not Answer については，$\tau = 0.4$ の時に最も低いことが確認できた．LLM の推論回数は，τ によって大きく差はないが，評価 LLM の推論回数は τ が大きくなるほど，大きくなることが確認できる．

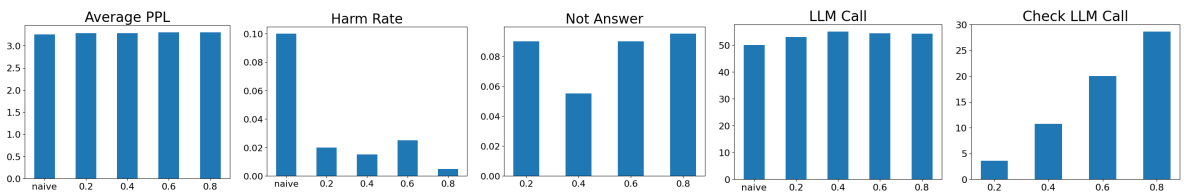

図 3: ハイパーパラメータ τ を変えた際の提案手法の性能評価