Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

A Project Report

on

"MeroPDFMitra"

[Code No: COMP 207]

(For partial fulfillment of Third Year/ First Semester in Computer Engineering)

Submitted by:

Aakriti Banjara(05)

Sabin Bhattarai (09)

Sudip Subedi (52)

Regal Adhikari(64)

Submitted to

(Mr Dhiraj Shrestha)

Department of Computer Science and Engineering

Submission Date: January 1, 2024

Bona fide Certificate

This project work on

"MeroPDFMitra"

is the bona fide work of

"Sabin Bhattarai, Regal Adhikari, Aakriti Banjara and Sudip Subedi"

who carried out the project work under my supervision.

Project Supervisor

_____

Prof Dr Sudan Jha
Department of Computer Science and Engineering (DoCSE)

Date: January 1, 2024

ACKNOWLEDGEMENT

We would like to express our sincere appreciation to our supervisor, Dr. Sudan Jha, for his constant encouragement, guidance, understanding, and suggestions during the advancement of our project. Without his valuable help this project would not have been possible. We were able to elevate the idea due to his guidance. We are highly indebted to him for believing in us and for being a constant source of motivation. We would also like to thank our project coordinator Mr. Dhiraj Shrestha for approving our project proposal and assigning an appropriate supervisor for the project. We extend our gratitude to the Kathmandu University Department of Computer Science and Engineering for providing us with a platform where students can work on a topic as a semester-long project and demonstrate our findings.

Abstract

In today's fast-paced world, the efficient retrieval of information from PDF documents is crucial for decision-making and problem-solving. This project proposes a PDF Answer Retrieval System that leverages natural language processing (NLP), machine learning (ML), and user-friendly interfaces to streamline information retrieval. The system integrates LangChain for document indexing and retrieval, Flutter for the frontend, and incorporates features like OCR for text extraction from images and a chatbot for user interactions.By combining the capabilities of natural language processing, machine learning, and user-friendly interfaces, this system will enable users to access specific information effortlessly. The proposed system leverages Langchain for document indexing and retrieval, utilizing Flutter as the frontend framework. On the features part, it integrates an OCR model for text extraction from images and incorporates a chatbot for intuitive user interactions. The application aims to finetune the preexisting models and integrate the required features to make it as interactive and interactive as possible. As technology continues to advance, the proposed system lays the groundwork for efficient and intelligent information retrieval coupled with engaging user interactions.

Keywords: *Machine Learning, Langchain, OCR, Flutter, NLP*

Table of Contents

List of Figures

Acronyms/Abbreviations

OCR: Optical Character Recognition

ML: Machine Learning

PDF: Portable Document Format

NLP: Natural Language Processing

LLM: Large Language Model

I/O: Input/Output

API: Application Programming Interface

NLTK: Natural Language Toolkit

AI: Artificial Intelligence

CPU: Central Processing Unit

GPU: Graphics Processing Unit

RAM: Random-Access Memory

SSD: Solid-State Drive

HDD: Hard Disk Drive

Chapter 1: Introduction

MeroPDFMitra is a project based in processing PDF documents, extracting relevant information from them, and providing efficient retrieval based on user queries. The user is provided with an easy-to-use and practical mobile app which they can use to upload a particular PDF they would like to query about or scan a document which will be converted into text and sent to the server for query and answer retrieval.

1.1 Background

In today's digital age, efficient document retrieval and management are essential for businesses and individuals alike. Document retrieval systems have traditionally relied on simple keyword-based searches, resulting in limited accuracy and relevance. More complex retrieval techniques have become possible as a result of recent developments in natural language processing. However, finding relevant information from a multitude of documents can be time-consuming and challenging. To address this issue, we propose the development of a Multiple Document retrieval System, integrating advanced technologies like LangChain, fine-tuning the pre-existing models, and finally integrating with Flutter on the frontend to develop a mobile application.

To support text extraction from images, an OCR (Optical Character Recognition) model is integrated into the system. This feature will enable users to upload images containing text, and the system will convert the image text into searchable content. Users can then retrieve relevant documents based on both text and image-based queries. The frontend is developed using the Flutter framework, ensuring a cross-platform user interface that is visually appealing and responsive. The API calls between Flutter and the server programs (LangChain, OCR) are made via HTTP, hosting the backend on a Flask server. Flask is perfect for hosting Python applications as it provides higher flexibility than other frameworks and highly encourages scalability.

LangChain forms the backbone of the document retrieval system, enabling advanced natural language processing capabilities. LangChain assists in query expansion, suggesting synonyms, related terms, and relevant concepts to broaden the search scope. LangChain's natural language understanding powers the chatbot's ability to comprehend user queries and generate meaningful responses. This integration enables the chatbot to provide accurate assistance, answer queries about the document retrieval process, and suggest refinements to user queries. This chatbot will enhance user engagement and provide real-time assistance. This will involve techniques such as named entity recognition, sentiment analysis, and semantic similarity to ensure that users will be able to interact with the system, search for documents, preview content, and manage their documents effortlessly. The ultimate goal of this project is to provide a user-centric approach to document query and answer retrieval, enhancing overall satisfaction and productivity.

1.2 Objective

Following are the specific objectives we aimed to achieve upon completion of this project.

1. To design and develop a user-friendly frontend using Flutter that provides an intuitive interface for document management and retrieval
2. To develop a functional language model that can accurately predict answers based on the PDF feed.
3. To implement an OCR model to extract text from images, so to enable users to search through image-based documents.
4. To integrate a chatbot to assist users in answering queries and offer personalized recommendations.
5. To learn machine learning through hands-on experience.

1.3 Motivation and Significance

In this world, quick access to relevant information is essential for making smart decisions. MeroPDFMitra allows professionals to quickly gather information from reports and trends, allowing them to make informed choices. For lawyers dealing with complex legal documents, this tool can make it easier to find specific sections, save time and effort in legal research, and much more. It is also especially useful for students, researchers, professionals, and everyone who wants to retrieve the information they are looking for without going through lengthy PDFs. The implementation of the project involves the use of advanced NLP techniques. These technologies help the system understand the context and meaning of words in PDF documents. By analyzing the content, the system can learn what users are looking for and provide accurate answers from PDF files. This project effectively tackles the challenge of finding the correct answer from PDF files. It aims to simplify this process by using NLP to quickly understand the PDF content and give accurate answers. This project is important because it saves time, improves research and decision-making processes, improves accessibility, and removes language barriers. With the help of advanced NLP techniques, the project is an example of technology's ability to make communicating information more efficient and easier.

Chapter 2 Related Works

AI has been present in our lives for a while now and we have increasingly become dependent on them. Due to this, there have been various works related to our project on the mobile storefronts. However, each application has some variance and not all of them are the same. In our project, we have included OCR that converts physical documents clicked by the cameras into PDF that can be passed for further processing, which has not been implemented yet in some of the related projects that are shown as follows:

2.1. AskPDF- PDF AI chat Assistance

AskPDF is a relatively new app that enables the user to upload their PDF to build a chatbot for them to retrieve the answers from the PDF. It's a very handy app, but the functions are relatively simple and users are bound to insert a PDF, i.e., there is no option to directly feed an image of a document.

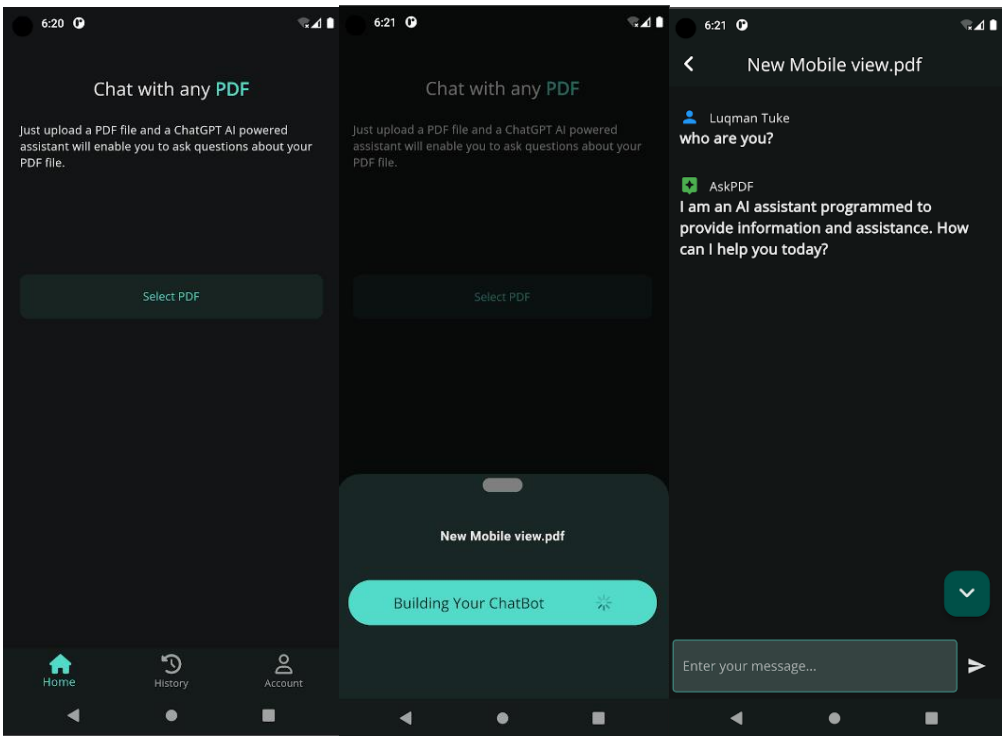


Figure 1: AskPDF- PDF AI chat Assistance

2.2 PDF ChatUp - Chat with any PDF

PDF ChatUp is more advanced PDF upload and answer retrieval application as in addition to PDF files, it also supports '.docx', '.pptx', '.epub', '.md', '.txt', and '.odt' files. It also has multilingual support. However, it is quite pricey and has limited PDF processing capabilities in the free tier. It is quite popular among the other alternatives as of now.

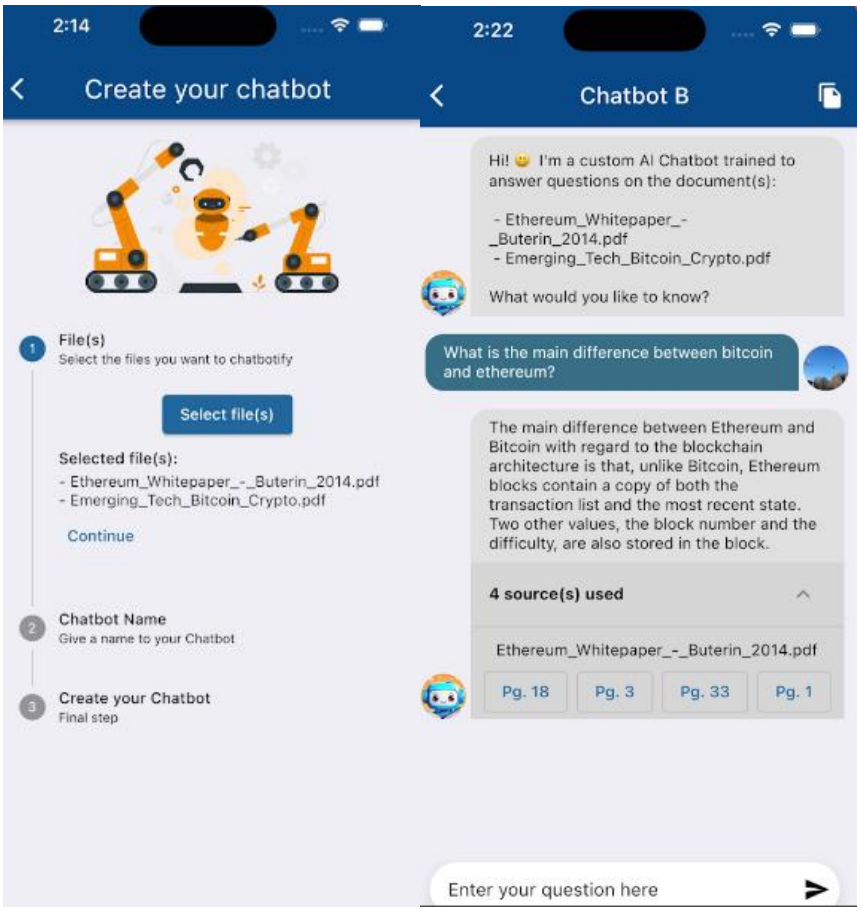


Figure 2 PDF ChatUp Android app

Chapter 3: Methodology

The project was to be completed in below mentioned stages:

3.1 Literature review

First of all, we went through a thorough literature review to understand the notion of question-answering systems, language models, and libraries for manipulating PDFs. This involved studying academic research papers, articles, and real-world projects that address similar challenges. The goal is to build a strong foundation of knowledge and identify the best methodologies, and technologies that we could implement in our project. And ultimately to fix the shortcomings of the preexisting practices.

3.2 Data Collection and Preprocessing

The team identified the type of data that aligns with the model's intended application. The focus is on PDF documents, and the process involves extracting text from these documents. The team uses PDF parsing libraries like PyPDF2 for this purpose. The focus is however on text extraction, splitting, overlapping, embedding and retrieval.

3.3 Feature Engineering

In the context of the project, it involved fine-tuning existing LangChain models, such as those available from Hugging Face. These models can be embedded to chat with provided files. On the basis of datasets and pre existing models we identified features best suitable for our project. We tried to implement different features from different models to provide users with essential system. Feature extraction creates new features that are relevant for improving the user experience and the model's performance.

3.4 Model Selection and Training

This phase involves selecting a suitable Natural Language Processing (NLP) model for the project. The team considers models like GPT-3, available from OpenAI, and models from Hugging Face. We had to consider the cost factor and compatibility in our project. The chosen model is then developed and fine-tuned to suit the specific requirements of the project. Fine-tuning involves adjusting hyperparameters such as the size of text splitting and overlapping to achieve optimal results. The goal is to create a model capable of understanding and responding to user queries effectively.

3.5 Model Evaluation

The model underwent evaluation to determine efficiency of its performance. This evaluation is conducted using a separate set of texts files and the responses generated. The purpose was to provide an unbiased assessment of the model's ability to make accurate responses. The team analyzes various metrics to ensure the model meets the desired level of accuracy and efficiency. Time and accuracy are the major factors taken into consideration. To some extent, we had to face the speed-accuracy tradeoff.

3.6 Model Workflow

To ensure effective querying, the model underwent various steps. Initially, all text is extracted from the document, and thus obtained text is passed into an LLM (Large Language Model) prompt. The texts are splitted into chunks of different sizes whichever gives efficient results, alongside the overlapping to ensure the context of the text is not missed during splitting . In order to organize the texts based on semantic meanings, the texts are then embedded. These embeddings are stored in vector stores along with the corresponding chunks of text. This allows for efficient querying and retrieval of relevant information.

3.7 Application Architecture

LangChain supports the querying process and extraction of information from PDFs based on user prompts. The model is integrated in a mobile application with the use of flutter. This application serves as the frontend,where the users are able to ask, and receive accurate answers to their queries. The architecture ensures a seamless integration of LangChain's capabilities into an intuitive user interface.

3.8 Deployment

The phase involves the integration of the trained AI model into an application. The team developed user-friendly interfaces, interactions, and functionalities. The process includes thorough debugging to identify and resolve any issues. Beta testing is conducted by showing the application to a limited audience to gather user feedback. Based on the feedback, necessary adjustments were made to improve the application's performance and user experience. The user interface is designed to be minimalist and interactive, ensuring users can easily understand and access the results of the model.

3.9 OCR

Optical Character Recognition (OCR) describes the process of converting printed or handwritten text into a digital format with image processing. We use different packages while doing OCR on an image, such as Pillow, Tesseract, and OpenCV.

There are various steps involved in performing OCR on a document, and they are as follows:

1. **Grayscale Conversion:** Convert the color image to grayscale, which simplifies the image for subsequent processing by focusing on brightness information.
2. **Inversion:** Invert the grayscale image that makes dark text on a light background appear as light text on a dark background, often enhancing OCR accuracy.
3. **Blurring:** Apply Gaussian blur, which reduces noise and smooths out small variations, making text edges more distinct.
4. **Thresholding:** Convert the image into a binary image that isolates text regions from the background, creating a clear distinction between text and non-text areas.
5. **Kernel:** Define a structuring element (kernel) that shape is used in morphological operations to modify image structures.
6. **Dilation:** Expand text regions, which enlarges the text areas, making them more prominent for contour detection.
7. **Noise Removal:** Remove small, isolated regions (noise) that eliminate non-text elements that could interfere with OCR recognition.
8. **Contours:** Find outlines of text regions and these contours define the shapes of individual text elements for further processing or extraction.

3.10 System Requirement Specifications

3.10.1. Software Specification

We were able to create this website with the help of many software such as:

- Python
- Visual Studio Code
- Jupyter Notebook
- Flutter
- Flask

Here is a list of all the Python modules that we used to create our project:

- Langchain ver 0.0.331
- HuggingFace ver 0.0.1
- Tensorflow ver 2.15.0
- PyPDF2 ver 3.0.1

3.10.2 Hardware Specifications

A PC with the following Hardware Specifications is required to implement our project:

- A multi-core CPU
- GPU
- RAM with 8 GB or more memory

Chapter 4: Discussion of the Achievements

4.1 Discussion

The project has helped us understand the concept of Natural language processing and mobile application integration alongside different soft skill enhancements in different phases of the project accomplishment. The literature review established a solid foundation, providing insights into existing systems and technologies. The creative approach to feature engineering enhanced the pre existing models for better use cases. Successful selection, training, and evaluation of NLP models, including GPT-3 and Hugging Face, emphasized the necessity of a balanced and efficient approach in integration . The optimized workflow for efficient querying, along with the development of a user-friendly application using Flutter and LangChain integration, led to a seamless user experience. The deployment phase, which included debugging, beta testing, and user feedback, reflected a commitment to delivering a robust PDF Answer Retriever system. Collectively, these achievements signify the project's success in advancing the field of document retrieval and natural language processing.

4.2 Implications

4.1.1 Advancements in Document Retrieval

The successful integration of LangChain, Flutter, and OCR in the PDF Answer Retrieval System implies significant advancements in document retrieval methodologies. The system achieves a fine understanding of the content within PDF documents with the implementation of NLP.

4.1.2 User-Friendly Interfaces

Developing minimalist yet efficient user interfaces for the application ensures that users, regardless of technical expertise, can interact effortlessly with the system.

This user-centric approach has implications for enhancing accessibility and usability, making information retrieval a seamless experience.

4.1.3 Enhanced Information Extraction from Images

The incorporation of OCR technology in the system only increases its capabilities to extract text from images. This achievement holds implications for diverse applications of the PDF Answer Retriever system.

4.1.4 Real-Time Assistance with Chatbot Integration

The successful integration of a chatbot within the system implies real-time assistance for users. The chatbot, powered by LangChain, has implications for improved user engagement and a more interactive information-seeking experience.

4.1.5 Free to use and no character limitation

Unlike the ones available, MeroPDFMitra project is freely accessible, allowing users to utilize its services without any cost. Additionally, there are no character limitations imposed, enabling users to process and extract information from documents without restrictions on text length.

Chapter 5: Conclusion and Recommendation

The development and implementation of the Retrieve Augment Generation system mark a significant achievement in the domain of question-answering chatbots, leveraging advanced technologies and frameworks to facilitate intelligent document analysis and retrieval. Through the integration of langchain, a versatile framework enabling seamless interaction with large language models, this system adeptly addresses user queries related to both PDF documents and images.

The utilization of vector store for storing text embeddings demonstrates a sophisticated approach to efficiently manage and retrieve textual information. This strategic choice not only enhances the system's scalability but also contributes to its robustness in handling diverse datasets. Additionally, the creation of a custom OCR system using pytesseract and OpenCV showcases an innovative solution for extracting text from images, further broadening the scope of document analysis.

The mobile application frontend developed using Flutter offers a user-friendly interface, ensuring accessibility and ease of use for a wider audience. This intuitive interface serves as a gateway for users to interact seamlessly with the system, providing a streamlined experience in querying and retrieving information from documents and images.

In conclusion, the Retrieve Augment Generation system represents a culmination of interdisciplinary efforts, amalgamating computer vision, natural language processing, and mobile development. Its successful implementation signifies a milestone in harnessing cutting-edge technologies to create a comprehensive solution for document-based queries. Moving forward, the system stands as a testament to the potential of AI-driven applications in simplifying information retrieval processes and holds promise for further advancements in the field.

5.1 Limitations

Despite the achievements and successes during the development phase, some areas could be further explored and enhanced in future iterations. Some of the limitations include:

- The model's accuracy is not very efficient when considering the speed-accuracy tradeoff.
- Chatbot responses may require further fine-tuning.
- When providing very large files, the processing time may be impacted.

5.2 Future Enhancements

Advanced OCR Capabilities: Implement advanced Optical Character Recognition (OCR) techniques to significantly enhance the accuracy and efficiency of extracting text from images. Investigate deep learning approaches and sophisticated preprocessing methods to improve the overall OCR performance.

Multi-Document Retrieval: Extend the system to support the retrieval of answers from multiple documents simultaneously. This enhancement will provide users with the ability to analyze information across various files in a single query, improving the system's versatility and user experience.

Fine-Tuning and Model Optimization: Continuously fine-tune the language model used in the system, focusing on optimizing its speed without compromising accuracy. Implement techniques to address the speed-accuracy tradeoff, ensuring a more responsive and efficient system.

Intelligent Chatbot Interactions: Enhance the natural language understanding capabilities of the chatbot component to better comprehend complex queries. Implement sentiment analysis and context-aware responses to provide more personalized and engaging interactions, improving overall user satisfaction.

Scalability and Performance Optimization: Conduct a comprehensive analysis of system performance and optimize its architecture for scalability. Explore cloud-based solutions, distributed computing, and parallel processing to efficiently handle increased user loads, ensuring a seamless and responsive user experience.

References

[1] Adith Sreeram A S, Pappuri Jithendra Sai. (2022, June 15). An Effective Query System Using LLMS and Langchain. IJERT.

https://www.ijert.org/an-effective-query-system-using-llms-and-langchain

[2] Maameri Sami. (2023, May 20). Building a Multi-Document Reader and Chatbot with Langchain and ChatGPT. Better Programming. https://betterprogramming.pub/building-a-multi-document-reader-and-chatbot-with-langchain-and-chatgpt-d1864d47e339

[3] PDF.AI.*PDF.ai | Chat with your PDF documents*. (n.d.). https://pdf.ai

https://play.google.com/store/apps/details?id=com.tukesolutions.askpdf

https://play.google.com/store/apps/details?id=com.fabudable.pdfchatup

[4] Sophia Yang. (2023, April 10).Building a Question Answering PDF Chatbot. Towards Data Science

https://towardsdatascience.com/building-a-question-answering-pdf-chatbot-3e3b6372528c

APPENDIX

APPENDIX-1: GANTT CHART

GANTT CHART

	Tasks	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12
Planning Phase	Scope Definition												
	Research												
	Work Division												
AI modeling Phase	AI Model Selection												
	Data Collection												
	Model Development												
	Model Validation												
Deployment and Beta Testing Phase	Model Deployment												
	Beta Testing												
Documentation and Delivery Phase	Bug and Error Resolution												
	Final Testing												
	Documentation												

Figure 3 Gantt Chart

APPENDIX-2: MODEL WORKFLOW

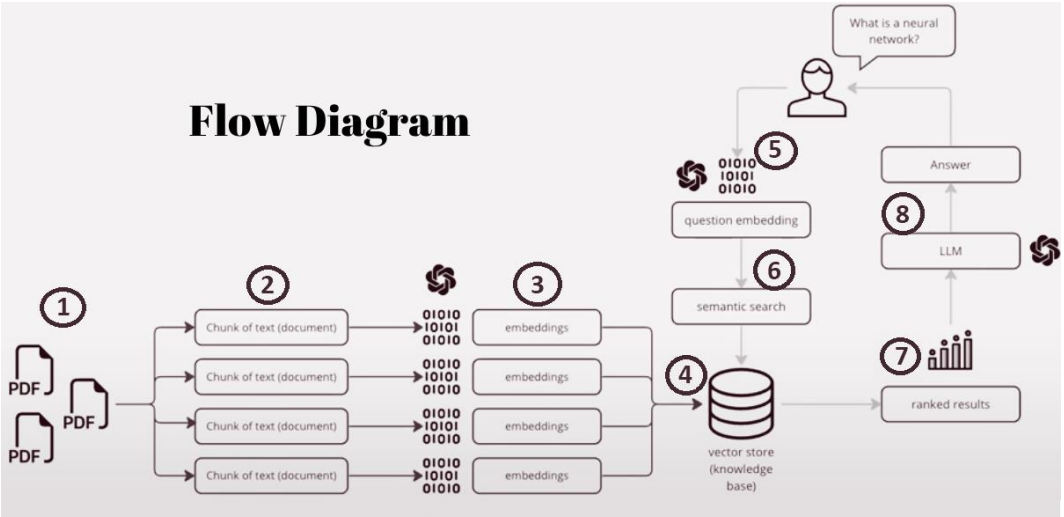


Figure 4 Model Workflow

APPENDIX-3: OCR WORKFLOW

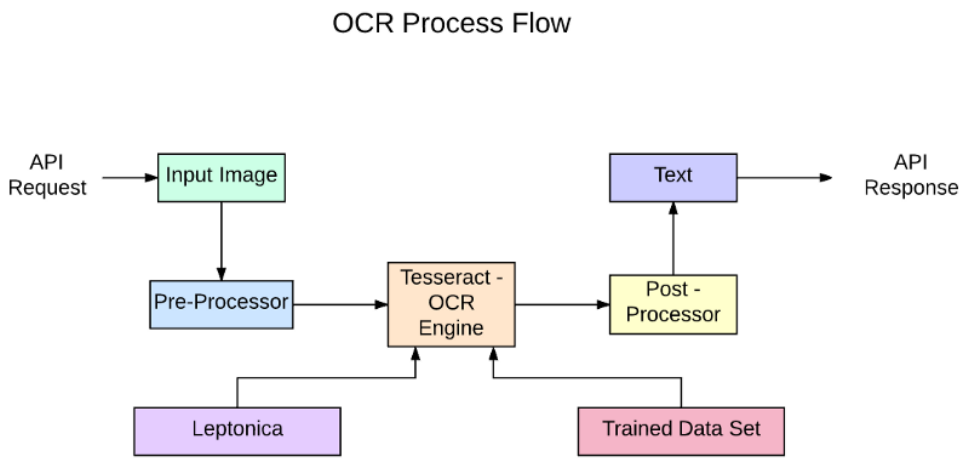
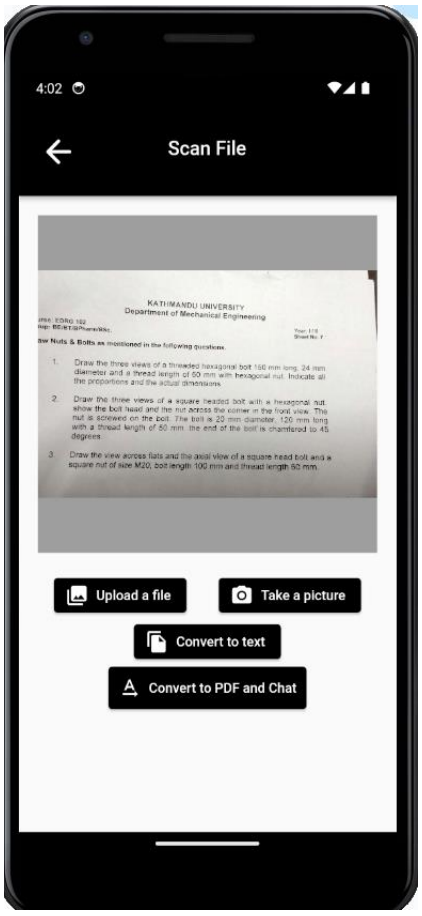
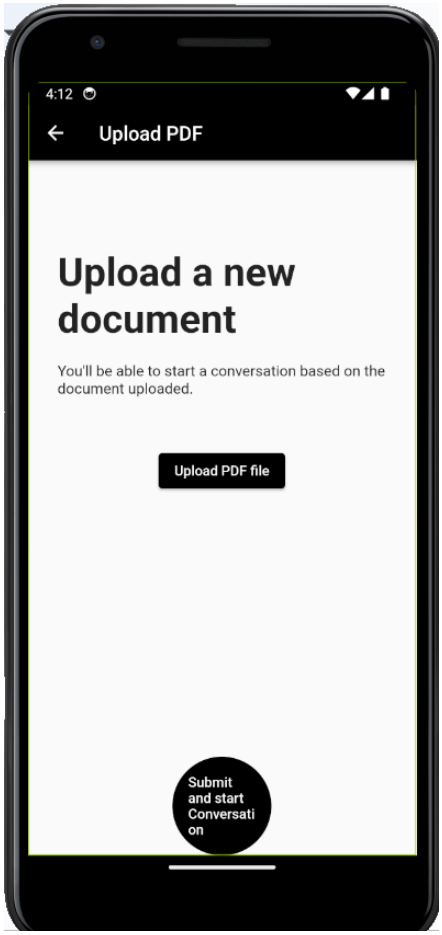


Figure 5 OCR Workflow

APPENDIX-4: MOBILE APP SCREENSHOTS





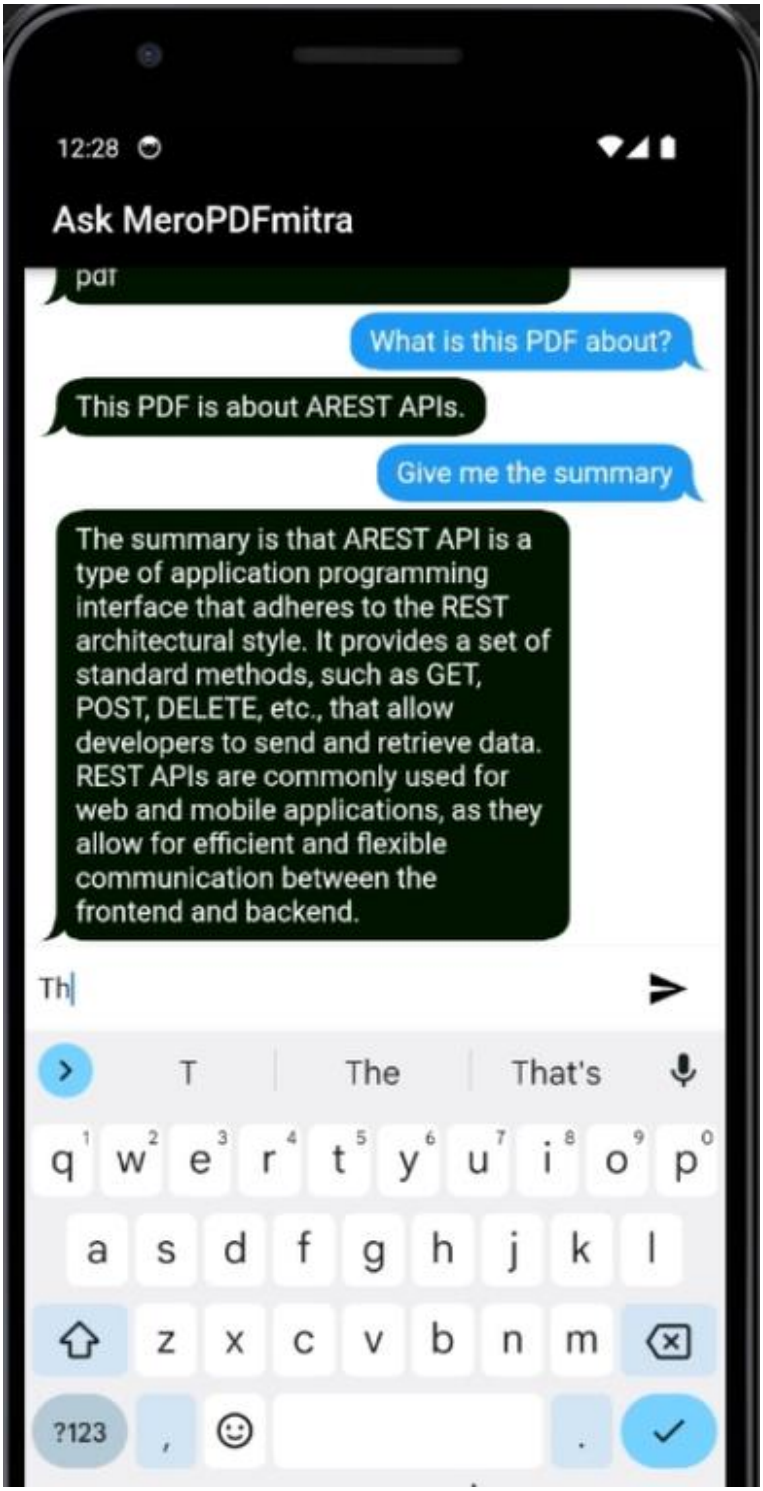


Figure 6 MeroPDFMitra Chatbot