

Chat with multiple pdfs by using Langchain.

Chakravarthi Jada, Vinod babu Pusuluri,

Teja Anand Pedapudi, RGUKT-NUZVID,

n180845@rguktn.ac.in

Abstract—Summarization of multi pdfs by using Langchain,Gemini API key,Streamlit. A chat bot is implemented by using streamlit frame work.According to the questions the responses are generated

Index Terms—component, formatting, style, styling, insert

I. INTRODUCTION

LangChain enables building application that connect external sources of data and computation to LLMs.This allows you to interact in a chat manner with this LLM, so it remembers previous questions. Finally, we will build an agent - which utilizes an LLM to determine whether or not it needs to fetch data to answer questions.

II. LITERATURE SURVEY PAPER-1

A. Chat with multiple pdfs by using Langchain,Gemini, Streamlit

B. Large Language Models (LLMs) have drawn a lot of attention due to their strong performance on a wide range of natural language tasks, since the release of ChatGPT in November 2022. LLMs' ability of general-purpose language understanding and generation is acquired by training billions of model's parameters on massive amounts of text data, as predicted by scaling laws\cite{The research area of LLMs, while very recent, is evolving rapidly in many different ways. In this paper, we review some of the most prominent LLMs, including three popular LLM families (GPT, LLaMA, PaLM), and discuss their characteristics, contributions and limitations. We also give an overview of techniques developed to build, and augment LLMs. We then survey popular datasets prepared for LLM training, fine-tuning, and evaluation, review widely used LLM evaluation metrics, and compare the performance of several popular LLMs on a set of representative benchmarks. Finally, we conclude the paper by discussing open challenges and future research directions

In the digital age, the dynamics of customer service are evolving, driven by technological advancements and the integration of Large Language Models (LLMs). This research paper introduces a groundbreaking approach to automating customer service using LangChain, a custom LLM tailored for organizations. The paper explores the

obsolescence of traditional customer support techniques, particularly Frequently Asked Questions (FAQs), and proposes a paradigm shift towards responsive, context-aware, and personalized customer interactions. The heart of this innovation lies in the fusion of open-source methodologies, web scraping, fine-tuning, and the seamless integration of LangChain into customer service platforms. This open-source state-of-the-art framework, presented as "Sahaay," demonstrates the ability to scale across industries and organizations, offering real-time support and query resolution. Key elements of this research encompass data collection via web scraping, the role of embeddings, the utilization of Google's Flan T5 XXL, Base and Small language models for knowledge retrieval, and the integration of the chatbot into customer service platforms. The results section provides insights into their performance and use cases, here particularly within an educational institution. This research heralds a new era in customer service, where technology is harnessed to create efficient, personalized, and responsive interactions. Sahaay, powered by LangChain, redefines the customer-company relationship, elevating customer retention, value extraction, and brand image. As organizations embrace LLMs, customer service becomes a dynamic and customer-centric ecosystem.

C. results

the result of a chat with multiple PDFs involves the extraction, processing, and presentation of information from the uploaded documents based on user interactions. The chat application acts as a versatile tool for users to interact with PDF content in a conversational manner, enabling efficient information retrieval and analysis.

III. INTRODUCTION GEMINI AI

The Gemini AI represents a cutting-edge artificial intelligence platform designed to revolutionize industries through intelligent automation and data-driven insights. Leveraging state-of-the-art machine learning algorithms and natural language processing techniques, Gemini AI offers a versatile suite of solutions tailored to address complex challenges across diverse domains.

capabilities encompass a range of functionalities, including predictive analytics, anomaly detection, sentiment analysis,

IV. INTRODUCTION TO FAISSU CPU

Faissu CPU, an extension of the Faiss library optimized for CPU architectures, offers high-performance similarity search capabilities for large-scale datasets. Leveraging advanced algorithms and efficient data structures, Faissu CPU enables rapid indexing, retrieval, and clustering of high-dimensional vectors on standard CPU hardware. Key features of Faissu CPU include support for various similarity measures, including Euclidean distance and inner product, as well as the ability to handle both dense and sparse vectors. The library provides a flexible and user-friendly interface for indexing and querying large datasets, making it suitable for a wide range of applications, including information retrieval, recommendation systems, and data mining. Faissu CPU's performance optimizations ensure efficient memory utilization and multi-threaded parallelism, allowing for fast and scalable similarity search operations on CPUs with multiple cores. Additionally, the library supports incremental indexing and online updates, enabling real-time applications to adapt to changing data streams. With its focus on performance, scalability, and ease of use, Faissu CPU serves as a valuable tool for researchers and practitioners seeking to implement efficient similarity search solutions on CPU architectures, without the need for specialized hardware or GPU acceleration.

V. INTRODUCTION TO STREAMLIT

Streamlit is a Python library designed to streamline the development of interactive web applications for data science and machine learning projects. By providing a simple and intuitive interface, Streamlit enables developers to create powerful web applications with minimal effort, eliminating the need for complex web development frameworks or front-end programming languages. Key features of Streamlit include its declarative syntax, which allows developers to build applications using familiar Python scripting, as well as its automatic widget generation, which dynamically updates the user interface based on changes to the underlying code. Additionally, Streamlit offers seamless integration with popular data science libraries such as Pandas, Matplotlib, and TensorFlow, enabling developers to leverage their existing tools and workflows. Streamlit's real-time feedback loop facilitates rapid prototyping and experimentation, allowing developers to iterate quickly and visualize their data in interactive dashboards or applications. Furthermore, Streamlit's built-in caching and optimization features ensure smooth performance, even when working with large datasets or computationally intensive tasks.

With its focus on simplicity, productivity, and flexibility, Streamlit empowers data scientists and machine learning engineers to build sophisticated web applications that showcase their work and insights, making it a valuable tool for both individual developers and collaborative

and personalized recommendation

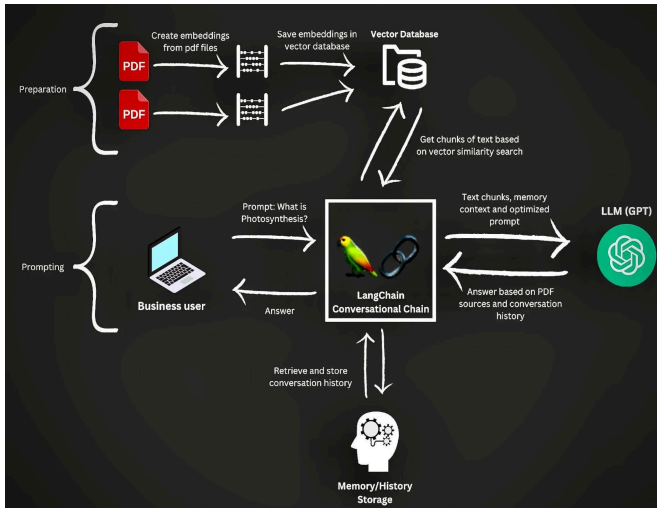
teams for diagnosis of disease.

VI. INTRODUCTION TO PyPDF2

PyPDF2 is a versatile Python library designed to facilitate the processing of PDF files with ease and efficiency. Leveraging PyPDF2, developers can perform a wide range of tasks such as extracting text, merging or splitting PDF documents, adding watermarks, and much more. Key features of PyPDF2 include its intuitive API, which allows developers to interact with PDF files using familiar Python syntax and conventions. This makes it easy to integrate PyPDF2 into existing projects and workflows, whether for data extraction, document generation, or archival purposes. PyPDF2's comprehensive functionality extends to handling encrypted PDFs, navigating through document pages, and extracting metadata such as author, title, and creation date. Additionally, the library offers support for both text-based and image-based PDF content, ensuring flexibility in processing diverse types of PDF documents. With its robust capabilities and user-friendly interface, PyPDF2 serves as a valuable tool for developers, researchers, and data scientists working with PDF files in various domains, including document management, data extraction, and report generation. PyPDF2 is a Python library facilitating easy manipulation of PDF files. With PyPDF2, users can perform tasks such as extracting text, merging or splitting documents, and adding watermarks. It offers an intuitive API, making integration into Python projects straightforward. PyPDF2 handles encrypted PDFs, extracts metadata, and supports both text and image-based content. Ideal for various domains including document management and data extraction, PyPDF2 simplifies PDF file interactions within Python applications.

VII. INTRODUCTION TO DOTENV

Dotenv is a Python library that simplifies the management of application configuration settings stored in environment variables. With dotenv, developers can easily load environment variables from a .env file into their application's runtime environment. This allows for a clean and portable approach to managing sensitive or environment-specific configuration data. By abstracting away the complexities of environment variable handling, dotenv streamlines the development process and enhances application portability across different deployment environments.



VIII. INTRODUCTION TO API

The Gemini API key is a secure credential issued by the Gemini cryptocurrency exchange, granting users programmable access to their account functionalities. It facilitates seamless integration of Gemini's services into external applications, enabling automated trading, retrieval of account data, and other interactions. These keys are generated through the Gemini platform with varying levels of access and security measures. Proper handling of API keys is crucial to safeguarding user accounts and funds. Gemini provides extensive documentation and support to assist developers and traders in utilizing API keys effectively.

IX. INTRODUCTION TO OS ENVIRONMENT

An OS environment abstract encapsulates the essential components and functionalities that enable the operation, management, and interaction of computer systems, providing a platform for running applications and facilitating user productivity.

X. FUTURESCOPE

The future scope of a project integrating LangChain, Gemini API key, and the Streamlit framework for chatting with multiple PDFs holds immense promise in revolutionizing how users interact with and extract insights from textual and financial data. By combining the capabilities of these technologies, the project could offer a multifaceted and dynamic platform for collaborative discussions, data analysis, and cryptocurrency-related insights.

LangChain's integration would enable advanced natural language processing functionalities within the chat interface. Users could engage in conversations about the content of multiple PDFs, leveraging LangChain's capabilities for summarization, sentiment analysis, and keyword extraction. This would facilitate efficient knowledge sharing and exploration, empowering users to extract key information and insights from complex textual data effortlessly.

The integration of the Gemini API key would introduce a new dimension to the chat interface, allowing users to access real-time cryptocurrency data and insights seamlessly. Users could query information such as current market prices, portfolio balances, and trading trends directly within the chat, enhancing their understanding of the cryptocurrency landscape and enabling informed decision-making. This integration would add depth and relevance to the conversations, catering to users interested in financial markets and cryptocurrency investments.

Utilizing the Streamlit framework would enable the development of a user-friendly and interactive interface for the chat application. Streamlit's simplicity and flexibility would allow for the seamless navigation between different PDFs, visualization of LangChain-generated summaries and analyses, and integration of Gemini API data. The application could be tailored to meet the specific needs and preferences of users, providing a personalized and engaging experience for exploring and discussing textual and financial content.

In addition to basic functionalities, the project could incorporate advanced features such as collaborative annotation tools for PDFs, personalized recommendations based on user preferences and browsing history, and integration with external services for additional insights and functionalities. Scalability, performance optimization, and robust security measures would be prioritized to ensure a smooth and secure user experience, even under high loads and concurrent usage.

Overall, the future scope of a project combining LangChain, Gemini API key, and the Streamlit framework for chatting with multiple PDFs is vast and promising. It has the potential to revolutionize how users interact with textual and financial data, facilitating collaborative exploration, knowledge sharing, and informed decision-making in diverse domains.

XI. Conclusion

The integration of LangChain, Gemini API key, and the Streamlit framework in the "Chat with Multiple PDFs" project presents an innovative solution that leverages artificial intelligence and collaborative technologies to enhance the user experience. Through LangChain's AI capabilities, multiple PDFs are trained and summarized, enabling users to extract key insights efficiently. The chat interface responds to user questions, providing relevant information sourced from the analyzed PDFs. The Streamlit framework offers a user-friendly interface where inputs can be entered, submitted, and processed, with outputs displayed seamlessly. Together, these components create a dynamic platform for interactive discussions, knowledge sharing, and informed decision-making, catering to diverse user needs and preferences.

XII. References

- [1] H. Naveed et al., "A Comprehensive Overview of Large Language Models", arXiv, Aug. 2023, [online] Available: <http://arxiv.org/abs/2307.06435>.
- [2] H. Chase, "Python LangChain", Python LangChain, Jan. 2023, [online] Available: https://python.langchain.com/docs/get_started/introduction.
- [3] B. Casselden and R. Pears, "Higher education student pathways to ebook usage and engagement and understanding: Highways and cul de sacs", *J. Librariansh. Inf. Sci.*, vol. 52, no. 2, pp. 601-619, Jun. 2020.- URL {https://scholar.google.com/scholar?q=Higher+education+student+pathways+to+ebook+usage+and+engagement%2C+and+understanding%3A+Highways+and+cul+de+sacs&as_occt=title&hl=en&as_sdt=0%2C31#d=gs_qab&t=1712521138612&u=%23p%3DAzUjkJqHcDoJ.}
- [4] C. Zhou et al., "A Comprehensive Survey on Pretrained Foundation Models: A History from BERT to ChatGPT", arXiv, May 2023, [online] Available: <http://arxiv.org/abs/2302.09419>.
- [5] K. Singhal et al., "Large language models encode clinical knowledge", *Nature*, vol. 620, no. 7972, pp. 172-180, Aug. 2023.
- [6] S. Kamnis, "Generative pre-trained transformers (GPT) for surface engineering", *Surf. Coat. Technol.*, vol. 466, pp. 129680, Aug. 2023.
- [7] P. I. Prayitno, R. P. Pujo Leksono, F. Chai, R. Aldy and W. Budiharto, "Health Chatbot Using Natural Language Processing for Disease Prediction and Treatment", 2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI), pp. 62-67, Oct. 2021.