Padding vs. Packing: 大規模言語モデルのファインチューニングにおける学習効果の検証

塩野 大輝 [1]　田中 涼太 [1,2]　宮脇 峻平 [1]　工藤 慧音 [1]　鈴木 潤 [1]
[1] 東北大学　[2] 日本電信電話株式会社 NTT 人間情報研究所
daiki.shiono.s1@dc.tohoku.ac.jp

概要

大規模言語モデル (LLM) のファインチューニングにおける入力系列の作成時には，1 つのサンプルに [PAD] トークンを最大系列長まで連結する Padding 戦略が広く採用される．しかしファインチューニング時には，最大系列長に収まるように複数サンプルを連結し入力系列を作成する Packing 戦略を採用する選択肢も考えられる．この Packing 戦略は，複数のサンプルを連結して入力するので，Padding 戦略と比較して学習効率が高い利点がある．しかし，Packing 戦略が，Padding 戦略を採用した時と同様の学習効果が得られるかどうかは明らかになっていない．本研究では，LLM のファインチューニング段階における，Packing 戦略の学習効果を検証し，特定の学習データに対しては，Padding 戦略で学習した場合と同等の学習効果が得られることを示す．

1　はじめに

大規模言語モデル (LLM) は，事前学習によって，大規模なテキストコーパスから言語に関する知識や常識的な知識を獲得する [1, 2]．さらに事前学習済み LLM に対して，入力文（指示文）と回答文のペアデータでファインチューニングすることにより，自然言語で記述された指示文に従って回答を生成する能力（指示追従能力）を獲得する [3]．これにより，LLM は指示追従能力が向上し，多様な自然言語処理タスクに対して高い汎化性能を示す．

一般的に『ファインチューニング』時の入力系列の作成には，**Padding 戦略**が広く採用される．Padding 戦略は実装が容易である一方で，1 サンプル分のトークン以外を最大系列長まで全て [PAD] トークンで埋めるため，計算上無駄な [PAD] トークンが多く含まれてしまい，メモリ効率や計算効率が最適ではない [4]．一方で，『事前学習』時には複数のサンプルを連結し，最大系列長を超えない範囲で入力系列を作成する **Packing 戦略** [1)] が広く用いられてきた [1, 5]．Packing 戦略では，複数のサンプルを 1 つの入力系列にまとめるので，Padding 戦略と比較して，使用する [PAD] トークン数が減少し，学習効率が向上することが期待される．この Packing 戦略は，事前学習段階において，性能や計算効率における優れた効果が実証されている [6, 1, 5] 一方で，ファインチューニング段階においてどのような学習効果が得られるのか完全には明らかになっていない．

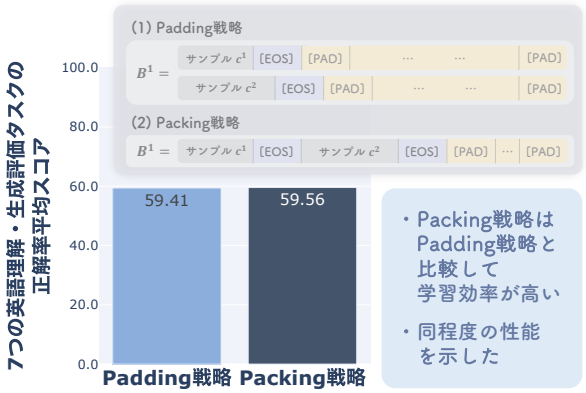

図 1　Padding 戦略及び Packing 戦略で，前処理済みの WildChat を用いてファインチューニングした LLM の最終チェックポイントにおける，7 つの英語理解・生成評価データセットの正解率平均スコア．7 スコア正解率平均の誤差は約 0.15 ポイントであり，Padding 戦略と Packing 戦略の両者間において，顕著な差はほとんど見られない．

本研究では，ファインチューニングにおける Packing 戦略の導入が LLM の性能に及ぼす影響を，下流タスクの定量評価および定性評価から明らかにすることを目的とする．具体的には，Padding 戦略と Packing 戦略の学習効果の同一性を厳密に比較調査するため，1 ステップあたりの損失計算対象となるトークン数を揃えた上で，Padding 戦略と Packing 戦略を用いてファインチューニングした 100 学習ステップごとのチェックポイントを 7 つの英語理解・生成評価タスクで評価する．その結果，Padding 戦

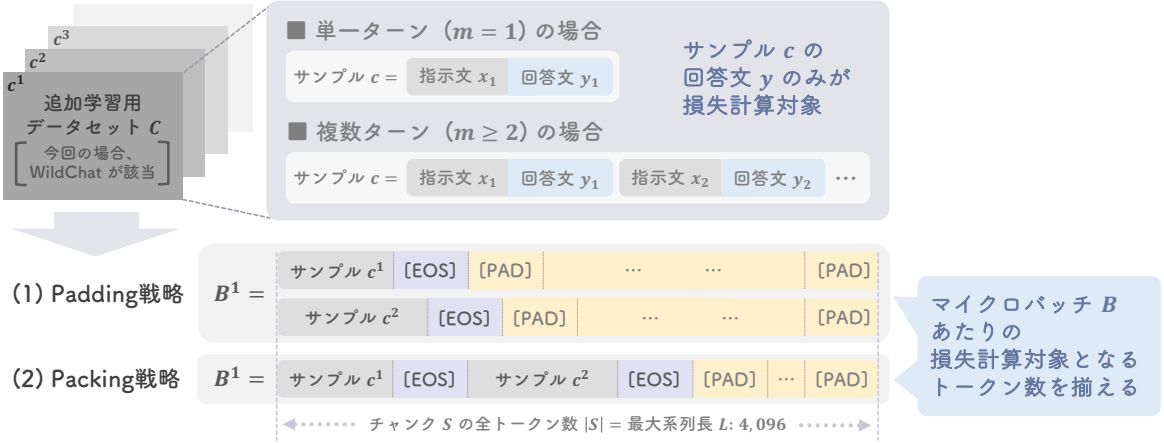

図 2　本研究の実験設定における，1 学習ステップあたりの損失計算対象となるトークン数を固定した場合の，Padding 戦略と Packing 戦略のマイクロバッチ B の作成例．(1) Padding 戦略: 1 チャンクに 1 サンプルを含めてマイクロバッチサイズを 2 に設定し，最大系列長 L に満たない部分を特殊トークン [PAD] で埋める．(2) Packing 戦略: 1 チャンクに 2 サンプルを含めてマイクロバッチサイズを 1 に設定し，最大系列長 L に満たない部分を特殊トークン [PAD] で埋める．

略と Packing 戦略が特定の学習データの下では，ほとんど同等の学習効果を持つことを示す（図 1）．

2　関連研究

事前学習において，Packing 時に，関連するサンプル同士を連結して因果的マスキングを適用することで，LLM の文脈内学習能力を向上させることができるという報告が複数挙げられている [7, 8]．また，連結した複数のサンプルが互いに干渉しないように各サンプル内の先行トークンのみに条件付けして次のトークンを予測できるようにする文書内因果的マスキング手法も使われ始めている [9, 2]．文書内因果的マスキング手法は，不要なサンプルからの干渉を完全に排除できる利点があるが，学習効率に悪影響を及ぼすことが主張されている [1, 10]．本研究では，因果的マスキング適用時のファインチューニング段階における Packing 戦略の効果検証のみに焦点を当て，文書内因果的マスキングやその他のマスキング手法については効果検証調査の範囲外とする．

我々の知る限り，Wang ら [4] が初めて，ファインチューニング段階における Padding 戦略と Packing 戦略の性能を比較調査している．彼らは，Packing 戦略で学習した LLM は，Padding 戦略で学習した LLM と比較して，さまざまなベンチマークにおいて平均的に優れた性能を示すと報告している．しかし，彼らの実験設定では，Padding 戦略と Packing 戦略で，1 学習ステップあたりの損失計算対象となるトークン数が異なるため，正確な比較ができていない可能性がある．本研究では，1 学習ステップあたりの損失計算対象となるトークン数を厳密に揃えた上で，Padding 戦略と Packing 戦略の学習効果の同一性を比較調査することを目指す．

3　入力系列の構築戦略

本研究における，ファインチューニング時の入力系列を作成するための手法である Padding 戦略と Packing 戦略 [1] の違いを説明する．図 2 に示すように，WildChat [11] などのファインチューニング用の学習データサンプル集合を $C = \{c^1, \ldots, c^N\}$ と定義する．ここで，N はデータサンプル数を表す．また，$c^i = \left((x_1^i, y_1^i), \ldots, (x_m^i, y_m^i)\right)$　$(m \geq 1)$ であり，m はサンプル c^i に含まれる指示文 x と回答文 y のペアの個数（複数ターン数）を指す．ファインチューニング段階においては，Padding 戦略と Packing 戦略のいずれも，回答文 y に含まれるトークンのみが損失計算の対象となる．

Padding 戦略　Padding 戦略は，学習データサンプル集合 C 中の，各データサンプル c^i に対して，最大系列長 L に満たない部分を特殊トークン [PAD] で埋めることで，全入力系列の長さを L に揃えたチャンク S を作成する手法である．Padding されたチャンク S は以下のように表される:

$$S^i = (c^i\,[\mathrm{EOS}]\,[\mathrm{PAD}]\ldots[\mathrm{PAD}]) \quad (|S^i| = L) \qquad (1)$$

Packing 戦略　Packing 戦略は，学習データサンプル集合 C の中から，最大系列長 L を超えない範囲

1)　本研究で使用する Packing 戦略は，事前学習時に頻繁に使用される複数サンプルを連結していき最大系列長を超えたサンプルは切り詰める戦略とは異なり，最大系列長を超えない範囲でサンプルを連結し，残りの部分は [PAD] トークンで埋めるような実装になっていることに注意したい．

で，複数のデータサンプル c を抽出・連結し，最後に最大系列長 L に満たない部分を特殊トークン [PAD] で埋めることで，全入力系列の長さを L に揃えたチャンク S を作成する手法である．Packing されたチャンク S は以下のように表される:

$$S^i = (c^k [\mathrm{EOS}] \dots c^{k+l} [\mathrm{EOS}] [\mathrm{PAD}] \dots [\mathrm{PAD}]) \quad (|S^i| = L) \tag{2}$$

ただし，$l+1$ は L を超えない範囲で Packing されたチャンク S に含まれるデータサンプル数を表す．また，一度チャンク S に含まれたデータサンプル c は，その後の学習データサンプル集合 C から削除され，複数回抽出されることはない．

Packing 戦略と Padding 戦略の利点と欠点 Padding 戦略は，1 サンプル分のトークン以外を最大系列長まで全て [PAD] トークンで埋める簡単な実装となる利点がある一方で，系列長が短いサンプル c が多く含まれる学習データサンプル集合 C ほど学習時の計算効率が悪くなる．一方，Packing 戦略は，チャンク S に各サンプル c を密に詰め込むことができるので，学習上無駄なトークン [PAD] を最小限に抑えられ，Padding 戦略に比べて計算効率が高い利点がある．しかし，2 章で説明した文書内因果的マスキングのような特殊なマスキング手法を用いない限り，連結した複数のサンプル中の不要なサンプルからの干渉を受ける可能性がある．本研究では，Padding 戦略と比較した Packing 戦略の学習効果の同一性を厳密に比較調査することに焦点を当てるため，**因果的マスキングに特別な工夫は施さず，Packing 戦略と Padding 戦略で 1 学習ステップあたりの損失計算対象となるトークン数を揃える．**

4 実験設定

本研究では，Padding 戦略と Packing 戦略の学習効果の同一性を比較調査するため，**1 学習ステップあたりの損失計算対象となるトークン数を固定**した（図 2）．具体的には，Padding 戦略では 1 チャンクに 1 サンプルを含めてマイクロバッチサイズを 2 に設定し，Packing 戦略では 1 チャンクに 2 サンプルを含めてマイクロバッチサイズを 1 に設定した．LLM には，Llama 3 8B [2] を採用し，データセットとして，トークン数が 2,000 を超えるサンプルを除いた WildChat（約 3.6 万件）を使用した（A.1）．ファインチューニング時には，100 ステップごとにモデルのチェックポイントを保存した．なお最終学習ステップ数は，2,287 となった．

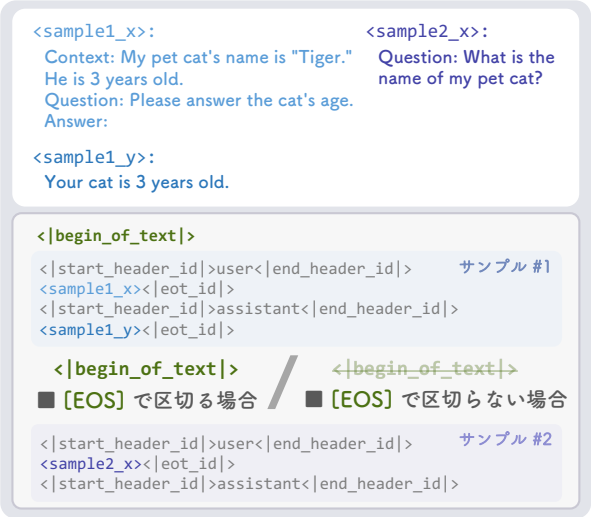

図 3 定性分析に使用した 2 つのサンプル同士を [EOS] で区切るか否かの違いを持つ 2 つのテストプロンプトの実例．<|begin_of_text|>が [EOS] トークンに相当する．

4.1 定量評価: 学習効果の同一性検証

Padding 戦略と Packing 戦略の学習効果の同一性を比較調査するため，2 つの戦略で学習した全てのチェックポイントで，MMLU [12]，TriviaQA [13]，GSM8K [14]，OpenBookQA [15]，HellaSwag [16]，XWINO [17]，SQuAD2 [18] の 7 つの英語理解・生成評価データセット（表 2）を用いて性能を測定した．MMLU は，5-shot，それ以外の評価データセットでは，4-shot で評価を実施した．評価指標には，正解率を用いた．

4.2 定性分析: [EOS] トークンの機能分析

Packing 戦略で学習する場合，連結した同一チャンク内のサンプル間を [EOS] トークンで区切るので，サンプル c^{k+1} の回答生成時に，[EOS] トークンが出現するより前のサンプル c^k 内のトークンを参照しないように LLM の学習が進むことが期待される．そこで 5.2 節では，2 つのサンプル間を [EOS] で区切ったプロンプトと区切らないプロンプトをそれぞれ用意し（図 3），Packing 戦略で学習された LLM が期待される振る舞いをするか確認した．

5 実験結果と分析

5.1 定量評価: 学習効果の同一性検証

Packing 戦略，Padding 戦略ともに，それぞれ 100 学習ステップごとに 7 つの評価データセットを用い

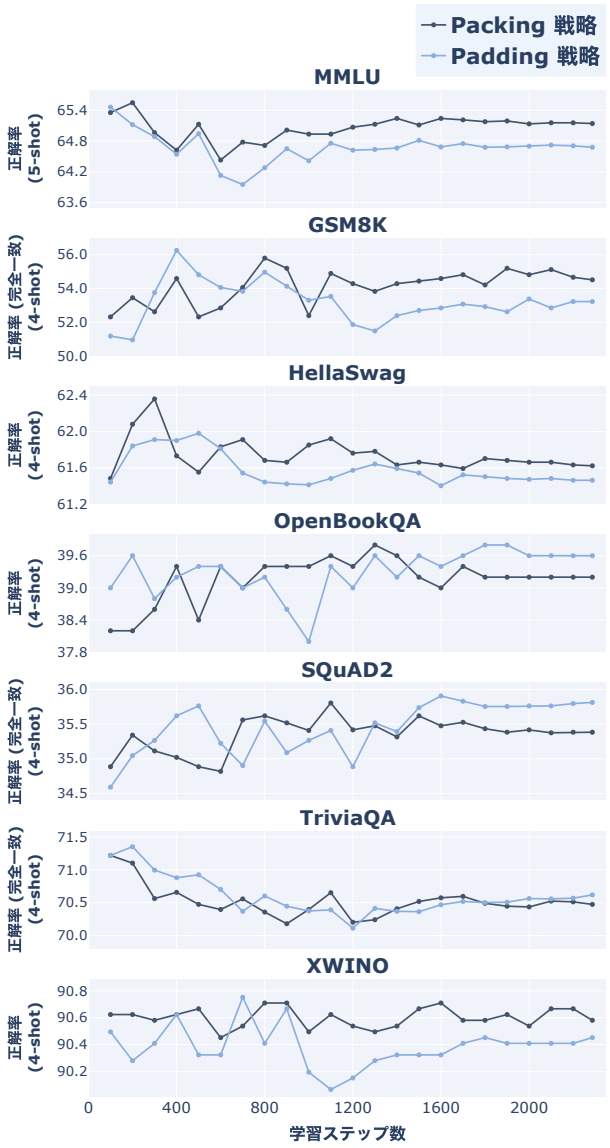

図 4　Packing 戦略，Padding 戦略ともに，それぞれ 100 学習ステップごとに 7 つの英語理解・生成評価データセットを用いて，LLM を評価した結果．

て，LLM の評価実験を行った結果を図 4 に示す．また，最終チェックポイント（学習ステップ数: 2,287）での評価結果を図 1 に示す．多様な英語理解・生成評価タスクにおいて，同じ学習ステップ数における Padding 戦略と Packing 戦略のスコア間の 7 タスク平均誤差は約 0.49 ポイントであった．このことから，特定の学習データ（WildChat）において，Packing 戦略と Padding 戦略の両者は，同様の学習効果を有している可能性があるといえる．そのため，ファインチューニングデータセットに含まれるサンプル数が大きいかつ各サンプルあたりのトークン数が少ない場合には，Packing 戦略を採用することで，Padding 戦略を採った場合と同程度の性能をより短い学習時間で達成できる可能性がある．

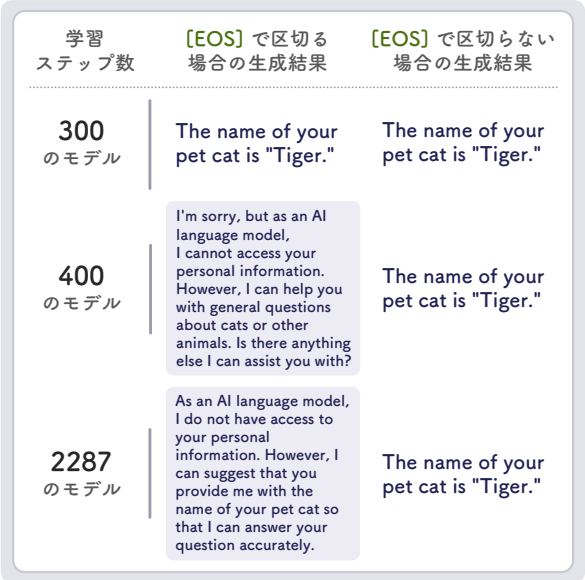

図 5　図 3 に示した 2 つのテストプロンプトに対する，Packing 戦略で学習したステップ数: 300, 400, 2,287(最終チェックポイント) の LLM の出力結果．学習ステップ数 400 以降の LLM は，連結したサンプル同士を [EOS] トークンで区切ることで，1 つ目のサンプル内に与えらている猫の名前を回答することができなくなる挙動を示した．

5.2　定性分析: [EOS] トークンの機能分析

定性分析の結果を図 5 に示す．Packing 戦略で学習した最終チェックポイント（学習ステップ数: 2,287）を用いて，定性分析を実施した結果，連結したサンプル間を [EOS] トークンで区切ることで，LLM が 1 つ目のサンプル内に与えられた猫の名前を回答しなくなった．この結果は，Packing 戦略で LLM を学習することによって，[EOS] トークンが自身以降のトークン出力時に自身以前のトークンを参照しないようにする目印としての機能を獲得している可能性を示唆する．加えて，学習ステップ数が 300 から 400 の箇所で，[EOS] ありの生成結果が変化したことから，Packing 戦略では学習の比較的早い段階で，この機能が発現している可能性が高い．

6　おわりに

本研究では，LLM のファインチューニングにおける Packing 戦略の学習効果を調査した．定量評価より，特定の学習データに対しては，Packing 戦略と Padding 戦略は，同様の学習効果を有している可能性があることが示唆された．これより Packing 戦略は，Padding 戦略を採った場合と同程度の性能をより短い学習時間で達成できる可能性がある．

謝辞

本研究は，JST ムーンショット型研究開発事業 JPMJMS2011-35 (fundamental research)，文部科学省の補助事業「生成 AI モデルの透明性・信頼性の確保に向けた研究開発拠点形成」，JST 国家戦略分野の若手研究者及び博士後期課程学生の育成事業（博士後期課程学生支援）JPMJBS2421 の支援を受けたものです．

本研究の一部は九州大学情報基盤研究開発センター研究用計算機システムの一般利用を利用しています．

研究遂行にあたりご協力を賜りました Tohoku NLP グループの皆様に感謝申し上げます．また評価の細かい技術的知見に関して，Swallow チーム（東京科学大）の解説記事を大いに参考にさせていただきました．この場を借りて，感謝申し上げます．

参考文献

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In **Advances in Neural Information Processing Systems (NeurIPS)**, pp. 1877–1901, 2020.

[2] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 Herd of Models. **arXiv preprint**, cs.CL/2407.21783v3, 2024.

[3] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned Language Models are Zero-Shot Learners. In **International Conference on Learning Representations (ICLR)**, 2022.

[4] Shuhe Wang, Guoyin Wang, Yizhong Wang, Jiwei Li, Eduard Hovy, and Chen Guo. Packing analysis: Packing is more appropriate for large models or datasets in supervised fine-tuning. **arXiv preprint**, cs.CL/2410.08081v3, 2024.

[5] Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. **arXiv preprint**, cs.CL/2401.02385v2, 2024.

[6] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism. **arXiv preprint**, cs.CL/1909.08053v4, 2019.

[7] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Pre-Training to Learn in Context. In **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 4849–4870, 2023.

[8] Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Xi Victoria Lin, Noah A. Smith, Luke Zettlemoyer, Wen tau Yih, and Mike Lewis. In-Context Pretraining: Language Modeling Beyond Document Boundaries. In **The Twelfth International Conference on Learning Representations (ICLR)**, 2024.

[9] Yu Zhao, Yuanbin Qu, Konrad Staniszewski, Szymon Tworkowski, Wei Liu, Piotr Mi l oś, Yuxiang Wu, and Pasquale Minervini. Analysing The Impact of Sequence Composition on Language Model Pre-Training. In **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 7897–7912, 2024.

[10] Matteo Pagliardini, Daniele Paliotta, Martin Jaggi, and François Fleuret. Faster causal attention over large sequences through sparse flash attention. **arXiv preprint**, cs.CL/2306.01160v1, 2023.

[11] Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. WildChat: 1M ChatGPT Interaction Logs in the Wild. In **The Twelfth International Conference on Learning Representations (ICLR)**, 2024.

[12] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding. In **International Conference on Learning Representations (ICLR)**, 2021.

[13] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In **Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 1601–1611, 2017.

[14] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. **arXiv preprint**, cs.CL/2110.14168v2, 2021.

[15] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 2381–2391, 2018.

[16] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a Machine Really Finish Your Sentence? In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 4791–4800, 2019.

[17] Alexey Tikhonov and Max Ryabinin. It's All in the Heads: Using Attention Heads as a Baseline for Cross-Lingual Transfer in Commonsense Reasoning. In **Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021**, pp. 3534–3546, 2021.

[18] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know What You Don't Know: Unanswerable Questions for SQuAD. In **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 784–789, 2018.

[19] Kazuki Fujii, Taishi Nakamura, and Rio Yokota. llm-recipes, 2024.

[20] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 2024.

A　参考情報

A.1　前処理済み WildChat の基本統計量

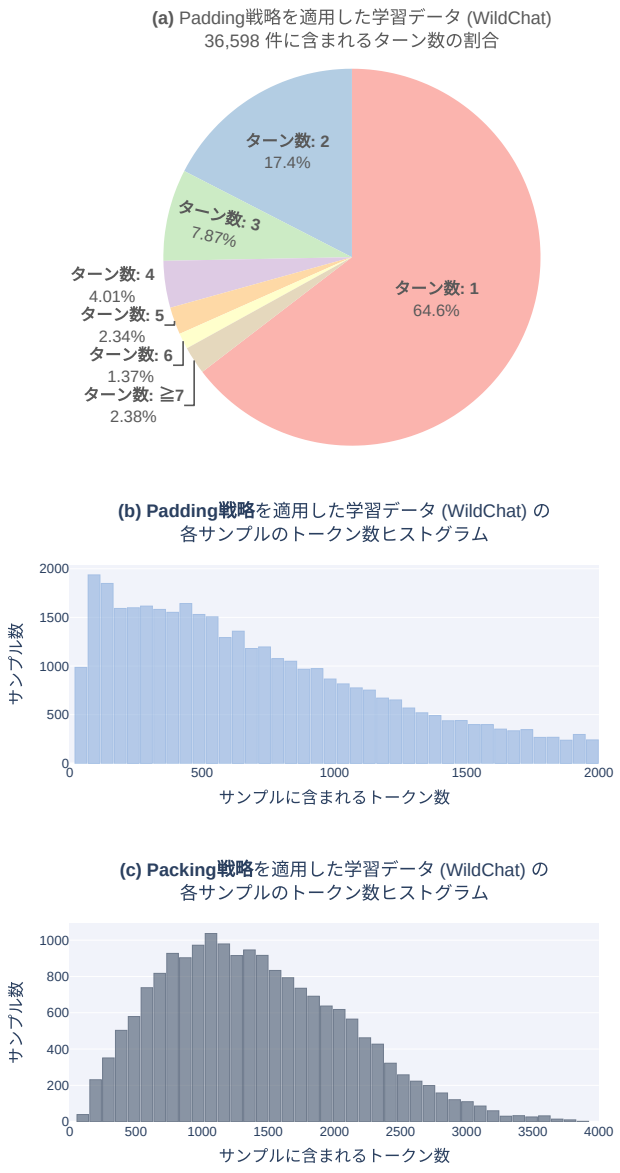

図 6　前処理後の WildChat データセットの種々の統計量. (a): 複数ターン数の割合，(b): Padding 戦略適用時のサンプルのトークン数ヒストグラム，(c): Packing 戦略適用時のサンプルのトークン数ヒストグラムを表す.

WildChat データセットから，言語が英語以外のサンプルのものと指示文が空文字列になっているものを除いた．さらに，4 章の実験に使用したいため，2,000 トークンより大きいサンプルは除外した．これにより，WildChat のサンプルが約 3.6 万件残った．このデータの種々の統計量を図 6 に示す.

A.2　学習設定の詳細

LLM は，事前学習済みの Llama 3 8B[2] を採用し，トークナイザーには，Llama 3.1 Instruct[3] のものを採用した．表 1 に詳細な学習設定を示す.

表 1　Padding 戦略及び Packing 戦略でファインチューニングした LLM の学習設定の詳細.

	Padding 戦略	Packing 戦略
GPU (H100) Num	8	4
Global Batch Size	16	8
Micro Batch Size	2	2
#Samples	36,598	18,299
#Samples / chunk	1	2
#Samples / step	16	16
Epoch	1	1
Max LR	1.0×10^{-5}	1.0×10^{-5}
Min LR	1.0×10^{-6}	1.0×10^{-6}
LR Warmup Steps	228 (10%)	228 (10%)
Scheduler	cosine	cosine
Optimizer	AdamW	AdamW
Optimizer Config	$\beta_1 = 0.9, \beta_2 = 0.95, \epsilon = 1.0 \times 10^{-8}$	
Weight Decay	0.1	0.1
Gradient Clipping	1.0	1.0
Sequence Length	4,096	4,096

A.3　評価データセットの詳細

表 2　評価タスクに含まれるデータ件数とタスクの説明.

タスク名	件数	タスクの説明
MMLU	14,042	57 科目からなる 4 値選択式の試験問題
GSM8K	1,319	小学校の数学の文章題データセット
HellaSwag	10,042	次に起こるイベントを予測する 4 択の選択式問題
OpenBookQA	500	科学的な知識と常識に基づく 4 択の選択式問題
SQuAD2	11,873	根拠文書に対して作成された自由記述式質問応答
TriviaQA	17,944	雑学的な知識に基づく自由記述式質問応答
XWINO	2,325	文中の代名詞の先行詞を推定する 2 択の選択式問題

A.4　使用したソフトウェアの詳細

LLM のファインチューニングには，llm-recipes [19] (v1.0.1)[4] を使用した．また，定量評価には，Language Model Evaluation Harness [20] (v0.4.5)[5] を用いた.

2) https://huggingface.co/meta-llama/Meta-Llama-3-8B
3) https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
4) www.github.com/okoge-kaz/llm-recipes/tree/v1.0.1
5) www.github.com/EleutherAI/lm-evaluation-harness/tree/v0.4.5