MultiHop-RAG: A Longitudinal Study on its Implementation and Benchmarks

Murtaza Asrani, Rishabh Agrawal, and Apurva Narayan
University of Western Ontario
{masrani2,ragrawa9,apurva.narayan}@uwo.ca

Abstract

In recent years, the popularization of large language model (LLM) applications such as ChatGPT has made it easy for anyone to access new knowledge and solve problems. However, these LLM applications come with precaution; often, the LLMs powering these applications can provide misleading or entirely incorrect answers referred to as hallucinations. Hallucinations can occur for many reasons, one of which is due to shortcomings in the dataset used to train the LLM. In combatance to such events, researchers have devised a new method of response generation known as Retrieval-Augmented Generation (RAG). However, inadequate response quality emerges in the system when handling complex multi-hop queries, which require retrieving and reasoning over multiple pieces of supporting evidence. In this paper, we will implement and benchmark a novel RAG system called MultiHop-RAG designed to handle multi-hop queries specifically. We will provide an instructive procedure for building the MultiHop-RAG system and demonstrate its utility by deriving benchmarks and comparing them against existing RAG systems.

1 Introduction

Large language model (LLM) applications have become popularized due to their extensive knowledge of various topics and specialized problem-solving abilities. However, LLMs have a reputation for providing erroneous or illogical answers, referred to as hallucinations. Many factors cause hallucinations, including insufficient information within the dataset used to train the LLM. In particular, insufficient information potentially refers to outdated or missing data, resulting in outdated or irrelevant responses to user queries. For instance, if our query were, "Who is the current president of the United States?" an outdated response would be "Donald Trump;" similarly, an irrelevant response due to missing data might return "The sixteenth president of the United States was Abraham Lincoln." Thus, researchers have introduced Retrieval-Augmented Generation (RAG) to mitigate hallucinations due to insufficient LLM pretrained datasets.

1.1 Retrieval Augmented Generation (RAG)

Ongoing developments in large language models (LLM) and natural language processing (NLP) have given rise to innovations such as ChatGPT. One innovation that has garnered significant attention in the artificial intelligence community is Retrieval-Augmented Generation (RAG). In its current state, even the most sophisticated Large language models have been known to provide entirely incorrect or irrelevant answers labelled by the community as hallucinations, plaguing their credibility. Retrieval-Augmented Generation utilizes an external knowledge base to optimize and reinforce the output of large language models (Borgeaud et al., 2022), mitigating the occurrence of hallucinations and thus restoring the credibility of responses (Gao et al., 2023). More accurately, RAG systems utilize semantic similarity matching between user queries and knowledge base to derive relevant responses; however, semantic similarity matching pales in comparison to analyzing the underlying concepts within the external knowledge base, a process used to answer multi-hop queries.

1.2 Multi-Hop Query

Retrieval-Augmented Generation systems are benchmarked by their response quality; however, until recently, benchmarks only evaluated simple queries where responses can be generated using a singular piece of evidence. These benchmarks must be revised in the face of multifaceted queries, often requiring numerous pieces of evidence to derive a response (Tang & Yang, 2024, p. 1). In recent months, researchers have delved into benchmarking the retrieval and reasoning capability of LLMs for complex multifaceted queries, referred to as multi-hop queries. A multi-hop query is defined as a query which necessitates the retrieval and ratiocination over various pieces of supporting evidence to derive an answer which yields more optimal results in comparison to conventional similarity search methods like cosine similarity between query and chunk embeddings (Tang & Yang, 2024, p. 3). The various pieces of evidence, r_n, retrieved as part of response derivation form a retrieval set, R_q, for a query, q, represented as $R_q = \{r1, r2, ..., r_k\}$. For instance, the multi-hop query, "Which G7 country has the lowest inflation rate as of 2024?" requires retrieving pieces of supporting evidence related to inflation rates of each of the G7 countries, then deriving a response by comparing and reasoning over the retrieved pieces of supporting evidence. Furthermore, a multi-hop query can be categorized as either inference, comparison, temporal, or null (Tang & Yang, 2024, p. 3); categorizing multi-hop queries allows for a more efficient method of benchmarking a MultiHop-RAG implementation.

Inference query: For query q, an answer is derived through reasoning from evidence in the retrieval set R_q. For example, an inference query might be: "Which company has spent billions to maintain its default search engine status on various platforms?"

Comparison query: For query q, an answer is derived by comparing evidence in the retrieval set R_q. For instance, a comparison query might ask: "Did Instagram or TikTok report a higher daily active user count for 2023?"

Temporal query: For query q, an answer is derived through analysis of time-related elements from evidence in the retrieval set R_q. An example of a temporal query would be: "Was Dave Grohl a drummer for the band Nirvana before he became a singer for the Foo Fighters?"

Null query: For query q, an answer cannot be derived from the retrieval set R_q. Null queries are used to assess hallucination issues in the MultiHop-RAG system. For example, assuming company XYZ does not exist, a null query might ask: "When was the company XYZ founded?"

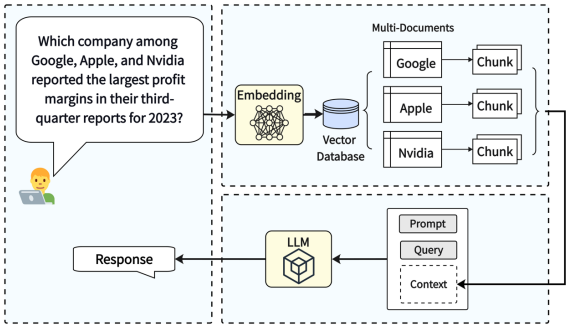

Figure 1: Multi-Hop Query Design (Tang & Yang, 2024, p. 1)

2 A Conceptual Breakdown: MultiHop-RAG

The MultiHop-RAG system, in many ways, differs from conventional RAG systems; one of these ways is in how the RAG dataset is formatted. We begin by examining external documents the user provides, extracting factual statements from each document. For example, a factual statement extracted from a document could be: "On June 29th, 2007, the entire world beheld the release of Apple's now decade-old and revolutionary product, the iPhone." Next, we prompt our large language model to generate a claim for each factual statement extracted from the documents. A claim is a statement or assertion expressing a belief, opinion, or fact, void of any ambiguous references to any person, place, or thing; each claim should have a topic and target, which act as bridges for connecting similar claims (Tang & Yang, 2024, p. 4). For instance, a claim regarding the aforementioned factual statement might look like this: "Apple first released the iPhone on June 29th, 2007, over a decade ago," where "Release of the iPhone" is the claim topic and "Apple" is the claim target. Afterwards, we follow a similar process, generating bridge targets for the query inputted by the user. Then, we search our dataset for claims similar to the query to form a retrieval set.

Evidence	A plunge in global markets combined with rising Japanese interest rates and crumbling tech stocks has created Japan's worst market crash since 1987.
Claim	The Japanese stock market experiences its worst crash since 1987.
Bridge-Topic	Japanese Stock Market
Bridge-Target	Market Crash

Table 1: An example of a claim with its bridge-target and bridge-topic. Evidence source: (Cooban et al., 2024)

3 Implementing the MultiHop-RAG

Researchers at the Hong Kong University of Science and Technology have leveraged the concept of MultiHop queries to implement a dataset of LLM-generated query, evidence and answer tuples to analyze an LLM's ability to ratiocinate over a retrieval set of evidence (Tang & Yang, 2024, p. 4). Our research intends to use the concepts used for designing queries instead to implement into an advanced Multihop-RAG system which can answer any user query, including non-multi-hop queries, using multi-hop retrieval and reasoning.

Step 1: Dataset Collection. The MultiHop-RAG system is designed to work with any text-based external documents. For our testing, we will utilize a corpus dataset[1] compiled by researchers at the Hong Kong University of Science and Technology (Tang & Yang, 2024); the corpus consists of news articles published from September 26th, 2023, to December 26th, 2023. Each news article has a max token length of 1,024 and is paired with its respective metadata: title, publish date, author, category, URL, and news source.

Step 2: Evidence Extraction. With a trained HuggingFace fact-or-opinion binary classifier model[2], we extract factual sentences from each news article to be later processed into evidence to populate the MultiHop-RAG dataset. News articles analogous in evidence with other news articles are retained within the MultiHop-RAG dataset to reinforce query responses because the evidence used to derive these responses is drawn from numerous sources.

Step 3: Claim, Bridge-Target, Bridge-Topic Generation. Our MultiHop-RAG system intends to have Mixtral-8x7B-Instruct (Jiang et al., 2024) compile a retrieval set using the evidence extracted from the previous step. However, the format of the extracted evidence needs to be revised for response generation due to inconsistencies in linguistic structure. To circumvent these inconsistencies, researchers have introduced the concept of a "claim": paraphrased evidence void of ambiguous pronouns or entities (Tang & Yang, 2024, pp. 2-4). We prompt Mixtral-8x7B-Instruct (Jiang et al., 2024) to convert each piece of extracted evidence into a claim and verify the consistency between the evidence and claim using the UniEval (Zhong et al., 2022) framework to ensure a precise conversion.

Bridge-Target and Bridge-Topic: Each claim is generated alongside a target and topic found within the evidence. For example, if our claim is "The Japanese stock market experiences its worst crash since 1987," the target would be "Japanese Stock Market," and the topic would be "market crash." Identifying a target and topic from each claim allows the MultiHop-RAG system to link claims together and append them to the retrieval set for query answering; thus, we refer to them as bridge-target and bridge-topic as they bridge claims together (Tang & Yang, 2024, p. 4). We prompt Mixtral-8x7B-Instruct (Jiang et al., 2024) to identify a bridge-target and bridge-topic for each claim.

Step 4: Prompt-Based Retrieval Set Generation. This step follows a similar process to the previous step: from the user prompt, 2-4 bridge-targets are extracted and used to retrieve n (where $n \leq 8$) claims and each of their relevant details (bridge-targets and bridge-topics). We prompt Mixtral-8x7B-Instruct (Jiang et al., 2024) to re-generate 1-2 bridge targets for each claim to avoid excluding secondary targets, which may be influential in linking claims. For instance, if our claim was "Apple introduces Apple Intelligence, a step towards challenging Nvidia's artificial intelligence platform," a bridge target might ini-

[1] https://huggingface.co/datasets/yixuantt/MultiHopRAG

[2] https://huggingface.co/lighteternal/fact-or-opinion-xlmr-el

tially be "Apple," excluding "Nvidia," through this more refined step, both targets are included with the claim. Altogether, a maximum number of 20 bridge targets are generated (4 query and 16 claim bridge targets); we then perform a cosine similarity search on the MultiHop-RAG dataset to retrieve a claim for each bridge target (a maximum number of 20 claims) to append to out retrieval set.

4 Benchmarks

In general, RAG-related tasks can be categorized as either retrieval-related or generation-related. Tasks focusing on retrieving relevant evidence from the RAG knowledge base are retrieval-related, while tasks focusing on response generation given the retrieved evidence are generation-related (Tang & Yang, 2024, p. 6). Our benchmarks will showcase the generation-related performance of a conventional RAG system alongside our implementation of the MultiHop-RAG system; both fed the same text corpus mentioned earlier in the paper.

For our experiment, we utilize the query and answer pair generated from the MultiHop dataset created by researchers from the Hong Kong University of Science and Technology (Tang & Yang, 2024). We set each answer as the ground truth for its respective query. To create our embeddings, we use the sentence embedding model, all-mpnet-base-v2[3], from HuggingFace without the assistance of a re-ranking model. When determining if a generated answer matches a ground truth answer, we prompt the respective LLM to output true if matching and false otherwise; these results are compiled to form an accuracy score of the model, from 0% to 100%.

Query Type	Count	F1 Score
Inference	816	0.89
Comparison	856	0.76
Temporal	300	0.77
Null	583	0.73
Total	2555	0.79

Table 2: Query type count and F1 score.

Experiment Setup: For each query, we retrieve the top-8 chunks to derive a response and include all four types of queries (inference, comparison, temporal, and null) in our testing (refer to table 2). We leverage the existing accuracy scores of the most robust LLMs, including GPT-4 (OpenAI, 2023), GPT-3.5, Claude-2 (Anthropic, 2023), Llama-2-70b-chat-hf (Touvron et al., 2023), and Google-PaLM (Google, 2023) published by researchers at the Hong Kong University of Science and Technology (Tang & Yang, 2024, p. 7); these accuracy scores are based on the MultiHop-RAG implementation used in their paper, which we will refer to as the "standard" implementation (a traditional RAG with MultiHop dataset). In Addition, we score the accuracy of Llama-3.1-8B-Instruct (Meta, 2024) and Mixtral-8x7B-Instruct (Jiang et al., 2024) based on our MultiHop-RAG implementation, which we will refer to as the "advanced" (adv) implementation.

	Models	Accuracy
Standard	GPT-4	**0.56**
	ChatGPT	0.44
	Llama-2-70b-chat-hf	0.28
	Mixtral-8x7B-Instruct	**0.32**
	Claude-2.1	0.52
	Google-PaLM	0.47
Adv.	Llama-3.1-8B-Instruct	**0.66**
	Mixtral-8x7B-Instruct	**0.37**

Table 3: Generation Accuracy of LLMs, Standard vs Advanced. Standard Accuracy Source: (Tang & Yang, 2024, p. 7)

Experiment Results: Table 3 displays the response accuracy of the LLMs used. Most notably, Llama-3.1-8B-Instruct (Meta, 2024) vastly outperforms GPT-4 (OpenAI, 2023) in terms of accuracy by a margin of 10%, which is accredited mainly to the difference in implantation; the advanced implementation returns a greater top-accuracy score of 66% compared to the top-accuracy score of 56% for the standard implementation. This deduction is further validated when comparing the accuracy scores of Mixtral-8x7B-Instruct (Jiang et al., 2024) used in both implementations; the advanced implementation, scoring 37%, outperforms the standard implementation, scoring 32%, by 5%. To make sure our system is robust in selecting the relevant chunks using the knowledge graph alone, we omit the use of any re-rank

[3]https://huggingface.co/sentence-transformers/all-mpnet-base-v2

model/algorithm. Our triumphant results were expected as the advanced implementation focuses on the semantic meaning of the inputted data and tailors its vector store to the query, thus pulling much more relevant information.

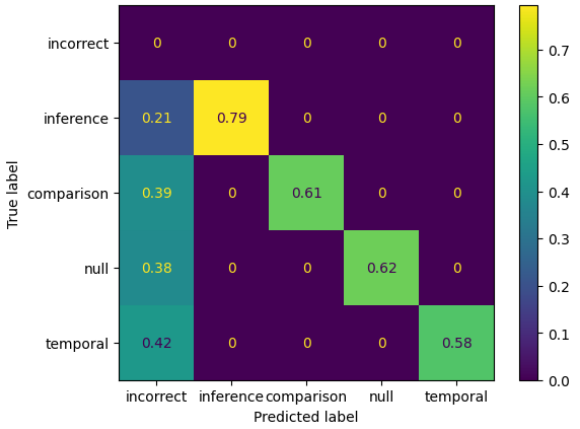

Figure 2: Llama-3.1-8B-Instruct generation accuracy for different query types.

Figure 2 shows the detailed results of all four query types for Llama-3.1-8B-Instruct (Meta, 2024). The model displays a consistently robust performance across all query types, with accuracy scores hovering in the 60% to 80% range. Furthermore, table 2 displays the F1 score of the model for classifying all four query types, hovering between the 70% to 90% range with an overall F1 score of 79%, ensuring the model can soundly identify correct performance for the correct class of query. These findings further verify the superiority of our advanced implementation over the standard implementation, which displays strong performance for null queries only (Tang & Yang, 2024). Although consistent, there is still room for improvement with the advanced implementation, especially temporal query responses, being accurate only 58% of the time in our testing.

5 Conclusion

In this paper, we build upon the research of the novel MultiHop-RAG system, suited for generating quality responses to multi-hop queries, which require the retrieval and reasoning over multiple pieces of supporting evidence and are frequently encountered in real-world scenarios (Tang & Yang, 2024, p. 1). Our research details the implementation approach, consisting of evidence extraction, claim generation, prompt handling, and retrieval set, followed by its respective benchmarks. We aim to provide a more instructive approach to implementing the MultiHop-Rag system so that artificial intelligence communities can leverage our findings to advance the effectiveness of RAG systems.

Limitations

Our implementation of the MultiHop-RAG system contains numerous limitations that could be ameliorated in future research. Unlike a conventional RAG system, MultiHop-RAG requires extensive preprocessing time to form its evidence database.t For each chunk of evidence, the selected LLM must generate a claim and its bridge-target and bridge-topic, followed by verifying the consistency between evidence and claim using another LLM. Future work could consider homogenizing this two-step process for a more streamlined preprocessing procedure. Additionally, the MultiHop-RAG system is limited to deriving its evidence database from the documents provided by the user, potentially compromising the RAG database with outdated or incorrect information, which contradicts our research goal of mitigating insufficient information within a RAG database. Future work could consider integrating a web search component into the MultiHop-RAG system, similar to a Corrective-RAG system, ensuring the MultiHop-RAG database is constantly fed accurate data.

References

Anthropic. 2023. Claude 2.1 (May version). https://api.anthropic.com/v1/messages. Claude 2.1.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of Proceedings of Machine Learning Research, pages 2206–2240. PMLR.

Anna Cooban, Laura He, Juliana Liu. 2024. Japanese stocks rebound from worst crash since 1987 while global markets are mixed.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations.

Google. 2023. PaLM 2 (May version). https://generativelanguage.googleapis.com/v1beta2/models/. Chat-bison-002.

Meta. 2024. Llama 3.1 (August version). https://llama.meta.com. Llama-3.1-8B-Instruct.

OpenAI. 2023. GPT4 (Nov 7 version). https://chat.openai.com/chat. gpt-4-1106-preview.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Yixuan Tang and Yi Yang. 2024. MultiHop-RAG: Benchmarking Retrieval-Augmented Generation for Multi-Hop Queries.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, An- thony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Di- ana Liskovich, Yinghai Lu, Yuning Mao, Xavier Mar- tinet, Todor Mihaylov, Pushkar Mishra, Igor Moly- bog, Yixin Nie, Andrew Poulton, Jeremy Reizen- stein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subrama- nian, Xiaoqing Ellen Tan, Binh Tang, Ross Tay- lor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Ro- driguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine- tuned chat models.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi- dimensional evaluator for text generation.