

Mini AI Email Summarizer & Task Logger

Overview

This project automates summarizing customer support emails and logging extracted insights as tasks into a Google Sheets tracker.

The system uses AI to:

- Summarize incoming email content.
- Extract key details such as customer name, urgency, and topic.
- Automatically append these details as a new row in a Google Sheet task tracker.

This saves time for support teams by streamlining email processing and task creation.

Features

- Input: Email-style message text.
- AI processing: Uses OpenAI's GPT-3.5 Turbo model via the ChatCompletion API.
- Extracts summary, customer name, urgency level, and main topic.
- Logs extracted data to Google Sheets via Google Sheets API. □ Easily testable with example email included.

Tools & Technologies

- **Python 3.7+**
- **OpenAI API** (`gpt-3.5-turbo`)
- **Google Sheets API** with service account authentication □ Libraries:
 - `openai` (OpenAI Python SDK)
 - `google-auth` and `google-api-python-client` (Google APIs)
 - `re`, `json`, `os` (standard Python libraries)

Setup Instructions

1. Prepare Google Cloud Project & Service Account

1. Go to Google Cloud Console.
2. Create a new project or select an existing one.

3. Enable the **Google Sheets API** for your project.
4. Go to **IAM & Admin > Service Accounts**.
5. Create a new Service Account.
6. In the Service Account details, go to the **Keys** tab.
7. Click **Add Key > Create new key**, select **JSON**, and download the JSON key file.
8. Share your Google Sheet with the service account's email address (found in the JSON file under "`client_email`"), granting Editor access.

2. Create Your Google Sheet

- Create a new Google Sheet.
- Copy the **Sheet ID** from the URL (the long string between `/d/` and `/edit`). □
Share it with your service account email as described above.

Example Sheet URL:

`https://docs.google.com/spreadsheets/d/1aeYq7pS3vMRvb3FJiMmpmlOhOfsU7UUY9SB_bJ8N4do/edit#gid=0` Sheet

ID:

`1aeYq7pS3vMRvb3FJiMmpmlOhOfsU7UUY9SB_bJ8N4do`

3. Set up Your Python Environment

- Install Python 3.7 or higher (recommended 3.8+).
- (Optional but recommended) Create a virtual environment:

```
bash CopyEdit
python -m venv
env
source env/bin/activate      # Linux/macOS

.\env\Scripts\activate      # Windows PowerShell □
```

Install required packages:

```
bash
CopyEdit
pip install --upgrade openai google-auth google-api-python-client
```

4. Prepare Your Project Folder

- Place your downloaded Google service account JSON file in your project folder.
- Create your Python script (`email_summarizer.py`) and paste the provided code (see below).
- Replace placeholders in the script:

- `SPREADSHEET_ID` with your Google Sheet ID. ◦
- `openai.api_key` with your actual OpenAI API key.

5. Running the Script

Run the script via terminal or your IDE:

```
bash    CopyEdit    python
email_summarizer.py
```

Expected output:

- The email is processed.
- Extracted data printed.
- Data logged as a new row in the Google Sheet.

Example Email Input

```
plaintext CopyEdit
Hello,
```

My name is Jane Doe. I'm having a serious issue with my account login, and I can't access it since yesterday. Please help as soon as possible! This is urgent. Thank you, Jane

Troubleshooting

- **No JSON service account file found:** Make sure your service account key JSON file is in the project directory.
- **Google Sheets permission errors:** Confirm your Google Sheet is shared with your service account email.
- **OpenAI API errors:** Check your API key is correct and has quota available. □
- **Python version errors:** Use Python 3.7+.

Optional Enhancements

- Add a simple UI (e.g., Flask web app) to input emails.
- Integrate with Zapier or n8n for automation triggers.
- Extend extraction fields or improve AI prompt.

RESULTS

The screenshot displays the Spyder Python IDE interface. The main editor window shows a Python script named `email_summarizer.py` with the following content:

```
1 import os
2 import json
3 import re
4 import openai
5 from google.oauth2 import service_account
6 from googleapiclient.discovery import build
7
8 # --- CONFIG SECTION ---
9
10 # Path to your Google service account file
11 SERVICE_ACCOUNT_FILE = "C:/Users/L3/Downloads/service_account.json"
12
13 # Your Google Sheet ID
14 SPREADSHEET_ID = "1oeYq7pS3vMRvb3FJ"
15
16 # OpenAI API Key loaded from environment variable
17 openai.api_key = os.getenv("OPENAI_API_KEY")
18 if not openai.api_key:
19     raise ValueError("OpenAI API key not found")
20
21 # --- Google Sheets Authentication ---
22 SCOPES = ['https://www.googleapis.com/auth/spreadsheets']
23 credentials = service_account.Credentials.from_service_account_file(
24     SERVICE_ACCOUNT_FILE, scopes=SCOPES)
25 service = build('sheets', 'v4', credentials=credentials)
26 sheet = service.spreadsheets()
27
28 # --- OpenAI Call ---
29 def ai_process_email(email_text):
30     prompt = f"""
31     You are a helpful assistant extracting info from customer support emails.
32     Extract the following from the email text:
33     """
34     response = openai.chat.completions.create(
35         model="gpt-4o",
36         messages=[{"role": "user", "content": prompt + email_text}],
37     )
38     return response.choices[0].message.content
```

A text editor window titled "Text editor - test_email" is open in the foreground, displaying the following text:

```
Hello,
My name is Jane Doe. I'm having a serious issue with my account login, and I can't
access it since yesterday.
Please help as soon as possible! This is urgent.
Thank you,
Jane
```

The background IDE shows the variable explorer on the right with the following data:

Name	Type	Size	Value
credentials	oauth2.service_account.Credentials	1	Credentials object of google.oauth2.service_account module
service	discovery.Resource	1	Resource object of googleapiclient.discovery module
sheet	discovery.Resource	1	Resource object of googleapiclient.discovery module

The status bar at the bottom indicates: LSP Python: ready | conda: base (Python 3.9.13) | Line 132, Col 1 | UTF-8 | CRLF | RW | Mem 64% | 1:45 PM