# User Guide for Python Script: GPT\_v3\_mee

## 1. Introduction

The data\_processor\_tool Python script extracts data from PDF files within a specified directory, processes the data using the OpenAI API, and saves the output in a CSV format. It is designed for users who need to extract specific information from SEC Form 8-K filings and save it in a structured format.

## 2. Requirements

Before running the script, ensure you have the following:

- Python (version 3.6 or higher)  
- The following Python libraries:  
 - openai  
 - os  
 - PyPDF2  
 - pandas  
 - csv  
 - shutil

You can install missing libraries using pip:

pip install openai PyPDF2 pandas

## 3. Setting Up the API Key

1. Obtain an OpenAI API key from OpenAI's website (https://platform.openai.com/).  
2. In the script, locate the line where openai.api\_key is set and replace the placeholder with your actual API key.  
3. Important: Keep your API key secure and do not share it publicly.

## 4. Directory Structure and Script Location

Place the script in a directory containing subfolders with PDF files. The script will automatically process all PDFs within the specified base directory and its subdirectories.  
  
Base Directory: The folder where the script is saved.  
Extracted Folder: After processing, the script will move each processed folder to a new directory named 'Extracted' within the base directory.

## 5. Running the Script

1. Open Terminal or Command Prompt:

Navigate to the directory where the script is saved using:  
cd path/to/script-directory

2. Run the Script:

Execute the script by typing:  
python script\_name.py

## 6. Script Workflow and Explanation

This section provides a high-level overview of each part of the script:  
  
Setting up the Base Directory: The script sets the base\_directory to the directory where the script is saved. This allows for relative file paths without hardcoding.

## 7. Explanation of the Main Processing Steps

- PDF Reading and Text Extraction:  
Each PDF file in the specified folders is read, and its content is extracted using PyPDF2.  
  
- Data Processing with OpenAI API:  
The script uses the OpenAI API to process extracted text and retrieve relevant information (e.g., Name, Effective Date, Role, and Change).  
  
- CSV Output:  
Each folder’s results are saved as a CSV file with headers ('Name', 'Effective Date', 'Role', 'Change'). The CSV file is saved in the 'Extracted' folder within the base directory.

## 8. Sample Output

The output CSV file will have the following structure:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Effective Date | Role | Change |
| Xxxxxxxxxxxx | 09/06/2015 | Chief Financial Officer | intends to retire |
| yyyyyyyyyyyy | 09/06/2015 | Class II Director | elected as director |
| zzzzzzzzzzzzz | 09/06/2015 | Class II Director | elected as director |

## 9. Troubleshooting

- If you encounter missing library errors: Ensure all required libraries are installed.  
- API Errors: Verify that your OpenAI API key is valid and correctly set in the script.  
- File Not Found Errors: Ensure that PDF files are located in the correct folders within the base directory.

## 10. Checking the OpenAI Library Version

To verify the OpenAI library version, run the following code in Python:

import openai  
print(openai.\_\_version\_\_)

This will display the version of the openai package, which should be compatible with the script requirements.

## 11. Modifying the Script for Customization

If you want to customize the script to handle different output formats or additional text extraction, consider editing the extract\_data function, specifically the parts that interact with the OpenAI API and process the CSV output.

## 12. Appendix

Python Libraries Used:  
- openai: For calling the OpenAI API to process extracted text.  
- os, shutil: For file and folder management.  
- PyPDF2: For reading PDF files.  
- pandas: For data manipulation and saving CSV files.