主語埋め込みの編集による効率的な未知知識追加手法の検討

北島祥平 [1]　井之上直也 [1,2]
[1] 北陸先端科学技術大学院大学　[2] 理化学研究所
{s2510043,naoya-i}@jaist.ac.jp

概要

大規模言語モデルの知識は学習完了時点で固定され，新たな知識を追加するにはモデルの再学習が要求される．この問題に対して Fine-Tuning や検索拡張生成が提案されてきたが，計算コストの高さなど課題は多い．そこで本研究では知識編集技術による効率的な知識追加手法を探求した．特に有望な手法として，主語埋め込みの変更のみで知識編集を実現した軽量な手法である SWEA⊕OS に注目し，モデルにとって未知の知識を追加するタスクにおける性能を既存の知識編集手法とを比較した．実験の結果，SWEA⊕OS は性能面で既存手法に劣るが，性能と各種コストのバランスに優れ，効率的な知識追加に対する有望な手法の 1 つであることが示唆された．

1　はじめに

大規模言語モデル (Large Language Models, LLM) は多くの分野で人間の専門家に匹敵する能力を発揮している [1, 2, 3, 4] が，ドメイン固有の知識を追加する際にモデルの再学習が要求されるという問題を抱えている．これは，LLM の知識が内部でパラメタとして静的に保持されるためであり，特に学習 cuttoff 以降に生じた未知の知識を追加すること（**知識追加**）は困難である．代表的な知識追加アプローチである Fine-Tuning [5] や検索拡張生成 [6] は一定の成果を挙げているものの，高い計算リソース要求，追加のストレージコストなど依然として未解決の課題も多い [7, 8, 9]．LLM が実社会へ浸透している現状において，各組織に固有の知識をモデルに導入したいというニーズは増加しており，効率的な知識追加手法の欠如は大きな問題である [10, 11]．

そこで本研究では，特定知識のみを正確に編集することで高い計算効率を達成した知識編集 [12, 13, 14] に注目し，モデルにとって未知の知識を効率的に追加する方法を探求する．知識編集の手法間には性能と効率に明確なトレードオフが存在するが，その中でも主語埋め込みの更新に基づく SWEA⊕OS [15] は複数の評価指標において他の手法を上回り，かつ圧倒的に軽量であるという優位性を示した．しかしながら，その検証は，モデルが既に持っている知識を更新すること（**知識修正**）に焦点が当てられており，知識追加においても同様の性能を発揮するか十分な検証が行われていない．そのため本稿では，SWEA⊕OS の未知知識追加の性能を検証し，他の知識編集手法との比較分析を行う．

分析の結果，既存の知識編集手法と SWEA⊕OS を比較し，SWEA⊕OS は性能と各種コストのバランスから，有望な手法であることが示唆された．未知知識の追加という観点で複数の知識編集手法を横断的に分析する試みは先行事例が少なく [16, 17]，こうした知見は効率的な知識追加手法の確立に向けて重要な示唆を与えるものと期待される．

2　既存の知識編集手法

知識編集は，特定の知識のみを正確に編集しつつ無関係な知識への影響を抑えることを目的とした技術である．複数の手法が存在するが，基本的なアプローチによって以下の 3 種類に大別できる [12]．

2.1　外部記憶ベース

LLM のパラメタは変更せず，新たな知識を保存・編集するための外部メモリを活用する．モデルパラメタの更新が不要なため計算効率に優れるが，追加のストレージやコンポーネントが要求されるのが欠点である．代表的な手法として SERAC [18]，SWEA⊕OS [15] などがある．

2.2　大域的最適化ベース

Fine-Tuning と同様に，新たな少数の知識によって内部パラメタを更新することで汎用的な知識組み込みを実現する．既存知識への影響を抑制するような工夫が施されている点で Fine-Tuning とは差別化されるが，更新対象のパラメタが膨大であるため

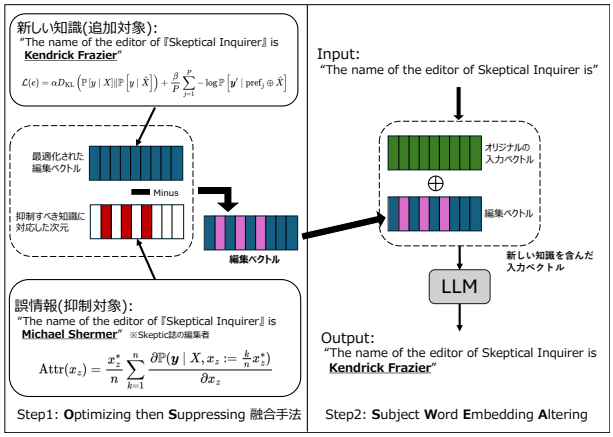

図 1　SWEA⊕OS による知識追加

計算コストの要求は重い．代表的な手法としては MEND [19] などが挙げられる．

2.3　局所編集ベース

特定の知識に対応する LLM の内部パラメタを特定・編集することで新たな知識を組み込む．モデルパラメタの一部のみを更新すればよいため計算効率が高く，追加のストレージ要求もないが，部分的なパラメタ変更によって無関係な知識やモデル自体の推論能力などに悪影響を及ぼす懸念がある．代表的な手法は ROME [13]，MEMIT [20] などがある．

3　SWEA⊕OS による知識追加

前述の通り SWEA⊕OS の性能検証は既存知識の修正に限定されており，未知知識の追加については未検証である．本節では，SWEA⊕OS を未知知識の追加タスクに適応させる方法を述べる．

3.1　前提: 知識追加タスク

知識の「追加」とは，主語 s，目的語 r，主語と目的語の関係 r からなる事実知識のトリプレット (s,r,o) [20] に対する次のような操作を指す:

$$(s,r,\phi) \rightarrow (s,r,o') \tag{1}$$

ここで，o' は $o \neq o'$ を満たす目的語であり，追加対象の未知知識である．直観的な説明としては，「Skeptical Inquirer 誌の編集者は不明です」と出力するモデルに対して何らかの操作を行うことで，「Skeptical Inquirer 誌の編集者はケンドリック・フレイザーです」と回答できるようにするのが知識追加タスクである．

式 (1) に示したように，知識追加は未知知識をモデル内部に取り込ませ，更に既存知識との関係性を構築することを要求する．この点において知識追加は，既にモデルが活用している知識を変更する修正とは性質が異なるタスクである．そのため，知識追加タスクにおいても SWEA⊕OS による主語埋め込み編集が十分な威力を発揮できるかは個別の検証が必要であり，これが本研究の主要な目的である．

3.2　知識追加の方法

SWEA⊕OS は，主語埋め込みへ僅かな編集ベクトルを加算するのみで知識を更新する手法であり，モデル内部への直接的な介入なしに知識編集手法全般の課題である局所性と汎化性 [12, 21, 22] の改善を実現した．訓練を要するのは編集ベクトルのみであり，かつこれも非常に小さいことから計算効率に優れ，追加ストレージの負荷も小さい．本研究では，図 1 に示すように，SWEA⊕OS を知識追加タスクに適応させる．

3.2.1　Step1: 知識編集ベクトルの学習

編集ベクトルが追加知識を出力できるよう（図 1, 左上），式 (2) の損失関数に基づいて最適化する:

$$\mathcal{L}(e) = \alpha D_{KL}(\mathbb{P}[y|X]||\mathbb{P}[y|\hat{X}]) + \frac{\beta}{P}\sum_{j=1}^{P} -\log \mathbb{P}[y'|pref_j \oplus \hat{X}] \tag{2}$$

ここで X は (s,r) のテキスト埋め込み，$\hat{X}$ は編集ベクトルを付加した X，y, y' はそれぞれ o, o' のすべてのトークン，D_{KL} は KL ダイバージェンス，e は最適化の目的である編集ベクトル，P はモデルによって生成された $\hat{X}$ を導く先行文，$\oplus$ は連結操作を表し，α, β はハイパーパラメタである．

次に最適化された編集ベクトルにおいて，主語 S の任意の事実知識に対応する次元を特定して抑制する（図 1, 左下）．知識帰属手法 [23] を用いると，$x^S = [x_s^S, ..., x_e^S] \in R^{|S| \times h}$ における任意の 1 つの埋め込みベクトル x_z の知識帰属スコアは式 (3) のように表せる:

$$\mathrm{Attr}(x_z) = \frac{x_z^*}{n}\sum_{k=1}^{n}\frac{\partial \mathbb{P}(\boldsymbol{y}|X, x_z := \frac{k}{n}x_z^*)}{\partial x_z} \tag{3}$$

ここで，x_z^* は埋め込みベクトルの元の値，n はリーマン積分のステップ数，$\mathbb{P}(\boldsymbol{y}|X, x_z := \frac{k}{n}x_z^*)$ は x_z を $\frac{k}{n}x_z^*$ で置き換えた後にモデルが $\boldsymbol{y}$ を生成する確率である．

この式 (3) を用いて主語 S のすべての埋め込み次元の知識帰属スコアを取得し，最大スコアの t 倍を

表 1　未知知識の追加結果

	外部記憶			大域的最適化	局所編集		その他			
評価指標	SWEA	(w/o supp.)	SERAC	MEND	ROME	MEMIT	ICE	AdaLoRA	FT-L	FT-M
Edit Succ.	86.74	86.20	98.68	95.75	97.18	97.05	60.74	**100.00**	55.75	**100.00**
Portability	51.63	51.65	63.52	55.88	55.25	56.37	36.93	64.69	40.86	**65.44**
Locality	56.57	56.94	**100.00**	94.76	54.77	52.15	33.34	56.42	43.70	64.33
Fluency	587.19	**587.47**	553.19	557.11	579.66	573.89	531.01	579.57	529.24	574.32

超える埋め込み次元を K_D として保持する．その後，最適化された編集ベクトル e から K_D に対応するオリジナルの埋め込みベクトル x^S の γ 倍を減算することで最終的な編集ベクトル e^S を得る:

$$e^S = e - \gamma \mathbb{O}_{\setminus K_D} \odot x^S \tag{4}$$

ここで $\mathbb{O}_{\setminus K_D}$ は K_D に対応する位置のみが 1 でそれ以外はすべて 0 であるようなベクトルであり，$\odot$ は要素ごとの乗算を表す．

既存知識の修正において，修正前の古い知識の影響を抑制するために設けられた抑制ステップは SWEA⊕OS の性能向上に大きく貢献していた．しかしながら知識追加タスクにおいては，理想的には式 (1) に示すように抑制すべき古い知識は存在しない．そのため知識追加において抑制ステップは省略可能だが，実際のモデルの挙動としては未知の知識に関する入力に対して，「不明」である旨を回答するか，誤情報を生成する可能性が考えられる (e.g. 「Skeptical Inquirer 誌の編集者は」という入力に対し，類似の雑誌である Skeptic 誌の編集者を答える)．そこで本研究では，SWEA⊕OS 適用前のモデルが誤情報を生成した場合のみ抑制を行うよう調整を施した．

3.2.2　Step 2: 主語埋め込みの更新

得られた編集ベクトル e^S と主語 S の埋め込みベクトル x^S を統合することで，最終的な入力ベクトル X を得る:

$$X = [x_0, ..., x_s^S + e_s^S, ..., x_e^S + e_e^S, ..., x_l] \tag{5}$$

ここで，$|S|$ は主語のトークン長，x_s^S と x_e^S はそれぞれ X 内の主語 S の最初と最後のトークンである．

4　実験

4.1　実験設定

データセット　様々な知識編集タスク向けに調整された KnowEdit [16, 14, 21] に収録されている知識追加のベンチマークである，Wiki_{recent} を用いた．

モデル　既に文献 [16] において行われた Wiki_{recent} を用いた実験結果と比較するために，同一のモデル (Llama2-7b-Chat [24]),

評価指標　知識編集の研究で標準的な，以下の 4 つの指標を用いる．

- Edit Succ.: 追加成功率．新たに追加された知識をモデルが生成できるか．
- Portability: 汎化性．追加による周辺知識への影響に対処できるか．
- Locality: 局所性．無関係な知識に影響を与えていないか．
- Fluency: 流暢性．編集後モデルの文章生成能力が低下していないか．

比較対象　誤情報抑制を行う SWEA に対し，抑制を行わない亜種 (w/o supp.) を用意した．加えて，文献 [16] より Wiki_{recent} を 8 つの知識編集手法で検証した結果を引用した．

4.2　結果と考察

結果を表 1 に示す．実験の結果，SWEA⊕OS は Fluency を除く指標において，他の知識編集手法と同等かそれ以下の性能しか達成できず，修正タスクにおける性能とのギャップが明らかになった．また SWEA と (w/o supp) の値を比較すると，すべての指標においてスコアに有意な差は見られず，誤情報抑制の有無が性能に影響を及ぼしていないことが明らかになった．これも既存知識の修正における先行研究 [15] の結果と明確に異なる点である．

外部記憶　同じ外部記憶ベースの手法である SERAC に対して SWEA は大きく差を付けられた．特筆すべきは SERAC の Locality スコア (100.00) であり，入力プロンプトが編集対象であるか否かをスコープすることで使用するモデルを変更する SERAC の構造的利点 [18] が顕著に表れている．また，Edit Succ. および Portability についても SERAC は知識編集手法内では最高スコアであり，これは

Fine-Tuning の改良である AdaLoRA や FT-M に迫る値である．この結果は知識追加における外部記憶ベース手法の優位性を示唆している．しかしながら，SERAC の性能は追加のコンポーネントを導入することによって達成されたものである．これに対して SWEA⊕OS が主語埋め込みの僅かな変更のみで知識追加を実現していることを考慮すれば，SWEA⊕OS はその簡便さによって実用的な性能を達成したと言える．また，Fluency についても SWEA⊕OS は SERAC を上回っており，これはモデル本来の文章生成能力がより維持されていることを示している．加えて SERAC には追加コンポーネント分のストレージコストと推論時オーバーヘッドの増加という課題が存在しているため [21]，実運用上は SWEA⊕OS に利点があると考えられる．

大域的最適化　MEND は勾配分解を利用した補助ネットワークから予測・生成された重み更新量に基づいてモデル内部のパラメタを編集する手法であり [19]，計算コストの高さ相応の性能を発揮している．特に Locality の高さは大域的最適化ベースの手法としては非常に高く，ベースモデルの出力を維持するという MEND の特性が明確に表れている．一方で Portability における SWEA⊕OS のスコアは MEND に及ばないながらも準ずる値を達成しており，主語埋め込み変更というアプローチの有用性を示唆している．

局所編集　代表的な知識編集手法である ROME と MEMIT だが，本稿における比較対象群の中では SWEA⊕OS に近い性能を示した．Edit Succ. では ROME，MEMIT が明確に上回っており，あくまでも入力埋め込みに追加知識を加算するだけの SWEA⊕OS に対して，モデル内部で知識を保持しているとされる MLP 層へダイレクトに追加を行うアプローチの強力さが表れている [13, 20]．Edit Succ. の差の大きさを考えれば ROME や MEMIT の優位は明らかだが，これらの手法による内部パラメタへの局所的な追加は無関係な知識へ影響を及ぼす可能性が指摘されており [22]，実際に Locality については若干ではあるが SWEA⊕OS が上回っている．この結果は，モデルのパラメタを変更しない SWEA⊕OS の利点を強調するものである．

その他　インコンテキスト学習手法である ICE は，モデルへの入力に対して操作を行うという点で SWEA⊕OS に似たアプローチだと言えるが，すべての指標で SWEA⊕OS が上回った．この結果は自然言語によって知識を追加する ICE に対して，埋め込みへ知識を追加する SWEA⊕OS の有効性が表れている．一方で Fine-Tuning の改良型の手法である AdaLoRA と FT-M はいずれも最大の Edit Succ. を示しており，知識追加における Fine-Tuning の威力を証明している．その他の評価指標についても SWEA⊕OS は下回ったが，部分的なモデルの再学習を行う AdaLoRA や FT-M との比較であると考えれば，一定の競争力を持った性能であると言える．

5　おわりに

本研究では未知知識を効率的に LLM へ追加する手法として，知識編集手法の中でも高いパフォーマンスを発揮した SWEA⊕OS に注目して未知知識追加タスクにおける性能を検証した．結果として SWEA⊕OS の知識追加タスクにおける性能は低かったものの，既存手法に準ずる性能を示した．加えて，その簡潔で軽量な手法は高い計算コストやストレージコストを要求する既存手法に対する明確な利点である．そのため，SWEA⊕OS は効率的な知識追加手法を実現するうえで有望な手法の 1 つであると言える．

今後の課題としては，知識追加において SWEA⊕OS がどのようにモデルへ作用しているかを分析し，修正と追加の間で性能にギャップが見られた原因を特定する．また特に有用であった既存手法との組み合わせが可能か検討する．これらの課題解決を通して，主語埋め込み編集による効率的な未知知識追加手法の確立を目指すものとする．

謝辞

本研究は，JST 創発的研究支援事業 JPMJFR232K，および中島記念国際交流財団の助成を受けたものです．

参考文献

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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. **Advances in Neural Information Processing Systems**, Vol. 2020-December, , 5 2020.

[2] OpenAI. Gpt-4 technical report. **arXiv preprint arXiv:2303.08774**, 3 2023.

[3] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet,

Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. **arXiv preprint arXiv:2302.13971**, 2 2023.

[4] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models. **arXiv preprint arXiv:2303.18223**, 3 2023.

[5] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. **NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference**, Vol. 1, pp. 4171–4186, 10 2018.

[6] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. **Proceedings - 2024 Conference on AI, Science, Engineering, and Technology, AIxSET 2024**, pp. 166–169, 12 2023.

[7] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. **Transactions on Machine Learning Research**, Vol. 2024, , 3 2024.

[8] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. **Advances in Neural Information Processing Systems**, Vol. 35, , 5 2022.

[9] Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. xrag: Extreme context compression for retrieval-augmented generation with one token. **Advances in Neural Information Processing Systems**, Vol. 37, pp. 109487–109516, 12 2024.

[10] Zirui Song, Bin Yan, Yuhan Liu, Miao Fang, Mingzhe Li, Rui Yan, and Xiuying Chen. Injecting domain-specific knowledge into large language models: A comprehensive survey. **Findings of the Association for Computational Linguistics: EMNLP 2025**, pp. 25297–25311, 11 2025.

[11] Jan-Micha Bodensohn, Ulf Brackmann, Liane Vogel, Anupam Sanghi, and Carsten Binnig. Unveiling challenges for llms in enterprise data engineering. **PVLDB**, Vol. 19, pp. 196–209, 11 2025.

[12] Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. Knowledge editing for large language models: A survey. **ACM Computing Surveys**, Vol. 57, , 10 2023.

[13] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. **In Proceedings of the Advances in Neural Information Processing Systems**, 2022.

[14] Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, Kangwei Liu, Yuansheng Ni, Guozhou Zheng, and Huajun Chen. Easyedit: An easy-to-use knowledge editing framework for large language models. **Proceedings of the Annual Meeting of the Association for Computational Linguistics**, Vol. 3, pp. 82–93, 8 2023.

[15] Xiaopeng Li, Shasha Li, Shezheng Song, Huijun Liu, Bin Ji, Xi Wang, Jun Ma, Jie Yu, Xiaodong Liu, Jing Wang, and Weimin Zhang. Swea: Updating factual knowledge in large language models via subject word embedding altering. **Proceedings of the AAAI Conference on Artificial Intelligence**, Vol. 39, pp. 24494–24502, 1 2024.

[16] Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, Xiaowei Zhu, Jun Zhou, and Huajun Chen. A comprehensive study of knowledge editing for large language models. **arXiv preprint arXiv:2401.01286**, 1 2024.

[17] Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. Evaluating the ripple effects of knowledge editing in language models. **Transactions of the Association for Computational Linguistics**, Vol. 12, pp. 283–298, 7 2023.

[18] Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. Memory-based model editing at scale. **Proceedings of Machine Learning Research**, Vol. 162, pp. 15817–15831, 6 2022.

[19] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale. **ICLR 2022 - 10th International Conference on Learning Representations**, 10 2021.

[20] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. **11th International Conference on Learning Representations, ICLR 2023**, 10 2022.

[21] Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities. **EMNLP 2023 - 2023 Conference on Empirical Methods in Natural Language Processing, Proceedings**, pp. 10222–10240, 5 2023.

[22] Yuval Pinter and Michael Elhadad. Emptying the ocean with a spoon: Should we edit models? **Findings of the Association for Computational Linguistics: EMNLP 2023**, pp. 15164–15172, 10 2023.

[23] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. **Proceedings of the Annual Meeting of the Association for Computational Linguistics**, Vol. 1, pp. 8493–8502, 4 2021.

[24] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael, Smith Ranjan, Subramanian Xiaoqing, Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. **arXiv preprint arXiv:2307.09288**, 7 2023.