RALLE: A Framework for Developing and Evaluating Retrieval-Augmented Large Language Models

星 康人 * 宮下 大輔* Youyang Ng 立野 賢登 森岡 靖太 鳥井 修 出口 淳
キオクシア株式会社
yasuto1.hoshi@kioxia.com

概要

Large Language Model (LLM) と検索を組み合わせた retrieval-augmented large language model (R-LLM) の効率的な開発と定量的な評価を可能とするフレームワーク，RALLE (Retrieval-Augmented LLM Development and Evaluation) を提案する[1)]．既存の R-LLM 開発用フレームワークとは異なり，RALLE は検索や生成などの個々の推論過程を動作させながら行うプロンプト開発や，構築した R-LLM に対して任意のベンチマークデータセットを用いた定量的な評価を実現する．

1 はじめに

Large Language Model (LLM) は，自然言語の理解と生成において高い能力を持つことが知られている [2, 3, 4]．しかし，LLM は事実に関する質問応答において，hallucination [5, 6]，モデルが持つ古い知識 [7]，記憶効率 [8] 等の課題を持つ．これらの課題の軽減策として，LLM と検索を組み合わせた retrieval-augmented large language model (以下，R-LLM) が有望視されている [9]．

R-LLM は，モデル外部のデータベースに保存された情報を検索して推論に利用することができ [9, 10]，オープンドメイン QA における精度向上をもたらすことが知られている [11]．さらに，R-LLM を用いることで，hallucination の抑制 [12]，知識原の更新の容易さ [13, 10]，参照した情報が明確，などの利点を追加学習なしで得られる．

しかし，R-LLM の効率的な開発や，R-LLM の応答性能の定量的な評価を実現する仕組みの整備は不十分である．R-LLM 構築に用いられる ChatGPT Retrieval Plugin[2)]，Guidance[3)]，LangChain [14] のようなフレームワークは抽象度が高く[4)]，検索や生成などの個々の推論ステップの機能の検証やプロンプトの最適化が困難である．また，これらのフレームワークでは任意のベンチマークを用いた R-LLM の定量的な性能評価をサポートしていない．

そこで，本論文では R-LLM の開発と評価のためのフレームワークである RALLE を提案する．RALLE を用いることで，R-LLM の効率的な開発，および客観的な評価指標を用いた性能評価が可能となる．

2 RALLE の使い方

図 1 に RALLE の主な特徴を示す[5)]．RALLE を用いた R-LLM 開発の主な流れは，(1) 知識源となる文書の埋め込みと indexing，(2) R-LLM の推論チェーンの設計，(3) 開発した R-LLM の評価，という 3 つの要素で構成される．

(1) **知識源となる文書の埋め込みと** indexing では，任意の文書検索器を用いて文書をエンコードし，任意のアルゴリズムで indexing する．RALLE では密ベクトルの index として Faiss Flat index [15]，HNSW [16]，DiskANN [17] がデフォルトで利用可能である．

(2) R-LLM **の推論チェーンの設計** では，検索器と LLM を組み合わせて，特定の用途に適した様々な推論パイプラインを設計できる．単一アクションのチェーンは，LLM を用いた closed-book QA システム，ないし検索器を用いて検索結果を返すシステムとなりうる．複数アクションで構成されるチェーンは，シンプルな retriever-reader の他，検索用にクエリ書き換えを伴う [18] のような複雑な R-LLM のワークフローとなりうる．推論チェーンの設計は，Gradio [19] ベースの GUI 上で行うことができる．

---

* These authors contributed equally to this work.

1） https://github.com/yhoshi3/RaLLe
本論文は，EMNLP2023 発表論文 [1] ©2023 Association for Computational Linguistics の抜粋・翻訳版です．

2） https://github.com/openai/chatgpt-retrieval-plugin

3） https://github.com/microsoft/guidance

4） 注：RALLE はこれらを使用せず実装されている．

5） RALLE の操作説明動画も参照されたい．

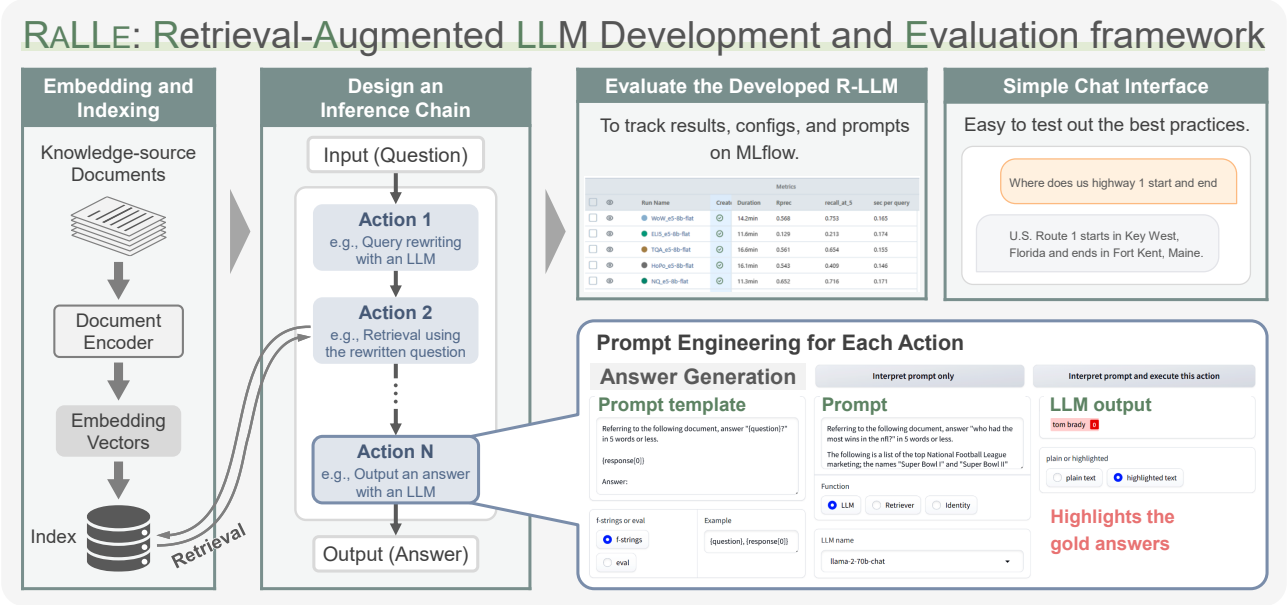

図 1　RALLE の概要．R-LLM を構成するアクションは任意の数定義できる．個々のアクションは個別に実行でき，そのアクションで定義されているプロンプトをテストできる．評価実験の設定や結果は MLflow で管理できる．チャット画面で，構築した R-LLM をテストできる．

RALLE では，LLM や検索（クエリの加工）に用いるプロンプトテンプレートをインタラクティブに開発できる．各アクションを独立に実行でき (図 1 右下)，各アクションの出力を確認しながらプロンプトテンプレートを改善できる．より汎用なプロンプトテンプレートの開発のために，RALLE は Python の f-strings と eval 関数をサポートしている．

(3) **開発した R-LLM の評価** は，独自の QA データセットを含む任意のベンチマークデータセットを用いて可能である[6]．開発した R-LLM の評価は GUI 上で開始できる．実験の追跡には MLflow [20] が利用でき，R-LLM の設定ファイルやプロンプトテンプレートを追跡できる．これにより，異なる実験設定における精度を比較でき，より優れた R-LLM の開発に寄与しうる．

さらに，RALLE はチャットインターフェースの構築もサポートしており，開発者は構築した R-LLM をチャット画面で試験できる（図 1 右上）．

3　実験設定

本章では，オープンソースの検索器と LLM を組み合わせて構築した R-LLM の性能を，知識集約型タスクで評価する際の実験設定を述べる．

3.1　タスクとデータセット

評価には KILT (Knowledge Intensive Language Tasks) ベンチマーク [21] を用いる．KILT ベンチマークは，計 5 つの知識集約型タスク (fact checking, entity linking, slot filling, open-domain question answering, and dialogue) にわたる計 11 のデータセットからなるベンチマークである．KILT の学習セットをプロンプト開発に使用し，開発セットを評価に使用する．

検索対象の知識源として，KILT が提供する前処理済みの Wikipedia パッセージ集を使用する．このパッセージ集は，2019 年 8 月 1 日の英語版 Wikipedia ダンプデータから作られており，計 590 万記事と 2220 万の 100 単語パッセージで構成される．本稿では，各パッセージに記事のタイトルを付加する前処理を行ったパッセージを検索に使用する．

3.2　ベースライン

モデル外部の知識を用いない *closed-book* 設定のベースラインとして **BART-large** モデル [22]，および *open-book* 設定のベースラインとして **RAG** モデル [23] の結果を用いる．これらのベースラインモデルは KILT ベンチマークで fine-tune されているのに対し，本稿の実験で用いた LLM および我々が構築した R-LLM はそのような fine-tune がなされていない点に注意されたい．

6)　詳細は https://github.com/yhoshi3/RaLLe/blob/main/docs/using_custom_datasets.md を参照．

	Fact Check.	Entity Linking			Slot Filling		Open Domain QA				Dial.
Dataset	**FEV**	**AY2**	**WnWi**	**WnCw**	**T-REx**	**zsRE**	**NQ**	**HoPo**	**TQA**	**ELI5**	**WoW**
Model / Metric	Accuracy						Exact Match			RL	F1
BART-large$^{\diamond}$ (*closed-book*)	80.7	**86.6**	47.9	**48.0**	43.8	3.0	26.2	16.9	32.5	22.7	**13.8**
Llama2-70B (*closed-book*)	33.6 (74.9)	39.8 (54.5)	42.8 (**53.8**)	39.2 (**55.7**)	28.5 (40.5)	11.3 (13.6)	19.6 (37.4)	13.9 (25.1)	67.4 (80.8)	**23.0**	13.3
RAG$^{\diamond}$	**87.7**	77.4	**49.0**	46.7	**61.5**	**47.4**	**48.8**	27.7	61.7	16.1	13.3
e5 + W-Vicuna-13B	10.6 (42.4)	51.2 (**57.9**)	48.6 (51.4)	45.6 (51.4)	31.6 (46.1)	23.0 (29.3)	18.7 (38.0)	19.7 (28.3)	43.1 (67.7)	21.4	12.3
e5 + Llama2-13B	66.3 (73.5)	51.2 (57.9)	48.6 (51.4)	45.6 (51.4)	17.2 (42.3)	31.7 (41.1)	36.1 (43.3)	14.3 (25.5)	56.3 (76.2)	20.9	12.3
BM25 + Llama2-70B	46.2 (86.3)	18.0 (35.9)	19.1 (32.2)	14.2 (30.9)	25.9 (43.0)	31.4 (37.8)	25.3 (34.3)	25.9 (33.4)	65.8 (80.0)	21.3	12.2
e5 + Llama2-70B	49.9 (**88.6**)	51.2 (57.9)	48.6 (51.4)	45.6 (51.4)	28.9 (49.2)	35.0 (**43.2**)	36.4 (**48.8**)	**28.1** (**35.8**)	71.1 (**83.9**)	21.5	13.2
`e5 (DiskANN)`	49.9 (87.9)	44.3 (50.5)	45.3 (48.1)	43.0 (48.8)	25.3 (43.9)	32.1 (37.9)	36.1 (48.4)	26.7 (34.3)	70.4 (83.2)	21.5	13.1
`top-2`	49.3 (88.1)	51.2 (57.9)	48.6 (51.4)	45.6 (51.4)	23.5 (44.9)	34.7 (43.0)	33.7 (46.2)	23.8 (34.2)	**71.3** (82.9)	21.6	13.3
`top-10`	50.2 (88.0)	51.2 (57.9)	48.6 (51.4)	45.6 (51.4)	31.1 (**49.3**)	35.4 (42.5)	35.2 (48.1)	24.9 (35.7)	59.3 (82.8)	21.5	13.2

表 1　KILT ベンチマーク開発セットにおける下流タスクの精度．太字と下線はそれぞれ最も良い値，2 番目に良い値を示す．括弧内の値は，出力に gold answer が含まれる割合を意味する has_answer の値を示す．灰色の数字は，与えられた設定で結果が変わらないため，上の行からコピーしたものである．$^{\diamond}$: KILT 論文 [21] で示されている値を引用．注：BART-large と RAG モデルは KILT で fine-tune 済みである．

3.3　文書検索器

文書検索器として，sparse 検索器と dense 検索器を使用する．Sparse 検索器として，Pyserini [24] の unigram **BM25** [25] を使用する[7]．Dense 検索器として，Massive Text Embedding Benchmark (MTEB) [26] のリーダーボード上で 2023 年 7 月時点で Retrieval タスクにおける精度が高いモデルを使用する（付録の表 5 を参照）．具体的には，e5-large-v2[8] (**e5**) [27] と multilingual-e5-large[9] (**m-e5**) を使用する．

検索結果を参照して回答する *open-book* 設定では，クエリとの関連度が最も高い上位 5 つの文書を取得して回答に使用する．検索性能の指標として，page-level R-precision [28] を用いる．これは，検索された上位 R の記事のうち，R 個の gold ページが含まれる割合を意味する．FEVER と HotPotQA（マルチホップデータセット）を除き，R-Precision は Precision@1 と等価である．

3.4　LLM

R-LLM で使用する LLM は，プロンプトで与えられる指示を理解し，適切な応答を生成しなければならないため，instruction tuning がなされた LLM を使用する．このため本稿では，Llama-2-chat [29] の 13B (**Llama2-13B**) と 70B (**Llama2-70B**) モデル，および WizardVicunaLM-13B[10] (**W-Vicuna-13B**) [30] を用いる．最適な性能と再現性を得るために，LLM の温度パラメータは 0 に設定する．

3.5　プロンプト

実験に用いるプロンプトテンプレートは，KILT の各データセットに対して，学習セットを用いて人手で設計する．RALLE では，自然言語以外の形式である Python の f-strings と eval 関数を用いたプロンプトテンプレートが使用でき，より柔軟なプロンプトテンプレートの設計を可能である．実験に使用したプロンプトテンプレートの一部を付録 A に示す．

4　評価結果

本章では，RALLE を用いて開発した R-LLM の評価結果について述べる．

4.1　下流タスクの性能

KILT ベンチマークにおける下流タスク性能を表 1 にまとめる．我々が構築した R-LLM (e5 + Llama2-70B) は，RAG モデルと異なり KILT で fine-tune されていないにもかかわらず，HoPo と TQA データセットにおいて RAG モデルの精度を上回った．また，我々が構築した R-LLM はこれ以外のデータセットでも許容可能な精度レベルを示していることから，

7）デフォルトのパラメータである $k_1 = 0.9$（単語頻度スケーリング）と $b = 0.4$（文書長正規化）を使用．
8）https://huggingface.co/intfloat/e5-large-v2
9）https://huggingface.co/intfloat/multilingual-e5-large
10）https://huggingface.co/junelee/wizard-vicuna-13b

	Fact Check.	Entity Linking			Slot Filling		Open Domain QA				Dial.	
Dataset	**FEV**	**AY2**	**WnWi**	**WnCw**	**T-REx**	**zsRE**	**NQ**	**HoPo**	**TQA**	**ELI5**	**WoW**	Avg.
Model	R-Precision											
RAG$^{\diamond}$	63.5	**77.4**	49.0	46.7	29.3	65.4	60.3	30.8	49.3	**16.4**	46.7	48.6
BM25	52.1	17.7	20.6	15.3	34.0	57.7	26.3	41.3	31.7	6.8	28.8	30.2
m-e5 (Flat)	81.7	41.8	45.8	41.6	**47.1**	81.4	63.0	54.0	**56.1**	11.9	**57.9**	52.9
m-e5 (HNSW)	57.0	2.0	0.1	1.3	23.3	45.5	50.7	28.4	42.5	10.0	52.8	28.5
e5 (Flat)	**82.0**	51.6	**51.6**	**49.2**	45.3	**81.9**	**65.2**	**54.3**	**56.1**	12.9	56.8	**55.2**
e5 (HNSW)	67.9	38.9	42.3	40.5	23.1	53.0	60.3	34.9	50.4	10.2	54.5	43.3
e5 (DiskANN)	78.8	44.7	47.8	46.0	37.1	74.5	64.9	49.1	55.4	12.9	56.6	51.6

表 2　KILT ベンチマークの開発セットにおける検索性能．*Avg.* は，各データセットにおける検索精度のマクロ平均．太字は最良の値を示す．$^{\diamond}$: KILT 論文 [21] で示されている値を引用．

Retrieval			
Model	Avg. R-Prec	Memory	sec/Q
BM25	30.2	-	0.121
e5 (Flat)	55.2	84.8 GB	0.169
e5 (HNSW)	43.3	90.4 GB	0.008
e5 (DiskANN)	51.6	10.9 GB	0.022
Completion in the Closed-Book Setting			sec/Q
Llama-70B			6.727
Retrieval + Generation			sec/Q
BM25 + Llama2-70B			3.637
e5 + Llama2-70B			3.793
e5 (DiskANN) + Llama2-70B			3.628

表 3　1 問あたりの実行時間 (sec/Q). *Memory* は，検索時の最大の DRAM 使用量を示す．

本研究で用いた LLM が検索結果を読解するための能力をある程度有することが示唆される．

さらに表 1 から，**検索を用いた生成**（ELI5 を除く），LLM **の大規模化**（FEV と T-REx を除く），**参照する関連文書数の増加**（NQ，HoPo，TQA，WoW を除く）が，下流タスクの性能向上に寄与しうることが示唆される．しかし，これらの傾向の例外が数多く見られること，また has_answer の値に比べて精度が低い場合があること（FEV，T-REx，NQ，TQA など）から，プロンプトや推論チェーンの改良，ないし読解に適した LLM の構築など，生成における改善の必要性が示唆される．いずれにせよ，R-LLM の定量的な評価結果は，構築した R-LLM の改善点の特定につながると考えられる．

4.2　検索精度

表 2 は KILT 開発セットにおける検索精度を示す．KILT における検索精度の平均は e5 (Flat index) が最も高いことが分かる．また，e5 の検索精度は RAG モデルより高いにもかかわらず，e5 を採用した R-LLM の下流性能は RAG モデルに及んでいない（表 1）．このことは，構築した R-LLM の生成の部分において改善の余地があることを示唆しており，この考察は 4.1 節の考察と一貫している．

4.3　実行時間の解析

表 3 に，実行時間と精度のトレードオフを示す．R-LLM の実行時間は LLM による生成が大部分を占めること，また HNSW や DiskANN などの近似最近傍探索アルゴリズムを使用することで，検索時間を短縮できる一方で検索精度は低下することが分かる．このため，近似最近傍探索による実行時間短縮の効果は限定的であると思われるが，DiskANN はメモリ使用量が比較的少ないことから，利用可能な DRAM 容量に制限がある場合には DiskANN などの技術が実用的な解決策となることも考えられる．速度と精度のバランスはアプリケーションの要件によって異なるが，このような様々な実験設定での評価結果は R-LLM 開発に貢献すると考えられる．

5　おわりに

本稿では，R-LLM を開発・評価するためのフレームワークである RALLE を提案した．RALLE を用いて R-LLM を構築し，KILT ベンチマークで評価したところ，我々が構築した R-LLM は KILT ベンチマークデータセットで fine-tune されていないにもかかわらず，ある程度の回答精度を示すことが判明した．RALLE が今後の検索拡張生成の研究開発に貢献することを期待する．

参考文献

[1] Yasuto Hoshi, Daisuke Miyashita, Youyang Ng, Kento Tatsuno, Yasuhiro Morioka, Osamu Torii, and Jun Deguchi. RaLLe: A Framework for Developing and Evaluating Retrieval-Augmented Large Language Models. In **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**, pp. 52–69, Singapore, December 2023. Association for Computational Linguistics.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, et al. Language Models are Few-Shot Learners. In **Advances in Neural Information Processing Systems**, Vol. 33, pp. 1877–1901, 2020.

[3] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, et al. PaLM: Scaling Language Modeling with Pathways. **Journal of Machine Learning Research**, Vol. 24, No. 240, pp. 1–113, 2023.

[4] OpenAI. GPT-4 Technical Report. **arXiv preprint arXiv:2303.08774**, Vol. abs/2303.08774, , 2023.

[5] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. **arXiv preprint arXiv:2302.04023**, 2023.

[6] Ali Borji. A categorical archive of ChatGPT failures. **arXiv preprint arXiv:2302.03494**, 2023.

[7] Adam Liska, Tomas Kocisky, Elena Gribovskaya, et al. StreamingQA: A Benchmark for Adaptation to New Knowledge over Time in Question Answering Models. In **Proceedings of the 39th International Conference on Machine Learning**, Vol. 162 of **Proceedings of Machine Learning Research**, pp. 13604–13622. PMLR, 17–23 Jul 2022.

[8] Benjamin Heinzerling and Kentaro Inui. Language Models as Knowledge Bases: On Entity Representations, Storage Capacity, and Paraphrased Queries. In **Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume**, pp. 1772–1791, Online, April 2021. Association for Computational Linguistics.

[9] Grégoire Mialon, Roberto Dessi, Maria Lomeli, Christoforos Nalmpantis, Ramakanth Pasunuru, Roberta Raileanu, et al. Augmented Language Models: a Survey. **Transactions on Machine Learning Research**, 2023. Survey Certification.

[10] Youyang Ng, Daisuke Miyashita, Yasuto Hoshi, Yasuhiro Morioka, Osamu Torii, Tomoya Kodama, and Jun Deguchi. SimplyRetrieve: A Private and Lightweight Retrieval-Centric Generative AI Tool. **arXiv preprint arXiv:2308.03983**, 2023.

[11] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. RePlug: Retrieval-augmented black-box language models. **arXiv preprint arXiv:2301.12652**, 2023.

[12] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, et al. WebGPT: Browser-assisted question-answering with human feedback. **arXiv preprint arXiv:2112.09332**, 2021.

[13] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval Augmented Language Model Pre-Training. In **Proceedings of the 37th International Conference on Machine Learning**, Vol. 119 of **Proceedings of Machine Learning Research**, pp. 3929–3938. PMLR, 13–18 Jul 2020.

[14] Harrison Chase. LangChain, 2023. https://langchain.com/.

[15] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. **IEEE Transactions on Big Data**, Vol. 7, No. 3, pp. 535–547, 2019.

[16] Yu A. Malkov and D. A. Yashunin. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. **IEEE Trans. Pattern Anal. Mach. Intell.**, Vol. 42, No. 4, pp. 824–836, apr 2020.

[17] Suhas Jayaram Subramanya, Fnu Devvrit, Harsha Vardhan Simhadri, Ravishankar Krishnawamy, and Rohan Kadekodi. DiskANN: Fast Accurate Billion-point Nearest Neighbor Search on a Single Node. In **Advances in Neural Information Processing Systems**, Vol. 32. Curran Associates, Inc., 2019.

[18] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. Query Rewriting in Retrieval-Augmented Large Language Models. In **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, pp. 5303–5315, Singapore, December 2023. Association for Computational Linguistics.

[19] Abubakar Abid, Ali Abdalla, Ali Abid, Dawood Khan, Abdulrahman Alfozan, and James Zou. Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild. **arXiv preprint arXiv:1906.02569**, 2019.

[20] LF Projects. MLflow – a platform for the machine learning lifecycle, 2023. https://mlflow.org/.

[21] Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, et al. KILT: a Benchmark for Knowledge Intensive Language Tasks. In **Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 2523–2544, Online, June 2021. Association for Computational Linguistics.

[22] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 7871–7880, Online, July 2020. Association for Computational Linguistics.

[23] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In **Advances in Neural Information Processing Systems**, Vol. 33, pp. 9459–9474. Curran Associates, Inc., 2020.

[24] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, et al. 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 2021)**, pp. 2356–2362, 2021.

[25] Stephen Robertson and Hugo Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. **Found. Trends Inf. Retr.**, Vol. 3, No. 4, pp. 333–389, apr 2009.

[26] Niklas Muennighoff, Nouamane Tazi, et al. MTEB: Massive Text Embedding Benchmark. In **Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics**, pp. 2014–2037, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[27] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, et al. Text embeddings by weakly-supervised contrastive pre-training. **arXiv preprint arXiv:2212.03533**, 2022.

[28] Nick Craswell. R-Precision. pp. 1–1, 2016.

[29] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, et al. LLaMA 2: Open Foundation and Fine-Tuned Chat Models. **arXiv preprint arXiv:2307.09288**, 2023.

[30] June Lee. WizardVicunaLM, 2023. https://github.com/melodysdreamj/WizardVicunaLM.

[31] Ori Ram, Liat Bezalel, Adi Zicher, Yonatan Belinkov, Jonathan Berant, and Amir Globerson. What Are You Token About? Dense Retrieval as Distributions Over the Vocabulary. In **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 2481–2498, Toronto, Canada, July 2023. Association for Computational Linguistics.

Open-book	*Closed-book*
AY2, WnWi, and WnCw f-strings ◉ eval	

Action 1: Retriever
`'What is "'+ '{}'.format(question).split('[START_ENT]')[1].split('[END_ENT]')[0][1:-1] + '"?'`
Action 2: Identity
`'{}'.format(wiki_id_title[0]).split(';')[0].split(', ')[0]`
	Action 1: LLM
	`'What is the most relevant Wikipedia title to the entity "'+ '{}'.format(question).split('[START_ENT]')[1].split('[END_ENT]')[0] + '" in the context of "'+ '{}'.format(question).split('[START_ENT]')[0][-100:] + '{}'.format(question).split('[START_ENT]')[1].split('[END_ENT]')[0] + '{}'.format(question).split('[END_ENT]')[1][:100] + '''..."?\n\nPlease answer only the Wikipedia title.\n\nAnswer: '''`

T-REx	

Action 1: Retriever (`f-strings`)
`{question}`
Action 2: LLM (`eval()`)
`'''Referring to the following document, answer "what is the '''+ '{}'.format(question).split('[SEP]')[1] + 'of '+ '{}'.format(question).split('[SEP]')[0] + '''?" in 5 words or less.\n\n'''+ '{}'.format(response[0]) + '''\n\n'''+ '{}'.format(question).split('[SEP]')[1] + ': '`
	Action 1: LLM (`eval()`)
	`'What is the '+ '"'+ '{}'.format(question).split('[SEP] ')[1] + '" of "'+ '{}'.format(question).split('[SEP]')[0] + '"'+ '''in 5 words or less?\n\n'''+ '{}'.format(question).split('[SEP] ')[1] + ': '`

NQ, HoPo, and TQA ◉ f-strings ○ eval	

Action 1: Retriever
`{question}`
Action 2: LLM
`Referring to the following document, answer "{question}?" in 5 words or less. ↩`
`↩`
`{response[0]} ↩`
`↩`
`Answer:`
	Action 1: LLM
	`Answer '{question}?' in 5 words or less. ↩`
	`↩`
	`Answer:`

表 4 本稿の評価実験で使用したプロンプトテンプレートの一部．左フック矢印 ↩ は，改行を意味する．より柔軟なプロンプトテンプレート開発のために，RALLE は Python の f-strings と eval 関数をサポートしている．

Model	dim.	max len.	MTEB Retrieval
BM25	-	-	42.3♠
m-e5	1,024	514	51.43
e5	1,024	512	50.56

表 5 実験に用いた検索器．Dense 検索器の埋め込みの次元数を *dim.*，モデルへの入力の最大系列長を *max len.* で示す．♠: 文献 [31] より引用．

A 評価で使用したプロンプトテンプレート

表 4 に，評価実験で使用したプロンプトテンプレートの一部を示す．詳細は 3.5 節を参照．*Retriever* はプロンプトを検索クエリとして関連文書を検索し，*LLM* はプロンプトを入力として出力を生成し，*Identity* はプロンプトの文字列をそのまま出力する．

B 実験に用いた検索器

表 5 に，実験に使用した文書検索器をまとめる．MTEB [26] の Retrieval タスクにおける評価指標は nDCG@10 であり，leaderboard[11] 上で公開されている値を示す．

11) https://huggingface.co/spaces/mteb/leaderboard