ITERKEY: LLM を用いた反復的キーワード生成による検索拡張生成の最適化

林 和樹 [1]　上垣外 英剛 [1]　幸田 慎也 [2]　渡辺 太郎 [1]
[1] 奈良先端科学技術大学院大学　[2]TDSE 株式会社
{hayashi.kazuki.hl4, kamigaito.h, taro}@is.naist.jp

概要

本論文では，LLM を活用し，疎検索を用いた検索拡張生成（RAG）を最適化する，反復的キーワード生成手法 IterKey を提案する．IterKey は，キーワード生成，回答生成，回答検証の 3 つの構成から成り，すべて LLM によって実行し，RAG の処理全体の最適化を目指す．この手法により，質問応答タスクにおいて，検索なしの手法や BM25 を用いた RAG と比較して精度が 5 %から 20 %向上し，密検索モデルを用いた RAG や先行研究と同等の性能を達成した．IterKey は，価値あるキーワードを生成しながら，解釈可能性を維持し，LLM による反復的なキーワード洗練と自己検証を通じて RAG 全体を最適化する可能性を示す新たな疎検索手法である．

1　はじめに

大規模言語モデル（LLM）[1, 2] は自然言語処理タスクにおいて高い性能を発揮する一方で，幻覚（hallucinations），知識の陳腐化，およびマルチホップ推論が必要となる複雑なクエリへの対応が難しいという課題がある [3, 4, 5]．特に，必要な情報がモデル内部に十分に記憶されていない知識集約型タスクにおいて，これらの問題は顕著である [6, 7]．

こうした課題に対応する手法として，検索拡張生成（RAG）がある．RAG は外部知識を統合することで，生成される応答の精度や関連性を向上させ，質問応答タスクで高い有効性を示している [8, 9, 10, 11, 12]．近年，クエリ拡張や検索モデルの改良といった要素技術の進展により，RAG の性能はさらに向上している．一方で，RAG の効果を最大化するには検索と生成の相互作用が重要だが，現状では，多くの手法が検索と生成の相互作用を十分に考慮しておらず，応答の一貫性や両者の連携に課題が残っている [13, 14, 15, 16, 17, 18, 19]．

本研究では **IterKey**（**Iter**ative **Key**word Generation with LLMs）を提案する．IterKey は，LLM を活用して RAG の処理を最適化する新しい手法であり，キーワード生成，回答生成，検証の 3 段階を通じて検索と生成を最適化し連携を強化する．4 つの質問応答データセットで実験を行った結果，IterKey は検索なしの手法や BM25 を用いた RAG と比較して精度が 5 %から 20 %向上し，密検索モデルを用いた RAG や先行研究と同等の性能を示した．これにより，IterKey が RAG の最適化に効果的であり，LLM が反復的なキーワード洗練と自己検証を通じて RAG を改善できる可能性が示された．

2　提案手法

BM25 [20] のような疎検索アルゴリズムは，キーワードの出現頻度や重要度に基づいて文書をランキングし，大規模データセットを効率的に処理する．このアルゴリズムは，検索結果に影響を与えたキーワードが明確で，可視性と解釈性の点で優れている．一方で，疎検索はクエリの細かなニュアンスや暗黙的な情報を捉える精度が低く，密検索手法と比べて性能面での制約がある [21, 22, 23]．IterKey は，LLM の最新技術を活用し，検索と生成を統合的に最適化することで，RAG システム全体の精度を向上させる手法である．LLM の自己検証機能 [24] を利用し，キーワード生成，文書検索，回答生成，回答検証の手順を反復的に改善することで，疎検索の制約を克服し，より正確で解釈可能な結果を実現する．各ステップの設計の背景や動機については，図 1 に示し，プロンプトの詳細は表 1 に記載する．

Step 1: キーワード生成　ユーザーのクエリ q に対し，LLM はクエリの回答に関連する文書を検索するための重要なキーワード集合 $\mathcal{K}^0$ を生成し，疎検索で捉えにくい意図やニュアンスを補完する．

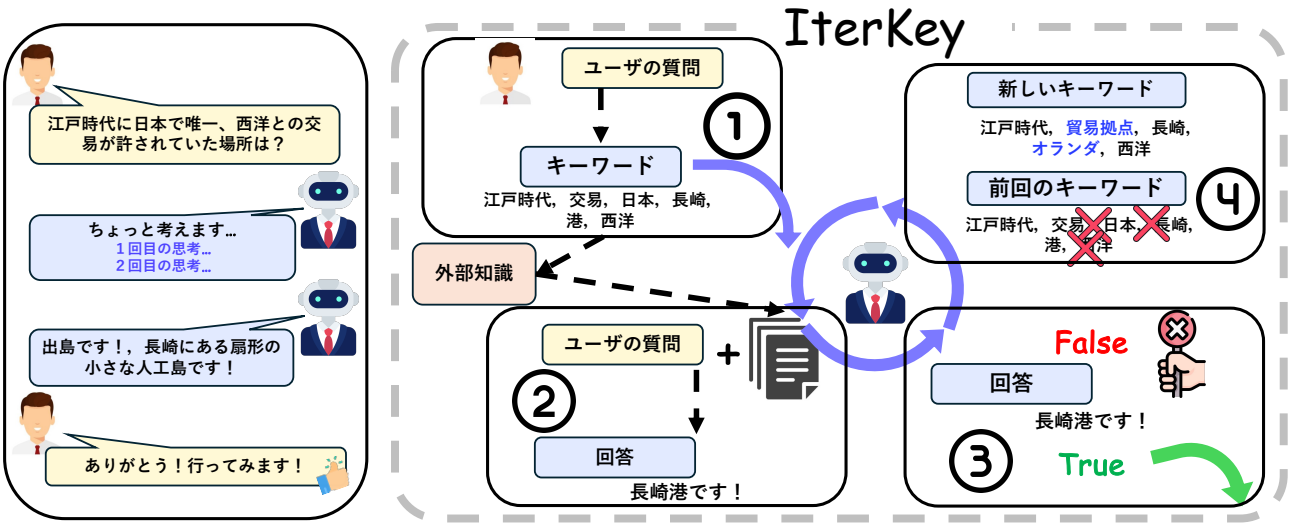

図 1 IterKey プロセスの概要：LLM を用いた反復的なキーワード生成，回答生成，および検証を行うプロセス．正しい回答が得られるまでキーワードと回答を洗練し，RAG の精度を向上させる．

表 1 IterKey で使用されるプロンプトは，キーワード生成と回答検証の反復的な改善に用いられる．

Step	Prompt
Step 1: Initial Keyword Generation	**System:** You are an assistant that generates keywords for information retrieval. **User:** Generate a list of important keywords related to the **query** { *q* }. Focus on keywords that are relevant and likely to appear in documents for BM25 search in the RAG framework. Output the keywords as: ["keyword1", "keyword2", "keyword3", ...]. Separate each keyword with a comma and do not include any additional text.
Step 2: Answer Generation using RAG	**System:** You are an assistant that generates answers based on retrieved documents. **User:** Here is a question that you need to answer: **Query:** { **q** } Below are some documents that may contain information relevant to the question. Consider the information in these documents while combining it with your own knowledge to answer the question accurately. Documents: { *D* } Provide a clear and concise answer. Do not include any additional text.
Step 3: Answer Validation	**System:** You are an assistant that validates whether the provided answer is correct. **User:** Is the following answer correct? **Query:** { **q** } Answer: { *a* } Respond 'True' or 'False'. Do not provide any additional explanation or text.
Step 4: Keyword Regeneration	**System:** You refine keywords to improve document retrieval for BM25 search in the RAG framework. **User:** Refine the keyword selection process to improve the accuracy of retrieving documents with the correct answer. **Query:** { **q** } Previous Keywords: { *K* } Provide the refined list of keywords in this format: ["keyword1", "keyword2", ...]. Separate each keyword with a comma and do not include any additional text.

Step 2: 回答生成 元のクエリ q は，生成されたキーワード $\mathcal{K}^i$ を加えて拡張クエリ $q + \mathcal{K}^i$ を形成する．この拡張クエリを使用して，BM25 ベースの検索システムから上位 k 件の文書集合 $\mathcal{D}^i = \{d_1, d_2, \ldots, d_k\}$ を取得する．取得した文書を基に，LLM は回答 a^i を生成する．

Step 3: 回答検証 LLM は生成された回答の正誤を検証し，RAG が正確に実行されているかを確認する．LLM は「True」または「False」で応答し，「True」の場合は回答 a^i が最終結果として返され，処理は終了する．「False」の場合は，キーワード生成と検索ステップが再調整する．この二値応答により，手順の明確化と反復処理の自動化が実現される．

Step 4: キーワード再生成 回答検証で「False」と判定された場合,LLM は新たなキーワード集合 $\mathcal{K}^{i+1}$ を生成し, 検索精度を向上させるための再検索を行う．この際, 元のクエリ q と前回生成されたキーワード $\mathcal{K}^i$ が入力として利用する．生成されたキーワード $\mathcal{K}^{i+1}$ に基づき，新たな拡張クエリ $q + \mathcal{K}^{i+1}$ が形成され，ステップ 2 から 4 が繰り返される．この処理は，「True」が返されるか，最大反復回数 N に達するまで繰り返す．

3 実験設定

本研究では，検索モデルとして BM25 [20] を採用し，BM25S [25][1)] を用いて実装した．また，密検索モデルとして E5 [23] を導入し，疎検索モデルと同条件で性能を比較した．LLM は Llama-3.1-(8B, 70B) [26]，Gemma-2 [27]，Phi-3.5-mini [2] の 4 種類を使用した．評価は 4 つの質問応答タスクで IterKey を評価した．使用したデータセットは，Natural Questions [28]，EntityQA [29]，WebQA [30]，および HotpotQA [31] である．先行研究 [32, 33, 16] に従い，各データセットから 500 件をランダムにサンプリングして評価を行った．評価はゼロショット環境で実施した．生成された回答は Exact Match (EM)[34] で評価した．

1) https://github.com/xhluca/bm25s

表 2　ŔAG (BM25) および ŔAG (E5) は，元のクエリに基づき BM25 および E5 を使用して 1 回の検索を適用する．ĺTRG (E5) は，5 回のクエリ拡張を伴う，E5(最適な密検索手法) を用いた手法である．ĺterKey (BM25) は，最大 5 回の検索を行って最適化する．本表では，下線付きの値 は各モデルのタスクごとの最良の性能を示し，**太字の値** はすべての手法の中で各タスクの最高精度を示す．

Method	Model	Entity	Hotpot	Natural	Web
Vanillla	Llama-3.1 (8B)	33.6	31.2	40.6	53.4
	Llama-3.1 (70B)	45.2	41.4	46.0	54.0
	Gemma-2	10.6	11.6	9.2	20.8
	Phi-3.5-mini	24.6	25.4	25.8	44.0
RAG (BM25)	Llama-3.1 (8B)	54.0	47.0	44.8	51.4
	Llama-3.1 (70B)	54.6	46.2	43.4	47.4
	Gemma-2	47.9	39.6	33.2	41.6
	Phi-3.5-mini	48.2	42.2	32.6	40.2
RAG (E5)	Llama-3.1 (8B)	52.9	47.7	49.6	48.2
	Llama-3.1 (70B)	57.0	51.0	49.4	48.8
	Gemma-2	52.2	40.8	41.5	41.8
	Phi-3.5-mini	50.2	44.6	37.0	41.4
ITRG (E5)	Llama-3.1 (8B)	60.6	53.4	53.6	**56.2**
	Llama-3.1 (70B)	60.7	52.9	53.3	51.6
	Gemma-2	54.2	47.6	47.4	48.5
	Phi-3.5-mini	54.3	47.1	36.2	44.6
IterKey (BM25)	Llama-3.1 (8B)	61.0	52.3	51.6	52.2
	Llama-3.1 (70B)	**62.1**	**54.5**	**54.7**	56.0
	Gemma-2	34.2	24.6	33.7	33.8
	Phi-3.5-mini	49.6	43.9	34.8	41.4

これは，正規化後の回答が参照解答と一致する場合に正解と見なす手法である．正規化では，大文字を小文字に変換し，冠詞や句読点の除去，および空白の統一を行った．また，クエリ拡張の精度を評価するため，取得した文書に正解が含まれているかを確認し，リコールを測定し検索性能を詳細に分析した．すべてのデータセットで，2018 年 12 月版の Wikipedia ダンプ [12] を検索コーパスとして使用した．これにより，先行研究 [16] の設定を再現した．

また，検索と生成を反復的に行うアプローチのベースラインとして，密検索モデルを用いた反復的クエリ精緻化を特徴とする ITRG (Iterative Retrieval-Generation Synergy) [16] を選択した．この比較を通じて，本研究の新規性と貢献を明確化した．

4　結果

表 2 は，IterKey がすべてのモデルでベースラインを一貫して上回り，10%から 20%の精度向上を達成した．特に，Llama-3.1 の 8B および 70B モデルでは顕著な改善が見られ，8B モデルは 70B モデルと同等の精度を示した．また，BM25 ベースの RAG と比較しても，Llama-3.1 モデルでは IterKey の適用により 5%から 10%の精度向上が確認された．さらに，Llama-3.1 モデルでは E5 ベースの RAG を上回り，70B モデルは 3 つのタスクで最高精度を達成した．密検索リトリーバーを用いた反復精緻化手法である ITRG とも同等以上の性能を示した．一方で，IterKey の有効性はモデルによって異なり，ITRG が Gemma-2 および Phi-3.5-mini で精度を安定して向上させたのに対し，IterKey では精度向上が見られず，E5 ベースの RAG を下回った．これらの結果から，IterKey は多くのモデルで効果を発揮するが，反復的なキーワード生成および自己検証の効果はモデルの性能に依存することが明らかとなった．

5　分析

IterKey におけるリコール率および QA 性能　図 2 は，Entity QA タスクにおける 3 手法と 4 モデルのリコール率（%）を，5 回の反復平均で示している．比較のため，BM25 および E5（最左列）は元のクエリを用いた単一の検索によるリコール率を示す．リコール率は，取得文書に正解が含まれる割合を表す．IterKey（BM25）は，BM25 と比較してリコール率が約 20%向上し，Llama-3.1 モデルでは E5 と比較してトップ 5 のリコール率が約 10%向上した．一方，Gemma および Phi は BM25 を上回るが，リコール率は E5 に及ばない．これは，効果的な検索用キーワードの生成がモデルの性能に依存することを示唆している．Gemma および Phi はトップ 3 で約 60%のリコール率を達成しているが，表 2 に示すように QA 精度は 20〜30%低い．検証ステップが取得候補の正誤を適切に判定できず，全体の精度を下げていることが原因と考えられる．IterKey（BM25）と IterKey（E5）の比較では，トップ 1 のリコール率は同等だが，トップ 5 で IterKey（BM25）が約 10%上回り，IterKey が疎検索モデルに特に適していることを示している．最後に，表 2 に基づき IterKey（BM25）と ITRG（E5）を比較すると，ITRG（E5）はリコール率で優れるが，IterKey（BM25）は Llama-70 モデルで 3 タスクにおいて最高の QA 精度を達成した．ITRG は 5 回の固定反復後に最終回答を出力するが，IterKey（BM25）は正解が見つかった時点で検証ステップにより早期終了する．この結果は，特に信頼性の高い回答検証が可能なモデルで，我々の手法の検証のステップの有効性を示している．

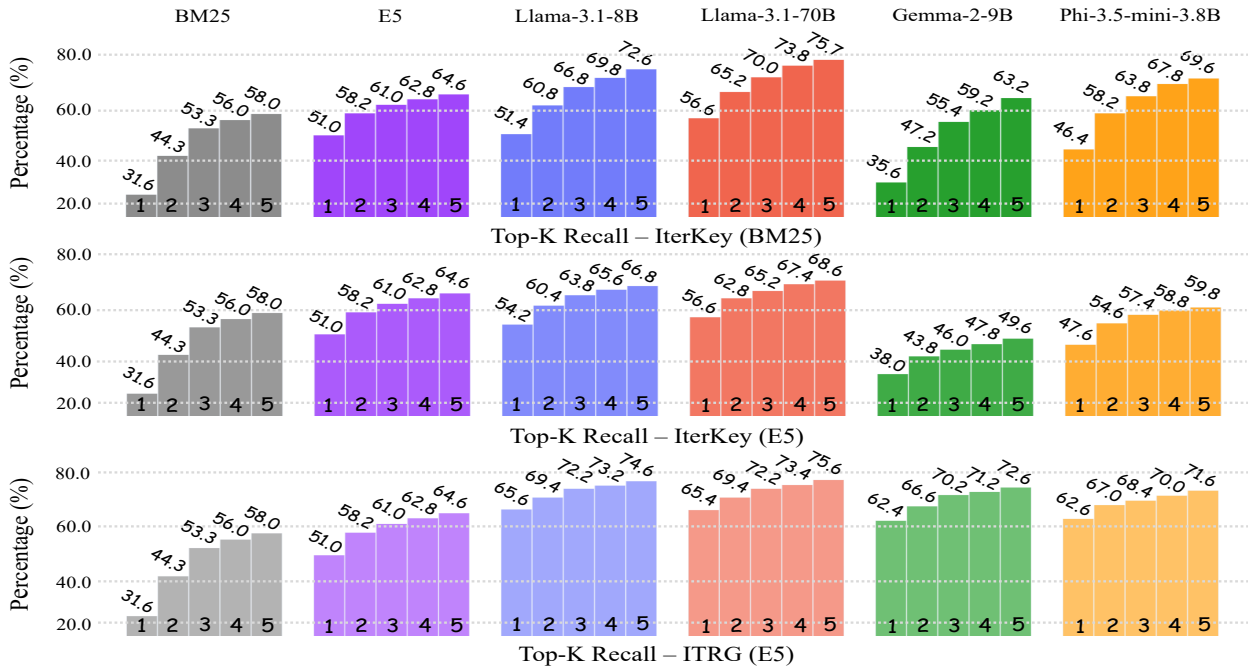

図 2　Entity QA タスクにおける各モデルの Top-K 検索リコール率を示す．‘BM25’ と ‘E5’ は単一の検索ステップを使用するベースライン手法である．提案手法 ‘IterKey (BM25)’ は,‘IterKey (E5)’ および ‘ITRG (E5)’ と比較される．リコール率は 5 回の反復の平均で算出されている．

表 3　IterKey のモデルごとにおける平均反復回数．

Model	Size (B)	Entity	Hotpot	Natural	Web
Llama-3.1	8B	1.27	1.38	1.35	1.36
Llama-3.1	70B	1.15	1.20	1.23	1.10
Gemma-2	9B	1.33	1.36	1.34	1.37
Phi-3.5-mini	3.8B	1.38	1.32	1.28	1.29

計算コスト　IterKey は精度を向上させるが，追加の計算ステップを必要とする．表 4 は，Entity QA データセットにおける Llama-3.1-8B を用いた実行時間を比較したものである．公平な比較を行うため，IterKey および ITRG の両方で検索には E5 を使用している．IterKey はクエリ拡張と回答検証の処理が追加されるものの，ITRG と比較して実行時間が 400 秒短縮された．表 3 は，すべての QA タスクおよびモデルにおける IterKey の平均反復回数を示している．平均反復回数は 1.5 未満であり，各反復はキーワード生成，回答生成，検証の 3 回の LLM 呼び出しで構成される．Llama-3.1-70B などの大規模モデルは，キーワード生成と検証の質が高いため，反復回数が少なく収束するなど，より効率的に動作する．これらの結果から，IterKey は疎検索モデルベースの RAG や他の反復的クエリ精緻化手法と同等の性能を維持しつつ，高い効率性と精度を両立していることが示される．

表 4　Llama-3.1 (8B) を使用した Entity QA の実行時間（秒）．推論は NVIDIA RTX 6000 Ada GPU，BM25 検索は Intel Xeon Platinum 8160 CPU で実行した．公平な比較のため，IterKey と ITRG の検索には E5 を使用した．

Step	RAG(BM25)	RAG(E5)	ITRG	IterKey
Query Expansion	–	–	–	663.8
Retrieval	222.4	52.7	437.5	167.3
Answer Generation	317.7	314.6	1694.3	523.5
Answer Validation	–	–	–	360.8
All	540.1	367.3	2131.8	1713.8

6　おわりに

本研究では，大規模言語モデル（LLM）を活用して RAG の処理全体を最適化する反復的キーワード生成手法 IterKey を提案した．IterKey は，LLM によるキーワード生成，回答生成，および検証を行うことで，検索と生成の連携を強化し，より正確な応答を実現する．4 つの質問応答タスクを用いた実験では，IterKey が BM25 を用いた RAG と比較して 5%から 20%の精度向上を達成し，密検索を用いた RAG や既存の反復的クエリ精緻化手法と同等の性能を示した．これらの結果は，LLM が反復的な洗練と検証を通じて，検索と生成の結び付きを強化し，RAG の処理を最適化できる可能性を示した．

参考文献

[1] OpenAI. Gpt-4 technical report. **ArXiv**, Vol. abs/2303.08774, , 2023.

[2] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024.

[3] Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, **Proceedings of the 40th International Conference on Machine Learning**, Vol. 202 of **Proceedings of Machine Learning Research**, pp. 15696–15707. PMLR, 23–29 Jul 2023.

[4] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. Siren's song in the ai ocean: A survey on hallucination in large language models, 2023.

[5] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024.

[6] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 6086–6096, Florence, Italy, July 2019. Association for Computational Linguistics.

[7] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing**, pp. 93–104, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[8] 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 **Proceedings of the 34th International Conference on Neural Information Processing Systems**, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[9] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, **Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume**, pp. 874–880, Online, April 2021. Association for Computational Linguistics.

[10] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, **Findings of the Association for Computational Linguistics: EMNLP 2021**, pp. 3784–3803, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[11] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. **Transactions of the Association for Computational Linguistics**, Vol. 11, pp. 1316–1331, 2023.

[12] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: few-shot learning with retrieval augmented language models. **J. Mach. Learn. Res.**, Vol. 24, No. 1, March 2024.

[13] Zhengbao Jiang, Luyu Gao, Zhiruo Wang, Jun Araki, Haibo Ding, Jamie Callan, and Graham Neubig. Retrieval as attention: End-to-end learning of retrieval and reading within a single transformer. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, **Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing**, pp. 2336–2349, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[14] Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. Retrieve anything to augment large language models, 2023.

[15] Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Findings of the Association for Computational Linguistics: EMNLP 2023**, pp. 9248–9274, Singapore, December 2023. Association for Computational Linguistics.

[16] Zhangyin Feng, Xiaocheng Feng, Dezhi Zhao, Maojin Yang, and Bing Qin. Retrieval-generation synergy augmented large language models. In **ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, pp. 11661–11665, 2024.

[17] Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. Generate rather than retrieve: Large language models are strong context generators. In **The Eleventh International Conference on Learning Representations**, 2023.

[18] Xin Cheng, Di Luo, Xiuying Chen, Lemao Liu, Dongyan Zhao, and Rui Yan. Lift yourself up: Retrieval-augmented text generation with self-memory. In **Thirty-seventh Conference on Neural Information Processing Systems**, 2023.

[19] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: Retrieval-augmented black-box language models. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, **Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)**, pp. 8371–8384, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[20] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. **Foundations and Trends in Information Retrieval**, Vol. 3, pp. 333–389, 01 2009.

[21] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning, 2022.

[22] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2023.

[23] 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 preprint arXiv:2212.03533**, 2022.

[24] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 9426–9439, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[25] Xing Han Lù. Bm25s: Orders of magnitude faster lexical search via eager sparse scoring, 2024.

[26] Abhimanyu Dubey. The llama 3 herd of models, 2024.

[27] Gemma Team. Gemma: Open models based on gemini research and technology, 2024.

[28] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. **Transactions of the Association for Computational Linguistics**, Vol. 7, pp. 452–466, 2019.

[29] Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. Entity-relation extraction as multi-turn question answering. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 1340–1350, Florence, Italy, July 2019. Association for Computational Linguistics.

[30] Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. Webqa: Multihop and multimodal qa. In **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**, pp. 16495–16504, June 2022.

[31] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing**, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[32] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 10014–10037, Toronto, Canada, July 2023. Association for Computational Linguistics.

[33] Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, pp. 7969–7992, Singapore, December 2023. Association for Computational Linguistics.

[34] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras, editors, **Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing**, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.

表 5 IterKey の性能比較（BM25 と密検索モデル E5）.

Method	Model	Entity		Hotpot		Natural		Web	
BM25	Llama-3.1 (8B)	61.0		52.3		51.6		52.2	
	Llama-3.1 (70B)	62.1		54.5		54.7		56.0	
	Gemma-2	34.2		24.6		33.7		33.8	
	Phi-3.5-mini	49.6		43.9		34.8		41.4	
E5	Llama-3.1 (8B)	-0.4	60.6	+1.4	53.7	-0.4	51.2	-2.1	50.1
	Llama-3.1 (70B)	-0.2	61.9	-0.7	53.8	-1.1	53.6	-3.0	53.0
	Gemma-2	-1.1	33.1	-0.4	24.2		33.7	-2.9	30.9
	Phi-3.5	-2.2	47.4	+0.6	44.5	-0.9	33.9	-0.2	41.2

A 実験設定の詳細

本研究では，複数のモデル間で性能を公平に比較するため，すべての実験を単一の NVIDIA RTX 6000 Ada GPU 上で実施し，モデル生成には半量子化を利用した．ただし，リソースの制約により，Llama-3.1 70B モデルは 4-bit モードでロードおよび推論を行った．本研究では，キーワード生成，回答生成，および回答検証の 3 つのステップで LLM を使用する．各ステップでは，表 7 に示すように，適切な最大トークン長が設定された．性能比較のため，各モデルに同一の設定が適用された．

表 6 実験で使用した言語モデル（LM）の詳細.

Model	Params	HuggingFace Name
Llama-3.1	8B	meta-llama/Llama-3.1-8B-Instruct
Llama-3.1	70B	meta-llama/Llama-3.1-70B-Instruct
Gemma-2	9B	google/Gemma-2-9b-it
Phi-3.5-mini	3.8B	microsoft/Phi-3.5-mini-instruct

表 7 各ステップにおける最大トークン長.

STEP	Max Token Length
Keyword Generation	50
Answer Generation	50
Answer validation	30

B IterKey の構成要素の分析

キーワード生成 textscIterKey の性能に対するキーワード品質の影響を調査するため，最も高いリコール率を達成した Llama-3.1-70B モデルによって生成された高品質なキーワードを用い，他のモデルに適用して検証を行った．表 8 に示すように，すべてのモデルで精度の向上が見られ，特に Gemma-2 モデルでは約 5%の改善が確認された．これらの結果から，キーワードの品質が IterKey の性能において重要であることが明らかとなった．

回答検証 LLM の回答検証能力を評価するため，‘True’ と判定された後でも最大 5 回の反復を行い，

表 8 Llama-3.1-70B からの IterKey w/ HQ Keywords．この手法は，最も高いリコールを達成した Llama-3.1-70B の高品質なキーワードを使用し，パラメータ数が 10 分の 1 の小規模モデルに適用する．

Method	Model	Entity		Hotpot		Natural		Web	
Base	Llama-3.1 (8B)	61.0		52.3		51.6		52.2	
	Llama-3.1 (70B)	62.1		54.5		54.7		56.0	
	Gemma-2	34.2		24.6		33.7		33.8	
	Phi-3.5	49.6		43.9		34.8		41.4	
HQ Keywords	Llama-3.1 (8B)	+2.6	63.6	+2.5	54.8	+0.6	52.2	+1.5	53.7
	Llama-3.1 (70B)		62.1		54.5		54.7		56.0
	Gemma-2	+4.9	39.1	+5.6	30.2	+3.8	37.5	+4.2	38.0
	Phi-3.5	+1.2	50.8	+0.7	44.6	+3.2	38.0	+2.4	43.8

表 9 異なるモデルと設定での 4 つの QA タスクにおける精度結果．**設定の説明:** ‘Base’: 検証ステップで最初に「True」と判定された時点で反復を停止．‘VerifiedTrue’: 「True」と判定された反復の中に正解が含まれるかを確認．‘VerifiedAll’: すべての反復結果（「True」と「False」）を確認し，正解が含まれるかを評価.

Setting	Model	Entity QA		HotpotQA		Natural QA		WebQA	
Base	Llama-3.1 (8B)	61.0		52.3		51.6		52.2	
	Llama-3.1 (70B)	62.1		54.5		54.7		56.0	
	Gemma-2	34.2		24.6		33.7		33.8	
	Phi-3.5-mini	49.6		43.9		34.8		41.4	
VerifiedTrue	Llama-3.1 (8B)	+2.9	63.9	+5.2	57.5	+5.3	56.9	+6.6	58.8
	Llama-3.1 (70B)	+3.7	65.8	+4.6	59.1	+4.1	58.8	+6.6	62.6
	Gemma-2	+8.2	42.4	+8.7	33.3	+5.4	39.1	+6.2	40.0
	Phi-3.5-mini	+4.9	54.5	+3.2	47.1	+5.9	40.7	+5.4	46.8
VerifiedAll	Llama-3.1 (8B)	+5.2	66.2	+6.3	58.6	+8.5	60.1	+8.5	60.7
	Llama-3.1 (70B)	+5.6	67.7	+6.0	60.5	+9.1	63.8	+8.2	64.2
	Gemma-2	+15.2	49.4	+16.9	41.5	+10.1	43.8	+13.9	47.7
	Phi-3.5-mini	+7.0	56.6	+5.3	49.2	+9.5	44.3	+9.0	50.4

各モデルの検証性能を 3 つの設定で比較した．表 9 に示すように，‘Base’ と ‘VerifiedAll’ を比較すると，すべてのモデルで検証精度の低下が確認され，特に Gemma-2 では HotpotQA で最大 16.9%，Entity QA で 15.2%の低下が見られた．この結果は，回答検証ステップにおけるエラーの影響が大きく，特に性能が低いモデルでは検証精度の向上が課題であることを示している．また，‘Base’ と ‘VerifiedTrue’ の比較により，‘True’ と誤判定される割合（miss True rate）を測定し，‘VerifiedTrue’ と ‘VerifiedAll’ の比較では，‘False’ を ‘True’ と誤判定する割合（miss False rate）を評価した．Gemma-2 以外のモデルでは，miss True rate が miss False rate を一貫して上回り，不正確な回答を正しいと認識する傾向が見られた．これが精度向上の障壁となっている．一方，Gemma-2 では miss False rate が高く，正解を見逃す傾向が示された．これらから，回答検証エラー，特に miss True rate が多くのモデルで精度向上の課題となっており，回答検証のさらなる改良が必要であることが示された．