This code provides a Python implementation for summarizing an assessment consultation using Azure OpenAI's language model. The code includes a wrapper class for interacting with the Azure OpenAI API, promoting clean and maintainable code, and a function to generate a structured SOAP (Subjective, Objective, Assessment, Plan) note from a given conversation transcription.

**Dependencies**

1. os: For accessing environment variables.
2. json: For handling JSON files.
3. dotenv: For loading environment variables from a .env file.
4. openai: Azure OpenAI SDK for interacting with Azure OpenAI services.
5. time: For measuring the response time of the API calls.

**AzureOpenAIWrapper Class**

The AzureOpenAIWrapper class is a wrapper for interacting with the Azure OpenAI API. It abstracts the configuration and interaction details, providing a clean and straightforward interface for performing chat completions.

**Initialization**

def \_\_init\_\_(self):

"""

Loads environment variables for API configuration securely.

"""

load\_dotenv()

self.api\_key = os.getenv("API\_KEY")

self.api\_version = os.getenv("API\_VERSION")

self.azure\_endpoint = os.getenv("AZURE\_ENDPOINT")

self.azure\_model\_deployment = os.getenv("AZURE\_MODEL\_DEPLOYMENT")

The \_\_init\_\_ method loads environment variables needed for the API configuration using the dotenv library, ensuring sensitive information like the API key is handled securely.

**Chat Completion**

def create\_chat\_completion(self, prompt, max\_tokens=3000, temperature=0.2, top\_p=0.95):

"""

Performs chat completion using the configured Azure OpenAI model.

Args:

prompt (str): The prompt text for the conversation construction.

max\_tokens (int, optional): The maximum number of tokens allowed in the response. Defaults to 3000.

temperature (float, optional): Controls the randomness of the generated text. Defaults to 0.2.

top\_p (float, optional): The probability of picking the top words in the vocabulary. Defaults to 0.95.

Returns:

str: The generated conversation reconstruction.

"""

client = AzureOpenAI(api\_key=self.api\_key,

api\_version=self.api\_version,

azure\_endpoint=self.azure\_endpoint)

start\_time = time()

response = client.chat.completions.create(

model=self.azure\_model\_deployment,

max\_tokens=max\_tokens,

temperature=temperature,

top\_p=top\_p,

messages=[

{"role": "system", "content": "Assistant is a conversation constructor between the doctor and patient."},

{"role": "user", "content": f"{prompt}"}

]

)

total\_time = time() - start\_time

hours, remainder = divmod(total\_time, 3600)

minutes, seconds = divmod(remainder, 60)

print(f"The time difference is: {int(hours)} hours, {int(minutes)} minutes, {int(seconds)} seconds")

print(f"The number of tokens being used are {response.usage.total\_tokens}")

return response.choices[0].message.content

The create\_chat\_completion method uses the Azure OpenAI client to generate a chat completion based on the given prompt. It measures and prints the response time and the number of tokens used.

**summarize\_text Function**

The summarize\_text function processes a JSON transcription of a conversation and generates a structured SOAP note using the Azure OpenAI model.

**Parameters**

* json\_transcription: A JSON object containing the conversation to be summarized.

**Workflow**

1. **Extract Text**: Extract the conversation text from the JSON transcription.
2. **Construct Prompt**: Build a detailed prompt instructing the model to generate a SOAP note.
3. **Generate Summary**: Use the AzureOpenAIWrapper to generate the summary.
4. **Return Summary**: Return the generated SOAP note as a dictionary.

def summarize\_text(json\_transcription):

text = json\_transcription["conversation"]

prompt = (

f"""

You are an AI assistant to a doctor. Your task is to interpret the spoken conversation {text}

between the doctor and a patient during a "geriatric assessment" consultation.

Your task is to consolidate all the available health information regarding the patient (including any available past

consultations, chatbot interactions, reports uploaded, Electronic Health records available etc)

with the information gathered during the latest consultation into one comprehensive view of the patient health.

Assimilate all the information collected so far into a detailed SOAP note that any clinician can use to quickly

understand the patient’s history and the current conversation.

Include the following sections in the detailed SOAP note:

Subjective: In succinct bullet points, provide the following, if available:

List of current symptoms reported by the patient (e.g., "Cough", "Fatigue", "Nausea").

Timeline of medical history (e.g., "Hypertension diagnosed 5 years ago").

Family history of relevant conditions (e.g., "Father with diabetes").

Social history including smoking, alcohol use, and occupational hazards (e.g., "Non-smoker", "Social drinker", "Works in a construction environment").

Objective: In succinct bullet points, include objective details as under, if available:

Vital signs (e.g., "Blood pressure: 140/90 mmHg", "Heart rate: 88 bpm", "Temperature: 37.5°C").

Physical exam findings using standardized medical terminology (e.g., "Normocephalic, atraumatic", "Lungs clear to auscultation bilaterally").

Results of any diagnostic tests mentioned during the consultation (e.g., "Urinalysis negative for leukocytes").

Assessment: In succinct bullet points, list out the following, if available:

Formulate a preliminary diagnosis or a list of potential diagnoses based on the subjective and objective data. (e.g., "Possible upper respiratory infection", "Differential diagnoses include pneumonia and bronchitis").

Briefly explain the reasoning behind the diagnosis/diagnoses using appropriate medical terminology (e.g., "Acute onset of cough and fatigue suggests upper respiratory infection. Negative urinalysis argues against urinary tract infection").

Plan: In succinct bullet points, list out suggested treatment plan for the patient, if available:

Recommendations for further tests if needed (e.g., "Chest X-ray to rule out pneumonia").

Explanation of prescribed medications or treatments (e.g., "Amoxicillin 500mg three times daily for 7 days").

Check for any adverse reactions of medications.

Instructions for the patient regarding self-care and follow-up (e.g., "Increase fluid intake", "Schedule a follow-up

appointment in 3 days if symptoms worsen").

Add a section on detailed references to evidence-based medical research literature that would be relevant to the

symptoms that the patient has experienced.

Maintain patient confidentiality by avoiding any personal details beyond what's necessary for medical documentation.

After you have generated a summary, check back with the original input text to confirm that the generated summary

accurately preserves the facts presented or discussed in the consultation.

Repeat this validation twice to ensure the AI is not hallucinating.

"""

)

client = AzureOpenAIWrapper()

output = client.create\_chat\_completion(prompt)

summarised\_output = {"conversation": output}

return summarised\_output

**Main Execution**

The main execution block reads a JSON file containing the conversation transcription, calls the summarize\_text function to generate the SOAP note, and prints the summary.

python

Copy code

if \_\_name\_\_ == '\_\_main\_\_':

with open("input\_file.json", "r") as f:

json\_text = json.load(f)

summary = summarize\_text(json\_text)

print(summary)

**Detailed Code Documentation**

**Class AzureOpenAIWrapper**

**\_\_init\_\_()**

* **Description**: Initializes the AzureOpenAIWrapper instance by loading environment variables required for API configuration.
* **Parameters**: None
* **Returns**: None

**create\_chat\_completion(prompt, max\_tokens=3000, temperature=0.2, top\_p=0.95)**

* **Description**: Performs chat completion using the configured Azure OpenAI model.
* **Parameters**:
  + prompt (str): The prompt text for the conversation construction.
  + max\_tokens (int, optional): The maximum number of tokens allowed in the response. Defaults to 3000.
  + temperature (float, optional): Controls the randomness of the generated text. Defaults to 0.2.
  + top\_p (float, optional): The probability of picking the top words in the vocabulary. Defaults to 0.95.
* **Returns**: str: The generated conversation reconstruction.

**Function summarize\_text(json\_transcription)**

* **Description**: Processes a JSON transcription of a conversation and generates a structured SOAP note using the Azure OpenAI model.
* **Parameters**:
  + json\_transcription (dict): A JSON object containing the conversation to be summarized.
* **Returns**: dict: A dictionary containing the summarized conversation.

**Main Execution**

* Reads a JSON file containing the conversation transcription.
* Calls the summarize\_text function to generate the SOAP note.
* Prints the generated summary.