Improving Multi-Hop Retrieval for Question Answering via Bipartite Question-Oriented Graphs

Micah McCollum
Department of Electrical Engineering and Computer Science
University of Arkansas
Fayetteville, Arkansas, United States of America
email: mlm132@uark.edu

Susan Gauch
Department of Electrical Engineering and Computer Science
University of Arkansas
Fayetteville, Arkansas, United States of America
email: sgauch@uark.edu

***Abstract*—Accurately answering multi-hop questions requires full retrieval of multiple, interdependent passages and is a long-standing problem in the area of natural language question answering. While retrieval-augmented generation helps address single-hop questions, many retrievers presently focus on semantic similarity in a dense vector space, which is insufficient for handling multi-hop questions specifically. To ameliorate this, we propose constructing a bipartite question graph composed of hypothetically generated questions connected to passage chunks at index time. The construction of the graph is guided by a large language model to prioritize the formation of edges that signal whether a question can be answered by a text passage. During retrieval, the graph is traversed starting from semantically similar seed questions and accrues relevant connected passage chunks after a set number of hops. Results from preliminary experiments on a challenging multi-hop dataset show promise in this approach. Full context retrieval accuracy was 9% for $k = 5$ and 32% for $k = 20$ compared to 5% and 21%, respectively, for the naive vector-only baseline. These results highlight the potential of graph-based retrievers in the area of multi-hop question answering, leading to improvements in downstream applications such as chat bots, search engines, web browsers, and other applications involving natural language interaction, knowledge discovery, and information retrieval.**

***Keywords*—information retrieval; question answering; retrieval-augmented generation; large language models; graphs.**

I. INTRODUCTION

Large Language Models (LLMs) have transformed the ways with which software is interacted. One key area that has been affected is information retrieval. Companies such as Google are supplementing their search engine results with Artificial Intelligence (AI) summarizations powered by LLMs, a synthesis of traditional information retrieval and knowledge acquisition techniques with recent developments in AI. However, despite rapid adoption, LLMs alone are not equipped to comprehensively handle all types of queries. Many complex natural language questions are multi-hop in nature, requiring the combination of multiple, interdependent pieces of information to produce a satisfactory answer. Techniques like Retrieval-Augmented Generation (RAG) aim to improve general question answering by integrating a retrieval component with the LLM but often remain insufficient for reliably answering multi-hop questions [1].

To address this limitation, we propose a novel graph approach that: (1) generates hypothetical questions via an LLM that can be answered by text chunks from the corpus; (2) connects these text chunks with the generated questions in a bipartite graph, according to whether the question can be answered by a chunk, as judged by the LLM; and (3) enables efficient query-time traversal over the graph to select the most relevant chunks. By prompting an LLM to form edges up front, the retriever maintains efficiency at query time while amortizing compute costs.

The rest of the paper is structured as follows. We examine related work in Section II and describe our approach in Section III. In Section IV, we present preliminary experimental results, finally concluding with a brief discussion thereof in Section V.

II. RELATED WORK

Information retrieval is an area concerned with the systematic storage and retrieval of unstructured data, typically from the web [2]. The extracted data is then preprocessed and tokenized to prepare for indexing. Traditionally, this involves building an inverted index mapping unique terms to the documents in which they are contained. Queries and documents are represented as sparse vectors and weighted according to schemes like Term Frequency-Inverse Document Frequency (TF-IDF) [2]. Similarity measures, such as cosine similarity, may be used to compare the query to documents for the final results [2].

While effective, traditional lexical searches use exact term matching and often cannot disambiguate matches that have no semantic relevance to each other, as in the case of "rock music" and "rock salt." Later approaches incorporate dense vector embeddings or learned representations to improve semantic understanding [2].

In 2020, Facebook AI Research (now Meta) introduced RAG to leverage the generative capabilities of LLMs in producing natural language answers from retrieved results ("non-parametric memory") [1]. RAG showed improvements in the final outputs and reducing hallucinations due to an incomplete, inaccurate, or outdated knowledge base from pretraining parameters. This underscores the importance of dynamic retrieval in capturing the context needed to accurately answer arbitrary questions, especially those that are domain specific or are not sufficiently accounted for in the training data.

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

In 2024, Microsoft Research followed up with GraphRAG, introducing community hierarchies and summarizations from a knowledge graph to perform global reasoning [3]. By leveraging the structured connectivity between contexts instead of only relying on semantics, GraphRAG demonstrated improvements in comprehensive and diverse question answering.

One pertinent method of improving retrieval performance is augmenting embeddings of real textual signals with synthetically generated text. Hypothetical Document Embeddings (HyDE) is one such approach, which uses a language model to create hypothetical documents based on the query and embeds them alongside real documents [4]. While HyDE is a query-time approach, Question-Oriented Text Embeddings (QuOTE) is an index-time approach which instead generates and embeds hypothetical questions for each chunked text passage in the document corpus, where the generated question can be answered by the passage on which it is conditioned [5].

Past studies have shown query decomposition as a viable technique for improving multi-hop reasoning, since multi-hop queries can be viewed as a sequence of multiple single-hop queries that are easier to answer individually [5][6][7]. Taken together, these works inform our methodology and touch on several key facets of it.

III. Approach

Our approach features a key contribution from QuOTE in the form of hypothetical question generation and takes it one step further by constructing a *Bipartite Question-Oriented Graph* (BiQOG). The combination of these two concepts reflects intuition from query decomposition; given a collection A of generated hypothetical questions, it is possible that an arbitrary multi-hop question can be decomposed into a set B of multiple single-hop questions such that there is overlap between A and B [6][7]. Through this lens, it is primarily a matter of mapping the multi-hop question to an initial set of seed questions after graph construction.

The graph construction process is facilitated by an LLM which is instructed to provide a binary answer for whether or not a given chunk answers a hypothetical question. If that binary answer is yes, then an undirected edge is formed. By forming explicit edge connections between these two types of data entities, latent relationships between different questions arise, thereby aiding multi-hop reasoning. In this way, query decomposition is moved from query time to index time. Note that while there is an extra graph construction step, parallel processing can significantly reduce runtime.

During retrieval, the retriever first embeds the query and matches it to semantically similar seed questions, from which the graph traversal starts. Over a predefined number of hops, the text chunks neighboring the seed questions are collected and reranked with a cross-encoder according to a width, which is the number of neighbors used to traverse the graph further. After the hops are finished, the list of seen text chunks is reranked a final time before the top-k are passed into the LLM's context window for downstream answer generation.

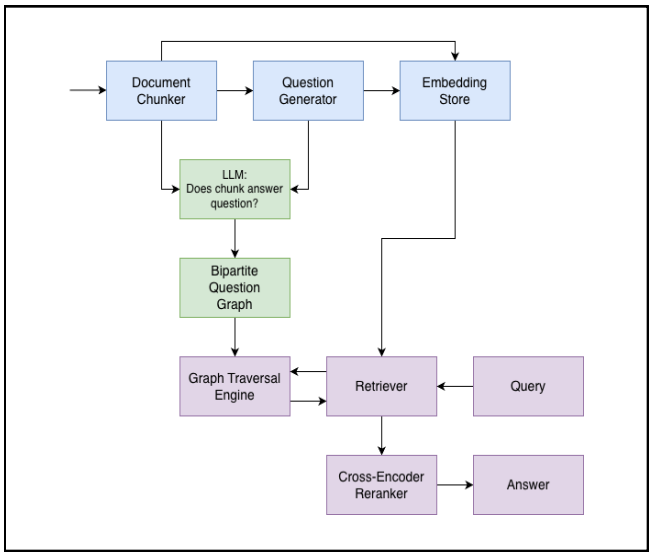

Figure 1. BiQOG High-Level Architecture.

Figure 1 illustrates the high-level architecture of our method and is comprised of text indexing, graph construction, and retrieval. Once the graph is constructed, it is used during the graph traversal stage of retrieval. Importantly, the graph's construction process is amenable to incremental updates and is left as an implementation detail.

IV. Experiments

The dataset used to evaluate BiQOG and the baselines is MultiHop-RAG, a challenging dataset specifically made for evaluating question answering and retrieval tasks in a multi-hop RAG setting [8]. It features four types of queries: *Inference, Comparison, Temporal,* and *Null.* For the purpose of scope, we exclude *Null* queries from the test set since they are unanswerable. Although we are principally motivated by RAG as a downstream application, the quality of RAG depends upon the quality of retrieval. Furthermore, RAG is difficult to accurately assess for end-to-end question answering. Thus, our evaluation focuses exclusively on retrieval quality to maintain a well-defined scope for experiments as well as generalizability. Our baselines include naive dense vector retrieval and QuOTE.

We follow the choice of metrics found in QuOTE for multi-hop evaluation, which is Full@k and Recall@k [5]. Recall@k measures the average fraction of gold evidence found within the top-k over all queries. Full@k indicates that all gold evidence is found within the top-k retrieved results; in other words, it measures how many queries on average retrieve 100% recall. Because the accuracy of an answer to a multi-hop question requires having all relevant passages, Full@k is the most important metric in determining the effectiveness of multi-hop retrieval. We conduct multiple experiments with different values of k for a comprehensive evaluation. Table 1 shows our preliminary results.

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

BiQOG demonstrates better performance in retrieving all necessary gold context compared to the baselines and is competitive with or better than baselines for queries where only part of the gold is retrieved. Notably, at higher values of *k*, BiQOG breaks away from the baselines in both metrics, showing that it is more adept at retrieving gold within the top-*k* altogether and achieving coverage, irrespective of exact ranking. QuOTE outperforms BiQOG in Recall@5, suggesting that BiQOG experiences a gap where it retrieves either all evidence (in the case of Full@5) or it retrieves a slightly lower fraction of gold compared to QuOTE. Despite this, the strong results from BiQOG illustrate the effectiveness of a graph structure for multi-hop retrieval compared to baseline approaches which do not leverage any graph techniques.

TABLE I. COMPARISON OF NAIVE, QUOTE, AND BIQOG BASELINES ACROSS RETRIEVAL TASKS IN MULTIHOP-RAG. BEST ENTRIES ARE IN **BOLD**.

Approach	MultiHop-RAG			
	Full@5	***Full@20***	***Recall@5***	***Recall@20***
Naive	5.00	21.00	27.08	51.50
QuOTE	8.00	29.00	**32.42**	59.00
BiQOG	**9.00**	**32.00**	31.92	**60.33**

For all LLM tasks, including question generation and edge formation, we use gpt-4o-mini due to its cost-effectiveness. Similarly, the embedding model and reranking model used in the experiments are all-MiniLM-L6-v2 and ms-marco-MiniLM-L6-v2, respectively. For this study, we elected not to use state-of-the-art language and embedding models due to cost concerns, but model choice should be agnostic [5].

A limitation of the approach is the heavy use of an LLM, resulting in higher token costs and inference latency. However, this usage is exclusively offline, so the impact is relegated to a one-time cost. Additionally, the graph is loaded into memory for these experiments. Finally, from failed cases in which recall was zero for a query, *Temporal* appeared as the most problematic query type, suggesting our approach alone may not be sufficient in handling all types of queries, and so other techniques may have to be supplemented to compensate.

Overall, the results show promising improvements in full context retrieval over baselines, demonstrating the importance of the graph structure in multi-hop settings.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose the use of a novel graph representation for multi-hop retrieval in RAG applications by encoding hypothetical questions conditioned on passage chunks into a bipartite graph structure at index time. We employed LLMs to generate the hypothetical questions and perform edge formations during offline graph construction and utilized simple but effective retrieval at query time. Experimental results demonstrate that in full context retrieval, a necessary requisite for accurate multi-hop question answering, our approach reports a Full@20 of 32% over the naive vector-only baseline 21% and a Full@5 of 9% over the baseline 5%. We plan to do further testing on a large dataset to get a fuller evaluation. In the future, we anticipate that incorporating query decomposition may yield improvements in retrieval accuracy, since multi-hop questions can be seen as a composition of multiple single-hop questions [6][7]. Thus, further investigation into the effective mapping between queries and single-hop seed question nodes in the bipartite graph could prove fruitful.

REFERENCES

[1] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive nlp tasks," in Advances in Neural Information Processing Systems, vol. 33, pp. 9459-9474. 2020.

[2] C. D. Manning, P. Raghavan, and H. Schütze, An Introduction to Information Retrieval. Cambridge, England: Cambridge University Press, 2009.

[3] D. Edge et al., "From local to global: a graphrag approach to query-focused summarization," arXiv preprint arXiv:2404.16130, 2024. [Online]. Available from: https://arxiv.org/pdf/2404.16130. [Retrieved: March, 2026]

[4] L. Gao, X. Ma, J. Lin, and J. Callan, "Precise zero-shot dense retrieval without relevance labels," Proc. 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Jul. 2023, pp. 1762-1777, doi:10.18653/v1/2023.acl-long.99.

[5] A. Neeser, K. Latimer, A. Khatri, C. Latimer, and N. Ramakrishnan, "Quote: question-oriented text embeddings," arXiv preprint arXiv:2502.10976, 2025. [Online]. Available from: https://arxiv.org/pdf/2502.10976. [Retrieved: March, 2026]

[6] S. Min, V. Zhong, L. Zettlemoyer, and H. Hajishirzi, "Multi-hop reading comprehension through question decomposition and rescoring," Proc. 57th Annual Meeting of the Association for Computational Linguistics, Jul. 2019, pp. 6097-6109, doi: 10.18653/v1/P19-1613.

[7] R. Fu, H. Wang, X. Zhang, J. Zhou, and Y. Yan, "Decomposing complex questions makes multi-hop qa easier and more interpretable," Findings of the Association for Computational Linguistics: EMNLP 2021, Nov. 2021, pp. 169-180, doi:10.18653/v1/2021.findings-emnlp.17.

[8] Y. Tang and Y. Yang, "Multihop-rag: benchmarking retrieval-augmented generation for multi-hop queries," arXiv preprint arXiv:2401.15391, 2024. [Online]. Available from: https://arxiv.org/pdf/2401.15391. [Retrieved: March, 2026]

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org