PDFs AI Helper

INFO 7375

Qianwei Yin 07/16/2024

Introduction

Overview

- The project involves developing **a web application** that allows users to **upload PDF documents** and subsequently **ask questions** about the content of these PDFs.

- The application leverages advanced technologies such as LangChain, OpenAI's gpt-3.5-turbo, and Pinecone for efficient document processing and retrieval-augmented generation (RAG).

Introduction

Objectives and Goals

- To create an intuitive web interface for users to upload and interact with PDF documents.
- To implement a robust backend system that can process these documents and provide accurate answers to user queries.
- To demonstrate the practical application of RAG in real-world scenarios.

Introduction

Importance and relevance of the project to the course and industry

- It combines theoretical knowledge from **file text processing** and **language models** with practical skills in web development and AI deployment.

- The ability to query documents effectively is a valuable tool in various industries, including legal, medical, and educational sectors.

Project Description

Description

- The web app allows users to upload PDF documents.
- Once uploaded, the app uses LangChain to parse the PDF content and store relevant embeddings in Pinecone.
- Users can then ask questions about the document, and the app uses OpenAI's gpt-3.5-turbo to generate answers based on the retrieved content.

Project Description

Specific problem the project aims to solve

- The project addresses the challenge of efficiently retrieving and understanding information from large PDF documents.

- This is particularly useful in contexts where manual search is impractical or time-consuming.

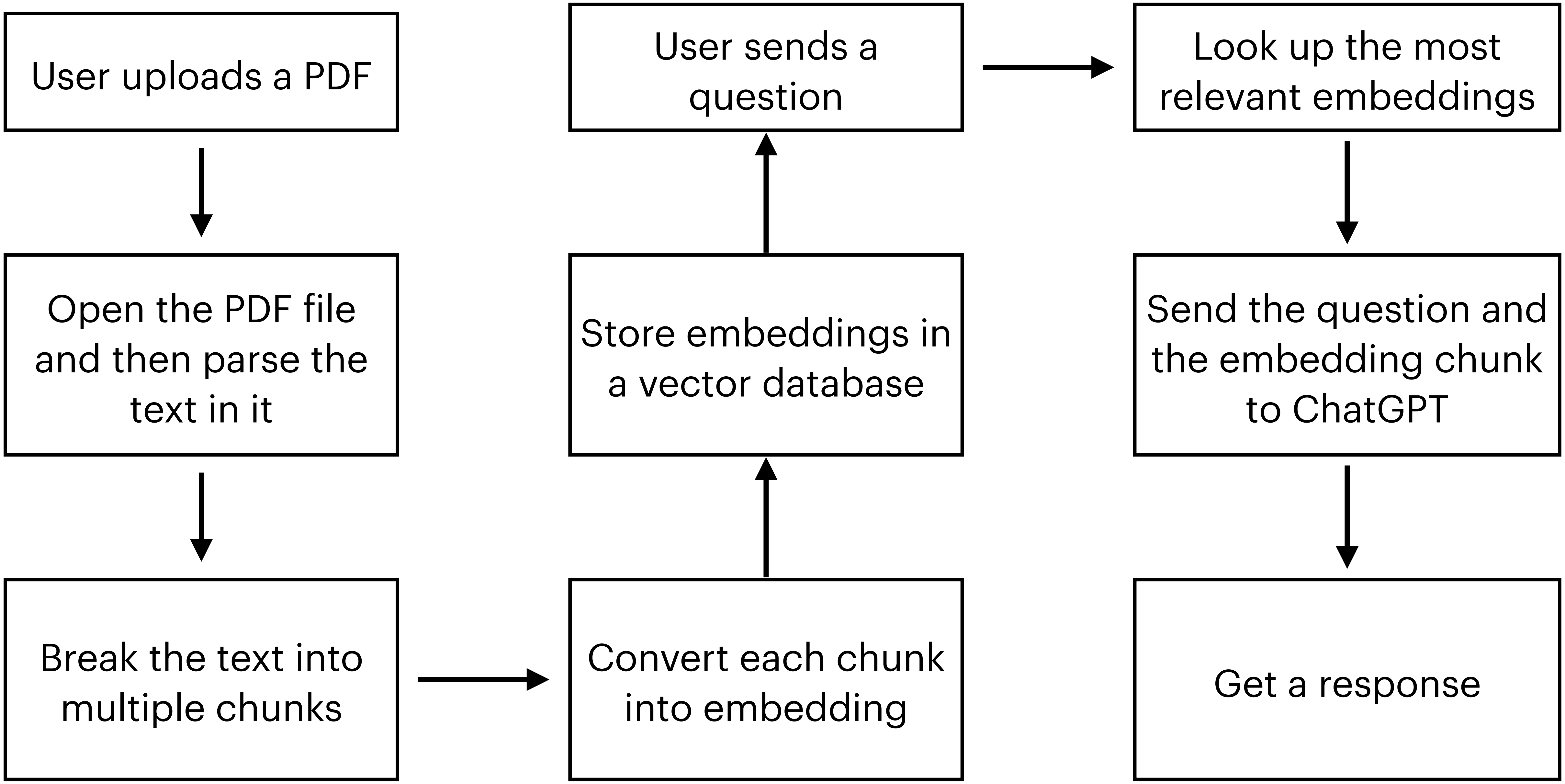
Project Description

Scope of the Project

- Frontend development for user interaction.
- Backend development for PDF processing and question answering.
- Integration of LangChain, OpenAI, and Pinecone.

Project Architecture

Diagram



User uploads a PDF

Open the PDF file and then parse the text in it

Break the text into multiple chunks

Convert each chunk into embedding

Store embeddings in a vector database

User sends a question

Look up the most relevant embeddings

Send the question and the embedding chunk to ChatGPT

Get a response

Project Architecture

Components and Their Interactions

- **Frontend (Streamlit)**: A web interface where users can upload PDFs and ask questions.
- **Backend**: Handles file uploads, PDF parsing, and interactions with AI models.
- **LangChain**: Processes the PDF content and generates embeddings.
- **OpenAI GPT-3-turbo**: Generates answers based on the retrieved content.
- **Pinecone**: Stores and retrieves embeddings for efficient content search.

Data Collection and Preprocessing

Source and nature of the data

- The data consists of user-uploaded PDF documents, which can vary in content and structure.

Data Collection and Preprocessing

Data Preprocessing Techniques and Steps

- User **uploads** PDFs via the web interface.
- PDF **parsing** to **extract** text content.
- **Embedding** generation using LangChain.
- **Stores** embeddings using Pinecone.

RAG Pipeline Implementation

Overview

- The RAG pipeline involves retrieving relevant document sections based on user queries and then generating answers using these sections.

RAG Pipeline Implementation

Steps

- User uploads PDF and submits a query.
- PDF is parsed, and text is embedded using LangChain.
- Pinecone retrieves relevant embeddings based on the query.
- GPT-4 generates an answer using the retrieved content.

RAG Pipeline Implementation

Challenges and Solutions

- Challenge: **Efficiently** processing large PDFs.
  - Possible Solution: Implementing batching and parallel processing techniques.
- Challenge: **Accurate** retrieval of relevant content.
  - Possible Solution: Fine-tuning embedding models and retrieval algorithms.

Performance Metrics

Key Metrics and Methods to Improve Metrics

- **Accuracy of answers**
  - Accuracy is measured by comparing generated answers to a set of reference answers.
- **Response time**
  - Response time is tracked from the moment a query is submitted to the answer being displayed.
- **User satisfaction**
  - User satisfaction is assessed through feedback forms and surveys.

Possible Methods to Improve Metrics

Strategies

- Implementing more advanced preprocessing techniques.
- Fine-tuning the language model with domain-specific data.
- Optimizing the retrieval algorithm for faster response times.
- Incorporating more robust error handling and fallback mechanisms.
- Enhancing the frontend for better user experience.
- Continuous model training and evaluation.

Deployment Plan

Steps

- Prepare application with necessary files.
- Create and configure a Heroku app.
- Set environment variables for API keys.
- Deploy the application using Git.
- Scale the app to run a web dyno and open it in the browser.

Deployment Plan

Plan for user testing and feedback

- Conduct initial beta testing with a small group.
- Collect user feedback via forms and surveys.
- Iterate and improve based on feedback.
- Gradually roll out to a wider audience.
- Provide ongoing support and updates.

Future Work

Potential extensions and improvements

- Support for additional document formats (e.g., Word, HTML).
- Multilingual support for non-English documents and queries.
- Advanced analytics and reporting features.

Future Work

Long-term vision and Further development

- Developing a comprehensive document management and query system.
- Integrating with other enterprise tools and platforms.
- Scaling the application for broader industry adoption.
- Expanding the dataset for better model training.
- Collaborating with industry partners for real-world applications.
- Exploring additional use cases and customization options.

Conclusion

Key Takeaways

- Practical application of RAG in document processing.
- Integration of multiple AI and web technologies.
- Importance of user-centric design and feedback.

Conclusion

Final Thoughts

- The project aims to build a web application that enables users to upload PDFs and ask questions about them, leveraging LangChain, OpenAI, and Pinecone. It addresses the need for efficient document querying and understanding.

- This project not only demonstrates the practical use of advanced AI technologies but also highlights the potential for future innovations in document management and retrieval systems. By continuing to refine and expand this application, it can become an invaluable tool for various industries.

Q&A