



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AI-Powered PDF Chatbot: Intelligent Document Interaction.

Mulinti Rohith Naidu
Computer Science and
Engineering
Vardhaman College of
Engineering
Hyderabad, India

Shouri Deshpande
Computer Science and
Engineering
Vardhaman College of
Engineering
Hyderabad, India

Narsuri Vaishnavi
Computer Science and
Engineering
Vardhaman College of
Engineering
Hyderabad, India

Ch.Bala Subramanyam
Computer Science and
Engineering
Vardhaman College of
Engineering
Hyderabad, India

Abstract

Modern digital information growth now requires PDF documents for essential operations such as education research and legal applications. Browsing PDF files by hand results in inefficient time-spending processes for locating necessary information. A paper explores the development of an artificial intelligence chatbot platform that makes PDF document interactions more efficient through natural language command inputs. Through Streamlit development the system delivers an easy-to-use interface which permits users to add PDF files then immediately retrieve context-specific responses. With LangChain for document parsing and retrieval and OpenAI's large language model for response generation and FAISS for semantic similarity search the system operates in the backend. The integrated system components extract meaningful answers which directly come from document content. Users gain improved text consumption experiences through the chatbot which operates as a better and quicker substitute for standard reading methods together with search tools. The evaluation shows users are pleased with this solution as it performs efficiently with different documents.

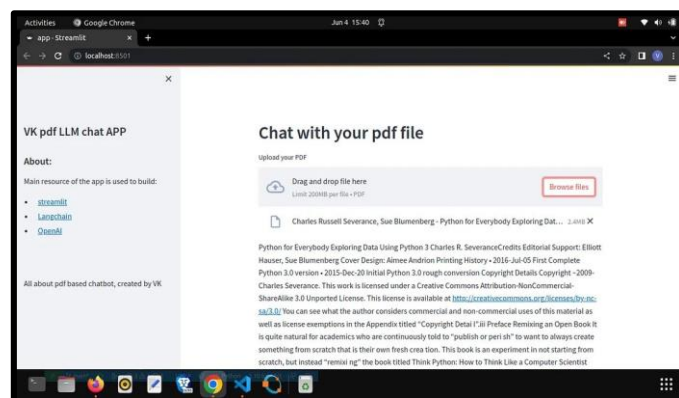
Key Words: PDF chatbot, Streamlit, Large Language Models, LangChain, Document Retrieval, Semantic Search, Natural Language Processing

I. Introduction

Individuals now use digital documents differently because they moved past traditional keyword searching toward intelligent dialogue systems. The modern information overload requires users to find quick effective methods for locating important document knowledge within lengthy PDFs without page-by-page manual browsing.

Modern users require rapid access to appropriate information since it reduces their spent effort on lengthy PDF documents. Through its interaction methods the chatbot enables users to work with PDF documents as though they were interactive texts. Our system revolutionizes document exploration because it combines large language models (LLMs) with the additional power of LangChain as well as FAISS and Streamlit tools. Through its interface users can submit PDF files which

the system converts into direct document answers in response to natural language queries. Productivity rises significantly because users no longer need to spend time reading documents or conducting manual searches particularly in academic fields and research as well as legal professionals.



1.1: PDF Chatbot (Homepage)

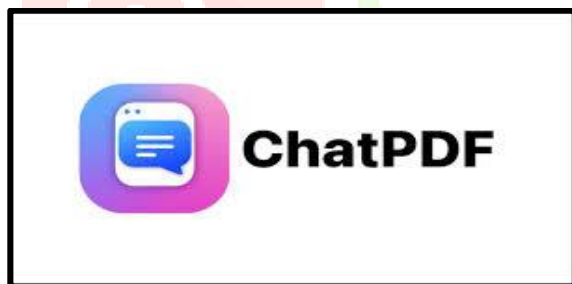
II. LITERATURE REVIEW

Document summarization and NLP techniques have previously been studied for extracting relevant datapoints from extensive text databases. The research field needs development regarding the integration of large language models with realtime PDF parsing along side semantic search functionality within chatbot applications. Our study adds advanced capabilities to fundamental innovations which leads to a full- scale AI-driven Q&A system for enhanced document interaction. A system described in “ChatPDF: Unlocking Knowledge from Documents Using AI” allows users to gain insights from PDFs through conversational question prompts. The system shows successful performance but it has limitations regarding time responsiveness and user-customizable features. The integration of Streamlit with real-time OpenAI embeddings forms a solution to overcome this interface and computational speed limitation.

A document published as “LangChain for Language Model Applications” presents a framework which uses prompt and memory systems and retrievers to manage complex NLP operations. The paper examines LangChain’s architectural structure but fails to develop it into an operational chatbot solution. Our system creates a working PDF chatbot interface by implementing LangChain’s Unstructured Loader together with RetrievalQA components.

A. **Existing Tools and Their Limitations** :This part analyzes existing PDF-related tools before examining our AI-Powered PDF Chatbot to demonstrate how our platform provides users with an advanced document experience.

■ ChatPDF



2.1 : ChatPDF

People can use ChatPDF as a web app to upload files which they can query through basic inquiries. As a basic tool ChatPDF serves its function yet fails to provide adaptable features and multiple document processing alongside active memory or running query optimization. The system uses LangChain memory components with customizable embedding pipelines along with improved control over chunking functions to eliminate these system restrictions.

■ AskYourPDF



2.2 : AskYourPDF

People can use ChatPDF as a web app to upload files which they can query through basic inquiries. As a basic tool ChatPDF serves its function yet fails to provide adaptable features and multiple document processing alongside active memory or running query optimization. The system uses LangChain memory components with customizable embedding pipelines along with improved control over chunking functions to eliminate these system restrictions.

III. METHODOLOGY

The implementation of the AI-Powered PDF Chatbot used structured modules which created document functionality with precise answers and an intuitive user interface. Building the system requires specific stages which are described in detail throughout the following sections.

1. User Interface and PDF Upload

The application welcomes users to an efficient interface that uses Streamlit as its framework. Users have an effortless method to submit PDF files before the chatbot system activates its user experience. The system displays immediate upload feedback while it verifies the successful processing for query purposes.

2. Text Extraction and Chunking

The PDF upload sequence starts with PyMuPDF extracting the text content from all its pages effectively. LangChain’s text splitter separates the extracted text into segments that improve retrieval precision. The system breaks each information segment into context windows which it handles either separately or in order.

3. Semantic Embedding Generation

The OpenAI embedding API creates dense vector embeddings from every chunk of received text. The system produces semantic embeddings that define the meaning behind document sections which aids in giving context-based responses to user inquiries. A FAISS vector store indexes the stored embeddings so that fast similarity searches become possible.

4. Intelligent Query Handling

The system processes questions from users through its submission system. The model embeds the search query then FAISS selects the most semantically suitable fragments from the database. The selected chunks accompany user queries to GPT- 3.5 for the creation of a natural language response.

5. Contextual Answer Generation

The RetrievalQA pipeline from LangChain unifies important parts of text with the question and transmits this combined information to the LLM. By doing this the system ensures that responses originated from the primary document content to fight hallucination while generating more accurate answers.

The system displays query solutions in real-time through the chat feature.

6. Streamlit-Based Real-Time Interaction

The Streamlit interface serves as the single platform for PDF uploading and question asking and receiving chatbot responses. The configuration allows for a nimble system with interactive features that operates across different platforms which include local development environments, Streamlit Cloud and Heroku deployment.

7. Session Management and Feedback Loop

During one user interaction the system keeps track of previous queries and responses through its maintained session state. A future implementation of multiple dialogue turns will be made possible by this feature. round the time users evaluate the accuracy of response predictions they create a feedback process that enables product development.

8. Admin Control and Logging

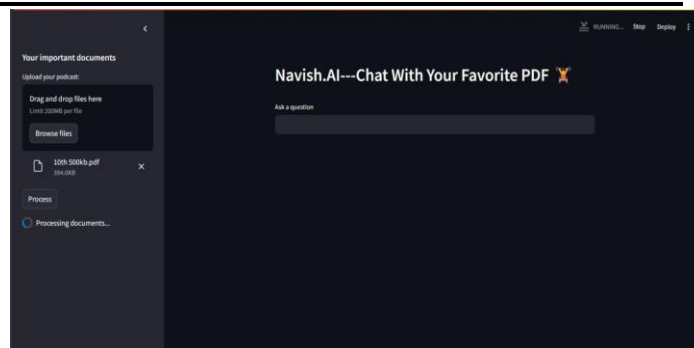
The system's administrator dashboard offers extensive features which enable effective control of operation systems. The dashboard lets the admin monitor user behavior patterns which aids in improved preference-based service delivery and engagement level assessment as well as trip pattern tracking. The admin interface section helps maintain user demands recognition while continuing uninterrupted functionality throughout the site's entire domain.

9. Contextual Memory for Query Continuity

Grouped memory functionality grants the chatbot ability to function with contextual understanding for natural intuitive dialogues. The system will retain previous session questions and answers for enhanced follow-up queries when the feature becomes operational. The system establishes basic components needed for natural human interaction in dialogue experiences.

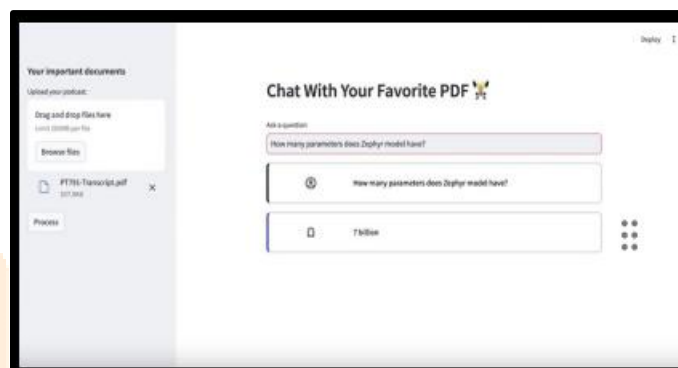
10. Error Handling and Fallback Responses

The system includes basic error detection mechanisms to handle cases where the document text cannot be processed or if the language model fails to generate a meaningful response. In such scenarios, the chatbot presents a friendly fallback message and suggests the user to rephrase or clarify the query, maintaining user trust and preventing confusion.



3.2: PDF Chatbot Interface

The **figure 3.2** displays a user-friendly dark-themed interface which enables comfortable document upload and question entry for travel PDF interaction.



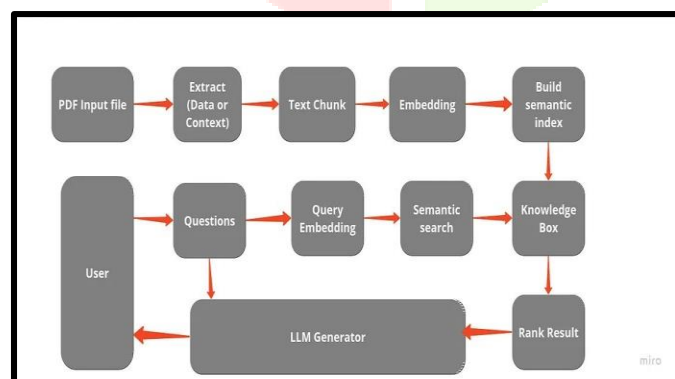
3.3: User Interaction with Chatbot

The chatbot interface shown in **Fig 3.3**, enables users to connect with AI for inquiry about travel routes

IV. RESULTS

Using AI technology the PDF Chatbot provides users with an automatic solution to search through lengthy electronic documents both quickly and intelligently. Through this application users no longer need to conduct manual PDF searches with basic tools because it provides users with both enhanced speed and personalized document experiences. Users can obtain real-time context-specific answers through this system that combines Streamlit front-end development with LangChain and FAISS along with OpenAI APIs.

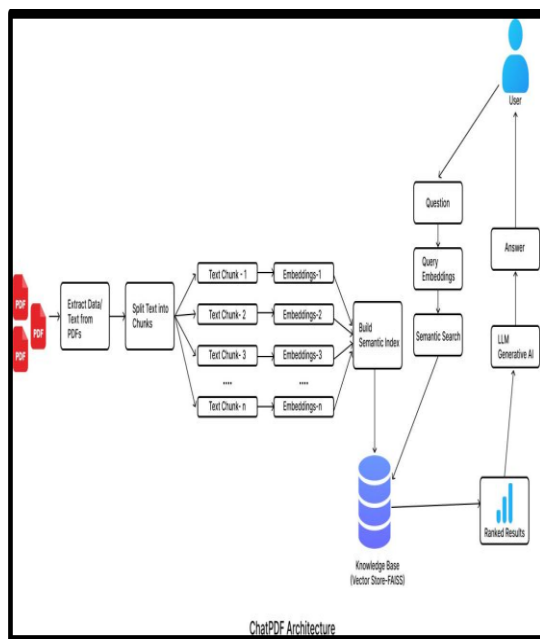
Users can access a **straightforward interface** through the chatbot to upload PDFs and ask questions that the system responds to using conversational dialogue. The system divides content into semantic parts before it generates embeddings as it retrieves sections from storage for presentation to the language model to create responses. The project distinguishes itself because its document understanding duplicates human processing capabilities making it useful to students up to professionals who work with challenging PDF documents.



3.1: Chunking and Embedding

The processing of PDF starts from extraction followed by chunking and embedding before constructing a semantic index for enhanced querying as depicted in **Fig 3.1**.

The chatbot includes feature developments that prioritize multi-document queries together with voice support as well as session-based memory which enables long-term scalability for additional functionality enhancements. Users can now avoid document scanning through complete pages because they receive valuable insights in response to a single query.



4.1: Data Flow Diagram

V. CONCLUSION:

People now have better access to static documents through PDF chatbots constructed with Streamlit and Large Language Models (LLMs). The combination of natural language processing and the Streamlit interface lets users obtain correct responses from PDF contents through a system that responds to direct questions. Through this combination users can enjoy easier access with greater engagement while the system creates new possibilities to review documents automatically and extract important insights for research applications. This

productivity result from AI fusion with user-friendly frameworks because the model demonstrates its capability to understand context and locate relevant information without requiring extensive user input.

The research demonstrates the power of contemporary NLP pipelines comprising LangChain alongside FAISS and OpenAI embeddings because they create time-sensitive intelligent document search capabilities. The system's flexible operation supports different domains including academic and legal as well as corporate organizations because it adapts to various settings. Additional features will be developed to achieve multilingual support as well as connections to cloud storage and handling multiple document types. The project demonstrates AI technology used in a practical application which advances the fundamental goal of developing information retrieval methods that are efficient and smart.

VI. REFERENCES

- [1] OpenAI, "OpenAI: Language Models and Applications," 2023. [Online]. Available: <https://www.openai.com>. [Accessed: Apr. 8, 2025].
- [2] LangChain, "LangChain: Building Applications with LLMs," 2023. [Online]. Available: <https://www.langchain.com>. [Accessed: Apr. 8, 2025].
- [3] Facebook AI Research, "FAISS: A Library for Efficient Similarity Search," 2023. [Online]. Available: <https://github.com/facebookresearch/faiss> [Accessed: Apr. 8, 2025].
- [4] Streamlit Inc., "Streamlit: The Fastest Way to Build and Share Data Apps," 2023. [Online]. Available: <https://streamlit.io>. [Accessed: Apr. 8, 2025].
- [5] Hugging Face, "Transformers and SentenceEmbeddings," 2023. [Online]. Available: <https://huggingface.co>. [Accessed: Apr. 8, 2025].
- [6] PyMuPDF, "Lightweight PDF Text Extraction in Python," 2023. [Online]. Available: <https://pymupdf.readthedocs.io>. [Accessed: Apr. 8, 2025].