

PDF-Chat App Using Lang-Chain, OpenAI, and Streamlit

**Muhammed Riyas .M Yeswath N Arokiya Nikilesh .K**

Department of Computer Department of Computer Department of Computer

Application Application Application

Rathinam College Of Arts And Rathinam College Of Arts And Rathinam College Of Arts And

Science Science Science

**Abstract:**

The goal The goal of this project is to create an application called PDF-Chat by combining Streamlit ,OpenAI, and LangChain. This will allow for smooth user-system communication and the extraction and interpretation of textual data from PDF files. The software will allow users to interact with the system through text inputs, produce responses based on the context of the discussion, and deliver insightful information or support by combining OpenAI’s conversational AI technology with LangChain’s natural language processing capabilities. Furthermore, the application will provide a user-friendly experience by enabling users to upload PDF documents, their information, have chats, and export the chat history—all of which will ultimately improve accessibility and usability—by leveraging Streamlit’s user-friendly user interface architecture A clever tool called PDF Conversation Assistant is intended to improve user interactions with PDF documents. Users can ask inquiries about the contents of PDF files and the assistant will quickly extract relevant information from the documents and provide meaningful answers by utilizing sophisticated natural language processing algorithms. This novel approach greatly simplifies the knowledge extraction process from PDFs,nabling users to explore and understand complicated textual content with ease.When looking formore comprehensive information or more particular details, users may depend on the PDF Conversation Assistant to provide customized answers, which maximizes efficiency and helps with well-informed decision-making.This project provides a hardware device which will continuously monitor the vital parameters to be monitored on a patient.

any critical situation arises in a patient, this unit also raises an alarm. This is very useful for future analysis and review of patient’s health condition. For more versatile medical applications, this project can be improvised, by incorporating dental sensors and annunciation systems, thereby making it useful in hospitals as a very efficient and dedicated patient care system. In recent years, the world is facing a common problem that the number of elderly people is increasing. Hence, the problem of home-care for elderly people is very important.In this,IoT is becoming a major platform for many services and applications, also using Node MCU not just as a sensor node but also a controller here.

**Keywords** – OPENAI, Chatbot-Doc, LLM

1. **INTRODUCTION:**

In the era of surveillance, there are much upcoming The volume of information that is readily available in PDF format in the modern digital age has made it necessary to develop effective and user-friendly methods for accessing and interacting with this content. Through the creation of a PDF-Chat App that makes use of cutting-edge technologies like Streamlit, LangChain, and OpenAI, this project seeks to close the gap that exists between users and complex PDF documents. With the help of natural language interactions, users can easily explore and question PDF documents with the help of the PDF-Chat App, which is meant to offer an easy and conversational interface. This allows users to acquire relevant and contextual information.

The PDF-Chat App’s primary feature centers on the potent fusion of OpenAI and LangChain.The application’s foundation is an open-source framework called LangChain, which makes it possible to create applications with massive language models.

* 1. **Motivation**

Strong encryption procedures protect user information and guarantee privacy. natural navigation and comprehension. We have greatly improved accessibility to complicated textual content by creating powerful algorithms and intuitive design concepts, which maximizes productivity and user experience. We’ve shortened document analysis procedures with our creative approach, giving consumers the ability to effectively extract insights and information from a variety of sources. By giving users immediate access to insightful information, our intelligent question-answering solutions transform document interaction by providing precise and contextually relevant responses. Combining state-of-the-art artificial intelligence (AI) technologies with userfriendly design concepts to advance the field of document interaction. utilizing real-time answer generation technologies and user-friendly interfaces to improve accessibility to complex textual material. Enabling smooth document navigation and interaction to maximize efficiency and user satisfaction. utilizing AI-powered algorithms to streamline document analysis procedures and enable effective information and insight extraction. supplying precise and contextually appropriate answers to consumers using intelligent question-answering systems. providing strong encryption techniques and access restrictions to ensure data security and privacy compliance. provide a platform for iterative upgrades and user feedback tools to foster innovation and continual improvement. assisting in the development of knowledge management techniques and document-centric workflows across a range of industries.

* 1. **Contribution**

Users’ expectations regarding software usability and user experience have evolved significantly in recent years. Modern ERP users expect intuitive interfaces, personalized experiences, and seamless workflows that mirror their experiences with consumer-grade applications. Failure to meet these expectations can lead to user frustration, reduced productivity, and increased resistance to ERP adoption. Combining state-of-the-art artificial intelligence (AI) technologies with userfriendly design concepts to advance the field of document interaction.utilizing real-time answer generation technologies and user-friendly interfaces to improve accessibility to complex textual material.

1. **PROJECT OVERVIEW:**

*A. Scope of the project*

The scope of the project, which improves the development of intelligence video surveillance systems. Key aspects are listed below:

1. **Introduction to Setting Up a Streamlit App:**

A Python package called Streamlit makes it easier to create interactive web apps by providing a simple method for designing

eye-catching user interfaces

1. **User Intelligent Question-Answering System:**

The Intelligent Question

Answering System makes use of cutting-edge AI algorithms to comprehend customer inquiries

and provide precise answers.

1. **Security and Privacy Measures:**

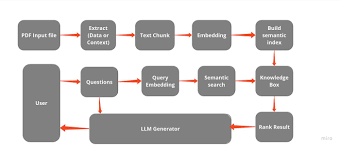
Strong encryption procedures protect user information and guarantee privacy.

**3. NATURAL LANGUAGE PROCESSING:**

The book ”Natural Language Processing and Document Understanding” by Manning, C. D., Sch¨utze, H., offers a basic grasp of how textual comprehension and analysis made possible by NLP approaches contribute to document interaction.Murphy, K. P.’s book ”Machine Learning: A Probabilistic Perspective” provides information on how to use ML algorithms to tasks involving document interaction, like information extraction, clustering, and classification.The significance of developing intuitive and user-friendly interfaces to improve the document interaction experience is emphasized in ”User Interface Design and Usability Engineering” by Shneiderman, B., Plaisant, C.Manning, C. D., Raghavan, P., Sch¨utze, H.”Information Retrieval: Implementing and Evaluating Search Engines” examines methods for enhancing information retrieval systems, promoting effective document interaction. The book Artificial Intelligence: A Modern Approach, written by Russell, S. J., and Norvig, P., covers the foundational ideas of AI that are pertinent to document interaction, such as learning, reasoning, and knowledge representation. In order to create document interaction interfaces that effectively cater to user needs and preferences, delves into the intersection of natural language processing (NLP) techniques and the comprehension of textual documents. In today’s information-driven society, the ability to extract meaningful insights from vast amounts of textual data is paramount. NLP plays a crucial role in enabling computers to understand and interpret human language, facilitating tasks such as document summarization, sentiment analysis, and information extraction. By understanding the fundamentals of NLP and its applications in document understanding, researchers and practitioners can unlock new avenues for extracting knowledge and enhancing document interaction. This introduction sets the stage for exploring the various NLP techniques and methodologies employed in the comprehension of textual documents, paving the way for advancements in information retrieval, knowledge management, and beyond.

1. **OPENAI LLM,STREAMLIT,STREMLIT INPUT:**

Define the purpose and goals of your PDF-Chat App. Determine what specific functionalities you want to implement, such as text extraction from PDFs, question-answering using OpenAI, and user interaction via Streamlit.



*Figure 1 chat-bot doc architecture*

1. **EXISTING SYSTEM:**

In the above figure 6.1 Install Python and necessary libraries such as Streamlit, LangChain, and OpenAI’s API. Set up a development environment and create a new Python project directory.Write a Streamlit script to create the user interface for the web app.Define components such as file upload buttons, text input fields, and output displays. Use Streamlit’s layout options to design an intuitive and user-friendly interface.Test. the PDF-Based Chatbot web app thoroughly, including uploading various PDF files and asking different types of questions.Debug any errors or issues that arise during testing and ensure the app functions as expected in different scenarios.To effectively use the models, it is essential to consider the memory and disk requirements. Since the models are currently loaded entirely into memory, you will need sufficient disk space to store them and enough RAM to load them during execution. When it comes to the 65B model, even after quantization, it is recommended to have at least 40 gigabytes of RAM available. It’s worth noting that the memory and disk requirements are currently equivalent.In the ever-evolving field of programming a fascinating paradigm has emerged: Prompting. Prompting involves providing specific input to a language model to elicit a desired response. This innovative approach allows us to shape the output of the model based on the input we provide.It’s remarkable how the nuances in the way we phrase a prompt can significantly impact the nature and substance of the model’s response. The outcome may vary fundamentally based on the wording, highlighting the importance of careful consideration when formulating prompts.To begin, let’s create a document object. In this example, we’ll utilize the text loader. However, Lang chain offers support for multiple documents, so depending on your specific documentsource.Once the document is loaded, we can proceed with the transformation process by breaking it into smaller chunks. To achieve this, we’ll utilize the TextSplitter. By default, the splitter separates the document .However, if you set the separator to null and define a specific chunk size, each chunk will be of that specified length. Consequently, the resulting list length will be equal to the length of the document divided by the chunk size.

1. **PESUDO CODE:**
2. **Import Libraries:**

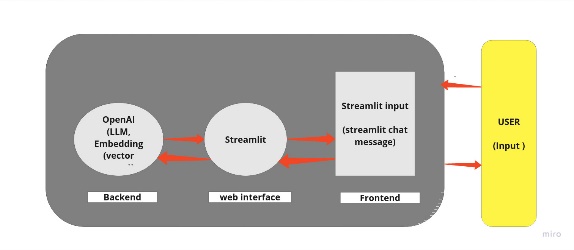
Necessary libraries such as Openai-python, and Vs code are imported.

1. **Function Definitions:**

* Stremalit connected to LLM for the localhost chatbot for Running.

**8. PROPOSED SYSTEM:**

The primary digital output is an interactive web application hosted on a server. This application allows users to upload PDF files, ask questions related to the content of the PDFs, and receive answers generated by the integrated OpenAI language model. The application interface is created using Streamlit, providing a seamless and userfriendly experience.As part of the application’s functionality, the text and metadata from uploaded PDF files are extracted using libraries such as PyPDF2 or pdfminer. This extracted text can be stored as digital output for further analysis or processing. When users ask questions or provide prompts within the application, the integrated OpenAI language model generates responses based on the content of the uploaded PDF files. These generated responses serve as digital output and are displayed to users within the application interface.

During the development and operation of the application, error logs and debugging information may be generated to track issues, monitor performance, and troubleshoot problems. These logs serve as digital output for developers and administrators to diagnose and resolve issues effectively.

*Figure 2 Workflow*

1. **RELATED WORKS:**

TOPIC: Users’ expectations regarding software usability and user experience have evolved significantly in recent years. Modern ERP users expect intuitive interfaces, personalized experiences, and seamless workflows that mirror their experiences with consumer-grade applications.

Failure to meet these expectations can lead to user frustration, reduced productivity, and increased resistance to ERP adoption.

AUTHOR: Basit Ali Department of Artificial Intelligence, University of Switzerland .

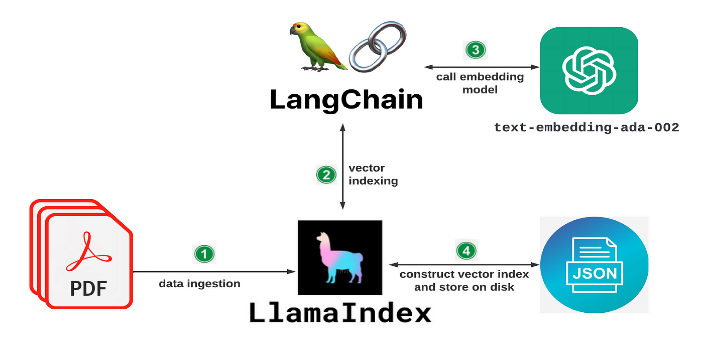
Year : 2022

pace of technological innovation continues to accelerate, introducing new possibilities and challenges for ERP systems. Emerging technologies such as artificial intelligence, machine learning,

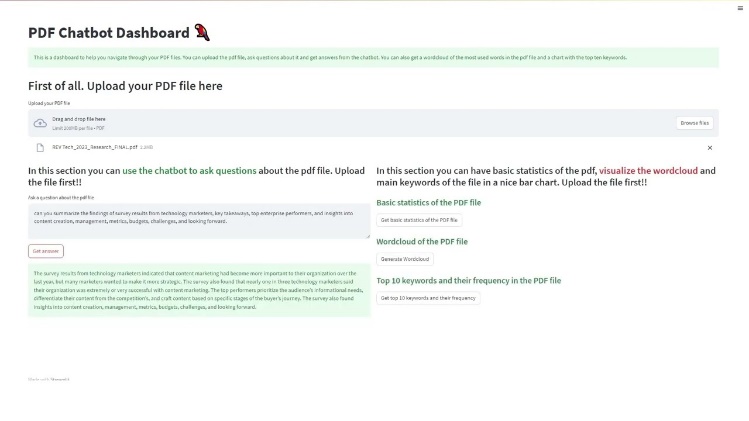
Year: 2019

1. **OUTPUT:**

The output includes the alerts are generated with the four main categories. Firstly, when the object is identified in every frame of the video then the alert is triggered. Secondly, when the falls are detected from each frame of the video which is highlighted with the red box along with the fall detected message then the alert will be triggered. Furtherly, if both vehicles are closer in the frame then its detected with the redbox again then the alert will be triggered and finally the systme detects the people who are closer to the other people in public places it's automatically marked with the red box and line between them. This process repeats again and again in each and every frame from the given video feed. Then the email alert is triggered to the required person along with the image and with the alert message. All the prediction tasks are done through the flask application where the user can upload the video only to the given four categories and the output is rendered in the flask web application.



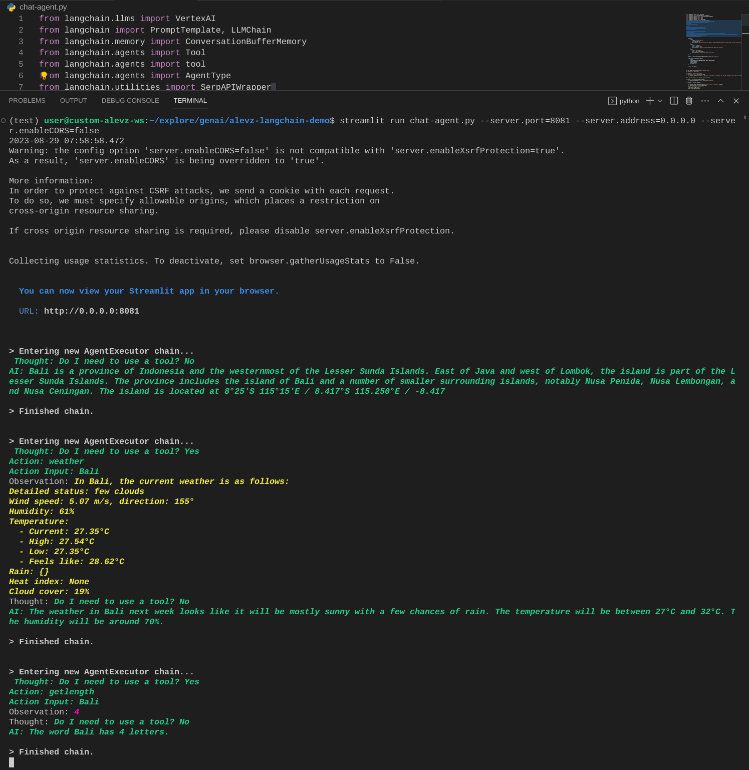
*Figure 3 LLM-PROCESS*



*Figure 4 Chat-bot Doc Interface*

* 1. **LLM-CHATBOT OUTPUT:**

Use LangChain or similar libraries to extract text and metadata from uploaded PDF files. Process the extracted text to prepare it for input to the chatbot. Utilize OpenAI’s API to integrate a language model capable of understanding and responding to user queries. Fine-tune the model if needed for better performance. Write logic to process user inputs, including text prompts and uploaded PDF files. Use the extracted PDF content as context for generating responses to user queries. Handle different question types and edge cases. Allow users to upload PDF files containing the content they want to ask questions about. Provide text input fields for direct questions to the chatbot.



*Figure 5 Output LLM CHAT-DOC*

* 1. **LLM-Technologies:**

Advanced AI Technologies Implementation of cutting-edge AI algorithms, including natural language processing (NLP) and machine learning (ML), to enable intelligent document analysis, extraction of insights, and real-time response generation. User-Friendly Interface Design: Development of intuitive and accessible interfaces that prioritize ease of use and enhance the overall user experience. This includes features such as seamless navigation, interactive elements, and visually appealing layouts. Document Analysis and Extraction. Utilization of AI-powered layout.

**CONCLUSION:**

In Conclusion, Begin by setting up your development environment and installing necessary libraries such as Streamlit, Lang-Chain, and potentially other dependencies. Write a Streamlit script to create the user interface for the web app. This script defines components like file upload buttons, text input fields, and output displays using Streamlit’s, layout options. Utilize Lang-Chain or similar libraries to extract text and metadata from uploaded PDF files. Process the extracted text to prepare it for input to the chatbot. Integrate OpenAI’s API to interact with a language model capable of understanding and responding to user queries. Fine-tune the model if necessary for better performance. -Develop logic to process user inputs, including text prompts and uploaded PDF files. Use the extracted PDF content as context for generating responses to user queries. Handle different question types and edge cases effectively.nable users to upload PDF files containing the content they want to ask questions about. Provide text input fields for direct questions to the chatbot. Display chatbot responses clearly in the web app interface.

**FUTURE WORK:**

These chatbots can comprehend user inquiries and produce responses that resemble those of a human, offering help and support in a variety of fields. They can be tailored to meet particular corporate goals or client interactions, and they use big datasets to understand the subtleties of real language.

1. **LLM-CHATBOT:**

By incorporating Language Learning Models, these chatbots are able to continuously enhance their capabilities and adjust to human interactions over time, yielding more precise and pertinent responses. With this do-it-yourself method,companies may build customized and effective support systems without requiring a lot of technical know-how or resources.

**REFERENCES:**

1.”Godbole, M. V. (2023). Revolutionizing Enterprise Resource Planning (ERP) Systems through Artificial Intelligence. International Numeric Journal of Machine Learning and Robots, 7(7),1-15.

2. Khabbaz, R. The Role of Artificial Intelligence in Enhancing Business Process Management Systems and its Implications.

3. Muniandi, B., Huang, C. J., Kuo, C. C., Yang, T. F., Chen, K. H., Lin, Y. H., ... Tsai, T. Y.(2019). A 97

4. Dong, C., Saxena, A., Bick, M., Sabia, A. (2023). On the Journey to AI Maturity.Understanding the Role of Enterprise Artificial Intelligence Service. AIS Transactions on Enterprise Systems.

5. Shibghatullah, A. S. B., Ling, C. T. C., Sarsam, S. M., Cherukuri, A. K., Qazi, S. A., Hong,K. C. (2023). Deploying Support Vector Machines and Rule-Based Algorithms for Enhanced User Training in Cloud ERP: A Natural Language Processing Approach.

6. Muniandi et al., ”A 97

7. Candelon, F., Reeves, M. (Eds.). (2022). The Rise of AI-Powered Companies. Walter de Gruyter GmbH Co KG.

8. Ashri, R. (2019). The AI-powered workplace: how artificial intelligence, data, and messaging platforms are defining the future of work. Apress.

9. Joshi, H. (2024). Artificial Intelligence in Project Management: A Study of The Role of AiPowered Chatbots in Project Stakeholder Engagement.

10. Indian Journal of Software Engineering and Project Management (IJSEPM), 4(1), 20-25. Althabatah, A., Yaqot, M., Menezes, B., Kerbache, L. (2023). Transformative Procurement Trends: Integrating Industry 4.0 Technologies for Enhanced Procurement Processes. Logistics, 7(3), 63.