日本語文埋め込みの文書検索性能と検索補助付き生成での評価

矢野 千紘 *　　塚越 駿 *　　笹野 遼平　　武田 浩一
名古屋大学大学院情報学研究科
{yano.chihiro.j3,tsukagoshi.hayato.r2}@s.mail.nagoya-u.ac.jp
{sasano,takedasu}@i.nagoya-u.ac.jp

概要

LLM の実応用が加速する中，LLM を訓練なしで拡張可能な検索補助付き生成 (Retrieval Augmented Generation: RAG) と，RAG に用いられる文埋め込みを利用した密ベクトル検索は，重要な技術として注目されている．しかし，日本語における密ベクトル検索や RAG の評価は十分とは言えない．そこで本研究では，日本語文書検索タスクにおける文埋め込み手法の性能を評価する．さらに，有望なモデルを RAG に利用した場合の性能評価も行い，日本語における密ベクトル検索と RAG の全般的な評価を行う．

1　はじめに

自然言語文のベクトル表現である文埋め込みを用いた密ベクトル検索は，文脈や意味を考慮した高度な情報検索を可能とすることから，盛んに研究が行われている [1, 2]．さらに近年では，情報検索と生成モデルを組み合わせた Retrieval Augmented Generation (RAG) が注目を集めている [3, 4]．RAG は大規模言語モデル (LLM) を用いて推論する際の幻覚を低減でき [5]，パラメータを更新せずモデルの知識を拡張可能であるため，LLM の実用上重要な技術となっている．RAG では LLM に与える参考文書が推論結果に直接影響を及ぼすと考えられるため，文書検索の性能が重要になる．しかし，密ベクトル検索，および，RAG の性能評価は英語を中心に行われており [2, 4]，日本語を対象とした評価はほとんど行われていない．そこで本研究では，日本語を対象に文埋め込みモデルを用いた文書検索タスクでの性能評価を行う．さらに，それらの文埋め込みモデルを RAG に用いた際の性能評価を行う．

2　文書検索タスクでの評価

密ベクトル検索は，埋め込み空間上で検索クエリと文書の類似度を測ることにより文書検索を行う手法である [1]．本研究では複数の文埋め込みモデルを用い，文書検索タスクでの性能評価を行う．[1]

表 1　密ベクトル検索に用いるデータセットの統計値

データセット名	クエリ数	文書数	平均正例数
AI 王	864	4,288,198	11.8
Mr.TyDi	928	7,000,027	1.0
MIRACL	860	6,953,614	2.1

2.1　評価実験

文書検索タスクの評価には，検索クエリと，それに対応する正例文書が必要となる．本研究では表 1 に示す 3 つのデータセットを用いた．

AI 王[2]　Wikipedia の記事タイトルが解答となるような質問と，それに対応する Wikipedia の文書が関連づけられたデータセットである．関連文書のうち，解答の文字列を含む文書を正例文書，含まない文書を負例文書としている．

Mr.TyDi [6]　多言語質問応答データセットである TyDi QA [7] に単言語の検索用コーパスを付与した，日本語を含む 11 言語に対応する単言語文書検索評価用のデータセットである．Mr.TyDi は各言語について人手で作成された質問と，Wikipedia の文書が関連づけられたデータセットである．関連文書は各質問について Google 検索上位の Wikipedia 記事を段落ごとに分割して作成され，解答の文字列を含む文書を正例文書，含まない文書を負例文書としている．

MIRACL [8]　日本語を含む 18 言語において，単言語文書検索評価を行うための多言語データセットである．Mr.TyDi を元に構築されており，Mr.TyDi 中の関連文書に解答が含まれないような質問を削除し，より多くの文書について人手でラベル付けを行った比較的高品質なデータセットである．

* Equal contribution.

1　検索タスクでは文書埋め込みを利用する場合もあるが，簡単のため本論文ではそれらも一律に文埋め込みと呼称する．

2　https://sites.google.com/view/project-aio/dataset

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

表 2　文書検索タスクでの評価結果．表中の値は全て 100 をかけたものであり，ρ はスピアマンの順位相関係数を，R@k は Recall@k を表す．◇♠♡ が付記されたモデルはそれぞれ順に AI 王，Mr.TiDy，MIRACL を訓練データセットとして用いていることを表す．AI 王データセットの関連文書は疎ベクトル検索によって取得されており，BM25 は極めて有利であるため † を付している．PKSHA 社の GLuCoSE は JSTS を訓練時の開発セットに利用しているため ∗ を付している．

システム	JSTS	AI 王 ◇			Mr.TyDi ♠			MIRACL ♡		
	ρ	R@5	R@10	R@30	R@5	R@10	R@30	R@5	R@10	R@30
ベースライン										
BM25	-	82.8†	88.8†	94.9†	31.9	41.6	56.6	53.7	63.6	77.6
cl-tohoku/bert-base-japanese-v3 (Mean)	74.1	48.6	58.2	68.8	3.5	5.7	9.2	5.5	9.9	16.6
日本語文埋め込みモデル										
llm-book/bpr-aio-base◇	77.8	77.2	83.8	88.3	18.4	25.9	38.0	33.8	42.9	56.5
cl-nagoya/unsup-simcse-ja-base	79.0	50.2	58.1	69.1	8.9	14.2	23.3	16.3	22.6	33.5
cl-nagoya/unsup-simcse-ja-large	81.4	58.3	66.2	74.9	10.1	15.5	25.2	16.6	24.1	36.2
cl-nagoya/sup-simcse-ja-base (jsnli)	80.9	57.2	65.3	76.4	16.3	22.3	34.2	26.0	34.5	47.2
cl-nagoya/sup-simcse-ja-large (jsnli)	83.1	57.1	65.5	76.3	16.8	24.1	36.3	29.8	37.7	50.0
cl-nagoya/sup-simcse-ja-base (miracl)♡	77.3	61.1	70.0	79.7	54.4	64.5	77.3	70.9	79.3	87.1
cl-nagoya/sup-simcse-ja-base (jsnli+miracl)♡	80.9	60.1	69.1	78.2	55.0	61.6	72.7	71.6	78.4	85.3
pkshatech/GLuCoSE-base-ja♠	81.8∗	54.9	64.2	74.4	43.3	54.3	67.8	53.3	64.3	76.5
pkshatech/simcse-ja-bert-base-clcmlp	80.1	55.9	64.6	74.2	18.0	24.9	40.6	26.6	37.7	52.1
多言語文埋め込みモデル										
intfloat/multilingual-e5-small♠♡	78.9	75.5	82.2	87.2	74.2	84.6	91.3	84.7	90.6	96.9
intfloat/multilingual-e5-base♠♡	79.7	77.1	82.3	88.8	77.5	84.8	90.8	84.2	91.0	96.6
intfloat/multilingual-e5-large♠♡	81.9	**80.8**	**85.6**	**90.9**	**81.4**	**87.8**	**93.6**	**89.2**	**93.4**	**98.1**
sentence-transformers/LaBSE	76.1	24.8	29.5	37.3	3.1	4.6	7.8	5.2	8.5	13.6
sentence-transformers/stsb-xlm-r-multilingual	78.4	21.9	26.0	34.1	2.4	3.2	5.2	4.3	5.7	9.0
商用システム										
OpenAI (text-embedding-ada-002)	79.0	61.9	70.9	80.4	35.0	45.8	62.0	47.8	60.8	77.3

評価対象モデル　日本語文埋め込みモデルとして名大 SimCSE [9]，PKSHA 社の GLuCoSE，多言語文埋め込みモデルとして multilingual E5 (mE5) [10]，LaBSE [11]，OpenAI 社の ada を含む，複数のモデルを評価した．埋め込み同士の類似度として，コサイン類似度を用いた．ベースラインとして疎ベクトル検索手法である BM25 を用いた場合と，事前学習済み言語モデルの埋め込みをそのまま用いた場合の性能も評価した．各モデルの詳細は付録 A に示す．

評価設定　評価には，各データセットのうち正例文書が一つ以上存在するデータを用いた．各データセットの統計値を表 1 に示す．評価指標には，検索上位 k 個の文書中に少なくとも一つ正例文書が含まれる割合を示す Recall@k を用いた．また参考として，意味的類似度 (Semantic Textual Similarity: STS) タスクにおける各システムの性能を，Tsukagoshi ら [9] と同様に JSTS [12] を用い評価した．

2.2　評価結果

実験結果を表 2 に示す．検索データセットで学習された mE5 や GLuCoSE，名大 SimCSE (miracl) などのモデルが，ベースラインや商用システムをほとんどの場合で上回る高い性能を示した．一方で，自然言語推論データセットのみを用いて訓練された名大 SimCSE (jsnli) などのモデルは，STS タスクにおいて高い性能を示しているものの，文書検索タスクにおいては比較的低い性能となった．

文書検索タスクは STS タスクと異なり，正例となる二つの文の意味が必ずしも一致しないため，非対称的な類似度を捉えることが重要である．すべてのモデルサイズで一貫して高い性能を示した mE5 は，他のモデルと異なり，クエリと文書に異なる接頭辞を付与した上で，大規模なデータセットで訓練されたモデルである。接頭辞により入力文の非対称性を考慮した埋め込みを生成できることが，mE5 が高い性能を示した要因であると考えられる．

3　RAG での評価

Retrieval Augmented Generation (RAG) とは生成モデルに情報検索を組み合わせ，検索した情報に基づく生成を可能とする技術である．本研究では 2 節にて評価した埋め込みモデルを用いて，日本語におけ

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

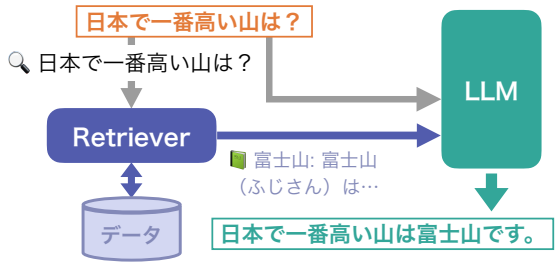

図 1　単純な質問応答の推論過程

る RAG の性能評価を行う．評価には，1 回の検索で回答できる単純な質問応答タスクと，複数回の検索を必要とする多段階質問応答タスクを用いる．

3.1　単純な質問応答

RAG を用いた質問応答の過程を図 1 に示す．まず，データセット中の質問文をクエリとして文書検索を行い，次に，質問文と検索された文書を言語モデルに入力して応答を生成する．

評価には AI 王データセットを利用した．言語モデルに入力するプロンプトには，書籍『大規模言語モデル入門』[13] と同じものを利用した．応答生成には OpenAI 社の GPT-3.5 を用いた．

3.2　多段階質問応答

多段階質問応答は，複数回の文書検索が必要となるような質問に回答するタスクである．推論の過程を図 2 に示す．

評価には日本語の多段階推論データセットである JEMHopQA [14] を用いた．JEMHopQA は，2 つの Wikipedia 記事の情報を統合する構成問題と，比較する比較問題からなるデータセットである．つまり，回答には 2 度の検索と，得られた情報の整理，常識的な比較推論が必要となる．

多段階質問応答タスクに対して，LLM を用いた推論手法が複数提案されている [15, 16]．本研究では，それらの中でも単純かつ高い性能を示す ReAct [17] を用いた．ReAct は LLM に所定の行動と推論を交互に行わせる手法であり，LLM は行動した結果を都度利用して推論を行い，次の行動を決定する．本研究では，検索クエリ生成と文書検索を繰り返し実施するため，以下の行動を定義した．

- Search[query]: query を検索システムに入力
- Finish[answer]: answer を回答とし推論を終了

また，GPT-3.5 では複雑な推論が困難であったため，応答生成には 3.1 節と異なり GPT-4 を用いた．

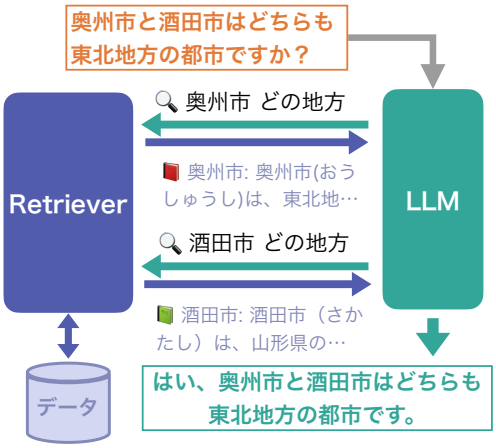

図 2　多段階質問応答の推論過程

表 3　RAG での評価に用いるデータセットの統計値

データセット名	推論段数	クエリ数	LLM
AI 王	1	1,000	GPT-3.5
JEMHopQA	2	120	GPT-4

3.3　評価実験

評価設定　各質問への回答時に関連文書として LLM に入力する文書は，ReAct にならい，質問に対する類似度が高い上位 5 文書とした．JEMHopQA に対応する Wikipedia コーパスは記事単位で分割されているが，1 記事が埋め込みモデルの最大系列長を超えることがあったため，同時期の Wikipedia から作成され，より細かな分割が施されている AI 王データセットを検索対象の文書群として利用した．各データセットの統計値を表 3 に示す．[3]

文書検索性能と質問応答の正答率の関係を調査するため，文書検索性能 (Recall@5) も評価した．JEMHopQA における文書検索性能の評価には，データセット中で正例とされている Wikipedia 記事と同一の記事に含まれるすべての文書を正例文書として扱った．検索を用いないベースライン手法として，AI 王データセットについては指示と質問文のみを入力する単純な質問応答を，JEMHopQA については Chain of Thought [15] を用いた質問応答を評価した．推論に用いた実際のプロンプトを付録 B に示す．

GPT-4 を用いた評価　生成モデルを質問応答タスクに適用する場合，文字列マッチングベースの評価指標はモデルの性能を過小に評価してしまうことが知られている [4]．そこで，文字列マッチングによ

3　AI 王データセットについては，2 節の文書検索タスクにおける評価と異なり，本節の実験では正例文書が存在しない質問も評価対象に含めているため，表 1 とクエリ数が異なる．

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

表 4 RAG での評価結果．AI 王の Recall は正解文書がない問題を除いた値である．

モデル	AI 王			JEMHopQA		
	R@5	Acc. (In)	Acc. (GPT-4)	R@5	Acc. (In)	Acc. (GPT-4)
検索なし	-	47.0	57.1	-	51.7	57.5
BM25	82.8†	68.2	76.2	56.7	66.7	69.2
cl-tohoku/bert-base-japanese-v3 (Mean)	48.6	59.4	68.0	2.5	28.3	33.3
cl-nagoya/sup-simcse-ja-large (jsnli)	57.2	62.5	70.7	21.7	41.7	45.0
cl-nagoya/sup-simcse-ja-base (jsnli+miracl)	60.1	62.1	70.7	32.5	59.2	60.8
pkshatech/GLuCoSE-base-ja	54.9	63.2	70.3	64.2	70.0	70.8
intfloat/multilingual-e5-large	80.8	**69.0**	**77.5**	60.8	**74.2**	**75.8**

表 5 AI 王データセットでの検索成否と回答性能の分布

	正答	誤答	合計	Acc. (GPT-4)
検索成功	641 件	57 件	698 件	91.8%
検索失敗	134 件	168 件	302 件	44.4%
合計	775 件	225 件	1000 件	77.5%

る評価に加え，GPT-4 を用いた評価を行った．文字列マッチングによる評価では，生成した応答に正解文字列が含まれている場合正解とした．[4]GPT-4 による評価では，質問，正答，生成された応答を入力し，生成された応答が質問に対する正答とみなせるかどうかを GPT-4 に判定させた．GPT-4 を用いた評価の妥当性を調査するため，AI 王データセットの先頭 300 件に対し著者らによる人手評価を実施し，GPT-4 による正誤判定との一致率を測った．その結果，文字列マッチングによる評価では 264/300 件，GPT-4 による評価では 294/300 件が人手評価と一致した．

3.4 評価結果

RAG での評価結果を表 4 に示す．検索なしと比較してほとんどの場合で性能の向上が見られたが，JEMHopQA において検索タスクで訓練していないモデルを用いた場合のみ性能の低下が見られた．これは，誤った参考文書が推論におけるノイズとなったためであると考えられる．2 節と同様，mE5 は一貫して最も高い性能を示した．

mE5 を用いた場合の，AI 王データセットにおける検索成否と正誤割合の関係を表 5 に示す．表から，検索成功時には正答率が 91.8%と高く，文書検索の成功が RAG の性能向上に寄与することがわかった．

3.5 検索クエリの形式が及ぼす影響

ReAct で生成される検索クエリは，「日本 最高峰」などのように検索キーワードを並べた単語列である

4 解答が「YES」もしくは「NO」である問題に関してはそれぞれ「はい」と「いいえ」も正解とした．

表 6 検索クエリを変化させた時の JEMHopQA での評価

クエリ	R@5	平均文書長	GPT-4
intfloat/multilingual-e5-large			
単語列	60.8	216.4	75.8
文	79.2	252.0	**77.5**
cl-nagoya/sup-simcse-ja-base (jsnli+miracl)			
単語列	32.5	165.9	60.8
文	50.8	268.4	75.0
pkshatech/GLuCoSE-base-ja			
単語列	64.2	210.7	70.8
文	64.2	242.1	65.8

ことが多い．しかし，文埋め込みモデルは文を訓練データとしているため，訓練時と推論時のデータ間に乖離があると考えられる．そこで生成する検索クエリが文になるようにプロンプティングした場合の実験結果を表 6 に示す．結果から，検索キーワードの羅列よりも文で検索することで，ほとんどのモデルの検索性能が大きく向上することがわかった．具体的には Recall@5 が改善し，また，検索された文書の平均文書長が大きくなった．ただし，GLuCoSE を用いた場合には検索性能と RAG の正答率に相関が見られなかった．これは，JEMHopQA の問題数が少ないことや，多段階質問応答の推論過程が複雑であることが理由であると考えられる．

4 おわりに

本研究では，文埋め込みモデルの文書検索タスク，および，RAG における性能を評価した．文書検索タスクによる評価では，検索データセットで訓練されたモデルが高い性能を示した．中でも，接頭辞を導入し，大規模なデータセットで訓練された mE5 が一貫して高い性能を示した．RAG での評価では，文書検索の成否が RAG の正答率に大きな影響を与えることを示した．また，検索クエリを単語の列挙ではなく文とすることで，多くのモデルで検索および RAG の性能が向上することがわかった．

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，JSPS 科研費 23KJ1134 の支援を受けたものである．

参考文献

[1] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense Passage Retrieval for Open-Domain Question Answering. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 3784–3803, 2020.

[2] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In **Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS)**, 2021.

[3] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In **Advances in Neural Information Processing Systems (NeurIPS)**, pp. 9459–9474, 2020.

[4] Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. Evaluating open-domain question answering in the era of large language models. In **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 5591–5606, 2023.

[5] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval Augmentation Reduces Hallucination in Conversation. In **Findings of the Association for Computational Linguistics (EMNLP)**, pp. 3784–3803, 2021.

[6] 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.

[7] Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. TyDi QA: A Benchmark for Information-Seeking Question Answering in Typologically Diverse Languages. **Transactions of the Association for Computational Linguistics (TACL)**, Vol. 8, pp. 454–470, 2020.

[8] 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.

[9] Hayato Tsukagoshi, Ryohei Sasano, and Koichi Takeda. Japanese SimCSE Technical Report. **arXiv:2310.19349**, 2023.

[10] 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.

[11] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic BERT Sentence Embedding. In **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 878–891, 2022.

[12] Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. JGLUE: Japanese General Language Understanding Evaluation. In **Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC)**, pp. 2957–2966, 2022.

[13] 山田育矢, 鈴木正敏, 山田康輔, 李凌寒. 大規模言語モデル入門. 技術評論社, 2023.

[14] 石井愛, 井之上直也, 関根聡. 根拠を説明可能な質問応答システムのための日本語マルチホップ QA データセット構築. 言語処理学会第 29 回年次大会, 2023.

[15] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. Chain of Thought Prompting Elicits Reasoning in Large Language Models. **arXiv:2201.11903**, 2023.

[16] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In **International Conference on Learning Representations (ICLR)**, 2023.

[17] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. In **International Conference on Learning Representations (ICLR)**, 2023.

[18] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In **Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)**, pp. 2356–2362, 2021.

[19] 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.

[20] Mandy Guo, Zihang Dai, Denny Vrandečić, and Rami Al-Rfou. Wiki-40B: Multilingual Language Model Dataset. In **Proceedings of the Twelfth Language Resources and Evaluation Conference (LREC)**, pp. 2440–2452, 2020.

[21] 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.

[22] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 6442–6454, 2020.

[23] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised Cross-lingual Representation Learning at Scale. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 8440–8451, 2020.

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

表 7　AI 王データセットでの RAG に用いたプロンプト

あなたには今からクイズに答えてもらいます。 問題を与えますので、その解答のみを簡潔に出力してください。 また解答の参考になりうるテキストを与えます。 解答を含まない場合もあるのでその場合は無視してください。 {retrieved passages} 問題：{query} 解答：

表 8　JEMHopQA での RAG に用いた ReAct のプロンプト．色付き文字の部分は文クエリで RAG を行う際に追加した文字列を示す．

Solve a question answering task with interleaving Thought, Action, Observation steps. Thought can reason about the current situation, and Action can be two types: (1) Search[query], which searches the query in a search engine and get references. The query should be a sentence, not a sequence of words. (2) Finish[answer], which returns the answer and finishes the task. Please conduct all output in Japanese. Here are some examples. Question: 『仮面ライダー電王』と『あまちゃん』，放送回数が 多いのはどちらでしょう？ Thought 1: 『仮面ライダー電王』の放送回数を検索した後， 『あまちゃん』の放送回数を検索し，それらの比較を行う． Action 1: Search[『仮面ライダー電王』の放送回数は 何回ですか？] Observation 1: 『仮面ライダー電王』の放送回数は 49 回です． Thought 2: 『仮面ライダー電王』の放送回数は 49 回である． 次に『あまちゃん』の放送回数を検索する． Action 2: Search[『あまちゃん』の放送回数は何回ですか？] Observation 2: 『あまちゃん』の放送回数は 156 回です． Thought 3: 『仮面ライダー電王』の放送回数は 49 回である． 『あまちゃん』の放送回数は 156 回である．『あまちゃん』の 放送回数が多いので，最終的な答えは『あまちゃん』である． Action 3: Finish[『あまちゃん』] Question: {query}

表 9　JEMHopQA での RAG に用いた Chain of Thought のプロンプト

Please conduct all output in Japanese. Question: 『仮面ライダー電王』と『あまちゃん』、放送回数が 多いのはどちらでしょう？ Answer: 『仮面ライダー電王』の放送回数は 49 回である。『あ まちゃん』の放送回数は 156 回である。『あまちゃん』の放送 回数が多いので、最終的な答えは『あまちゃん』である。 Question: {query}

表 10　GPT-4 での評価に用いたプロンプト

You are a professional quiz grader. Referring to the question and correct answer pairs, tell me if the studentanswer is correct or 1 or 0. Output 1 if the answer is correct, 0 if it is wrong, and nothing else. question: {query} correct answer: {answer} student answer: {pred} score:

A　評価対象モデルの詳細

ベースライン　疎ベクトル検索手法として BM25 を評価した．実装には Pyserini [18] を利用し，分かち書きには Kuromoji[5]を利用した．密ベクトル検索のベースラインとして東北大 BERT-base の出力埋め込み表現の平均をとったものを評価した．

日本語密ベクトル検索モデル　質問応答データセットで学習された既存の日本語密ベクトル検索モデルとして，東北大 BERT を AI 王データセットで訓練した `llm-book/bpr-aio` を評価した．

日本語文埋め込みモデル　名古屋大学の武田笹野研究室が公開する名大 SimCSE [9]，PKSHA 社が公開する GLuCoSE および，PKSHA SimCSE を評価した．名大 SimCSE は，対照学習を用いた文埋め込み手法である SimCSE [19] を用いて東北大 BERT を微調整したモデルである．教師なし設定には Wiki40B [20] を，教師あり設定には JSNLI [21]，MIRACL 及び双方を利用したモデルを評価した．JSNLI と MIRACL を併用したモデルでは，学習データの件数を等しくするため，MIRACL のアップサンプリングを行った．

GLuCoSE は日本語 LUKE [22] を弱教師あり学習によって事前学習したのち，自然言語推論，言い換え，質問応答データセットで教師あり対照学習を行った文埋め込みモデルである．PKSHA SimCSE は東北大 BERT に対し JSNLI を利用して教師あり対照学習を行った文埋め込みモデルである．

多言語文埋め込みモデル　Multilingual E5 (mE5)，LaBSE [11]，STS データセットで微調整された XLM-R [23] を評価した．mE5 は，大規模な弱教師あり学習ののち，教師あり対照学習によって微調整された文埋め込みモデルである E5 [10] の多言語版であり，XLM-R をベースとして，自然言語推論，質問応答，事実検証データセットなどを用いて対照学習されている．LaBSE [11] は対訳データを用いて大規模に事前学習を行った文埋め込みモデルである．

B　RAG に用いたプロンプト

実際に RAG の推論に用いたプロンプトを表 7，8，9 に示す．GPT-4 での評価に用いたプロンプトを表 10 に示す．

5　https://github.com/atilika/kuromoji

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