ACM-ICAIF ’24 FinanceRAG Challenge

Jing Wang[*]

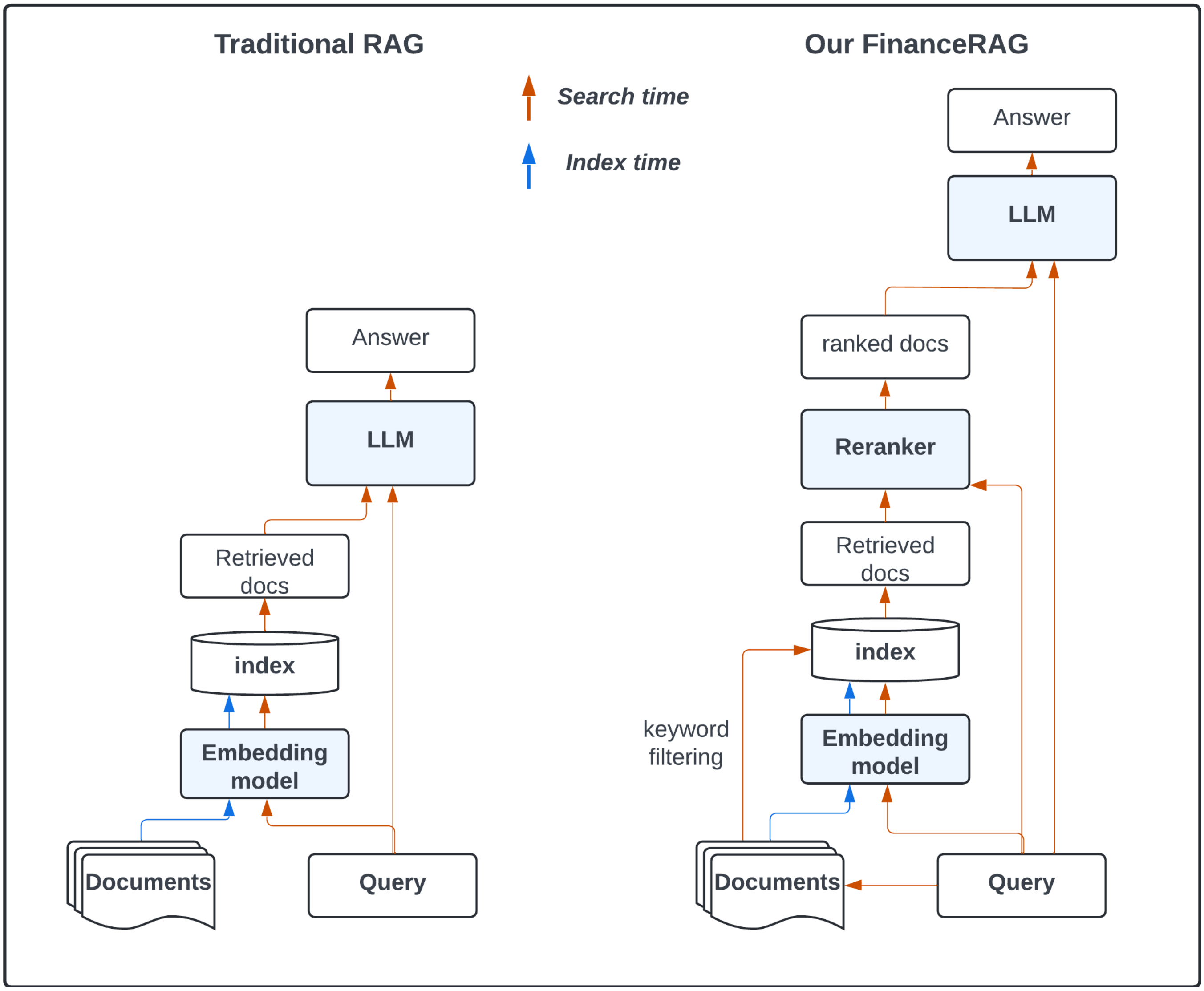

Figure 1: Overview of the RAG workflow

Abstract

In this project, we integrate hybrid search with document reranking to enhance Retrieval-Augmented Generation (RAG) systems. During the hybrid search phase, we retrieve documents that are not only semantically similar to the query but also contain specific keywords inferred from the query. Subsequently, the reranking step leverages a cross-attention mechanism to further assess the semantic similarity of the retrieved documents. The top-k most relevant documents identified through this combined approach are then used as contexts for a large language model (LLM) to generate the final answer.

[*] jing.wang@Accrete.ai

Conference’24, Nov 2024, NYC, NY, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Keywords

Retrieval-Augmented Generation, Hybrid Search, Text Embeddings, Approximate Nearest Neighbor (ANN) Search, Reranking, Generative Language Models

ACM Reference Format:

Jing Wang. 2024. ACM-ICAIF '24 FinanceRAG Challenge. In . ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Retrieval-Augmented Generation (RAG) frameworks leverage external knowledge sources to provide rich and accurate contexts for applications utilizing LLMs. This study introduces a method that synergistically combines hybrid search techniques with advanced reranking strategies to optimize document retrieval for RAG tasks. By employing dual-criteria and funnel-style retrieval, our approach enhances both the relevance and specificity of information retrieved from extensive corpora.

2 Method

The objective of Task 1 is to retrieve the most relevant document chunks (contexts) from a substantial document corpus in response to a given query. This process aims to maximize retrieval accuracy, specifically measured by the normalized Discounted Cumulative Gain at rank 10 (nDCG@10). The evaluation framework assesses the system's ability to rank retrieved contexts effectively compared to established ground truth relevance.

2.1 Retrieval

1. Data Preprocessing

Each document consists of an article title and chunked text. We remove unnecessary markdowns and punctuation and correct invalid bytes. Texts that are excessively long and exceed the token limit of the embedding model are truncated to ensure compatibility with subsequent processing stages.

2. Query Embedding Generation

Each input query is transformed into a high-dimensional vector representation (embedding) using OpenAI's `text-embedding-ada-002` model. This embedding captures the semantic nuances of the query, enabling effective similarity comparisons with document embeddings.

3. Initial Document Retrieval

The generated query embedding is utilized to perform a nearest neighbor search within a pre-indexed vector database containing embeddings of all document chunks in the corpus. This database is optimized for rapid similarity searches, facilitating the retrieval of an initial set of candidate documents that are semantically aligned with the query. To enhance retrieval precision, a hybrid search approach is employed that combines semantic similarity with keyword matching. This ensures that the retrieved documents are not only contextually relevant but also contain specific keywords inferred from the query.

4. Reranking Mechanism

The initial set of retrieved documents undergoes a reranking process using the ColBERT model. This step assesses the finer details of semantic similarity between the query and each document chunk, allowing the system to prioritize documents that exhibit higher relevance. Each document chunk is assigned a similarity score based on cross-attention evaluation. These scores are used to reorder the documents, ensuring that the most pertinent contexts are ranked higher.

5. LLM Answer Generation

The top-k documents, as determined by the reranking scores, are aggregated to serve as contextual input for subsequent processing tasks, such as answer generation by a LLM.

2.2 Retrieval Evaluation

The effectiveness of the retrieval system is primarily measured using nDCG@10, which evaluates the ranking quality of the top 10 retrieved document chunks against the ground truth on a subset of data. This metric considers both the position and relevance of each retrieved document, providing a comprehensive assessment of the system's retrieval accuracy. Table 1 provides the full metrics for the retrieval step across all seven datasets.

Table 1: Evaluation metrics across all datasets

Dataset	NDCG@10	MAP@10	Recall@10
ConvFinQA	0.522	0.498	0.595
FinDER	0.425	0.361	0.583
FinQA	0.331	0.311	0.395
FinQABench	0.871	0.850	0.933
FinanceBench	0.699	0.609	0.922
MultiHiertt	0.322	0.291	0.312
TATQA	0.801	0.762	0.922

2.3 Generation

In the answer generation step, we deployed OpenAI's `O1` model and Anthropic `Claude-3.5-Sonnet`. The quality of the generated responses varies based on the specific LLM and its application context. We evaluate the generated answers using the RAGAS framework, which will be detailed elsewhere.

3 Conclusion

In this challenge, we developed and evaluated a robust retrieval and reranking system designed to identify the most relevant document contexts from large-scale corpora in response to user queries. By integrating hybrid search techniques with a reranking mechanism, our approach effectively balances semantic similarity and keyword specificity, ensuring that retrieved documents are both keyword-anchored and contextually pertinent.

References

Chanyeol Choi, Jy-Yong Sohn, Yongjae Lee, Subeen Pang, Jaeseon Ha, Hoyeon Ryoo, Yongjin Kim, Hojun Choi, Jihoon Kwon. (2024). ACM-ICAIF '24 FinanceRAG Challenge. Kaggle.