From azure.core.credentials import AzureKeyCredential

From azure.search.documents import SearchClient

Import openai

# Step 1: Set up your Azure Cognitive Search and OpenAI credentials

Search\_service\_name = “<your-search-service-name>”

Index\_name = “finalversion2-empidx”

Admin\_api\_key = “<your-search-admin-api-key>”

Openai.api\_type = “azure”

Openai.api\_base = https://<your-openai-endpoint>.openai.azure.com/

Openai.api\_version = “2024-08-01-preview”

Openai.api\_key = “<your-openai-api-key>”

Deployment\_name = “gpt-4o”

# Step 2: Create a SearchClient

Endpoint = fhttps://{search\_service\_name}.search.windows.net

Search\_client = SearchClient(endpoint=endpoint, index\_name=index\_name, credential=AzureKeyCredential(admin\_api\_key))

# Step 3: Define your search query

Search\_query = “total number of awards won by sales and marketing”

# Step 4: Perform the search and collect results

Results = search\_client.search(search\_query)

# Step 5: Collect search results into a single string

Search\_results = [result[‘content’] for result in results if ‘content’ in result]

Search\_results\_text = “\n”.join(search\_results)

# Step 6: Function to chunk text into smaller sections to fit within OpenAI limits

Def chunk\_text(text, max\_length=4000):

“””Break down the text into chunks of a specified length.”””

Chunks = []

While len(text) > max\_length:

Split\_point = text[:max\_length].rfind(‘ ‘) # Split at the last space within limit

If split\_point == -1:

Split\_point = max\_length

Chunks.append(text[:split\_point])

Text = text[split\_point:].strip()

Chunks.append(text)

Return chunks

# Step 7: Chunk the search results text

Chunks = chunk\_text(search\_results\_text)

# Step 8: Function to process each chunk with Azure OpenAI

Def process\_chunk(chunk):

“””Query the OpenAI model with a given chunk.”””

Try:

Response = openai.ChatCompletion.create(

Engine=deployment\_name,

Messages=[

{“role”: “system”, “content”: “You are a helpful assistant.”},

{“role”: “user”, “content”: f”Based on the following data, calculate total number of awards won by sales and marketing:\n\n{chunk}”}

],

Max\_tokens=800 # Adjust as needed to control response size

)

Return response.choices[0].message[‘content’].strip()

Except Exception as e:

Print(f”Error processing chunk: {e}”)

Return None

# Step 9: Process each chunk and collect responses

All\_responses = []

For chunk in chunks:

Response = process\_chunk(chunk)

If response:

All\_responses.append(response)

# Step 10: Combine the results from all chunks

Final\_output = “\n”.join(all\_responses)

# Step 11: (Optional) Summarize the combined output if necessary

Summary\_response = openai.ChatCompletion.create(

Engine=deployment\_name,

Messages=[

{“role”: “system”, “content”: “You are a helpful assistant.”},

{“role”: “user”, “content”: f”Summarize the following information:\n\n{final\_output}”}

],

Max\_tokens=1000 # Adjust as needed

)

# Step 12: Print the final summarized output

Print(“Final Summary:”, summary\_response.choices[0].message[‘content’].strip())