2024 8th International Conference on Computing, Communication, Control and Automation (ICCUBEA)
Pimpri Chinchwad College of Engineering (PCCOE), Pune, India. Aug 23-24, 2024

2024 8th International Conference on Computing, Communication, Control and Automation (ICCUBEA) | 979-8-3503-9177-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCUBEA61740.2024.10774607

Retrieval Augmented Generation Approach for Multipdf Chatbot using LangChain

Anuj Jaiswal
Computer Department
G H Raisoni College of Engineering & Management
Pune, India
anuj.jaiswal.cs@ghrcem.raisoni.net

Garima Tiwari
Computer Department
G H Raisoni College of Engineerng & Management
Pune,India
garima.tiwari.cs@ghrcem.raisoni.net

Aakash Jha
Computer Department
G H Raisoni College of Engineering & Management
Pune,India
aakash.jha.cs@ghrcem.raisoni.net

Rushikesh Mangulkar
Computer Department
G H Raisoni College of Engineering & Management
Pune,India
rushikesh.mangulkar.cs@ghrcem.raisoni.net

Pushpi Rani
Computer Department
G H Raisoni College of Engineering & Management
Pune,India
pushpi.rani@ghrcem.raisoni.net

***Abstract*— This paper presents an AI-driven application designed to facilitate information retrieval and conversational interactions through uploaded PDF documents. In an era where a vast amount of information is contained within PDFs, it is essential to have efficient extraction and interaction mechanisms. The purpose of this paper is to bridge the gap between users and the wealth of knowledge contained in these documents. The research problem is addressed by utilizing advanced AI technologies to streamline document processing and enhance user engagement. The paper emphasizes the importance of efficient information retrieval and user-friendly interfaces in addressing existing challenges in accessing and discussing content within PDF documents.Methodologically, the paper integrates various technologies, including Google's Generative AI and FAISS, to enable seamless document processing, text segmentation, and embedding generation. Key findings underscore the effectiveness of the implemented methodologies in facilitating interactive communication and extracting relevant information from PDFs. The paper's conclusions highlight the it's utility as a versatile tool for knowledge sharing and information retrieval. Limitations are acknowledged, emphasizing the need for ongoing research and improvement in addressing emerging challenges. Overall, this research contributes to advancing AI-driven applications for document processing and interactive communication, paving the way for enhanced knowledge accessibility and user engagement in diverse domains.**

Index Terms : Information retrieval ,Conversational interactions ,PDF documents ,Google's Generative AI , FAISS ,Text segmentation ,Embedding generation ,User-friendly interfaces ,Knowledge accessibility.

I. INTRODUCTION

In the contemporary era of digital documents, the extraction and utilization of information from PDF files has become a significant challenge. This article introduces an AI-driven application specifically designed to enhance information retrieval and foster conversational interactions using uploaded PDF documents. The development, functionality, and importance of this application are examined, emphasizing its potential to increase user productivity and facilitate knowledge sharing.The main objective of the application is to improve accessibility and comprehension of information within PDF documents. With the help of advanced AI technologies, users can effortlessly upload PDF files through a user-friendly interface and extract crucial textual data. The application further optimizes efficiency by dividing the extracted text into manageable segments. By generating embeddings for each text segment using Google's Generative AI technology, advanced information retrieval techniques become possible. Additionally, the incorporation of a conversational AI model enables users to query the system for insights and engage in dynamic conversations.A diverse range of programming tools and languages are utilized in the development of the application to ensure robust functionality and user accessibility. A virtual environment is created to efficiently manage dependencies, while the Streamlit framework facilitates the creation of a user-friendly interface. Version control and collaborative development are made possible through GitHub, with Python serving as the primary programming language due to its flexibility and extensive library support. Visual Studio Code is employed as the integrated development environment (IDE). The integration of Google's Generative AI API enhances the application's conversational capabilities, enriching user interactions and information retrieval processes.The prevalence of digital documents, particularly in PDF format, underscores the necessity for efficient information extraction methods. Traditional approaches to parsing and interpreting textual content from PDF files often involve manual intervention or rudimentary automation, leading to inefficiencies and inaccuracies. This application aims to address these challenges by leveraging cutting-edge AI technologies to automate the extraction, segmentation, and comprehension of textual information within PDF documents. This application aims to address the challenges inherent in conventional PDF information retrieval methods by leveraging cutting-edge artificial intelligence (AI) technologies to automate the extraction, segmentation, and comprehension of textual

information within PDF documents. The primary research problem addressed by the application is the inefficiencies associated with manual parsing and segmentation processes, which are labor-intensive and error-prone, hindering productivity and knowledge dissemination. Existing automation solutions often lack sophistication and contextual understanding, making it difficult to extract relevant information from PDF documents. Therefore, the application seeks to bridge this gap by developing an AI-driven system capable of intelligently parsing, segmenting, and comprehending PDF content, thereby enhancing user accessibility and productivity.The overarching objective of the research is to develop an interactive AI-driven application that facilitates seamless information retrieval and conversational interactions. The specific objectives include implementing robust PDF processing algorithms, integrating Generative AI technology for advanced retrieval techniques, developing a conversational AI model, and designing a user-friendly interface.The significance of the paper lies in its potential to revolutionize information retrieval and knowledge sharing across various domains. By automating and optimizing the extraction and comprehension of PDF content, the application empowers users to access, understand, and discuss complex information with unprecedented ease and efficiency. The integration of advanced AI technologies augments productivity and fosters collaborative learning and decision-making processes.The paper contextualizes existing knowledge by conducting a literature review encompassing relevant studies in AI-driven information retrieval, natural language processing, and document parsing, ultimately proposing an integrated solution that combines advanced PDF processing with conversational AI capabilities.

II. Related works

A. Literature review

The fusion of retrieval and generation capabilities in AI systems, termed Retrieval Augmented Generation (RAG), has garnered significant attention. Projects like OpenAI's GPT-3 exemplify this integration, while prior research in information retrieval and NLP lays theoretical foundations. However, existing literature reveals gaps in contextual understanding, multi-modal integration, scalability, and user interaction. This paper addresses these limitations through an interactive AI application leveraging advanced technologies.[1]

B. Papers Related to RAG

One notable paper that aligns closely with the concept of Retrieval Augmented Generation is OpenAI's GPT-3 model. GPT-3 (Generative Pre-trained Transformer 3) is a state-of-the-art language model renowned for its ability to generate human-like text responses based on given prompts. However, what distinguishes GPT-3 is its capacity to incorporate retrieved information into the generation process. This retrieval-augmented generation capability enables the model to produce more contextually relevant and informative responses.[1] The study described in this paper also falls within the realm of RAG systems. By leveraging Google's Generative AI technology alongside advanced information retrieval techniques, the application facilitates conversational interactions grounded in the content extracted from uploaded PDF documents. This integration of retrieval and generation functionalities enhances the depth and relevance of responses provided by the conversational AI model.

C. Prior Research Theories and Methodologies.

Prior studies in the field of RAG systems have mostly concentrated on improving AI models' capacity to exploit retrieved data to provide replies that are logical and contextually appropriate. Using large-scale language models that have been pre-trained on enormous amounts of textual data—like GPT-3—and honing them on particular tasks or domains is one common way. [2] Through this process of fine-tuning, the model learns to produce responses that are appropriate for the context given by the information that has been obtained.Furthermore, information retrieval research has produced fundamental theories and techniques that support the creation of RAG systems. Relevant information is retrieved and presented to users using traditional information retrieval techniques such document indexing, query processing, and relevance rating. Developments in natural language processing (NLP) have made it easier to create increasingly complex retrieval and generation models that can comprehend and produce content that is similar to that of humans.

D. Gaps and Limitations in Existing Literature

Despite the progress made in the field of RAG systems, several gaps and limitations persist in the existing literature:

- Contextual Understanding: Many existing RAG systems lack the ability to comprehensively understand and utilize the context provided by retrieved information. While they may generate coherent responses, the relevance and depth of these responses are often limited by the model's inability to grasp the broader context of the conversation.
- Integration of Multi-Modal Information: Most RAG systems primarily rely on textual information for retrieval and generation tasks. However, real-world scenarios often involve multi-modal data sources, including images, audio, and video. The integration of such multi-modal information remains an underexplored area in RAG research.
- Scalability and Efficiency: As the volume of available data continues to grow exponentially, scalability and efficiency become crucial concerns for RAG systems. Many existing approaches may struggle to handle large datasets efficiently, leading to performance degradation or increased computational costs.
- User Interaction and Feedback: Effective user interaction and feedback mechanisms are essential for improving the performance and usability of RAG systems. However, limited research has been conducted on incorporating user feedback into the retrieval and generation processes to iteratively refine the system's responses.

E. Addressing Gaps and Limitations

The study outlined in this paper aims to address several of these gaps and limitations in existing literature. By leveraging advanced AI technologies, including Google's Generative AI and information retrieval techniques, the

application facilitates seamless interaction and information retrieval from PDF documents. The integration of conversational AI capabilities enables the system to understand user queries in context and generate relevant responses grounded in the content of the uploaded documents.

- Furthermore, the paper's emphasis on providing a user-friendly interface and incorporating features for managing chat history reflects a commitment to enhancing user experience and promoting iterative refinement through user feedback. Additionally, the use of Streamlit for building the web-based interface ensures scalability and efficiency in handling interactive data applications.
- The research represents a significant step towards bridging the gap between retrieval and generation in AI-driven applications. By addressing key limitations in existing literature and leveraging state-of-the-art technologies, the paper aims to contribute to the advancement of RAG systems and their practical applications in information retrieval and interactive communication. [1]

III. METHODOLOGY

The proposed system facilitates seamless interaction with PDF documents by employing advanced text processing and AI technologies. Users upload PDF files through a sidebar interface, initiating the extraction of textual content. Text chunking optimizes processing efficiency, followed by embedding into a semantic vector space via Google Generative AI. FAISS indexing enables rapid semantic similarity search for user queries, fostering effective information retrieval.

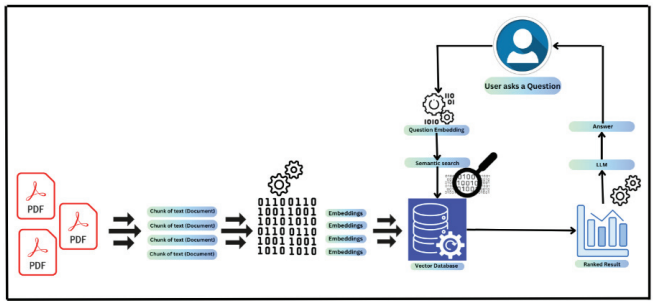

Fig.1.Block diagram of the proposed system

The system leverages a conversational AI model, RAG, to generate accurate responses from relevant text chunks. A user-friendly chat interface provides real-time interaction, maintaining a comprehensive conversation history for user reference and review. Overall, the system streamlines PDF document exploration through intuitive user interaction and advanced AI capabilities.

A. Framework

The project comprises two main components. Firstly, it focuses on text processing, which involves extracting, chunking, and embedding PDF text utilizing PyPDF2, langchain, and Google Generative AI.[9] This segment aims to structure and encode textual data for subsequent analysis. Secondly, the project incorporates query processing techniques. Here, FAISS is employed for similarity search, while a conversational chain handles question answering tasks, leveraging the embedded text for efficient retrieval and response generation.[5] These components collectively facilitate the project's goal of enabling interaction with PDF files through a chatbot interface.

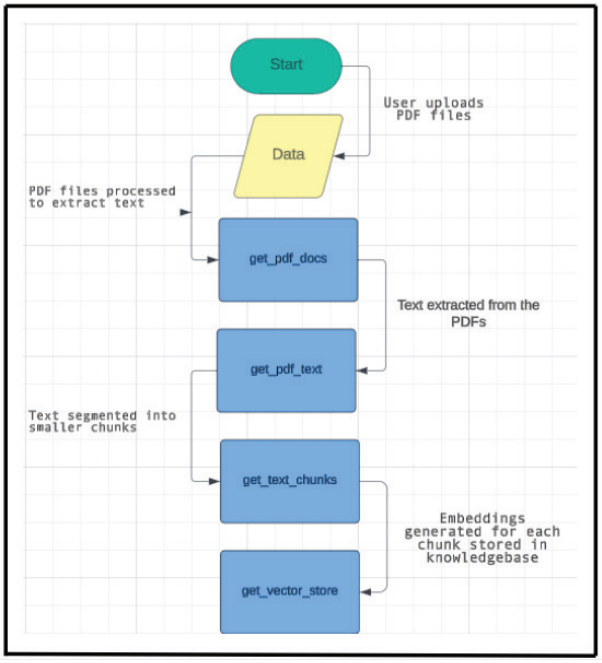

Fig.2. Text extraction & embedding flowchart

1. get_pdf_text(pdf_docs):
   - This function reads the uploaded PDF files using the PyPDF2 library and extracts textual content from each page.
   - It iterates over each PDF document, reads its pages, and concatenates the text into a single string.
   - The extracted text is returned for further processing.
2. get_text_chunks(text):
   - Once the text is extracted from the PDF documents, it can be quite large and unwieldy for processing.
   - This function splits the text into smaller, manageable chunks using the RecursiveCharacterTextSplitter.
   - The chunking process ensures efficient processing, particularly for large PDF documents, by breaking down the text into smaller segments.
3. get_vector_store(chunks):
   - Text chunks obtained from the previous step are embedded into a high-dimensional vector space using the Google Generative AI embeddings.
   - These embeddings capture the semantic meaning of the text, providing a robust representation of its underlying semantics.

- The embeddings are indexed using FAISS (Facebook AI Similarity Search) to enable fast and efficient semantic similarity search.[5]
- The indexed embeddings are saved locally.

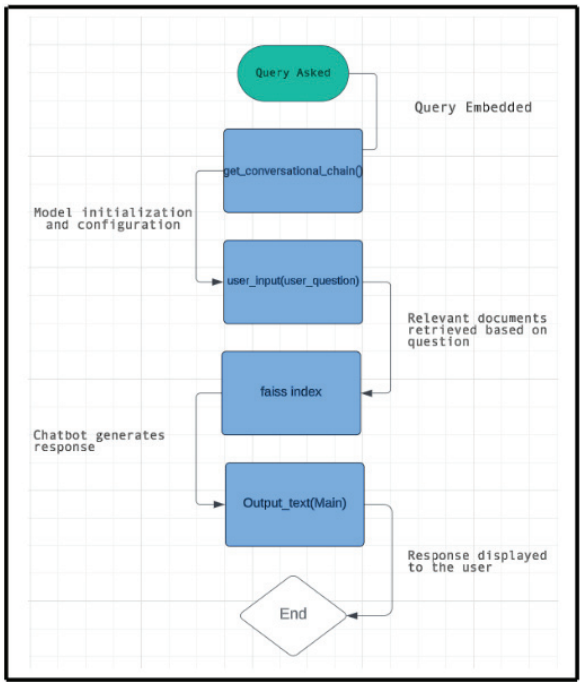

Fig.3. Query processing flowchart

4. get_conversational_chain():
   - This function initializes a conversational AI model, specifically the ChatGoogleGenerativeAI model, for generating responses to user queries.
   - It configures the model with appropriate parameters such as the model type (gemini-pro), client (genai), and temperature for response generation.
   - The model is equipped with a prompt template that guides the generation of responses based on the provided context and user questions.
5. user_input(user_question):
   - Upon receiving a user query, this function retrieves the embeddings of relevant text chunks from the FAISS index based on semantic similarity to the user question.[5]
   - It initiates the conversational AI model to generate responses by synthesizing information from the retrieved text chunks and the user query.
   - The generated responses are returned for display in the chat interface.
6. Output_text main():
   - The main function orchestrates the entire application flow, configuring the Streamlit user interface, handling user interactions, and displaying chat messages and bot responses.
   - It sets up the sidebar for PDF file upload, initializes the chat interface, and provides functionality for clearing the chat history.
   - User interactions, including uploading PDF files, asking questions, and receiving responses, are managed within the main function.

B. Faiss indexing

Vector embeddings are numerical representations of words, sentences, or documents in a high-dimensional vector space. Each dimension of the vector captures specific semantic or syntactic features of the text, allowing for the quantification of similarities and differences between different pieces of text. These embeddings are generated using advanced techniques such as word embeddings, sentence embeddings, or document embeddings, which leverage deep learning models trained on large corpora of text data.[3]

To illustrate the concept of vector embeddings and their application in our research, consider the table below, which presents a simplified representation of embeddings indexed using FAISS (Facebook AI Similarity Search):

TABLE I. SIMPLIFIED REPRESENTATION OF EMBEDDINGS INDEXED USING FAISS

Embedding ID	Vector (Reduced Dimension)	Metadata (Optional)
1	[0.1, 0.2, ..., 0.9]	Document ID: 123
2	[0.3, 0.1, ..., 0.8]	Document ID: 456
3	[0.2, 0.5, ..., 0.7]	Document ID: 789

- **Embedding ID**: Each embedding is assigned a unique identifier, allowing for easy reference in search operations or other parts of our application.[3]
- **Vector (Reduced Dimension)**: The embedding vector represents the textual information in a reduced dimension to conserve space and computational resources. These vectors contain the actual embedding values, typically represented as floating-point numbers. Each value in the vector captures specific semantic or syntactic aspects of the corresponding text.[3]
- **Metadata**: Additional metadata, such as document IDs or labels, provides context or supplementary information about each embedding. This metadata enhances the interpretability and usefulness of the embeddings in downstream tasks.

C. Gemini 1.0 pro Usecases

Gemini 1.0 Pro represents a cutting-edge development in the realm of natural language processing, boasting an array of features tailored to meet the demands of complex text and code generation tasks. This research endeavors to provide a comprehensive overview of Gemini 1.0 Pro, elucidating its underlying mechanisms and exploring the extensive repertoire of use cases it encompasses. By scrutinizing its capabilities in various domains, this paper aims to underscore the pivotal role of Gemini 1.0 Pro in advancing the frontiers of computational linguistics and AI.

1. Summarization: Gemini 1.0 Pro excels in the art of summarization, adeptly distilling key insights from lengthy documents to produce concise and informative summaries.

2. Question Answering: With its prowess in text comprehension, Gemini 1.0 Pro facilitates automated question answering, enabling the creation of comprehensive FAQs and providing prompt responses to user inquiries.

3. Digital Content Understanding: Leveraging advanced language processing techniques, Gemini 1.0 Pro facilitates the categorization of text content, enabling the assignment of relevant labels based on grammatical correctness and semantic context.

4. Classification: Gemini 1.0 Pro enables the generation of structured responses in various formats such as HTML and JSON, offering unparalleled flexibility in information organization and dissemination.

5. Info Seeking: Harnessing a wealth of world knowledge, Gemini 1.0 Pro seamlessly integrates information extracted from multimedia sources, offering comprehensive insights for information retrieval tasks.

6. Sentiment Analysis: Gemini 1.0 Pro offers robust sentiment analysis capabilities, enabling the identification and labeling of text sentiments, ranging from polarities such as positive or negative to nuanced emotions like anger or happiness.

7. Entity Extraction: By leveraging contextual cues, Gemini 1.0 Pro facilitates the extraction of entities from text, empowering users to generate customized texts tailored to specific requirements and tones.

8. Code Generation: Gemini 1.0 Pro revolutionizes code generation tasks, effortlessly translating textual descriptions into executable code snippets. From function generation to algorithm implementation, Gemini 1.0 Pro streamlines the coding process with unparalleled efficiency.

IV. Results & Discussions

A major advancement in the fields of information retrieval and natural language processing (NLP) has been made with the creation of the Multi PDF Chatbot. This section explains the main conclusions and their consequences for the paper goals and hypotheses by carefully examining the paper's architecture and functionality and exploring any potential ramifications or restrictions.

A. User Interface

The interface is designed to facilitate interaction with the PDF chatbot, specifically tailored for research purposes

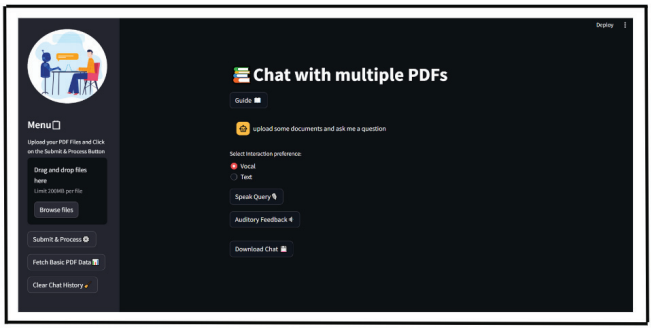

Fig.4. RAG's Graphical user interface

The layout is divided into two main sections: the sidebar and the main content area.

B. Sidebar

This section hosts the menu for uploading PDF files. Users can upload one or multiple PDF documents containing research materials by using the file uploader component. Once files are uploaded, users trigger the processing of these documents by clicking the "Submit & Process" button. This initiates the extraction of text from the uploaded PDFs, which serves as the context for subsequent chat interactions.

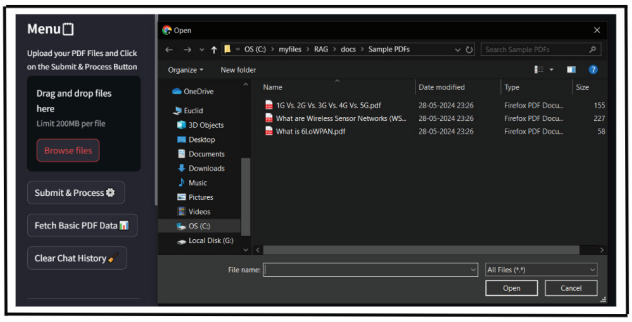

Fig.5. Browsing & processing PDFs

C. Main Content Area

The central part of the interface is dedicated to the chat interaction with the PDF chatbot. It starts with a welcome message, inviting users to engage in conversation. A button labeled "Clear Chat History" allows users to reset the chat history, providing a clean slate for new interactions.

- Chat Input: Users can type their questions or queries in the chat input box provided. As users type, their messages appear in the conversation thread, marked with the "user" role.
- Chat Messages: The conversation history is displayed as a series of chat bubbles, alternating between the user's messages and the assistant's responses. The interface maintains the continuity of the conversation, ensuring that users can easily track the flow of communication.
- Assistant Response: Upon receiving a user question, the assistant (PDF chatbot) processes the input using the provided context from the uploaded PDFs. It then generates a response based on the context and the nature of the query. The response is displayed in the chat interface as a message from the assistant, marked with the "assistant" role.

- •Thinking Indicator: During the assistant's response generation process, a spinner indicates that the AI is processing the input and formulating a reply. This visual cue informs users that the system is actively working on their request.

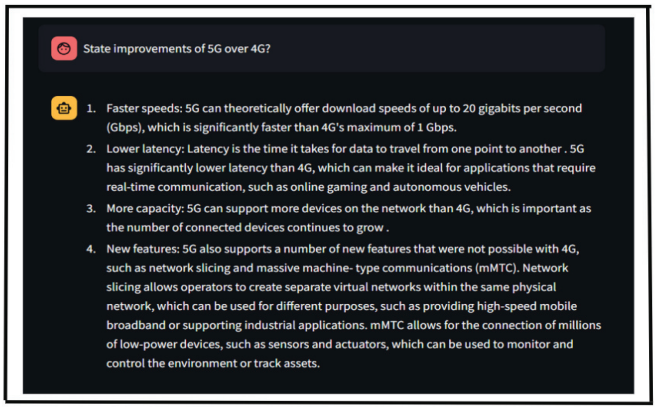

Fig 6. Chatbot composing response

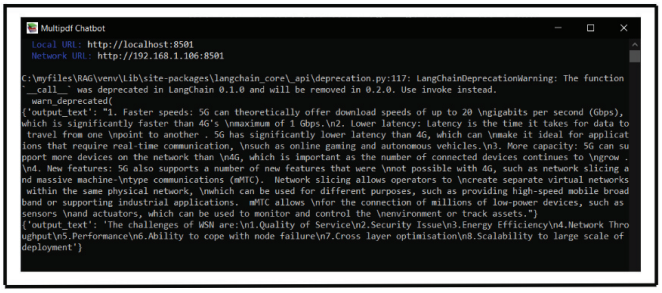

Fig 7.O/P Command Prompt

D. Addressing Limitations and Potential Biases

In examining the Gemini PDF Chatbot paper, certain limitations and potential biases emerge that warrant consideration in the Results & Discussion section of an IEEE research paper. Firstly, the effectiveness of the chatbot heavily relies on the quality and comprehensiveness of the PDF documents uploaded by users. If the documents lack relevant information or contain errors, the chatbot's performance may suffer. Additionally, the accuracy of the chatbot's responses is contingent upon the underlying language model and its training data. Any biases present in the training data could inadvertently influence the chatbot's answers, potentially leading to skewed or inaccurate responses. Moreover, the chatbot's performance may vary depending on the complexity and specificity of user queries, with more nuanced questions posing challenges for accurate interpretation. Furthermore, the reliance on third-party services, such as Google's Generative AI, introduces dependencies and potential vulnerabilities related to service uptime, data privacy, and future changes to the API. Addressing these limitations through robust data preprocessing, model training, and continuous monitoring can enhance the reliability and usability of the Gemini PDF Chatbot.

V. CONCLUSION

In conclusion, in this paper it has successfully developed a Gemini PDF Chatbot utilizing advanced natural language processing techniques. Through integration of PyPDF2 for PDF parsing, Google's Generative AI for text embeddings, and Streamlit for the user interface, we have created an interactive tool capable of extracting text from PDFs, segmenting it into chunks, and providing context-based conversational responses. By employing FAISS for efficient text similarity search and a conversational chain powered by Google's generative AI, our chatbot offers a seamless user experience. This research contributes to the field by bridging the gap between document processing and conversational AI, paving the way for enhanced document querying and interaction.

REFERENCES

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, and N. Goyal, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," Facebook AI Research, University College London, New York University, April 12, 2021.

[2] C. Dilmegani, "Large Language Models: Complete Guide in 2024," Jan 10 AI, January 10, 2024.

[3] D. S. Asudani, N. K. Nagwani, and P. Singh, "Impact of Word Embedding Models on Text Analytics in Deep Learning Environment: A Review," February 22, 2023.

[4] S. Maameri, "Building a Multi-Document Reader and Chatbot with Langchain & ChatGPT," Published in Better Programming, May 20, 2023.

[5] D. Danopoulos, D. Soudris, and C. Kachris, "Approximate Similarity Search with FAISS Framework Using FPGAs on the Cloud," August 2019.

[6] A. Sreeram and J. Sai, "An Effective Query System Using LLMS & Langchain," July 2023.

[7] T. Medeiros, M. Medeiros, M. Azevedo, M. Silva, I. Silva, and D. G. Costa, "Analysis of Language-Model-Powered Chatbots for Query Resolution in PDF-Based Automotive Manuals," October 16, 2023.

[8] Hongjin Su, Weijia Shi, "One Embedder, Any Task: Instruction-Finetuned Text Embeddings," May 30, 2023.

[9] Muhammad Usman Hadi, Qasem Al-Tashi, "Large Language Models: A Comprehensive Survey of its Applications, Challenges," July 2023.

[10] Cheonsu Jeong, "Generative AI service implementation using LLM application architecture: based on RAG model and LangChain framework," December 31, 2023.

[11] C. Jeong, "Fine-tuning and Utilization Methods of Domain-specific LLMs," January 2024.

[12] Koh Matsuda, Ian Frank, "LangChain Unleashed: Advancing Education Beyond ChatGPT's Limits," March 2024.