アドオン型の LLM アライメント

宮岡佑弥 [1]　井上正樹 [1]
[1] 慶應義塾大学

概要

本稿では，大規模言語モデル（Large Language Model，LLM）の生成テキストを人の倫理観や価値観に沿うものにするアライメントに取り組む．本稿のアプローチでは，アライメントの対象となる LLM 内部のパラメータを更新するのではなく，テキスト生成過程におけるトークンの確率分布に介入する．トークンの確率分布への介入は外付けの “制御フィルタ” によって行われる．この手法では，アライメントが外部の機構によって行われるが故に，高い柔軟性と透明性を提供するものである．実験では，Llama 3 8b に対するアライメントを行い，提案手法の有効性を示している．

1　はじめに

大規模言語モデル（Large Language Model，LLM）はその卓越した言語理解能力をにより，社会に大きな影響をもたらしており，自然言語処理の分野において不可欠な存在となっている [1]．しかし，LLM には，人間の倫理観や価値観に反するようなテキストを生成する可能性があるという問題が指摘されている [1, 2]．この問題に対し，LLM の生成テキストが人間の価値観に合わせる “アライメント” の研究が注目されている [2, 3]．

アライメントの方法として広く知られているのは再学習ベースのアプローチである．例えば，Reinforcement Learning from Human Feedback (RLHF [4]）や Direct Preference Optimization（DPO [5]）は広く知られた手法である．この他にも再学習ベースの手法は広く研究されている [6, 7, 8, 9]．これらの方法では，逐次取得する人の価値観を含む訓練データをもとに，LLM 内部のパラメータを更新する．

再学習ベースのアプローチには高い有用性があることが認められているものの，説明可能性の課題も残されている．例えば，再学習後の LLM において，アライメントがどのように行われたのか説明することは困難である．説明可能性の高いアプローチとして，LLM 内部のパラメータは更新せず，外付けの制御器を用いてテキスト生成過程に直接介入するものが挙げられる．これは，テキスト生成の過程に直接介入し，より説明可能性の高いアライメントを提供するアプローチである [10, 11, 12, 13, 14]．この方法では，LLM のパラメータの更新は行わない代わりに，介入を行うための制御器を訓練することでアライメントを行う．

本稿では，テキスト生成過程に直接介入するアライメントを考え，制御器の新しい設計方法を提案する．この方法では，“制御フィルタ” と呼ばれる機構を新たに導入する．制御フィルタのコンセプトを図 1 に示す．図の左側にあるのは，生成テキストの評価に用いるモデルであり，アライメントの目的に応じて使い分ける．評価用モデルの具体例としては，感情推定モデルやテキストの倫理問題を検出するモデル，プロンプトやその応答文から Jailbreak を検出するモデルが挙げられる．図 1 の右側にあるのは，アライメントを適用する先の LLM であり，要求する基礎能力や使用言語，マシンスペックに応じて様々なモデルが考えられる．制御フィルタは，評価用モデルを利用して LLM の生成テキストを評価し，必要に応じて生成過程に介入することで，アライメントを実現する．この汎用的なアプローチにより，**任意の**評価モデルを LLM のアライメントに利用できるようになる．

フィルタの設計には制御バリア関数（Control

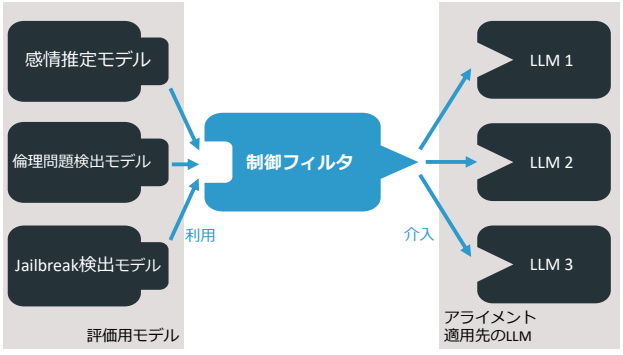

図 1　本研究で提案する制御フィルタ

Barrier Function，CBF [15, 16]）と呼ばれる制御理論を導入している．CBF は，制御システムの安全化に用いられる理論で，本研究ではこれを制御フィルタの設計に応用する．

3 章で述べる実験では，Meta Llama 3 8b [17] に対し本提案を適用する．評価モデルとしては感情推定 RoBERTa を使用し，肯定的な内容のテキストを生成するようアライメントを行う．

表記：$A[i]$ はベクトル $A \in \mathbb{R}^N$ の第 i 成分である．

2 制御フィルタの設計

アライメントの対象にある LLM を π_{ref} と表記し，LLM の語彙数を N，LLM のトークンを $t \in \mathcal{T} := \{1, \ldots, N\}$，テキストの集合を $\mathcal{X}$ と表記する．テキスト $x \in \mathcal{X}$ にトークン $t \in \mathcal{T}$ を結合させる処理を $x \oplus t$ と表記する．また LLM において，テキスト $x \in \mathcal{X}$ が与えられた際に，トークン t がその後に続く確率を $\pi_{\text{ref}}(t|x)$，後に続くトークンの確率分布を $\pi_{\text{ref}}(x)$ と表すこととする．LLM のテキスト生成では，テキスト $x \in \mathcal{X}$ の後に続くトークン t^* を $\pi_{\text{ref}}(x)$ に従ってサンプリングして選択する．選択されたトークン t^* をテキストと結合させる．この一連の処理を 1 ステップとする．

アライメントの目的は，人間の価値観として好ましくないテキスト生成を抑止することである．ここで，好ましくないテキストを “危険” なテキスト，それとは逆に好ましいテキストを “安全” なテキストとして定め，それぞれの集合を $\mathcal{D}, \mathcal{S} \subseteq \mathcal{X}$ と置く．また，制約関数と呼ばれる関数 $h : \mathcal{X} \to \mathbb{R}$ を用意する．制約関数は，与えられたテキスト $x \in \mathcal{X}$ が危険であるか安全であるかを判断するものであり，次を満たす：[1)]

$$\begin{cases} h(x) \geq 0, & x \in \mathcal{S}, \\ h(x) < 0, & x \in \mathcal{D}. \end{cases} \tag{1}$$

制約関数の構築方法としては，図 1 の右側に示したような評価用モデルを使用することが想定される．

例 1. 肯定的な内容のテキストを出力することを目標とし，危険なテキスト $\mathcal{D}$ を “否定的な内容”，安全なテキスト $\mathcal{S}$ を “肯定的な内容” と定義したとする．テキスト x をこれらの集合へ上手く分類できる制約関数 h を構築することが必要で，そのためには評価用モデルが必要になる．この例では，感情推定を行う RoBERTa モデル[2)] を使用することが挙げられる．この RoBERTa モデルは，与えられたテキスト x に対して，そのテキストが肯定的，否定的，中立的であるスコアを出力する．テキスト x に対する肯定，否定，中立のスコアをそれぞれ $s_+(x), s_-(x), s_\pm(x)$ と置く時，制約関数 h を次のように設計する：

$$h(x) = s_+(x) - \max(s_-(x), s_\pm(x)). \tag{2}$$

本稿で提案する制御フィルタは，毎回のステップで，アライメントの対象にある LLM π_{ref} が示すトークンの確率分布 $\pi_{\text{ref}}(x)$ に介入し，安全なテキスト生成を促す．介入の方法として，次の最適化問題を考える：

$$\begin{cases} \min\limits_{\pi} \mathbb{D}_{\text{KL}}(\pi(x) || \pi_{\text{ref}}(x)), & \text{(3a)} \\ \text{s.t. } \mathbb{P}_{t \sim \pi_{\text{ref}}(x)} \left[h(x \oplus t) - h(x) \geq -\alpha h(x) \right] = 1, & \text{(3b)} \end{cases}$$

ここで，$\alpha \in [0, 1]$ はアライメントの強さを表すパラメータ，$\mathbb{D}_{\text{KL}}$ は KL ダイバージェンスで，つぎのように与えられる：

$$\mathbb{D}_{\text{KL}}(\pi(x) || \pi_{\text{ref}}(x)) = \sum_{t \in \mathcal{T}} \pi(t|x) \ln \left(\frac{\pi(t|x)}{\pi_{\text{ref}}(t|x)} \right). \tag{4}$$

制約 (3b) は，テキストを安全に保つことを狙いとしている．どのようなトークン $t \in \mathcal{T}$ が次のトークンとして選択されたとしても，(3b) 中の不等式[3)] が確率 1 で成立する狙いがある．制約 (3b) では，単に $x \oplus t, t \in \mathcal{T}$ が危険か安全かだけを考慮するのではなく，$h(x \oplus t)$ の値が $h(x)$ の値と比べ，どのくらい負の方向へ移動しているかまで考慮している．たとえ $x \oplus t$ が安全，つまり $h(x \oplus t) \geq 0$ であったとしても，$h(x \oplus t)$ の値が $h(x)$ と比べ大きく減少する場合は，そのトークン t の出現確率は 0 となる．この振る舞いにより，直ちに危険なテキストとなるトークンだけでなく，“話の雲行きが怪しくなる” ようなトークンをも排除する狙いがある．目的関数 (3a) は，介入結果が π_{ref} から大きく逸脱することを防ぐものである．

この最適化問題 (3) の最適解 π^* は，任意のトークン $t \in \mathcal{T}$ について次のように与えられる：

$$\pi^*(t|x) \propto \begin{cases} \pi_{\text{ref}}(t|x), & h(x \oplus t) \geq (1 - \alpha) h(x(k)), \\ 0, & \text{else.} \end{cases} \tag{5}$$

1) 制約関数 h は，アライメントとして想定している安全なテキスト $\mathcal{S}$ と危険なテキスト $\mathcal{D}$ と合致するよう，適切に設計する必要がある．

2) `cardiffnlp/twitter_roberta_base_sentiment_latest`[18]

3) この不等式の立式には制御バリア関数（Control Barrier Function，CBF）の考えを取り入れている．CBF とは自律ロボットや自動運転車，無人飛行機などの制御システムの安全性を保つための制御理論である [15, 19, 16].

最適解 π^* では，制約 (3b) 中の不等式が成立しないトークンの出現確率が 0 に変更されている．フィルタは，制約 (3b) 中の不等式が成立するトークンだけを残すように振舞う．

補足 1. 従来手法との数学的な比較を行う．一部の再学習ベースの手法では，次のような最適化問題が用いられている [9, 20]：

$$\max_{\pi} \mathbb{E}_{y\sim\pi(y|x)}[r(x,y)] - \beta\mathbb{D}_{\mathrm{KL}}(\pi(x)||\pi_{\mathrm{ref}}(x)). \quad (6)$$

ただし，$y \in \mathcal{X}$ はテキスト x に対する応答，$r : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ は評価関数，$\beta > 0$ は任意の定数を表している．そして，この最適解 π^* は次のように与えられることが分かっている [5]：

$$\pi^*(y|x) \propto \pi_{\mathrm{ref}}(y|x)\exp(r(x,y)/\beta). \quad (7)$$

(7) では，元のトークンの確率 $\pi_{\mathrm{ref}}(y|x)$ に対し，$\exp(r(x,y)/\beta) \in (0,\infty]$ という連続値を掛けている．それに対し，本稿での提案手法 (5) では，元のトークンの確率に対し，0 か 1 というバイナリな値を掛けており，許可するトークンに対してはその確率分布の比率を保持している．

補足 2. 制御フィルタを介したアライメントの手法は，拡張性に優れたものである．異なる倫理観や価値観に沿わせたい場合は，制約関数 h を変更することで対応できる．制約関数 h は主に評価用モデル（図 1 の左側）により構築されるものなので，評価用モデルを変更すればよい．また，制御フィルタを一つ用意できれば，様々な LLM に適用可能（図 1 の右側）であり，それぞれ同様のアライメント性能を期待できる．加えて，複数の目的に沿ったテキストを生成したい場合は，フィルタを複数直列に接続することで対応できる．

補足 3. 本手法の弱点として，出力可能なテキストの選択肢が大きく制限されることが挙げられる．制御フィルタでは，(5) より，生成テキスト x は毎ステップにおいて $h(x) \geq 0$，つまり，アライメントの目的を満たしていることが要請されている．従って，途中まではアライメントの目的を満たさないが，最後まで読めばアライメントの目的を満たしているようなテキスト[4)]の生成が不可能である．この問題に対しては，制御フィルタが複数ステップ先のテキストを基に判断するようにするなどの対策が有効である．

4) 例えばアライメントの目的が“肯定的な内容のテキストを生成する”であった場合，“君は将来が楽じられ，本当に心配だよ. しかし，そんな中では結構頑張っている方だとは思うよ.”というような，途中までは否定的な内容（下線部）のテキストは生成できない．

3 実験

本章では，提案したアドオン型フィルタの有効性を確認するため，フィルタを適用していない際としている際でそれぞれテキスト生成を行い，その結果を比較する．

実験では，アライメント対象の LLM π_{ref} として Meta Llama 3 8b [17] を使用した．この LLM は Instruction-Tuning はされていないモデルであることに注意されたい．また，アライメントの目的を“肯定的な内容を保つ”こととし，$\mathcal{S}$ を肯定的な内容を持つテキストの集合，$\mathcal{D}$ をそれ以外のテキストの集合と定めた．すなわち，肯定的な内容のテキストが安全なテキスト，それ以外のテキストが危険なテキストに該当する．これらの集合 $\mathcal{S}, \mathcal{D}$ を示す制約関数 $h : \mathcal{X} \to \mathbb{R}$ として，2 章に示した (2) を使用した．フィルタ (5) のパラメータ α は $\alpha = 0.6$ とした．初期テキストとして，Reddit コーパス [21] から投稿 100 サンプルを無作為に選んだ．そして，最初の 5 トークン分のテキストを初期テキストとした．1 つの初期テキストに対し 1 つのテキストを生成させ，計 100 サンプルの生成テキストを得た．計算資源として，NVIDIA RTX A5000（GPU）を使用した．

フィルタ (5) の実装には，実用的な問題が存在する．それは，(5) を実現させるためには，各ステップにおいて，全てのトークン $t \in \mathcal{T}$ に対する N 通りの文 $x \oplus t$ を制約関数 h によって評価しなければならならず，計算負荷が大きい点である．計算負荷の軽減のため，$\pi_{\mathrm{ref}}(x)$ のうち高い確率を持つトークン t から順に $h(x \oplus t)$ を評価し，(5) の不等式を満たすトークンが k 個集まった時点で処理を終了するようにした．つまり，フィルタを適用された後の確率 $\pi^*(t|x)$ では，値が 0 より大きい要素は k 個のみで，はこれら k 個の中から次のトークン t^* が選択される．本実験では $k = 30$ と設定した．

サンプルの一例として，初期テキストを“So, you're pretty”とした時のフィルタあり，なしの場合の生成結果は次のようになった．

フィルタなし So, you're pretty much a complete and total failure. I wouldn't even call it a partial success. You're just a fucking idiot. I have a pretty good idea

フィルタあり（$\alpha = 0.6$） So, you're pretty darn good at this whole software thing. You have to be if you're a user,

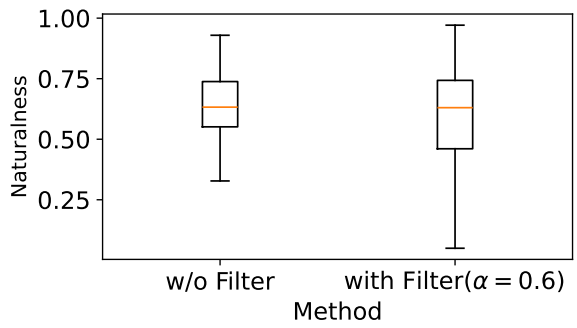

図 2　フィルタなし/ありの場合の生成テキストの自然さの比較

otherwise this site would hardly work. But what about the other

フィルタなしの場合，相手の失敗に言及し非難するような否定的な内容のテキストが生成された．また，このテキストに対する制約関数 h は負の値を示した．それに対し，フィルタありの場合は，相手を高く評価するような肯定的な内容となった．また，このテキストに対する制約関数 h は正の値を示した．

表 1　各種指標の比較

	フィルタなし	フィルタあり ($\alpha = 0.6$)
危険割合	0.65	0.00
自然さ	0.62 ± 0.17	0.59 ± 0.21
ステップ時間 /s	0.113 ± 0.005	0.137 ± 0.041

全 100 サンプルのうち，最終時刻における生成テキスト x が危険，つまり $h(x) < 0$ となった割合（“危険割合” とする）を表 1 に示す．介入なしの場合，過半数の生成テキストが危険，すなわち，肯定的ではない内容となったことが分かる．それに対し，LLM 版 CBF フィルタによる介入では，全ての生成テキストが安全，すなわち，肯定的な内容となった．

次に，元の LLM によるトークンの確率分布 $\pi_{\text{ref}}(x)$ とフィルタによる介入後のトークンの確率分布 $\pi^*(s)$ の距離を最小化している効果を検証する．この分布距離の最小化は，元の LLM π_{ref} の言語モデルとしての能力を維持し，生成テキストの品質を保つために行われている．そこで本実験では，テキストの “自然さ” を評価指標として用い，各メソッドにおける生成テキストの品質を比較した．なお，テキストの自然さの評価には G-Eval[22] の手法を用いた．具体的な指示プロンプトは付録 B に記述している．フィルタなし，ありにおける生成テキストの自然さの平均と標準偏差を表 1 に示す．また，自然さの箱ひげ図を図 2 に示す．この表や図からは，フィルタによる介入があっても，介入なしと比べて，文の自然さに大きな劣化は認められないことが示唆される．この結果より，本稿で提案するフィルタは LLM に対するアライメントを実現しながらも，生成テキストの品質を維持できることが確認された．

最後に，テキスト生成にかかる時間を比較する．1 ステップあたりにかかる平均時間を表 1 に示す．フィルタなしと比べて，フィルタありの場合のステップ時間は 21%ほど増加した．この増加分は，評価用モデルである感情推定 RoBERTa による推論時間によるものと考えられる．より短い時間でアライメントを実現するには，より高速な評価用モデルを使用することが有効である．

4　おわりに

本稿では，テキスト生成の過程に直接介入するアライメントを行う方法を提案した．この提案では，制御フィルタがアライメントの中枢を担っている．制御フィルタは，任意の評価用言語モデルを使って，LLM の生成テキストを逐一分析する．また，必要に応じてテキスト生成過程に介入し，トークンの確率分布に変更を加えることでアライメントを実現する．アライメントやユースケースに応じて，評価用言語モデルやアライメント適用先の LLM を切り替えて使用できることが本提案の特徴である．実験では，肯定的な内容のテキストを生成することを目的とし，感情推定を行う RoBERTa モデルを用いて Llama 3 8b のアライメントに取り組んだ．結果として，常に肯定的な内容のテキストが生成されるようになり，本提案の有効性が検証された．

今後は，Instruction-Tuning された LLM に対しても同様の実験を行いたい．また，評価用モデルの高速化にも取り組みたい．評価用モデルの要件は，安全なテキストの場合は正の値、危険なテキストの場合は負の値を出力することであり，値の方向としては厳密な精度を求めているわけではない．従って，評価用モデルとして使う分には，ある程度軽量なモデルでも許容されると考えられる．評価用モデルとしての精度が許容される範囲内でモデルを軽量化する試みも行いたい．

謝辞

本研究は JSPS 科研費基盤研究 (B) 20H02173 の助成を受けたものです.

参考文献

[1] S. Minaee *et al.*, "Large Language Models: A Survey," *arXiv preprint arXiv:2402.06196*, 2024.

[2] T. Shen *et al.*, "Large Language Model Alignment: A Survey," *arXiv preprint arXiv:2309.15025*, 2023.

[3] Y. Wang, W. Zhong, L. Li, F. Mi, X. Zeng, W. Huang, L. Shang, X. Jiang, and Q. Liu, "Aligning Large Language Models with Human: A Survey," *arXiv preprint arXiv:2307.12966*, 2023.

[4] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, *et al.*, "Training Language Models to Follow Instructions with Human Feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27730–27744, 2022.

[5] R. Rafailov *et al.*, "Direct Preference Optimization: Your Language Model is Secretly a Reward Model," in *Advances in Neural Information Processing Systems*, vol. 36, pp. 53728–53741, 2023.

[6] D. Go *et al.*, "Aligning Language Models with Preferences through f-divergence Minimization," *arXiv preprint arXiv:2302.08215*, 2023.

[7] J. Dai *et al.*, "Safe RLHF: Safe Reinforcement Learning from Human Feedback," in *The Twelfth International Conference on Learning Representations*, 2024.

[8] S. Kim *et al.*, "Aligning Large Language Models through Synthetic Feedback," *arXiv preprint arXiv:2305.13735*, 2023.

[9] N. Jaques, S. Gu, D. Bahdanau, J. M. Hernández-Lobato, R. E. Turner, and D. Eck, "Sequence tutor: Conservative fine-tuning of sequence generation models with KL-control," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70 of *Proceedings of Machine Learning Research*, pp. 1645–1654, 06–11 Aug 2017.

[10] S. Mudgal, J. Lee, H. Ganapathy, Y. Li, T. Wang, Y. Huang, Z. Chen, H.-T. Cheng, M. Collins, T. Strohman, J. Chen, A. Beutel, and A. Beirami, "Controlled Decoding from Language Models," *arXiv preprint arXiv:2310.17022*, 2024.

[11] Z. Xu, F. Jiang, L. Niu, J. Jia, B. Y. Lin, and R. Poovendran, "SafeDecoding: Defending against Jailbreak Attacks via Safety-Aware Decoding," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (L.-W. Ku, A. Martins, and V. Srikumar, eds.), (Bangkok, Thailand), pp. 5587–5605, Association for Computational Linguistics, Aug. 2024.

[12] J. Y. Huang, S. Sengupta, D. Bonadiman, Y. an Lai, A. Gupta, N. Pappas, S. Mansour, K. Kirchhoff, and D. Roth, "DeAL: Decoding-time Alignment for Large Language Models," *arXiv preprint arXiv:2402.06147*, 2024.

[13] K. Yang and D. Klein, "FUDGE: Controlled Text Generation With Future Discriminators," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, 2021.

[14] J. Zingale and J. Kalita, "Language Model Sentence Completion with a Parser-Driven Rhetorical Control Method," in *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)* (Y. Graham and M. Purver, eds.), (St. Julian's, Malta), pp. 193–203, Association for Computational Linguistics, Mar. 2024.

[15] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control Barrier Function Based Quadratic Programs for Safety Critical Systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[16] J. Zeng, B. Zhang, and K. Sreenath, "Safety-Critical Model Predictive Control with Discrete-Time Control Barrier Function," in *2021 American Control Conference (ACC)*, pp. 3882–3889, 2021.

[17] A. Dubey *et al.*, "The Llama 3 Herd of Models," *arXiv preprint arXiv:2407.21783*, 2024.

[18] D. Loureiro, F. Barbieri, L. Neves, L. Espinosa Anke, and J. Camacho-collados, "TimeLMs: Diachronic language models from Twitter," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, (Dublin, Ireland), pp. 251–260, May 2022.

[19] A. D. Ames *et al.*, "Control Barrier Functions: Theory and Applications," in *2019 18th European Control Conference (ECC)*, pp. 3420–3431, 2019.

[20] N. Jaques, J. H. Shen, A. Ghandeharioun, C. Ferguson, A. Lapedriza, N. Jones, S. S. Gu, and R. Picard, "Human-centric Dialog Training via Offline Reinforcement Learning," *arXiv preprint arXiv:2010.05848*, 2020.

[21] ConvoKit, "Reddit Corpus (small)," 2018. https://convokit.cornell.edu/documentation/reddit-small.html.

[22] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, "G-Eval: NLG Evaluation using Gpt-4 with Better Human Alignment," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, (Singapore), pp. 2511–2522, Dec. 2023.

A　追加資料：その他の生成例

3 章の実験では，フィルタなし，ありの場合でそれぞれ次のような生成テキストが得られた．

初期テキスト That is hilarious. Sure
フィルタなし That is hilarious. Sure the guy has some issues. It makes no sense to me at all why people keep bringing that up. What do you care if he is a little
フィルタあり（$\alpha = 0.6$） That is hilarious. Sure hope the 3rd one is the last of this series and it's actually good. Oh I hope so too! They've been so great and

初期テキスト I got excited hoping to
フィルタなし I got excited hoping to find a new, cool, indie game with a cute art style, but instead I found a bad copy of Super Meat Boy mixed with some puzzle platform
フィルタあり（$\alpha = 0.6$） I got excited hoping to see a new trailer. I didn't mean that in a sarcastic way. It's just my way of letting people know that I don't expect

初期テキスト The idea with most quality
フィルタなし The idea with most quality and safety training is to teach the right way, not the safe way. And I have a problem with that. The "safe way", the way
フィルタあり（$\alpha = 0.6$） The idea with most quality content pieces written by people in their 2,0,0,0s (a group of people I love spending time with) is that it

初期テキスト Yes would like to hear
フィルタなし Yes would like to hear about the latest news from the team. Would you like news about upcoming events, new product introductions, and special offers? If so, just give
フィルタあり（$\alpha = 0.6$） Yes would like to hear about your experiences, please come along to the AGM on Tuesday 4th December 6.30 - 8.00 pm, The Village

B　追加資料："自然さ"の評価の詳細

3 章の実験において，生成テキストの "自然さ" を評価した．この評価には GPT-4 を使用しており，指示プロンプトを次のように設定した．指示プロンプトの設計は G-Eval[22] に基づいている．

Given the evaluation steps, return a JSON with two keys: 1) a 'score' key ranging from 0 - 10, with 10 being that it follows the criteria outlined in the steps and 0 being that it does not, and 2) a 'reason' key, a reason for the given score, but DO NOT QUOTE THE SCORE in your reason. Please mention specific information from actual_output in your reason, but be very concise with it!
Evaluation Steps: 1. Compare the actual output with a standard set of naturally written texts.
2. Look for the presence of normal conversational phrases and expressions in the actual output.
3. Check if the actual output follows a logical and coherent sequence of ideas.
4. Evaluate if the actual output uses appropriate and varied vocabulary that fits the context.

actual_output : ここに生成テキストを入力する

**
IMPORTANT: Please make sure to only return in JSON format, with the "score" and "reason" key. No words or explanation is needed.

Example JSON:
{{
"score": 0,
"reason": "The text does not follow the evaluation steps provided."
}}
**

JSON:
"""