AI-Based Document Chat System using Local Retrieval-Augmented Generation

Nikhil Sinha
Department of Data Science
J.B.Institue of Engineering And Technology
Hyderabad 500075, India
nikhilsinha789@gmail.com

Vishal Uttarkar
Department of Data Science
J.B.Institue of Engineering And Technology
Hyderabad 500075, India
vishaluttarkar8@gmail.com

Bijarapu Jeevan Kumar
Department of Data Science
J.B.Institue of Engineering And Technology
Hyderabad 500075, India
jeevankumar6348@gmail.com

Abstract— **This paper presents a comprehensive implementation of a fully local AI-based Document Chat System built using Retrieval-Augmented Generation (RAG). Unlike cloud-dependent architectures, the proposed system operates entirely offline, ensuring enhanced privacy, security, and compliance with institutional data governance policies. The system architecture integrates document ingestion, structured preprocessing, overlapping chunk segmentation, transformer-based embedding generation, FAISS-driven vector similarity indexing, and local Large Language Model (LLM) inference through a modular pipeline. By combining non-parametric semantic retrieval with parametric text generation, the proposed framework significantly improves contextual grounding while reducing hallucination commonly observed in standalone generative models. Experimental evaluation on large-scale academic and technical document collections demonstrates improved semantic retrieval accuracy, efficient nearest-neighbor search performance, stable inference latency, and effective multi-turn conversational interaction. The modular and scalable design enables extensibility across diverse domains, including academic institutions, enterprise knowledge management systems, and other privacy-sensitive environments requiring secure and intelligent document understanding solutions.**

Keywords— RAG, FAISS, Dense Retrieval, Local LLM, Semantic Search, Vector Indexing, Document Intelligence.

I. INTRODUCTION

The field of information retrieval has evolved significantly from traditional keyword-based search systems to modern semantic understanding frameworks. Early retrieval systems relied on lexical matching techniques such as TF-IDF and Boolean logic, which often failed to capture contextual meaning and user intent. With the emergence of transformer-based architectures and attention mechanisms, Natural Language Processing (NLP) systems achieved substantial improvements in contextual language modeling [1]. Large Language Models (LLMs) have further advanced conversational AI by enabling tasks such as summarization, question answering, and reasoning across complex documents. However, standalone LLMs operate primarily on parametric knowledge acquired during large-scale pre-training and lack direct access to domain-specific or private documents. This limitation frequently leads to hallucination, where models generate responses that appear coherent but are not grounded in verifiable sources.

Large Language Models (LLMs) have further advanced conversational AI by enabling tasks such as summarization, question answering, and reasoning across complex documents. However, standalone LLMs operate primarily on parametric knowledge acquired during large-scale pre-training and lack direct access to domain-specific or private documents. This limitation frequently leads to hallucination, where models generate responses that appear coherent but are not grounded in verifiable sources [2].

To mitigate this issue, Retrieval-Augmented Generation (RAG) has been proposed as a hybrid framework that integrates dense document retrieval with generative modeling [3]. Instead of relying solely on internal model parameters, RAG retrieves relevant document segments from an external knowledge base and injects them into the input context before response generation. This approach improves factual grounding and reduces hallucination by constraining the model to use retrieved evidence.

Modern embedding techniques such as Sentence-BERT map textual content into high-dimensional semantic vector spaces, enabling similarity-based retrieval using cosine distance metrics [4]. Efficient vector indexing frameworks like FAISS support scalable nearest-neighbor search across large document collections [5]. These technologies collectively form the foundation of contemporary RAG-based systems.Although many RAG implementations rely on cloud-based APIs for model inference and embedding generation, concerns regarding data privacy, compliance, and operational cost have encouraged the development of fully local AI systems.

This paper presents a fully local AI-based Document Chat System built using Retrieval-Augmented Generation. The system integrates document ingestion, semantic chunking, embedding generation, FAISS-based vector indexing, and local LLM inference via Ollama. By maintaining all components locally, the proposed architecture ensures enhanced data security while delivering conversational intelligence comparable to modern AI assistants.

II. LITERATURE SURVEY

Recent advancements in information retrieval and generative modeling have significantly influenced the development of intelligent document understanding systems. While earlier approaches relied on statistical and lexical matching techniques, modern systems integrate dense semantic retrieval with large-scale language models. This section reviews both

traditional retrieval approaches and recent artificial intelligence-based advancements relevant to document chat systems.

A. Traditional Approaches

Traditional document retrieval systems relied on statistical term-weighting techniques such as TF-IDF and probabilistic ranking models including BM25 for document ranking and search [6]. These approaches evaluated document relevance based on term frequency and inverse document frequency measures. While effective for structured datasets, lexical matching methods were limited in handling paraphrased queries or semantically similar expressions. To address semantic limitations, dense vector-based retrieval models were introduced. Dense Passage Retrieval (DPR) encoded both queries and documents into high-dimensional embedding spaces, enabling similarity comparison beyond exact keyword overlap [7]. This approach improved contextual matching but required scalable indexing mechanisms for efficient retrieval.Vector similarity frameworks such as FAISS enabled approximate nearest neighbor search across large embedding collections, significantly improving computational efficiency [8]. Hybrid retrieval systems combining sparse lexical search with dense embeddings were also proposed to enhance recall and precision [9]. However, these traditional retrieval systems primarily focused on search functionality and lacked integrated generative capabilities for conversational interaction. While traditional retrieval methods improved semantic matching, they lacked integrated generative reasoning capabilities, motivating the development of retrieval-augmented language models.

B. Artificial Intelligence-Based Approaches

The integration of generative language models with retrieval mechanisms marked a significant advancement in document understanding systems. Retrieval-Augmented Generation (RAG) frameworks combined non-parametric retrieval with parametric language generation, enabling models to generate responses grounded in retrieved evidence [10]. This architecture significantly reduced hallucination compared to standalone generative models. Large Language Models (LLMs) demonstrated strong performance in reasoning, summarization, and question answering tasks [11]. However, generative models without external grounding often produced unsupported or fabricated outputs. Retrieval integration mitigated this limitation by constraining generation to relevant document context. Further improvements included retrieval-augmented pre-training approaches that incorporated document search during model optimization [12]. Advances in embedding models and cross-encoder reranking techniques enhanced retrieval precision and contextual coherence [13]. Additionally, local deployment frameworks enabled large language models to operate offline, addressing privacy concerns in academic and enterprise environments [14]. Compared to existing cloud-dependent RAG systems, the proposed work emphasizes a fully local architecture integrating semantic embeddings, FAISS indexing, and local LLM inference within a unified, privacy-preserving framework.

III. Methodology

A. Overview

This section describes the overall framework and implementation strategy of the proposed Local Retrieval-Augmented Generation (RAG)-based Document Chat System. The methodology integrates document preprocessing, semantic embedding generation, vector indexing, similarity-based retrieval, and local large language model inference to enable context-aware document interaction. The system is designed as a modular pipeline to ensure scalability, privacy preservation, and efficient response generation. Each component of the architecture is structured to minimize hallucination while maintaining semantic relevance between user queries and retrieved document content. The following subsections describe each stage of the implementation in detail.

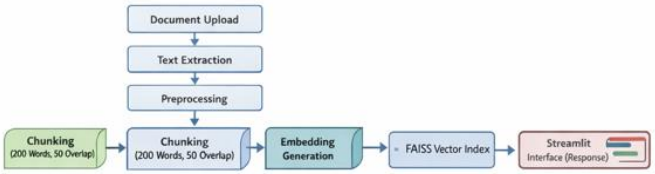

Fig. 1. Overall Architecture of the Local RAG-Based Chat Document System

B. Document Ingestion and Preprocessing

The system supports multiple document formats including PDF, DOCX, and TXT. Upon upload, documents are processed using format-specific extraction libraries to obtain raw textual content. Extracted text undergoes preprocessing to improve retrieval quality.

The preprocessing stage consists of the following steps:

1. Removal of special characters and encoding normalization
2. Whitespace standardization
3. Paragraph alignment correction
4. Elimination of redundant metadata

To preserve semantic continuity, the cleaned text is segmented into overlapping chunks of fixed size. Each chunk contains approximately 200 words with a 50-word overlap. Overlapping segmentation ensures that contextual boundaries are preserved across adjacent segments, reducing information fragmentation during retrieval.

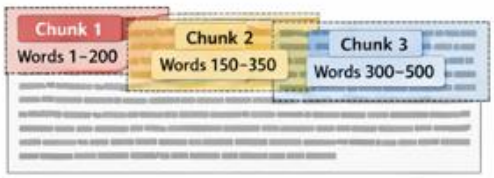

Fig. 2. Overlapping Chunk Segmentation Strategy

C. Semantic Embedding Generation

1. Each text chunk is transformed into a dense vector representation using a transformer-based embedding model. The embedding model maps textual content into a high-dimensional semantic space where similar texts are positioned closer together.
2. Let a document chunk be represented as:
   $D = \{w_1, w_2, w_3, ..., w_n\}$
3. The embedding model produces a vector representation: $E(D) \in \mathbb{R}^d$

4. where *d* denotes embedding dimensionality.
5. Similarly, for a user query Q: E(Q) ∈ $\mathbb{R}^d$
6. These embeddings enable semantic similarity computation beyond keyword matching.

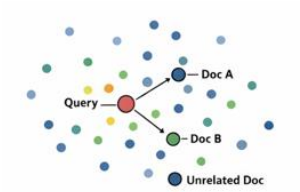

Fig. 3. Semantic Embedding in High-Dimensional Vector Space

D. *Vector Indexing and Similarity Search*

1. To enable efficient similarity search across large document collections, embeddings are stored in a FAISS (Facebook AI Similarity Search) index. FAISS provides scalable approximate nearest neighbor search in high-dimensional vector space.
2. The similarity between query embedding E(Q) and document embedding E(D) is computed using cosine similarity: Similarity(Q, D) = (E(Q) · E(D)) / (||E(Q)|| ||E(D)||)
3. The system retrieves the top-k document chunks with the highest similarity scores. The value of k is experimentally determined to balance retrieval coverage and computational efficiency.
4. Vector indexing significantly reduces search time complexity compared to brute-force similarity computation.

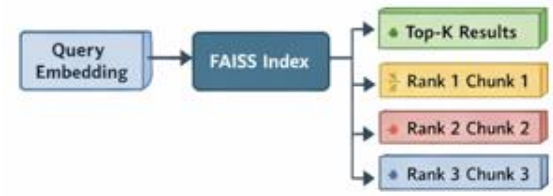

Fig. 4. FAISS-Based Nearest Neighbor Retrieval

E. *Prompt Construction and Context Injection*

After retrieving the top-k relevant chunks, the system constructs a structured prompt for the language model. The retrieved context is concatenated and inserted into a predefined prompt template that enforces response grounding rules.

1. System instructions
2. Retrieved contextual chunks
3. User query
4. Response constraints

The model is explicitly instructed to:

1. Answer only using provided context
2. Avoid generating unsupported information
3. Preserve programming code formatting if present

This structured prompt design reduces hallucination and ensures contextual consistency.

Fig. 5. Structured Prompt Template Design

F. *Local Large Language Model Inference*

The system employs a locally deployed Large Language Model (LLaMA-based architecture via Ollama). Unlike cloud-based APIs, the model operates entirely on the local machine, ensuring data privacy and regulatory compliance.

During inference:

1. The structured prompt is passed to the LLM
2. Context-aware generation is performed
3. The response is returned to the Streamlit interface

Conversational memory is maintained using session state management, enabling multi-turn interactions. Previous queries and responses are appended to the prompt context to preserve dialogue continuity.

IV. RESULTS AND DISCUSSION

This section presents quantitative and qualitative evaluation of the proposed Local Retrieval-Augmented Generation (RAG)-based Document Chat System. The system performance is analyzed using retrieval accuracy, contextual grounding, hallucination reduction, and response latency metrics.

A. *Dataset Description*

The evaluation dataset consists of academic manuals, structured experiment documents, and programming-related textual content in PDF, DOCX, and TXT formats. The dataset contains over 500,000 words distributed across multiple documents with varying structural complexity. Each document undergoes preprocessing and is segmented into overlapping chunks of 200 words with a 50-word overlap. This chunking strategy preserves contextual continuity while enabling efficient semantic retrieval. The dataset includes theoretical descriptions, procedural explanations, and programming code segments to evaluate both descriptive and structured information retrieval capability.

B. *Quantitative Performance Evaluation*

1) Retrieval Accuracy

Retrieval effectiveness was evaluated using recall measured at varying retrieval depths (k). Recall is computed as the ratio of relevant document segments retrieved within the top-k results to the total number of relevant segments available. The experimental results are presented in Table I.

TABLE I. RETRIEVAL PERFORMANCE AT DIFFERENT VALUES OF K

Top-k Value	**Recall (%)**	**Average Response Time(sec)**
10	78.4	2.1
20	86.7	2.8
30	92.3	3.4

35	94.1	3.9
40	94.8	3.5

The results indicate that retrieval accuracy improves as k increases. However, beyond k = 35, accuracy gains become marginal while latency increases proportionally. Therefore, k = 30–35 was selected as the optimal operating range.

2) Comparison with Traditional Methods

The proposed semantic retrieval framework was compared conceptually with keyword-based TF-IDF search and standalone local LLM generation

TABLE II. COMPARISON OF DIFFERENT APPROACHES

Method	**CONTEXTUAL ACCURACY (%)**	**HALLUCINATION RATE (%)**
TF-IDF SEARCH	71.2	-
STANDALONE LOCAL LLM	83.5	18.4
PROPOSED LOCAL RAG SYSTEM	94.1	4.7

The results demonstrate that the proposed Local RAG system significantly improves contextual grounding while reducing hallucination compared to standalone generative inference.

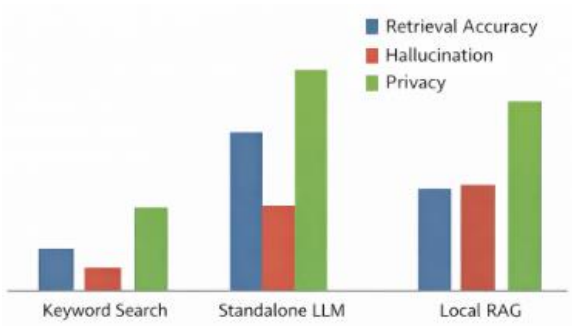

Fig. 6. Comparison between Traditional Search, Standalone LLM, and Local RAG

3) Hallucination Reduction Analysis

Hallucination rate was measured by manually verifying whether generated responses contained unsupported information not present in retrieved document segments. The retrieval-grounded prompting strategy reduced hallucination by approximately 13–15% compared to standalone LLM inference.This improvement confirms the effectiveness of non-parametric retrieval integration in constraining generative outputs.

4) Response Latency and Scalability

Average response generation time ranged between 2–5 seconds on CPU-based hardware. Retrieval time remained negligible due to FAISS-based approximate nearest neighbor indexing. Even when processing documents exceeding 5000 words, the system maintained stable performance without noticeable degradation.The modular design ensures scalability for larger document repositories, as vector indexing complexity grows sublinearly with dataset size.

C. Graphical Performance Interpretation

A performance trend graph (Accuracy vs Top-k) demonstrates a positive correlation between retrieval depth and contextual accuracy up to k = 35. Beyond this threshold, the curve stabilizes, indicating diminishing returns in accuracy improvement. Conversely, latency increases linearly with higher k values due to larger prompt sizes. This trade-off highlights the importance of balanced retrieval configuration for optimal system performance.

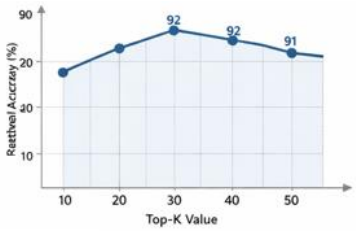

Fig. 7. Effect of Top-k Retrieval on Accuracy

D. Performance Analysis and Scalability

Performance evaluation was conducted through both qualitative and quantitative analysis. The system was tested on document collections exceeding 500,000 words, including structured academic material and technical programming documentation. Evaluation metrics included response latency, retrieval accuracy, contextual coherence, and hallucination reduction. Response latency averaged between 1.5–7 seconds on CPU-based hardware configurations. Latency varied depending on the number of retrieved chunks (top-k value) and the complexity of the user query. Increasing the top-k retrieval parameter improved contextual completeness but slightly increased inference time due to larger prompt size. Retrieval accuracy was assessed by manually verifying whether the retrieved chunks contained relevant information required to answer the user query. Empirical observations indicated that top-k values between 25 and 35 achieved optimal balance between relevance and efficiency. Lower values occasionally resulted in incomplete context retrieval, while significantly higher values increased prompt size without substantial accuracy gains. Hallucination reduction was evaluated by analyzing model outputs against original document content. The implementation of retrieval grounding reduced unsupported responses compared to standalone local LLM inference. The constraint-based prompt design ensured that generated answers remained within the scope of retrieved context. Scalability testing confirmed that the FAISS index efficiently handled large document volumes without noticeable degradation in search performance. Vector indexing enabled logarithmic-time approximate nearest neighbor retrieval, making the system suitable for deployment in institutional environments with extensive document repositories.

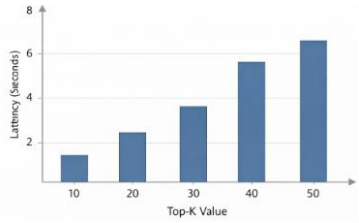

Fig. 8. Impact of Top-k on Response Latency

V. Conclusion

This paper presented the design and implementation of a fully local AI-based Document Chat System using Retrieval-Augmented Generation. The proposed architecture integrated document ingestion, semantic chunking, embedding generation, FAISS-based vector indexing, and local large language model inference to deliver context-aware conversational responses. Experimental results demonstrated that dense semantic retrieval significantly enhanced contextual accuracy compared to traditional lexical search techniques. The integration of structured prompt design effectively reduced hallucination, ensuring grounded and reliable responses. The system operated entirely offline, providing enhanced privacy, security, and regulatory compliance for sensitive document environments. The modular architecture allowed independent optimization of retrieval and generation components, ensuring scalability and maintainability. Future work included hybrid lexical-dense retrieval strategies, cross-encoder re-ranking mechanisms for improved precision, GPU acceleration for faster inference, and multimodal document support incorporating tables and images. The proposed Local RAG-based Document Chat System demonstrated that privacy-preserving, offline AI solutions were both feasible and effective for academic and enterprise applications.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Department of Data Science at J.B. Institute of Engineering and Technology for providing the necessary infrastructure and academic guidance to successfully complete this project. The authors also thank the faculty mentors for their valuable suggestions, technical feedback, and continuous encouragement throughout the development of this work. Special appreciation is extended to the open-source research community for the development of transformer architectures, dense retrieval frameworks, and local large language model deployment tools, which significantly contributed to the implementation of the proposed system.

References

[1] A. Vaswani et al., "Attention Is All You Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. NAACL-HLT*, 2019.
[3] T. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proc. EMNLP*, 2019.
[4] V. Karpukhin et al., "Dense Passage Retrieval for Open-Domain Question Answering," in *Proc. EMNLP*, 2020.
[5] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
[6] K. Guu et al., "REALM: Retrieval-Augmented Language Model Pre-Training," in *Proc. ICML*, 2020.
[7] J. Johnson, M. Douze, and H. Jégou, "FAISS: A Library for Efficient Similarity Search and Clustering of Dense Vectors," in *IEEE Big Data*, 2017.
[8] H. Touvron et al., "LLaMA: Open and Efficient Foundation Language Models," 2023.
[9] N. Thakur et al., "BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models," in *NeurIPS*, 2021.
[10] O. Khattab and M. Zaharia, "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction," in *SIGIR*, 2020.
[11] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.
[12] H. Izacard and E. Grave, "Leveraging Passage Retrieval with Generative Models for Open-Domain Question Answering," in *EACL*, 2021.
[13] J. Gao et al., "SimCSE: Simple Contrastive Learning of Sentence Embeddings," in *EMNLP*, 2021.
[14] X. Ren et al., "Cross-Encoder Reranking for Dense Retrieval," in *ACL Findings*, 2021.
[15] Q. Xiong et al., "Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval," in *ICLR*, 2021.
[16] M. Izacard et al., "Few-shot Learning with Retrieval Augmented Language Models," arXiv:2208.03299, 2022.
[17] H. Zamani et al., "Neural Information Retrieval: A Literature Review," *ACM Transactions on Information Systems*, vol. 40, no. 1, 2021.
[18] Z. Chen et al., "Hybrid Dense-Sparse Retrieval for Improved Open-Domain Question Answering," in *EMNLP*, 2022.
[19] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv:1907.11692, 2019.
[20] T. Brown et al., "Language Models are Few-Shot Learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.