LeapQuery: Enhancing Multi-Hop Question Answering

Kashob Kumar Roy
University of Illinois at Urbana Champaign
Champaign, Illinois, USA
kkroy2@illinois.edu

Palvi Shroff
University of Illinois at Urbana Champaign
Champaign, Illinois, USA
pps6@illinois.edu

Shreya Matta
University of Illinois at Urbana Champaign
Champaign, Illinois, USA
matta6@illinois.edu

Taobo Liao
University of Illinois at Urbana Champaign
Champaign, Illinois, USA
taobol2@illinois.edu

ABSTRACT

Retrieval-augmented generation has raised extensive attention as it is promising to address the limitations of large language models including outdated knowledge and hallucinations. However, retrievers struggle to capture relevance, especially for queries with complex information needs. Recent work has proposed to improve relevance modeling by having large language models actively involved in retrieval, i.e., to guide retrieval with generation. In this paper, we present a novel iterative query formulation approach termed "LeapQuery" aimed at enhancing retrieval-augmented multi-hop question-answering systems. Current methods often treat input questions as complete queries, which may not suffice for complex inquiries that require pulling information from multiple sources. Our research addresses this limitation by integrating retrieval-augmented techniques with large language models (LLMs) to fill knowledge gaps effectively and prevent redundancy in data retrieval. This approach systematically enhances the ability of models to leverage external, up-to-date knowledge, crucial for addressing complex, long-tail questions and reducing model-generated hallucinations. Experimental results demonstrate significant improvements in retrieval and question-answering performance, showcasing the potential of LeapQuery to transform the landscape of complex query handling in AI-driven systems.

KEYWORDS

Open-domain Question-Answering, Retrieval-Augmented Generation, Information Retrieval

ACM Reference Format:
Kashob Kumar Roy, Palvi Shroff, Shreya Matta, and Taobo Liao. 2024. LeapQuery: Enhancing Multi-Hop Question Answering .

1 INTRODUCTION

Generative Large Language Models (LLMs) have become integral to numerous applications due to their impressive utility. Despite their capabilities, LLMs often lack representation of under-represented knowledge in their training data and are susceptible to generating inaccurate or hallucinated information, particularly in open-domain scenarios. Retrieval-augmented LLMs, therefore, have raised widespread attention as LLM outputs can be potentially grounded on external knowledge.

Previous retrieval-augmented LLMs, such as those cited in [1, 2], typically employed a one-time retrieval strategy. This method retrieves information once using the input question as the sole query. Such question-answering (QA) systems often fall short when dealing with queries that require understanding and integrating information from multiple data sources. This limitation is particularly evident in retrieval-augmented systems that treat the input question as a comprehensive query, assuming it to be fully formed and self-contained. Such an approach often fails to capture the nuances necessary for addressing more complex inquiries, which are increasingly prevalent in real-world applications.

To meet these complex information needs, recent advancements have introduced iterative strategies that involve retrieving knowledge multiple times throughout the generation process. These include using the full intermediate answer [3], the last generated response [4], or a follow-up question [5] as search queries. However, these approaches typically generate queries based on previous LLM responses, which may not effectively capture the comprehensive knowledge necessary to answer the main question. While they might retrieve new passages in initial iterations, these systems often quickly resort to reiterating the same passages that were previously retrieved, limiting their ability to continuously access new and relevant information. This repetitive retrieval highlights a critical gap in current methodologies, underscoring the need for more dynamic and nuanced retrieval strategies in QA systems.

To address the above-mentioned drawbacks of existing approaches, our research introduces "LeapQuery," a novel framework designed to enhance multi-hop question answering by employing iterative query formulation. This method systematically identifies and addresses 'knowledge gaps' within the query process, thereby refining the retrieval of information and ensuring the integration of relevant and up-to-date knowledge. By doing so, LeapQuery aims to minimize the redundancy of data retrieval and reduce the likelihood of model-generated hallucinations, issues that commonly plague current large language model (LLM) implementations in QA systems.

The significance of LeapQuery lies in its potential to transform the operational framework of question-answering systems, making them more robust and capable of handling the intricacies of

CS510, Spring 2024, University of Illinois Urbana-Champaign

'long-tail' questions. These are queries that involve less frequent, more specific scenarios, which traditional QA systems often handle poorly due to their reliance on more generalist, frequently encountered information. As the volume and complexity of data continue to grow, the ability of AI systems to adapt and respond to such challenges becomes increasingly critical.

In summary, our contributions in this work are as follows:

- We have introduced a model-agnostic framework that uniquely formulates queries based on identified knowledge gaps. This framework not only self-verifies these queries but also dynamically assesses when sufficient knowledge has been gathered to adequately answer questions. This approach allows for adaptive learning and improved accuracy in query handling, independent of the underlying model used.
- We have conducted a comprehensive experimental analysis to validate the effectiveness of our proposed framework. These experiments demonstrate a marked improvement in recall of information retrieval, highlighting the framework's potential to enhance question-answering systems.

In the following sections, we will explore the existing methodologies in multi-hop question answering, detail the LeapQuery approach, and present experimental results that demonstrate its efficacy in enhancing the performance of QA systems.

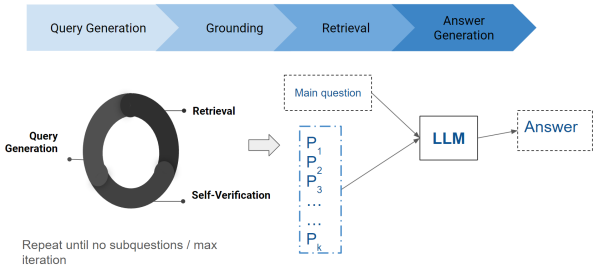

Figure 1: Overview of our framework LeapQuery

2 RELATED WORK

Our methodology is inspired by the work of Xiong et al.[6], which proposes a novel way for training a query encoder by ground-truth evidence passages and conceptualizing these passages as links in a chain. The goal there is to sequentially retrieve the next relevant passage in a multi-hop questions answering fashion. The reason for using such method is to guide the model navigate through multiple sources of information to answer complex questions. The sequential style of logic mimics human's reasoning process when tackling similar questions. Moreover, such process also enhances the relevance of the retrieved passage and lead to a more effective and efficient utilization of available information. Additionally, in the works of Trivedi et al.[4] and Shao et al. [3], both papers emphasize the utility of generating intermediate answers to aid in the discovery of subsequent relevant passages. Trivedi et al. suggests that iteratively alternating between extending the chain of thoughts (CoT) and retrieved information, the model can effectively discover the subsequent relevant passages with the aid of intermediate answers. Therefore, this methodology leads to more accurate and contextually relevant answers to complex multi-step questions. Shao et al. extends this idea by arguing that such iterative process involves leveraging generated output of previous iteration could bridge semantic gaps and improve retrieval in subsequent iterations. Despite these advantages, both cited work do not explicitly focus on the iterative refinement of queries based on identified knowledge gaps, which we think should be another important factor affecting the performance. Often, these methodologies treat almost all newly generated content as useful and beneficial information, which we think may not be necessarily true. We, therefore, build on the original ideas of these studies and treat the intermediate answers more carefully and strategically. Approach with similar fashion has actually been done in the work of Press et al. [5], which proposes a method called Self-ask. This method can not only leverage large language model (LLMs) to iteratively generate follow up questions, but also decompose the main question into sub-questions and use a search engine to find answers to these sub-questions. As a result, the model can gather more information and refine its understanding before answering the main question. This iterative process allows the LLM to engage in a dialogue with itself and incorporate external source information. However, it has certain limitations. The simplicity of the prompting technique may not consistently guarantee the questions are sufficiently targeted to the knowledge gap. Thus, the information retrieved, while relevant, may not be entirely focused or deep to satisfactorily answer the main query. Moreover, the redundant information gathered from internet can lead to inefficient questioning process and potentially cause the model to iterate over broad or irrelevant areas before honing in on the real queried information. To address these issues, in this paper, we propose a more systematic and structured approach to improve the precision and relevance of the follow-up questions. Our methodology advanced beyond the basic promoting technique and leverage another decision-making model that assesses the potential relevance of follow-up questions before they are refilled into the original model. This pre-assessment ensures the questions are aligned with the identified knowledge gaps. Moreover, we further enhance the performance of our approach through an iterative refinement loop. It will keep looping until the knowledge gaps are filled adequately. This process allows the model to generate questions directly targeting to discover critical information that fills the knowledge gaps. This leads to a more efficient and effective use of the large language model's capabilities, significantly enhancing the overall quality of the multi-hop question-answering process. Another work that inspire our methodology is the work by Chuang et al. [7]. Their work uses a query expansion model to generate a diverse set of queries related to the original question. After the generation of expand queries, a reranker is trained to predict the most relevant passages based on the expanded queries. Expanded queries will be evaluated according to the prediction of the reranker. Once the queries are expanded and reranked, the system leverage those to further perform information retrieval. However, work of Chuang et al. focused on open-domain questions while ours targets multi-hop question answering.

3 METHODOLOGY

3.1 Overview of the Framework

LeapQuery utilizes a structured approach that sequentially processes and refines user queries through a multi-component framework. This system is specifically designed to address the challenges of multi-hop question answering by iteratively refining the query process. Each component of the framework, namely Query Generation, Verification, Retrieval, and Answer Generation, works in concert to progressively close the information gaps identified during the question-answering process as shown in Figure 1.

3.2 Query Generation

Our framework stands out from existing methods by generating sub-queries that pinpoint knowledge gaps in the retrieved supporting passages thus far to address the main question as shown in Figure 2. In the initial phase, it retrieves a set of passages from an external corpus using the input question as a query. The system then leverages the advanced analytical capabilities of LLMs to identify the potential knowledge gaps inherent within the current context passages to answer the input question. Afterward, this LLM generates a series of sub-questions aimed at exploring these gaps. This set of sub-questions will be used in later stages to retrieve relevant passages from the corpus to fill the knowledge gaps.

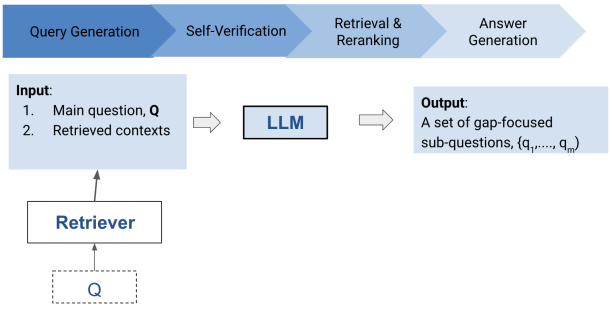

Figure 2: Query Generation

3.3 Verification

Following query generation, each sub-question undergoes a rigorous verification process shown in Figure 3. This step is crucial for maintaining the efficiency of the retrieval process by ensuring that only relevant and potentially fruitful sub-questions are pursued further. This verification step assesses how well each sub-question aligns with the intent of the main query and its potential to fill the identified knowledge gaps. Drawing inspiration from [8, 9], we employ a large language model (LLM) to check the alignment of each sub-question with the main question. The LLM returns 'True' if the sub-question is aligned, and 'False' otherwise. Rather than relying on a single verification instance, we implement a majority voting technique across multiple verification outcomes for each sub-question. Sub-questions that are predominantly verified as positive are then selected for the next stage.

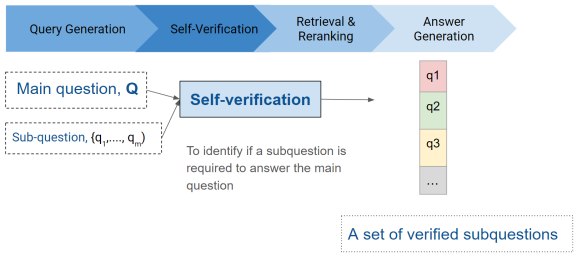

Figure 3: Self-Verification

3.4 Retrieval and Reranking

In the retrieval phase, each verified sub-question is used as a query to fetch relevant passages from an external corpus, with each sub-question retrieving l passages. Later on, we combine all newly retrieved passages with previously retrieved passages. Due to the input length limitations of large language models (LLMs), we cannot process an arbitrary number of passages. Hence, it becomes necessary to select the top k passages that are most relevant to the main question. To accomplish this, we employ a reranker model that ranks the passages based on their relevance scores. The reranker model is basically a cross-encoder that takes the input question and a passage and assesses them jointly to compute a score. It is superior in capturing the dynamics between the question and passages compared to a traditional retriever model, which encodes the query and passage separately and then measures their similarity score.

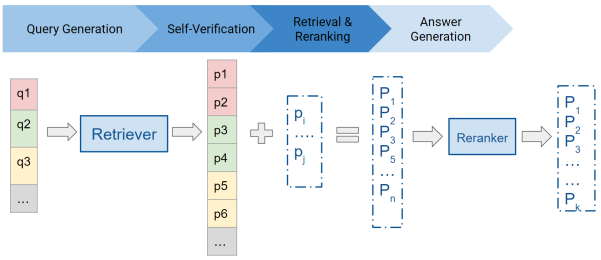

Figure 4: Retrieval & Reranking

These top k passages will be used in the next iteration of query generation.

3.5 Iterative Context Refinement and Answer generation

Our framework iteratively repeats query-generation, verification, and retrieval-reranking stages until the query-generation doesn't generate any sub-questions or it reaches the predefined maximum number of iterations. It is to be noted that if query generation stops generating sub-questions before the limit, it indicates that the current set of retrieved sufficiently addresses the main question. This criterion allows our system to dynamically manage the extent of the retrieval process as needed. The iterative design of the LeapQuery system facilitates ongoing refinement of both the queries

and the retrieval mechanisms. Each cycle in the query-verification-retrieval sequence builds on the insights from the preceding cycles, progressively diminishing the knowledge gaps. Upon concluding the iterations, we use the final top k passages as the context to formulate the answer to the main question.

4 EXPERIMENTAL ANALYSIS

To validate the effectiveness of LeapQuery, we employ a structured experimental setup involving benchmark datasets specifically designed for evaluating multi-hop question-answering systems. Comparative analysis is conducted against several baseline models to establish benchmarks for performance improvements. Metrics such as precision, recall, and F1 score are used to quantitatively assess retrieval accuracy, while qualitative assessments are made through user studies focusing on the relevance and usability of the answers generated.

4.1 Dataset

To evaluate the efficacy of our proposed framework, we have used the MuSiQue [10] dataset, which is a commonly used multi-hop question-answering dataset. This dataset is particularly suitable because its questions necessitate synthesizing information from multiple sources, demanding an effective multi-hop retrieval-augmented approach to generate answers. We use validation samples to compare the performance of our model with other existing baselines. For corpus preparation, we incorporated evidence passages provided for all training, and validation samples, resulting in a corpus comprising 101,962 passages. This comprehensive dataset allows for robust testing and validation of our retrieval and processing methodologies.

4.2 Baselines

For performance comparison, we have use existing recent baselines as follows:

- Retrieve-then-Answer [1]: The approach retrieves relevant passages from the corpus once using the main input question as a query and then uses them to answer the question.
- Self-Ask [5]: This model employs an explicit prompt to generate a single follow-up question at each iteration, which is then used to retrieve an answer. Traditionally, answers would be obtained using Google search or generated through a pre-trained large language model (LLM). In our setup, instead of using Google search or LLM pre-trained to find the answer to the follow-up question, we use the follow-up question to retrieve contexts first and then use these contexts to generate the answer for the follow-up question.
- IterRetGen [3]: This technique leverages answers generated at intermediate stages to fetch subsequent relevant passages. In the original implementation, the retriever is fine-tuned with the assistance of a reranker model to enhance its performance. However, for our experimental analysis, we opted to use only the pretrained retriever. This decision allows us to evaluate the baseline capabilities of the retriever without the additional influence of reranking, providing a clearer view of its standalone effectiveness in retrieving pertinent information.

4.3 Implementation Details

For our query generation, verification, and answer generation processes, we employed the LLama3-8B model [11]. To ensure a fair evaluation, we utilized the same model for generating answers and follow-up questions across all respective baselines. Additionally, we used the pretrained Contriever [2] as our primary retriever and a sentence transformer [12]-based cross-encoder as our reranker. This setup allows us to maintain consistency in the performance and capabilities of our models, providing a robust framework for comparative analysis.

4.4 Evaluation Metrics

- Recall @ K: his metric measures the ability of a retrieval system to retrieve all relevant documents within the top K results. It is defined as the proportion of relevant documents retrieved in the top K results out of all relevant documents available in the dataset. This metric is crucial for understanding how effectively the retrieval system captures necessary information.
- ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [13]: ROUGE is used to evaluate the quality of summaries or machine-generated text against a set of reference texts. It includes several measures such as ROUGE-N (which calculates the overlap of N-grams between the generated and reference texts), ROUGE-L (which measures the longest common subsequence). These metrics primarily focus on the recall of the reference's content by the generated text, making them suitable for assessing the completeness and accuracy of content generation in question answering.
- BERTScore [14]: BERTScore leverages the embeddings from models like BERT to evaluate the semantic similarity between machine-generated text and reference text. It computes the cosine similarity between the tokens' embeddings of the candidate and the reference texts, considering contextual information from all the words in the sentence. This metric is useful for assessing the quality of answers in question answering systems, as it measures how semantically close the generated answers are to a reference standard.

4.5 Retrieval Performance Analysis

To verify the efficacy of our gap-focused subquestions in sourcing more pertinent passages, we measured the recall performance for the top 5 passages. The comparative results presented in Table 1 demonstrate that our approach achieves significantly higher recall scores. In contrast, the Retrieve-then-Answer strategy shows lower performance because it retrieves passages only once using the main question as a query. This indicates that a single input question is inadequate for identifying all relevant passages, especially within a large corpus. Although the Self-Ask approach generates some follow-up questions, its performance is on par with a single retrieval attempt. We observed that simple prompt instructions such as *"Are follow-up questions needed here:"* do not effectively prompt LLMs to identify knowledge gaps.

Moreover, the IterRetGen approach outperforms Self-Ask, suggesting that intermediate responses from LLMs are beneficial for uncovering new relevant passages. These responses provide a richer

Table 1: Comparative Retrieval Performance

Methods	Recall @ 5
Retrieve-then-Answer	26.95
Self-Ask	27.31
IterRetGen	37.54
Our Approach (LeapQuery)	49.16

context for the input question than the question itself, aiding the retriever in identifying relevant passages. However, a significant limitation of this approach is that it reaches a plateau in retrieval performance quickly. In the initial iterations, it retrieves new relevant passages, but it fails to do so in later iterations. This is often because the intermediate responses contain a lot of overlapping information with already retrieved passages, leading to the repetition of the same set of passages in subsequent iterations. Thus, this model falls short for questions requiring a long chain of retrieved passages.

Conversely, our method consistently focuses on knowledge gaps and generates new subquestions targeting these gaps, which significantly increases the likelihood of retrieving new relevant passages at each iteration. The superior retrieval performance underscores the effectiveness of our retrieval techniques. Specifically, our approach achieved a Recall @ 5 of 49.16%, significantly outperforming the next best result of 37.54% by IterRetGen.

4.6 Question Answering Performance Analysis

LeapQuery has exhibited superior performance across all metrics for question answering. It achieved a score of 19.73 in ROUGE-L and 58.65 in BERTScore, indicating a significant enhancement in generating coherent and contextually accurate responses compared to the baseline models. Since we employ the same Llama3 model as the answer generator, the retrieval performance has a direct impact on the QA performance. This correlation underscores the importance of effective retrieval in enhancing the quality of the generated answers, as better context passages lead to more accurate responses.

Table 2: Comparative Evaluation of Question Answering Models Using ROUGE-L and BertScore Metrics

Models	ROUGE-L	BertScore
Retrieve-then-Answer	12.29	32.38
Self-Ask	14.28	44.57
IterRetGen	16.29	53.90
Our Approach (LeapQuery)	19.73	58.65

Our experimental results demonstrate that iterative query formulation, with its emphasis on identifying and filling knowledge gaps, effectively enhances the retrieval and processing capabilities of QA systems. The observed improvement in recall metrics indicates that LeapQuery excels at accessing relevant information, a critical factor for managing complex queries that span multiple data sources. This ability to accurately target and retrieve necessary information ensures that the system is well-equipped to handle the intricacies of multi-source queries, contributing to a more robust and effective QA performance.

4.7 Qualitative Analysis

The MuSiQue dataset features a set of subquestions that were instrumental in formulating the main complex input question. To perform a qualitative analysis, we utilize the ground-truth set of subquestions provided by the dataset. This involves manually examining the subquestions generated by our approach to assess how well they align with the ground-truth subquestions.

Input Question	Ground Truth Subquestions	LeapQuery Generated Subquestions
What company succeeded the owner of Empire Sports Network?	What company succeeded Adelphia Communications Corporation?	What company succeeded Adelphia Communications Corporation in owning the Empire Sports Network?
What record label did the person who is part of The Bruce Lee Band start?	Who is the part of The Bruce Lee Band? What record label Mike did?	Who is part of The Bruce Lee Band? Who is Mike Park and what is his connection to The Bruce Lee Band? What is the name of the record label run by Mike Park, the person who is part of The Bruce Lee Band?
Who won the 1993 Indy Car race in the city with the largest population in the state where Poachie Range is located?	In which state is Poachie Range located? What is the largest populated city in Arizona? who won the indy car race in Phoenix?	What city is located in the state where Poachie Range is located? What was the winner of the 1993 Indy Car race?

Figure 5: Examples of LeapQuery generated subquestions

In Figure 5, we showcase several examples of subquestions generated by our approach, LeapQuery. As illustrated, LeapQuery is capable of producing a set of subquestions that not only encompass various aspects of the main input question but also provide more contextual information compared to the ground-truth questions. This enhanced context in the subquestions significantly improves the effectiveness of the retrieval process. By integrating additional relevant details, these subquestions enable a more targeted and comprehensive search for pertinent information, thereby increasing the likelihood of retrieving highly relevant passages that can aid in formulating more accurate answers. This demonstrates the utility of our method in generating queries that are not just aligned with, but also augmentative to, the knowledge requirements posed by complex questions.

5 FUTURE SCOPE

The promising results of LeapQuery open several avenues for further research and development. Future work will focus on several key areas:

1. Scalability: Extending the framework to handle larger datasets and more complex queries to test the robustness and scalability of LeapQuery.
2. Real-Time Applications: Adapting the methodology for real-time question answering environments, where speed and efficiency are crucial.
3. Language Model Enhancement: Integrating LeapQuery with more advanced LLMs and exploring the effects of different architectures on its performance.
4. Domain-Specific Adaptations: Tailoring LeapQuery for specialized fields such as medical, legal, or financial information retrieval, where accuracy and reliability are especially important.

5. Multilingual Capabilities: Expanding the framework to support multilingual question answering to cater to a global user base.
6. User Interaction Models: Developing more intuitive user interfaces that allow non-experts to interact effectively with the system, potentially increasing the accessibility and usability of LeapQuery.
By addressing these areas, we aim to not only refine the capabilities of LeapQuery but also broaden its applicability and impact in the field of AI-driven question answering.

6 CONCLUSION

The implementation of LeapQuery marks a significant advancement in the field of question answering systems. Our research not only introduces a novel approach with iterative query formulation but also successfully demonstrates its efficacy through substantial improvements in retrieval performance and answer accuracy across multiple evaluation metrics.
LeapQuery significantly surpasses traditional methods by employing a mechanism that dynamically identifies and fills knowledge gaps during the retrieval process. This not only increases the efficiency of information retrieval but also enhances the quality of the responses generated, as evidenced by superior scores in Recall @ 5, ROUGE-L, and BertScore metrics compared to established benchmarks. Such enhancements are crucial for dealing with complex queries that require understanding and synthesizing information from disparate sources.
The practical applications of LeapQuery extend beyond academic research, offering promising enhancements for real-world AI systems that demand high levels of accuracy and contextual sensitivity, such as in healthcare diagnostics, legal research, and financial forecasting. The ability of LeapQuery to reduce the occurrence of redundant data retrieval and mitigate model-generated errors can lead to more reliable AI interactions and improved user trust.
Looking ahead, the focus will be on expanding LeapQuery's capabilities to include real-time processing, increasing its adaptability to different domains, and testing its effectiveness across larger and more diverse datasets. Additionally, enhancing the system's ability to handle multilingual content and developing more intuitive user interfaces are key goals that will make LeapQuery more accessible and effective for a global audience.
In summary, LeapQuery not only sets a new standard for multi-hop question answering systems but also offers a scalable and robust framework poised to drive significant improvements in the AI field, making sophisticated and accurate question answering more attainable and practical.

7 TEAM CONTRIBUTIONS

Shreya Matta (matta6): Literature Review, Conceptualization, Data Collection, Experiment Design.
Palvi Shroff(pps6): Literature review, Testing, Feedback Integration.
Taobo Liao(taobol2): Literature review, Model development, Data analysis.
Kashob Kumar Roy(kkroy2): Literature review, Model development, Experimental Analysis.

REFERENCES

[1] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.
[2] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.
[3] 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*, pages 9248–9274, Singapore, December 2023. Association for Computational Linguistics.
[4] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions, 2023.
[5] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models, 2023.
[6] Wenhan Xiong, Xiang Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Scott Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oguz. Answering complex open-domain questions with multi-hop dense retrieval. In *International Conference on Learning Representations*, 2021.
[7] Yung-Sung Chuang, Wei Fang, Shang-Wen Li, Wen-tau Yih, and James Glass. Expand, rerank, and retrieve: Query reranking for open-domain question answering. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12131–12147, Toronto, Canada, July 2023. Association for Computational Linguistics.
[8] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.
[9] Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*, 2022.
[10] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.
[11] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
[12] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.
[13] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.
[14] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.