Malwal Shreyas Ravindra, 2026, 14:5
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Retrieval-Augmented Generation For Document Question Answering: A Review Of Retrieval, Embedding, Vector Search And Grounded Answer Generation

Nalawade Soham Sandip[1]Prof. J. M. Patade[2]Prof. M. C. Kolse[3]Prof. S. K. Said[4]
Department of Artificial Intelligence and Data Science
Jaihind College of Engineering
Pune, Maharashtra, India

Abstract—The rapid growth of digital documents creates a need for systems that can find relevant information and answer questions accurately. Traditional question-answering systems rely on predefined knowledge bases or keyword search and may fail to understand the meaning of complex questions, while large language models alone can produce fluent but unsupported answers. Retrieval-Augmented Generation (RAG) combines information retrieval with the generation abilities of large language models. This paper reviews RAG for document question answering. It explains sparse retrieval with BM25, dense retrieval with dual encoders and sentence embeddings, approximate nearest-neighbour vector search, re-ranking, and the integration of retrieved passages into generation, including fusion-in-decoder and self-reflective retrieval. Practical issues such as chunking, context length and the tendency of models to overlook information in the middle of long contexts are discussed, together with evaluation of faithfulness and relevance. A five-layer system is proposed in which users upload PDF documents that are extracted, cleaned and chunked, embedded with a sentence-transformer model and stored in a vector database; queries retrieve relevant chunks that are passed to an LLM to generate grounded answers through a Streamlit interface. Research gaps, objectives and a comparison of retrieval methods are presented.

Keywords—Retrieval-augmented generation, question answering, large language models, semantic search, vector database, text embeddings, document retrieval

I. INTRODUCTION

1. Motivation

Students, researchers and employees spend much time searching through PDFs such as textbooks, manuals, policies and research papers. Keyword search returns documents but not answers, and it fails when the question uses different words from the text. Large language models answer fluently, but without access to the specific documents they may guess or hallucinate.

Retrieval-Augmented Generation addresses both problems. Lewis *et al.* [1] introduced RAG, in which a retriever finds relevant passages and a generator produces an answer conditioned on them. Gao *et al.* [2] reviewed how RAG has evolved into a family of techniques for building reliable knowledge-grounded applications.

This paper reviews the components of RAG for document question answering and proposes a practical system for answering questions about uploaded PDFs.

2. Objectives Of The Review

- **O1:** To review retrieval, embedding and vector search methods for RAG.
- **O2:** To examine how retrieved context is used in generation.

© 2026Malwal Shreyas Ravindra. his is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited.

- **O3:** To propose a PDF question-answering system based on RAG.
- **O4:** To compare sparse, dense and hybrid retrieval.
- **O5:** To identify challenges and evaluation practices.

II. OVERVIEW OF KEY CONCEPTS

1. Question Answering Tasks

Extractive question answering selects an answer span from a passage, as in the SQuAD benchmark [3]; generative question answering composes an answer in natural language. RAG supports generative answers grounded in retrieved text.

2. Rag Components

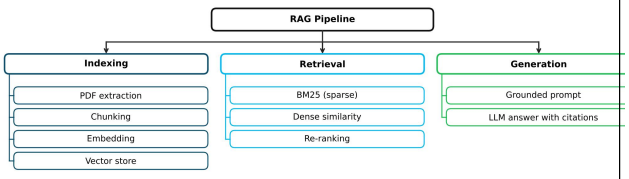


Fig. 1 summarizes the components: document processing, embedding, indexing, retrieval, re-ranking and generation.

Fig. 1. Components of a retrieval-augmented generation system.

III. REVIEW OF EXISTING WORK

Robertson and Zaragoza [4] described the probabilistic relevance framework behind BM25, the strong sparse retrieval baseline that matches query terms with document terms. Karpukhin *et al.* [5] showed that dense passage retrieval with dual BERT encoders outperforms BM25 on open-domain question answering. Reimers and Gurevych [6] proposed Sentence-BERT, producing sentence embeddings suitable for efficient semantic similarity search.

For efficient search, Johnson *et al.* [7] presented FAISS for billion-scale similarity search on GPUs, and Malkov and Yashunin [8] proposed HNSW graphs for fast approximate nearest-neighbour search, the basis of many vector databases. Nogueira and Cho [9] showed that re-ranking retrieved passages with a BERT cross-encoder significantly improves relevance.

On the generation side, Izacard and Grave [10] proposed fusion-in-decoder, which processes many retrieved passages jointly. Liu *et al.* [11] found that models use information at the beginning and end of long contexts better than information in the middle, which affects how retrieved chunks should be ordered. Asai *et al.* [12] introduced Self-RAG, where the model decides when to retrieve and critiques its own output. Es *et al.* [13] proposed RAGAS metrics for faithfulness and relevance. Table I summarizes these works.

Ref.	Authors	Year	Technique Used	Merits	Demerits
[4]	Robertson and Zaragoza	2009	BM25 relevance framework	Strong sparse retrieval	Misses synonyms
[6]	Reimers and Gurevych	2019	Sentence-BERT	Efficient semantic embeddings	Domain shift
[9]	Nogueira and Cho	2019	BERT re-ranking	Large relevance gains	Extra latency
[5]	Karpukhin et al.	2020	Dense passage retrieval	Outperforms BM25 for QA	Needs training pairs
[7]	Johnson et al.	2021	FAISS GPU search	Billion-scale similarity search	Memory for large indexes
[10]	Izacard and Grave	2021	Fusion-in-decoder	Uses many passages jointly	Training cost

Ref.	Authors	Year	Technique Used	Merits	Demerits
[11]	Liu et al.	2024	Long-context usage study	Middle context under-used	Model-dependent
[12]	Asai et al.	2024	Self-RAG	Adaptive retrieval and critique	Special training needed

TABLE I Summary of Literature Survey

IV. IDENTIFIED RESEARCH GAPS

- **G1:** Chunking strategies strongly affect quality but are chosen heuristically.
- **G2:** Retrieval misses answers when questions are vague or need multiple passages.
- **G3:** Generated answers may still go beyond retrieved content.
- **G4:** Tables, figures and scanned pages in PDFs are poorly handled.
- **G5:** Evaluation of RAG systems often relies on small manual tests.

V. SYSTEM ARCHITECTURE

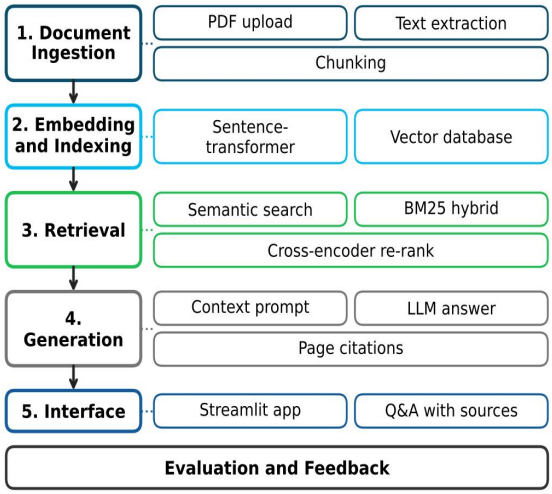


Fig. 2 shows the proposed system.

Fig. 2. Proposed RAG-based PDF question-answering system.

1. Document Ingestion Layer

Users upload PDFs; text is extracted, cleaned of headers and footers, and split into overlapping chunks of a few hundred tokens with page references.

2. Embedding And Indexing Layer

Each chunk is converted into a vector with a sentence-transformer model and stored in a vector database with its metadata.

3. Retrieval Layer

The query is embedded and the most similar chunks are retrieved by semantic similarity; optionally BM25 results are merged and a cross-encoder re-ranks them.

4. Generation Layer

Top chunks are inserted into a prompt that instructs the LLM to answer only from the provided context and cite page numbers.

5. Interface Layer

A Streamlit application lets users upload documents, ask questions and view answers with source excerpts.

VI. IMPLEMENTATION METHODOLOGY

Fig. 3 describes the workflow. Similarity between query embedding q and chunk embedding d is computed by cosine similarity

$$sim(q, d) = (q \cdot d) / (||q|| \, ||d||) \quad (1)$$

and the top-k chunks are retrieved. BM25, used for hybrid retrieval, scores term matches with saturation and length normalization. Answer quality is assessed by faithfulness to retrieved chunks, answer relevance and retrieval recall.

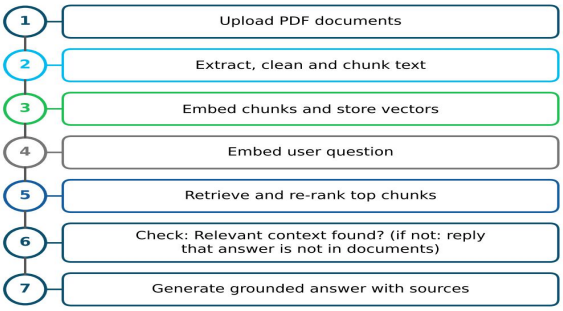


Fig. 3. Document question-answering workflow.

VII. PERFORMANCE COMPARISON

Table II compares retrieval methods. BM25 is fast and strong for exact terms, dense retrieval captures meaning, hybrid retrieval combines both, and re-ranking improves precision at extra cost.

Method	Strength	Weakness	Speed
BM25 (sparse)	Exact terms, no training	Misses paraphrases	Very fast
Dense retrieval	Semantic matching	Needs embeddings	Fast
Hybrid	Best recall	More complex	Fast
Cross-encoder re-rank	High precision	Expensive	Slow

TABLE II Comparison of Retrieval Methods

VIII. OPEN ISSUES AND FUTURE DIRECTIONS

Table III summarizes challenges. Future work includes layout-aware extraction of tables and figures, multilingual retrieval, adaptive chunking, conversational memory for follow-up questions and automated evaluation.

Challenge	Mitigation
Poor chunking	Overlap; semantic chunking
Missed answers	Hybrid retrieval; query rewriting
Unsupported answers	Strict prompts; faithfulness checks
Tables and scans	Layout-aware extraction; OCR
Evaluation	RAGAS-style automated metrics

TABLE III Challenges and Mitigation

IX. CONCLUSION

Retrieval-augmented generation turns a collection of documents into a question-answering assistant whose answers can be traced to their sources. The reviewed literature shows how dense and hybrid retrieval, efficient vector search, re-ranking and grounded generation work together. A simple pipeline of PDF extraction, chunking, sentence embeddings, a vector database and an LLM, wrapped in a Streamlit interface, can serve academic, educational and business needs while reducing irrelevant answers and hallucinations.

1. Acknowledgment

The authors express their sincere gratitude to the Department of Artificial Intelligence and Data Science, Jaihind College of Engineering, Pune, for providing the facilities, guidance and encouragement needed to carry out this work. The authors also thank the faculty members and colleagues whose valuable suggestions helped improve the quality of this review.

References

1. P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Proc. NeurIPS, vol. 33, 2020, pp. 9459–9474.
2. Y. Gao et al., "Retrieval-augmented generation for large language models: A survey," arXiv:2312.10997, 2023.
3. P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in Proc. EMNLP, 2016, pp. 2383–2392.
4. S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," Found. Trends Inf. Retr., vol. 3, no. 4, pp. 333–389, 2009.

5. V. Karpukhin et al., "Dense passage retrieval for open-domain question answering," in Proc. EMNLP, 2020, pp. 6769–6781.
6. N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in Proc. EMNLP-IJCNLP, 2019, pp. 3982–3992.
7. J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," IEEE Trans. Big Data, vol. 7, no. 3, pp. 535–547, 2021.
8. Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs," IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 4, pp. 824–836, 2020.
9. R. Nogueira and K. Cho, "Passage re-ranking with BERT," arXiv:1901.04085, 2019.
10. G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," in Proc. EACL, 2021, pp. 874–880.
11. N. F. Liu et al., "Lost in the middle: How language models use long contexts," Trans. Assoc. Comput. Linguist., vol. 12, pp. 157–173, 2024.
12. A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to retrieve, generate, and critique through self-reflection," in Proc. ICLR, 2024.
13. S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "RAGAS: Automated evaluation of retrieval augmented generation," in Proc. EACL System Demonstrations, 2024, pp. 150–158.