Engineering Advances, 2025, 5(4), 166-171
https://www.hillpublisher.com/journals/ea/
ISSN Online: 2768-7961

A Study on Question-answering Retrieval System Based on Large Language Models and RAG Technology

Xingang Wu*, Yajie Huang, Yu Li, Yifei Duan, Xiaowei Li, Mingze Wu

Yantai Institute of Technology, Yantai 265699, Shandong, China.

How to cite this paper: Xingang Wu, Yajie Huang, Yu Li, Yifei Duan, Xiaowei Li, Mingze Wu. (2025) A Study on Question-answering Retrieval System Based on Large Language Models and RAG Technology. *Engineering Advances*, 5(4), 166-171.
DOI: 10.26855/ea.2025.10.007

Received: September 24, 2025
Accepted: October 21, 2025
Published: November 17, 2025

***Corresponding author:** Xingang Wu, Yantai Institute of Technology, Yantai 265699, Shandong, China.

Abstract

This paper aims to explore and implement a question-answering system utilizing Retrieval-Augmented Generation (RAG) technology with Large Language Models (LLMs)to enhance the accuracy and reliability of information retrieval. The paper first compares and analyzes the performance characteristics and application scenarios of mainstream large models. It then systematically reviews the evolution of RAG technology and its key optimization strategies. Building on this foundation, the paper delves into the development of question-answering retrieval systems, from traditional template matching to end-to-end deep learning models, and finally to the current integration of large models and RAG. The core section of this paper designs and implements a question-answering retrieval system based on large models and RAG technology, detailing its system architecture, workflow, and key implementation technologies using LangChain and vector databases. Through the construction of experimental prototypes and comparative testing, the system's significant advantages in answer accuracy, factual consistency, and resistance to "hallucinations" compared to pure large model solutions are verified. Finally, the paper summarizes the current system's shortcomings and looks forward to future research directions.

Keywords

Large Language Models; Retrieval-Augmented Generation; Question-Answering System; Vector Database; LangChain

1. Introduction

In recent years, large language models (LLMs)represented by GPT, LLaMA, and ChatGLM have achieved breakthrough progress in numerous natural language processing tasks due to their powerful language generation and context understanding capabilities. However, when applied to scenarios that require precise factual basis, such as knowledge Q&A and content summarization, LLMs reveal three core bottlenecks, including hallucination generation, outdated knowledge, and limited domain expertise [1].

To address the above issues, Retrieval-Augmented Generation (RAG)technology has emerged. Retrieval-Augmented Generation (RAG) represents a major advancement in natural language processing (NLP) [2]. Its core idea is to retrieve relevant information fragments from external knowledge bases (such as documents, databases, and knowledge graphs) before the LLM generates answers and provide them as context prompts to the LLM, thereby guiding the model to generate content that is more factual, accurate, and traceable. Research on question-answering systems based on large models and RAG is of great theoretical value and broad practical significance for building a new generation of trustworthy, reliable, and updatable intelligent applications.

2. Review of Related Technologies

2.1 Comparison of Large Language Models

Large models serve as the “brain” of RAG systems, and their selection directly determines the system’s ultimate performance. The following is a comparative analysis of mainstream open-source and closed-source models from multiple dimensions.

Table 1. Comparative analysis of mainstream open-source and closed-source models from multiple dimensions

Model Name	Type	Main Characteristics	Advantages	Disadvantages	Applicable Scenarios
GPT-4 Turbo	Closed Source	General-purpose model Large context window Strong reasoning capability	Top-tier performance Mature ecosystem	High cost Data privacy concerns	Commercial applications requiring extremely high performance
ChatGLM3-6B	Open Source	Optimized for Chinese & English Efficient inference framework	Lightweight Suitable for Chinese scenarios Supports private deployment	Overall capability lags behind top closed-source models	Chinese enterprise applications, resource-constrained environments
LLaMA 2-7B/1.3B	Open Source	Excellent architecture design Active open-source community	Good balance of performance and efficiency Abundant fine-tuning resources	Native Chinese support not as strong as GLM	Academic research, scenarios requiring customized fine-tuning
ERNIE (Baidu)	Closed/Open Source	Developed by Baidu Deep Chinese understanding Knowledge-enhanced	Tight integration with Baidu ecosystem Strong Chinese capability	Performance gap between open-source and closed-source versions	Domestic business integration, Chinese language tasks

Conclusion of Model Selection: Considering performance, cost, Chinese support, and the need for private deployment, this study prioritizes ChatGLM3-6B as the core generation model in the system prototype implementation, which achieves a good balance between parameter size, Chinese-English capabilities, and deployment costs.

2.2 Current Development of RAG Technology

RAG technology was initially proposed by Lewis et al. in 2020 [3]. It combines information retrieval with sequence-to-sequence models, enabling models to utilize external knowledge bases when generating answers, thereby improving the accuracy and factuality of the generated content. This core idea laid the foundation for the subsequent development of RAG. Currently, there are numerous research results on RAG in the industry. For example, Chen et al. (2023) systematically studied the impact of RAG on different LLMs to identify potential bottlenecks in RAG capabilities, which is crucial for optimizing RAG systems [4]. Danuarta et al. (2024) explored the potential of RAG methods in educational chatbots, enhancing the accuracy, relevance, and security of educational Q&A by integrating LLMs with real-time external data [5]. Muludi et al. (2022) demonstrated the application breakthrough of RAG methods in document Q&A, emphasizing their capability to overcome traditional Q&A system barriers [6].

Overall, the research progress of RAG technology is not static, and its development has gone through three stages [1]:

• Naive RAG (the classic "retrieve-then-read" pipeline),
• Advanced RAG (introducing various optimization strategies),
• Modular RAG (modularizing and intelligentizing the RAG process).

Table 2. Comparison of Core Features of Three Generations of RAG Technology

Dimension	**Naive RAG (Traditional/Basic)**	**Advanced RAG (Enhanced)**	**Modular RAG (Modularized)**
Core Architecture	Linear "Retrieve-Generate" pipeline	Enhanced pipeline with front-end/back-end optimization	Intelligent orchestration, modular, customizable
Retrieval Method	Basic model retrieval (e.g., BERT)	Hybrid retrieval, re-ranking, refined strategies	Recursive retrieval, intelligent routing, multi-step dynamic retrieval
Processing Logic	Fixed, simple	Fixed, optimized	Flexible, adaptive (Agent/routing-based)
Core Innovation	Proposed basic paradigm	Process optimization	Architectural revolution
Advantages	Simple implementation; low cost	High recall; improved answer quality	Extremely flexible; handles complex tasks
Limitations	Low retrieval precision; hallucinations; no multi-hop reasoning	Relatively fixed process; limited complex logic handling	Complex system; high design/debugging difficulty

2.3 Development of Question-Answering Retrieval Systems

The development of question-answering retrieval systems can be summarized as follows:

• Rule-based, keyword matching, and template-based methods: Early systems mainly relied on preset rules and information retrieval techniques, such as TF-IDF or BM25 algorithms, to find information matching the query from structured databases or unstructured texts. However, they have limitations in handling complex semantics, multi-hop reasoning, and generating natural language answers.

• End-to-end deep learning models: Models like BiDAF and BERT-based QA models directly learn answers from question-text pairs, but their knowledge is confined within the model parameters.

• Open-domain question answering: Combining large-scale search engines with reading comprehension models, but the process is complex, and each module is trained independently.

• Large models and RAG-integrated question-answering systems: This is the current mainstream paradigm. It leverages the powerful generalization and generation capabilities of large models, combined with the real-time and accurate external knowledge provided by RAG, to achieve end-to-end, updatable, high-performance question-answering systems. The latest research focuses on optimizing various components of RAG systems, including smarter retrieval strategies (such as multi-granularity indexing, hybrid retrieval, and multi-hop retrieval), context management and knowledge graph integration, and model architecture optimization.

3. System Design and Implementation

3.1 System Architecture

The question-answering retrieval system designed in this paper adopts a layered architecture, which mainly includes the data preprocessing layer, the RAG retrieval layer, the generation layer, and the application layer. The core workflow of the system is shown in the figure below:

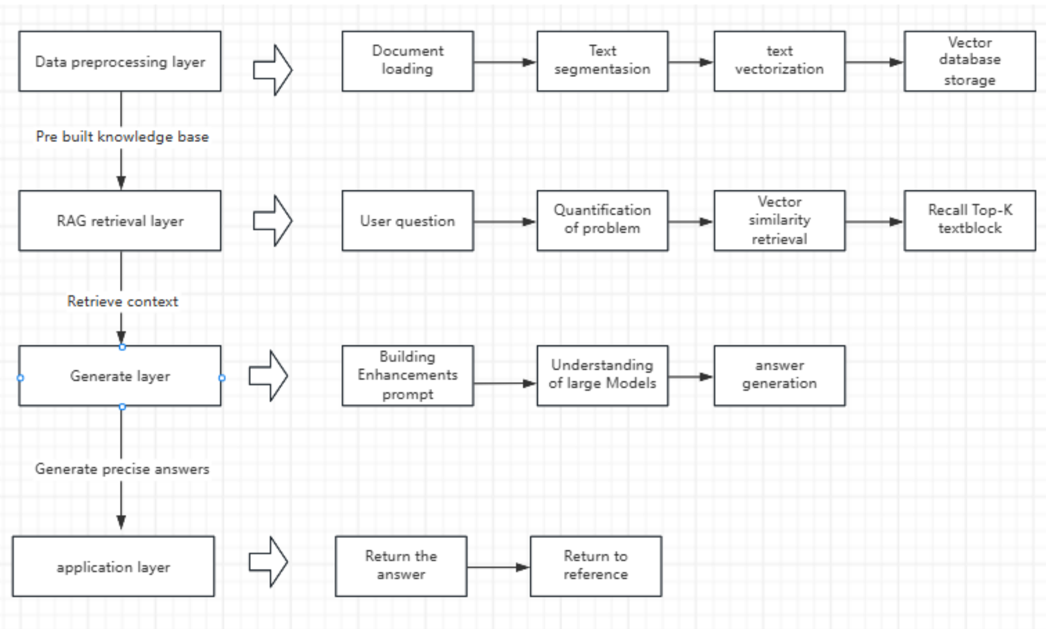

Figure 1. The core workflow of the system.

3.2 Detailed Core Modules

3.2.1 Data Preprocessing and Vectorization Module

• Function: To process raw unstructured documents (such as PDF, Word, TXT) into vectors available for retrieval.

• Implementation:

• Document Loading and Splitting: Using the document loader of LangChain [7] and employing the ‘Recursive Character Text Splitter’ for intelligent splitting to ensure semantic integrity.

• Vector Embedding: Selecting the high-performance Chinese embedding model`text2vec`to convert text chunks into vectors.

• Vector Storage: Storing vectors along with metadata (such as source and page number) into Chroma/Milvus vector databases.

Note: The preprocessing steps for video files are as follows:(1) Utilizing FFmpeg technology to process video files for frame extraction and conversion into images;(2) Using the open-source OCR project PaddleOCR by Baidu’s PaddlePaddle team for text extraction;(3) Vectorizing the extracted text based on the aforementioned steps.

3.2.2 RAG Retrieval Module

• Function: To quickly find the most relevant text fragments from the vector database according to user questions.
• Implementation:
• Query Encoding: Using the same model as document embedding to encode user questions into vectors.
• Similarity Search: Conducting approximate nearest neighbor search in the vector database to return the top-K most similar text chunks and their metadata.
• Re-ranking: Introducing more sophisticated models such as cross-encoders to re-rank the initial search results, further enhancing the accuracy of the top results.

3.2.3 Prompt Engineering and Answer Generation Module

• Function: To integrate the retrieved context with user questions to construct effective prompts that guide the large model to generate high-quality answers.
• Implementation:
• Prompt Template Design: Designing structured system prompts with parameters`{context}` to clarify the model's role and task.
• Context Injection: Filling the text chunks retrieved by RAG into the`{context}` placeholder.
• LLM Invocation: Sending the constructed prompt to ChatGLM3 and other large models to obtain generated answers.

3.3 System Implementation

The system prototype is implemented based on the LangChain framework and Chroma vector database in this study. LangChain provides a rich set of component chains, greatly simplifying the orchestration of the RAG process. The system backend is built using Python Flask/FastAPI to construct RESTful APIs, and the frontend provides a simple web interface to support users in uploading documents and conducting Q&A.

The code, please refer to: https://github.com/winstonwuxingang/RAG_QA

4. Experiments and Analysis

4.1 Experimental Settings

• Dataset: A self-constructed small-scale domain-specific knowledge document set (the life and literature of Qi Jiguang, approximately 30 documents) was used as the external knowledge base.
• Baseline Models:
• Baseline A: Pure ChatGLM3-6B (without external retrieval).
• Baseline B: Naive RAG (using simple vector retrieval).
• Our System: Advanced RAG (including optimized Prompt and preprocessed text splitting).
• Evaluation Metrics: Manual evaluation was adopted, scoring from three aspects—answer accuracy, factual consistency (resistance to hallucination), and answer relevance—using a 5-point scale.

4.2 Results and Analysis

4.2.1 Analysis Conclusions

• Limitations of Pure LLMs: Baseline A scored the lowest in factual consistency, frequently generating "hallucination" content inconsistent with domain knowledge, which validates the necessity of incorporating external knowledge.
• Effectiveness of RAG: Both RAG systems outperformed pure LLM in factual consistency and answer accuracy, showing RAG’s core value.
• Gains from Optimization Strategies: Our system, with more refined data preprocessing and Prompt engineering, outperformed Naive RAG in all metrics, indicating that the optimization measures of Advanced RAG can bring tangible performance improvements.

Table 3. System Performance Manual Evaluation Results (Average Scores)

Model	Answer accuracy	Factual consistency	Answer relevance
Baseline A (pure LLM)	2.8	2.5	3.5
Baseline B (Naive RAG)	3.6	3.8	3.9
Our system (Advanced RAG)	4.2	4.4	4.3

5. Conclusion and Future Work

This paper systematically investigates the question-answering system integrating large language models (LLMs)with Retrieval-Augmented Generation (RAG)technology. Through comparison, it selects a suitable mid-sized open-source model, reviews RAG evolution, and successfully designs an end-to-end question-answering prototype. Experiments have proven that the system can effectively leverage external knowledge to significantly enhance the factual accuracy and reliability of answers.

While the current system has demonstrated promising results, there is still room for improvement. Future work will focus on the following directions:

• Hybrid Retrieval: Combining keyword-based retrieval (e.g., BM25) with vector-based retrieval to improve recall and robustness of the retrieval process.

• Iterative Retrieval and Agents: Enabling the system to perform multi-step reasoning and iterative querying, which will allow it to tackle more complex questions by breaking them down into simpler sub-questions and retrieving relevant information incrementally.

• Integration of RAG with Fine-tuning: Exploring the combination of RAG and parameter-efficient fine-tuning techniques to better adapt large models to specific domain language styles and knowledge structures, especially in niche areas where domain expertise is crucial.

• Knowledge Graph Integration: Combining structured knowledge (knowledge graphs) with unstructured text(documents)to create a more powerful knowledge base, which can provide richer context and improve the quality of generated answers.

In summary, the integration of large language models and RAG technology holds great potential for advancing question-answering systems. By addressing the limitations of pure LLMs and enhancing the system's ability to utilize external knowledge, we can move towards more reliable, accurate, and up-to-date intelligent applications. Future research will continue to explore optimizations and innovations to further improve the performance and applicability of such systems.

References

[1] Jiang P, Ouyang S, Jiao Y, Zhong M, Tian R, Han J. A Survey on Retrieval And Structuring Augmented Generation with Large Language Models. arXiv [Preprint] 2025 [cited 2025 Sep 12]. Available from: http://arxiv.org/abs/2509.10697

[2] Oche AJ, Folashade AG, Ghosal T, Biswas A. A Systematic Review of Key Retrieval-Augmented Generation (RAG) Systems: Progress, Gaps, and Future Directions. arXiv [Preprint] 2025 [cited 2025 Jul 25]. Available from: http://arxiv.org/abs/2507.18910

[3] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv [Preprint] 2020 [cited 2020 May 22]. Available from: http://arxiv.org/abs/2005.11401

[4] Chen J, Lin H, Han X, Sun L. Benchmarking Large Language Models in Retrieval-Augmented Generation. arXiv [Preprint] 2023 [cited 2023 Sep 4]. Available from: http://arxiv.org/abs/2309.01431

[5] Danuarta L, Mawardi VC, Lee V. Retrieval-Augmented Generation (RAG) Large Language Model For Educational Chatbot. In: 2024 Ninth International Conference on Informatics and Computing (ICIC) [Internet]. Medan, Indonesia: IEEE; 2024 [cited 2025 Apr 15]. p. 1-6. Available from: https://doi.org/10.1109/ICIC64337.2024.10957676

[6] Muludi K, Fitria KM, Triloka J, Sutedi. Retrieval-Augmented Generation Approach: Document Question Answering using Large Language Model. Int J Adv Comput Sci Appl. 2024;15(3). Available from: https://thesai.org/Publications/ViewPaper?Volume=15&Issue=3&Code=IJACSA&SerialNo=46

[7] LangChain [Internet]. [place unknown]: LangChain Inc.; c2024. LangChain overview; [cited 2024 Date]. Available from: https://www.langchain.com/

[8] Du Z, Qian Y, Liu X, Ding M, Qiu J, Yang Z, et al. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. arXiv [Preprint] 2021 [cited 2021 Mar 18]. Available from: http://arxiv.org/abs/2103.10360