Ruri: 日本語に特化した汎用テキスト埋め込みモデル

塚越 駿　　笹野 遼平
名古屋大学大学院情報学研究科
tsukagoshi.hayato.r2@s.mail.nagoya-u.ac.jp
sasano@i.nagoya-u.ac.jp

概要

近年，英語や多言語の汎用的なテキスト埋め込みモデルの開発が盛んに行われている．しかし，日本語でのモデル開発の取り組みは限定的であり，その理由としてはデータセットの不足やモデル開発のための知見が少ないことが挙げられる．本稿では，日本語汎用テキスト埋め込みモデル Ruri を開発し，その過程について述べる．具体的には，訓練データの不足を補うための大規模言語モデルによる合成データセット構築，対照事前学習によるベースモデルの訓練，そして高品質データを用いた微調整について説明する．構築したテキスト埋め込みモデル Ruri は，日本語テキスト埋め込みのベンチマークにおいて既存のモデルを上回る性能を達成した．

1　はじめに

テキスト埋め込みは，検索拡張生成や類似文書検索などのタスクで広く利用されており [1, 2]，特に近年は多様なデータセットで学習した汎用的なテキスト埋め込みモデルの開発が盛んに行われている [3, 4, 5, 6, 7]．しかし，これらの取り組みは主に英語モデルや多言語モデルの構築を目標としており，日本語の語彙・データは相対的に小さな割合にとどまっている．大規模な日本語データセットを用いて埋め込みモデルを学習すれば，これらの多言語モデルより日本語を対象としたタスクにおいて高性能なモデルの実現が期待できるが，テキスト埋め込みの学習に利用できる日本語資源は限定的である．

本稿では，テキスト埋め込みモデルの学習に利用できるような大規模日本語データセットを構築し，日本語に特化したテキスト埋め込みモデルを開発する．具体的には，既存データセットの収集・整理と，大規模言語モデル (LLM) による合成データセットの構築を行い，構築されたデータセットに基づき，対照学習による埋め込みモデルの事前学習と高品質な人手データによる微調整を行った．構築されたテキスト埋め込みモデル Ruri は，日本語テキスト埋め込みのベンチマークで既存のモデルを大きく上回る性能を達成した．本研究で構築したモデルおよびデータセットは公開している[1)]．

2　対照事前学習

近年のテキスト埋め込みモデル開発では，2 段階の学習アプローチを取ることが一般的である [3, 5, 6]．具体的には，まず大規模な弱教師データセットで対照学習損失 [8] に基づく事前学習 (対照事前学習) を行い，その後，人手で作成された高品質なデータを用いてモデルの微調整を行う．本稿ではこの流れにしたがい，対照事前学習に基づいて日本語テキスト埋め込みモデルの強力なベースモデルを構築し，微調整によって性能を高めることを目指す．しかし，英語や多言語モデルと異なり，日本語の汎用テキスト埋め込みモデル構築には，訓練データ量が不足している．そこで，本稿では既存の資源を収集するとともに，LLM を用いてデータ合成を行うことでその不足を補う．本研究では，対照事前学習用のデータセットを v1 と v2 の 2 通り構築する．それぞれのデータセットに含まれるデータの内訳を表 1 と表 2 に示す．既存のデータセットについては付録 A に記載する．

2.1　合成データセット

近年，テキスト埋め込みの学習に合成データを活用する取り組みが盛んであり [9, 10, 11, 12]，特に日本語のような公開データセットの少ない言語では有用と考えられる．本稿でも，LLM を用いてテキスト埋め込みモデル学習用のデータを合成する．

本稿では，検索・QA データセットや自然言語推論 (NLI) データセットなど複数種類のデータを

1) https://huggingface.co/collections/cl-nagoya/ruri-japanese-general-text-embeddings-66cf1f3ee0c8028b89d85b5e

表 1　対照事前学習に用いる v1 データセット

カテゴリ	詳細	合成	ペア数
Wikipedia	タイトル–1 段落		19,361,464
Wikipedia	タイトル–3 段落		10,010,462
Wikipedia 要約	タイトル–要約		7,889,486
Wiktionary	タイトル–本文		697,405
WikiBooks	タイトル–1 段落		314,207
MQA	質問と回答		25,165,824
CC-News	タイトル–本文		6,248,336
CC-News-topic	同記事中の 2 文		2,795,632
JRC	タイトル–本文		131,072
言い換え・NLI	Wiki Atomic Edits		3,679,939
言い換え・NLI	AutoWikiNLI	✓	203,147
言い換え・NLI	JSNLI	✓	180,146
Wikipedia 検索	本文由来 QA	✓	12,058,624
合計			88,735,744

合成した．検索・QA データセットの構築については，まず，試験的に小規模な検索・QA データセットを日本語 Wikipedia をもとに構築した[2]．この合成には，Mixtral-8x7B[3] をもとに，日本語大規模コーパスの継続学習を施したモデル [13, 14] である Swallow-MX[4] と，Nemotron-4 340B [15] を用いる．結果として 2,377,503 事例からなる日本語検索・QA データセットを構築した[5]．このデータセットを用いて対照事前学習用データセット v1 とし，予備実験を行ったところ，合成データセットを用いた結果が有望だったため，さらに大規模な合成を行った．具体的には，LLM-jp [16] の 13B モデル[6]，Phi-3.5-mini[7]，Qwen2.5 [17] の 14B, 32B モデル，CALM-3 22B [18] を加えて合成を行った．

合成データセット生成は日本語 Wikipedia の本文を LLM に与え，そこから質問を生成させるという流れで行うが，合成データの量が増えるにつれ生成元文章の方が分量が小さくなり，同じデータから繰り返し合成データを作成することに対するデータセットの多様性に関する懸念がある．そこで，LLM を用いて日本語 Wikipedia 中のテキストを言い換え，その言い換えをもとに質問を生成させることで，データセットの表層的な多様性を増大させることを試みる．結果として，日本語 Wikipedia 中の文から生成された 23,394,737 事例と，日本語 Wikipedia 中の文の言い換えから生成された 42,247,260 事例を合わせた 65,641,997 事例からなる合成データセットを構築した．さらに，データセットの多様性を増大させるため，日本語高品質データセットを対象に言い換えや要約を LLM を用いて生成し，事前学習用のデータに加えた．これらのデータは対照事前学習用データセット v2 に含め，公開している[8]．

表 2　対照事前学習に用いる v2 データセット

カテゴリ	詳細	合成	ペア数
Wikipedia	タイトル–1 段落		9,718,042
Wikipedia	タイトル–3 段落		5,017,846
Wikipedia 要約	タイトル–要約		1,063,924
MQA	質問と回答		12,576,291
CC-News	タイトル–本文		1,659,944
CC-News-topic	同記事中の 2 文		822,938
JRC	タイトル–本文		64,738
JRC	タイトル–言い換え	✓	1,219,273
JRC	言い換え–本文	✓	1,219,273
Kaken	タイトル–本文		5,303,408
Kaken	タイトル–要約	✓	14,937,345
Kaken	要約–本文	✓	14,937,345
Warp-HTML	タイトル–本文		1,058,092
Warp-HTML	タイトル–要約	✓	10,955,338
Warp-HTML	要約–本文	✓	10,955,338
読み・異表記	小説タイトル		16,319
読み・異表記	日本語 WordNet		16,504
読み・異表記	sudachi 辞書		62,797
読み・異表記	異表記		2,496,799
言い換え・NLI	SNOW		62,758
言い換え・NLI	JWTD		696,188
言い換え・NLI	Wiki Atomic Edits		3,601,689
言い換え・NLI	AutoWikiNLI	✓	198,895
言い換え・NLI	JSNLI	✓	176,309
Wikipedia 検索	本文由来 QA	✓	23,394,737
Wikipedia 検索	言い換え由来 QA	✓	42,247,260
合計			199,657,488

2.2　学習設定

本研究では，性能と推論コストのトレードオフをモデルの利用者が選択できるよう，small, base, large の 3 種類のモデルを構築する．small は LINE 社 (現 LINE ヤフー社) の DistilBERT[9] を，base と large は東北大学の BERT[10),11)] をベースモデルとして用い，それぞれのモデルを微調整する形で対照事前学習を行う．学習時は，先行研究 [3, 4, 5, 6] にならい接頭辞を各系列に付加した．学習には Li ら [5] が用いた改良版対照学習損失を利用する．v1 データセットでの対照事前学習では BM25 を用いて取得したハード負例を各事例ごとに 1 事例付加した．v2 データセットでの対照事前学習ではハード負例は用いなかった．small モデルは NVIDIA A6000 を 4 つ，base, large モデルは NVIDIA A100 (80GB) を 4 つ用いて訓

2) https://huggingface.co/datasets/cl-nagoya/auto-wiki-qa
3) https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1
4) https://huggingface.co/tokyotech-llm/Swallow-MX-8x7b-NVE-v0.1
5) https://huggingface.co/datasets/cl-nagoya/auto-wiki-qa
6) https://huggingface.co/llm-jp/llm-jp-3-13b
7) https://huggingface.co/microsoft/Phi-3.5-mini-instruct
8) https://huggingface.co/datasets/cl-nagoya/ruri-dataset-v2-pt
9) https://huggingface.co/line-corporation/line-distilbert-base-japanese
10) https://huggingface.co/tohoku-nlp/bert-base-japanese-v3
11) https://huggingface.co/tohoku-nlp/bert-large-japanese-v2

練した．その他の設定は付録 B に記載する．

2.3 モデルマージ

LLM を中心に，モデルのパラメータに対する算術操作がモデルの性能を向上させることが知られている [19, 20]．後述する実験の結果，v1 データセットを用いて対照事前学習されたモデルと，v2 データセットを用いて対照事前学習されたモデルを単一のモデルに統合することで全体的な性能向上が見られた．本稿では，v1 データセットのみを用いて構築されたモデルを v1 モデル，v1 データセットと v2 データセットを用いて構築されたそれぞれのモデルをマージしたモデルを v2 モデルと呼称する．

3 高品質データによる微調整

先行研究 [3, 5, 6] にならい，高品質なデータを用いて対照事前学習されたモデルの微調整を行う．

3.1 データセットの構築

対照事前学習したモデルを微調整するため，検索・QA データセットと自然言語推論 (NLI) データセットの二種類のデータセットを収集する．検索・QA データセットについては，検索クエリと文書の関連度をスコア付けするリランカ[12] を用いて，ペアになっている検索クエリと文書の関連どのスコアが一定以上になるようフィルタリングする．高品質データについても，v1 と v2 の二つのデータセットを構築する．実際に収集したデータセットは，日本語の QA データ 2 種[13),14)]，日本語 QA データセットの JQaRA [21]，多言語検索データセットの Mr. TyDi [22] と MIRACL [23] の日本語サブセット，LLM により日本語に翻訳された SNLI [24] と MNLI [25] である NU-SNLI[15] と NU-MNLI[16]，日本語自然言語推論データセットの JaNLI [26] である．構築した高品質データセットの内訳を表 3 に示す．検索・QA データセットのうち高品質な JQaRA，Mr. TyDi，MIRACL についてはアップサンプリングを行っているほか，データセットごとにデータ拡張の方法やフィルタリングの設定が異なるため，同じデータをもとにしていても v1 と v2 で事例数が異なる．検索・QA データセットについては，多言語 E5 [4] を用いて検索クエリに対するハード負例を収集し，偽の負例を取り除くためリランカによるスコアが一定値以上のハード負例をフィルタリングし，各検索クエリ–正解文書ごとに 15 のハード負例を付与した．

12) https://huggingface.co/cl-nagoya/ruri-reranker-large
13) https://huggingface.co/datasets/hpprc/quiz-no-mori
14) https://huggingface.co/datasets/hpprc/quiz-works
15) https://huggingface.co/datasets/cl-nagoya/nu-snli
16) https://huggingface.co/datasets/cl-nagoya/nu-mnli

表 3 高品質データでの微調整に用いるデータセット

データセット名	v1	v2
クイズの杜	31,232	24,576
Quiz Works	26,624	16,384
JQaRA [21]	13,824	24,576
MIRACL [23]	12,800	32,768
Mr. TyDi [22]	7,168	24,576
NU-SNLI	109,568	114,688
NU-MNLI	77,824	81,920
JaNLI [26]	13,824	8,192
合計	292,864	327,680

3.2 学習設定

v1 データセットを用いて対照事前学習を行ったモデル Ruri-PT と，v2 データセットを用いて対照事前学習を行ったモデルと Ruri-PT をマージした Ruri-PT v2 をそれぞれ高品質データで微調整する．v1 モデルについては，検索・QA データのミニバッチに対してリランカからの知識蒸留 [27, 28] を適用する．v2 モデルについては，知識蒸留は行わず，単純な対照学習を行った．学習時の最大系列長は 512 に設定し，その他のハイパーパラメータや設定については付録 C に記載する．

4 JMTEB による評価

日本語のテキスト埋め込みベンチマークである JMTEB [29] を用いて，構築した日本語汎用テキスト埋め込みモデル Ruri を評価した．

4.1 評価設定

JMTEB は，英語中心のベンチマークである MTEB [30] の日本語版であり，分類・検索など 16 の評価データセットから構成される．評価には公式実装を用いる．ただし，Ruri や多言語 E5 など接頭辞を用いるテキスト埋め込みモデルは，タスクごとに適切な接頭辞を付加して評価する．

4.2 結果

Ruri と既存のテキスト埋め込みモデルを評価した結果を表 4 に示す．全体として，Ruri は多言語 E5 などの多言語モデルや，既存の日本語の埋め込みモデルを安定して上回った．Ruri と Ruri v2 の差を比較すると，特に small サイズでの性能向上が大きく，

表 4 JMTEB での評価結果．“Model” はモデル名を，“#Param” はモデルのパラメータ数を表し，“Avg.” は 16 タスクのマイクロ平均を表す．それ以外の列は JMTEB に含まれるタスクの種別ごとの平均性能を表す．

Model	#Param.	Retrieval	STS	Class.	Reranking	Clustering	Pair.	Avg.
cl-nagoya/sup-simcse-ja-base	111M	49.64	82.05	73.47	91.83	51.79	62.57	63.36
cl-nagoya/sup-simcse-ja-large	337M	37.62	83.18	73.73	91.48	50.56	62.51	58.88
cl-nagoya/unsup-simcse-ja-base	111M	40.23	78.72	73.07	91.16	44.77	62.44	58.39
cl-nagoya/unsup-simcse-ja-large	337M	40.53	80.56	74.66	90.95	48.41	62.49	59.58
pkshatech/GLuCoSE-base-ja	133M	59.02	78.71	76.82	91.90	49.78	66.39	67.29
pkshatech/GLuCoSE-base-ja-v2	133M	73.36	82.96	74.21	93.01	48.65	62.37	72.23
sentence-transformers/LaBSE	472M	40.12	76.56	72.66	91.63	44.88	62.33	58.01
intfloat/multilingual-e5-small	118M	67.27	80.07	67.62	93.03	46.91	62.19	67.71
intfloat/multilingual-e5-base	278M	68.21	79.84	69.30	92.85	48.26	62.26	68.61
intfloat/multilingual-e5-large	560M	70.98	79.70	72.89	92.96	51.24	62.15	70.90
OpenAI/text-embedding-ada-002	-	64.38	79.02	69.75	93.04	48.30	62.40	67.21
OpenAI/text-embedding-3-small	-	66.39	79.46	73.06	92.92	51.06	62.27	69.18
OpenAI/text-embedding-3-large	-	74.48	82.52	77.58	93.58	53.32	62.35	74.05
$\text{Ruri}_{\text{small}}$ (cl-nagoya/ruri-small)	68M	69.41	82.79	76.22	93.00	51.19	62.11	71.53
$\text{Ruri}_{\text{base}}$ (cl-nagoya/ruri-base)	111M	69.82	82.87	75.58	92.91	54.16	62.38	71.91
$\text{Ruri}_{\text{large}}$ (cl-nagoya/ruri-large)	337M	73.02	83.13	77.43	92.99	51.82	62.29	73.31
$\text{Ruri}_{\text{small}}$ v2 (cl-nagoya/ruri-small-v2)	68M	73.94	82.91	76.17	93.20	51.58	62.32	73.30
$\text{Ruri}_{\text{base}}$ v2 (cl-nagoya/ruri-base-v2)	111M	72.33	83.03	75.34	93.17	51.38	62.35	72.48
$\text{Ruri}_{\text{large}}$ v2 (cl-nagoya/ruri-large-v2)	337M	76.34	83.17	77.18	93.21	52.14	62.27	**74.55**

表 5 v1 モデル構築における実験条件ごとの性能

Model	Retrieval	Avg.
$\text{Ruri-PT}_{\text{large}}$	71.48	72.46
$\text{Ruri-PT}_{\text{large}}$ w/o 合成検索データ	68.08	71.11
$\text{Ruri-PT}_{\text{large}}$ w/o ハード負例	71.37	72.52

表 6 マージによるモデルの性能向上

Model	Retrieval	Avg.
$\text{Ruri-PT}_{\text{small}}$ w/ データセット v1	67.39	70.41
$\text{Ruri-PT}_{\text{small}}$ w/ データセット v2	68.16	70.43
$\text{Ruri-PT}_{\text{small}}$ v2	**70.58**	**71.43**
$\text{Ruri-PT}_{\text{base}}$ w/ データセット v1	**69.82**	**71.91**
$\text{Ruri-PT}_{\text{base}}$ w/ データセット v2	67.69	70.25
$\text{Ruri-PT}_{\text{base}}$ v2	69.58	71.39
$\text{Ruri-PT}_{\text{large}}$ w/ データセット v1	71.48	72.46
$\text{Ruri-PT}_{\text{large}}$ w/ データセット v2	69.58	71.50
$\text{Ruri-PT}_{\text{large}}$ v2	**73.40**	**73.32**

$\text{Ruri}_{\text{small}}$ v2 は $\text{Ruri}_{\text{large}}$ や多言語 E5 と同等以上の性能を示した．さらに，$\text{Ruri}_{\text{large}}$ v2 は全体で最も高い性能を示し，同規模のモデルとしては初めて，OpenAI 社の埋め込みモデルを上回る性能を達成した．

4.3 アブレーション実験

v1 データセットを用いた対照事前学習において，1) 合成検索データセットの有無，2) ハード負例の有無，を変えた場合の性能を表 5 に示す．結果から，合成検索データセットは検索タスクの性能を向上させ，有用であり，対照事前学習時のハード負例は性能に必ずしも寄与しないことがわかった．

次に，それぞれのデータセットごとに訓練されたモデルと，モデルマージを行ったモデルの性能を表 6 に示す．単独のデータセットで訓練されたモデルの性能を比較すると，v1 データセットを用いた場合の方が v2 データセットのみを用いた場合より性能が高い傾向にあることがわかる．v2 データセットは合成データが非常に多く含まれるデータセットになっており，データの多様性が低減したことが性能低下の理由として考えられる．ただし，単独のデータセットでは性能が下がってしまう場合にも，モデルマージを行うことで性能向上に寄与する場合がある点に注意が必要である．実際に，small と large モデルについては，マージモデルの性能が単独モデルの性能を 1 ポイント程度上回っている．

5 まとめ

本稿では，日本語に特化した汎用テキスト埋め込みモデルを構築するため，合成データセットを含む大規模データセットの構築と，様々な条件でのモデル開発を行った．その結果，合成データセットを用いることによる検索性能の向上や，異なる事前学習用データセットにより構築された二つのモデルを統合することによる性能向上が確認できた．最終的に構築されたテキスト埋め込みモデル Ruri は，JMTEB による評価の結果，同規模の既存日本語・多言語テキスト埋め込みモデルを上回る性能を達成した．

謝辞

本研究は JSPS 科研費 23KJ1134，24H00727 の助成を受けたものです．

参考文献

[1] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**, pp. 3982–3992, 2019.

[2] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 6894–6910, 2021.

[3] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text Embeddings by Weakly-Supervised Contrastive Pre-training. **arXiv:2212.03533**, 2022.

[4] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual e5 text embeddings: A technical report. 2024.

[5] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards General Text Embeddings with Multi-stage Contrastive Learning. **arXiv:2308.03281**, 2023.

[6] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-Pack: Packaged Resources To Advance General Chinese Embedding. In **The 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)**, 2024.

[7] Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, Maximilian Werk, Nan Wang, and Han Xiao. Jina Embeddings 2: 8192-Token General-Purpose Text Embeddings for Long Documents. **arXiv:2310.19923**, 2024.

[8] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. **arXiv:1807.03748**, 2019.

[9] Junlei Zhang, Zhenzhong Lan, and Junxian He. Contrastive Learning of Sentence Embeddings from Scratch. In **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023)**, pp. 3916–3932, December 2023.

[10] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving Text Embeddings with Large Language Models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 11897–11916, 2024.

[11] Soma Sato, Hayato Tsukagoshi, Ryohei Sasano, and Koichi Takeda. Improving Sentence Embeddings with Automatic Generation of Training Data Using Few-shot Examples. In Xiyan Fu and Eve Fleisig, editors, **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics Volume 4: Student Research Workshop (ACL SRW)**, pp. 519–530, 2024.

[12] Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R. Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, Yi Luan, Sai Meher Karthik Duddu, Gustavo Hernandez Abrego, Weiqiang Shi, Nithi Gupta, Aditya Kusupati, Prateek Jain, Siddhartha Reddy Jonnalagadda, Ming-Wei Chang, and Iftekhar Naim. Gecko: Versatile Text Embeddings Distilled from Large Language Models. **arXiv:2403.20327**, 2024.

[13] Naoaki Okazaki, Kakeru Hattori, Hirai Shota, Hiroki Iida, Masanari Ohi, Kazuki Fujii, Taishi Nakamura an Mengsay Loem, Rio Yokota, and Sakae Mizuki. Building a Large Japanese Web Corpus for Large Language Models. In **Proceedings of the First Conference on Language Modeling (COLM)**, COLM, 2024.

[14] 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)**, COLM, 2024.

[15] Nvidia. Nemotron-4 340B Technical Report. **arXiv:2406.11704**, 2024.

[16] LLM jp Team. LLM-jp: A Cross-organizational Project for the Research and Development of Fully Open Japanese LLMs. **arXiv:2407.03963**, 2024.

[17] Qwen Team. Qwen2.5 Technical Report. **arXiv:2412.15115**, 2025.

[18] Ryosuke Ishigami. cyberagent/calm3-22b-chat, 2024.

[19] 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 **Proceedings of the 39th International Conference on Machine Learning (ICML)**, Vol. 162 of **Proceedings of Machine Learning Research**, pp. 23965–23998, 17–23 Jul 2022.

[20] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In **The Eleventh International Conference on Learning Representations (ICLR)**, 2023.

[21] Yuichi Tateno. JQaRA: Japanese Question Answering with Retrieval Augmentation - 検索拡張 (RAG) 評価のための日本語 Q&A データセット, 2024.

[22] Xinyu Zhang, Xueguang Ma, Peng Shi, and Jimmy Lin. Mr. TyDi: A Multi-lingual Benchmark for Dense Retrieval. In **Proceedings of the 1st Workshop on Multilingual Representation Learning (MRL)**, pp. 127–137, 2021.

[23] Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. Making a MIRACL: Multilingual Information Retrieval Across a Continuum of Languages. **arXiv:2210.09984**, 2022.

[24] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In **Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 632–642, 2015.

[25] Adina Williams, Nikita Nangia, and Samuel Bowman. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)**, pp. 1112–1122, 2018.

[26] Hitomi Yanaka and Koji Mineshima. Assessing the Generalization Capacity of Pre-trained Language Models through Japanese Adversarial Natural Language Inference. In **Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP (BlackboxNLP)**, pp. 337–349, 2021.

[27] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. SimLM: Pre-training with Representation Bottleneck for Dense Passage Retrieval. In **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 2244–2258, 2023.

[28] Benjamin Clavié. JaColBERTv2.5: Optimising Multi-Vector Retrievers to Create State-of-the-Art Japanese Retrievers with Constrained Resources. **arXiv:2407.20750**, 2024.

[29] Shengzhe Li, Masaya Ohagi, and Ryokan Ri. JMTEB: Japanese Massive Text Embedding Benchmark. https://huggingface.co/datasets/sbintuitions/JMTEB%7D%7D, 2024. [Accessed 31-08-2024].

[30] Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. MTEB: Massive Text Embedding Benchmark. In **Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL)**, pp. 2014–2037, 2023.

[31] Hitoshi Isahara, Francis Bond, Kiyotaka Uchimoto, Masao Utiyama, and Kyoko Kanzaki. Development of the Japanese WordNet. In **Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC)**, 2008.

[32] Kazuma Takaoka, Sorami Hisamoto, Noriko Kawahara, Miho Sakamoto, Yoshitaka Uchida, and Yuji Matsumoto. Sudachi: a Japanese Tokenizer for Business. In **Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)**, 2018.

[33] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying Conditional Random Fields to Japanese Morphological Analysis. In **Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 230–237, 2004.

[34] 佐藤敏紀, 橋本泰一, 奥村学. 単語分かち書き辞書 mecab-ipadic-NEologd の実装と情報検索における効果的な使用方法の検討. 言語処理学会 第 23 回年次大会, 2017.

[35] Takumi Maruyama and Kazuhide Yamamoto. Simplified Corpus with Core Vocabulary. In **Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)**, 2018.

[36] Akihiro Katsuta and Kazuhide Yamamoto. Crowdsourced Corpus of Sentence Simplification with Core Vocabulary. In **Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)**, 2018.

[37] Yu Tanaka, Yugo Murawaki, Daisuke Kawahara, and Sadao Kurohashi. Building a Japanese Typo Dataset from Wikipedia's Revision History. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop (ACL SRW)**, pp. 230–236, 2020.

[38] Manaal Faruqui, Ellie Pavlick, Ian Tenney, and Dipanjan Das. WikiAtomicEdits: A Multilingual Corpus of Wikipedia Edits for Modeling Language and Discourse. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)**, pp. 305–315, 2018.

[39] 吉越, 卓見 and 河原, 大輔 and 黒橋, 禎夫. 機械翻訳を用いた自然言語推論データセットの多言語化. 第 244 回自然言語処理研究会（NL 研）, 2020.

[40] Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. Nomic Embed: Training a Reproducible Long Context Text Embedder. **arXiv:2402.01613**, 2024.

[41] Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. One Embedder, Any Task: Instruction-Finetuned Text Embeddings. In **Findings of the Association for Computational Linguistics: ACL 2023**, pp. 1102–1121, 2023.

[42] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking. **Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)**, 2021.

表 7　対照事前学習におけるハイパーパラメータと設定

	v1 データセット			v2 データセット		
サイズ	Small	Base	Large	Small	Base	Large
学習率	1×10^{-4}	5×10^{-5}	3×10^{-5}	1×10^{-4}	5×10^{-5}	3×10^{-5}
系列長	256	256	192	256	256	256
バッチサイズ	8192	8192	8192	8192	8192	8192
温度	0.01	0.01	0.01	0.01	0.01	0.01
ハード負例数	1	1	1	0	0	0

表 8　微調整におけるハイパーパラメータと学習設定

	v1 データセット			v2 データセット		
サイズ	Small	Base	Large	Small	Base	Large
学習率	1×10^{-5}	5×10^{-6}	3×10^{-6}	1×10^{-5}	5×10^{-6}	3×10^{-6}
系列長	512	512	512	512	512	512
バッチサイズ	512	512	512	512	512	512
温度	0.01	0.01	0.01	0.01	0.01	0.01
ハード負例数	15	15	15	15	15	15
知識蒸留	✓	✓	✓			

A　既存データセット

対照事前学習に適したデータを収集した．対象は次の通り: 日本語 Wikipedia[17]，日本語 WikiBooks[18]，日本語 Wiktionary[19]，MQA の日本語サブセット[20]，CC News の日本語サブセット[21]，Japanese Research Corpus (JRC)[22),23)], LLM-jp Corpus v3 の kaken サブセット[24] と warp-html サブセット[25]，青空文庫の小説タイトルとその読みをペアにしたデータセット[26]，日本語 WordNet [31] の同義語集合，日本語形態素解析器 Sudachi [32] の同義語辞書[27]，MeCab [33] と IPA 辞書および mecab-ipadic-NEologd [34] を用いて日本語 Wikipedia の記事タイトルの読みを推定したデータセット[28]，やさしい日本語コーパス [35]，および，やさしい日本語拡張コーパス [36] (本稿では総称して SNOW と呼ぶ)，JWTD [37]，Wiki Atomic Edits [38]，JSNLI [39].

B　対照事前学習の詳細

ハイパーパラメータと実装　対照事前学習におけるハイパーパラメータや学習設定を表 7 に一覧する．対照事前学習では非常に大きなバッチサイズでモデルを訓練する．Li ら [5] を参考に，学習時のバッチサイズは 8192 とした．E5 [3] と同様，位置埋め込みは学習せず固定した[29]．また，正例文章の文の順序をランダムにシャッフルするデータ拡張を導入した．その際の文分割には Konoha[30] を利用した．

接頭辞　近年の埋め込みモデルでは，テキストを埋め込む際に接頭辞を付与する手法が広く採用されている [3, 4, 5, 6, 40]．例えば，検索クエリの文頭に query: を，検索対象文章に passage: などを付加するもので，特に非対称な類似度を計算する必要がある検索タスクで性能が向上することが知られている．本稿ではこれらの研究を参考にしつつ，日本語に特化したモデルのための接頭辞として「クエリ: 」と「文章: 」をテキストに付加し訓練した．

ミニバッチ作成の工夫　GTE [5] や InstructOR [41] と同様，1 つのバッチに同種のデータセットの事例のみを含める dataset-homogeneous batching を用いた．これは複数のデータセットを混在させると，データセットの分布の違いを手がかりにモデルが好ましくない学習経過を辿る可能性があるためである．また，ひとつのミニバッチ内で系列長を揃えられるため，パディングが減り学習を効率化できるという利点も存在する．さらに，ミニバッチ内に重複するテキストが存在すると偽の負例 (本来は正例となるべきであるが負例として学習されてしまう事例) になってしまうため，あらかじめ重複文をミニバッチ内から除く前処理も行った．

C　高品質データでの微調整の詳細

高品質データを用いた微調整におけるハイパーパラメータや学習設定を表 8 に一覧する．高品質データでの微調整についても，対照事前学習と同様，接頭辞の付加やミニバッチ作成時の工夫を行った．リランカを用いたフィルタリングは日本語 SPLADE [42] に関する取り組み[31]を参考にした．予備実験の結果，リランカからの知識蒸留を用いない場合の方が検索タスクにおける性能が高くなる傾向が観察されたため，v2 データセットを用いた微調整ではリランカからの知識蒸留を導入しなかった．また，合成データセットを微調整の段階に含めると全体的な性能が低下する傾向にあったため，人手データのみを用いて微調整を行った．

17) https://huggingface.co/datasets/hpprc/jawiki
18) https://huggingface.co/datasets/hpprc/jawiki-books
19) https://huggingface.co/datasets/hpprc/jawiki-wiktionary
20) https://huggingface.co/datasets/clips/mqa
21) https://huggingface.co/datasets/intfloat/multilingual_cc_news
22) https://huggingface.co/datasets/kunishou/J-ResearchCorpus
23) Japanese Research Corpus は日本語論文を集めた高品質なデータセットである．ただし本データセットには，評価ベンチマーク JMTEB [29] に含まれる論文誌「自然言語処理」のデータが存在している．そこで評価時のリークを避けるため，該当のデータは学習データから除外した．
24) https://gitlab.llm-jp.nii.ac.jp/datasets/llm-jp-corpus-v3/-/tree/main/ja/kaken
25) https://gitlab.llm-jp.nii.ac.jp/datasets/llm-jp-corpus-v3/-/tree/main/ja/ja_warp_html/level2
26) https://huggingface.co/datasets/hpprc/aozora-yomi
27) https://github.com/WorksApplications/SudachiDict/tree/develop
28) https://huggingface.co/datasets/hpprc/ihyoki
29) https://github.com/microsoft/unilm/issues/1120
30) https://github.com/himkt/konoha
31) https://secon.dev/entry/2024/10/23/080000-japanese-splade-tech-report/