Embedding タスクでの継続事前学習によるドメイン特化の検討

藤本裕之 [1]　高田直幸 [1]　今田翔平 [2]　青木秀行 [2]
[1] セコム株式会社　IS 研究所
[2] セコム株式会社　技術開発本部
{hiroyu-fujimoto, n-takada, sho-konta, hid-aoki}@secom.co.jp

概要

汎用的な Embedding モデルでは未知のドメイン知識の埋め込みが困難である。本研究では、事前学習済みモデルに継続事前学習で新規のドメイン知識を付加することで特定ドメインに特化させることは可能なのかを検証した。Wikipedia の新規記事を疑似ドメインデータを作成し、ベンチマークを提案した。その結果、継続事前学習によってドメイン知識に関する能力が向上することを確認した。今後、より Embedding 能力が高いモデルでの効果、異なる性質のドメイン知識について検討したい。

1　はじめに

学習されていない専門領域の知識（ドメイン知識）に基づいた回答を LLM にさせたいというニーズがある。一般に Prompting[1] や RAG[2] で入力プロンプトに情報付加することで回答に反映させる方法が主流である。一方でプロンプトにドメイン知識を記述しきれない、回答の質は検索性能に依存するという課題がある [3]。さらに Embedding モデルは知らないドメイン知識を説明なしにはうまく埋め込めないという課題もある [4]。

そこで事前学習後にドメイン知識を含んだデータで学習しドメイン知識をモデルに付加する継続事前学習（中間学習）[5, 6, 7] や事後学習 [8, 9, 10] というアプローチがある。継続事前学習ではドメイン知識を含んだコーパスがあればよい。また LLM では継続事前学習によりドメイン知識による能力を獲得した例が報告されている [11, 12, 13]。一方で事後学習では事後タスク用のデータが必要となり、一般にドメイン知識を含んだコーパスから事後タスク用のデータを作成する必要がある [14, 15]。

そこで本研究では、Embedding モデルにおいても、ドメイン知識に基づいた継続事前学習のみによって特定ドメインに特化することが可能かを検証する。

2　関連研究

ドメインに特化した知識の学習には継続事前学習が有効であることが知られている [5, 16, 17]。自然言語生成 (NLG) タスクでは、英語モデルを日本語ドメインに特化させる試み [16] や、日本語モデルを金融ドメインに特化させる試み [17] などの報告がある。RoBERTa で様々なドメイン知識による継続事前学習を実施することで事後タスク (Classification) でドメイン特化ができることが確認されている [5]。

MTEB[18] や JMTEB[19] で上位の Embedding モデルは Decoder-only アーキテクチャを採用しており [8, 9, 10]、Embedding モデルのベースモデルとして NLG タスク用の事前学習済みモデルを利用することが主流になりつつある。また、NLG タスクにおいては継続事前学習で獲得した能力を事後学習で学習することなく事後タスクで発揮した例も数多く報告されている [11, 12, 13]。

一方で事後学習で対照学習等を使って Embedding タスクを行っている先行研究はあるが、継続事前学習と事後学習で同一ドメインデータを使用する [14]、あるいは事前学習の段階から同一ドメインデータを使用するという条件であり [15]、継続事前学習のみによるドメイン特化は検証されていない。

Embedding タスクのベンチマークには様々なサブタスクが存在する [19]。Retrieval は query と query に対応した positive 文書を用意し、query をつかって文書集合から positive 文書を検索する性能を評価する。このとき文書集合は数十万から数百万という規模とする場合が多い [19]。また Reranking は top-N 検索結果を想定した positive 文書と negative 文書の文書集合をつかって、query と positive 文書の類似度が高く、かつ query と negative 文書の類似度は低くなるかを評価する。このとき N は数十から百程度の規模とする場合が多い [19]。

Embedding タスクの評価データは Wikipedia[20] が

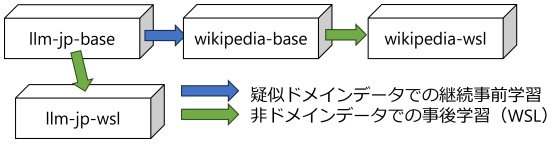

図 1　本研究の学習パイプライン

ベースとなる場合が多い。作成方法としては、記事タイトルと記事本文の関係性をそのまま利用 [20, 21]、検索エンジン [22]、人手でラベルを付与 [22, 23]、データ合成を活用するもの [24, 25] がある。

3　提案手法

ベースモデルに対してドメイン知識を含むデータによる継続事前学習を行い、その後、ドメイン知識を含まない事後タスク用データで事後学習を実施する。このときベースモデルに事後学習を実施したモデルも比較のために作成する。

本研究では、疑似ドメイン知識データの作成方法、特定ドメインデータでの Embedding タスクにおけるベンチマーク WSEBench(Wiki-Style Embedding Benchmark) を提案し、上記 2 モデルを作成、比較することで継続事前学習の効果を評価する。

3.1　疑似ドメイン知識データ

使用するベースモデルの事前学習に使用されていない、かつ汎用ベンチマークにも使用されていないデータをドメインデータとして採用する必要がある。本研究では、Wikipedia[20] の 2025/10/1 のダンプファイルから 2024/1/5 以降に新規に作成された記事のみを採用することで疑似的にベースモデルや汎用ベンチマークに含まれていないドメインデータを取得した。記事は計 78894 件となった。

3.2　WSEBench

本研究では、Wikipedia[20] などの記事タイトルと記事本文からなる title-text コーパスを対象とする。

gpt-oss-20b[26] で title と text を基に title が回答となるような質問を記事ごとに 5 件ずつ生成した。これを question とする。question 生成対象の記事は記事本文が記事タイトルより文字数が多く、100 文字以上の記事とした。また question に title がそのまま入っているもの、「タイトルは何ですか？」など question 単体で回答を導けないもの、他のレコードの question と重複しているものは機械的に除外した。question は LLM による合成によって記事本文とは異なる表現が用いられることを期待した。

表 1　WSEBench の構成

タイプ	query(#)	document(#)
title-text	title(65K)	text(79K)
question-text	question(65K)	text(79K)
question-title	question(65K)	title(65K)

本ベンチマークは query-document の形式でいくつかのタイプから構成する。これを表 1 に示す。title-text の構成は Wikipedia[20] や llm-japanese-dataset[21]、question-text の構成は Mr.TyDi[22] や MLDR[25] を参考とした。なお、question-text と question-title では、5 件の question を機械的に除外したものから 1 件の question を選び採用した。なお、title は question が存在する記事のみを使用し、text は全記事を対象とした。

この評価データセットごとに、Retrieval と Reranking のサブタスクについて評価する。Reranking では計 N 個になるように、文書集合のうち positive 文書を除いた文書 (非 positive 文書) から Negative Mining[27] を実施した。本研究では N=50、Negative Mining はランダムサンプリングとした。

4　実験

4.1　モデル

Decoder-only アーキテクチャである LLM-jp[28] の llm-jp-3-150M[1] をベースモデルとする。事前学習データが公開されており、本研究での実験に適していたためである。pooling は last token とした [9, 10]。

図 1 に示す通り、ベースモデルに対して継続事前学習を行ったモデルを wikipedia-base とし、それに対して、事後学習を行ったモデルを wikipedia-wsl とする。また、継続事前学習を行っていないモデルを llm-jp-base とし、それに対して、wikipedia-wsl と全く同じ事後学習を行ったモデルを llm-jp-wsl とする。

4.2　学習設定

継続事前学習のデータは 3.1 節で述べた疑似ドメインデータを使用し、記事情報の title と text を結合して作成した。

Embedding タスクの事後学習は、弱教師あり学習 (WSL) と教師あり学習 (SFT) の二段階で行われる場合が多いが [24]、本研究は WSL のみを実施した。WSL のデータは [10][24] を参考に公開されている計

1）　https://huggingface.co/llm-jp/llm-jp-3-150m

表 2　JMTEB(v1.4.0) の結果

Model	#Params	Avg.	Retrieval	STS	Classification	Reranking	Clustering
llm-jp-wsl	152M	0.615	0.521	0.785	0.724	0.721	0.488
wikipedia-wsl	152M	0.602	0.504	0.762	0.712	0.725	0.468

表 3　WSEBench の結果

Model	title-text			question-text			question-title		
	Avg.	Retrieval	Reranking	Avg.	Retrieval	Reranking	Avg.	Retrieval	Reranking
llm-jp-wsl	0.919	0.818	1.000	0.636	0.301	0.904	0.488	0.214	0.707
wikipedia-wsl	0.961	0.912	1.000	0.686	0.366	0.941	0.493	0.223	0.710

1735 万ペアのデータを採用した。詳細は付録 A に示す。また使用した事前学習済みモデルの事前学習データ、WSL で使用している Wikipedia 由来のデータは最新でも 2024/01/01 のダンプファイルであることを確認した。

その他学習設定の詳細は付録 B に示す。

4.3　評価

JMTEB[19] で汎用的な Embedding 能力、3.2 節で述べた WSEBench によってドメイン特化能力を測る。WSEBench では学習データから評価データを作成している。過学習の恐れがあるが、これはドメイン知識に特化したことを評価する上で避けらないため、JMTEB によって汎用的な能力を測ることで過学習の度合いを確認する。なお、JMTEB の一部データセットを除外して評価した[2)]。

WSEBench の評価指標は nDCG@k[29] を採用した。Retrieval は $k = \{1, 5, 10, 50, 100\}$、Reranking では $k = \{1, 3, 5, 10\}$ として、サブタスクごとのスコアはそれら全スコアの平均、全体の平均は両サブタスクの全スコア平均とした。なお Retrieval に関しては、nDCG@k は各クエリについて上位 100 件の文書に基づいて算出し、上位 100 件に positive 文書が含まれない場合は 0 として扱った。

4.4　結果

JMTEB の結果を表 2、WSEBench の結果を表 3 に示す。WSEBench の全タイプのベンチマークにおいて、llm-jp-wsl に比べて wikipedia-wsl が高いスコアとなった。JMTEB では、llm-jp-wsl に比べて wikipedia-wsl が低いスコアとなった。

2)　ライセンスが不明な Japanese Sentiment Classification、Academic Purpose Only の AmazonReviewClassification を除外。

5　考察

5.1　継続事前学習の有効性

WSEBench で wikipedia-wsl がより高いスコアであるため、継続事前学習の効果を確認した。ただし、question-title では効果は小さかった。

wikipedia-wsl の JMTEB スコアは、llm-jp-wsl に比べて総合スコアでは 1.3 ポイントの低下となった。付録 C のデータセットごとの結果によると、Retrieval と Reranking において、Wikipedia ベースのデータセットで wikipedia-wsl のスコアが高い傾向を確認した。継続事前学習によって、Wikipedia のドメインにより特化した可能性がある。このドメイン特化によって、Wikipedia ベースでないデータセットのスコアが低下し、総合スコアも低下したと考えられる。

5.2　Embedding ベクトルの類似度分布

llm-jp-wsl モデルと wikipedia-wsl モデルが、WSEBench のベンチマークタイプ毎に示した類似度の平均と標準偏差の値を、positive 文書、非 positive 文書に分けてそれぞれ表 4、表 5 に示す。なお、非 positive 文書は文書集合のうち positive 文書を除いた文書である。

llm-jp-wsl モデルと wikipedia-wsl モデルが示した類似度分布の差異について t 検定を行ったところ、いずれの分布においても有意な差が認められた ($p < 0.05$)。ただし、この結果はサンプル数の多さに起因する可能性がある。そこで、実質的な差の大きさを評価するため、効果量もあわせて算出した。効果量は Cohen's d[30] を用いた。

表 4, 5 に示した通り、positive 文書の title-text、question-text に関しては効果量が非常に小さく、実質的に差があるのは positive 文書の question-title と

表 4　positive 文書での類似度の平均と標準偏差

	title-text	question-text	question-title
llm-jp-wsl	0.714± 0.080	0.543± 0.140	**0.534± 0.143**
wikipedia-wsl	0.718± 0.076	0.534± 0.130	0.494±0.152
Cohen's d	-0.063	0.065	0.267

表 5　非 positive 文書での類似度の平均と標準偏差

	title-text	question-text	question-title
llm-jp-wsl	0.086 ± 0.108	0.110 ± 0.115	0.209 ± 0.119
wikipedia-wsl	**0.066 ± 0.105**	**0.076 ± 0.108**	**0.179 ± 0.107**
Cohen's d	0.193	0.306	0.266

非 positive 文書の全タイプのみと考えられる。

positive 文書の question-title では、llm-jp-wsl モデルに比べて wikipedia-wsl モデルの類似度が低く、非 positive 文書においても、全タイプで llm-jp-wsl モデルに比べて wikipedia-wsl モデルの類似度が低い結果となった。

継続事前学習によってドメイン知識間の共起を学習することで、近い意味のドメイン知識 (positive 文書) が高い類似度に、遠い意味のドメイン知識 (非 positive 文書) が低い類似度となる Embedding 空間が形成されることを期待したが、結果はいずれの類似度も下がる傾向であった。ただ、非 positive 文書に対してより低い類似度を示す傾向が見られ、これによって WSEBench のスコア向上につながった可能性がある。

5.3　ベンチマークの性質

WSEBench で query のトークン集合 Q が document のトークン集合 D にどの程度含まれているかを次の Overlap 率で算出し、表 6 に示した。

$$\text{Overlap}(Q, D) = \frac{|Q \cap D|}{|Q|}$$

title-text や question-text では Overlap 率が高い。text に title そのものや question に含まれる表現が多く含まれていることが原因と考える。つまり、text にはドメイン知識を説明するコンテキストが多く含まれていると考えられる。一方で question-title はお互いに共有している表現が少ないため Overlap 率が低い。

WSEBench の結果をみると、ベンチマークタイプによって結果が大きく異なる。query-document 間で共通するコンテキストの量によって必要な能力が異なる可能性がある。

表 6　WSEBench での Overlap 率

タイプ	Overlap 率 [%]
title-text	90.1
question-text	71.3
question-title	9.2

5.4　疑似ドメインデータの性質

本研究で提案した疑似ドメインデータには既存の学習データに含まれる可能性のあるドメイン知識、新規で生まれたドメイン知識が含まれている。例えば、疑似ドメインデータに含まれていた「ソロモン朝」はエチオピア帝国の過去の実在した王朝であるため、既存の学習データに含まれている可能性がある。また事件や映画など新規で生まれた概念もあることを確認している。

一方で疑似ドメインデータにはあまり含まれていないが、同じ単語でも一般と意味の異なる概念となるようなドメイン知識もある。例えば、警備業界ではセットという単語は警備システムの開始という意味となることがある。このようなドメイン知識は既存知識と重複しているため、ドメイン特化の効果がより高いことが予想される。

6　おわりに

本研究では、ドメイン知識に基づいた継続事前学習の Embedding タスクの効果を検証した。また、事前学習、事後学習に含まれない Wikipedia の新規記事のみを対象とした疑似ドメインデータを作成し、疑似ドメインデータにおける Embedding 能力を測るベンチマークを提案した。その結果、Embedding タスクにおいては継続事前学習のみによるドメイン知識の学習でドメイン特化できることを確認した。ただしベンチマークのタイプによっては効果は小さかった。

一方で JMTEB[19] の上位の Embedding モデルに比べると、本研究のモデルのスコアが低いことを確認しており、Embedding 能力自体が不足していると考えられる。より Embedding 能力が高いモデルにおける継続事前学習の効果については、今後検証する必要がある。

本研究では、Wikipedia から疑似ドメインデータを作成したが、同じ単語でも一般と意味の異なる概念となるようなドメイン知識も存在することを確認している。今後、こういった異なる性質をもったドメイン知識での継続事前学習の効果を検証したい。

参考文献

[1] Tom B. Brown, et al. Language Models are Few-Shot Learners, July 2020. arXiv:2005.14165 [cs].

[2] Patrick Lewis, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, April 2021. arXiv:2005.11401 [cs].

[3] Chaitanya Sharma. Retrieval-Augmented Generation: A Comprehensive Survey of Architectures, Enhancements, and Robustness Frontiers, May 2025. arXiv:2506.00054 [cs] version: 1.

[4] Kelong Mao, et al. RAG-Studio: Towards In-Domain Adaptation of Retrieval Augmented Generation Through Self-Alignment. In **Findings of the Association for Computational Linguistics: EMNLP 2024**, pp. 725–735, Miami, Florida, USA, January 2024. Association for Computational Linguistics.

[5] Suchin Gururangan, et al. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks, May 2020. arXiv:2004.10964 [cs].

[6] Tongtong Wu, et al. Continual Learning for Large Language Models: A Survey, February 2024. arXiv:2402.01364 [cs].

[7] Haizhou Shi, et al. Continual Learning of Large Language Models: A Comprehensive Survey, November 2024. arXiv:2404.16789 [cs].

[8] Jinhyuk Lee, et al. Gemini Embedding: Generalizable Embeddings from Gemini, March 2025. arXiv:2503.07891 [cs].

[9] Yanzhao Zhang, et al. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models, June 2025. arXiv:2506.05176 [cs].

[10] SB Intuitions. Sarashina-Embedding-v2-1B: 日本語に特化した指示を付与できるテキスト埋め込みモデル. https://www.sbintuitions.co.jp/blog/entry/2025/08/20/160139, 2025. SB Intuitions TECH BLOG. Accessed: 2025-12-17.

[11] OpenAI, et al. GPT-4 Technical Report, March 2024. arXiv:2303.08774 [cs].

[12] Tiffany H. Kung, et al. Performance of ChatGPT on USMLE: Potential for AI-assisted medical education using large language models. **PLOS Digital Health**, Vol. 2, No. 2, p. e0000198, February 2023.

[13] Ross Taylor, et al. Galactica: A Large Language Model for Science, November 2022. arXiv:2211.09085 [cs].

[14] Jinhyuk Lee, et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. **Bioinformatics**, Vol. 36, No. 4, pp. 1234–1240, February 2020. arXiv:1901.08746 [cs].

[15] Xuan Xu, et al. FinBERT2: A Specialized Bidirectional Encoder for Bridging the Gap in Finance-Specific Deployment of Large Language Models, May 2025. arXiv:2506.06335 [cs] version: 1.

[16] 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 **Proceedings of the First Conference on Language Modeling**, COLM, p. (to appear), University of Pennsylvania, USA, October 2024.

[17] Preferred Networks. 継続事前学習による金融ドメイン特化 LLM の構築の検証. https://tech.preferred.jp/ja/blog/qfin-llm-continual-pretraining/, 2024. Blog. Accessed: 2025-12-03.

[18] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. MTEB: Massive Text Embedding Benchmark, March 2023. arXiv:2210.07316 [cs].

[19] Shengzhe Li, Masaya Ohagi, and Ryokan Ri. JMTEB: Japanese Massive Text Embedding Benchmark. https://huggingface.co/datasets/sbintuitions/JMTEB, 2024.

[20] Wikimedia Foundation. Wikimedia Downloads. https://dumps.wikimedia.org. Accessed: 2026-1-5.

[21] Masanori Hirano, Masahiro Suzuki, and Hiroki Sakaji. llm-japanese-dataset v0: Construction of Japanese Chat Dataset for Large Language Models and its Methodology, 2023.

[22] Xinyu Zhang, Xueguang Ma, Peng Shi, and Jimmy Lin. Mr. TyDi: A multi-lingual benchmark for dense retrieval. **arXiv:2108.08787**, 2021.

[23] Yusuke Oda and Baobab. wikipedia-human-retrieval-ja. https://huggingface.co/datasets/baobab-trees/wikipedia-human-retrieval-ja, 2024. Accessed: 2025-12-17.

[24] Hayato Tsukagoshi and Ryohei Sasano. Ruri: Japanese General Text Embeddings, September 2024. arXiv:2409.07737 [cs].

[25] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. BGE m3-embedding: Multilingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2024.

[26] OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025.

[27] Gabriel de Souza P. Moreira, Radek Osmulski, Mengyao Xu, Ronay Ak, Benedikt Schifferer, and Even Oldridge. NV-Retriever: Improving text embedding models with effective hard-negative mining, February 2025. arXiv:2407.15831 [cs].

[28] LLM-jp, et al. LLM-jp: A cross-organizational project for the research and development of fully open japanese llms, 2024.

[29] Järvelin Kalervo and Kekäläinen Jaana. Cumulated gain-based evaluation of ir techniques. **ACM Transactions on Information Systems**, Vol. 20, No. 4, pp. 422–446, 10 2002.

[30] Peter A. Lachenbruch and Cohen Jacob. Statistical power analysis for the behavioral sciences (2nd ed.). **Journal of the American Statistical Association**, Vol. 84, No. 408, p. 1096, 12 1989.

表 7　WSL の学習データセット

データセット名	レコード数
hpprc/mqa-ja	12944450
cl-nagoya/auto-wiki-qa	2377503
wikipedia/20231101.ja	1389467
sentence-transformers/NQ-retrieval	307373
JSNLI 1.1	176142
sbintuitions/JSQuAD	62859
SNOW(T15+T23)	50000
SkelterLabsInc/JaQuAD	31748
apple/mkqa	6758
計	17346300

表 8　学習パラメータ

Parameters	継続事前学習	事後学習 (WSL)
learning rate	3e-4	5e-5
max length	512	512
lr scheduler	cosine	cosine
warmup ratio	0.1	0.1
batch size	50	4096
batch sampler	-	task-homogeneous[24]
epochs	2	1
loss function	Cross Entropy	MNRL(w/ GradCache)
optimizer	AdamW	AdamW
weight decay	0.1	0.1
beta1	0.9	0.9
beta2	0.95	0.95
epsilon	1e-8	1e-8

A　WSL の学習データ

表 7 に使用した学習データセットを示す。なお、JSNLI では negative 文書がレコードごとに計 30 件になるように Hard Negative Mining した。Hard Negative Mining は positive 文書との類似度スコアに 0.95 をかけたスコアを上限とした top-k mining を採用した [27]。モデルは ruri-v3-310m[24] を使用した。

B　学習設定

表 8 に本研究で使用した学習パラメータを示す。llm-jp の事前学習で使用された学習パラメータを参考に設定した。継続事前学習の learning rate は、事前学習の 3e-4 と同じ値を設定した[3)]。予備実験で 3e-5 と比較した結果、3e-4 の方が WSEBench においては良い性能だったためである。

なお、継続事前学習に用いたデータの総トークン数は 3677 万トークンである。

表 9　JMTEB スコア（Retrieval, Reranking）

task	model dataset_name	llm-jp-wsl (nDCG@10)	wikipedia-wsl (nDCG@10)
Retrieval	jacwir_retrieval	0.653	0.648
	jagovfaqs_22k	0.678	0.617
	jaqket*	0.313	0.250
	mintaka_retrieval	0.359	0.316
	miracl_retrieval*	0.285	**0.347**
	mldr_retrieval*	0.247	0.243
	mrtydi*	0.177	**0.195**
	nlp_journal_abs_article	0.760	**0.766**
	nlp_journal_abs_intro	0.781	0.772
	nlp_journal_title_abs	0.860	0.837
	nlp_journal_title_intro	0.621	0.555
Reranking	esci	0.922	0.919
	jacwir_reranking	0.741	**0.749**
	jqara*	0.350	0.343
	miracl_reranking*	0.723	**0.728**
	mldr_reranking*	0.872	**0.884**

C　JMTEB スコアの詳細

表 9 に Retrieval と Reranking のデータセットごとの結果を示す。Wikipedia ベースのデータセットに* を付与した [19]。Wikipedia ベースのデータセットで wikipedia-wsl のスコアが高い傾向がある。

3）https://github.com/llm-jp/scripts/blob/main/pretrain/scripts/v3-152m-sakura/train.sh#L44