Title: Code Genei AI

Name: Ranjan Kumar

Milestone 3: OCR Chatbot

Github:Click here

This documentation provides a guide explains how to set up and use the OCR Chatbot. It uses Streamlit for the interface, PyTesseract for reading text from images, and Ollama for chatting about the extracted content.

⚙️ Prerequisites

Before you begin, ensure you have the following installed on your system:

- **Python 3.7 or higher**: This is the core language for the application.
- **Tesseract OCR Engine**: This is a powerful OCR engine required for text extraction from images.
    - **Windows**: Download and install the Tesseract installer from https://github.com/UB-Mannheim/tesseract/wiki. Note the installation path, as you'll need it in the code.
- **Ollama**: This tool runs large language models (LLMs) locally. Download and install it from the https://ollama.com/ .
- **A local LLM**: After installing Ollama, pull a model to use with the application. The code uses **llama2**. Open your terminal and run: **ollama pull llama2**.

🚀 Installation and Setup

Follow these simple steps to get the application running.

1. Install Python Libraries

Open your terminal or command prompt and install the necessary Python packages using pip:

```
pip install streamlit pillow pytesseract ollama
```

- **streamlit:** The framework for creating the web application.
- **pillow:** A library for handling image files (part of the Python Imaging Library).
- **pytesseract:** The Python wrapper for the Tesseract OCR engine.
- **Ollama :** The Python client for communicating with the Ollama server.

2. Configure PyTesseract

The Python code needs to know where your Tesseract installation is located.

- Must update the path in the script:

  `pytesseract.pytesseract.tesseract_cmd = r"C:\Program Files\Tesseract-OCR\tesseract.exe"`

  Change the path to match your installation directory if it's different.

3. Run the Application

Navigate to the directory containing your Python script and run the following command in your terminal:

`streamlit run Ocr.py`

This will start a local web server and open the application in your default web browser .

🖼️ Usage Guide

The OCR Chatbot's user interface is split into two main sections: the sidebar and the main content area.

Sidebar: Session Management

The sidebar allows you to manage different chat sessions.

- **Create Session**: Enter a name in the text box and click "Create Session" to start a new, isolated conversation. This is useful for working on different images or topics without mixing the chat history.
- **Clear Current Session**: The "Clear Current Session" button will erase all messages and extracted text from the currently active session, but the session itself will remain.

Main Content: Image & Chat

The main area of the application is where you interact with the OCR and chatbot features.

- **Upload Image**: Use the file uploader to select a JPG or PNG file. Once uploaded, the application will display the image and automatically run **OCR** on it.

    - The extracted text will appear in the "Extracted Text" box.
- **Extracted Text (Editable)**: The OCR text appears in an editable area. You can fix errors or adjust the text, which the chatbot then uses as its knowledge base.
- **Chat Input**: Use the "Ask something..." box to chat. Ask questions, get summaries, or request details, and the chatbot will answer using the extracted text.
    - **Example questions**:
        - "What is the total annual income mentioned in the document?"
        - "Who is the certificate holder?"

The chat history will be displayed above the input box, showing both your questions and the chatbot's responses.

Approach

The OCR Chatbot is a user-friendly app that lets you upload images, extract text, and chat about it using a local LLM. It runs on Streamlit for an easy web interface.

Methodology

The application's functionality is built on a few key components that work together in a streamlined workflow:

- **Image Upload and Text Extraction:** The user uploads a JPG or PNG file through the Streamlit interface. Once the image is uploaded, the application uses **PyTesseract**, a Python wrapper for the **Tesseract OCR Engine**, to automatically extract all the text from the image.
- **Editable Extracted Text:**The extracted text appears in an editable box, so users can fix any OCR errors. This corrected text becomes the chatbot's knowledge base.
- **Chatbot Interaction:** The chatbot uses Ollama with the llama2 model to process questions. It combines the chat history and extracted text to give answers based only on the document.

Outcomes

The OCR Chatbot successfully delivers a unified solution for document analysis and interaction. Users can:

- **Accurately extract text from images**, which can be easily corrected for greater precision.
- **Ask questions** about the document's content, such as summarizing key details or finding specific information.
- **Run the entire process locally**, as both the OCR engine and the LLM are installed on the user's system. This ensures data privacy and allows for offline use