NAME: BHAVYA TANEJA

MILESTONE-4 REPORT

Objective

The system provides an automated solution to analyze financial and document data, extract relevant information, and generate visual insights using llm model. It focuses on handling three document types: payslips, balance sheets, and bank statements.

Technologies and Libraries Used

1. **Gradio**:
   - Purpose: Provides a user-friendly interface to interact with the system.
   - Usage: Designed the front-end for the user to upload inputs, visualize results, and retrieve outputs seamlessly.
   - Features:
     - Dropdown menus for document type selection.
     - Options for image count and visualization type (Bar Chart or Pie Chart).
     - Clear button for resetting the system.
     - Display of extracted text, comparison, and visualizations.
2. **EasyOCR** :
   - Purpose: Optical Character Recognition (OCR) for text extraction from images.
   - Features: Supports multiple languages (here, en for English).
   - Usage:
     - Detects and extracts text from images after preprocessing.
     - Maps text to predefined categories such as "Basic Salary," "HRA," etc.
3. **Cohere**:
   - Purpose: NLP capabilities for advanced text analysis.

   - Configuration: Initialized with the Cohere API key.
   - Usage: Placeholder for potential language-based enrichment tasks.

4. **Cloudinary**:
   - Purpose: Handles cloud storage and retrieval of document images.
   - Usage:
     - Retrieves image URLs using a prefix-based search.
     - Securely downloads and processes images for analysis.
5. **Matplotlib**:
   - Purpose: Visualization of extracted data.
   - Features:
     - Generates bar and pie charts based on extracted values.
     - Enhances user understanding of financial metrics.
6. **OpenCV** :
   - Purpose: Image preprocessing for OCR enhancement.
   - Usage:
     - Applied grayscale conversion, noise reduction, and adaptive thresholding.
     - Improves OCR accuracy for low-quality documents.
7. **Cloudinary API** :
   - Purpose: Secure interaction with cloud-hosted resources.
   - Usage:
     - Queries cloud storage to retrieve financial document images by prefix.
8. **Pandas** :
   - Purpose: Tabular representation of extracted data.
   - Features:
     - Organizes the data into a user-friendly format.
     - Supports creation of comparison tables.

9. **FuzzyWuzzy** :
    - Purpose: Text similarity checking using fuzzy matching.
    - Features:
        - Matches text in the document to predefined field names with variations.
        - Configurable similarity threshold (default: 70%).
10. **Tempfile**:
    - Purpose: Temporary file management during processing.
    - Usage:
        - Stores intermediate files like processed images and visualizations.
11. **OS**:
    - Purpose: Environment variable management and file operations.
    - Usage:
        - Accesses API keys and file paths.
12. **Requests**:
    - Purpose: HTTP requests for downloading images from URLs.
    - Features: Retrieves images and saves them temporarily for OCR.
13. **Re** :
    - Purpose: Regular expressions for data validation.
    - Usage:Ensures extracted values are numeric or formatted correctly.

Key Features and Functionalities

1. **Image Retrieval and Preprocessing**:
    - Images are fetched from Cloudinary based on a user-defined prefix.
    - Enhanced preprocessing ensures high OCR accuracy:
        - Noise reduction via cv2.fastNlMeansDenoising.
        - Adaptive thresholding for contrast improvement.

2. **Text Extraction and Mapping**:
   - EasyOCR extracts text from processed images.
   - Fuzzy matching associates extracted text with predefined financial terms.
3. **Visualization**:
   - **Bar Chart**: Displays comparisons of metrics across multiple images.
   - **Pie Chart**: Highlights proportional data distribution.
4. **Data Validation and Parsing**:
   - Ensures extracted values are numeric and properly formatted.
   - Avoids errors during visualization and statistical computation.
5. **Comparison Analysis**:
   - Highlights the highest and lowest values for each financial category across images.
   - Provides users with actionable insights.
6. **Interactive User Interface**:
   - Clear, user-friendly Gradio interface with real-time feedback.

Code Workflow

1. **Input Handling**:
   - User selects the document type, the number of images, and the chart type.
   - System fetches matching images from Cloudinary.
2. **Image Processing**:
   - Images are downloaded and preprocessed.
   - Text is extracted using EasyOCR.
3. **Data Mapping**:
   - Extracted text is mapped to categories based on similarity scores.
4. **Analysis and Visualization**:
   - Data is structured into a Pandas DataFrame.

   - Visualizations are generated using Matplotlib.

5. **Output Display**:
   - Displays retrieved images, extracted data, visualizations, and comparison metrics.

6. **Reset Functionality**:
   - Clears all inputs and outputs for a fresh start.

System Configuration and Dependencies

- **Python Version**: Compatible with Python 3.7 and above.
- **Package Installation**:
   - pip install gradio easyocr cohere matplotlib opencv-python-headless pandas fuzzywuzzy cloudinary requests
- **Environment Variables**:
   - COHERE_API_KEY: API key for Cohere.
   - Cloudinary keys (cloud_name, api_key, api_secret).