日本語継続事前学習モデルを対象とした暗記の定量化

高橋寛武 [1*] 石原祥太郎 [2]
[1] 独立研究者 [2] 株式会社日本経済新聞社
hiromu.takahashi56@gmail.com shotaro.ishihara@nex.nikkei.com

概要

大規模言語モデルによる訓練データの暗記の懸念に注目が集まる一方，非英語や産業界のコーパスを用いる条件下での分析は十分に進んでいない．そこで本研究では，日本語特化の大規模言語モデル構築で一般的な継続事前学習に着目し，訓練データに対する暗記の定量化に取り組む．具体的には，Llama 3 に対して Wikipedia と日本語ニュースメディア「日経電子版」の記事データを用い，2 種類の継続事前学習モデルを構築・分析した．実験では英語の実証的知見と同様，学習が進むごとに暗記量が増える傾向が観測された．この傾向は日経電子版の場合により明確で，一般的でない産業界のコーパスを用いる際の懸念が示唆された．

1 はじめに

大規模言語モデルの実用性が高まるにつれ，事前学習に用いられた訓練データの暗記に関する懸念が浮上している [1]．暗記は訓練データと同じ，または類似の文字列が出力される現象を指す．大規模言語モデルの暗記はプライバシや著作権の侵害に繋がる他，汎用性の低下も引き起こす可能性がある．プライバシについて，最初期の研究である Carlini et al. [2] は，GPT-2 から暗記された個人情報が抽出できると警鐘を鳴らした．著作権に関して Lee et al. [3] は剽窃の観点から GPT-2 を分析し，実用面での倫理的な問題を指摘した．汎用性については，大規模言語モデルが評価ベンチマークを暗記することで評価の正当性が損なわれるとの議論がある [4]．

大規模言語モデルの暗記に関する研究は，主に英語の一般的なコーパスを対象として精力的に進んでいる．訓練データが明らかになっている**オープン**な設定では，訓練データの一部をプロンプトとして与えた際の続きの生成結果を用いる枠組みが一般的である（図 1 上）．実証的知見として，暗記が (1) 訓練データ内の文字列の重複 (2) モデルサイズ (3) プロンプト長の 3 つと強く関連すると報告されている [5]．メンバーシップ推論 [6] 手法を用いて，テキストが訓練データに含まれるか分からない**クローズド**な設定で検出を試みる研究も盛んである（図 1 下）．Shi et al. [7] は Wikipedia の日付情報を用いた評価ベンチマークを構築すると共に，トークンごとの生成確率を用いた予測手法 MIN-K% Prob を提案した．

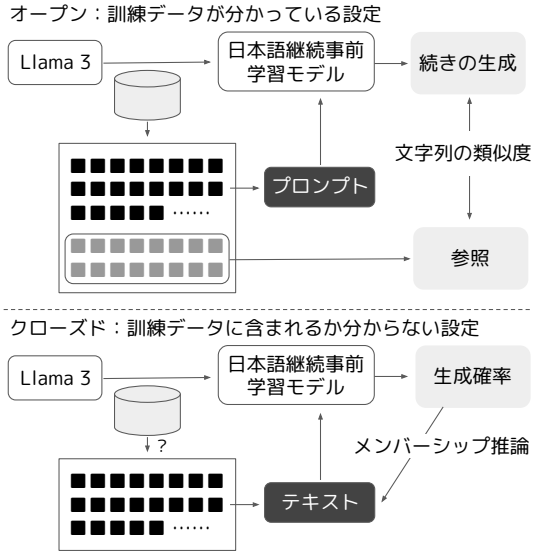


図 1: 本研究では，日本語コーパスを用いて継続事前学習された大規模言語モデルに対し，訓練データの暗記の定量化を試みる．訓練データが分かっているオープンな設定（上）では，訓練データをプロンプトと参照用に分割し，続きの生成結果と参照用の類似度を算出する．テキストが訓練データに含まれるか分からないクローズドな設定（下）では，生成確率を用いて訓練データに含まれる可能性を予想する．

非英語や産業界のコーパスを用いた大規模言語モデルの構築も進み，同時にそれらを対象とした暗記に関する議論の需要も高まっている．日本語に関しては日本語に強いモデルの開発を目的に，学術界主導で組織横断プロジェクト LLM-jp [8] が始動した．Kiyomaru et al. [9] は LLM-jp で構築された GPT-2 モデルを分析し，英語での実証的知見が日本語でも再現されると確認した．産業界では Ishihara et al. [10]

* 株式会社日本経済新聞社での業務委託

表 1: 日本語を用いた先行研究と本研究での問題設定.

	対象のモデル	
問題設定	事前学習	継続事前学習
オープン	一般的なコーパス [9] 産業界のコーパス [10]	本研究 §2.1
クローズド	一般的なコーパス [11] 産業界のコーパス [10]	本研究 §2.2

が日本語ニュースメディア「日経電子版」の記事データを用いて GPT-2 モデルを事前学習し，独特な文体の日本語でも英語での実証的知見が再現されると報告している．クローズドな設定でも，メンバーシップ推論手法を用いることで AUC 0.60 程度の性能が出ると実証した．日本語・英語で，メンバーシップ推論手法の性能を比較した研究 [11] もある．

しかし，非英語や産業界のコーパスを用いる条件下での主要な手法である**継続事前学習** [12] では，暗記が十分に調査されていない．継続事前学習では，事前学習済みモデルを起点に追加学習することで，比較的小規模のコーパスでもドメイン特化の優れたモデルを構築できる [13, 14]．このように微調整されたモデルの暗記に関する既存研究の主な対象は英語 [15, 16, 17, 18] で，限られた日本語を対象とする暗記の研究 [9, 10, 11] では議論されていない（表 1）．

そこで本研究では，一般的な日本語と産業界のコーパスの 2 種類を用いて日本語継続事前学習モデルを構築し暗記を定量化する（§2）．具体的には日本語の一般的なウェブコーパスである Wikipedia と独自の表記規定を持つ日経電子版の記事データを用い，Llama 3 [19] を Low-Rank Adaptation (LoRA) [20] で継続事前学習したモデルを構築[1)]・分析した（§3）．分析の結果，日本語継続事前学習の設定でも部分的に英語の実証的知見が再現され，特に日経電子版を用いた場合に顕著であると分かった（§4）．

2 問題設定

本節では，本研究で取り組む暗記の定量化の手順や手法を説明する．本研究では，データセットの暗記に関する解説論文 [22] による体系的な分類に従い，オープン・クローズドの両者の設定で検証する．

2.1 オープンな設定

手順 訓練データが分かっている設定では，先行研究 [2, 9, 10] と同様に暗記を定量化する．

1. 学習済みのモデルと，訓練データを用意する．
2. 学習に用いた訓練データから，評価データを抽出する．評価データ内の各テキストはプロンプトと参照用に分割する．
3. プロンプト用のテキストを与え，モデルに続きを生成させる．貪欲法を用い，一つのプロンプトから一つの文字列を生成する．
4. 生成結果と参照用のテキストを比較し，逐語暗記・近似暗記の度合いを算出する．

暗記の定義 日本語を対象とした先行研究 [10] での 2 つの定義を利用する．具体的には，大きいほど暗記量が多い値として次のように定量化した．

逐語暗記 前方一致の文字数

近似暗記 1 -（編集距離 / 文字列の長さ[2)]）

逐語暗記は多くの先行研究 [2, 5] を参考に，文字列の部分一致に基づく定義を採用する．加えて，文字列の類似性を考慮した近似暗記の定義 [23, 24] も用いる．単語間に空白がない日本語を扱う点を考慮し，類似度は文字単位の編集距離 [25] で算出する．

2.2 クローズドな設定

手順 この設定では，学習済みの大規模言語モデルを用いて与えられたテキストの生成確率を算出し，訓練データに含まれるかをメンバーシップ推論手法で予測する．

1. 学習済みのモデルを用意する．
2. 評価データ用に訓練データの一部から正例，訓練データに含まれないコーパスから負例を選ぶ．評価データ内の各テキストからは，一定の長さのテキストを抽出する．
3. モデルを用いてテキストの生成確率を算出し，メンバーシップ推論手法で予測を実行する．
4. 予測値と 2 値の正解ラベルから，AUC などの評価指標を算出する．

メンバーシップ推論手法 本研究では，以下の 5 つの手法を検証する．より詳しい手法の説明は，Appendix A に示す．性能は，先行研究 [7, 10, 11] に従い AUC で評価する．

LOSS 損失（負の対数尤度）が閾値より小さい場合，訓練データに含まれると判定 [26].

PPL/zlib Perplexity（PPL）と，zlib 圧縮を通じて計算される情報量の比 [2].

Min-K% Prob 生成確率の低い K % のトークンの

1) 計算量の削減のため，先行研究 [14, 17, 21] と同様に，継続事前学習には LoRA を用いた．

2) 比較する 2 つの文字列の長い方.

みを用いた際の平均の対数尤度 [7].

Min-K%++ 生成確率の正規化と標準化で Min-K% Prob を改善 [27].

ReCaLL 訓練データに含まれないテキストをプロンプトに追加した際の対数尤度の変化率 [28].

3 実験

本節では，実験の実行内容と結果を報告する．

3.1 データセット

本研究では，一般的な日本語として Wikipedia，産業界のコーパスとして日経電子版の記事データを用いた．共通処理としてクローズドな設定では，MeCab [3] を用いてプロンプトを単語単位に分割し，冒頭から {32, 64, 128, 256} 単語を抽出して 4 種類の入力を作成した．辞書は mecab-ipadic-NEologd [29] の 2020 年 9 月 10 日時点版を用いた．

Wikipedia 2023 年 7 月 20 日版の日本語 Wikipedia が前処理されたデータセット [4] を訓練データ（約 13 億トークン）として用いた．オープンな設定では，重複する形で 1000 記事を評価データとして選び，各記事の最初の 200 文字をプロンプト，続きを参照用に分割した．クローズドな設定の負例として，学習に用いない検証データから 1000 記事を抽出した．

日経電子版 2010〜2022 年公開の記事から重複排除などの前処理を経て，訓練データ（約 7 億トークン）を構築した．オープンな設定では，重複する形で 1000 記事を評価データとして選んだ．日経電子版では例外 [5] を除き「冒頭 200 文字」か「記事全体の文字数の半分」の短い方が公開部分，残りが有料会員限定の非公開部分として定義されている．本研究では公開部分をプロンプト，非公開部分を参照用に利用した．クローズドな設定の負例として，2023 年公開の記事から 1000 記事を抽出した．

3.2 モデルの継続事前学習

Wikipedia と日経電子版から構築した訓練データを用いて，Llama 3 の指示学習済み 8B モデル [6] を

3) https://taku910.github.io/mecab/
4) https://gitlab.llm-jp.nii.ac.jp/datasets/llm-jp-corpus-v3/-/tree/main/ja/ja_wiki
5) ニュースの公共性などさまざまな事情に応じて非公開部分も含めた記事全体を公開している場合がある．
6) meta-llama/Meta-Llama-3-8B-Instruct

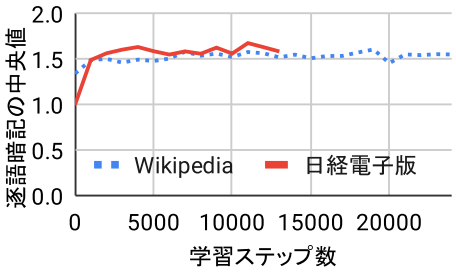


図 2: 学習ステップごとの逐語暗記の中央値の推移．

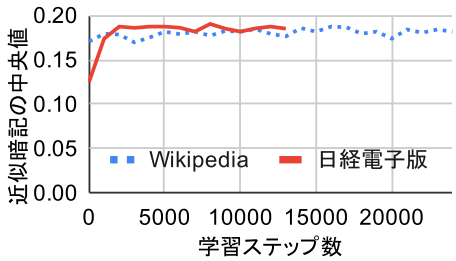


図 3: 学習ステップごとの近似暗記の中央値の推移．

LoRA でそれぞれ継続事前学習した．LoRA のランク数は 16 で 4 エポック学習し，1000 ステップごとに重みを保存した．最終的な検証データに対する損失は，Wikipedia で 1.97，日経電子版で 1.96 とほぼ同じだった．学習設定の詳細や損失の推移は，Appendix B に示す．

3.3 実験結果：オープンな設定

学習ステップ数が 0 の Llama 3 に比べて，特に日経電子版で大きく暗記量が増加していく傾向が読み取れた．継続事前学習のステップ数ごとの逐語暗記・近似暗記の中央値の推移を，それぞれ図 2・3 に示す．たとえば日経電子版では，Llama 3 での逐語暗記が最大 15 文字だったが，学習が進むにつれ最大 27 文字の一致が確認された．Wikipedia でも Llama 3 に対し 4 エポックの継続事前学習をする中で，比較的緩やかではあるが，逐語暗記の中央値が 1.34 から 1.53，近似暗記の中央値が 0.17 から 0.18 に増加した．逐語暗記が最大だった生成例は，Appendix C に示す．

3.4 実験結果：クローズドな設定

学習が進むごとに性能が高まる傾向が観測された．表 2 に，各手法・学習ステップ数・入力単語数での AUC の性能を示す．Min-K% Prob と Min-K%++では，先行研究 [7] を参考に K=20 と設定した．全般に ReCaLL や LOSS で良い性能が出ている．特に日経電子版では学習が進むごとに性能が高まり，入力単

表 2: メンバーシップ推論の各手法・学習ステップ数・入力単語数での AUC．太字は各列での最良を示す．

手法	継続学習のステップ数	Wikipedia 32	Wikipedia 64	Wikipedia 128	Wikipedia 256	日経電子版 32	日経電子版 64	日経電子版 128	日経電子版 256
LOSS	0	0.506	0.490	0.462	0.465	0.504	0.515	0.528	0.526
	1000	0.518	0.512	0.485	0.484	**0.641**	0.642	0.640	0.578
	12000	0.515	0.514	0.479	0.486	**0.641**	0.647	0.650	0.590
	24000	0.513	0.512	0.478	0.485	-	-	-	-
PPL/zlib	0	0.485	0.479	0.470	0.491	0.491	0.502	0.516	0.535
	1000	0.498	0.494	0.484	0.503	0.638	0.642	0.641	0.595
	12000	0.497	0.498	0.490	0.504	0.635	0.648	0.647	0.601
	24000	0.494	0.497	0.489	0.503	-	-	-	-
Min-K% Prob (K=20)	0	0.481	0.493	0.527	**0.535**	0.514	0.488	0.467	0.485
	1000	0.431	0.441	0.475	0.496	0.381	0.382	0.383	0.439
	12000	0.432	0.441	0.483	0.492	0.387	0.379	0.376	0.426
	24000	0.433	0.441	0.484	0.491	-	-	-	-
Min-K%++ (K=20)	0	0.486	0.496	0.522	0.513	0.502	0.489	0.482	0.482
	1000	0.425	0.420	0.443	0.447	0.494	0.514	0.491	0.473
	12000	0.424	0.419	0.456	0.450	0.513	0.531	0.517	0.497
	24000	0.425	0.419	0.451	0.449	-	-	-	-
ReCaLL	0	0.561	0.502	0.483	0.437	0.484	0.535	0.546	0.542
	1000	**0.613**	**0.605**	**0.569**	0.520	0.611	0.651	0.572	**0.630**
	12000	0.608	0.569	0.460	0.494	0.637	**0.660**	**0.689**	0.603
	24000	0.601	0.560	0.454	0.484	-	-	-	-

語数が小さい際に LOSS，大きい際に ReCaLL が最良の結果だった．一方で Min-K% Prob や Min-K%++ では AUC が 0.50 未満で，良い結果が確認できなかった．

4 考察

本節では先行研究を踏まえて，本研究で得られた知見を考察する．

英語での実証的知見の再現 英語での実証的知見として，暗記は (1) 訓練データ内の文字列の重複 (2) モデルサイズ (3) プロンプト長と関連する [5]．(1) について本研究のオープンな設定では，学習が進む[7]につれ暗記量が増えている．クローズドな設定でも ReCaLL や LOSS で学習ステップ数が多いほど性能が高い傾向があった．(2) について，Llama 3 の 8B モデルを用いた本研究での日経電子版での暗記量は，類似の設定で GPT-2 の 0.1B モデルを用いた場合 [10] より多かった．たとえば 30 エポック学習した 0.1B モデルの近似暗記の中央値は約 0.15 だったが，8B モデルの場合は 1000 ステップ（約 0.25 エポック）で同水準を超え，最終的には 0.20 に近づいている（図 3）．(3) については，本研究ではプロンプト長がほぼ同一のため分析の対象外とした．

日本語特有の傾向 クローズドな設定に関して，日本語の場合に Min-K% Prob は K が大きい方が性能が高いという報告 [11] がある．本研究でも Min-K% Prob や Min-K%++に比べ，全トークンの生成確率を用いる LOSS の方が優れた結果となり，先行研究の報告を支持する形となった．

一般的でない産業界のコーパスの暗記 オープン・クローズドな設定の両者で，Wikipedia に比べ日経電子版で顕著に，学習が進むごとの顕著な変化が観測された．要因の一つとして，Llama 3 の事前学習に用いられた訓練データとの性質の違いが考えられる．日経電子版には独特の表記規定があり，ウェブ上で収集される大規模コーパスに含まれる類似のテキスト量は相対的に少ない．特徴的なコーパスを用いた継続事前学習で，大規模言語モデルがより効率的に訓練データを暗記する可能性が示唆された．この現象は，一般的でない産業界のコーパスで大規模言語モデルを微調整する危険性に警鐘を鳴らすものである．

5 おわりに

本研究では，Wikipedia と日経電子版を用いた日本語継続事前学習モデルを構築し，訓練データの暗記の定量化に取り組んだ．今後の展望として，LoRA のランク数の変更や全層を対象とした継続事前学習など，様々な学習設定を考えている．日本語特有の傾向を踏まえた手法の改良といった議論も進めていく必要がある．

7) 同じ訓練データで学習を繰り返し文字列の重複が増える．

謝辞

本稿を丁寧にレビューしてくださった日本経済新聞社の大村和正さんにお礼申し上げます.

参考文献

[1] Shotaro Ishihara. Training data extraction from pre-trained language models: A survey. In **Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)**, pp. 260–275, Toronto, Canada, July 2023. Association for Computational Linguistics.

[2] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In **30th USENIX Security Symposium (USENIX Security 21)**, pp. 2633–2650. USENIX Association, August 2021.

[3] Jooyoung Lee, Thai Le, Jinghui Chen, et al. Do language models plagiarize? In **Proceedings of the ACM Web Conference 2023**, WWW '23, p. 3637–3647, New York, NY, USA, 2023. Association for Computing Machinery.

[4] Inbal Magar and Roy Schwartz. Data contamination: From memorization to exploitation. In **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)**, pp. 157–165, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[5] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, et al. Quantifying memorization across neural language models. In **The Eleventh International Conference on Learning Representations**, 2023.

[6] Reza Shokri, Marco Stronati, Congzheng Song, et al. Membership inference attacks against machine learning models. In **2017 IEEE Symposium on Security and Privacy (SP)**, pp. 3–18, 2017.

[7] Weijia Shi, Anirudh Ajith, Mengzhou Xia, et al. Detecting pretraining data from large language models. In **The Twelfth International Conference on Learning Representations**, 2024.

[8] Akiko Aizawa, Eiji Aramaki, Bowen Chen, et al. LLM-jp: A cross-organizational project for the research and development of fully open japanese LLMs. **arXiv [cs.CL]**, July 2024.

[9] Hirokazu Kiyomaru, Issa Sugiura, Daisuke Kawahara, et al. A comprehensive analysis of memorization in large language models. In **Proceedings of the 17th International Natural Language Generation Conference**, pp. 584–596, Tokyo, Japan, September 2024. Association for Computational Linguistics.

[10] Shotaro Ishihara and Hiromu Takahashi. Quantifying memorization and detecting training data of pre-trained language models using Japanese newspaper. In **Proceedings of the 17th International Natural Language Generation Conference**, pp. 165–179, Tokyo, Japan, September 2024. Association for Computational Linguistics.

[11] 小柳響子, 佐藤美唯, 梶浦照乃ほか. LLM の事前学習データ検知法の日英比較. 人工知能学会全国大会論文集, Vol. JSAI2024, pp. 4Xin298–4Xin298, 2024.

[12] Zixuan Ke, Yijia Shao, Haowei Lin, and Tothers. Continual pretraining of language models. In **The Eleventh International Conference on Learning Representations**, 2023.

[13] Kazuki Fujii, Taishi Nakamura, Mengsay Loem, et al. Continual pre-training for cross-lingual LLM adaptation: Enhancing japanese language capabilities. In **First Conference on Language Modeling**, 2024.

[14] Minato Kondo, Takehito Utsuro, and Masaaki Nagata. Enhancing translation accuracy of large language models through continual pre-training on parallel data. In **Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT 2024)**, pp. 203–220, Bangkok, Thailand (in-person and online), August 2024. Association for Computational Linguistics.

[15] Yuren Mao, Yuhang Ge, Yijiang Fan, et al. A survey on LoRA of large language models. **Frontiers of Computer Science**, Vol. 19, No. 7, pp. 1–19, July 2025.

[16] Shenglai Zeng, Yaxin Li, Jie Ren, et al. Exploring memorization in fine-tuned language models. In **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 3917–3948, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[17] Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, et al. LoRA learns less and forgets less. **Transactions on Machine Learning Research**, 2024. Featured Certification.

[18] Fatemehsadat Mireshghallah, Archit Uniyal, Tianhao Wang, et al. An empirical analysis of memorization in fine-tuned autoregressive language models. In **Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing**, pp. 1816–1826, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[19] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The llama 3 herd of models. **arXiv [cs.AI]**, July 2024.

[20] Edward J Hu, yelong shen, Phillip Wallis, et al. LoRA: Low-rank adaptation of large language models. In **International Conference on Learning Representations**, 2022.

[21] Kan Hatakeyama-Sato, Yasuhiko Igarashi, Shun Katakami, et al. Teaching specific scientific knowledge into large language models through additional training. **arXiv [cs.CL]**, December 2023.

[22] Mathieu Ravaut, Bosheng Ding, Fangkai Jiao, et al. How much are large language models contaminated? a comprehensive survey and the LLMSanitize library. **arXiv [cs.CL]**, March 2024.

[23] Katherine Lee, Daphne Ippolito, Andrew Nystrom, et al. Deduplicating training data makes language models better. In **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 8424–8445, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[24] Daphne Ippolito, Florian Tramer, Milad Nasr, et al. Preventing generation of verbatim memorization in language models gives a false sense of privacy. In **Proceedings of the 16th International Natural Language Generation Conference**, pp. 28–53, Prague, Czechia, September 2023. Association for Computational Linguistics.

[25] Li Yujian and Liu Bo. A normalized levenshtein distance metric. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Vol. 29, No. 6, pp. 1091–1095, 2007.

[26] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, et al. Privacy risk in machine learning: Analyzing the connection to overfitting. In **2018 IEEE 31st Computer Security Foundations Symposium (CSF)**, pp. 268–282, 2018.

[27] Jingyang Zhang, Jingwei Sun, Eric Yeats, et al. Min-k%++: Improved baseline for detecting pre-training data from large language models. **arXiv [cs.CL]**, April 2024.

[28] Roy Xie, Junlin Wang, Ruomin Huang, et al. ReCaLL: Membership inference via relative conditional log-likelihoods. In **Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing**, pp. 8671–8689, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[29] 佐藤敏紀, 橋本泰一, 奥村学. 単語分かち書き辞書 mecab-ipadic-NEologd の実装と情報検索における効果的な使用方法の検討. 言語処理学会第 23 回年次大会発表論文集, pp. 875–878, 2017.

A　メンバーシップ推論手法

T 個のトークンの系列からなる文 $s_n = c_1 c_2 \ldots c_T$ の生成確率 $p(s_n)$ は次のように計算できる．

$$p(s_n) = \prod_{t=1}^{T} p(c_t | c_1, \ldots, c_{t-1})$$

一般に $p(s_n)$ を直接計算すると非常に小さい値となるため，対数をとった値（対数尤度）を扱う場合が多い．次のように T で割って平均の対数尤度とすると，他との大小比較にも利用しやすくなる．

$$\frac{1}{T} \log p(s_n) = \frac{1}{T} \log \sum_{t=1}^{T} p(c_t | c_1, \ldots, c_{t-1})$$

平均予測確率の逆数をとった Perplexity（PPL）も，言語モデルの評価で一般的な指標の一つである．

$$\mathrm{PPL}(s_n) = \frac{1}{p(s_n)^{\frac{1}{T}}} = p(s_n)^{-\frac{1}{T}} = \exp(\log(p(s_n)^{-\frac{1}{T}}))$$

$$= \exp(-\frac{1}{T} \log \sum_{t=1}^{T} p(c_t | c_1, \ldots, c_{t-1}))$$

訓練データに含まれるテキストに対しては予測確率が大きくなると期待され，負の対数尤度で定義される損失は小さくなると考えられる．最も素朴な手法である **LOSS** は，損失が閾値より小さい場合に訓練データに含まれると判定する [26]．**Min-K% Prob** は，生成確率の低い K % のトークンのみに着目し平均の対数尤度を計算すると，メンバーシップ推論の性能が上がることを実証的に示した [7]．**Min-K%++**は生成確率の正規化と標準化による改良版である [27]．

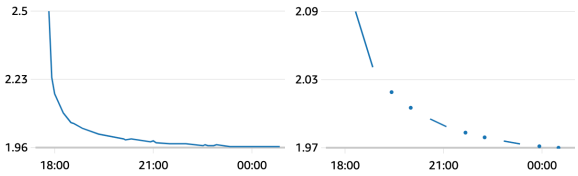
訓練データに含まれないテキストに対しては，予測確率が比較的小さくなり，繰り返しなど冗長な表現が現れやすい．この特徴を踏まえた **PPL/zlib** は，zlib 圧縮を通じて計算される情報量で PPL を割った値である [2]．**ReCaLL** は，訓練データに含まれないテキストをプロンプトに追加で与えた場合の対数尤度の変化の比率を計算する [28]．

B　LoRA による継続事前学習

Llama 3 の 8B モデルを LoRA で継続事前学習した際の学習設定を以下に示す．トークナイザは学習済みモデルを利用した．Python は 3.10，Transformers は 4.36.0，PyTorch は 2.1.0 のバージョンを用いた．Wikipedia と日経電子版で同じ設定とした．q_proj と v_proj は自己注意機構の query と value に関する層，fc_in と fc_out は全結合層である．

- 学習率：1e-4
- 最大トークン長：512
- ミクロバッチサイズ：8
- エポック数：4
- LoRA ランク：16
- LoRA 対象層：q_proj, v_proj, fc_in, fc_out

訓練・検証データに対する損失は滑らかに減少し，実行時間は約 6 時間だった（図 4）．計算資源には，GPU の A100 を 8 個含む Amazon EC2 P4 Instances の `ml.p4d.24xlarge14` を 8 個使った．検証データは，評価データとは重複しないテキスト 5000 文を訓練データから切り出した．



(a) 訓練データに対する損失．　**(b)** 検証データに対する損失．

図 4: Wikipedia を用いた継続事前学習時の損失の推移．

C　暗記量が多かった生成例

3.3 節での実験において，Wikipedia と日経電子版を用いた継続事前学習モデルで，それぞれ最も逐語暗記が大きかった生成結果を表 3・4 に示す．前者では，参照用のテキストに当たる部分が完全一致で再現された．後者では「国連気候変動枠組み条約締約国会議（COP26）」のイベント名が，日経電子版での表記規定の通りに生成された．

表 3: Wikipedia での逐語暗記が最大の生成結果．

種類	文字列
入力	（前略…）英語でハードウェア (hardware) は、本来は「金物類、金属製品」の意味であり、かつては木材製品などとの対比語として用いられた。例えば英語で ”hardware store” は、日本で言う「金物屋」を意味する。パーソナルコンピュータのハードウェア
参照	などを「ソフトウェア」と呼ぶことがある。
生成	などを「ソフトウェア」と呼ぶことがある。

表 4: 日経電子版での逐語暗記が最大の生成結果．

種類	文字列
入力	（前略…）日本政府は 4 月、30 年度に温暖化ガス排出を 13 年度比 46 %減らす目標を打ち出した。秋に開かれた第 26
参照	回国連気候変動枠組み条約締約国会議（COP26）では、「世界の平均気温の上昇を 1.5 度に抑える努力を追求することを決意する」ことで合意した。
生成	回国連気候変動枠組み条約締約国会議 (COP26) で も、世界各国は脱炭素の実行を急ぐ姿勢を鮮明にした。