医療分野に特化した日本語の小規模言語モデルの開発

渡辺翔吾[1] 連乃駿[1] 今岡幸弘[1] 飯原弘二[1]
[1]国立循環器病研究センター
{watanabe.shogo,n.ren,imaoka.yukihiro,kiihara}@ncvc.go.jp

概要

本研究では，日本語の医療分野に特化した小規模言語モデルの開発を行った．日本語に焦点を絞ったテキストクリーニングと形態素解析，教科書相当の品質に分類されたテキストデータ，疾病・薬剤に関する話題についての合成テキストデータを使用し，軽量化に重きをおいた 1B の言語モデルの事前学習を実施した．このモデルをベースモデルとして指示ファインチューニングしたモデルを IgakuQA と JMED-LLM を用いて評価を行った．ファインチューニングモデルにおいて，JMED-LLM の 8 つの評価指標のうち，6 つのタスクで既存の大規模言語モデルより高いスコアを示した．この結果から，ネットワークや計算資源が限られた環境において，特定の分野に特化した小規模言語モデルの運用が選択肢の 1 つになり得る可能性が示唆された．

1 はじめに

2023 年から 2024 年にかけて，Transformer [1]ベースの大規模言語モデル(LLM)の開発は各社・研究機関が精力的に開発を行っており，多様なタスクや評価指標において，人間レベルに匹敵あるいは上回る結果を出している．これらの成果を基にして，医学知識に富んだモデルの開発も行われている．[2,3,4]

しかしながら，LLM は高い性能を発揮する反面，計算コストが高いなど，運用面での欠点も抱えており，個人情報漏洩のリスクなどの観点からネットワークから切り離されている環境下での運用が難しく，費用面においても導入が容易ではない場合が考えられる．

LLM が注目される一方で，Microsoft の phi シリーズを筆頭に，パラメータを抑えて計算コストを削減した小型言語モデル（SLM）において，パフォーマンスの向上が報告されている．

特に，phi-1 [5]で提案された Textbook approach は，言語モデルのトレーニングに教科書品質のデータのみを使用することでモデルの品質を高めることに加え，より大規模な言語モデルを用いて生成した合成テキストデータを用いることで，コード生成タスクにおいて，より大きなモデルと同等の優れたパフォーマンスを発揮することを示した．

本研究では，計算資源の限られた医療機関等においても，高い専門性と軽快動作を両立する小規模言語モデルを目指して開発を行った．

2 方法

2.1 データセット

本研究において開発した SLM の事前学習では，広く使用されるコーパスである日本語 Wikipedia と OSCAR を，一般的な日本語のコーパスとして使用するとともに，医療分野の内容に絞ってスクレイピングしたテキストデータと，よりパラメータ数の多い既存の言語モデルを用いて生成した疾病・薬剤に関する合成テキストデータを用いることで，医学知識の増強を図った．

2.2.1 日本語コーパス

日本語 Wikipedia [6]は，各記事の本文のみを抽出して使用した．

OSCAR[7]は，不適切な内容を含む文章，内容が途切れている文章のスクリーニング，重複する表現や文の削除など，いくつかの強力な追加のフィルタリングを適用した．加えて，既存の言語モデルを使用した品質フィルタリングを行った．最初に，フィルタリング済みコーパスから 10 万件の文章をランダムサンプリングし，Llama-3-ELYZA-JP-8B-instruct[8]を用いてテキストの品質アノテーションを実施した．

次に，すべての文書に対して，同モデルの入力の最終トークンの最終層ベクトルを，入力された文章の特徴量として求めた．

分類器は Random Forest [9]を使用し，埋め込みベクトルとアノテーションラベルで訓練した後，テキストの品質フィルタリングを行った．

2.1.2 医学テキスト

2 つ目のコーパスとして，Web からスクレイピングした教科書品質の医学に関するテキストを収集した．しかし，テキスト量はわずか 49MB であったため，医学に関するテキストの不足を補う目的で，合成教科書アプローチを実施した．

合成教科書には，万病辞書[10]に収録されている多様な病名と，厚生労働省のウェブサイトで公表されている薬価基準品目リストに収録されている薬剤について，テキスト生成を行った．合成教科書の作成には，5 つの言語モデルを使用して，91000 個の病名と 9000 種類の薬に関する合成教科書を作成した．

さらに，ELYZA-Llama3-8B-instruct を用いて，日本の医師国家試験や薬剤師国家試験を模した合成問題集を作成した．

表 1 に，事前学習に使用したデータとトークン数をまとめている．最終的に，ユニークなトークンの総数は約 9B となった．

表 1　事前学習に使用したテキストデータ内訳

	トークン数
日本語コーパス	
日本語 Wikipedia	0.55 B
OSCAR	8.20 B
医学テキスト	
Web スクレイピング	0.01 B
合成教科書	0.16 B
合成問題集	0.07 B
	8.99 B

2.2 トークナイザー

トークン化にあたって，テキストデータの品質担保と日本語への特化を目的とした，テキストクリーニング，形態素解析，サブワード化を行った．

2.2.1 テキストクリーニング

句読点は全角の句読点記号に統一した．Unicode 正規化を Normalization Form Compatibility Composition (NFKC) で行い，ハイフン，長音符，チルダなどの表記ゆれを統一した．同時に，NFKC により分解された三点リーダーや温度記号を復元し，正規表現を用いて電話番号やメールアドレス，URL，@アカウント名，#タグなどの個人情報や Web 固有の表記を削除し，類似記号の統一，必要となる文字以外の削除を実施した後，冗長な表現に使用される文字・記号を削除する．最後に，半角スペースをメタスペースに置き換える．

2.2.2 形態素解析

先行研究に基づき，形態素解析器として MeCab [11]を使用した．Neologd などの最新語を含むカスタム辞書は使用しない代わりに，医学用語に特化するため，万病辞書を参考にしてカスタム辞書を作成した．

2.2.3 トークン化

トークナイザーには Unigram [12]を採用した．事前に形態素解析により半角スペースで区切られたテキストを分割してからサブワード化を行う．語彙サイズは，3 つの特殊トークン（<|begin_of_text|>, <|end_of_text|>, 改行コード ¥n）を含めて，32768 とした．

トークナイザーの学習には，医学的な内容を優先するため，日本語 Wikipedia と医学コーパスを使用した．

2.3 言語モデル

SLM は phi シリーズや LLaMA シリーズをベースとした．トークン埋め込み層，最終出力層を除いて，24 層の Transformer Decoder のみで構成され，次元数は 2048，Feedforward Network 内の次元数は 8192，Self-Attention においては各 64 次元の 32 マルチヘッドとした．事前学習時の最大シーケンス長は 2048 とした．位置エンコーディングには Rotary Positional Encoding (RoPE) [13]を用いて，学習を安定させるため， pre-normalization [14]を採用した．パラメータの総数は約 1B（1.2B）となった．

活性化関数は，Sigmoid Linear Units (SiLU) [15]を使用し，Layer normalization は，LLaMA シリーズに倣って，root mean square normalization (RMSNorm) [16]に置き換えた．同様に，Multi-head attention (MHA) は，Grouped Query Attention (GQA) [17]を採用した．グループ数は 8 とした．

ほぼすべてのパラメータの初期化に small init[18]を使用した．トークン埋め込み層の直後に scaled embedding を適用し，scaled initialization[19]を併用した．また，最終出力層のパラメータは Xavier normal で初期化した．なお，バイアスパラメータはすべての層で無効とした．

2.4 学習

2.4.1 事前学習

事前学習は Chinchill 則[20]や phi-1 に基づき, 3200 ステップごとに保存しながら，合計 24000 ステップで実施した．これは約 50B トークン（9B のユニークなトークン）を見たことに相当する．

学習には flash attention 2[21]，分散データ並列化，DeepSpeed の zero redundant optimizer (ZeRO) stage 2[22] を用いて，バッチサイズ 1024（= 1GPU あたり 8 つのミニバッチ，4GPU，32 の勾配蓄積）とした．他の LLM の学習と同様に，計算効率向上のため，入力トークンは最大系列長 2048 に詰め込んだ．

また，繰り返しトークンを用いた場合の汎化性能低下を軽減するため，Dropout を 0.1 で適用した．損失関数はクロスエントロピーを使用した．

最適化アルゴリズムは AdamW[23] を選択し，$\beta 1$, $\beta 2$，重み減衰はそれぞれ 0.9，0.95，0.1 とした．学習率のスケジューリングには WarmupCosineLR を使用し, 750 グローバルステップで 1e-3 まで上昇させ，その後コサイン減衰で 0 とした．

事前学習は，4 台の NVIDIA 6000 Ada を使用して 14 日間を要した．

2.4.2 ファインチューニング

ファインチューニングは，instruction tuning [24]で行った．データセットは，テストサンプルを除く JMED-LLM データセットを使用し，ベースモデルとして, 合計で 20B トークンを見たもの, 合計で 50B トークンを見たものを使用した．

instruction tuning を適用する際，トークナイザーに 3 つの特殊トークン <|system|>, <|user|>, <|assistant|> を追加し，トークン埋め込み層の次元を 32768 から 32771 に拡張した．

ファインチューニング時のバッチサイズは 256, とし，損失関数は事前学習と同様にクロスエントロピーを使用した．ただし，損失は<|assistant|>の直後から<|end_of_text|>までの位置のみ計算した．

ファインチューニング時のバッチサイズは 256, とし，損失関数は事前学習と同様にクロスエントロピーを使用した．ただし，損失は<|assistant|>の直後から<|end_of_text|>までの位置のみ計算した．

最適化アルゴリズムは事前学習と同様に AdamW を使用し，重み減衰のみ 0.01 に変更した．学習率のスケジューリングは WarmupCosineLR を使用して，50 グローバルステップで 1e-4 まで上昇させ，その後コサイン減衰で 0 とした．1 台の NVIDIA A6000 を使用して 890 ステップの完了に 9 時間を要した．

また，モデルの汎化性能を向上させるため，訓練時のトークン埋め込み層適用後に noisy embeddings fine-tuning (NEFTune) [25]を適用した．ノイズの強さは α=5 に設定した．

2.5 評価

性能評価には IgakuQA[26]と JMED-LLM[27]の 2 つのベンチマークを使用した．IgakuQA は日本の医師国家試験のベンチマークであり, 2018 年から 2022 年までの 5 回の試験問題が含まれている．

JMED-LLM は LLM-JP によって提案されているベンチマークであり，医療言語処理における言語モデルの性能をさまざまな側面から評価するのに適している. 6 つのタスクのうち, JMMLU-Med, CRADE, RRTNM，SMDIS，JCSTS は Cohen's kappa と正解率で評価され，固有表現抽出(NER)タスクは，部分的または完全な F1 スコアで評価される．

3 結果

3.1 IgakuQA

表 2 に，主要な言語モデルで IgakuQA を評価した 5 年度の平均スコアを示す. 他の LLM のスコアは元論文とブログ記事[28]を基に算出している．本研究で開発したベースモデルおよびファインチューニングモデルは，合格基準と推定される 75%に達しなかった．

表 2　IgakuQA の 5 年度の平均スコア

	Avg.
GPT-4	**78.2%**
ChatGPT (gpt-3.5-turbo)	55.0%
GPT-3 (text-davinci-003)	42.1%
Llama3-Preferred-MedSwallow-70B	**79.5%**
Llama3-Swallow-70B-v0.1	70.1%
Meta-Llama-3-70B	67.3%
gemma-2-27b	63.6%
Our base (seen 20B tokens)	13.7%
Our base (seen 50B tokens)	13.8%
Our instruct (seen 20B tokens)	19.0%
Our instruct (seen 50B tokens)	18.0%

3.2 JMED-LLM

表 3 に報告されている主要 LLM の JMED-LLM におけるスコア[29]とともに開発した SLM のスコアを示す. SLM のベースモデルはすべてタスクで最低スコアであったが，インストラクトモデルは，8 つのタスクのうち 6 つ（CRADE，SMDIS，JCSTS，MRNER-disease，MRNER-disease，NRNER）で最高スコアを達成した．一方，JMMLU-Med と RRTNM のスコアは十分なファインチューニングを行った場合でも，同等程度に留まった．

4 考察

事前学習に合成問題集を含めていたものの，その効果は期待通りには得られなかった．また学習したトークン数を 20B と 50B で比較した場合にも明らかな差は見られなかった．これは，限られたユニークトークン数と，臨床医学コンテンツの少なさが，原因として考えられる．ファインチューニング後においても, スコアの大きな改善は見られなかったため, 医師国家試験の合格基準には，より大規模なモデルが必要である可能性がある．

JMED-LLM ベンチマークでは, ファインチューニングにより，6 つのタスクで高スコアを達成した．この結果は, SLM であっても特定の分野に焦点を当ててファインチューニングすることが，その分野の特定のタスクを実行するために有効な選択肢であることを示唆している．しかしながら，JMMLU-Med と RRTNM の性能は他のタスクほど向上が見られなかったため，IgakuQA 同様，モデルサイズの限界が起因していると考えられる．

5 おわりに

特定の言語，特定の分野に特化し，適切にファインチューニングされた SLM は，特定のタスクに対して有用な働きをすることができます．このタイプの SLM は，ネットワーク接続が隔離されている，あるいは計算コストが限られているローカル環境において，人間の仕事を支援することが期待される．

今後はより専門性の高い医学知識や実際の医療テキストによるファインチューニングを行いつつ，画像や音声の入力可能とするマルチモーダルモデルを行っていくことを予定している．

表 3　JMED-LLM ベンチマーク

	JMMLU-Med	CRADE	RRTNM	SMDIS	JCSTS	MRNER disease	MRNER medicine	NRNER
	Kappa(Accuracy)					Partial F1(Exact F1)		
GPT-4o-2024-08-06	**0.82(0.87)**	0.54(0.53)	**0.85(0.90)**	0.76(0.88)	0.60(0.48)	0.54(0.15)	0.42(0.26)	0.39(0.20)
GPT-4o-mini-2024-07-18	0.77(0.83)	0.21(0.37)	0.58(0.71)	0.56(0.78)	0.57(0.51)	0.48(0.13)	0.52(0.32)	0.48(0.25)
gemma-2-9b-it	0.52(0.64)	0.33(0.42)	0.54(0.68)	0.62(0.81)	0.16(0.24)	0.61(0.16)	0.65(0.42)	0.53(0.30)
Llama-3-ELYZA-JP-8B	0.34(0.51)	0.01(0.26)	0.29(0.52)	0.54(0.77)	0.02(0.18)	0.83(0.31)	0.51(0.31)	0.47(0.26)
Meta-Llama-3.1-8B-Instruct	0.31(0.49)	0.11(0.32)	0.41(0.57)	0.28(0.64)	0.13(0.23)	0.82(0.30)	0.54(0.32)	0.36(0.18)
Meta-Llama-3-8B-Instruct	0.42(0.57)	0.00(0.25)	0.37(0.54)	0.43(0.72)	0.16(0.24)	0.60(0.20)	0.44(0.25)	0.41(0.21)
Llama-3-Swallow-8B-Instruct-v0.1	0.33(0.50)	0.31(0.37)	0.33(0.55)	0.26(0.63)	0.01(0.17)	0.56(0.17)	0.37(0.21)	0.42(0.24)
Qwen2-7B-Instruct	0.42(0.57)	0.11(0.29)	0.31(0.53)	0.33(0.67)	0.37(0.31)	0.24(0.06)	0.29(0.14)	0.33(0.17)
gemma-2-2b-it	0.17(0.38)	0.00(0.25)	0.24(0.42)	0.14(0.57)	0.12(0.21)	0.66(0.20)	0.46(0.23)	0.46(0.26)
Llama-3-youko-8b-insturct	0.31(0.49)	0.02(0.28)	0.28(0.47)	0.50(0.75)	0.01(0.20)	0.02(0.00)	0.05(0.02)	0.11(0.07)
Our base (seen 20B tokens)	0.03(0.27)	-0.00(0.16)	-0.07(0.10)	0.01(0.01)	0.05(0.18)	0.00(0.00)	0.00(0.00)	0.00(0.00)
Our base (seen 50B tokens)	0.13(0.35)	-0.02(0.21)	-0.10(0.06)	0.02(0.09)	-0.03(0.12)	0.00(0.00)	0.00(0.00)	0.00(0.00)
Our instruct (seen 20B tokens)	0.19(0.39)	**0.61(0.67)**	0.36(0.53)	**0.98(0.99)**	**0.75(0.67)**	**0.87(0.35)**	**0.65(0.38)**	**0.86(0.61)**
Our instruct (seen 50B tokens)	0.26(0.44)	0.50(0.59)	0.42(0.58)	**0.98(0.99)**	**0.76(0.66)**	**0.90(0.34)**	0.55(0.34)	**0.88(0.66)**

謝辞

本研究はイドルシアファーマシューティカルズジャパンの寄付金により行われた.

参考文献

1. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. http://arxiv.org/abs/1706.03762

2. Han, T., Adams, L. C., Papaioannou, J.-M., Grundmann, P., Oberhauser, T., Löser, A., Truhn, D., & Bressem, K. K. (2023). MedAlpaca -- An Open-Source Collection of Medical Conversational AI Models and Training Data. http://arxiv.org/abs/2304.08247

3. Chen, Z., Cano, A. H., Romanou, A., Bonnet, A., Matoba, K., Salvi, F., Pagliardini, M., Fan, S., Köpf, A., Mohtashami, A., Sallinen, A., Sakhaeirad, A., Swamy, V., Krawczuk, I., Bayazit, D., Marmet, A., Montariol, S., Hartley, M.-A., Jaggi, M., & Bosselut, A. (2023). MEDITRON-70B: Scaling Medical Pretraining for Large Language Models. http://arxiv.org/abs/2311.16079

4. Saab, K., Tu, T., Weng, W.-H., Tanno, R., Stutz, D., Wulczyn, E., Zhang, F., Strother, T., Park, C., Vedadi, E., Chaves, J. Z., Hu, S.-Y., Schaekermann, M., Kamath, A., Cheng, Y., Barrett, D. G. T., Cheung, C., Mustafa, B., Palepu, A., … Natarajan, V. (2024). Capabilities of Gemini Models in Medicine. http://arxiv.org/abs/2404.18416

5. Gunasekar, S., Zhang, Y., Aneja, J., Mendes, C. C. T., del Giorno, A., Gopi, S., Javaheripi, M., Kauffmann, P., de Rosa, G., Saarikivi, O., Salim, A., Shah, S., Behl, H. S., Wang, X., Bubeck, S., Eldan, R., Kalai, A. T., Lee, Y. T., & Li, Y. (2023). Textbooks Are All You Need. http://arxiv.org/abs/2306.11644

6. GUO, Mandy, et al. Wiki-40b: Multilingual language model dataset. In: Proceedings of the Twelfth Language Resources and Evaluation Conference. 2020. p. 2440-2452.

7. Ortiz Suárez, P. J., Romary, L., & Sagot, B. (n.d.). A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages. Association for Computational Linguistics. https://oscar-corpus.com

8. https://huggingface.co/elyza/Llama-3-ELYZA-JP-8B

9. Breiman, L. (2001). Random Forests (Vol. 45).

10. Ito, K., Nagai, H., Okahisa, T., Wakamiya, S., Iwao, T., & Aramaki, E. (n.d.). J-MeDic: A Japanese Disease Name Dictionary based on Real Clinical Usage. http://www.snomed.org/snomed-ct/

11. 工藤拓. MeCab. http://mecab. sourceforge. net/, 2006.

12. Kudo, T. (n.d.). Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. Association for Computational Linguistics.

13. Su, J., et al. (2021). RoFormer: Enhanced Transformer with Rotary Position Embedding. http://arxiv.org/abs/2104.09864

14. Xiong, R., Yang, Y., He, D., Zheng, K., heng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., & Liu, T.-Y. (2020). On Layer Normalization in the Transformer Architecture. http://arxiv.org/abs/2002.04745

15. Elfwing, S., Uchibe, E., & Doya, K. (2017). Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning. http://arxiv.org/abs/1702.03118

16. Zhang, B., & Sennrich, R. (2019). Root Mean Square Layer Normalization. http://arxiv.org/abs/1910.07467

17. Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebrón, F., & Sanghai, S. (n.d.). GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. https://github.com/google/flaxformer

18. Nguyen, T. Q., & Salazar, J. (2019). Transformers without Tears: Improving the Normalization of Self-Attention. https://doi.org/10.5281/zenodo.3525484

19. Takase, S., Kiyono, S., Kobayashi, S., & Suzuki, J. (2023). Spike No More: Stabilizing the Pre-training of Large Language Models. http://arxiv.org/abs/2312.16903

20. Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. de las, Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., Driessche, G. van den, Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., … Sifre, L. (2022). Training Compute-Optimal Large Language Models. http://arxiv.org/abs/2203.15556

21. Dao, T., Fu, D. Y., Ermon, S., Rudra, A., & Ré, C. (2022). FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. http://arxiv.org/abs/2205.14135

22. Rajbhandari, S., Rasley, J., Ruwase, O., & He, Y. (2019). ZeRO: Memory Optimizations Toward Training Trillion Parameter Models. http://arxiv.org/abs/1910.02054

23. Loshchilov, I., & Hutter, F. (2017). Decoupled Weight Decay Regularization. http://arxiv.org/abs/1711.05101

24. Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., & le Google Research, Q. v. (2022). FINETUNED LANGUAGE MODELS ARE ZERO-SHOT LEARNERS Closed-Book QA. https://github.com/google-research/flan.

25. Jain, N., Chiang, P., Wen, Y., Kirchenbauer, J., Chu, H.-M., Somepalli, G., Bartoldson, B. R., Kailkhura, B., Schwarzschild, A., Saha, A., Goldblum, M., Geiping, J., & Goldstein, T. (2023). NEFTune: Noisy Embeddings Improve Instruction Finetuning. http://arxiv.org/abs/2310.05914

26. Kasai, J., Kasai, Y., Sakaguchi, K., Yamada, Y., & Radev, D. (2023). Evaluating GPT-4 and ChatGPT on Japanese Medical Licensing Examinations. http://arxiv.org/abs/2303.18027

27. https://github.com/sociocom/JMED-LLM

28.https://speakerdeck.com/fta98/ri-ben-yu-yi-liao-llmping-jia-bentimakunogou-zhu-toxing-neng-fen-xi

29. https://tech.preferred.jp/ja/blog/llama3-preferred-medswallow-70b/