Documentation

This script demonstrates how to integrate a Neo4j graph database with an OpenAI language model to query movie data using natural language.

The script involves several key steps: setting up environment variables, loading datasets, connecting to the Neo4j database, inserting data, and querying the database using a language model.

Here is a detailed breakdown of each part of the script:

* Importing Library and Modules
  + langchain\_community.graphs: This module provides a Neo4jGraph class to interact with the Neo4j database.
  + datasets: This library from Hugging Face is used to load datasets.
  + neo4j: This is the official Neo4j driver for Python, used to connect and interact with a Neo4j database.
  + langchain.chains: This module contains chain classes for creating complex workflows, including GraphCypherQAChain.
  + langchain\_openai: This module includes integration with OpenAI’s language models, specifically ChatOpenAI.
  + os: This standard library module is used to manage environment variables.
* Load OpenAI API Key from a Text File and Set It as an Environment Variable
* openaiAPI.txt: This file contains the OpenAI API key in the format API\_KEY=<your\_openai\_api\_key>.
  + The script reads the API key from the file and sets it as an environment variable using os.environ.
* Load Neo4j Credentials from a Text File
  + Neo4jCredentials.txt: This file contains the Neo4j connection details in the format. The script reads the first three lines of this file and stores the key-value pairs in a dictionary called config.
  + Extract Neo4j Connection Details from the Config Dictionary.
  + The script extracts the URI, username, and password from the config dictionary for connecting to the Neo4j database.
* Initialize a Neo4jGraph Object with the Connection Details
* Neo4jGraph: This class is initialized with the connection details to interact with the Neo4j database.
* Load the Movie Dataset from Hugging Face
  + load\_dataset: This function from the datasets library loads the movie dataset hosted on Hugging Face.
* Insert the Dataset into the Neo4j Database
* It connects to the Neo4j database and inserts the movie dataset into it. Each record is inserted as a Movie node.
* GraphDatabase.driver: Creates a driver instance to connect to the Neo4j database.
* Initialize a GraphCypherQAChain with OpenAI's Chat Model and the Neo4j Graph
* GraphCypherQAChain.from\_llm: Initializes a chain for querying the graph database using natural language.
* ChatOpenAI: The language model used for generating Cypher queries based on natural language input. The temperature parameter controls the randomness of the model's output, with lower values making it more deterministic.
* verbose: Enables detailed logging for debugging and tracing.
* validate\_cypher: Ensures that the generated Cypher queries are valid.
* Execute a Query and Print the Result.