Performance Trade-offs in Retrieval Augmented Generation Systems for Multi-Hop Question Answering

Lok Yee Joey Cheung
University of Queensland

1 Introduction

Multi-hop question answering (QA) has emerged as a significant challenge in natural language processing (NLP), particularly in the context of generative AI advancements. It involves combining information from multiple relevant documents to derive an answer for each query. While a Retrieval-Augmented Generation (RAG) system is an effective architecture by leveraging retrieval and generation mechanisms for this task, they come with trade-offs. It is typically divided into two stages: first, retrieving the top-10 most relevant documents, and then combining the retrieved documents to generate a comprehensive answer for each query. These trade-offs could stem from the quality of the retrieved documents, the complexity of multi-hop reasoning, balancing between retrieval and generation, cost-effectiveness of LLM models and more.

In this project, I will explore different alternatives for the retrieval and generation stages using rankers, re-rankers and large language models (LLM) from Hugging Face. I will then compare and contrast different combinations and their corresponding performances. A small factoid question answering dataset designed for multi-hop QA tasks is used which contains fact-based questions and short, factual answers as ground truth. It serves as an excellent benchmark for assessing how well a system retrieves relevant documents and performs multi-hop QA. The aim of this project is to find the best overall system which not only retrieves high-quality documents but also performs accurate multi-hop reasoning to generate precise answers.

2 Methodology

2.1 Retrieval Stage

In the first stage retrieval, I have selected a diverse set of models for my experiments, which includes combinations of 3 retrievers and 2 re-rankers. These models are mainly from Beijing Academy of Artificial Intelligence.

2.1.1 Ranker A

Ranker A utilizes the **"BAAI/llm-embedder"**. It is selected for its capability to effectively retrieve relevant information in diverse contexts. It also provides a solid baseline for retrieval [9, 11].

2.1.2 Re-ranker A

Re-ranker A leverages **"BAAI/bge-reranker-base"** to re-rank top-10 documents returned by Ranker A to enhance ranking performance. It is a cross-encoder model introduced at the end of 2023, which is expected to outperform traditional embedding models and enhance ranking performance with increased precision [9].

2.1.3 Ranker B

I chose **"BAAI/bge-m3"** as the second retriever based on the hypothesis that M3-Embedding is the latest and most versatile embedding model. It is described to excel in multi-linguality, multi-functionality and multi-granularity [3]. It is robust in handling over 100 languages and perform dense, multi-vector, and sparse retrieval simultaneously [3]. It, therefore, may provide an optimal performance in this retrieval task.

2.1.4 Re-ranker B

"BAAI/bge-reranker-large" is selected to refine top-10 documents retrieved by ranker B. It differs from re-ranker A by being trained with around 280M more parameters. According to evaluation results in Hugging Face, this re-ranker has slightly better performance than re-ranker A in mean reciprocal ranking [9]. I assume that it could further refine the retrieval results, leveraging its advanced cross-encoder capabilities for higher accuracy.

2.1.5 Ranker C

Ranker C features **"BAAI/bge-large-en-v1.5"**, a popular model on Hugging Face, hypothesized to offer strong performance based on its widespread adoption with over 3 millions download last month. It is the version 1.5 of the model, "BAAI/bge-large-en", which addresses issues with similarity distribution and offers the highest dimensional embeddings [9]. It has

proven to achieve the top performance on the MTEB leaderboard [9].

2.1.6 Re-ranker C

Once again, I re-rank the top-10 documents retrieved by ranker C using **"BAAI/bge-reranker-large"** since it is a lightweighted model with fast deployment and inference [3]. With limited computational resources, it is important to balance between accuracy and efficiency of the retrieval process. It is hypothesized that the result would achieve even higher accuracy for improved ranking than that from pure ranker C.

2.2 RAG Stage

In the second stage, I leverage 4 different LLM from diverse organizations namely Meta, Alibaba Cloud and Mistral AI. Below shows the details of each LLM used:

2.2.1 LLM A

"meta-llama/Llama-2-7b-chat-hf" is a fine-tuned model of Llama 2, designed for dialogue use cases. It is an auto-regressive language model that uses an transformer architecture and reinforcement learning with human feedback [8]. It is well-suited for generating concise answers for complex questions in a Multi-Hop QA setup.

2.2.2 LLM B

"meta-llama/Meta-Llama-3.1-8B-Instruct" is a recently updated versatile model designed for both commercial and research use across multiple languages. It also supports the generation of synthetic data and model distillation to enhance the performance of other models [1]. It is expected to be flexibility in real-world applications and deliver robust performance for precise multi-step question answering.

2.2.3 LLM C

"Qwen/Qwen2-7B-Instruct" is a instruction-tuned model with 7 billions parameters by Alibaba Cloud. It has proven to outperform most language models including Llama 3 across a series of benchmarks for text generation and language understanding, especially in English and Chinese datasets [7].

2.2.4 LLM D

I choose **"mistralai/Mistral-7B-Instruct-v0.3"** as Mistral 7B is a cutting-edge model that surpasses even larger models like Llama 2 (13B) in NLP tasks [5]. It uses innovative techniques like grouped-query attention and sliding window attention to improve inference speed and memory usage. This variant is fine-tuned for instruction-following tasks, making it an ideal choice for high-performance text generation [5].

Table 1: RAG Model and LLM Setup for the Second Stage.

RAG Model Setup		
RAG Model	**Ranker**	**LLM**
RAG A	Best Ranker	LLM A
RAG B	Best Ranker	LLM B
RAG C	Best Ranker	LLM C
RAG D	Best Ranker	LLM D
RAG E	Second Best Ranker	LLM A
RAG F	Second Best Ranker	LLM B
RAG G	Second Best Ranker	LLM C
RAG H	Second Best Ranker	LLM D

2.3 Model Setup

Different rankers and LLMs exhibit varying strengths and weaknesses based on their architectures and training data. Through exploring various combinations, I assess how these differences impact the overall performance of the RAG models and the possible trade-offs. As shown in Table 1, I combined the the four LLM with 'best' ranker in RAG models A, B, C, and D and with the 'second-best' ranker in RAG models E, F, G, and H. The analysis of the "best" rankers will be discussed in section 3.

In my solution, I used LlamaIndex to facilitate the RAG pipeline, which combines the strengths of document retrieval systems with LLMs to generate answer to queries. First, the LlamaIndex processes the query, efficiently indexes and searches through the large corpus of documents dataset. It then retrieves the most relevant pieces of information. Once the relevant documents are retrieved, they are passed to the LLM to generate a coherent and contextually accurate answer. In my RAG setup, I leveraged LlamaIndex's integration with several pre-trained LLMs and fine-tune them for multi-hop QA.

2.4 Evaluation Metrics

2.4.1 Retrieval Evaluation

To evaluate the retrieval results of each rankers in the first stage, I utilize 5 metrics which are discussed as follows:

(i) Hits@10: It measures the fraction of relevant items from the gold list that are included in the top 10 retrieved results. The higher, the better the model's performance.

(ii) Hits@4: Similar to Hits@10, it measures how often a relevant result is retrieved in the top 4.

(iii) MAP@10: It computes the mean of the average precision calculated at each rank up to 10 for all queries [2]. Precision is the ratio of relevant items retrieved. A higher score reveals that relevant

Table 2: Performance Evaluation Statistics of Rankers and Rerankers in First Stage Retrieval.

Multi-Hop QA Retrieval Stage					
Model	**Hits@10**	**Hits@4**	**MAP@10**	**MRR@10**	**NDCG@10**
Ranker A	0.5348	0.4018	0.1408	0.3299	0.5348
Reranker A	0.6204	0.5410	0.2149	0.4936	0.6204
Ranker B	0.6856	0.5322	0.2007	0.4489	0.6856
Reranker B	0.7481	0.6563	**0.2703**	0.6016	0.7481
Ranker C	0.6896	0.5463	0.1997	0.4520	0.6896
Reranker C	**0.7490**	**0.6661**	0.2687	**0.6029**	**0.7490**

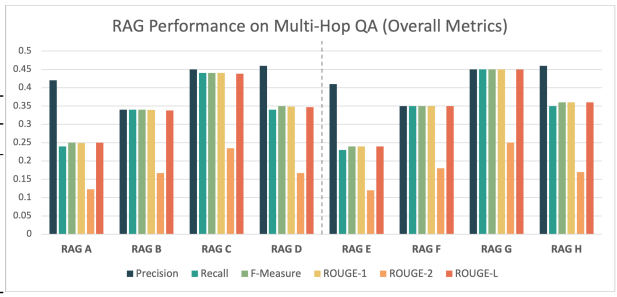

Figure 1: Performance comparison of RAG systems using different LLMs on a multi-hop QA task.

items are ranked higher within the top 10 [2].

(iv) MRR@10: Mean Reciprocal Rank emphasizes how early the first relevant result appears [2]. MRR@10 measures the average of the reciprocal rank of the first relevant item in the top 10 results for all queries [2]. Higher MRR means relevant results tend to be found at the top of the ranking list.

(v) NDCG@10: This metric measures the quality of top 10 documents retrieved by taking into account both the relevance of the items and their positions [10]. The higher the score, the more effective the ranking model.

2.4.2 RAG Evaluation

The evaluation of the RAG models is crucial to reflect their performance in multi-hop question answering, and several key metrics are used for this purpose:

(i) Precision: It measures the fraction of retrieved answers that overlap with the gold standard answers [4].

(ii) Recall: It assesses the proportion of ground truth answer that is successfully retrieved [4].

(iii) F-measure: It combines precision and recall to provide a balanced measure of model's performance [4].

(iv) ROUGE Score: It is a set of metrics that evaluate the quality of generated text. I include ROUGE-1 and ROUGE-2 which measure the overlap of unigram and bi-grams between the generated output and reference respectively [6]. ROUGE-L captures the longest common sub-sequence (LCS)[6].

3 Experiments

In this section, we conduct a comprehensive analysis of the performance of rankers in Stage One retrieval as well as RAGs across query types in Stage Two text generation. The figures illustrate key metrics which highlighting the strengths and weaknesses of each model. We aim to draw insights into how each model or combination of models retrieve relevant documents and responds to diverse query structures.

3.1 Retrieval Analysis

According to Table 2, it illustrates the statistical evaluation on rankers performance when retrieving relevant top-10 documents. Reranker B and C consistently outperforms the other models across almost all metrics, with Reranker C achieving slightly higher scores in Hits@10 (0.75), Hit@4 (0.67), MRR@10 (0.60) and NDCG@10 (0.75), as shown by bold text. This indicates that highly relevant results are present at the top few retrieved items using Reranker C. Reranker B ("BAAI/bge-m3" reranked with "BAAI/bge-reranker-large") is renowned for handling longer and support multilingual texts while Reranker C ("BAAI/bge-large-en-v1.5" reranked with "BAAI/bge-reranker-large") addresses issues with similarity distribution. It also offers improved retrieval capabilities, performing effectively without requiring instruction-based prompts. This could be the reasons why these two models excel.

On the contrary, Ranker A performed the weakest in all aspects, having the lowest score across all the metrics, particularly with low Hits@10 (0.5348) and MAP@10 (0.1408). This suggests that the initial retrieval stage was less effective at identifying and ranking relevant items. This could be due to its relatively simple architecture with fewer parameters used for pre-training.

The Reranker models show better performance than the Ranker models across the board which proves and emphasizes the importance of reranking in improving answer retrieval accuracy. Reranking helps to prioritize the most contextual and relevant documents to the top of the list. Consequently, we have chosen the best retriever, Re-ranker C, to produce the input for the next stage, RAG A to D. Reranker B, the 'second best' ranker is further used as the input for RAG E to H.

Table 3: Overall Performance Evaluation Statistics of RAG in Second Stage Text Generation.

Multi-Hop QA Text Generation Stage						
Model	**Precision**	**Recall**	**F1 Score**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
RAG A	0.42	0.24	0.25	0.25	0.12	0.25
RAG B	0.34	0.34	0.34	0.34	0.17	0.34
RAG C	0.45	0.44	0.44	0.44	0.23	0.44
RAG D	**0.46**	0.34	0.35	0.35	0.17	0.35
RAG E	0.41	0.23	0.24	0.24	0.12	0.24
RAG F	0.35	0.35	0.35	0.35	0.18	0.35
RAG G	0.45	**0.45**	**0.45**	**0.45**	**0.25**	**0.45**
RAG H	**0.46**	0.35	0.41	0.36	0.17	0.36

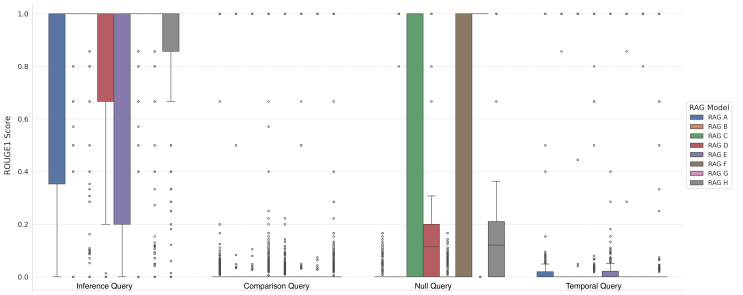

Figure 2: Distribution of per-query ROUGE-1 score for different query types.

3.2 RAG Analysis

3.2.1 Overall Average Performance

Figure 1 presents the performance of RAG models A to H across all query types, with key metrics displayed as colored bars. A dashed line separates models that use different retrieval models. Table 3 provides a statistical reference for Figure 1.

We observe that RAG G has the best performance as it achieve the highest scores across Recall, F-Measure and all ROUGE scores, slightly outperforms RAG C. Both RAGs use Qwen2 which generate content that closely align with the reference answers, as measured by overlapping n-gram and LCS in ROUGE. Since this model is fine-tuned on instructions and support lengthy context with its improved tokenizer, it follows prompts more accurately and generate exact keywords from documents in question-answering tasks [7]. Given that RAG G does not uses best ranker but delivers the most outstanding results, it further proves that the optimal combination of ranker and LLM does not guarantee the best RAG performance.

Unexpectedly, RAG D and RAG H which both use Mistral-7B-Instruct, have the best precision for overall query at 0.46. This model seems to have an edge over the others in terms of giving high proportion of relevant words in relation to reference answer. It might have prioritized accuracy over completeness, leading to fewer false positives but more false negatives.

On the other hand, RAG A and RAG E have relatively weak performance, with low recall, F1 scores, and ROUGE metrics. Unlike Qwen-2-7B in RAG C and G, Llama 2 in these RAG A and E is not instruction-tuned which may result in less accurate responses in text generation. Moreover, it has not been optimized specifically for text generation benchmarks as it is more likely designed for dialogue-based use cases [8]. This may explain why these two RAGs achieve almost half the ROUGE scores of RAG C and G.

3.2.2 Per-Query Level Performance

The box plot in Figure 2 shows the distribution of per-query ROUGE-1 scores for different query types and RAG models. In QA tasks, the primary goal is to ensure that the concise terms in the response align with the expected output. Therefore, I chose to plot per-query ROUGE-1 scores which measures the overlap of unigrams between the generated and the reference answers for each query. Per-query scores distribution allows for a more granular analysis of model behavior e.g. consistency across a diverse range of tasks.

For inference queries, at least 50% of the data points have ROUGE-1 scores near or equal to 1.0 across nearly all models. RAG models B, C, F, and G, which use Llama-3.1 and Qwen2, show less variance. The median ROUGE-1 score consistently remains at 1, indicating that more than half of the responses for inference queries are perfect matches. While it's expected for Qwen2, the best model, to achieve a high ROUGE-1 score, it's surprising that Llama-3.1, which has a lower mean ROUGE-1 overall, also performs favorably. As for RAG A and E, there are some queries performing poorly, dropping down to around 0.2 to 0.3 for the lower quarter of cases.

As for null queries, performance varies widely between models, from very poor (mostly below 0.2 for RAG A) to quite good (median equals 1 with low variance for RAG G). It further suggests that Qwen2 in RAG C and G are effectively trained to handle situations with incomplete or unclear information. Poor performance on comparison and temporal queries across all models are observed, with mean ROUGE-1 scores ranging from 0.022 to 0.183. The median is consistently 0 but the mean is non-zero, which suggests skewed distribution and more than half of the responses fail completely. In general, there is high variability in performance across query types regarding ROUGE-1 scores.

3.2.3 Average Performance across Query Types

Figure 3 to Figure 7 demonstrate the performance evaluation of RAG A to D across different query types.

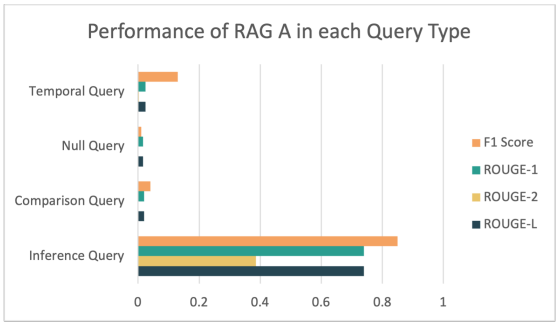

Figure 3: Performance comparison of RAG A across different query types.

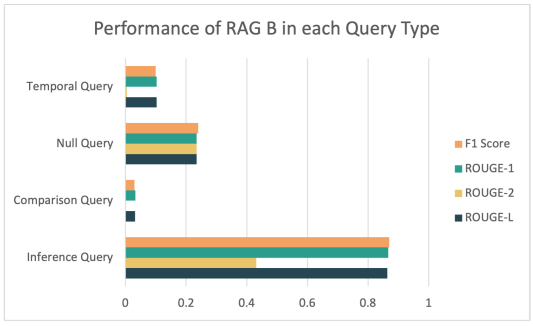

Figure 4: Performance comparison of RAG B across different query types.

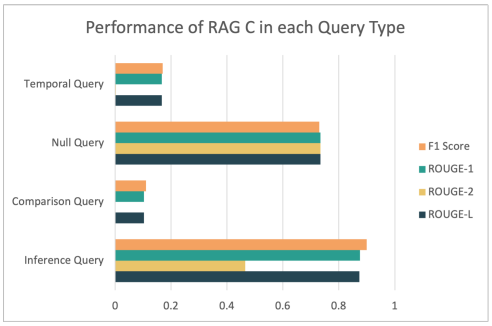

Figure 5: Performance comparison of RAG C across different query types.

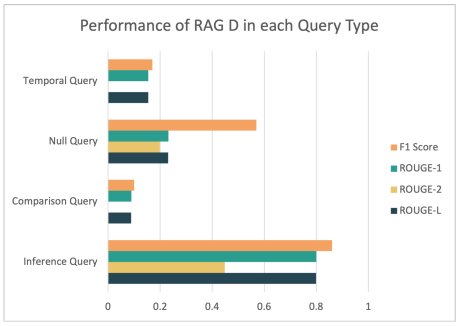

Figure 6: Performance comparison of RAG D across different query types.

We focus on analysing RAG A to D that generate text using LLM A to D respectively on results retrieved by reranker C. RAG E to H that uses the same list of LLMs but on a slightly different retrieved results have similar patterns.

Inference queries generally perform best across all RAG models, while temporal and comparison queries yield lower scores. RAG B, C, and D slightly outperform RAG A on comparison queries, likely due to recent updates to Llama-3.1, Qwen2, and Mistral-7B that emphasize multi-hop reasoning and broader training on diverse tasks. Temporal queries remain challenging for all models. As for null queries, RAG C and D appear to outperform RAG A and B, especially in the F1 Score. This indicates that Qwen2 in RAG C and Mistral-7B in RAG D have been well-trained on handling incomplete or ambiguous information. The weakest model, Llama 2 model in RAG A is only capable of handling inference queries as it is the least update model and tailored for dialogues use cases. As a result, Qwen2-7B-Instruct is the top generator, aided by Bge-large-en-v1.5 and Bge-reranker-large (Reranker C).

4 Conclusion

In this project, this study highlights the critical interplay between rankers and LLMs in the performance of RAG systems. Different experiments were carried out to combine rankers with LLMs from diverse organizations and capabilities. We gained valuable insights into each of their strengths and weaknesses, as well as the importance of the selection of choices that significantly influence the quality of generated answers in a multi-hop QA tasks. One of the biggest discovery was that even though the 'best' first stage ranker, bge-reranker-large was combined with the best second stage LLM, Qwen2-7B-Instruct, they did not necessarily produce the best overall result. Per-query level analysis also reveals that models tend to perform better on inference queries and null queries but fall short of handling comparison and temporal queries. It is also obvious that RAG C and G which utilize Qwen2 perform well on null queries, while Llama-2 in RAG A struggles. Since we were limited to only test the four LLMs on the two rankers, more alternative combinations could be further explored for future improvement in retrieval and generation optimization tasks.

References

[1] AI@Meta. "Llama 3 Model Card". In: (2024). URL: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

[2] Laura Carnevali. *Evaluation measures in information retrieval*. URL: https://www.pinecone.io/learn/offline-evaluation/#Mean-Reciprocal-Rank-(MRR).

[3] Jianlv Chen et al. "Bge m3-embedding: Multilingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation". In: *arXiv preprint arXiv:2402.03216* (2024).

[4] George Hripcsak and Adam S Rothschild. "Agreement, the f-measure, and reliability in information retrieval". In: *Journal of the American medical informatics association* 12.3 (2005), pp. 296–298.

[5] Albert Q Jiang et al. "Mistral 7B". In: *arXiv preprint arXiv:2310.06825* (2023).

[6] Chin-Yew Lin. "Rouge: A package for automatic evaluation of summaries". In: *Text summarization branches out*. 2004, pp. 74–81.

[7] "Qwen2 Technical Report". In: (2024).

[8] Hugo Touvron et al. "Llama 2: Open foundation and fine-tuned chat models". In: *arXiv preprint arXiv:2307.09288* (2023).

[9] Shitao Xiao et al. *C-Pack: Packaged Resources To Advance General Chinese Embedding*. 2023. arXiv: 2309.07597 [cs.CL].

[10] Wang Yining et al. "A theoretical analysis of ndcg ranking measures". In: *Proceedings of the 26th annual conference on learning theory*. 2013.

[11] Peitian Zhang et al. *Retrieve Anything To Augment Large Language Models*. 2023. arXiv: 2310.07554 [cs.IR].

A Appendix

The RAG performance of RAG E, F, G, H across four types of queries:

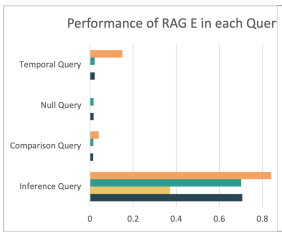

Figure 7: Performance comparison of RAG E across different query types.

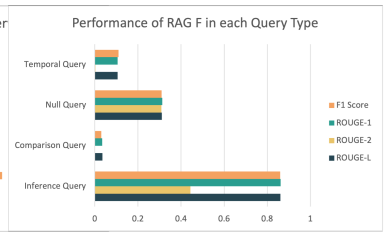

Figure 8: Performance comparison of RAG F across different query types.

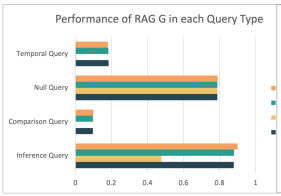

Figure 9: Performance comparison of RAG G across different query types.

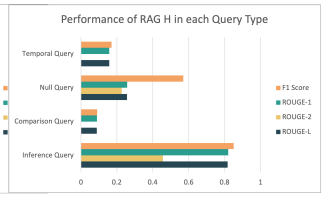

Figure 10: Performance comparison of RAG H across different query types.