Readme: Text - Mining - Final Project. (PDF-Bot with LLMs)

Our Streamlit application allows users to interact with PDF documents using conversational artificial intelligence (AI). Users can upload PDF files, ask questions related to the content of these documents, and receive responses generated by a pre-trained AI model.

Features:

- PDF File Upload: Users can upload one or more PDF files containing text content.
- AI Model Selection: Users can select from a variety of pre-trained AI models to generate responses to their questions.
- User Question Input: Users can input questions related to the content of the uploaded PDF files.
- Temperature Adjustment: Users can adjust the temperature parameter for response generation, controlling the creativity and randomness of the AI-generated responses.
- Response Display: The application displays the generated response, broken down into paragraphs.
- Chat Memory: Users can view a history of their interactions, including questions asked and AI-generated responses.

Code Structure:

Dependencies:

- Streamlit: Used for creating web applications.
- PyPDF2: Used for extracting text from PDF files.
- pdfplumber: Used for more robust PDF text extraction.
- pdfminer: Used for additional PDF text extraction functionality.
- Pillow: Used for image processing.
- tempfile: Used for creating temporary files.
- Hugging Face Transformers: Used for natural language processing and model loading.
- langchain: Custom library used for various text processing and conversational AI functionalities.
- Hugging Face Hub: Used for accessing and loading pre-trained models from the Hugging Face Model Hub.

Components:

- **PDF Text Extraction:** The read_pdf function extracts text content from the uploaded PDF files using various PDF processing libraries.
- **AI Model Setup:** The setup_qa_chain function sets up the conversational retrieval chain, which includes creating loaders, a vector store index, and initializing the language model (LLM).
- **Response Generation:** The generate_response function generates a response to the user's question using the configured conversational retrieval chain.
- **Streamlit App:** The main function defines the Streamlit application interface, including file upload, model selection, question input, response display, and chat memory.

AI Model Selection:

1. T5 Base:

The T5 Base model is part of the Text-To-Text Transfer Transformer (T5) architecture, capable of handling a wide range of natural language processing tasks. It excels in understanding and generating text-based responses, making it suitable for diverse conversational contexts.

2. GPT2:

GPT2, short for "Generative Pre-trained Transformer 2," is a highly versatile language model known for its ability to generate coherent and contextually relevant text. With various pre-trained versions available, users can select the GPT2 model size that best suits their needs.

3. DialoGPT Small:

DialoGPT Small, developed by Microsoft, is specifically designed for generating human-like responses in conversational settings. Trained on large-scale conversational datasets, it excels in understanding dialogue context and producing contextually appropriate responses.

4. GPT2 Medium:

GPT2 Medium is a mid-sized variant of the GPT2 model, striking a balance between model size and computational resources. It offers a good compromise between response quality and inference speed, making it suitable for real-time conversational applications.

5. GPT2 Large:

GPT2 Large is a larger version of the GPT2 model, equipped with more parameters and enhanced language generation capabilities. It excels in handling complex language tasks and generating detailed responses, with higher computational requirements.

6. GPT2 XL:

GPT2 XL represents the largest variant of the GPT2 model, designed for applications requiring extensive context understanding and long-form text generation. While offering unparalleled performance, it demands substantial computational resources for inference.

Usage:

- **Running the Application:** Execute the provided Python script to launch the Streamlit application.
- **File Upload:** Upload one or more PDF files containing text content.
- **Ask a Question:** Input a question related to the content of the uploaded PDF files.
- **Model Selection:** Choose an AI model from the available options.
- **Temperature Adjustment:** Adjust the temperature slider to control the creativity of the AI-generated responses.
- **View Response:** The application displays the generated response, broken down into paragraphs.
- **Chat Memory:** View the history of interactions, including questions asked and AI-generated responses, in the sidebar.

Note:

- Ensure that the required dependencies are installed before running the application.
- The performance of the AI model and response quality may vary depending on the selected model and the complexity of the questions asked. Adjusting the temperature parameter can also affect response creativity.

Feel free to explore and interact with the application to experience the capabilities of conversational AI in analyzing PDF documents!