Code Documentation : asp.py

Overview

This file implements a semantic quote retrieval system that uses machine learning models and vector search for retrieving quotes based on NLQ. It integrates:

- Sentence Transformers for semantic embeddings
- FAISS for similarity search
- Streamlit for the user interface
- Google Gemini for generating answers

1. Imports

We import pandas, numpy, faiss, streamlit, sentence_transformer and genai

2. Data Preprocessing

This block:

- Loads quotes from a Hugging Face dataset
- Cleans missing values
- Converts text to lowercase
- Combines quote, author, and tags into a single string for embedding

---

3. Embedding Model and FAISS Index

```
model1 =
SentenceTransformer(r"C:\\Users\\lovyv\\Downloads\\ai_ass\\fine_tuned_quote_model")
embeddings = model1.encode(df['combined'].tolist(), show_progress_bar=True)
embeddings = np.ascontiguousarray(embeddings, dtype=np.float32)
index = faiss.IndexFlatL2(embeddings.shape[1])
index.add(embeddings)
```

- Loads a local fine-tuned model
- Generates embeddings for all quotes
- Creates and populates a FAISS index for similarity search

4. Quote Retrieval Function

def retrieve(query, top_k=5)

- Encodes the input query
- Searches the FAISS index for top-k similar quotes
- Returns matching quotes with metadata and similarity score

5. Google Gemini Integration

genai.configure(api_key="<API_KEY>")
model = genai.GenerativeModel("gemini-1.5-flash")

- Configures Gemini with API key
- Loads the generative model

6. Structured Answer Generation

def generate_answer(query, context_quotes):

- Builds a context-rich prompt from retrieved quotes
- Asks Gemini to generate a structured JSON response

7. Streamlit Web Interface

st.title("Semantic Quote Retriever")
query = st.text_input("Enter your query (e.g., quotes about courage by women authors):")

- Renders a web interface
- Takes user query input

- Shows retrieved quotes
- Provides an option to generate a structured answer