PDFs Question Answering Chatbot using Langchain, Llama 2

This is a Python GUI application that demonstrates how to build a custom PDF chatbot using LangChain and GPT 3.5 / Llama 2.

Important Files and Folders

1. `app/.env`: File with environment variables.
2. `app/requirements.txt`: File with dependencies to be installed for the project.
3. `app/environment.yml`: File with dependencies to create conda environment.
4. `app/`: Directory containing the source code for the streamlit app.
5. `docs/`: Documentation including meeting minutes and README in PDF format.

How it works (GPT 3.5)

1. The application GUI is built using streamlit
2. The application reads text from PDF files, splits it into chunks
3. Uses `HuggingFace Embedding Inference API` to generate embedding vectors used to find the most relevant content to a user's question
4. Build a conversational retrieval chain using Langchain
5. Use Locally installed `llama-2-7b-chat` to generate respond based on content in PDF

Requirements

1. Install the following Python packages:

```
conda env create -f environment.yml
conda activate llm
```

2. Create a `.env` file in the root directory of the project and add the following environment variables:

```
OPENAI_API_KEY=<# Your OpenAI API key>
HUGGINGFACEHUB_API_TOKEN=<# Your HUggingface hub access token>
```

> **Note**: Make sure you are in the `/app` directory

3. Download the `llama-2-7b-chat` model

```
# Create a models directory, cd into it and download the model
mkdir models
cd models
wget https://huggingface.co/TheBloke/Llama-2-7B-Chat-
```

```
GGUF/resolve/main/llama-2-7b-chat.Q3_K_S.gguf?download=true -O llama-2-7b-chat.Q3_K_S.gguf

# Move back to /app dir
cd ..
```

Code Structure

The code is structured as follows:

- `app.py`: The main application file that defines the Streamlit gui app and the user interface.
  - get_pdf_text function: reads text from PDF files
  - get_text_chunks function: splits text into chunks
  - get_vectorstore function: creates a FAISS vectorstore from text chunks and their embeddings
  - get_conversation_chain function: creates a retrieval chain from vectorstore
  - handle_userinput function: generates response from `llama-2-7b-chat`
- `htmlTemplates.py`: A module that defines HTML templates for the user interface.

How to run

```
streamlit run app.py
```

> **Note**: Make sure you are in the `/app` directory