CHAT WITH MULTIPLE PDFS BY USING LANGCHAIN

A report submitted in partial fulfillment of the requirements for award of the Degree of

BACHELOR OF TECHNOLOGY
in
ELECTRONICS AND COMMUNICATION ENGINEERING

by

PEDAPUDI TEJA ANAND(N180845)

Under the Esteem Guidance of
Mr.Chakravarthi Jada
Assistance Professor,EEE Dept

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES
NUZVID, INDIA

DECLARATION

I, **P.Teja Anand(N180845)** hereby declare that the project report entitled done by us under the guidance of Mr.Chakravarthi Jada is submitted for the partial fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering during the academic session September 2023 - April 2024 at RGUKT-Nuzvid. We also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from any websites are mentioned in the references. The results embodied in this project report have not been submitted to any other university or institute for the award of any degree or diploma.

.....

Project Guide: Pedapudi.Teja Anand(N180845)

Date:08-042024

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES, NUZVID, INDIA

THESIS CERTIFICATE

This is to certify that the report entitled **"CHAT WITH MULTIPLE PDFS BY USING LANGCHAIN "** submitted by **P.Teja Anand** to the Department of Electronics and Communication Engineering, Rajiv Gandhi University of Knowledge Technologies - Nuzvid campus, AP for the award of the degree of Bachelor of Technology in ECE is a bonafide record of work carried out by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Mr. Chakravarthi Jada
Project Guide
Asst.Professor
Department of EEE
RGUKT NUZVID

Mr.Shyam Peraka
Head of the Department
Department of ECE
RGUKT NUZVID

ACKNOWLEDGEMENT

I would like to express our profound gratitude and regards to our guide **Mr.Chakravarthi Jada** , for his exemplary guidance, monitoring and constant encouragement to us throughout the B.Tech course. I shall always cherish the time spent with her during the course of this work due to the invaluable knowledge gained in the field of Deep Learning. I extremely grateful for the confidence bestowed in us and entrusting our project entitled "Traffic Sign Board Recognition And Voice Alert System". I express gratitude to Mr.Shyam Peraka(HOD of ECE) and other faculty members for being a source of inspiration and constant encouragement which helped us in completing the project successfully.Finally, yet importantly, I would like to express our heartfelt thanks to our beloved God and parents for their blessings, our friends for their help and wishes for the successful completion of this project.

ABSTRACT

The emergence of LangChain signifies a significant leap forward in facilitating seamless interaction with multiple PDF documents. Leveraging advanced technology, LangChain is adept at comprehending the intricate content of PDFs, adeptly extracting essential information such as key topics and interconnections between ideas. Users are empowered to engage in natural language conversations with LangChain, effortlessly posing queries or discussing topics of interest. In response, LangChain meticulously sifts through the PDFs, retrieving relevant information, and presents it in an intuitive chat-like format. What sets LangChain apart is its remarkable capability to manage dialogues spanning across multiple PDFs simultaneously, making it an invaluable asset for grappling with complex subjects. By providing users with a user-friendly avenue to explore and comprehend information from disparate sources, LangChain streamlines the process, obviating the need for manual document navigation.

CONTENTS

Chapter 1

Introduction

he advent of digitalization has exponentially increased the accessibility of information, yet it has also posed formidable challenges in effectively navigating and synthesizing this abundance of data. Amidst this backdrop, LangChain emerges as a groundbreaking solution, poised to revolutionize the way we interact with multiple PDF documents. In an era where information retrieval often necessitates cumbersome manual searches and disjointed document navigation, LangChain offers a paradigm shift by seamlessly integrating natural language processing (NLP) techniques with the complexities of PDF content. This introduction sets the stage for exploring the transformative potential of LangChain in facilitating intuitive and efficient communication with multi-PDF repositories. By delving into its core principles, functionalities, and implications, this paper aims to illuminate the pivotal role LangChain plays in bridging the gap between information silos and human understanding, thus heralding a new era of streamlined information access and comprehension.

1.1 Objective of the work

The primary objective of "Chat with Multi PDFs by Using LangChain" is to revolutionize the way users interact with and extract information from multiple PDF documents. By harnessing advanced natural language processing (NLP) techniques, LangChain aims to streamline the process of information retrieval by offering a user-friendly chat interface. Through this interface, users can express queries and engage in conversations using everyday language, eliminating the need for complex search syntax or manual document navigation. LangChain also endeavors to enhance users' comprehension of complex topics by synthesizing insights from multiple PDF documents and presenting them in a coherent manner. Additionally, the system aims to support multi-document conversations, allowing users to seamlessly transition between different sources and explore interconnected topics within a single dialogue. Overall, the objective of LangChain is to empower users with efficient tools for accessing, understanding, and leveraging information stored in PDF documents, thus facilitating more informed decision-making and knowledge discovery processes.

1.2 Motivation of the work

1.2.1 Information Overloaded

With the exponential growth of digital data, individuals and organizations are inundated with vast amounts of information stored across numerous PDF documents. This abundance of data often leads to information overload, making it challenging to locate and extract relevant insights efficiently.

1.2.2 Fragmentation of Information

Relevant information on a particular topic may be dispersed across multiple PDF documents, requiring users to sift through numerous files to compile a comprehensive understanding. This fragmentation of information increases the time and effort required for knowledge acquisition and synthesis.

1.2.3 Need for Natural Language Interaction

Traditional search interfaces often require users to formulate precise search queries or navigate complex menus, posing barriers to individuals who may not be familiar with technical search syntax or terminology. There is a growing demand for more intuitive and user-friendly methods of interacting with digital information.

1.2.4 Desire for Comprehensive Insights

Researchers, students, professionals, and individuals across various domains often need to explore complex topics that span multiple documents. However, existing tools may lack the capability to integrate insights from diverse sources, limiting the depth and breadth of analysis.
The motivation behind LangChain is to develop a sophisticated yet user-friendly solution that empowers individuals to efficiently navigate, search, and comprehend information stored across multiple PDF documents.

1.3 Methodology

It encompasses a systematic approach that integrates natural language processing (NLP) techniques with document analysis and retrieval mechanisms. Firstly, LangChain ingests a collection of PDF documents provided by the user, employing optical character recognition (OCR) techniques to extract textual content. This raw text undergoes pre-processing steps to standardize formats and remove noise. Advanced NLP models process and understand user queries expressed in natural language, including tasks such as intent classification, named entity recognition (NER), and sentiment analysis.

Chapter 2

Literature Review

It involves several intersecting areas of research, each contributing to the understanding and development of such a tool. In the realm of natural language processing (NLP) and document analysis, extensive literature exists on techniques for parsing, analyzing, and extracting information from textual data, including PDF documents. Studies on named entity recognition (NER), sentiment analysis, topic modeling, and relationship extraction provide foundational knowledge for processing PDF content. Additionally, research on chatbots and conversational agents offers insights into designing interactive interfaces and managing dialogue flows, which are essential components of LangChain's functionality. Moreover, literature on information retrieval and knowledge graphs informs the structured representation of information within LangChain, facilitating efficient retrieval and synthesis of information from multiple PDF sources. While there may not be specific literature on "Chat with Multiple PDFs Using LangChain" as a standalone concept, synthesizing findings from these related areas provides a foundational understanding and sets the stage for the development and exploration of LangChain's capabilities in facilitating intuitive communication and information retrieval with PDF documents.

Chapter 3

Proposed Model

It involves a multi-step process that integrates advanced natural language processing (NLP) techniques with sophisticated document analysis and retrieval mechanisms. At its core, LangChain employs a combination of machine learning algorithms, semantic analysis, and information extraction methods to understand and respond to user queries in a conversational manner while seamlessly accessing and synthesizing information from multiple PDF documents.

Firstly, LangChain ingests a collection of PDF documents, employing optical character recognition (OCR) and parsing techniques to extract the textual content from each document. This step ensures that LangChain can access and analyze the information contained within the PDFs, regardless of their format or structure.

Next, LangChain utilizes advanced NLP models to process and understand the textual content extracted from the PDFs. This involves tasks such as named entity recognition (NER), topic modeling, sentiment analysis, and relationship extraction. By analyzing the content at both the document and sentence levels, LangChain can identify key topics, entities, and relationships within the documents, facilitating more accurate information retrieval.

Once the textual content has been processed and analyzed, LangChain constructs a structured representation of the information landscape, which serves as the basis for responding to user queries. This representation may take the form of a knowledge graph, where entities and their relationships are organized hierarchically, or a semantic index that maps concepts to their corresponding documents and contexts.

When a user interacts with LangChain through a chat interface, the system interprets the user's query using natural language understanding techniques. By analyzing the semantics and intent behind the query, LangChain determines the user's information needs and identifies relevant documents or passages that contain the requested information.

LangChain then retrieves the relevant information from the PDF documents, either

by directly accessing the text or by referencing the structured representation generated during the preprocessing stage. The system synthesizes the retrieved information into a coherent response, presenting it to the user in a chat-like format that is easy to understand and engage with.

One of the key features of LangChain is its ability to handle conversations that span multiple PDF documents. By maintaining context across multiple interactions, LangChain can seamlessly integrate information from different sources and provide comprehensive responses to complex queries or topics.

Chapter 4

Requirement And Analysis

4.1 Requirements

4.1.1 Software Requirements

- Operating System: Windows, macOS
- Programming language: Python 3.7 or above
- Frame works : TensorFlow, PyTorch, or Keras
- Development Environment: IDEs (Integrated Development Environments) like Jupyter Notebook, PyCharm, or Visual Studio Code
- Dataset: Day2night, Seasonal,and face attribute modification etc...

4.1.2 Hardware Requirements

- Computer or Device
- CPU: multi core CPU
- RAM: 16GB or higher
- Storage: Solid State Drive (SSD) 256 GB or higher for fast accessing data
- Network Interface: WIFI or Ethernet

Chapter 5

Langchain

5.1 Langchain

LangChain operates as an integrated system that combines advanced natural language processing (NLP) models with sophisticated document analysis and retrieval mechanisms to enable seamless interaction with multiple PDF documents. At its core, LangChain comprises several interconnected components and models, each playing a crucial role in facilitating efficient communication and information retrieval.

5.1.1 Models in Langchain

Input Processing and Preprocessing:
LangChain begins by ingesting a collection of PDF documents provided by the user. Optical character recognition (OCR) techniques are applied to extract the textual content from the PDFs. This raw text undergoes preprocessing steps, including cleaning, tokenization, and normalization, to standardize formats and remove noise, ensuring consistency across documents.

Natural Language Understanding (NLU):
Advanced NLP models form the backbone of LangChain's natural language understanding capabilities. These models are trained on vast amounts of text data and are capable of parsing and understanding user queries expressed in natural language. Key NLP tasks include named entity recognition (NER), sentiment analysis, intent classification, and syntactic parsing, enabling LangChain to interpret the semantics and intent behind user queries accurately.

Document Analysis Models:

LangChain employs various NLP models for analyzing the textual content of PDF documents. Topic modeling techniques, such as Latent Dirichlet Allocation (LDA) or Non-negative Matrix Factorization (NMF), are utilized to identify key topics and themes within the documents. Entity recognition models extract named entities, such as persons, organizations, and locations, while relationship extraction models identify connections and associations between entities and concepts mentioned in the documents. Additionally, sentiment analysis models assess the sentiment expressed in the text, providing insights into the tone and mood of the content.

Structured Representation:

The information extracted from the PDF documents is organized into a structured representation of the information landscape. This may take the form of a knowledge graph, where entities, topics, and relationships are mapped hierarchically, or a semantic index that captures the semantic similarities and connections between different documents and concepts.

Conversation Management:

LangChain maintains context across user interactions to facilitate multi-turn conversations. Contextual information from previous interactions is stored and utilized to provide relevant responses and maintain coherence in the dialogue. Dialog management models help manage the flow of conversation, ensuring that user queries are addressed appropriately and comprehensively.

Response Generation and Presentation:

Retrieved information is synthesized into coherent responses using natural language generation models. These models generate human-like text based on the extracted information and context, presenting responses to the user in a chat-like format. Response presentation models ensure that the information is formatted and displayed effectively within the chat interface, making the interaction intuitive and user-friendly.

Multi-Document Integration:

LangChain seamlessly integrates information from multiple PDF documents to provide comprehensive responses. Models for semantic linking and summarization aggregate insights from diverse sources, synthesizing them into a unified response that addresses the user's query or topic of interest comprehensively.

5.2 Langchain Architecture

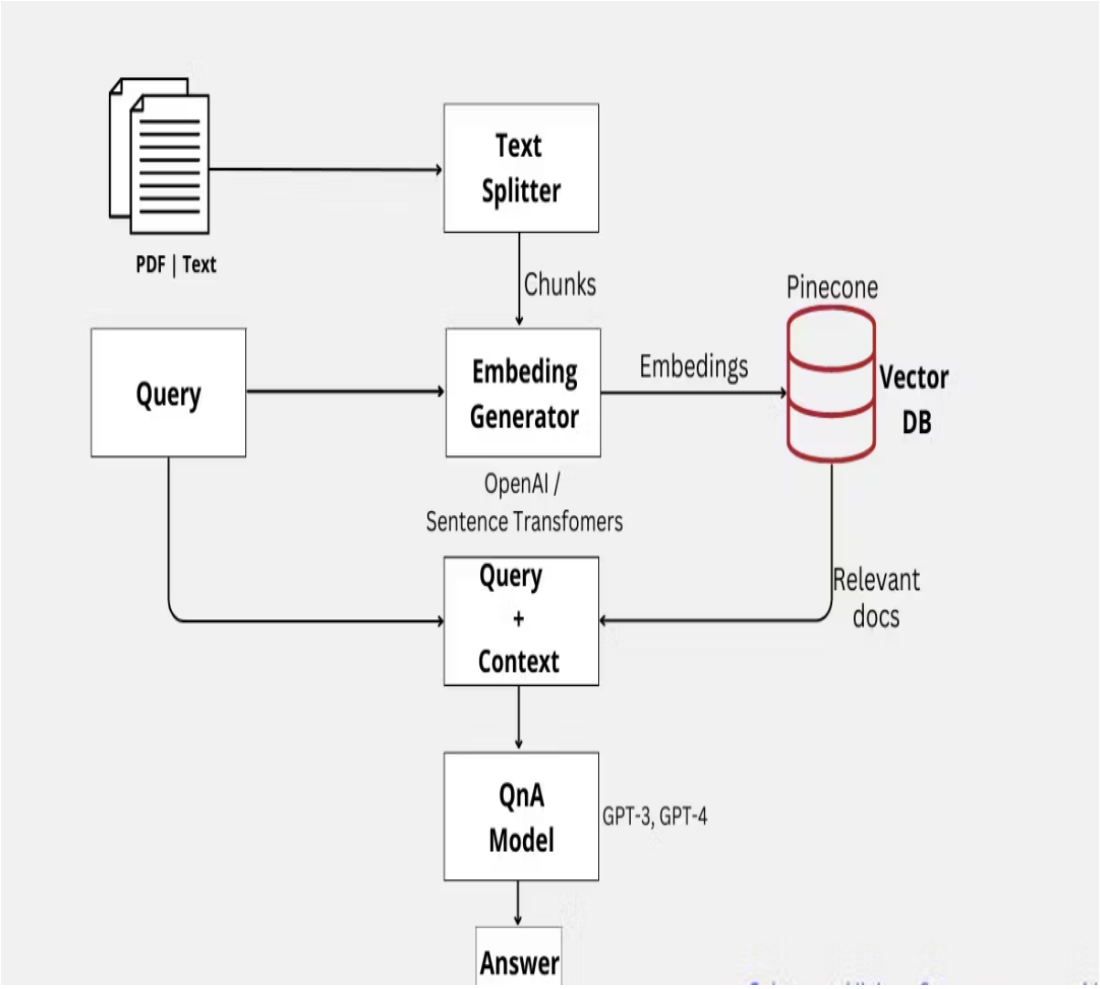


Figure 5.1: Langchain Architecture

5.3 Loss Functions in Langchain

5.3.1 Natural Language Understanding (NLU):

For tasks like intent classification and named entity recognition (NER), cross-entropy loss is commonly used. This loss function measures the discrepancy between predicted labels and ground truth labels. Additionally, for sentiment analysis tasks, binary cross-entropy loss or categorical cross-entropy loss may be employed, depending on the number of sentiment classes.

5.3.2 Document Analysis:

Loss functions for document analysis tasks such as topic modeling, named entity recognition (NER), relationship extraction, and sentiment analysis could include cross-entropy loss or other appropriate metrics based on the specific models used for each task.

5.3.3 Dialog Management:

Dialog management in LangChain may involve reinforcement learning-based approaches where the loss function is defined in terms of maximizing a reward signal. Reinforcement learning algorithms such as policy gradient methods can be used to train dialog management policies by maximizing cumulative rewards over dialog interactions.

5.3.4 Response Generation:

For generating responses in a conversational context, sequence-to-sequence models may be employed. In such cases, sequence-level loss functions like cross-entropy loss can be used to measure the discrepancy between predicted responses and ground truth responses.

5.3.5 Semantic Linking and Summarization:

Loss functions for tasks such as semantic linking and summarization could vary based on the specific methods and models used. For example, sequence-to-sequence models may use cross-entropy loss for sequence generation tasks, while graph-based models may employ specialized loss functions tailored to graph-based representations.

5.4 Implementation

```
[10] pip install streamlit

Requirement already satisfied: streamlit in /usr/local/lib/python3.10/dist-packages (1.33.0)
Requirement already satisfied: altair<6,>=4.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (4.2.2)
Requirement already satisfied: blinker<2,>=1.0.0 in /usr/lib/python3/dist-packages (from streamlit) (1.4)
Requirement already satisfied: cachetools<6,>=4.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (5.3.3)
Requirement already satisfied: click<9,>=7.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (8.1.7)
Requirement already satisfied: numpy<2,>=1.19.3 in /usr/local/lib/python3.10/dist-packages (from streamlit) (1.25.2)
Requirement already satisfied: packaging<25,>=16.8 in /usr/local/lib/python3.10/dist-packages (from streamlit) (24.0)
Requirement already satisfied: pandas<3,>=1.3.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (2.0.3)
Requirement already satisfied: pillow<11,>=7.1.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (9.4.0)
Requirement already satisfied: protobuf<5,>=3.20 in /usr/local/lib/python3.10/dist-packages (from streamlit) (3.20.3)
Requirement already satisfied: pyarrow>=7.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (14.0.2)
Requirement already satisfied: requests<3,>=2.27 in /usr/local/lib/python3.10/dist-packages (from streamlit) (2.31.0)
Requirement already satisfied: rich<14,>=10.14.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (13.7.1)
Requirement already satisfied: tenacity<9,>=8.1.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (8.2.3)
Requirement already satisfied: toml<2,>=0.10.1 in /usr/local/lib/python3.10/dist-packages (from streamlit) (0.10.2)
Requirement already satisfied: typing-extensions<5,>=4.3.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (4.10.0)
Requirement already satisfied: gitpython!=3.1.19,<4,>=3.0.7 in /usr/local/lib/python3.10/dist-packages (from streamlit) (3.1.43)
Requirement already satisfied: pydeck<1,>=0.8.0b4 in /usr/local/lib/python3.10/dist-packages (from streamlit) (0.8.1b0)
Requirement already satisfied: tornado<7,>=6.0.3 in /usr/local/lib/python3.10/dist-packages (from streamlit) (6.3.3)
Requirement already satisfied: watchdog>=2.1.5 in /usr/local/lib/python3.10/dist-packages (from streamlit) (4.0.0)
Requirement already satisfied: entrypoints in /usr/local/lib/python3.10/dist-packages (from altair<6,>=4.0->streamlit) (0.4)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from altair<6,>=4.0->streamlit) (3.1.3)
Requirement already satisfied: jsonschema>=3.0 in /usr/local/lib/python3.10/dist-packages (from altair<6,>=4.0->streamlit) (4.19.2)
Requirement already satisfied: toolz in /usr/local/lib/python3.10/dist-packages (from altair<6,>=4.0->streamlit) (0.12.1)
Requirement already satisfied: gitdb<5,>=4.0.1 in /usr/local/lib/python3.10/dist-packages (from gitpython!=3.1.19,<4,>=3.0.7->streamlit) (4.0.11)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas<3,>=1.3.0->streamlit) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas<3,>=1.3.0->streamlit) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas<3,>=1.3.0->streamlit) (2024.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.27->streamlit) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.27->streamlit) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.27->streamlit) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.27->streamlit) (2024.2.2)
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from rich<14,>=10.14.0->streamlit) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from rich<14,>=10.14.0->streamlit) (2.16.1)
Requirement already satisfied: smmap<6,>=3.0.1 in /usr/local/lib/python3.10/dist-packages (from gitdb<5,>=4.0.1->gitpython!=3.1.19,<4,>=3.0.7->streamlit) (5.0.1)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->altair<6,>=4.0->streamlit) (2.1.5)
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair<6,>=4.0->streamlit) (23.2.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair<6,>=4.0->streamlit) (2023.12.1)
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair<6,>=4.0->streamlit) (0.34.0)
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair<6,>=4.0->streamlit) (0.18.0)
Requirement already satisfied: mdurl<=0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.0->rich<14,>=10.14.0->streamlit) (0.1.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas<3,>=1.3.0->streamlit) (1.16.0)

[11] pip install -q -U google-generativeai
```

0s completed at 8:49 AM

```
[11] pip install -q -U google-generativeai
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 137.4/137.4 kB 1.4 MB/s eta 0:00:00

[12] pip install python-dotenv

Collecting python-dotenv
  Downloading python_dotenv-1.0.1-py3-none-any.whl (19 kB)
Installing collected packages: python-dotenv
Successfully installed python-dotenv-1.0.1

[13] pip install langchain
  Downloading langchain-0.1.14-py3-none-any.whl (812 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 812.8/812.8 kB 4.9 MB/s eta 0:00:00
Requirement already satisfied: PyYAML>=5.3 in /usr/local/lib/python3.10/dist-packages (from langchain) (6.0.1)
Requirement already satisfied: SQLAlchemy<3,>=1.4 in /usr/local/lib/python3.10/dist-packages (from langchain) (2.0.29)
Requirement already satisfied: aiohttp<4.0.0,>=3.8.3 in /usr/local/lib/python3.10/dist-packages (from langchain) (3.9.3)
Requirement already satisfied: async-timeout<5.0.0,>=4.0.0 in /usr/local/lib/python3.10/dist-packages (from langchain) (4.0.3)
Collecting dataclasses-json<0.7,>=0.5.7 (from langchain)
  Downloading dataclasses_json-0.6.4-py3-none-any.whl (28 kB)
Collecting jsonpatch<2.0,>=1.33 (from langchain)
  Downloading jsonpatch-1.33-py2.py3-none-any.whl (12 kB)
Collecting langchain-community<0.1,>=0.0.30 (from langchain)
  Downloading langchain_community-0.0.31-py3-none-any.whl (1.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.9/1.9 MB 24.3 MB/s eta 0:00:00
Collecting langchain-core<0.2.0,>=0.1.37 (from langchain)
  Downloading langchain_core-0.1.40-py3-none-any.whl (276 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 276.8/276.8 kB 24.4 MB/s eta 0:00:00
Collecting langchain-text-splitters<0.1,>=0.0.1 (from langchain)
  Downloading langchain_text_splitters-0.0.1-py3-none-any.whl (21 kB)
Collecting langsmith<0.2.0,>=0.1.17 (from langchain)
  Downloading langsmith-0.1.40-py3-none-any.whl (87 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 87.5/87.5 kB 10.2 MB/s eta 0:00:00
Requirement already satisfied: numpy<2,>=1 in /usr/local/lib/python3.10/dist-packages (from langchain) (1.25.2)
Requirement already satisfied: pydantic<3,>=1 in /usr/local/lib/python3.10/dist-packages (from langchain) (2.6.4)
Requirement already satisfied: requests<3,>=2 in /usr/local/lib/python3.10/dist-packages (from langchain) (2.31.0)
Requirement already satisfied: tenacity<9.0.0,>=8.1.0 in /usr/local/lib/python3.10/dist-packages (from langchain) (8.2.3)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (1.3.1)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (23.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (1.4.1)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (6.0.5)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (1.9.4)
Collecting marshmallow<4.0.0,>=3.18.0 (from dataclasses-json<0.7,>=0.5.7->langchain)
  Downloading marshmallow-3.21.1-py3-none-any.whl (49 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 49.4/49.4 kB 6.4 MB/s eta 0:00:00
Collecting typing-inspect<1,>=0.4.0 (from dataclasses-json<0.7,>=0.5.7->langchain)
```

0s completed at 8:49 AM

```
Requirement already satisfied: annotated-types>=0.4.0 in /usr/local/lib/python3.10/dist-packages (from pydantic<3,>=1->langchain) (0.6.0)
Requirement already satisfied: pydantic-core==2.16.3 in /usr/local/lib/python3.10/dist-packages (from pydantic<3,>=1->langchain) (2.16.3)
Requirement already satisfied: typing-extensions>=4.6.1 in /usr/local/lib/python3.10/dist-packages (from pydantic<3,>=1->langchain) (4.10.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchain) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchain) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchain) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchain) (2024.2.2)
Requirement already satisfied: greenlet!=0.4.17 in /usr/local/lib/python3.10/dist-packages (from SQLAlchemy<3,>=1.4->langchain) (3.0.3)
Collecting mypy-extensions>=0.3.0 (from typing-inspect<1,>=0.4.0->dataclasses-json<0.7,>=0.5.7->langchain)
  Downloading mypy_extensions-1.0.0-py3-none-any.whl (4.7 kB)
Installing collected packages: packaging, orjson, mypy-extensions, jsonpointer, typing-inspect, marshmallow, jsonpatch, langsmith, dataclasses-json, langchain-core, langchain-text-splitter
  Attempting uninstall: packaging
    Found existing installation: packaging 24.0
    Uninstalling packaging-24.0:
      Successfully uninstalled packaging-24.0
Successfully installed dataclasses-json-0.6.4 jsonpatch-1.33 jsonpointer-2.4 langchain-0.1.14 langchain-community-0.0.31 langchain-core-0.1.40 langchain-text-splitters-0.0.1 langsmith-0.1.
```

```
[14] pip install PyPDF2

Collecting PyPDF2
  Downloading pypdf2-3.0.1-py3-none-any.whl (232 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 232.6/232.6 kB 1.9 MB/s eta 0:00:00
Installing collected packages: PyPDF2
Successfully installed PyPDF2-3.0.1
```

```
[15] pip install chromadb

Collecting chromadb
  Downloading chromadb-0.4.24-py3-none-any.whl (525 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 525.5/525.5 kB 3.4 MB/s eta 0:00:00
Requirement already satisfied: build>=1.0.3 in /usr/local/lib/python3.10/dist-packages (from chromadb) (1.2.1)
Requirement already satisfied: requests>=2.28 in /usr/local/lib/python3.10/dist-packages (from chromadb) (2.31.0)
Requirement already satisfied: pydantic>=1.9 in /usr/local/lib/python3.10/dist-packages (from chromadb) (2.6.4)
Collecting chroma-hnswlib==0.7.3 (from chromadb)
  Downloading chroma_hnswlib-0.7.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.4/2.4 MB 22.6 MB/s eta 0:00:00
Collecting fastapi>=0.95.2 (from chromadb)
  Downloading fastapi-0.110.1-py3-none-any.whl (91 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 91.9/91.9 kB 12.2 MB/s eta 0:00:00
Collecting uvicorn[standard]>=0.18.3 (from chromadb)
  Downloading uvicorn-0.29.0-py3-none-any.whl (60 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 60.8/60.8 kB 7.7 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.22.5 in /usr/local/lib/python3.10/dist-packages (from chromadb) (1.25.2)
Collecting posthog>=2.4.0 (from chromadb)
  Downloading posthog-3.5.0-py2.py3-none-any.whl (41 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 41.3/41.3 kB 4.6 MB/s eta 0:00:00
Requirement already satisfied: typing-extensions>=4.5.0 in /usr/local/lib/python3.10/dist-packages (from chromadb) (4.10.0)
Collecting pulsar-client>=3.1.0 (from chromadb)
  Downloading pulsar_client-3.4.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (5.4 MB)
```

✓ 0s completed at 8:49 AM

```
  Attempting uninstall: importlib-metadata
    Found existing installation: importlib_metadata 7.1.0
    Uninstalling importlib_metadata-7.1.0:
      Successfully uninstalled importlib_metadata-7.1.0
Successfully installed asgiref-3.8.1 backoff-2.2.1 bcrypt-4.1.2 chroma-hnswlib-0.7.3 chromadb-0.4.24 coloredlogs-15.0.1 deprecated-1.2.14 fastapi-0.110.1 h11-0.14.0 httptools-0.6.1 humanfr
```

```
[16] pip install faiss-cpu

Collecting faiss-cpu
  Downloading faiss_cpu-1.8.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (27.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 27.0/27.0 MB 28.1 MB/s eta 0:00:00
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from faiss-cpu) (1.25.2)
Installing collected packages: faiss-cpu
Successfully installed faiss-cpu-1.8.0
```

```
pip install langchain_google_genai

Collecting langchain_google_genai
  Downloading langchain_google_genai-1.0.1-py3-none-any.whl (28 kB)
Requirement already satisfied: google-generativeai<0.5.0,>=0.4.1 in /usr/local/lib/python3.10/dist-packages (from langchain_google_genai) (0.4.1)
Requirement already satisfied: langchain-core<0.2,>=0.1 in /usr/local/lib/python3.10/dist-packages (from langchain_google_genai) (0.1.40)
Requirement already satisfied: google-ai-generativelanguage==0.4.0 in /usr/local/lib/python3.10/dist-packages (from google-generativeai<0.5.0,>=0.4.1->langchain_google_genai) (0.4.0)
Requirement already satisfied: google-auth>=2.15.0 in /usr/local/lib/python3.10/dist-packages (from google-generativeai<0.5.0,>=0.4.1->langchain_google_genai) (2.27.0)
Requirement already satisfied: google-api-core in /usr/local/lib/python3.10/dist-packages (from google-generativeai<0.5.0,>=0.4.1->langchain_google_genai) (2.11.1)
Requirement already satisfied: protobuf in /usr/local/lib/python3.10/dist-packages (from google-generativeai<0.5.0,>=0.4.1->langchain_google_genai) (3.20.3)
Requirement already satisfied: pydantic in /usr/local/lib/python3.10/dist-packages (from google-generativeai<0.5.0,>=0.4.1->langchain_google_genai) (2.6.4)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from google-generativeai<0.5.0,>=0.4.1->langchain_google_genai) (4.66.2)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packages (from google-generativeai<0.5.0,>=0.4.1->langchain_google_genai) (4.10.0)
Requirement already satisfied: proto-plus<2.0.0dev,>=1.22.3 in /usr/local/lib/python3.10/dist-packages (from google-ai-generativelanguage==0.4.0->google-generativeai<0.5.0,>=0.4.1->langcha
Requirement already satisfied: PyYAML>=5.3 in /usr/local/lib/python3.10/dist-packages (from langchain-core<0.2,>=0.1->langchain_google_genai) (6.0.1)
Requirement already satisfied: jsonpatch<2.0,>=1.33 in /usr/local/lib/python3.10/dist-packages (from langchain-core<0.2,>=0.1->langchain_google_genai) (1.33)
Requirement already satisfied: langsmith<0.2.0,>=0.1.0 in /usr/local/lib/python3.10/dist-packages (from langchain-core<0.2,>=0.1->langchain_google_genai) (0.1.40)
Requirement already satisfied: packaging<24.0,>=23.2 in /usr/local/lib/python3.10/dist-packages (from langchain-core<0.2,>=0.1->langchain_google_genai) (23.2)
Requirement already satisfied: tenacity<9.0.0,>=8.1.0 in /usr/local/lib/python3.10/dist-packages (from langchain-core<0.2,>=0.1->langchain_google_genai) (8.2.3)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from google-auth>=2.15.0->google-generativeai<0.5.0,>=0.4.1->langchain_google_genai) (5.3.
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from google-auth>=2.15.0->google-generativeai<0.5.0,>=0.4.1->langchain_google_genai) (0.4.0
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.10/dist-packages (from google-auth>=2.15.0->google-generativeai<0.5.0,>=0.4.1->langchain_google_genai) (4.9)
Requirement already satisfied: jsonpointer>=1.9 in /usr/local/lib/python3.10/dist-packages (from jsonpatch<2.0,>=1.33->langchain-core<0.2,>=0.1->langchain_google_genai) (2.4)
Requirement already satisfied: orjson<4.0.0,>=3.9.14 in /usr/local/lib/python3.10/dist-packages (from langsmith<0.2.0,>=0.1.0->langchain-core<0.2,>=0.1->langchain_google_genai) (3.10.0)
Requirement already satisfied: requests<3,>=2 in /usr/local/lib/python3.10/dist-packages (from langsmith<0.2.0,>=0.1.0->langchain-core<0.2,>=0.1->langchain_google_genai) (2.31.0)
Requirement already satisfied: annotated-types>=0.4.0 in /usr/local/lib/python3.10/dist-packages (from pydantic->google-generativeai<0.5.0,>=0.4.1->langchain_google_genai) (0.6.0)
Requirement already satisfied: pydantic-core==2.16.3 in /usr/local/lib/python3.10/dist-packages (from pydantic->google-generativeai<0.5.0,>=0.4.1->langchain_google_genai) (2.16.3)
Requirement already satisfied: googleapis-common-protos<2.0.dev0,>=1.56.2 in /usr/local/lib/python3.10/dist-packages (from google-api-core->google-generativeai<0.5.0,>=0.4.1->langchain_goo
Requirement already satisfied: grpcio<2.0dev,>=1.33.2 in /usr/local/lib/python3.10/dist-packages (from google-api-core->google-generativeai<0.5.0,>=0.4.1->langchain_google_genai) (1.62.1)
Requirement already satisfied: grpcio-status<2.0.dev0,>=1.33.2 in /usr/local/lib/python3.10/dist-packages (from google-api-core->google-generativeai<0.5.0,>=0.4.1->langchain_google_genai)
Requirement already satisfied: pyasn1<0.7.0,>=0.4.6 in /usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1->google-auth>=2.15.0->google-generativeai<0.5.0,>=0.4.1->langchai
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langsmith<0.2.0,>=0.1.0->langchain-core<0.2,>=0.1->langchain_google
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langsmith<0.2.0,>=0.1.0->langchain-core<0.2,>=0.1->langchain_google_genai) (3.6
```

✓ 0s completed at 8:49 AM

```
[17] Requirement already satisfied: grpcio-status<2.0.dev0,>=1.33.2 in /usr/local/lib/python3.10/dist-packages (from google-api-core->google-generativeai<0.5.0,>=0.4.1->langchain_google_genai)
     Requirement already satisfied: pyasn1<0.7.0,>=0.4.6 in /usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1->google-auth>=2.15.0->google-generativeai<0.5.0,>=0.4.1->langchai
     Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langsmith<0.2.0,>=0.1.0->langchain-core<0.2,>=0.1->langchain_google
     Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langsmith<0.2.0,>=0.1.0->langchain-core<0.2,>=0.1->langchain_google_genai) (3.6
     Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langsmith<0.2.0,>=0.1.0->langchain-core<0.2,>=0.1->langchain_google_genai
     Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langsmith<0.2.0,>=0.1.0->langchain-core<0.2,>=0.1->langchain_google_genai
     Installing collected packages: langchain_google_genai
     Successfully installed langchain_google_genai-1.0.1
```

```python
import streamlit as st
from PyPDF2 import PdfReader
from langchain.text_splitter import RecursiveCharacterTextSplitter
import os
from langchain_google_genai import GoogleGenerativeAIEmbeddings
import google.generativeai as genai
from langchain.vectorstores import FAISS
from langchain_google_genai import ChatGoogleGenerativeAI
from langchain.chains.question_answering import load_qa_chain
from langchain.prompts import PromptTemplate
from dotenv import load_dotenv
load_dotenv()
from google.colab import userdata
userdata.get('GOOGLE_API_KEY')


os.getenv("GOOGLE_API_KEY")
genai.configure(api_key=os.getenv("GOOGLE_API_KEY"))




def get_pdf_text(pdf_docs):
    text=""
    for pdf in pdf_docs:
        pdf_reader= PdfReader(pdf)
        for page in pdf_reader.pages:
            text+= page.extract_text()
    return  text



def get_text_chunks(text):
    text_splitter = RecursiveCharacterTextSplitter(chunk_size=10000, chunk_overlap=1000)
    chunks = text_splitter.split_text(text)
```

✓ 0s completed at 8:49 AM

```python
def get_text_chunks(text):
    text_splitter = RecursiveCharacterTextSplitter(chunk_size=10000, chunk_overlap=1000)
    chunks = text_splitter.split_text(text)
    return chunks


def get_vector_store(text_chunks):
    embeddings = GoogleGenerativeAIEmbeddings(model = "models/embedding-001")
    vector_store = FAISS.from_texts(text_chunks, embedding=embeddings)
    vector_store.save_local("faiss_index")


def get_conversational_chain():

    prompt_template = """
    Answer the question as detailed as possible from the provided context, make sure to provide all the details, if the answer is not in
    provided context just say, "answer is not available in the context", don't provide the wrong answer\n\n
    Context:\n {context}?\n
    Question: \n{question}\n

    Answer:
    """

    model = ChatGoogleGenerativeAI(model="gemini-pro",
                                   temperature=0.3)

    prompt = PromptTemplate(template = prompt_template, input_variables = ["context", "question"])
    chain = load_qa_chain(model, chain_type="stuff", prompt=prompt)

    return chain



def user_input(user_question):
    embeddings = GoogleGenerativeAIEmbeddings(model = "models/embedding-001")

    new_db = FAISS.load_local("faiss_index", embeddings)
    docs = new_db.similarity_search(user_question)

    chain = get_conversational_chain()


    response = chain(
        {"input_documents":docs, "question": user_question}
        return only outputs=True)
```

✓ 0s completed at 8:49 AM

```
[19]
def main():
    st.set_page_config("Chat PDF")
    st.header("Chat with PDF using Gemini💁")

    user_question = st.text_input("Ask a Question from the PDF Files")

    if user_question:
        user_input(user_question)

    with st.sidebar:
        st.title("Menu:")
        pdf_docs = st.file_uploader("Upload your PDF Files and Click on the Submit & Process Button", accept_multiple_files=True)
        if st.button("Submit & Process"):
            with st.spinner("Processing..."):
                raw_text = get_pdf_text(pdf_docs)
                text_chunks = get_text_chunks(raw_text)
                get_vector_store(text_chunks)
                st.success("Done")


if __name__ == "__main__":
    main()
```

```
2024-04-07 03:13:23.983
  Warning: to view this Streamlit app on a browser, run it with the following
  command:

    streamlit run /usr/local/lib/python3.10/dist-packages/colab_kernel_launcher.py [ARGUMENTS]
```

```
!streamlit run /usr/local/lib/python3.10/dist-packages/colab_kernel_launcher.py
```

```
Collecting usage statistics. To deactivate, set browser.gatherUsageStats to false.


  You can now view your Streamlit app in your browser.

  Network URL: http://172.28.0.12:8501
  External URL: http://34.42.105.229:8501

  Stopping...
^C
```

✓ 0s completed at 8:49 AM

```
[24] !pip install streamlit -q
```

```
[25] !npm install localtunnel
```

```
npm WARN saveError ENOENT: no such file or directory, open '/content/drive/MyDrive/chat_with_pdfs/package.json'
npm WARN enoent ENOENT: no such file or directory, open '/content/drive/MyDrive/chat_with_pdfs/package.json'
npm WARN chat_with_pdfs No description
npm WARN chat_with_pdfs No repository field.
npm WARN chat_with_pdfs No README data
npm WARN chat_with_pdfs No license field.

+ localtunnel@2.0.2
updated 1 package and audited 36 packages in 1.852s

3 packages are looking for funding
  run `npm fund` for details

found 2 moderate severity vulnerabilities
  run `npm audit fix` to fix them, or `npm audit` for details
```

```
[26] !streamlit run app.py &>/content/logs.txt &
```

```
[27] !wget -q -O - ipv4.icanhazip.com
```

```
34.42.105.229
```

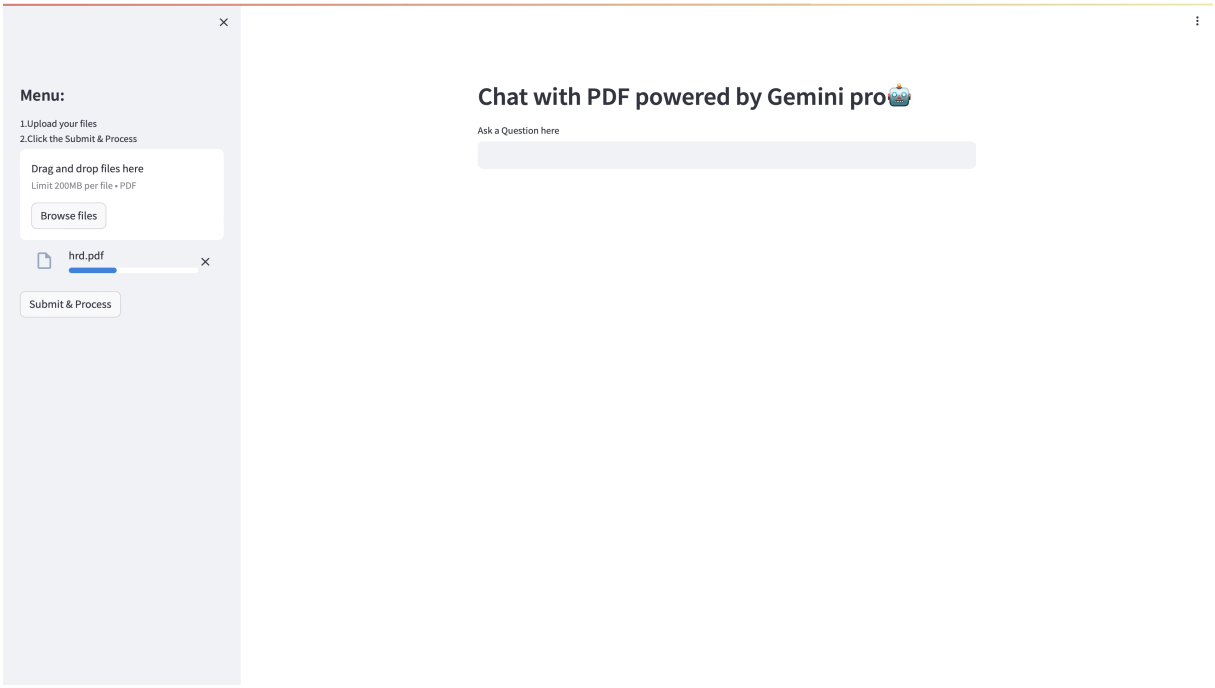
```
! streamlit run app.py & npx localtunnel --port 8501
```

```
Collecting usage statistics. To deactivate, set browser.gatherUsageStats to false.


  You can now view your Streamlit app in your browser.

  Network URL: http://172.28.0.12:8502
  External URL: http://34.42.105.229:8502

npx: installed 22 in 3.776s
your url is: https://spotty-pans-relax.loca.lt
```

✓ 0s completed at 8:49 AM

Chapter 6

Result



6.1 Future Scope

might understand what you're asking for even more accurately, making it easier to find what you need in PDFs. It could also start understanding pictures and tables in PDFs, not just text. LangChain might become a hub for all kinds of information, pulling data from different sources to give you comprehensive answers. Plus, it could remember your preferences and adapt to how you like to chat. And don't worry, it'll always prioritize fairness and privacy as it gets smarter.

Chapter 7

Conclusion

It presents an innovative and promising approach to accessing and interacting with PDF documents. By harnessing the power of natural language processing and document analysis, LangChain offers a user-friendly solution for navigating through the wealth of information contained within PDFs. Its ability to understand user queries, extract relevant information, and present it in a conversational format has the potential to revolutionize the way users engage with digital content. Looking ahead, the future of LangChain holds exciting possibilities for further enhancements and advancements. With continued development and refinement, LangChain can evolve into a versatile and indispensable tool for accessing, exploring, and synthesizing information from multiple PDF documents seamlessly. As technology progresses and user needs evolve, LangChain stands poised to play a pivotal role in facilitating intuitive and efficient information retrieval, empowering users to make informed decisions and extract insights with ease.

Chapter 8

Refernece

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$_s$*tarted*/*introduction*

3. K. Singhal et al., "Large language models encode clinical knowledge", Nature, vol. 620, no. 7972, pp. 172-180, Aug. 2023.

4. S. Kamnis, "Generative pre-trained transformers (GPT) for surface engineering", Surf. Coat. Technol., vol. 466, pp. 129680, Aug. 2023.

5. 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.

6. 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.