IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14449

VectorChat AI

Pratham Avhad[1], Snehal Koli[2], Shruti Bhuvad[3], Avishkar Gole[4], Shilpali Bansu[5]

Student, Department of Computer Engineering, A.C. Patil College of Engineering, Navi Mumbai, India[1]

Student, Department of Computer Engineering, A.C. Patil College of Engineering, Navi Mumbai, India[2]

Student, Department of Computer Engineering, A.C. Patil College of Engineering, Navi Mumbai, India[3]

Student, Department of Computer Engineering, A.C. Patil College of Engineering, Navi Mumbai, India[4]

Professor, Department of Computer Engineering, A.C. Patil College of Engineering, Navi Mumbai, India[5]

Abstract: This Document centers on the development of an advanced chatbot system that seamlessly integrates with PDF documents, significantly enhancing users' ability to extract information using natural language queries. It addresses the growing need for efficient information retrieval from textual content, particularly in academic and professional contexts. The system provides a user-friendly platform designed to quickly and accurately extract relevant information from PDFs. To achieve this, it incorporates several modern technologies. Streamlit is used to build an intuitive and interactive user interface. For PDF parsing and text extraction, the system employs PyPDF2. LangChain is responsible for text processing and generating semantic embeddings, which improve the efficiency and relevance of indexed data. Google's Generative AI powers the chatbot, enabling it to understand complex user queries and generate accurate, context-aware responses. Additionally, FAISS is integrated to support similarity-based search, ensuring fast and precise information retrieval from the vectorized content. The system workflow begins with users uploading PDF files, which are then parsed, processed, and indexed. The chatbot interacts with users by understanding their queries and providing targeted responses based on the indexed content. The primary aim of this project is to offer a highly interactive and user-centric experience, simplifying how users engage with and extract insights from PDF documents. Future enhancements may include support for more complex queries, broader document format compatibility, and advanced features to improve user engagement. Ultimately, this project contributes to the advancement of natural language processing and intelligent information retrieval, offering value to a wide range of domains requiring effective document analysis.

Keywords: LangChain, Vector Database, Multi-PDF Chat, AI, FAISS, OCR, Streamlit, GPT.

I. INTRODUCTION

In the digital age, the exponential growth of textual information has created an urgent need for efficient and intelligent information retrieval methods. Among various digital formats, PDF documents are widely adopted for storing and sharing content, serving as essential repositories of knowledge across academic, professional, and industrial domains. However, extracting meaningful and relevant information from PDFs remains a challenge, particularly when dealing with large volumes of text or complex document structures. To address this issue, we propose an advanced chatbot system that integrates seamlessly with PDF documents, allowing users to retrieve information using natural language queries. This system is designed to provide a highly intuitive and efficient solution for extracting key insights from PDFs, substantially reducing the time and effort involved in manual searching. The motivation for this research lies in the increasing demand for user-friendly tools that can streamline document interaction, especially in environments where timely and accurate information access is critical. Existing approaches often involve tedious manual processes and are limited in their scalability and contextual understanding. Our proposed chatbot overcomes these limitations by offering an automated, intelligent, and interactive method for document querying. This paper outlines the system's methodology, detailing the technologies employed—such as natural language processing, vector search, and large language models—along with its workflow and expected outcomes. Furthermore, we highlight possible enhancements, including advanced query handling, support for various document formats, and improved user experience. By leveraging cutting-edge language models and semantic retrieval techniques, this project marks a significant advancement in document analysis technology, transforming how users interact with PDF content and making information retrieval more accessible, accurate, and efficient.

act the conference publications committee as indicated on the conference website. Information about final paper submission is available from the conference website.

II. MOTIVATION

In an era characterized by information overload, professionals, researchers, and students frequently face the challenge of navigating through extensive document collections to locate specific, relevant insights. Traditional keyword-based search

DOI: 10.17148/IJARCCE.2025.14449

systems often prove inadequate in such scenarios, as they lack the ability to understand contextual nuances, interpret user intent, or process multiple documents simultaneously. This shortfall underscores the pressing need for intelligent systems that can efficiently comprehend and extract meaningful information from diverse data sources. VectorChatAI was conceptualized in response to this demand. By utilizing advanced technologies such as vector embeddings, semantic search, and natural language understanding through platforms like LangChain and OpenAI's large language models, the system offers a conversational interface designed for intelligent interaction with multiple PDF documents. Rather than requiring users to sift through lengthy text, VectorChatAI enables them to pose natural language questions and receive precise, context-aware responses, dramatically enhancing the efficiency and accessibility of information retrieval.

III. PROBLEM STATEMENT

Accessing and extracting relevant information from multiple PDF documents remains a significant challenge in both academic and professional settings. Traditional approaches—such as manual reading, basic keyword searches, and static text extraction—often fall short when it comes to understanding nuanced context or synthesizing insights from diverse documents simultaneously. These conventional tools are limited by their inability to perform deep semantic searches, their lack of contextual comprehension, and their inefficiency and poor scalability when dealing with large document collections. Moreover, many existing solutions lack intuitive, user-friendly interfaces that support natural language interaction. This highlights the urgent need for a more advanced system capable of understanding, indexing, and querying multiple documents semantically, all while interacting with users in natural language. VectorChatAI directly addresses this gap by leveraging state-of-the-art natural language processing (NLP) techniques, vector embeddings, and large language models to deliver a powerful, scalable, and intelligent solution for document analysis and retrieval.

IV. LITERATURE REVIEW

Several recent studies have explored the integration of large language models (LLMs) and retrieval-based techniques for querying PDF documents.

Deekshita et al. [1] proposed a chatbot system using LLMs combined with Retrieval-Augmented Generation (RAG), allowing the model to retrieve relevant chunks from documents before generating accurate, context-aware responses. While effective for unstructured PDFs, the system requires significant computational resources and lacks adaptability to real-time content updates.

Prem Jacob et al. [2] implemented a LangChain-based PDF query system using OpenAI's ChatGPT, showcasing a modular pipeline for document loading, chunking, embedding, and querying. Although this design simplifies document-based question answering, it demands technical expertise and lacks evaluation on large datasets.

Khadija et al. [3] developed a domain-specific chatbot for faculty guideline documents using OpenAI's GPT and structured PDF parsing. While effective in academic FAQs, the system struggles with less structured content and lacks multilingual and adaptive capabilities.

Kotiyal et al. [4] created a LangChain-powered chatbot that processes PDFs into semantic embeddings for relevant answer generation. Despite its promising prototype, the system lacks benchmarking and real-world validation.

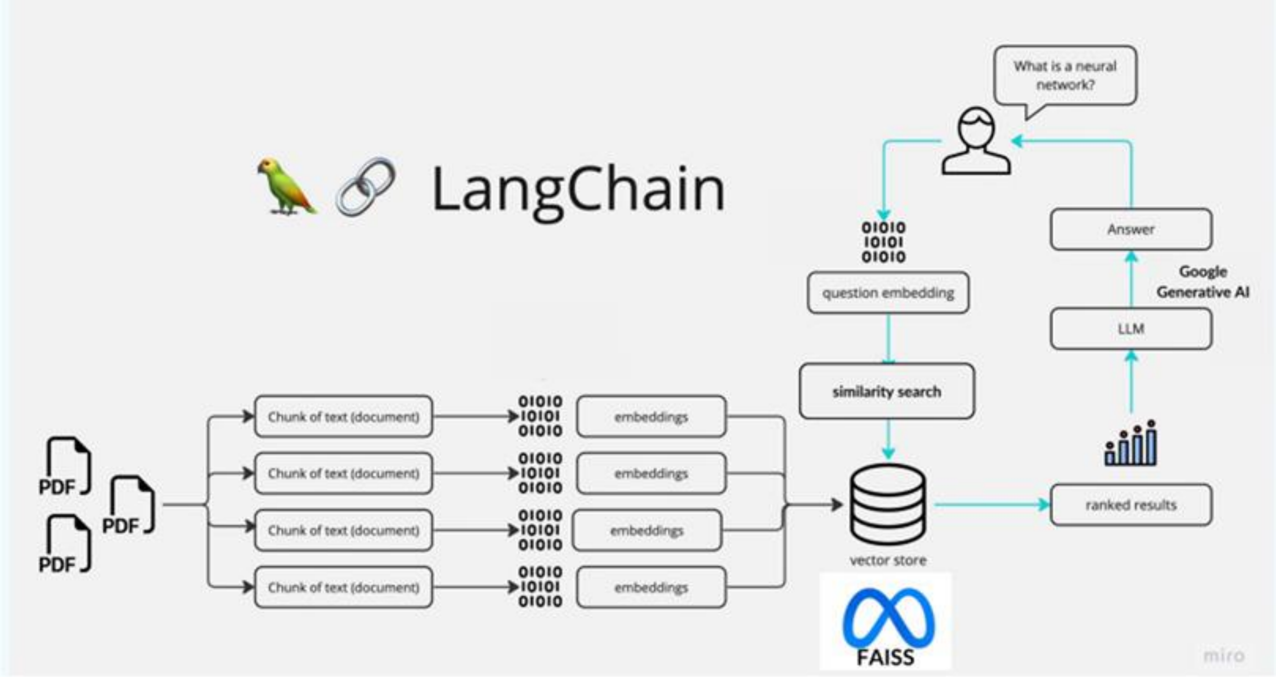
Zhang et al. [5] focused on designing a modular LangChain-based chatbot architecture, emphasizing reusable components for parsing, retrieval, and response synthesis, though it remains largely theoretical without performance metrics. Finally, Mansurova et al. [6] introduced a domain-specific chatbot tailored for blockchain-related queries, achieving high precision due to specialized training data. However, its limited scope hinders generalization and its utility is constrained by the lack of publicly available datasets. Collectively, these studies highlight the potential and ongoing challenges of building intelligent PDF-based query systems.

V. CONCEPTUAL FRAMEWORK

This workflow describes a document processing and retrieval system that leverages embeddings, vector databases, and large language models (LLMs) for providing users with relevant, synthesized responses. Here's a breakdown of how each step works:

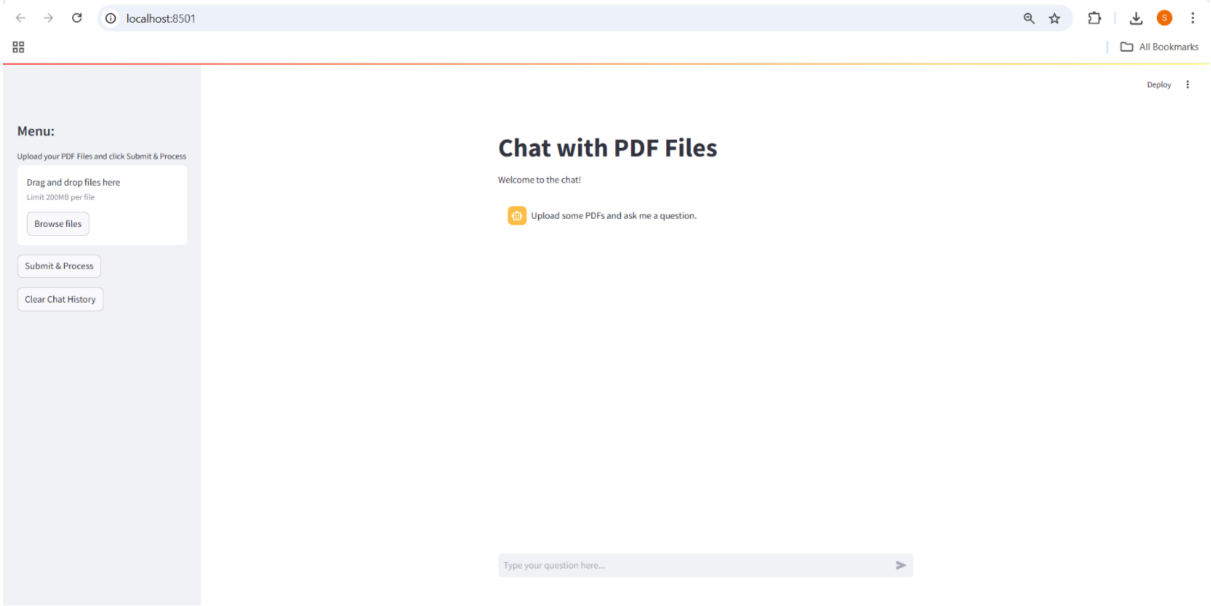
1. **User uploads one or more PDF documents**: This is the initial input phase where the user provides the documents that will be processed.
2. **Document parsing and chunking**: The system parses the PDF files to extract text. This text is then chunked into semantically meaningful segments, typically by dividing it into paragraphs, sentences, or sections based on content relevance.

DOI: 10.17148/IJARCCE.2025.14449



3. **Embedding transformation**: Each chunk of text is then transformed into an embedding, which is a mathematical representation of the text. These embeddings capture semantic meaning, allowing the system to understand the content beyond simple keyword matching.
4. **Vector database storage**: The embeddings are stored in a vector database. This database allows the system to efficiently search for and retrieve relevant chunks based on their similarity to a query.
5. **Search and retrieval of relevant chunks**: When a user submits a query, the system searches the vector database for the most relevant text chunks using similarity measures (such as cosine similarity) between the query and the stored embeddings.
6. **Response synthesis by LLM**: The retrieved chunks are passed to a large language model, which synthesizes the most relevant information to generate a comprehensive, accurate response to the user's query.
7. **Displaying the response**: The LLM's response is then presented to the user through the interface.

VI. RESULT AND DISCUSSION



localhost:8501

All Bookmarks

Deploy

Menu:

Upload your PDF Files and click Submit & Process

Drag and drop files here

Limit 200MB per file

Browse files

Submit & Process

Clear Chat History

Chat with PDF Files

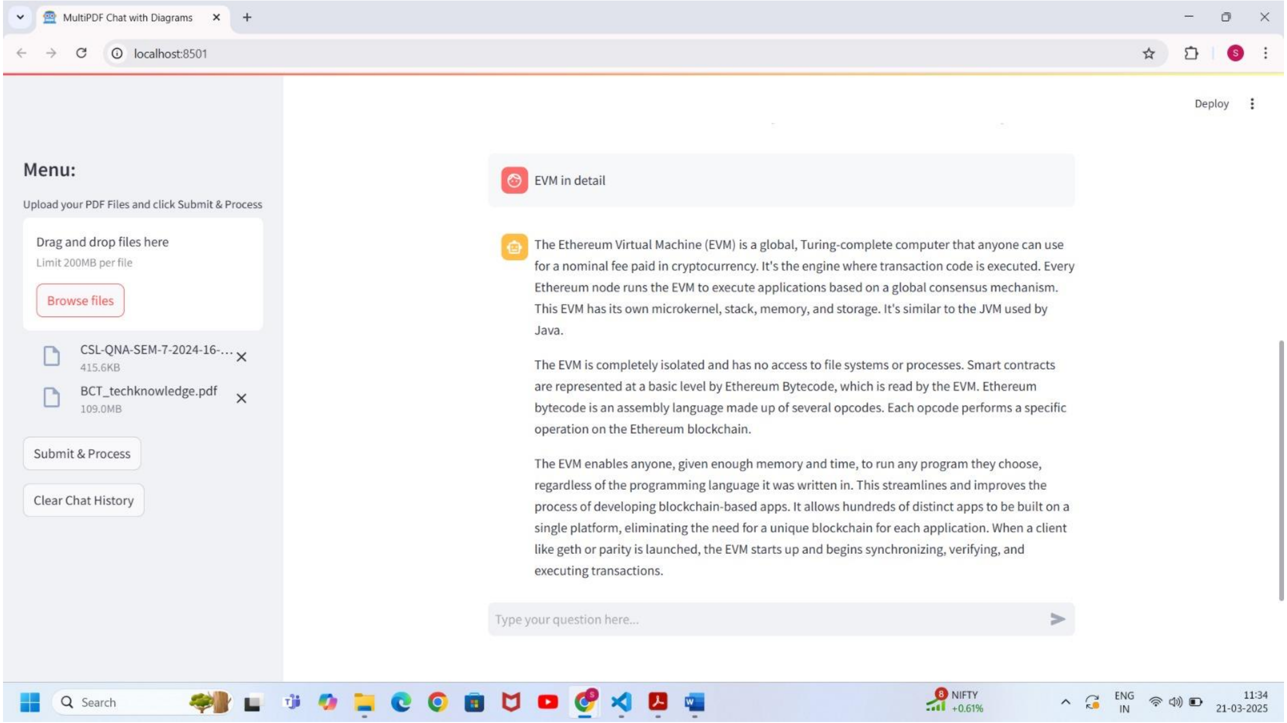
Welcome to the chat!

Upload some PDFs and ask me a question.

Type your question here...

DOI: 10.17148/IJARCCE.2025.14449

The "Chat with PDF Files" Streamlit app offers a user-friendly interface divided into two main sections. On the left sidebar, users can easily upload PDF files (up to 200MB), initiate the processing of the document, or clear the chat history. The center panel serves as the primary chat area, where users can interact with the content of the PDFs.
At the bottom of the center panel, there's an input box where users can type their questions, enabling seamless communication and exploration of the document's contents.



EVM in detail

The Ethereum Virtual Machine (EVM) is a global, Turing-complete computer that anyone can use for a nominal fee paid in cryptocurrency. It's the engine where transaction code is executed. Every Ethereum node runs the EVM to execute applications based on a global consensus mechanism. This EVM has its own microkernel, stack, memory, and storage. It's similar to the JVM used by Java.

The EVM is completely isolated and has no access to file systems or processes. Smart contracts are represented at a basic level by Ethereum Bytecode, which is read by the EVM. Ethereum bytecode is an assembly language made up of several opcodes. Each opcode performs a specific operation on the Ethereum blockchain.

The EVM enables anyone, given enough memory and time, to run any program they choose, regardless of the programming language it was written in. This streamlines and improves the process of developing blockchain-based apps. It allows hundreds of distinct apps to be built on a single platform, eliminating the need for a unique blockchain for each application. When a client like geth or parity is launched, the EVM starts up and begins synchronizing, verifying, and executing transactions.

The interface of the multi-PDF chat application is designed for ease of use, allowing users to upload multiple PDF files (up to 200MB each) either by dragging and dropping or browsing their device. Once uploaded, the files are listed, and users can remove them individually if needed. The session can be controlled through the "Submit Process" button to analyze the documents, or by using the "Clear Chat History" button to reset the session. The main chat area displays AI-generated responses based on the content of the uploaded PDFs. For example, if a user asks about the Ethereum Virtual Machine (EVM), the system provides a detailed explanation based on the documents. The interface facilitates real-time, user-friendly interaction, enabling users to seamlessly explore and query the content of their documents.

VII.CONCLUSION

The development of VectorChatAI showcases the potential of combining LangChain, vector databases, and large language models to create an intelligent, interactive system for handling multi-PDF document queries. The system allows users to upload multiple documents, including scanned or image-based PDFs. By utilizing OCR and embeddings, it extracts and vectorizes content, enabling accurate semantic retrieval and natural language responses. The integration of components such as FAISS for vector similarity search, Streamlit for the user interface, and LangChain for constructing LLM pipelines ensures a seamless and user-friendly experience. This project highlights the value of AI-driven tools in educational, research, and enterprise settings, where quick and context-aware information retrieval is essential.

ACKNOWLEDGEMNT

We express our sincere appreciation to our project mentor**, Prof. S. P. Bansu**, for her expert advice, consistent support, and encouragement throughout the development of this project. We are also thankful to our families and friends for their patience and motivation. Special thanks to everyone involved in this project whose contributions were vital in reaching our milestones.

DOI: 10.17148/IJARCCE.2025.14449

REFERENCES

[1] P. Deekshita, K. Neeharika, M. Haritha, P. Mohan, and Y. Bindu, "Pdf chat bot using generative ai (llms&rag)," *Journal of Nonlinear Analysis and Optimization*, vol. 15, no. 1, 2024.

[2] T. Prem Jacob, B. L. S. Bizotto, and M. Sathiyanarayanan, "Constructing the chatgpt for pdf files with langchain – ai," in *2024 International Conference on Inventive Computation Technologies (ICICT)*, 2024, pp. 835–839.

[3] M. A. Khadija, A. Aziz, and W. Nurharjadmo, "Automating information retrieval from faculty guidelines: Designing a pdf-driven chatbot powered by openai chatgpt," in *2023 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, 2023, pp. 394–399.

[4] A. Kotiyal, P. G. J, G. P. M. S, R. M. Devadas, V. Hiremani, and P. Tangade, "Chat with pdf using langchain model," in *2024 Second International Conference on Advances in Information Technology (ICAIT)*, vol. 1, 2024, pp. 1–4.

[5] H. Zhang, Z. Li, F. Liu, Y. He, Z. Cao, and Y. Zheng, "Design and implementation of langchain-based chatbot," in *2024 International Seminar on Artificial Intelligence, Computer Technology and Control Engineering (ACTCE)*, 2024, pp. 226–229.

[6] A. Mansurova, A. Nugumanova, and Z. Makhambetova, "Development of a question answering chatbot for blockchain domain."