コーパスの逆蒸留

盧 慧敏 [1]　磯沼 大 [1,2,3]　森 純一郎 [1,4]　坂田 一郎 [1]
[1] 東京大学　[2] エディンバラ大学　[3] 国立情報学研究所　[4] 理研 AIP
{luhuimin1999, isonuma, isakata}@ipr-ctr.t.u-tokyo.ac.jp　mori@mi.u-tokyo.ac.jp

概要

本稿では、学習データ蒸留を逆向きに適用する学習データの逆蒸留を導入し、大規模言語モデルへの応用について議論する。学習データの逆蒸留では、ある学習データがもたらすモデルの変化と逆の変化をもたらす学習データを生成する。例えば有害な文章を含むコーパスを逆蒸留することで、モデルから有害な表現を忘却させる学習データが得られる。本稿では、一般的なコーパスに適用可能な、非常に軽量・単純な逆蒸留手法を導入し、モデルの有害性除去を例にその有効性を検証した。逆蒸留された文章が、有害性を除去するための学習データとして機能することを、様々なモデルを通じて示した。

1　はじめに

機械学習モデルの学習データを圧縮する方法として、近年学習データ蒸留が注目を集めている [1]。図 1 (a) に示すように、学習データ蒸留とは、元の学習データがもたらすモデルの変化と同様の変化をもたらす学習データを生成する技術である。これにより、例えば大規模な学習データで学習した場合と同様の影響をもたらす少量の学習データが得られる。

本稿では、学習データ蒸留を逆向きに適用する学習データの逆蒸留を導入し、大規模言語モデル（LLM）への応用について議論する。図 1 (b) に示すように、学習データの逆蒸留では、元の学習データがもたらす変化と逆の変化をもたらす学習データを生成する。例えば差別的な文章を含むコーパスを逆蒸留することで、LLM から有害な表現を忘却させる学習データが得られることが期待される。同様の発想に立った方法として勾配上昇法による逆学習 [2] があるが、単純に勾配を反転させて学習させると、モデルの性能が著しく毀損してしまう。一方、逆蒸留は一般的な文章でありつつも逆向きにモデルを更新する学習データを生成することで、言語モデルの性能を維持しつつ、特定のデータの忘却を試みる。

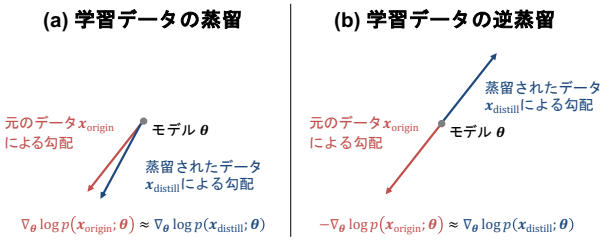

図 1　(a) 学習データの蒸留と (b) 学習データの逆蒸留。

学習データ蒸留/逆蒸留を LLM の学習コーパスに適用するにあたり、いくつかの障壁が存在する。従来の学習データ蒸留手法は、学習データを連続値として扱い、その勾配と元のデータがもたらす勾配が一致するように学習データを最適化する。したがって蒸留で得られるデータは文章ではなく連続値であり、蒸留に用いたモデルと異なるモデルの学習に用いることができない。また、最適化に際し 2 階微分の計算を伴うため、大規模なモデルやデータに適用するには膨大な計算量を要する。実際、既存研究の多くは画像データを対象としており [1, 3, 4, 5, 6]、自然言語に適用した研究もあるものの、文章分類のファインチューニングデータに留まり、一般的なコーパスの蒸留には用いられていない [7, 8, 9, 10]。

そこで本稿では、一般的なコーパスの蒸留/逆蒸留に適用可能な、対照的デコーディング [11, 12] を用いた単純・軽量な学習データ蒸留手法を提案する。対照的デコーディングとは、二つの言語モデルによる文章の生成確率の差分をもとに文章を生成する方法であり、既存手法よりも極めて軽量ながら、一般的な文章の形式でデータを蒸留できる。対照的デコーディングにより生成された文章は、蒸留されたコーパスとしてみなせることを数学的に示す。

実験では、逆蒸留の応用例として LLM の有害性除去を取り上げ、言語モデルの性能を維持しつつ、有害な文章の生成を抑制できることを示す。また、ある LLM を用いて逆蒸留したコーパスは、異なる LLM の有害性も除去でき、逆蒸留で得られる学習データが汎用的に機能することを示す。

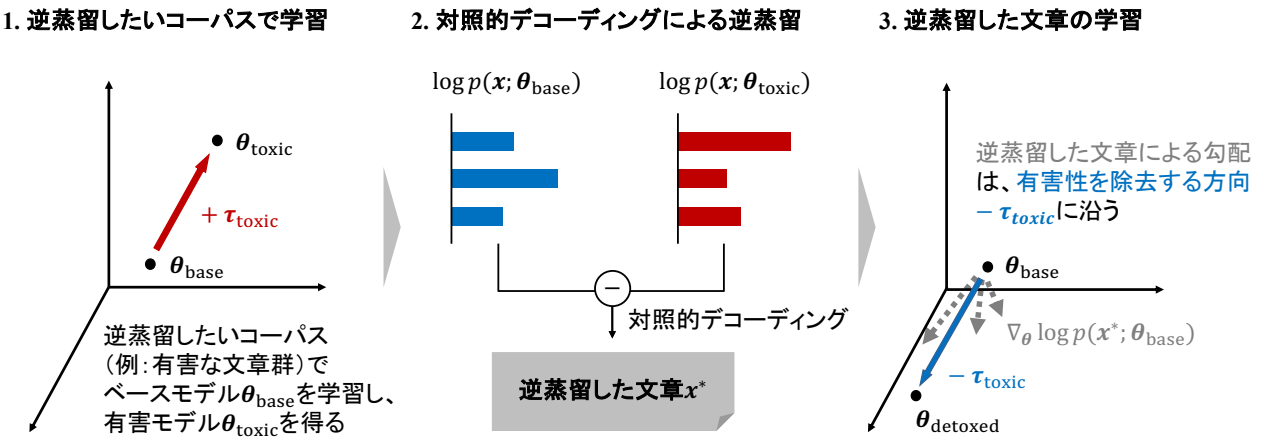

図 2　有害なコーパスの逆蒸留により、元のモデル θ_{base} から有害性を除去したモデル θ_{detoxed} を得るまでの流れ。

2　コーパスの逆蒸留

ここから、対照的デコーディングによりコーパスを逆蒸留する方法について説明する。本節では有害性除去を例に解説するが、プライバシー侵害防止、著作権侵害抑制などの用途にも適用可能である。

有害モデルの学習　有害性を除去したい学習済み言語モデルをベースモデル $\boldsymbol{\theta}_{\text{base}}$ と呼ぶ。まず、逆蒸留したい有害な文章を含むコーパスでベースモデルを学習し、有害モデル $\boldsymbol{\theta}_{\text{toxic}}$ を得る（図 2-1）。

対照的デコーディングによる逆蒸留　ベースモデルと有害モデルを用いた対照的デコーディングにより、有害性を除去するための学習データを逆蒸留する（図 2-2）。式 (1) 及び (2) に示すように、両モデルにより計算される対数確率の差分 $s(\boldsymbol{x})$ をもとに文章をサンプルして、データ $\boldsymbol{x}^*$ を逆蒸留する：

$$s(\boldsymbol{x}) = \log p(\boldsymbol{x};\boldsymbol{\theta}_{\text{base}}) - \log p(\boldsymbol{x};\boldsymbol{\theta}_{\text{toxic}}) \tag{1}$$

$$\boldsymbol{x}^* \sim \text{softmax}(s(\boldsymbol{x})) \tag{2}$$

実際に $s(\boldsymbol{x})$ から文章をそのままサンプルすると、生成される文章は文法性・流暢性を欠く。先行研究 [11, 12] に倣い、式 (4) のように、ベースモデルが低確率とみなすトークンを生成しないようにする。

$$p_{\max} = \max_{x'} p(x'|\boldsymbol{x}_{<t};\boldsymbol{\theta}_{\text{base}}) \tag{3}$$

$$s'(x_t|\boldsymbol{x}_{<t}) = \begin{cases} s(x_t|\boldsymbol{x}_{<t}) & \text{if } p(x_t|\boldsymbol{x}_{<t};\boldsymbol{\theta}_{\text{base}}) \geq \alpha p_{\max}, \\ -\text{inf} & \text{otherwise.} \end{cases} \tag{4}$$

ただし $\alpha \in [0,1]$ はハイパーパラメータで、大きな値を設定するほど確率が上位のトークンのみが生成されるようになり、流暢な文章が生成される。

逆蒸留したデータの学習　生成した文章 $\boldsymbol{x}^*$ でベースモデルを学習し、有害な表現を忘却させる。この学習により、有害モデルとベースモデルの差分で定義される有害性ベクトル $\boldsymbol{\tau}_{\text{toxic}} = \boldsymbol{\theta}_{\text{toxic}} - \boldsymbol{\theta}_{\text{base}}$ に逆らう方向へモデルが更新されることを以下に説明する。一次近似により、式 (1) は下記式で表せる。

$$\begin{aligned} s(\boldsymbol{x}) &\approx (\boldsymbol{\theta}_{\text{base}} - \boldsymbol{\theta}_{\text{toxic}})^\top \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{x};\boldsymbol{\theta}_{\text{base}}) \\ &= (-\boldsymbol{\tau}_{\text{toxic}})^\top \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{x};\boldsymbol{\theta}_{\text{base}}) \end{aligned} \tag{5}$$

ここで、$\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{x};\boldsymbol{\theta}_{\text{base}})$ はベースモデルについて $\boldsymbol{x}$ がもたらす勾配であり、$-\boldsymbol{\tau}_{\text{toxic}}$ との内積が大きい勾配をもたらす文章が重点的に $s(\boldsymbol{x})$ からサンプルされることがわかる。従って、$s(\boldsymbol{x})$ からサンプルされた文章 $\boldsymbol{x}^*$ による勾配は、有害性ベクトルと逆方向になる（図 2-3）。これは、モデルのパラメータから $\boldsymbol{\tau}_{\text{toxic}}$ を差し引くことで有害性を軽減する既存研究 (Task Arithmetic [13]) に類似する。

2.1　既存の学習データ蒸留手法との関係

既存手法の多くは、蒸留されたデータ $\boldsymbol{x}$ による勾配が、元のデータによるパラメータ変化に一致するように、$\boldsymbol{x}$ を勾配降下法で最適化する（勾配マッチング [4, 5]）。学習前のモデルを $\boldsymbol{\theta}$、元のデータで学習したモデルを $\boldsymbol{\theta}^*$ とすると、勾配マッチングでは式 (6) を最大化する $\boldsymbol{x}$ を勾配降下法で得る。

$$f(\boldsymbol{x}) = d(\boldsymbol{\theta}^* - \boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{x};\boldsymbol{\theta})) \tag{6}$$

ここで、d はコサイン類似度や二乗誤差、内積などの類似度指標である。例えば先行研究 [4, 5, 10] では、元のデータ $\boldsymbol{x}_{\text{origin}}$ により 1 ステップ更新した場合（$\boldsymbol{\theta}^* - \boldsymbol{\theta} = \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{x}_{\text{origin}};\boldsymbol{\theta})$）を仮定し、式 (6) を最大化するように $\boldsymbol{x}$ を最適化している。

式 (6) における $\boldsymbol{\theta}^*$ と $\boldsymbol{\theta}$ をそれぞれ $\boldsymbol{\theta}_{\text{toxic}}$、$\boldsymbol{\theta}_{\text{base}}$ とし、d として内積を採用すると、一次近似のもとで、$s(\boldsymbol{x}) = -f(\boldsymbol{x})$ が成り立つ。これは、提案法が学習データ蒸留とは逆に、元の（有害な）学習データがもたらすパラメータ更新と逆向きの勾配を持つ文章を重点的にサンプルすることを示している。

勾配マッチングを含む既存の蒸留手法の多くは、文章ではなく連続値としてデータを蒸留するため、蒸留に用いたモデル以外の学習に蒸留したデータを用いることができない。また、最適化の際に 2 階微分 $\nabla_{\boldsymbol{x}}\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{x};\boldsymbol{\theta})$ の計算を伴うため、大規模なモデル・データでは膨大な計算コストを要する。一方、提案法により蒸留したデータは離散的な文章であり、異なるモデルの学習に利用できる。また、対照的デコーディングを用いることで計算量を大幅に抑え、大規模モデルのコーパスにも適用可能である。学習データ蒸留に関する先行研究 [4, 5, 6] では、あるモデルで蒸留したデータが他のモデルの学習にも有効であることが報告されており、モデル間での汎化が逆蒸留でもみられるか実験にて検証する。

3 評価実験

本節では、逆蒸留で生成された文章を学習させることで、言語モデルの性能を維持しつつ、有害な文章の生成を抑制できるか検証する。本実験では GPT-2 XL を用いて逆蒸留を行い、逆蒸留した文章が、OPT-6.7B、Falcon-7B、LLaMA2-7B など GPT-2 以外のモデルの有害性も除去できるか調べる。

3.1 データセット

本実験では、ヘイトスピーチを格納したデータセットである DGHS[14] を有害モデルの学習に、ToxiGen [15] を検証・評価に用いた。有害モデルの学習に用いられた「性別」「性的指向」「人種」「宗教」の 4 ドメインを in-domain（ID）、用いられていない「障害者」のドメインを out-of-domain（OOD）とし、OOD についても有害性が除去できるか検証する。896 件の検証データでハイパーパラメータを探索し、940 件の評価データにて結果を報告する。また、有害性除去後にモデルの下流タスク性能が維持されるか測るために、MMLU [16] を用いた。

3.2 比較手法

GPT-2 Samples　GPT-2 XL から無作為にサンプルした文章でモデルを学習する。これは式 (1) にて $-\log p(\boldsymbol{x};\boldsymbol{\theta}_{\text{toxic}})$ の項を除いた場合の提案法に相当する。この比較により、提案法で用いている対照的デコーディングの必要性を確認する。

表 1　逆蒸留に用いたモデルの有害性抑制。GPT-2 から逆蒸留した文章を用いて GPT-2 を学習した結果。有害な文章の生成確率（TP）の最小値を太字で示す。

Model	TP (↓)		PPL (↓)	Acc. (↑)
	ID	OOD		MMLU
GPT-2 XL	$0.53_{0.01}$	$0.41_{0.02}$	17.28	32.07
$\text{Samples}_{\text{GPT-2}}$	$0.48_{0.02}$	$0.35_{0.03}$	15.71	32.20
LM-Steer	$0.44_{0.01}$	$0.32_{0.01}$	18.73	29.72
DExperts	$0.50_{0.02}$	$0.35_{0.03}$	18.12	30.83
Task Arithmetic	$0.52_{0.01}$	$0.38_{0.02}$	17.64	29.92
提案法	$\mathbf{0.36}_{0.01}$	$\mathbf{0.28}_{0.02}$	12.23	30.37

LM-Steer [17]　トークン x_t の埋め込み $\boldsymbol{e}(x_t)$ を $\boldsymbol{e}'(x_t) = \boldsymbol{e}(x_t) - \epsilon \boldsymbol{W}_{\text{toxic}} \boldsymbol{e}(x_t)$ と置き換えることで有害な表現の生成を抑制する。$\boldsymbol{W}_{\text{toxic}}$ は有害データから学習した行列、ϵ はハイパーパラメータである。

DExperts [11]　ベースモデルと有害モデルの対照的デコーディングにより有害な表現の生成を抑制する：$x_t \sim (1+\beta) \log p_{\boldsymbol{\theta}_{\text{base}}}(x_t|\boldsymbol{x}_{<t}) - \beta \log p_{\boldsymbol{\theta}_{\text{toxic}}}(x_t|\boldsymbol{x}_{<t})$。ただし、$\beta$ はハイパーパラメータである。

Task Arithmetic [13]　有害モデルとベースモデルのパラメータの差分 $\boldsymbol{\tau}_{\text{toxic}} = \boldsymbol{\theta}_{\text{toxic}} - \boldsymbol{\theta}_{\text{base}}$ をパラメータから直接減算する：$\boldsymbol{\theta}_{\text{detoxed}} = \boldsymbol{\theta}_{\text{base}} - \lambda \boldsymbol{\tau}_{\text{toxic}}$。ただし、$\lambda$ はハイパーパラメータである。

既存手法は有害性を抑制したいモデルごとに $\boldsymbol{\theta}_{\text{toxic}}$ や $\boldsymbol{W}_{\text{toxic}}$ を得る必要がある。一方、提案法は一つのモデルから逆蒸留すれば、生成した文章は他のモデルの学習に転用できるため、汎用性に優れる。ハイパーパラメータの詳細は付録 A で議論する。

3.3 評価指標

有害性抑制　既存研究 [11, 17] にならい、ToxiGen の各サンプルに対し、最大 20 トークンの生成を 25 回行う（nucleus sampling、$p = 0.9$ を使用）。それぞれの生成文について Detoxify [18] で有害スコアを算出し、25 回のうち 1 回でも有害スコアが 0.5 を超えるサンプルの割合を Toxicity Probability (TP) とする。

性能維持　既存研究 [11, 17] と同様に、以下の指標でモデル性能を評価する。1) Perplexity (PPL): 生成された文章の流暢性を LLaMA2-7B によるパープレキシティで測る。2) Accuracy (Acc.): MMLU を用いた few-shot 正解率で下流タスク性能を評価する。各モデルの入力長の制約から、GPT-2 XL は 1-shot、

表 2　逆蒸留に用いていないモデルの有害性抑制。GPT-2 から逆蒸留した文章を用いて各モデルを学習した結果。有害な文章の生成確率（TP）の最小値を太字で示す。

Model	TP (↓)		PPL (↓)	Acc. (↑)
	ID	OOD		MMLU
OPT-6.7B	$0.78_{0.01}$	$0.82_{0.02}$	17.30	34.36
$\text{Samples}_{\text{GPT-2}}$	$0.61_{0.01}$	$0.59_{0.01}$	21.37	34.16
LM-Steer	$0.74_{0.01}$	$0.78_{0.03}$	24.69	30.83
DExperts	$0.62_{0.02}$	$0.65_{0.02}$	28.19	35.40
Task Arithmetic	$0.58_{0.01}$	$0.56_{0.04}$	25.89	30.70
提案法	$\mathbf{0.20}_{0.01}$	$\mathbf{0.14}_{0.02}$	13.57	31.16
Falcon-7B	$0.60_{0.01}$	$0.53_{0.03}$	10.69	39.32
$\text{Samples}_{\text{GPT-2}}$	$0.46_{0.01}$	$0.40_{0.03}$	17.15	34.49
LM-Steer	$0.37_{0.02}$	$0.32_{0.03}$	29.05	34.75
DExperts	$0.30_{0.01}$	$0.25_{0.01}$	28.71	37.88
Task Arithmetic	$0.52_{0.01}$	$0.47_{0.02}$	32.71	29.85
提案法	$\mathbf{0.26}_{0.01}$	$\mathbf{0.22}_{0.02}$	8.78	34.23
LLaMA2-7B	$0.58_{0.01}$	$0.49_{0.02}$	8.56	41.74
$\text{Samples}_{\text{GPT-2}}$	$0.57_{0.02}$	$0.47_{0.02}$	8.37	37.75
LM-Steer	$0.47_{0.03}$	$0.40_{0.03}$	10.18	40.82
DExperts	$0.45_{0.03}$	$0.35_{0.01}$	9.91	39.71
Task Arithmetic	$0.58_{0.01}$	$0.47_{0.03}$	9.39	41.02
提案法	$\mathbf{0.20}_{0.01}$	$\mathbf{0.16}_{0.01}$	9.44	36.25

その他のモデルは 3-shot で正解率を測定する。

3.4　実験結果

逆蒸留に用いたモデルの有害性抑制　GPT-2 自身から逆蒸留した文章を用いて GPT-2 を学習し、有害性を除去した結果を表 1 に示す。異なるシード値について、その平均値と標準偏差を記載した。ID は逆蒸留に用いたドメインを示し、OOD は逆蒸留時に利用していないドメインへの汎化性能を示す。

提案法はベースモデル（GPT-2 XL）の性能を大きく損なわずに有害性を除去できている。一方、既存手法は MMLU の正解率が提案法と同等ながら、提案法より有害性を抑制できていないことがわかる。

逆蒸留に用いていないモデルの有害性抑制　GPT-2 から逆蒸留した文章で、OPT-6.7B、Falcon-7B、LLaMA2-7B といった異なるモデルを学習し、有害性を抑制できるか検証した（表 2）。

提案法はいずれのモデルにおいても高い有害性除去性能を示した。これは、GPT-2 から逆蒸留した文章が当該モデルに過適合せず、モデル間で汎化することを示唆している。一方、既存手法はいずれも有害性を抑制できるものの、モデルによって性能のばらつきが大きい。例えば、Task Arithmetic は OPT では DExperts よりも高い性能を示す一方、LLaMA にはさほど有効ではない。また、LM-Steer は OPT では性能が低いものの、他のモデルでは比較的高い有効性を示している。

表 3　GPT-2 から無作為にサンプルした文章と、逆蒸留により生成された文章の有害性・政治的バイアスの比較。

テキスト	有害性	政治的バイアス		
		左派 (%)	右派 (%)	中立 (%)
GPT-2 XL	$0.008_{0.002}$	50.81	23.31	25.88
逆蒸留	$0.003_{0.001}$	44.56	30.19	25.25

4　逆蒸留した文章の分析

逆蒸留によって生成された文章について、どのような特徴が有害性の抑制に寄与したのか、有害性と政治的バイアスの側面から分析を行った。

有害性　GPT-2 から無作為にサンプルした文章と、逆蒸留した文章について、その有害性スコアを比較する。前節と同様に、生成された文章について Detoxify により有害性スコアの平均と標準偏差を算出した。逆蒸留で生成された文章は、GPT-2 から生成された文章と比較して、有害スコアが低いことがわかる（表 3）。逆蒸留により有害な表現が少ない文章が生成され、それを学習することでモデルが有害な表現を生成しにくくなったことが示唆される。

政治的バイアス　先行研究 [19] では、政治的に偏向した LLM は、有害コンテンツを検知しにくいことが報告されており、政治的バイアスと有害性の有無に関連があることを示唆している。この知見から、PoliticalBiasBERT [20] を用いて、逆蒸留した文章を左派・右派・中立に分類した。表 3 に示すように、GPT-2 からサンプルした文章は左派寄りの傾向が見られる一方、提案法で蒸留された文章は、左派が減少し、右派の割合が増えることで、より政治的に中立的な傾向を示した。逆蒸留により生成された文章を学習することで、LLM の政治的バイアスが中和され、有害性抑制に寄与した可能性がある。

5　おわりに

本稿では、対照的デコーディングを用いてコーパスを逆蒸留する手法を提案し、LLM の有害性抑制に有効であることを示した。本研究が、学習済モデルからの著作権侵害データの除去など、幅広い文脈における LLM の安全性向上に役立つことを期待する。

謝辞

本研究は、NEDO JPNP20006、JST CREST JPMJCR21D1 及び JSPS 科研費 JP23K16940 の支援を受けたものである。

参考文献

[1] Jiahui Geng, Zongxiong Chen, Yuandou Wang, Herbert Woisetschläger, Sonja Schimmler, Ruben Mayer, Zhiming Zhao, and Chunming Rong. A survey on dataset distillation: approaches, applications and future directions. In **Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence**, pp. 6610–6618, 2023.

[2] Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 14389–14408. Association for Computational Linguistics, 2023.

[3] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. Dataset distillation. **arXiv preprint arXiv:1811.10959**, 2018.

[4] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In **International Conference on Learning Representations**, 2020.

[5] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In **International Conference on Machine Learning**, pp. 12674–12685. PMLR, 2021.

[6] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**, pp. 10718–10727, 2022.

[7] Yongqi Li and Wenjie Li. Data distillation for text classification. **arXiv preprint arXiv:2104.08448**, 2021.

[8] Ilia Sucholutsky and Matthias Schonlau. Soft-label dataset distillation and text dataset distillation. In **2021 International Joint Conference on Neural Networks**, pp. 1–8. IEEE, 2021.

[9] Aru Maekawa, Naoki Kobayashi, Kotaro Funakoshi, and Manabu Okumura. Dataset distillation with attention labels for fine-tuning bert. In **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)**, pp. 119–127, 2023.

[10] Aru Maekawa, Satoshi Kosugi, Kotaro Funakoshi, and Manabu Okumura. Dilm: Distilling dataset into language model for text-level dataset distillation. **arXiv preprint arXiv:2404.00264**, 2024.

[11] 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 **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. Association for Computational Linguistics, 2021.

[12] 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 **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 12286–12312. Association for Computational Linguistics, 2023.

[13] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In **The Eleventh International Conference on Learning Representations**, 2023.

[14] Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. Learning from the worst: Dynamically generated datasets to improve online hate detection. 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)**, pp. 1667–1682. Association for Computational Linguistics, 2021.

[15] Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. Toxigen: A large-scale machine-generated dataset for implicit and adversarial hate speech detection. In **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics**, 2022.

[16] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In **Proceedings of the International Conference on Learning Representations (ICLR)**, 2021.

[17] Chi Han, Jialiang Xu, Manling Li, Yi Fung, Chenkai Sun, Nan Jiang, Tarek Abdelzaher, and Heng Ji. Word embeddings are steers for language models. In **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 16410–16430. Association for Computational Linguistics, 2024.

[18] Laura Hanu and Unitary team. Detoxify. Github. https://github.com/unitaryai/detoxify, 2020.

[19] Shangbin Feng, Chan Young Park, Yuhan Liu, and Yulia Tsvetkov. From pretraining data to language models to downstream tasks: Tracking the trails of political biases leading to unfair NLP models. In **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 11737–11762. Association for Computational Linguistics, 2023.

[20] Ramy Baly, Giovanni Da San Martino, James Glass, and Preslav Nakov. We can detect your bias: Predicting the political ideology of news articles. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 4982–4991. Association for Computational Linguistics, 2020.

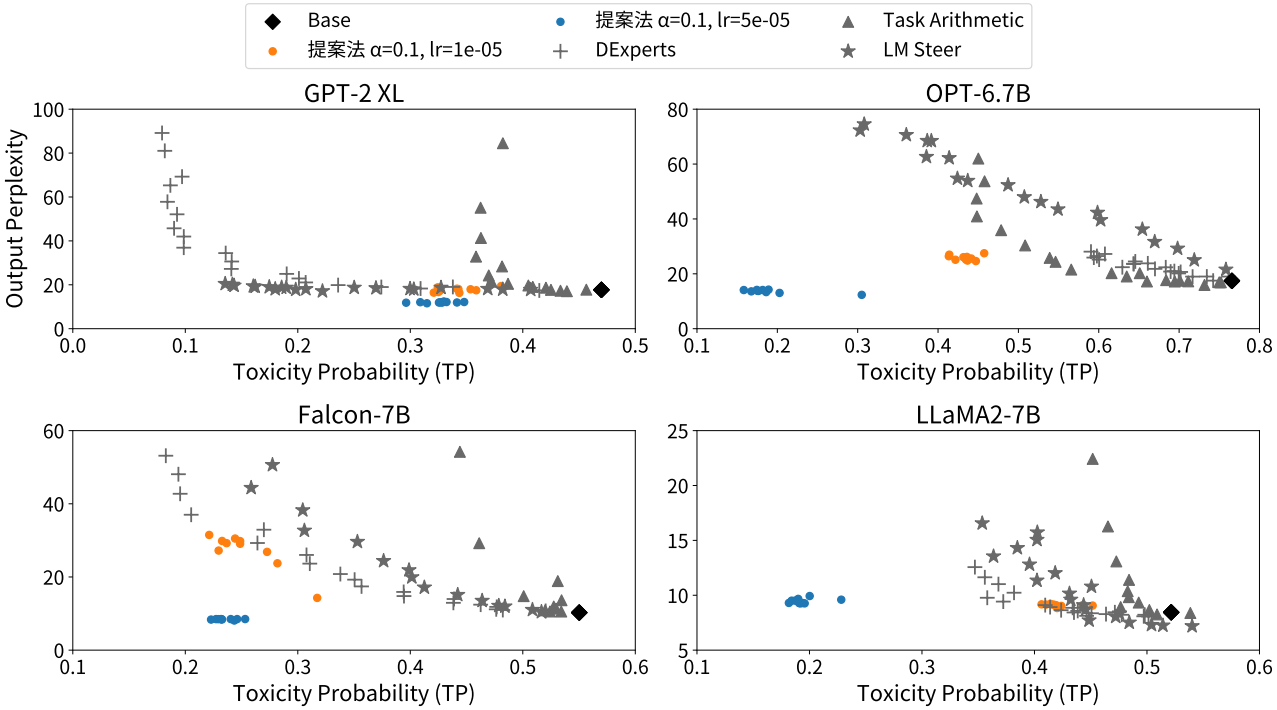

図 3　ハイパーパラメータ感度。異なるハイパーパラメータに対する、Perplexity と Toxicity Probability (TP) の変化を示す。

A　ハイパーパラメータ

ハイパーパラメータ探索　提案法で蒸留したテキストを GPT-2 XL や OPT-6.7B、Falcon-7B、LLaMA2-7B に適用し、学習率 5×10^{-5} または 1×10^{-5} で追加学習を行う。学習率と学習ステップはアラインメント強度を制御するハイパーパラメータとして設定する。GPT-2 Samples についても同様に、GPT-2 XL および他モデルに適用し、学習率 1×10^{-5} 固定で学習ステップをハイパーパラメータとする。提案法および GPT-2 Samples については、ToxiGen の検証セットに含まれる全ドメインの平均 Toxicity Probability (TP) を基準として、GPT-2 XL で最適な追加学習ステップを決定する。蒸留したアラインメント用テキストと決定したハイパーパラメータを、モデルごとの蒸留や探索を行わずに他のモデルにも適用する。

一方、LM-Steer、DExperts、Task Arithmetic については、各モデルごとに個別のハイパーパラメータ探索を行う。具体的には、提案法において観測された最大 PPL より 10 %以内の範囲で、TP を最小化できるハイパーパラメータを探索する。つまり、流暢性を大きく損なわない範囲で最良のアラインメント性能を達成するパラメータを探索している。

ハイパーパラメータ感度　図 3 は、各モデルに対して異なるハイパーパラメータを設定した際の、Perplexity（縦軸）と Toxicity Probability (TP, 横軸) の関係を示しており、それぞれの値を全偏見分野で平均している。提案法の結果はいずれのモデルでも左下に集まっており、流暢性の劣化を最小限に抑えながら高いアラインメント性能を達成しており、提案法がモデルごとのハイパーパラメータ探索を必要とせず、異なるモデルに対して頑健なアラインメントを実現することを示唆する。

これに対し、LM-Steer や DExperts、Task Arithmetic はいずれもモデルごとに変動が大きい。たとえば、LM-Steer を $\epsilon = -1.1 \times 10^{-3}$ という設定で OPT-6.7B に適用すると、Perplexity は 52.35 にまで上昇する一方で、LLaMA2-7B の場合は 10.16 程度に収まる。同様に、DExperts を $\beta = 1.8$ で GPT-2 XL に適用すると、Perplexity が 69.27 に急上昇するのに対し、OPT-6.7B では 25.92 にとどまる。Task Arithmetic はさらにばらつきが大きく、$\lambda = 0.14$ で Falcon-7B では Perplexity が 275.51 に、LLaMA2-7B では 72.77 に上昇するのに対し、OPT-6.7B では 25.81 にとどまる。こうしたばらつきから、同一のハイパーパラメータを異なるモデルに適用することは、著しい性能劣化を引き起こす可能性があることがわかる。