PDF Chatbot with Ollama: Project Documentation

1. Executive Summary
The PDF Chatbot with Ollama is an intelligent interactive system engineered to enable users to upload PDF documents and engage with their content through natural language questioning. This solution utilizes Ollama as a locally-hosted Large Language Model (LLM) server to produce responses that are contextually relevant to the document's content. By merging advanced PDF text extraction capabilities with conversational artificial intelligence, this project strives to streamline document analysis and enhance information accessibility.

2. Goals & Core Functionality
- Deliver an intuitive and accessible user interface built with Streamlit.
- Implement robust PDF text extraction utilizing multiple libraries (PyPDF2, pdfplumber) for reliability.
- Facilitate natural language questioning against the document's content.
- Integrate seamlessly with the local Ollama instance for LLM-powered inference.
- Provide user-adjustable model parameters such as temperature and top-p for response customization.
- Enable export functionality for both chat history and extracted text in JSON and TXT formats.
- Incorporate robust error handling for connection issues and request timeouts.

3. Technical Prerequisites- **Supported OS:** Windows 10/11 or a Linux distribution running Python 3.9 or newer.
- **Python Libraries:** Streamlit, Requests, PyPDF2, pdfplumber, ReportLab.
- **Ollama:** Must be installed and actively running (via `ollama serve` command).
- **Hardware:** Minimum of 8 GB RAM (higher capacity recommended for larger LLM models).
- **Client:** A modern web browser like Chrome, Edge, or Firefox to access the web interface.

4. System Design & Operational Flow**
The chatbot architecture is composed of several key modules:

PDF Chatbot with Ollama: Project Documentation

1. **User Interface (UI):** A Streamlit-based web application handling PDF uploads, chat interaction, and export features.
2. **PDF Processing Engine:** Employs PyPDF2 and pdfplumber to convert uploaded PDFs into searchable, structured text.
3. **Conversation Manager:** Constructs prompts and communicates with the Ollama server via its REST API, sending the user's query along with the relevant PDF context.
4. **Ollama LLM Server:** Operates locally, processing incoming requests and generating intelligent, context-aware replies.
5. **Output Handler:** Manages the display of responses, maintains chat history, and facilitates data export.

The step-by-step user interaction is as follows:
- The user uploads a PDF document through the UI.
- The system extracts the text content from each page of the PDF.
- The user submits a question related to the document in the chat interface.
- The application sends the question and the relevant text context to the Ollama server.
- Ollama processes the request and streams the generated answer back to the UI.
- The conversation is updated in real-time, and the full history is available for export.

5. User Instructions
1. Ensure the Ollama server is running by executing `ollama serve` in your terminal.
2. Launch the Streamlit application using the command `streamlit run app.py`.
3. Navigate to the provided local URL and upload your target PDF file using the interface.
4. Begin interacting by typing questions about the document's content into the chat input box.
5. Utilize the sidebar to adjust model parameters like temperature and top-p to fine-tune responses.
6. Use the provided buttons to clear the current conversation or reset the session.

7. Download the extracted text or the complete chat history in JSON/TXT format for record-keeping.

6. Planned Future Developments
- Support for uploading and querying across multiple documents simultaneously.
- Implementation of GPU acceleration to significantly improve inference speed.
- Introduction of additional features like automatic summarization and keyword extraction.
- Integration of cloud-based LLM APIs (e.g., OpenAI) as an optional or fallback service.
- Enhancement of PDF processing to handle scanned documents and images through Optical Character Recognition (OCR) technology.

7. Concluding Remarks
The PDF Chatbot with Ollama effectively illustrates the practical synergy between local LLM inference and sophisticated document processing, resulting in a potent tool for knowledge retrieval. This system not only simplifies access to information locked within PDFs but also establishes a scalable foundation for developing more advanced AI-assisted applications in academic, research, and corporate environments. Subsequent development will focus on expanding its functional scope and optimizing performance.