Local LLM-Powered PDF Q&A System

This project provides an **offline**, **private**, and **secure** way to interact with PDF documents using a **local Large Language Model (LLM)**. With PDF parsing, smart chunking, embedding via `sentence-transformers`, and semantic search powered by **FAISS**, you can query any uploaded PDF — entirely on your own machine.

Table of Contents

Features

- **Offline Operation**: All tasks are performed locally — no cloud needed.
- **Privacy-Focused**: No API calls. Your data stays on your device.
- **PDF Text Extraction**: Extracts text from standard PDFs.
- **Smart Text Chunking**: For better context preservation.
- **Semantic Search**: Powered by FAISS for fast and relevant chunk retrieval.
- **Local LLM Integration**: Uses `llama-cpp-python` to run LLMs offline.
- **Streamlit UI**: Intuitive and interactive browser-based interface.
- **Session Management**: Retains chat history and PDF state.
- **Configurable**: Tune parameters like chunk size, overlap, top-k chunks, and model path.

Prerequisites

Make sure your system meets these requirements:

- **Python** 3.9 or higher – Download Python
- **Git** – Download Git
- **5–10 GB Disk Space** – For model files and processed data
- **8 GB+ RAM Recommended** – For 7B quantized models (e.g., Mistral, Llama 2)

Installation Guide

Step 1: Clone the Repository

```
git clone https://github.com/your-username/your-repo-name.git
cd your-repo-name
```

Step 2: Create and Activate a Virtual Environment

🔹 Create a Virtual Environment

```
python -m venv venv
.\venv\Scripts\activate  #For Windows
```

Step 3: Install Dependencies

🔹 **Create `requirements.txt`**

Add the following content to a file named `requirements.txt`:

```
streamlit
PyMuPDF
langchain
langchain-community
sentence-transformers
faiss-cpu
llama-cpp-python
numpy
```

Step 4: Download the Local LLM Model

🔹 **Download a `.gguf` model file**

You can download a model like **Mistral** or **LLaMA 2 7B** from:

🔗 [TheBloke's Hugging Face Models]

🔹 **Recommended file:**

After downloading, move the file into the `models/` directory you just created:

```
mkdir models
# Move the downloaded file here
mv mistral-7b-instruct-v0.2.Q4_K_M.gguf models/
```

🔹 Update llm_utils.py with the model filename and path

```
LLM_MODEL_FILENAME = "mistral-7b-instruct-v0.2.Q4_K_M.gguf"
LLM_MODEL_PATH = os.path.join(os.path.dirname(__file__), "..", "models",
LLM_MODEL_FILENAME)
```

🔹 Project Structure

```
your-project-name/
├── app.py # Main Streamlit app
├── pdf_utils.py # PDF text extractor
├── embed_utils.py # Chunking and embedding logic
```

```
├── vector_store.py # FAISS-based vector DB
├── llm_utils.py # LLM loading & inference
├── requirements.txt # Dependency file
├── venv/ # Python virtual environment
├── data/ # Processed text/PDF data
└── models/ # LLM model (.gguf)
```

How to Use

Activate the environment:

```
# On Windows:
.\venv\Scripts\activate

# On macOS/Linux:
source venv/bin/activate
```

Run the app:

```
streamlit run app.py
```

Then open your browser to: http://localhost:8501

Upload a PDF

- Click "Choose a PDF file" in the app.
- Wait for processing (text extraction, chunking, embedding).
- Ask your question in the input box and get instant answers!

Configuration Options

Setting	Location	Description
`chunk_size`	`embed_utils.py`	Max characters per text chunk
`chunk_overlap`	`embed_utils.py`	Overlap (in chars) between text chunks
`k` (top-k chunks)	`app.py`, `vector_store.py`	Number of most relevant chunks retrieved
`LLM_MODEL_FILENAME`	`llm_utils.py`	Name of your `.gguf` local LLM model file

Troubleshooting Common Issues

llama-cpp-python Build Fails

Windows: Install Visual Studio Build Tools Choose the "Desktop development with C++" workload during installation.

macOS/Linux: Make sure you have gcc, clang, or xcode installed.

Model Not Found Error: LLM model not found at...

Make sure your .gguf file is placed inside the models/ folder.

Ensure the LLM_MODEL_FILENAME in llm_utils.py exactly matches the file name.

No Text Extracted from PDF Cause: The PDF may be image-based (e.g., scanned).

Solution: Use OCR to convert it into a searchable PDF.

Tools: Tesseract OCR, Adobe Acrobat.

 CUDA Out of Memory Use a smaller quantized model (e.g., Q4, Q2).

Reduce chunk_size and k from the Streamlit sidebar.

Close other GPU-intensive apps.

 Streamlit Won’t Start Make sure the virtual environment is activated.

```
Try reinstalling dependencies:
pip install -r requirements.txt
```
