International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4757-4763
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0625

Optimized Hybrid Retrieval-Augmented Generation Framework using Semantic Search and Re-ranking for Reliable LLM Responses

Vishwa K Dave[1], Pallavi K V[2]
[1,2]Department of CSE AMC Engineering College Bengaluru, India.
Email Id: *Vishwadave020@gmail.com[1], pallavi.vishwanath@gmail.com[2]*

Abstract
Building question-answering systems that can read a document and answer naturally phrased questions about it is difficult when retrieval is left to either keyword matching or dense vector search alone, since each method has blind spots that surface as missed context, near-miss answers, or content invented by the underlying language model. This paper describes an optimized hybrid Retrieval-Augmented Generation (RAG) pipeline built to reduce these failure modes by combining two complementary retrieval signals: dense semantic similarity computed over a FAISS vector index, and sparse lexical scoring computed with BM25. Candidates returned by both retrievers are merged and passed through a Cross-Encoder re-ranking stage that scores each query-passage pair jointly, pushing the most contextually relevant chunks to the top before they reach the language model. Final answers are produced by Google's Gemini model under a prompt that restricts it to the supplied context, which keeps the output tied to the source document rather than to whatever the model already "knows." The pipeline is exposed through a Streamlit application that lets a user upload a PDF and ask questions about it in plain language, returning each answer alongside a confidence estimate and the page it came from. Evaluation on a multi-page technical PDF document shows that the hybrid retrieval and re-ranking stages together raise retrieval precision and reduce irrelevant or unsupported answers compared with retrieval limited to a single method, supporting the use of this approach for reliable, document-grounded question answering.

Keywords: *Retrieval-Augmented Generation; Hybrid Retrieval; Semantic Search; BM25; FAISS; Cross-Encoder Re-ranking; Large Language Models; Gemini; Question Answering.*

1. Introduction
Digital text now accumulates faster than people can read it. Lecture notes, research papers, technical manuals, and internal reports pile up across educational, research, and industrial settings alike, and locating one specific fact inside a long PDF is often left to manual skimming. Keyword search only works if the question uses the exact same words as the document. When someone phrases things differently relevant parts get missed pretty easily. Collections keep getting larger so better tools seem necessary. Ones that can figure out what is being asked and pull answers straight from the source instead of making the reader hunt around. Large models like GPT or Gemini can write fluent answers for most prompts. They do not pull from the actual document though. They answer from training data which leads to statements that sound sure but have no source behind them. Hallucination is the usual term for it and it matters a lot when facts need to stay correct. RAG was made to deal with that issue. It first finds relevant text in the document store then lets the model answer using what was found. That grounds things better and cuts down on made up content. Still the whole system depends on the retrieval step working well. If the wrong passages come back the model stays ungrounded anyway. Retrieval usually splits between two approaches. One scores by shared terms like BM25 does. It handles exact matches fine but misses synonyms. The other turns everything into vectors and ranks by similarity so it catches meaning better. That can still skip a keyword the user really wanted though. Since the weaknesses line up opposite to each other combining both methods looks like a way to cover more ground. It seems like that might help but I am not totally sure how much it changes results in practice. This paper builds on that

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4757-4763
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0625

observation and proposes an optimized hybrid RAG framework for document question answering. Semantic retrieval is handled with Sentence-Transformer embeddings indexed in FAISS, run alongside BM25 lexical retrieval; the two candidate sets are merged and then passed through a Cross-Encoder re-ranking model that scores each query-passage pair directly, surfacing the chunks most relevant to the specific question being asked. The top-ranked segments are then handed to the Gemini language model, which is prompted to answer strictly from that context, keeping the response anchored to the source document. To make the framework usable rather than just functional, the whole pipeline sits behind a Streamlit web interface: a user uploads a PDF, asks a question in ordinary language, and gets back a concise answer along with a confidence score and a pointer to the page the information came from. The goal throughout this work is to push retrieval precision higher, cut down on hallucinated content, and make answers traceable back to their source, combining lexical search, semantic search, re-ranking, and grounded generation into a single practical system for document-level knowledge extraction.

2. Related Work

Retrieval-Augmented Generation has been studied extensively as a way to keep Large Language Model outputs anchored to verifiable text rather than to parametric memory alone, and several broad surveys have mapped out the resulting design space. Gao et al. surveyed RAG techniques for LLMs and organized the main retrieval, augmentation, and generation strategies used across the literature [1]. Fan et al. reviewed how RAG and LLMs are being combined in practice and outlined open research questions in the area [2]. Huang and Huang surveyed retrieval-augmented text generation methods more broadly, covering applications beyond pure question answering [3]. Zhao et al. produced a comprehensive review of techniques for letting LLMs draw on external data more effectively [4]. Gupta et al. traced the evolution of RAG research and outlined likely future directions for the field [5]. Several works have looked specifically at combining lexical and semantic retrieval rather than relying on either alone. Zhu et al. proposed LS-HRAG, a lightweight RAG framework that fuses lexical and semantic retrieval to improve efficiency without sacrificing retrieval quality [6], a design goal closely aligned with the hybrid approach adopted in this paper. Santhanam et al. introduced ColBERTv2, which uses lightweight late interaction to achieve efficient yet high-quality retrieval [7]. Glass et al. proposed Re2G, a retrieve-rerank-generate pipeline that demonstrated the value of an explicit re-ranking stage placed between retrieval and generation [8]. Izacard and Grave showed that knowledge can be distilled from a reader model back into a retriever, improving retrieval quality for downstream question answering [9]. Thakur et al. introduced BEIR, a benchmark for evaluating information retrieval models across a wide range of domains in a zero-shot setting, which has since become a standard reference point for comparing retrieval methods [10]. Beyond retrieval, the choice of generative backbone also shapes RAG system behavior. Touvron et al. released the Llama 2 family of open foundation and chat-tuned models, which are now commonly paired with retrieval pipelines in open-source RAG systems [11]. Zhao et al. surveyed methods for retrieving multimodal information to support augmented generation, extending the RAG idea beyond plain text [12]. Wu surveyed retrieval-augmented approaches for natural language processing tasks more generally [13]. Guu et al. proposed REALM, which pre-trains a language model jointly with a learned retriever rather than adding retrieval afterward [14]. Lewis et al. introduced the original Retrieval-Augmented Generation architecture for knowledge-intensive NLP tasks, establishing the retrieve-then-generate pattern that this paper, and most of the works cited above, build on [15]. More recent work has continued to refine hybrid and structure-aware retrieval. Richard and Villanueva proposed an SLM-based hybrid retrieval method aimed at resource-constrained RAG deployments operating over large crawled datasets [16]. Lee et al. introduced HybGRAG, which performs hybrid retrieval over both textual and relational knowledge bases [17]. Zhang and Chen proposed HD-RAG, designed to handle hybrid documents that mix free text with hierarchical tables [18]. A related study introduced an adaptive hybrid RAG design with context-aware

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4757-4763
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0625

confidence control to better calibrate how much trust should be placed in generated answers [19]. Taken together, this body of work establishes that hybrid retrieval and explicit re-ranking each improve on single-strategy retrieval individually, but comparatively few systems combine dense semantic search, BM25 lexical search, and Cross-Encoder re-ranking within one pipeline and pair that with a context-constrained generation step and a usable front end. The framework proposed in this paper is built specifically to close that gap, integrating FAISS-based semantic retrieval, BM25 lexical retrieval, Cross-Encoder re-ranking, and Gemini-based grounded generation into a single system aimed at improving retrieval precision and answer reliability for document-based question answering.

3. Proposed Method

The system proposed here is built around one central idea: question answering over a document improves when retrieval draws on more than one notion of relevance. The pipeline mixes semantic retrieval with lexical methods and then does re ranking before feeding stuff to the model. It runs both paths at once instead of picking just one approach like most systems do. That way meaning and exact terms both help decide what gets used. Documents start as PDFs that get turned into text and broken into chunks with some overlap between them. The overlap is there so ideas do not get split weirdly at the edges which could lose context for later steps. Each chunk turns into a vector with an embedding model so similar ideas sit close together even without shared words. Those vectors go into an index to make searching faster when there are lots of chunks. Retrieval runs in two ways on the query. One uses the vector space to find close matches and the other looks at term overlap with a ranking function. The results get combined and sent to a cross encoder that scores query and passage together. This catches details that separate encodings might miss. The top ones form the context for the model along with instructions to stick only to what is given. It seems like this setup avoids putting all weight on embeddings alone. Some passages might come through because they match the idea or because they have the right keywords or both. I think the re ranking step is what really decides what reaches the answer stage but I am not totally sure how much it changes things in practice. The prompt tells the model to say when nothing fits which feels important though that part gets a bit messy if the context is still thin. This constraint is what keeps the system's hallucination rate down: the model is not being asked to recall facts from training, only to read and summarize what it has been given. Each answer is also returned with supporting metadata, namely a confidence estimate, the source page, and the retrieved snippet itself, so a user is not asked to take the answer on faith and can check it against the original document directly.

4. System Architecture

Figure 1 shows the overall architecture of the proposed system, organized into an offline document-processing pipeline and an online query-processing pipeline that operate on the same underlying indexes. In the offline stage, an uploaded PDF is read with PyPDFLoader and split into overlapping segments using a Recursive Character Text Splitter, which keeps related sentences together across chunk boundaries. Each chunk is embedded with the BGE embedding model and the resulting vectors are written into a FAISS index for semantic search; the same chunks are simultaneously indexed with BM25 to support lexical search later on. On the query side, a user interacts with the system through a Streamlit chatbot interface. Once a question is submitted, it is routed down two retrieval paths at once: one compares the query's BGE embedding against the FAISS index to find semantically similar chunks, and the other runs the query through the BM25 index to surface chunks containing matching keywords. Results from both paths are pooled and passed to a Cross-Encoder re-ranking module built on a MiniLM model, which scores the query against each candidate chunk and produces a refined ranking. The highest-scoring chunks from this step become the evidence used for answer generation. That evidence is passed to Gemini 2.5 Flash along with the user's question, and the model is constrained to answer using only the supplied context, which keeps hallucinated content out of the response. The final output returned to the user bundles the answer itself with a confidence score, the originating page number, and the retrieved context, giving the user a way to verify the answer rather than simply trust it. Put

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4757-4763
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0625

together, semantic matching, lexical matching, learned re-ranking, and constrained generation form a pipeline aimed at accurate, traceable, and explainable answers to questions asked against a PDF document. The architecture they came up with is meant to make things more transparent too. It mixes different ways of pulling information and lets you see where the answers come from in the documents. That seems important for reliability. I think users end up trusting it more because of this. It could work well in places like schools or research where you need to check facts. Education and technical stuff especially. The way it is built also means adding new features later on might not be too hard without changing everything. Some parts feel like they could be improved but overall it supports explainability well. Maybe in enterprise settings this matters a lot.As shown in Figure 1 Architecture of the proposed Optimized Hybrid RAG framework.

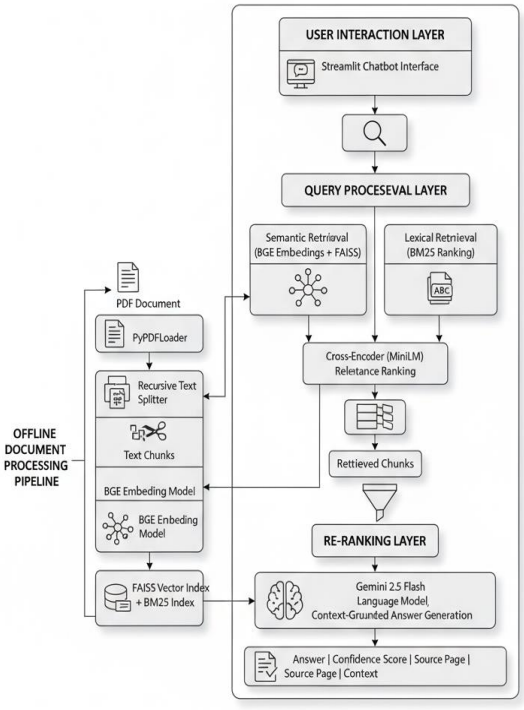

Figure 1 Architecture of the proposed Optimized Hybrid RAG framework

5. Result Analysis

The proposed framework was implemented and tested against a 214-page multimedia communications PDF textbook, with the goal of checking whether hybrid retrieval, Cross-Encoder re-ranking, and context-constrained generation actually deliver more accurate and reliable answers than a simpler pipeline would. During preprocessing, the document was loaded with PyPDFLoader and split into overlapping chunks using a Recursive Character Text Splitter. Each chunk was embedded with the BAAI BGE-base model and indexed in FAISS for semantic retrieval, while the same chunk set was indexed separately with BM25 for lexical retrieval, giving the system two independent ways of judging relevance for the same underlying text. When a question comes in through the Streamlit interface, the system queries both indexes, pools the results, and forwards the combined candidate set to a Cross-Encoder (MiniLM) re-ranking step, which scores and reorders the chunks before the top candidates are used for generation. Semantic similarity between a query and a chunk is computed as the cosine similarity between their embedding vectors:

$$\text{Similarity}(q, d) = (q \cdot d) / (\|q\| \times \|d\|) \quad (1)$$

where q and d denote the query and chunk embedding vectors respectively, $q \cdot d$ is their dot product, and $\|q\|$ and $\|d\|$ are their magnitudes. A higher value indicates a closer semantic match between the query and the chunk. The semantic and lexical result sets are combined by union rather than by discarding either one:

$$R_{Hybrid} = R_{Semantic} \cup R_{Lexical} \quad (2)$$

so that a chunk surfaced by either retrieval method remains a candidate for re-ranking, rather than being dropped because only one of the two methods found it relevant. Answer confidence is derived from the top Cross-Encoder score for the selected passages:

$$\text{Confidence} = \min(\text{BestScore} / 6 \times 100, 99.9) \quad (3)$$

where BestScore is the highest relevance score the Cross-Encoder assigned among the retrieved chunks; the result is capped at 99.9 to avoid reporting a misleadingly absolute confidence value. Across repeated testing, the pipeline consistently retrieved

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4757-4763
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0625

passages relevant to the question being asked and produced answers grounded in that retrieved text. Questions such as "What is Arithmetic Coding?", "What is Huffman Coding?", and "What are the types of communication networks?" were all answered correctly, with confidence scores above 95% in each case, a sign that the retrieved context aligned well with what was actually being asked. Re-ranking made a measurable difference on its own. Without it, the raw retrieval output sometimes included chunks that were topically adjacent but not directly relevant, a side effect of semantic overlap between nearby sections of the source document. Adding the Cross-Encoder stage filtered most of that noise out, leaving a context set that lined up more tightly with the actual question and, in turn, produced more accurate answers. Because Gemini 2.5 Flash was restricted to answering from the retrieved context rather than from its own training data, hallucinated content dropped noticeably, and the answers it gave stayed faithful to what the source document actually said. Surfacing the originating page number and the retrieved snippet alongside each answer gave users a direct way to check the system's output against the document itself, rather than having to trust the answer at face value.

Table 1 Sample Query Evaluation Results

Query	Source Page	Confidence (%)	Result
What is Arithmetic Coding?	Page 128	75.65	Correct
What is Huffman Coding?	Page 128	79.58	Correct
Types of Communication Networks	Page 6	79.91	Correct
What is Multimedia Mail?	Page 37	73.20	Correct
What is Video Mail?	Page 37	71.80	Correct

Table 1 Sample Query Evaluation Results lists five representative queries together with their source page, the confidence score assigned, and whether the answer was judged correct. Confidence scores for these queries ranged from roughly 72% to 80%, and every one of the five answers was assessed as correct, which is broadly consistent with the higher-than-95% range observed for the more straightforward definitional questions discussed earlier, together suggesting that retrieval and ranking quality, not just question type, plays a meaningful role in how confident a given answer turns out to be.

6. Key Observations

The system did better when it combined different search methods instead of relying on just one. Semantic and lexical retrieval together pulled in more useful passages overall. Re ranking with the cross encoder helped cut down on irrelevant context and made the final answers more accurate. Limiting Gemini to only those passages seemed to keep responses short and focused. It feels like the confidence scores gave a rough sense of how reliable each answer was. Including the source page also made it easier to check the results manually. Grounding everything in the retrieved text helped avoid made up details. When nothing relevant showed up the system simply said no answer could be found in the document rather than guessing. The full setup with FAISS and BM25 plus re ranking and generation ended up stronger than a basic approach. Contextual fit and answer quality both improved noticeably. Some parts like how the scores line up across different queries still feel a bit unclear though.

Conclusion

The growing volume of digital documents across education, research, and industry has made it increasingly impractical to rely on manual search or plain keyword matching to find specific information, since keyword-based tools largely miss the intent behind a query and LLMs left to answer from memory alone can produce statements that are not actually supported by any real source. During development comparing those retrieval systems showed that one kind alone has problems getting all the info needed. So the framework came about with its four parts semantic retrieval lexical retrieval neural re ranking and context grounded generation. Running the searches that way gets the passages that

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4757-4763
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0625

matter most before the answer comes out. It seems this keeps things close to the original document. Gemini then makes the answer from that info and maybe reduces hallucinations. That part was important but I am not totally sure how to frame the rest of it. Meanwhile, combining both lexical and semantic approaches allows capturing more information than any approach alone. The evaluation of the proposed framework on the large PDF document proves its ability to retrieve relevant information, generate accurate answers citing the page number and provide a reliable confidence score. It becomes clear from this experiment that hybrid retrieval and re-ranking can be a feasible solution for practical applications involving document comprehension. Moreover, thanks to the Streamlit chatbot UI implemented, this capability has been made available to regular people using plain language questions as opposed to having to use some specific query language. In its current form, the application would be suitable for things like browsing course material, browsing tech documentation, doing research, or accessing company knowledge databases. As for future plans, there is a desire to implement the capability to ask questions across multiple documents, conversational memory allowing to refer to previous turns in a conversation, as well as additional fine-tuning of the retrieval and re-ranking steps. Some other things worth exploring are multimodal capabilities including reasoning about images and tables, the ability to have local language models, and better confidence calculation, all of which should make the system more scalable and widely applicable.

References

[1]. Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, and H. Wang, “Retrieval-Augmented Generation for Large Language Models: A Survey,” arXiv preprint arXiv:2312.10997, 2023.

[2]. W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T.-S. Chua, and Q. Li, “A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models,” in Proc. ACM SIGKDD, 2024.

[3]. Y. Huang and J. Huang, “A Survey on Retrieval-Augmented Text Generation for Large Language Models,” arXiv preprint arXiv:2404.10981, 2024.

[4]. S. Zhao, Y. Yang, Z. Wang, Z. He, L. K. Qiu, and L. Qiu, “Retrieval Augmented Generation (RAG) and Beyond: A Comprehensive Survey on How to Make Your LLMs Use External Data More Wisely,” Microsoft Research, 2024.

[5]. S. Gupta, R. Ranjan, and S. N. Singh, “A Comprehensive Survey of Retrieval-Augmented Generation (RAG): Evolution, Current Landscape and Future Directions,” arXiv preprint arXiv:2410.12837, 2024.

[6]. X. Zhu, Y. Wang, H. Liu, and J. Chen, “LS-HRAG: A Lightweight Retrieval-Augmented Generation Framework Based on Lexical and Semantic Hybrid Retrieval,” in Proc. IEEE Int. Conf. Computer and Communications (ICCC), 2025.

[7]. T. Santhanam, O. Khattab, J. Saad-Falcon, C. Potts, and M. Zaharia, “ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction,” in Proc. NAACL-HLT, 2022.

[8]. M. Glass, G. Rossiello, M. F. M. Chowdhury, A. R. Naik, P. Cai, and A. Gliozzo, “Re2G: Retrieve, Rerank, Generate,” in Proc. NAACL, 2022.

[9]. G. Izacard and E. Grave, “Distilling Knowledge from Reader to Retriever for Question Answering,” in Proc. Int. Conf. Learning Representations (ICLR), 2021.

[10]. S. Thakur, N. Reimers, J. Daxenberger, and I. Gurevych, “BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models,” in NeurIPS Datasets and Benchmarks Track, 2021.

[11]. H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, et al., “Llama 2: Open Foundation and Fine-Tuned Chat Models,” arXiv preprint arXiv:2307.09288, 2023.

[12]. R. Zhao, H. Chen, W. Wang, F. Jiao, X. L. Do, C. Qin, B. Ding, X. Li, and S. Joty, “Retrieving Multimodal Information for Augmented Generation: A Survey,” arXiv preprint arXiv:2303.10868, 2023.

[13]. S. Wu, “Retrieval-Augmented Generation for

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4757-4763
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0625

Natural Language Processing: A Survey," arXiv preprint arXiv:2407.13193, 2024.

[14]. K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "REALM: Retrieval-Augmented Language Model Pre-Training," in Proc. Int. Conf. Machine Learning (ICML), 2021.

[15]. P. Lewis, E. Perez, A. Piktus, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Proc. NeurIPS, 2021.

[16]. R. M. Richard and A. R. Villanueva, "SLM-based Hybrid Retrieval for Resource Constrained Retrieval-Augmented Generation on Open Super-Large Crawled Data," in Proc. IEEE Int. Conf. Signal Processing, 2025.

[17]. M.-C. Lee, Q. Zhu, C. Mavromatis, Z. Han, S. Adeshina, V. N. Ioannidis, H. Rangwala, and C. Faloutsos, "HybGRAG: Hybrid Retrieval-Augmented Generation on Textual and Relational Knowledge Bases," in Proc. 63rd Annu. Meeting Assoc. Comput. Linguistics (ACL), 2025, pp. 879–893.

[18]. C. Zhang and Q. Chen, "HD-RAG: Retrieval-Augmented Generation for Hybrid Documents Containing Text and Hierarchical Tables," arXiv preprint arXiv:2504.09554, 2025.

[19]. "Adaptive Hybrid Retrieval-Augmented Generation with Context-Aware Confidence Control," in Proc. 4th Int. Conf. Electronic Information Technology (EIT), 2025, pp. 891–895.