4a_MongoDB_Vector_SQL

November 11, 2024

MongoDB Vector Set up

link: https://cloud.mongodb.com/

Loading packages and libraries into notebook

[ ]:
```
# Importing the required libraries
from pymongo import MongoClient
from sentence_transformers import SentenceTransformer # https://huggingface.co/
 ↪thenlper/gte-large
import os
from dotenv import load_dotenv
from datasets import load_dataset
import pandas as pd
from pymongo.mongo_client import MongoClient
```

Accessing secrets

[ ]:
```
# Accessing the secrets from the environment variables
load_dotenv()
MONGO_URI_SQL = os.getenv("MONGO_URI_SQL")
HF_Token = os.getenv("HF_TOKEN")
```

Prepare and load dataset and transform to dataframe

[ ]:
```
# Upload the dataset and transform to dataframe
# Define the dataset path
dataset_path = "SpiderTrain_complete.csv"
print("Dataset Path:", dataset_path)

# Check if the file exists at the specified path
if not os.path.isfile(dataset_path):
    raise FileNotFoundError(f"Unable to find the file at {dataset_path}")

# Load the dataset
dataset = load_dataset('csv', data_files=dataset_path)

# Convert the dataset to a pandas dataframe
dataset_df = pd.DataFrame(dataset["train"])
```

```
# Print a few rows to verify
print(dataset_df.head())
```

Setting up embedding model and creating embeddings

[5]:
```
# Setting the embedding model and getting the embeddings for the dataframe
embedding_model = SentenceTransformer("thenlper/gte-large")
def get_embedding(text: str) -> list[float]:
    if not text.strip():
        print("Attempted to get embedding for empty text.")
        return []

    embedding = embedding_model.encode(text)

    return embedding.tolist()
dataset_df["embedding"] = dataset_df["Query"].apply(get_embedding)
```

Connecting to Vector Database

[ ]:
```
# MongoDB setup
client = MongoClient(MONGO_URI_SQL)
dbName = "MVector"
collectionName = "MTSQL"
collection = client[dbName][collectionName]
index_name = "vector_index_sql"

# Send a ping to confirm a successful connection
try:
    client.admin.command('ping')
    print("Pinged your deployment. You successfully connected to MongoDB!")
except Exception as e:
    print(e)
```

Delete all Content from Vector DB before Data Ingestion

[ ]:
```
# Delete any existing records in the collection before loading the new data
collection.delete_many({})
```

Load Data into Vector DB

[ ]:
```
# Insert the documents into the collection
documents = dataset_df.to_dict("records")
collection.insert_many(documents)
print("Data ingestion into MongoDB completed")
```