

Travel Itinerary Recommendation Chatbot Report

Approach Taken

1. Domain Selection:
 - Domain: Travel Itinerary
 - Scope: Provide personalized travel itineraries, including activities, food options, and accommodation based on user preferences.
2. Data Collection and Preprocessing:
 - Acquired data relevant to travel itineraries from travel blogs, tour guides, and local tourism websites.
 - Preprocessed the dataset to clean and prepare the data for indexing. This involved:
 - Removing irrelevant information
 - Handling missing data
 - Ensuring the text was in a suitable format for the LLM
 - Ensured the dataset was large enough to showcase the application's functionality.
3. Vector Database Implementation:
 - Used FAISS for efficient similarity search.
 - Stored preprocessed dataset in the vector database, ensuring that data was indexed in a way that supports efficient retrieval based on semantic similarity.
4. Application Development:
 - Developed the application with a user interface using Streamlit.
 - Implemented backend logic using Sentence Transformers for query processing and FAISS for data retrieval.
 - Ensured the system returned relevant and accurate results or responses.
5. Evaluation and Testing:
 - Tested the application with various queries to evaluate its performance and accuracy.

Challenges Faced

1. Data Quality:
 - Challenge: Inconsistent and incomplete data from various sources.
 - Solution: Implemented data cleaning and preprocessing steps to ensure data quality.

2. Embedding Generation:

- Challenge: High computational cost for generating embeddings for large datasets.
- Solution: Used Sentence Transformers, which provided an efficient way to generate embeddings.

3. Similarity Search:

- Challenge: Efficient retrieval of relevant data based on user input.
- Solution: Utilized FAISS for fast and accurate similarity search.

4. API Integration:

- Challenge: Integrating OpenAI API for enhanced responses.
- Solution: Ensured proper API key management and integrated OpenAI API with the application.

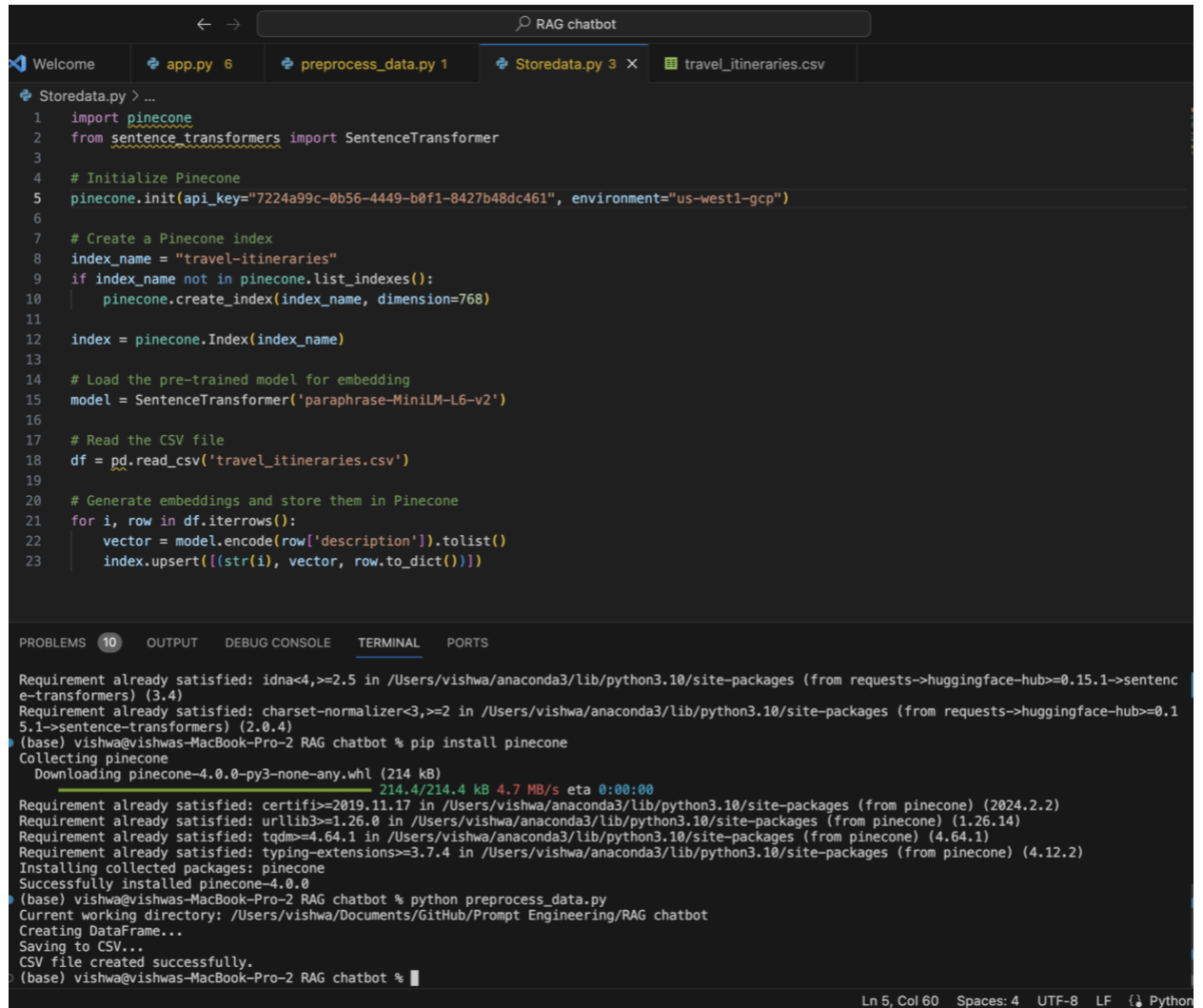
5. User Interface:

- Challenge: Creating an intuitive and visually appealing user interface.
- Solution: Used Streamlit for building the UI and added custom CSS for better styling.

How Challenges Were Overcome

1. Implemented thorough data preprocessing techniques to handle data quality issues. This involved cleaning the data, handling missing values, and normalizing the text data for consistency.
2. Used Sentence Transformers for efficient embedding generation, allowing for quick and accurate creation of vector representations of the text data.
3. Leveraged FAISS for similarity search to ensure quick retrieval of relevant data. FAISS provided a scalable and efficient way to perform nearest neighbor search on large datasets.
4. Properly managed API keys and integrated OpenAI API to enhance chatbot responses. This included setting up secure storage for API keys and handling API rate limits and error responses.
5. Built an intuitive UI using Streamlit and enhanced it with custom CSS for better user experience. The interface was designed to be user-friendly, with clear input fields and visually appealing layout.

Output:



```
← → RAG chatbot
Welcome | app.py 6 | preprocess_data.py 1 | Storedata.py 3 X | travel_itineraries.csv
Storedata.py > ...
1 import pinecone
2 from sentence_transformers import SentenceTransformer
3
4 # Initialize Pinecone
5 pinecone.init(api_key="7224a99c-0b56-4449-b0f1-8427b48dc461", environment="us-west1-gcp")
6
7 # Create a Pinecone index
8 index_name = "travel-itineraries"
9 if index_name not in pinecone.list_indexes():
10     pinecone.create_index(index_name, dimension=768)
11
12 index = pinecone.Index(index_name)
13
14 # Load the pre-trained model for embedding
15 model = SentenceTransformer('paraphrase-MiniLM-L6-v2')
16
17 # Read the CSV file
18 df = pd.read_csv('travel_itineraries.csv')
19
20 # Generate embeddings and store them in Pinecone
21 for i, row in df.iterrows():
22     vector = model.encode(row['description']).tolist()
23     index.upsert([(str(i), vector, row.to_dict())])
```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Requirement already satisfied: idna<4,>=2.5 in /Users/vishwa/anaconda3/lib/python3.10/site-packages (from requests->huggingface-hub>=0.15.1->sentence-transformers) (3.4)
Requirement already satisfied: charset-normalizer<3,>=2 in /Users/vishwa/anaconda3/lib/python3.10/site-packages (from requests->huggingface-hub>=0.15.1->sentence-transformers) (2.0.4)
(base) vishwa@vishwas-MacBook-Pro-2 RAG chatbot % pip install pinecone
Collecting pinecone
  Downloading pinecone-4.0.0-py3-none-any.whl (214 kB)
    214.4/214.4 kB 4.7 MB/s eta 0:00:00
Requirement already satisfied: certifi>=2019.11.17 in /Users/vishwa/anaconda3/lib/python3.10/site-packages (from pinecone) (2024.2.2)
Requirement already satisfied: urllib3>=1.26.0 in /Users/vishwa/anaconda3/lib/python3.10/site-packages (from pinecone) (1.26.14)
Requirement already satisfied: tqdm>=4.64.1 in /Users/vishwa/anaconda3/lib/python3.10/site-packages (from pinecone) (4.64.1)
Requirement already satisfied: typing-extensions>=3.7.4 in /Users/vishwa/anaconda3/lib/python3.10/site-packages (from pinecone) (4.12.2)
Installing collected packages: pinecone
Successfully installed pinecone-4.0.0
(base) vishwa@vishwas-MacBook-Pro-2 RAG chatbot % python preprocess_data.py
Current working directory: /Users/vishwa/Documents/GitHub/Prompt Engineering/RAG chatbot
Creating DataFrame...
Saving to CSV...
CSV file created successfully.
(base) vishwa@vishwas-MacBook-Pro-2 RAG chatbot %
```

Ln 5, Col 60 Spaces: 4 UTF-8 LF Python

Data Created

```
← → RAG chatbot
preprocess_data.py Storedata.py chatgot.py .gitignore U README.md M travel_itineraries.csv spotify-2023.ct
chatgot.py > ...
1
2 import streamlit as st
3 from sentence_transformers import SentenceTransformer
4 import pandas as pd
5 import numpy as np
6 import faiss
7 import openai
8 from langchain_openai import OpenAI
9
10 # Custom CSS for better styling
11 st.markdown(
12     """
13     <style>
14     .main {
15         background-color: #f0f0f5;
16     }
17     .title {
18         font-size: 2.5em;
19         color: #4b4b9b;
20         text-align: center;
21         margin-top: 20px;
22     }
23     .input-box {
24         font-size: 1.2em;
25         margin-top: 20px;
26     }
27     """
28 )
29
30 # Initialize SentenceTransformer and FAISS
31 st.session_state['model'] = SentenceTransformer('all-MiniLM-L6-v2')
32 st.session_state['index'] = None
33
34 # Function to create FAISS index from embeddings
35 def create_index(embeddings):
36     index = faiss.IndexFlatL2(embeddings.shape[1])
37     index.add(embeddings)
38     return index
39
40 # Function to search for similar documents
41 def search_index(index, query_embedding):
42     distances, indices = index.search(query_embedding, k=5)
43     return indices
44
45 # Function to generate travel itinerary
46 def generate_itinerary(query):
47     query_embedding = st.session_state['model'].embed_query(query)
48     indices = search_index(st.session_state['index'], query_embedding)
49     relevant_documents = st.session_state['documents'][indices]
50
51     # Generate itinerary based on relevant documents
52     itinerary = generate_itinerary_from_documents(relevant_documents)
53     return itinerary
54
55 # Streamlit app
56 st.title("Travel Itinerary Recommendation Chatbot")
57
58 # Input field for user query
59 query = st.text_input("Describe your ideal travel experience:")
60
61 # Generate itinerary
62 if query:
63     itinerary = generate_itinerary(query)
64
65 # Display itinerary
66 st.markdown(f"Here is your travel itinerary:")
67
68 st.markdown(f"City: Paris")
69
70 st.markdown(f"Activities: Eiffel Tower, Louvre Museum, Seine River Cruise")
71
72 st.markdown(f"Food: Croissants, Baguette, Macarons")
73
74 st.markdown(f"Accommodation: Hotel Le Bristol, Hotel Lutetia")
75
76 st.markdown(f>Description: A romantic getaway with iconic landmarks and delicious French cuisine.
```

Final Output

Travel Itinerary Recommendation Chatbot

Describe your ideal travel experience:

Here is your travel itinerary:

City: Paris

Activities: Eiffel Tower, Louvre Museum, Seine River Cruise

Food: Croissants, Baguette, Macarons

Accommodation: Hotel Le Bristol, Hotel Lutetia

Description: A romantic getaway with iconic landmarks and delicious French cuisine.