モデルマージを用いた LLM 翻訳における破滅的忘却の抑制

岩川光一 [1]　Haocheng Zhu[1]　鈴木潤 [1]　永田昌明 [2]
[1] 東北大学　[2] 日本電信電話株式会社
{iwakawa.koichi.q5，zhu.haocheng.r4}@dc.tohoku.ac.jp　jun.suzuki@tohoku.ac.jp
masaaki.nagata@ntt.com

概要

本稿では翻訳能力を向上させるデータを用いて継続事前学習を行ったモデルを対象に，モデルマージを用いて破滅的忘却を抑制する方法について検討する．継続事前学習前後のモデルをマージすることにより，一般的タスク能力の忘却を抑えつつ，翻訳能力をベースモデルよりも向上させられることを示す．さらに，モデルマージの手法間の比較や，各マージ手法のパラメータ設定による結果の変化についても調査を行い，今後の研究の方向性を示す．

1　はじめに

事前学習済みの言語モデルに対し，特定のタスクやドメインに特化したデータを用意し，それらを学習させることにより，当該タスクやドメインに対する性能を向上させることができる．例えば，文献 [1] では，事前学習済みの英語中心の言語モデルに対し，日本語のコーパスを用いて継続事前学習を行うことにより，日本語能力を高めている．このように事前学習済み言語モデルに対して，別のデータを用意して継続して事前学習を行うことを一般的に継続事前学習と呼ぶ．

継続事前学習を用いることで，言語モデルをランダム初期値から訓練する場合に比べ，少ない計算コストで特定のタスクの性能が高い言語モデルを獲得することができる．一方で，継続事前学習を行うことで，事前学習済み言語モデルが獲得していた一部の能力を失ってしまう破滅的忘却が発生することが報告されている [2]．このような背景から，文献 [3] や [4] など，継続事前学習の欠点となる破滅的忘却を解消または軽減する方法が盛んに研究されるようになった．本稿でも継続事前学習における破滅的忘却を抑制する方法について議論する．

本研究では，これまで主に提案されていきた手法は，計算資源を多く必要とする方法であり，限られた計算資源しかない状況で効果を得ることが難しいという課題があることに着目し，限られた計算資源で継続事前学習における破滅的忘却を抑える方法を検討する．具体的には，モデルマージ [5] の考えを活用した方法を提案し，翻訳能力を向上させるデータを用いて継続事前学習を行い，ベースとなる事前学習済みモデルよりも翻訳能力を向上させつつ，他のタスクの性能を維持できるかを実験により検証する．

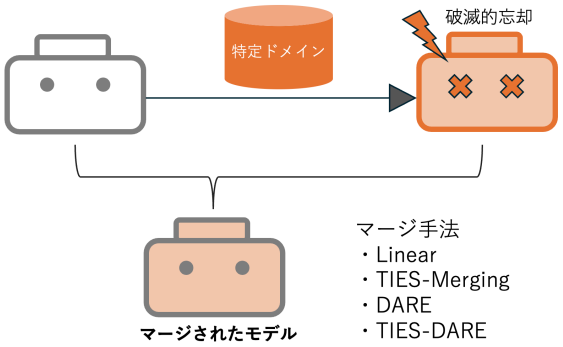

図 1　破滅的忘却を抑制するモデルマージ

2　LLM 翻訳

機械学習モデルを用いて翻訳を行う方法については文献 [6] など，従来より盛んに研究されている．近年においても，大規模言語モデルを用いた翻訳に関する手法が [7]，[8] などで提案されている．[7] では，対訳コーパスを用いてモデルをファインチューニングする以前に，翻訳の対象となる言語の単言語コーパスを用いて継続事前学習することにより，著しい翻訳精度の向上が報告されている．[8] では，対訳データを用いた継続事前訓練により，日英方向の翻訳能力と英日方向の翻訳能力をそれぞれ向上させる方法が提案されている．一方，これらの研究において，事前学習済みモデルに翻訳能力を与える目的で訓練する以前の，一般タスクに対する性能の維持については十分に考慮されていない．そのため，

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

本研究では，事前学習済みモデルに対しフルパラメータで継続事前学習を行い，モデルマージの考えを用いた各手法を用いて，破滅的忘却を抑える方法について検証を行う．

3 モデルマージ

本章ではモデルマージに一般的に用いられ，本研究で検証を行う手法を説明する．

3.1 Linear

複数のモデルのパラメータを加算平均する手法である．単純な手法ではあるが，[5] において，モデルの性能を向上させている．以下の数式で示すことができる．

$$\theta_m = \sum_{i=1}^{N} \alpha_i \theta_i \tag{1}$$

ここで，θ_m はマージを行った後のモデルパラメータ，N はマージを行うモデルの数，α_i は i 番目のモデルの重み，θ_i はマージの元とするそれぞれのモデルのパラメータとする．

3.2 TIES-Merging

[9] において提案された，ファインチューニングを行なったモデル同士のマージを効率化するための手法である．θ_{init} をベースモデルとし，θ_1，θ_2 をそれぞれ独立にファインチューニングを行なったモデルとする．(1) まず，ファインチューニングを行なったモデルとベースモデルのパラメータの差分 $\tau_1 = \theta_1 - \theta_{\text{init}}$，$\tau_2 = \theta_2 - \theta_{\text{init}}$ を計算する．その後，τ_1，τ_2 からそれぞれ絶対値の大きい上位 k% のパラメータのみを残し，$\hat{\tau}_1$，$\hat{\tau}_2$ とする．(2) $\hat{\tau}_1$，$\hat{\tau}_2$ それぞれに含まれる n 番目のパラメータについて，符号の異なるパラメータを足し合わせて小さな値になることを防ぐために，$\gamma^n = \text{sgn}(\hat{\tau}_1^n + \hat{\tau}_2^n)$ を n 番目のパラメータの符号とする．ここで，$\text{sgn}(a)$ は a の符号を表す．(3) $\hat{\tau}_1$，$\hat{\tau}_2$ それぞれに含まれる，n 番目のパラメータに関して，(2) で選択した符号側にあるパラメータの平均を最終的にベースモデルに足し合わせるベクトル τ_m^n とする．この時，$\tau_m^n = avg(S^n)$ と表すことができる．ただし，$S^n = \{\hat{\tau}_i^n | \text{sgn}(\hat{\tau}_i^n) = \gamma^n, i = 1, 2\}$．そして，最終的にマージを行なったモデルパラメータ θ_m は，

$$\theta_m = \theta_{\text{init}} + \tau_m \tag{2}$$

と表される．

3.3 DARE

異なるタスクに向けてファインチューニングを行なったそれぞれのモデルをマージすることを前提に考案された手法 [10] である．ファインチューニングを行なったモデルのパラメータの一部を，ランダムにファインチューニングを行う以前のモデルの初期値に変更し，出力の大きさを一定に保つようにスケーリングを行った後，マージを行う．TIES-Merging と組み合わせて用いられることもあり，以降ではそれを TIES-DARE と表す．

4 実験設定

本実験では，事前学習済みモデルに対して対訳コーパスにより継続事前学習を行い，継続事前学習を行う前後のモデルをマージした時の一般タスクに対する性能と翻訳タスクに対する性能の変化を観察する．

4.1 翻訳コーパスによる継続事前学習

事前学習済みモデルとして，llm-jp-v3-3.7b，llm-jp-v3-3.7b-instruct[11] の 2 種類を用いる．llm-jp-v3-3.7b は日本語と英語合わせて 3T トークン程度で事前学習された，バイリンガルな大規模言語モデルである．llm-jp-v3-3.7b-instruct は上記モデルに対して指示チューニングを行ったもので，各種一般タスクでより高い性能を示す．以降，それぞれのモデルを通常モデル，指示チューニングモデルと呼ぶことにする．本実験では，モデルが元々持っている性能を落とさずに新たな能力を付与できるかを検証するため，指示追従能力が加わった後者のモデルに対しても検証を行う．今後，継続事前学習を行ったモデルと区別する目的で，これらのモデルをベースモデルと呼ぶ．ベースモデルそれぞれに対し，翻訳能力を向上させるために JParaCorpus-v3[12] を用いて継続事前学習を行う．JParaCorpus-v3 は日英の対訳を集めた高品質なデータセットであり，日英翻訳と英日翻訳の双方向に対してそれぞれ独立に 300 万文対（約 0.18B トークン）の学習を行う．継続事前学習における設定は付録 A に示す．以上の設定で継続事前学習を行うことにより，それぞれのベースモデルに対し，日英翻訳モデル，英日翻訳モデルが完成し，ベースモデル含め 6 つのモデルを使用してモデルマージの検証を行う．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

表 1　継続事前学習を行ったモデルとその後にマージを行なったモデルの比較

		llm-jp-eval		ALT					
				ja-en			en-ja		
		AVG	MT	BERTscore	BLEU	COMET	BERTscore	BLEU	COMET
Base	base	**0.395**	0.824	0.938	12.5	0.856	0.850	10.1	0.892
	ja-en	0.378	0.816	**0.951**	**16.8**	**0.881**	0.852	10.3	0.892
	en-ja	0.370	0.783	0.838	7.07	0.771	**0.863**	**11.0**	0.897
	merged	0.388	**0.830**	0.941	14.7	0.859	0.860	10.3	**0.900**
Instruct	base	**0.471**	0.838	0.942	13.0	0.864	0.851	8.86	0.898
	ja-en	0.393	0.753	**0.948**	**15.5**	**0.874**	0.787	6.02	0.752
	en-ja	0.411	0.798	0.899	10.8	0.818	**0.862**	**10.6**	0.899
	merged	0.448	**0.841**	0.945	14.4	0.868	0.857	10.4	**0.900**

4.2　モデルマージの検証

モデルマージは各ベースモデルごとに独立して行う．例えば，通常モデルに対しては，通常モデルをもとに継続事前学習した日英翻訳モデル，英日翻訳モデルの 3 種類のモデルに対してモデルマージの検証を行う．モデルマージには mergekit[13] を用いる．mergekit は，モデルマージの様々な手法が実装されたパッケージで，本稿で触れている手法に関しては全て実装されている．各モデルマージ手法を適用する際に用いたパラメータ設定については付録 B に示す．モデルの評価には，llm-jp-eval[14] を用いる．llm-jp-eval は様々な日本語タスクに関して評価を行うことができるツールキットであり，多様な日本語タスクに対する言語モデルの性能を測ることができる．本実験では，llm-jp-eval の各タスクにおける性能を平均した AVG と，翻訳タスクの性能を示す MT を主に扱う．llm-jp-eval に含まれるタスクについては付録 C で述べる．

5　実験結果

5.1　モデルマージによる破滅的忘却の抑制

はじめに，翻訳モデルを継続事前学習した結果と，継続事前学習を行う以前のモデル（ベースモデル）を含めてモデルマージを行った結果を確認する．各モデルを llm-jp-eval で評価した結果を表 1 に示す．表の最左列はベースモデルの種類を示し，Base は通常モデル，Instruct は指示チューニングモデルを表す．左から 2 列目には，それぞれのモデルの状態を示している．base は継続事前学習を行う前のモデル，ja-en, en-ja はそれぞれ独立に，日英，英日翻訳コーパスで継続事前学習したモデルである．また，merged は，base, ja-en, en-ja をそれぞれ均等に Linear マージしたモデルである．評価指標に関しては，llm-jp-eval の各タスクの平均である AVG，翻訳精度の指標 MT に加え，llm-jp-eval の翻訳タスクに含まれる， ALT 対訳コーパスにおける，各方向の翻訳精度を示す指標（付録 D）を表に載せている．ベースモデルの種別ごとに，各指標ごとに最も数値が高いものは太字，次点のものに下線を引いている．まず， 継続事前学習のみを行ったモデル ja-en, en-ja に関しては，それぞれの方向の翻訳性能が向上する一方，一般タスクの性能と，逆方向の翻訳精度が落ちており，破滅的忘却が起こっているといえる.. 特に，指示チューニングを行なったモデルに対して継続事前学習を行った場合，元モデルよりも大きく一般タスクに対する性能が劣化している．そして，各種マージを行なったモデルに関しては，継続事前学習を行ったモデルに比べ llm-jp-eval における AVG が向上している．さらに，双方向の翻訳タスクにおいても，ベースとなるモデルを全ての指標で上回り，各翻訳方向に継続事前学習を行ったモデルに次ぐ精度を示している．このことから，モデルマージにより，モデルの破滅的忘却を抑制しつつ，特定タスクに対する能力を得られることがわかる．

5.2　各モデルマージ手法の比較

次に各種マージ手法における性能の差について表 2 を確認する．全体的な結果として，Linear マージ以外の手法では，一般タスクの性能を示す AVG が

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

表 2　マージ手法間での一般タスクと翻訳タスクにおける性能の比較

		llm-jp-eval	
	設定	AVG	MT
Base	base	**0.395**	0.824
	Linear	0.388	**0.830**
	TIES	0.375	0.819
	DARE	0.377	0.827
	TIES-DARE	0.380	0.824
Instruct	base	**0.471**	0.838
	Linear	0.448	**0.841**
	TIES	0.395	0.831
	DARE	0.407	0.834
	TIES-DARE	0.396	0.832

継続事前学習のみを行ったモデルとほとんど変わらない．つまり，破滅的忘却を抑制できていないといえる．また，翻訳タスクの性能においても Linear マージを上回る結果は出ていない．Linear マージに関しては，一般タスクと翻訳タスクの性能において継続事前学習以前のモデルと比べてトレードオフの関係になっている．この結果を受け，Linear マージと TIES-Merging に関して，異なるパラメータ設定における検証を行った．

5.3　モデルマージのパラメータ探索

本節では，マージ手法のうち Linear と TIES-Merging について，マージにおけるパラメータ設定を変更することによる一般性能と翻訳性能の変化を確認する．マージのベースモデルとしては，Instruct モデルを使用する．はじめに，Linear におけるパラメータの変化における一般性能と翻訳性能の変化を確認する．Linear マージでは，モデルを混ぜる割合を変化させる．図 2 から，ベースモデルの割合を増やすと，一般タスクに対する性能が上がり，日英，英日両方向の翻訳における BLEU が減少していることがわかる．このことから，本実験におけるベースモデル，そして継続事前学習を行ったモデル間での Linear マージでは，一般タスクに対する性能と翻訳タスクにおける BLEU がトレードオフの関係にあることがいえる．続いて，TIES-Merging におけるパラメータの変化における結果を確認する．TIES-Merging のパラメータについては，前節では一般タスクの性能をマージによって回復できていないことが示された．よって，一般タスクの性能を伸ばすために，ベースモデルの割合を増やす方向への探索を行う．図 2 下部に結果を示している．ベースモデルのパラメータの割合を増やしても，一般タスクに対する性能が大きく向上することはなく，Linear よりも低い結果となった．一方，翻訳タスクに関してはベースモデルの割合を増やしても単調に下がることはなかった．よって，TIES-Merging に関しては，ベースモデルの比率がタスクにおける性能に与える影響は，単純ではないことがわかる．

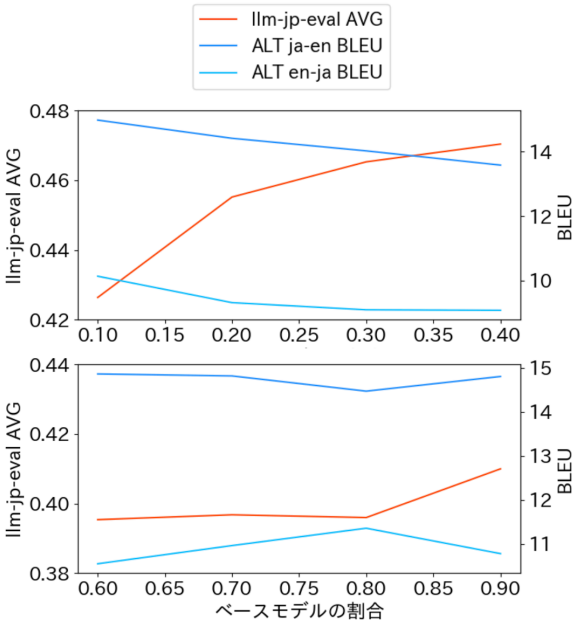

図 2　Linear（上）と TIES-Merging（下）のベースモデルの割合ごとの一般タスクと翻訳タスクに対する性能

6　おわりに

本稿では，継続事前学習における破滅的忘却を，モデルマージを用いて抑制する方法について検証を行った．実験の結果から，継続事前学習を行う以前のベースモデルと，継続事前学習を行った後のモデルのパラメータ平均を取ることにより，ベースモデルの持つ一般的タスクに対する性能を保ちつつ，新たな能力を得られることがわかった．今後の課題として，本実験で得られたモデルは，ベースモデルの持っていた能力を完全に保てているとはいえず，トレードオフの関係にあるため，新たなタスクへの性能を求めれば求めるほど，基本性能は悪化する可能性がある．加えて，本実験では継続事前学習を限られたトークン数で実行したので，より大規模な継続事前学習においても，ベースモデルの性能を保つ方法を検討する必要がある．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

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) の支援を受けたものです．本研究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ました．本研究成果（の一部）は，データ活用社会創成プラットフォーム mdx [15] を利用して得られた物です．

参考文献

[1] Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. Continual pre-training for cross-lingual LLM adaptation: Enhancing japanese language capabilities. In **First Conference on Language Modeling**, 2024.

[2] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. **Advances in neural information processing systems**, Vol. 30, , 2017.

[3] Kshitij Gupta, Benjamin Thérien, Adam Ibrahim, Mats Leon Richter, Quentin Gregory Anthony, Eugene Belilovsky, Irina Rish, and Timothée Lesort. Continual pre-training of large language models: How to re-warm your model? In **Workshop on Efficient Systems for Foundation Models @ ICML2023**, 2023.

[4] Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben allal, Leandro Von Werra, and Martin Jaggi. Scaling laws and compute-optimal training beyond fixed training durations. In **Workshop on Efficient Systems for Foundation Models II @ ICML2024**, 2024.

[5] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, **Proceedings of the 39th International Conference on Machine Learning**, Vol. 162 of **Proceedings of Machine Learning Research**, pp. 23965–23998. PMLR, 17–23 Jul 2022.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In **Proceedings of the 31st International Conference on Neural Information Processing Systems**, NIPS'17, p. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[7] Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. A paradigm shift in machine translation: Boosting translation performance of large language models. In **The Twelfth International Conference on Learning Representations**, 2024.

[8] Minato Kondo, Takehito Utsuro, and Masaaki Nagata. Enhancing translation accuracy of large language models through continual pre-training on parallel data. In Elizabeth Salesky, Marcello Federico, and Marine Carpuat, editors, **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.

[9] Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In **Thirty-seventh Conference on Neural Information Processing Systems**, 2023.

[10] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In **Forty-first International Conference on Machine Learning**, 2024.

[11] Akiko Aizawa, Eiji Aramaki, Bowen Chen, Fei Cheng, Hiroyuki Deguchi, Rintaro Enomoto, Kazuki Fujii, Kensuke Fukumoto, Takuya Fukushima, Namgi Han, Yuto Harada, Chikara Hashimoto, Tatsuya Hiraoka, Shohei Hisada, Sosuke Hosokawa, Lu Jie, Keisuke Kamata, Teruhito Kanazawa, Hiroki Kanezashi, Hiroshi Kataoka, Satoru Katsumata, Daisuke Kawahara, Seiya Kawano, Atsushi Keyaki, Keisuke Kiryu, Hirokazu Kiyomaru, Takashi Kodama, Takahiro Kubo, Yohei Kuga, Ryoma Kumon, Shuhei Kurita, Sadao Kurohashi, Conglong Li, Taiki Maekawa, Hiroshi Matsuda, Yusuke Miyao, Kentaro Mizuki, Sakae Mizuki, Yugo Murawaki, Ryo Nakamura, Taishi Nakamura, Kouta Nakayama, Tomoka Nakazato, Takuro Niitsuma, Jiro Nishitoba, Yusuke Oda, Hayato Ogawa, Takumi Okamoto, Naoaki Okazaki, Yohei Oseki, Shintaro Ozaki, Koki Ryu, Rafal Rzepka, Keisuke Sakaguchi, Shota Sasaki, Satoshi Sekine, Kohei Suda, Saku Sugawara, Issa Sugiura, Hiroaki Sugiyama, Hisami Suzuki, Jun Suzuki, Toyotaro Suzumura, Kensuke Tachibana, Yu Takagi, Kyosuke Takami, Koichi Takeda, Masashi Takeshita, Masahiro Tanaka, Kenjiro Taura, Arseny Tolmachev, Nobuhiro Ueda, Zhen Wan, Shuntaro Yada, Sakiko Yahata, Yuya Yamamoto, Yusuke Yamauchi, Hitomi Yanaka, Rio Yokota, and Koichiro Yoshino. Llm-jp: A cross-organizational project for the research and development of fully open japanese llms. **CoRR**, Vol. abs/2407.03963, , 2024.

[12] Makoto Morishita, Katsuki Chousa, Jun Suzuki, and Masaaki Nagata. JParaCrawl v3.0: A large-scale English-Japanese parallel corpus. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk, and Stelios Piperidis, editors, **Proceedings of the Thirteenth Language Resources and Evaluation Conference**, pp. 6704–6710, Marseille, France, June 2022. European Language Resources Association.

[13] Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vladimir Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. Arcee's MergeKit: A toolkit for merging large language models. In Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina, editors, **Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track**, pp. 477–485, Miami, Florida, US, November 2024. Association for Computational Linguistics.

[14] llm-jp-eval: 日本語大規模言語モデルの自動評価ツール, 言語処理学会第 30 回年次大会 (NLP2024), March 2024.

[15] Toyotaro Suzumura, Akiyoshi Sugiki, Hiroyuki Takizawa, Akira Imakura, Hiroshi Nakamura, Kenjiro Taura, Tomohiro Kudoh, Toshihiro Hanawa, Yuji Sekiya, Hiroki Kobayashi, Yohei Kuga, Ryo Nakamura, Renhe Jiang, Junya Kawase, Masatoshi Hanai, Hiroshi Miyazaki, Tsutomu Ishizaki, Daisuke Shimotoku, Daisuke Miyamoto, Kento Aida, Atsuko Takefusa, Takashi Kurimoto, Koji Sasayama, Naoya Kitagawa, Ikki Fujiwara, Yusuke Tanimura, Takayuki Aoki, Toshio Endo, Satoshi Ohshima, Keiichiro Fukazawa, Susumu Date, and Toshihiro Uchibayashi. mdx: A cloud platform for supporting data science and cross-disciplinary research collaborations. In **2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)**, pp. 1–7, 2022.

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

表 3　継続事前学習のハイパーパラメータ

最適化手法	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95, \epsilon = 1.0e-8$)
学習率（最大）	$3.5e-05$
学習率（最小）	$3.5e-06$
学習率のウォームアップ	70 ステップ
学習率のスケジューラ	コサインスケジューラ
勾配のクリッピング	1.0
ドロップアウト	0.1
バッチサイズ	256
窓長	1024 トークン

表 4　モデルマージのパラメータ設定

手法名	モデルの種類	weight	density
Linear	base	0.33	-
	ja-en	0.33	-
	en-ja	0.33	-
TIES	ja-en	0.50	0.50
	en-ja	0.50	0.50
DARE	base	0.33	-
	ja-en	0.33	-
	en-ja	0.33	-
TIES-DARE	ja-en	0.50	0.50
	en-ja	0.50	0.50

A　継続事前学習の設定

継続事前学習の際に用いたハイパーパラメータを表 3 に示す．継続事前学習には NVIDIA Tesla A100 40GB 2 枚，もしくは NVIDIA H100 80GB 1 枚を用いた．コードに関しては，llm-recipes[1] をベースに改変したものを用いた．

B　各マージ手法のパラメータ設定

5.2 節で各マージ手法の比較の際に用いたそれぞれのマージ手法のパラメータ設定を表 4 に示す．weight は，マージの元となる各モデルにつける重みの大きさである．また，density は，3.2 節で説明した TIES-Merging において，絶対値の多い上位 k% のパラメータを選ぶプロセスでの，パラメータを選ぶ割合である．

C　llm-jp-eval の評価指標

llm-jp-eval で評価を行うタスクのカテゴリの一覧を表 5 に示す．各カテゴリごとに複数個のデータセットを用いて主に正答率で評価を行う．それぞれのカテゴリに含まれるデータセットに関しては [14] に記されている．これら 10 カテゴリの評価の平均を AVG として本稿では扱っている．翻訳評価のカテゴリ MT に関して，このカテゴリにはアジア言語ツリーバンクと Wikipedia 日英京都関連文書対訳コーパスそれぞれのデータセットが用いられている．

表 5　llm-jp-eval の評価カテゴリ一覧

NLI	自然言語推論
QA	質問応答
RC	読解
MC	選択式質問応答
EL	エンティティリンキング
FA	基礎解析
MR	数学的推論
MT	機械翻訳
HE	試験問題
CG	コード生成

D　翻訳精度の評価指標

BERTscore　事前学習済みモデルから得られるトークンのベクトル表現を利用して，テキスト間の類似度を測る手法．従来手法と比べ人手評価との相関が高いという結果が出ている．[2]

BLEU　生成したテキストの中の n-gram が正解テキストの中にどれだけ含まれているかを測る手法．従来より機械翻訳の評価にしばしば使われている．

COMET　人間による評価を学習した機械学習モデルを用いて評価を行う指標．[3]

1）https://github.com/okoge-kaz/llm-recipes

2）https://arxiv.org/pdf/1904.09675

3）https://arxiv.org/pdf/2009.09025

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).