Documentation for Resume Parsing and Generative AI Integration

Project Overview

This project is aimed at automating the extraction of key details from resumes using Python. The extracted information is then structured into a CSV file, making it easier for further analysis or decision-making. The core of the project relies on Generative AI, specifically Google's Gemini model, for text extraction, and the data is processed in batches for efficiency. The program reads resumes stored in PDF format, extracts relevant details, and organizes them in a structured format using predefined headers.

Approach

1. **Data Extraction:**
   - Resumes are in PDF format and stored in a specific directory on the system.
   - The `PyPDF2` library is used to read the contents of each resume PDF file and extract text.
2. **Prompt Generation:**
   - A prompt is dynamically constructed for each resume using the extracted text and predefined headers. This prompt is then sent to the Generative AI model to parse and structure the information.
   - The headers for the parsed data include essential details such as:
     - Name
     - Contact details (phone number)
     - University
     - Year of Study
     - Course
     - Discipline
     - CGPA/Percentage
     - Key Skills
     - Generative AI Experience Score
     - AI/ML Experience Score
     - Supporting Information (certifications, internships, projects)
3. **Generative AI Model:**
   - Google's Gemini-1.5-Flash model is used for text understanding and response generation.
   - The prompt is processed by the Generative AI to extract the specified headers and output the data in JSON format.
4. **Batch Processing for Efficiency:**
   - To handle multiple resumes at once, a batch processing approach using `ThreadPoolExecutor` from the `concurrent.futures` module is implemented.
   - This allows the system to process multiple resumes concurrently, improving performance and speeding up the output generation.
5. **Data Formatting and Storage:**

- The JSON responses are converted into a Pandas DataFrame for easy manipulation.
- The structured data is saved into an Excel file (`resumes_output.xlsx`), which contains all the extracted details from the resumes.

Generative AI Features and Innovations

- **Generative AI Integration:**
  - The core innovation lies in using Generative AI (specifically the Gemini model) for automatic extraction of structured information from unstructured resume text. The AI is prompted with a natural language description of the desired fields, allowing it to generate a well-structured JSON output.
- **Dynamic Prompt Construction:**
  - The prompt is dynamically constructed for each resume based on the extracted text, allowing for context-specific extraction without requiring predefined templates. This is a flexible and scalable approach, making the model adaptable to different types of resumes.
- **Batch Processing with Thread Pool:**
  - By using `ThreadPoolExecutor`, multiple resumes can be processed simultaneously. This feature improves performance significantly, especially when handling large volumes of resumes, enabling faster parsing of data.

Innovative Features Implemented

1. **Automatic Resume Parsing:**
   - The program handles a batch of resumes and extracts structured information from them using Generative AI. This automation reduces the manual effort involved in resume screening.
2. **Generative AI Customization:**
   - The Gemini-1.5-Flash model is specifically chosen for its ability to generate detailed and structured outputs, ensuring accurate extraction of the required resume information.
3. **Multi-threaded Processing:**
   - The batch processing with multiple threads speeds up the extraction process, handling multiple resumes at once without significantly slowing down the system.
4. **Output in CSV Format:**
   - The parsed data is stored in a user-friendly CSV format, which can be easily analyzed or imported into other applications, making the system highly efficient for HR professionals or recruiters.

Challenges and Solutions

1. **Text Extraction from PDFs:**
    - Extracting clean and accurate text from PDFs can sometimes be difficult due to the quality and format of the resume. The solution was to use `PyPDF2`, a robust library for text extraction, and handle any parsing issues programmatically by cleaning up the text before sending it to the AI model.
2. **Batch Processing and Performance:**
    - Processing resumes one by one can be time-consuming, especially for a large dataset. By implementing batch processing using `ThreadPoolExecutor`, the project ensures that multiple resumes are processed concurrently, which greatly improves the performance.

Conclusion

This project successfully demonstrates how Generative AI can be leveraged to automate the extraction of key details from resumes, significantly improving the efficiency of resume processing workflows. The combination of AI-powered text extraction, dynamic prompt construction, and multi-threaded processing allows for scalable and adaptable solutions to resume screening. By saving the structured data into a CSV format, the project enables further data analysis and decision-making.