Interactive PDF Chat System for Multi-Document Insights

Dr. David Raj Micheal
Department of Mathematics
School of Advanced Sciences
Vellore Institute of Technology Chennai
Tamil Nadu – 600127
davidraj.micheal@vit.ac.in

Ramya V
Department of Mathematics
School of Advanced Sciences
Vellore Institute of Technology Chennai
Tamil Nadu – 600127
ramya.v2023@vitstudent.ac.in

***Abstract*—In today's information-driven world, PDFs are widely used for storing and sharing information across various fields, but extracting useful information from large or complex PDF documents can be a challenging task. Large Language Models (LLMs) offer an innovative solution by enabling conversational AI systems to interact with and retrieve information from these files in a more efficient and intuitive manner. This study presents a solution that leverages LLMs, such as Google's Gemini model, along with LangChain and FAISS (Facebook AI Similarity Search), to process, embed, and index the text from uploaded PDF documents. Users can upload multiple PDFs, which are processed to extract text, split into manageable chunks, and indexed using FAISS. When a user asks a question, the system uses semantic search to find the most relevant text segments and generates answers based on the context using the LLM. By combining large language models with text extraction and semantic search, this system enhances document retrieval, providing accurate and contextually relevant responses to user queries. This solution improves the user experience and can be applied across various fields such as research, legal analysis, and business document management.**

***Index Terms*—Large Language Models (LLMs); PDF Document Retrieval; Conversational AI; Facebook AI similarity search (FAISS) ; Semantic Search**

I. Introduction

Interacting with PDFs through conversational interfaces has become an exciting frontier in information retrieval. Traditionally, PDF files are static and require manual effort to extract relevant details, often leading to time-consuming searches. However, by using Large Language Models (LLMs), such as GPT-based models, it is now possible to ask natural language questions directly related to the content of a PDF and receive detailed, context-aware responses. These models understand and process the text within the documents, allowing users to engage with the content in a dynamic, intuitive manner. This technology transforms PDFs from static files into interactive tools, streamlining information retrieval across various fields, from academic research to legal and business applications.

II. Objective

The primary objective is to develop an interactive system that allows users to upload multiple PDF files and query their contents through a conversational interface. The system extracts text from the PDFs, splits it into manageable chunks, and stores these chunks in a FAISS vector store using Google Generative AI embeddings for efficient text retrieval. Using a question-answering model powered by Google's Gemini, the system then responds to user queries by pulling relevant information from the indexed content, providing detailed and accurate answers. This study aims to simplify document analysis and improve information retrieval from large PDF files in an intuitive, user-friendly Streamlit app.

III. Related Works

Zürcher (2024) [1] explores the development of a chatbot for internal document management, focusing on improving the accessibility and efficiency of retrieving organizational knowledge. The system uses natural language processing (NLP) models to handle various document types and user queries, allowing employees to quickly access relevant information through a conversational interface. The paper emphasizes the chatbot's ability to streamline internal workflows, reduce search times, and enhance overall productivity by automating document retrieval, which can save time in large corporate environments where document management is often a cumbersome task.

Farinetti and Canale (2024) [2] provide a case study on the use of LangChain for developing a chatbot that fosters critical thinking and creativity. The paper delves into how chatbots built with LangChain can serve as effective educational tools, engaging students in deep, interactive learning experiences. By using natural language models to generate and respond to questions, the chatbot stimulates active learning and encourages users to think critically. The authors discuss the challenges and benefits of integrating LangChain into educational frameworks, demonstrating its ability to improve user interaction and enhance cognitive skills through AI-driven dialogues.

Chandra et al. (2024) [3] propose an innovative approach for large-scale PDF document retrieval by utilizing a neural embedding pipeline. Their method integrates distributed FAISS (Facebook AI Similarity Search) with Sentence Transformer models to create embeddings for document collections, making document search faster and more efficient. The paper explores the scalability of this approach in handling large volumes of

text, making it suitable for environments that require quick access to vast repositories of documents, such as legal firms, academic research, or enterprise-level document management systems. Their work shows how AI-based search technologies can improve the retrieval process by understanding the semantic meaning of document content rather than relying solely on keyword matching.

Singh and Gupta (2024) [4] introduce an AI-based social summarizer designed to condense and summarize content from social media platforms. The tool is designed to address the challenge of data overload by providing concise summaries of social interactions and posts. By using AI algorithms to analyze large sets of social media data, the system extracts key themes and topics, helping users navigate and understand the vast amounts of information on social media without becoming overwhelmed. This work highlights the potential of AI in the realm of social media analysis, where the sheer volume of content can hinder users' ability to extract meaningful insights.

Mujawar and Vidya (2024) [5] explore the generation of educational content and questions from books using large language models (LLMs), LangChain, and FAISS. Their approach aims to enhance the learning experience by creating tools that automatically generate questions and summaries based on the content of textbooks. By applying LLMs to book contents, their system provides educators and learners with interactive resources that stimulate critical thinking and foster engagement with reading materials. This research suggests that the combination of LLMs and AI-powered frameworks like LangChain can revolutionize the way educational content is presented and consumed, enabling more personalized and dynamic learning paths.

Topsakal and Akinci (2023) [6] provide an in-depth guide to creating large language model (LLM) applications using LangChain. Their paper serves as an instructional resource for developers interested in leveraging LangChain's capabilities to rapidly build LLM-powered applications. The authors explain how LangChain streamlines the development of conversational AI applications, reducing the complexity of integrating LLMs into real-world use cases. They discuss how LangChain's framework enables the rapid prototyping of AI-powered tools, making it an essential resource for developers looking to build scalable and efficient LLM applications across a variety of domains, from customer service to content generation.

Reddy et al. (2024) [7] introduce Docu Detective, an AI-powered PDF referencing chatbot designed to facilitate interactions with PDF documents. This system automates the process of citing and referencing information from PDFs, which is particularly valuable in academic and professional contexts. The chatbot uses natural language understanding to identify relevant sections of a document based on user queries, allowing for seamless content retrieval. This work contributes to the growing field of intelligent document interaction tools, enabling users to quickly extract and reference information from vast collections of digital documents.

Jacob et al. (2024) [8] describe the development of a ChatGPT-based application designed for querying PDF files using LangChain. Their work focuses on how large language models can be utilized to create intelligent agents capable of answering user queries related to the content of PDF documents. The system allows users to interact with documents in a conversational manner, making it easier to find specific information within lengthy and complex documents. This approach is particularly useful in academic and research settings where users need to access and process large amounts of information contained in PDFs. The integration of LangChain enhances the scalability and efficiency of the solution.

Tin et al. (2024) [9] discuss the creation of an interactive chatbot that allows users to converse with PDF documents using large language models. Their research explores how LLMs can enable users to ask questions and retrieve relevant content directly from PDF files, transforming the way people interact with documents. The chatbot enhances user experience by providing dynamic, real-time responses to queries, making it possible to navigate dense and information-rich documents with ease. This work demonstrates the potential of combining LLMs with document-based applications to improve information accessibility and simplify the process of finding specific details within PDFs.

Vishnu et al. (2023) [10] propose an interactive framework for querying large PDF files, enabling users to efficiently search and retrieve data from extensive document collections. The framework utilizes natural language queries to extract information from PDFs, making it easier for users to navigate complex documents without having to manually sift through them. This is particularly useful in industries like law, finance, and research, where professionals often need to access specific details from large collections of legal texts, financial reports, or research papers. Their system provides a solution to the challenge of handling large amounts of text data efficiently.

Das et al. (2024) [11] introduce a custom LLM-based chatbot developed using LangChain, aimed at providing intelligent document retrieval and interaction services. The system utilizes large language models to respond to user queries, summarize information, and facilitate deep engagement with document content. The chatbot's ability to understand context and generate relevant responses makes it a powerful tool for domains like legal research, academic studies, and technical documentation, where users need quick, accurate, and interactive access to information stored in large text corpora.

Iqbal (2024) [12] presents InnovatED, a doctoral dissertation that investigates the role of AI in enhancing educational systems. InnovatED focuses on the development of AI-powered tools, such as chatbots and intelligent tutoring systems, that personalize learning experiences for students. The dissertation explores various ways in which AI can be integrated into educational settings to foster better engagement, streamline communication, and improve learning outcomes. Through its comprehensive study, InnovatED provides a roadmap for educators and technologists to leverage AI in creating more effective, adaptive, and inclusive learning environments.

Richard et al. (2024) [13] introduce a client-server-based

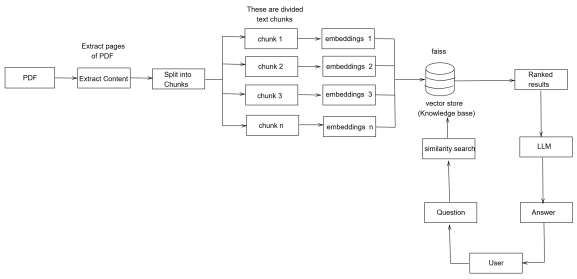
educational chatbot designed for use in academic institutions. This chatbot provides students and faculty with real-time responses to academic queries, assists with administrative tasks, and offers support in learning activities. The system leverages AI to enhance communication within educational institutions, making it easier for students to access information, track their academic progress, and engage with learning materials. The chatbot's versatility and user-friendly interface make it a valuable tool for educational environments, where timely and accurate information is crucial for student success.

Ramprasad and Sivakumar (2024) [14] explore context-aware summarization techniques for PDF documents using large language models. Their system aims to understand the context of a document and generate summaries that are tailored to the user's needs and the document's content. The research focuses on improving the relevance and coherence of automatic summaries, making them more useful in a variety of settings where quick and accurate information retrieval is important. By adapting the summarization process to the context, the authors aim to enhance the efficiency of document analysis and increase the utility of AI in document management.

Lavanya et al. (2024) [15] discuss the integration of LangChain, large language models, and vector databases such as FAISS for advanced video transcription and summarization. Their system combines these technologies to efficiently transcribe and summarize video content, making it easier to process and search multimedia data. The paper highlights the potential of AI to streamline workflows in industries that rely heavily on video content, such as media, entertainment, and education. By leveraging LangChain and FAISS, the authors demonstrate how AI can facilitate the extraction of key insights from video files and improve the accessibility of video-based information.

IV. Methodology

This section outlines the comprehensive methodology utilized for the automated PDF information extraction and retrieval system. An overview is depicted in Figure 1.



A. Data Acquisition

The initial stage in this project involves acquiring data from multiple PDF documents uploaded by the user. This step is essential as it allows the system to work with various types of PDF files that may vary in formatting, encoding, and structure. The PDF files are processed to ensure compatibility with downstream natural language processing tasks. In a typical workflow, the PDFs are gathered through a user interface that supports bulk uploads, enabling users to easily add multiple documents for simultaneous processing. This setup forms the foundation for the entire chat system, as each document must be correctly accessed and read before further processing can occur.

B. Data Preprocessing

1) Text Extraction: This stage involves extracting text data from each PDF document, which is necessary for converting the document content into a format suitable for natural language processing. The PyPDF2 library is employed for this task, offering robust functionality to parse text while handling the complexities of diverse PDF structures. PyPDF2 enables the extraction of text content across various PDF layouts, ensuring that essential information is captured from each document for further processing.

2) Text Chunking: Text chunking is the process of dividing large text bodies into manageable segments or "chunks." This step is crucial to accommodate the memory limitations of the model and to improve the efficiency of the retrieval process in subsequent steps.

a) Chunk Size Configuration:: The chunk size is configured to 10,000 characters, setting a consistent length for each text chunk. This size helps manage memory limitations while processing large documents, allowing the system to handle extensive content efficiently. The fixed chunk size is carefully chosen to capture substantial context within each segment, providing a balanced approach to ensuring that each chunk contains enough information for meaningful analysis without exceeding resource constraints.

b) Recursive Splitting:: For documents with sections that exceed the designated chunk size, a recursive splitting approach is applied. This process ensures that each text chunk remains within the configured limits while maintaining the structural integrity of the original content. Recursive splitting enables the chat system to work with documents of varying lengths and complexity without compromising the quality of information retrieval.

c) Overlap Management:: To preserve context across chunk boundaries, an overlap of 1,000 characters is applied between adjacent chunks. This overlap maintains sentence continuity and retains semantic context, which is essential when user queries may refer to content spanning multiple chunks. Overlap management ensures that critical information is not lost between sections, enhancing the chat system's ability to deliver accurate, contextually relevant responses by providing a smooth transition between adjacent segments.

C. Compact Embeddings

The GoogleGenerativeAIEmbeddings transform text chunks into compact, dense vector representations. The embedding model captures the semantic content of the text, encoding it into high-dimensional vectors that represent contextual relationships and meaning. This approach enables the system to understand similarities in meaning across different parts of the document, facilitating effective information retrieval even when phrasing varies. Each text chunk, processed through the embedding model, is converted into a vector, which preserves essential contextual details while reducing the text to a manageable numerical format. This vectorized data is then stored in a FAISS index, an efficient library for similarity search. By storing embeddings in the FAISS index, the system can rapidly search for and retrieve relevant information, ensuring responsive and contextually accurate answers to user queries across multiple PDF documents.

D. Conversational Chain Setup

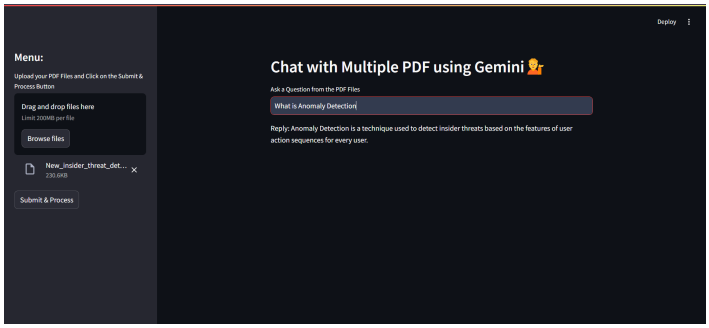
To enable the chat system to provide detailed and context-aware responses, we set up a conversational chain using the ChatGoogleGenerativeAI model. The core of this system lies in a custom prompt template designed to ensure that the model answers questions based on the provided context, offering as much detail as possible. The `prompt_template` directs the model to answer the user's question by extracting relevant information from the given context. If the information is not available, the model is instructed to respond with "answer is not available in the context," ensuring that the system does not provide misleading or incorrect answers. The conversational chain is constructed by first defining the prompt with placeholders for the context and the question. The ChatGoogleGenerativeAI model, set with a temperature of 0.3 to control the creativity of responses, is used for generating answers. The prompt and model are then passed into the `load_qa_chain` function, which builds the final question-answer chain that can be used to process and respond to user queries. This setup ensures that the system can dynamically interact with the uploaded PDF documents, pulling relevant information from the indexed embeddings and generating precise answers in real time.

E. Information Retrieval

To facilitate rapid and accurate information retrieval, the study employs FAISS (Facebook AI Similarity Search) indexing. FAISS is a highly optimized library for similarity search and nearest neighbor search, which helps in finding the most relevant chunks based on user queries. By indexing the embeddings generated in the previous step, FAISS enables the system to quickly retrieve relevant information from large document collections, making it highly suitable for interactive chat applications. This retrieval mechanism allows users to obtain contextually appropriate responses based on the content of the uploaded documents.

F. Streamlit App Interface for PDF Chat

The user interface for the chat system is built using Streamlit, providing an intuitive and interactive platform for users to interact with multiple PDF documents. The main function is the entry point, setting up the Streamlit page configuration and rendering the user interface elements such as text inputs, buttons, and file uploaders. At the top of the page, a header is displayed, welcoming users to the "Chat with PDF using Gemini" interface. Users can enter their questions through a text input field labeled "Ask a Question from the PDF Files," and upon submitting their question, the system processes the input via the `user_input` function to generate a response based on the uploaded PDF content. On the sidebar, users are provided with an option to upload multiple PDF files using the file uploader component. Once the PDFs are uploaded, users can click the "Submit & Process" button to initiate the processing of the documents. During this process, the system extracts text from the PDFs, divides it into manageable chunks, and stores it in a vector database for efficient retrieval. A spinner is displayed to indicate that the system is working on the processing, and upon completion, a success message is shown to confirm that the documents have been successfully processed. This interface allows users to seamlessly upload documents and receive detailed, contextually accurate responses based on the content of their PDFs, all through a simple and efficient web-based interface.



V. Conclusion

In conclusion, this methodology provides a structured approach to building a chat system capable of handling multiple PDF documents. Through data acquisition, preprocessing with careful chunking and overlap management, compact embedding generation, and efficient information retrieval using FAISS indexing, the system is designed to deliver accurate and contextually relevant responses. Each step is crafted to ensure that users can interact with their documents in a seamless and meaningful way, enhancing accessibility to information within large and complex PDF collections.

REFERENCES

[1] Zürcher, A. (2024). Developing a Chatbot for Internal Documents.

[2] Farinetti, L., Canale, L. (2024). Chatbot Development Using LangChain: A Case Study to Foster Critical Thinking and Creativity. In Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1 (pp. 401-407).

[3] Chandra, B., Preethika, P., Challagundla, S., Gogireddy, Y. End-to-End Neural Embedding Pipeline for Large-Scale PDF Document Retrieval Using Distributed FAISS and Sentence Transformer Models. Journal ID, 1004, 1429.

[4] Singh, S. B., Gupta, D. (2024). AI Based Social Summarizer.

[5] Mujawar, K. M., Vidya, S. Content and Question Generation from Book using LLM, Lang Chain, and FAISS.

[6] Topsakal, O., Akinci, T. C. (2023, July). Creating large language model applications utilizing langchain: A primer on developing llm apps fast. In International Conference on Applied Engineering and Natural Sciences (Vol. 1, No. 1, pp. 1050-1056).

[7] Reddy, M. L., Hemanth, G. N., Harsha, K., Pranav, S., Manoj, K. H. (2024). Docu Detective. AI-A Pdf Referencing Chatbot. International Research Journal of Innovations in Engineering and Technology, 8(4), 318.

[8] Jacob, T. P., Bizotto, B. L. S., Sathiyanarayanan, M. (2024, April). Constructing the ChatGPT for PDF Files with Langchain–AI. In 2024 International Conference on Inventive Computation Technologies (ICICT) (pp. 835-839). IEEE.

[9] Tin, T. T., Xuan, S. Y., Ee, W. M., Tiung, L. K., Aitizaz, A. (2024). Interactive ChatBot for PDF Content Conversation Using an LLM Language Model LLM-Based PDF ChatBot. International Journal of Advanced Computer Science Applications, 15(9).

[10] Vishnu, B. V., Rao, S. S., Netravathi, B. (2023, November). An Interactive Framework for Querying Data from Large Pdf Files. In 2023 International Conference on Recent Advances in Information Technology for Sustainable Development (ICRAIS) (pp. 25-30). IEEE.

[11] Das, D., Rath, R. L., Singh, T., Mishra, S., Malik, V., Sobti, R., Brahma, B. (2024, February). LLM-Based Custom Chatbot Using LangChain. In International Conference On Innovative Computing And Communication (pp. 257-267). Singapore: Springer Nature Singapore.

[12] Iqbal, M. J. (2024). InnovatED (Doctoral dissertation).

[13] Richard, R. P., Veemaraj, E., Thomas, J. M., Mathew, J., Stephen, C., Koshy, R. S. (2024, June). A Client-Server Based Educational Chatbot for Academic Institutions. In 2023 4th International Conference on Intelligent Technologies (CONIT) (pp. 1-5). IEEE.

[14] Ramprasad, A., Sivakumar, P. (2024, April). Context-Aware Summarization for PDF Documents using Large Language Models. In 2024 International Conference on Expert Clouds and Applications (ICOECA) (pp. 186-191). IEEE.

[15] Lavanya, K., Aravind, K., Dixit, V. (2024, February). Advanced Video Transcription And Summarization A Synergy of Langchain, Language Models, And VectorDB with Mozilla Deep Speech. In 2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE) (pp. 1-9). IEEE.