組織を超えたLLM学習データの目的外利用を防げるか

相馬菜生[1]　小林美結[2]　宮田侑佳[2]　倉光君郎[2]
[1]日本女子大学大学院 理学研究科　[2]日本女子大学 理学部
m1916045sn@ug.jwu.ac.jp　kuramitsuk@fc.jwu.ac.jp

概要

高性能な大規模言語モデル（Large Language Model, LLM）の開発には，高品質な学習データの収集が不可欠である．しかし，データ収集には多くの課題が伴い，データ提供者とモデル開発者が異なるケースでは，組織間でのデータ共有が避けられない場合がある．この共有プロセスにおいて，データの不適切な利用や漏洩リスクは重大な課題となる．そのため，データに適切な保護処理を施し，LLMの学習以外での利用を防ぐことが必要不可欠となる．同時に，保護されたデータを用いて構築したLLMにおいて，モデル性能が大幅に低下しないことも求められる．本研究では，保護処理を施したデータを学習データとして使用した場合におけるLLMへの影響を，特にコード生成能力を指標として評価する．

1　はじめに

LLMの性能向上において，学習データは極めて重要な要素である．データの品質がモデルの能力を大きく左右することは広く認識されており [1, 2]，高品質なデータセットの確保が研究や開発の鍵となる．LLMの開発に携わる開発者であれば，他の組織が所有する高品質データを活用したいと考えたことがあるだろう．しかし，このようなデータは，プライバシー保護や知的財産権の観点から厳格な制約の下に管理されており，組織を超えた共有は容易ではない [3]．そのため，次の要件を満たす学習データの保護法が必要となる．

- 学習データをオリジナルデータに復元できない形に黒塗りし，目的外利用を防ぐこと
- 保護されたデータを用いてLLMを構築した際に，モデル性能が顕著に低下しないこと

我々は先行研究として，各トークンの出現頻度に基づきデータを黒塗りする保護法を提案した．この黒塗り手法で保護された事前学習データを使用してLLMを構築し，JGLUE [4] などを用いて言語理解能力を確認した [5]．しかし，より影響を及ぼすと想定される言語生成能力は未評価のままであった．

本研究は，データ保護を強化するため，黒塗りの実装方法を再評価し，さらにLLMのモデル性能に与える影響を生成能力から評価する．我々は，生成能力の評価としてコード生成タスクを採用した．コード生成は，一文字でも正しくない生成を行うとパスしないため厳密に生成能力を見るのに適している．

2　学習データ保護

自然言語処理分野においても，機密性の保証や個人情報保護の観点から，テキストデータに対する保護技術が広く研究されてきた．LLM開発の需要が高まるにつれ，新たな課題も含めて，よりデータ保護技術の重要性 [6] が高まっている．以下，我々の提案手法に関連するプライバシー保護，匿名化に関する研究を概観する．

テキストデータの匿名化は，機密情報を保護しつつデータの利活用を可能にするための重要な研究領域である．Lisonらは，テキストデータの匿名化における現状，課題，および将来的な方向性を包括的に調査しており，その中で匿名化手法の一例として，NLP技術を活用した手法を挙げている [7]．

NLP手法とは，NLP技術を活用した手法であり，名前付きエンティティ認識 (NER) [8] やシーケンスラベリングによって直接識別可能な情報 (e.g. 名前，住所) を検出・削除する．これらの手法は，特定のカテゴリ (e.g. Protected Health Information, PHI) に依存する形で設計されており，ドメインや言語が異なる場合には適応が難しいという課題がある [9]．

このように様々なテキスト保護法が提案されてきたが，基本的に対人間向けの保護法を想定したもの

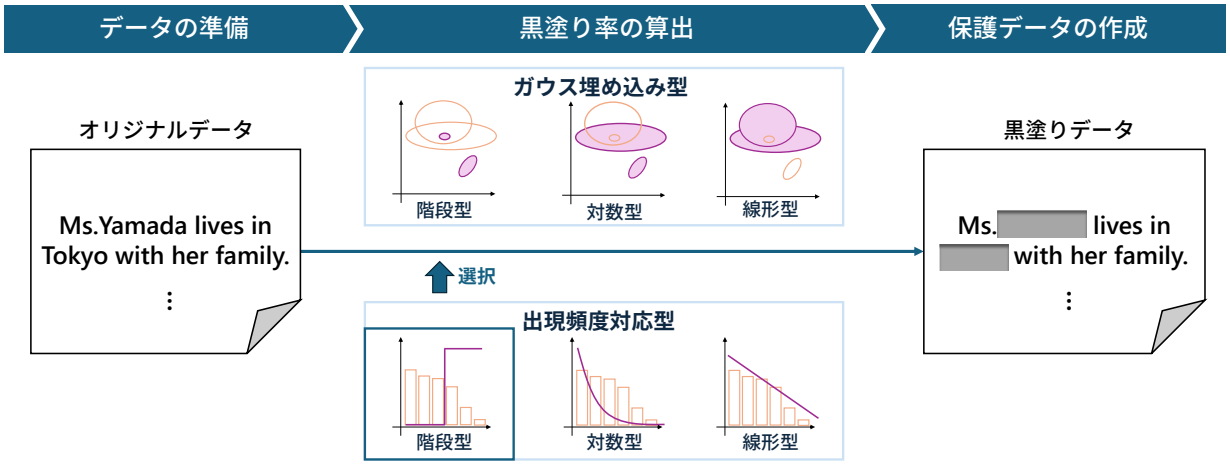

図 1　統計的黒塗りデータ保護手法の概要図

であり，LLM で学習したときの影響については十分に検証されていない．本研究では，LLM でデータを学習することを前提に独自の保護法を提案する．

3　統計的黒塗りデータ保護

本節では，統計的黒塗りデータ保護手法を提案する (図 1)．我々のアイデアとしては，先行研究 [5] で示されているが，今回, データ保護の強化の観点で改良されている．

3.1　黒塗りとは

黒塗りとは，機密文書を公開するとき，文の一部を黒塗りにして見えないように編集することに由来している．我々は，同様に，学習データの一部を黒塗りすることで，開発者側に知られたくない情報を隠すことができる．

黒塗りの実現方法としては，マスクとドロップが考えられる．

- **マスク**: LLM の事前学習手法の一つである MLM を参考に黒塗り箇所をマスクトークンに置き換える手法
- **ドロップ**: 黒塗り箇所を完全に削除する手法

先行研究では，マスクを採用した．これは，過去の BERT [10] や T5 [11] など LLM の事前学習で採用されており，学習効果に対する悪い影響よりむしろ良い影響が期待できるためである．しかし，CLM の事前学習においてマスクはほとんど使用されていない．また，マスクは黒塗りにした箇所がマスクトークンで明らかになるため，より復元しやすいという課題もある．本研究では，マスクの代わりにより復元が難しいドロップを採用した．ただし，黒塗り自体はマスクであれドロップであれ，どちらを採用しても同じように適用することができる．

3.2　統計的な黒塗り

LLM の事前学習では，基本的にマスク箇所はランダムに決定されてきた．我々の黒塗り手法は，目的の保護を実現するため，より統計的に黒塗りする箇所を決定する．

我々は，学習データにおける各トークンの出現頻度に基づいて，黒塗りする字句を決定した (出現頻度対応型)．

我々は，トークン出現頻度と与えられた黒塗り率 [5] に基づいて，黒塗り箇所を決定した．

- **線形型**: 各トークンの出現回数に関わらず，出現頻度の高いトークンから線形状に黒塗り率を減少させる．
- **対数型**: 各トークンの出現回数を考慮し，助詞や句読点のような極端に出現頻度の高いトークンの黒塗り率を対数状に高くする．
- **階段型**: 各トークンの出現回数に関わらず，出現頻度の低いトークンに一律で高い黒塗り率を設定し，階段関数を示すようにする．

3.3　ガウス埋め込み

我々は，黒塗り箇所をより統計的かつ機密情報や個人情報を選択的に選ぶため，ガウス埋め込み (Gaussian Embedding) に着目した．

ガウス埋め込みとは，トークンの意味を表現空間のベクトルとして表現するだけでなく，その意味の広がりをガウス分布の分散として表現する手法である [12]．この手法を用いることにより単語の意味

の広がりや含意関係を捉えた表現を構成できる．その特性を利用して，様々な応用が検討されている [13, 14]．

本研究では，ガウス埋め込みによって得られる単語の共分散行列のトレース (対角和) を元に，以下の 3 つの型でトークンを黒塗りする確率を定めている．

- **線形型**: 抽象的な単語 (=トレース値の高い単語) に高い黒塗り率をかけ，線形状に黒塗り率を減少させる．
- **対数型**: 抽象的な単語に高い黒塗り率をかけ，トレース値に対応しながら黒塗り率を減少させる．
- **階段型**: 具体的な単語 (=トレース値の低い単語) に極端に高い黒塗り率をかける．

4 実験

データ保護が LLM の性能に与える影響を評価するため，3 節で説明した方法でデータ保護を行った学習データを用いて LLM を構築し，そのコード生成能力について評価を行う．

黒塗り箇所の決定には，線形型，対数型，階段型に加え，全てのトークンにおいて黒塗り率を一定にして決定する一定型を用いた．また，今回は，黒塗り手法としてドロップを用いる．

4.1 実験設定

評価実験では，LLM の性能向上手法として広く採用されている指示調整を対象に実験を行った．ベースとなる LLM には，Qwen2.5-0.5B[1] を採用した．このモデルは 18T トークンの大規模なデータで事前学習済みの多言語対応モデルである．

また，指示調整データセットとして，OpenCoder の educational-instruct データセット[2] を用いた．このデータセットは，アルゴリズム関連のコーパスを基に生成された 11 万件の指示調整データセットであり，主にコード生成能力の向上を目的としている．

ファインチューニングの設定は文献 [15] を参考に，コサインスケジューラを使用して学習率 5e-5 で 3 エポックの学習を実施した．

1） https://huggingface. co/Qwen/Qwen2.5-0.5B
2） https://huggingface.co/datasets/OpenCoder-LLM/opc-sft-stage2

4.2 ドロップデータの作成

3 節で示したデータ保護手法に基づき，ドロップデータを作成した．本実験ではデータの全体トークン数に対して 5%をドロップすることとした．作成手順は以下の通りである．

1. ベースの LLM の語彙 (vocab) に含まれる各トークンの出現頻度及びガウス埋め込みのトレース値を算出
2. 1 で算出した出現頻度及びトレース値を基に，各トークンのドロップ率を決定
3. educational-instruct データにおける instruction と output を対象に，指定のドロップ率に従いトークンを削除

4.3 評価

LLM 性能の評価方法と黒塗りされたデータの保護強度について説明する．

4.3.1 LLM 性能

LLM の性能は，指示調整後のコード生成性能をベースライン LLM と比較することで評価した．本実験では，コード生成性能の指標として HumanEval の pass@1 を用いた．HumanEval [16] とは，OpenAI 社が Codex を発表する際に使用した評価基準で，ソフトウェアテストに基づいて生成されたコードを実行ベースで評価する．

4.3.2 復元困難性

データ保護強度は，ドロップされたデータからオリジナルデータへの復元困難性を，データ間の類似度として測定した．復元困難性が高いほど，データ保護の効果が高いとみなす．そのため，本研究では，編集距離と Jaccard 係数を用いて類似度を測定した．編集距離は，1 つの文字列を別の文字列に変換する際に必要な最小の操作回数 (挿入，削除，置換) で測定される指標である．Jaccard 係数は，2 つの集合間での共通要素の割合で測定される指標である．これらの類似度は値が 1 に近いほど類似度が高いと評価する．

4.4 実験結果

実験結果を表 1 に示す．表 1 より，ドロップデータによるファインチューニングにおいて，オリジナ

表 1　データ保護手法の比較結果

Model	指示調整	ドロップ手法	ドロップ率の変化	HumanEval	類似度	
					編集距離	Jaccard
Qwen2.5 (basemodel)	-	-	-	28.66	-	-
+educational-instruct (ベースライン)	有	無		37.20	1.0000	1.0000
+educational-instruct	有	一定		35.37	0.9807	0.9745
+educational-instruct	有	出現頻度対応型	線形型	34.15	0.9792	0.9724
+educational-instruct	有		対数型	34.76	0.9811	0.9750
+educational-instruct	有		階段型	29.88	0.9746	0.9672
+educational-instruct	有	ガウス埋め込み型	線形型	34.15	0.9832	0.9794
+educational-instruct	有		対数型	34.15	0.9811	0.9753
+educational-instruct	有		階段型	34.76	0.9535	0.9151

ルデータを学習した LLM(ベースライン) と比較しても，HumanEval のスコアに関して大幅な低下が抑えられていることが確認できた．

さらに，出現頻度対応型 (階段型) の手法では類似度が低い値であることから，頻度の低いトークンを優先的にドロップする設計がデータ保護強度の向上に寄与していることが示された．また，ガウス埋め込み型 (階段型) の手法は，データ間の類似度が最も低くなる一方で，性能低下を最小限に抑える結果を示した．

類似度のみでデータ保護の強度を正確に測ることは困難ではあるが，頻度が低いトークンを優先的にドロップする出現頻度対応型 (階段型) や，具体性の高いトークンをドロップするガウス埋め込み型 (階段型) はいずれも高いデータ保護強度を持つと予測される．特に，ガウス埋め込み型 (階段型) が最も低い類似度を示していることから，類似度を用いた評価が一定の有効性を持つと仮定できる．

5　関連研究

本研究は，LLM の学習に使用するデータを黒塗りすることによってデータを保護することを目的としているが，LLM からの出力に関するセキュリティに注目した研究として，M.Abadi らの研究が挙げれられる [17]．M.Abadi らは確率的勾配降下法においてパラメータ更新の際，勾配にノイズを加える方法を提案し，それが差分プライバシー [18] を達成することを示した．本研究では，学習データを黒塗りすることで学習データの保護を図っているのに対し，M.Abadi らの研究ではニューラルネットワークの勾配にノイズを加えることで，ニューラルネットワークからの出力に関するプライバシーを保護している．これらの 2 つの手法はノイズを与える場所がそれぞれ異なっており，組み合わせて使うことが可能であると考えられる．

6　むすびに

本研究では，LLM の学習データを保護するためにデータに黒塗りを施した場合，LLM の性能にどのような影響が出るのかについて評価実験を行った．データの黒塗りにあたっては，出現頻度とガウス埋め込みの二つの指標を用いてドロップさせるトークン選定・学習データの復元困難性を高めつつ，LLM の性能を維持することを目指した．

実験の結果，ドロップ手法が高いデータ保護効果を示すとともに，LLM の性能劣化を抑える可能性を示唆した．特に，ガウス埋め込みを利用した欠損選定は，語彙の文脈的意味を考慮できる点で有用であり，プライバシーや機密性を重視したデータ利用において実践的価値があると考えられる．

今後は，統計的黒塗りデータ保護手法として，線形，対数，階段型以外の方法を新たに開拓し，データ保護とモデル性能の最適化を目指す．また，本研究では評価のしやすさ，性能向上の観点から，コードを学習データとして採用したが，今後は他のデータにも適用していきたい．

謝辞

本研究は JSPS 科研費 JP23K11374 の助成を一部受けたものです.

参考文献

[1] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need, 2023.

[2] Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. **arXiv preprint arXiv:2406.17557**, 2024.

[3] Da Yu, Sivakanth Gopi, Janardhan Kulkarni, Zinan Lin, Saurabh Naik, Tomasz Lukasz Religa, Jian Yin, and Huishuai Zhang. Selective pre-training for private fine-tuning. **arXiv preprint arXiv:2305.13865**, 2023.

[4] Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. Jglue: Japanese general language understanding evaluation. In **Proceedings of the Thirteenth Language Resources and Evaluation Conference**, pp. 2957–2966, 2022.

[5] 小柳響子, 小林美結, 相馬菜生, 倉光君郎. ノイズ付与による llm 事前学習データセットの保護. コンピュータセキュリティシンポジウム 2024 論文集, pp. 289–294, 10 2024.

[6] Biwei Yan, Kun Li, Minghui Xu, Yueyan Dong, Yue Zhang, Zhaochun Ren, and Xiuzhen Cheng. On protecting the data privacy of large language models (llms): A survey. **arXiv preprint arXiv:2403.05156**, 2024.

[7] Pierre Lison, Ildikó Pilán, David Sánchez, Montserrat Batet, and Lilja Øvrelid. Anonymisation models for text data: State of the art, challenges and future directions. 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. 4188–4203, 2021.

[8] Behrang Mohit. **Named Entity Recognition**, pp. 221–245. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[9] Tzvika Hartman, Michael D Howell, Jeff Dean, Shlomo Hoory, Ronit Slyper, Itay Laish, Oren Gilon, Danny Vainstein, Greg Corrado, Katherine Chou, et al. Customization scenarios for de-identification of clinical notes. **BMC medical informatics and decision making**, Vol. 20, pp. 1–9, 2020.

[10] Kenton Lee Kristina Toutanova Jacob Devlin, Ming-Wei Chang. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018.

[11] Adam Roberts Katherine Lee Sharan Narang Michael Matena Yanqi Zhou Wei Li Peter J. Liu Colin Raffel, Noam Shazeer. Exploring the limits of transfer learning with a unified text-to-text transformer. **arXiv preprint arXiv:1910.10683**, 2019.

[12] Luke Vilnis and Andrew McCallum. Word representations via gaussian embedding. 2015.

[13] 時武孝介, 村脇有吾, 黒橋禎夫. ガウス埋め込みに基づく単語の意味の史的変化分析. 言語処理学会 第 24 回年次大会 発表論文集, 2018.

[14] Shohei Yoda, Hayato Tsukagoshi, Ryohei Sasano, and Koichi Takeda. Sentence representations via gaussian embedding. **arXiv preprint arXiv:2305.12990**, 2023.

[15] Siming Huang, Tianhao Cheng, Jason Klein Liu, Jiaran Hao, Liuyihan Song, Yang Xu, J Yang, JH Liu, Chenchen Zhang, Linzheng Chai, et al. Opencoder: The open cookbook for top-tier code large language models. **arXiv preprint arXiv:2411.04905**, 2024.

[16] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. **arXiv preprint arXiv:2107.03374**, 2021.

[17] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In **Proceedings of the 2016 ACM SIGSAC conference on computer and communications security**, pp. 308–318, 2016.

[18] Cynthia Dwork. Differential privacy. In **International colloquium on automata, languages, and programming**, pp. 1–12. Springer, 2006.