import os

os.environ['TRANSFORMERS\_CACHE'] = "/apps/opt/data"

os.environ['HTTP\_PROXY'] = "[http://proxy.ebiz.x.com:80"](http://proxy.ebiz.x.com:80%22%20)

from langchain.llms.huggingface\_pipeline import HuggingFacePipeline

from langchain.utilities import SQLDatabase

from langchain\_experimental.sql import SQLDatabaseChain

from transformers import AutoTokenizer, AutoModelForCausalLM, pipeline

from langchain.prompts import PromptTemplate

model\_id = "EleutherAI/gpt-neo-2.7B"

tokenizer = AutoTokenizer.from\_pretrained(model\_id)

model = AutoModelForCausalLM.from\_pretrained(model\_id)

pipeline = pipeline("text-generation", model=model, device=-1, tokenizer=tokenizer, max\_length=2048)

llm = HuggingFacePipeline(pipeline=pipeline, model\_kwargs={'temperature': 0})

###############################################################

conuri = f"oracle+oracledb://{user}:{password}@{dsn}"

print(f"conuri: {conuri}")

db = SQLDatabase.from\_uri(conuri)

#############################################################

#db\_chain = SQLDatabaseChain.from\_llm(llm, db, verbose=True)

#query = "select PO\_ID from PO\_WKFLW\_VIEW where PO\_ID = '1000269452'"

# sql\_query = db\_chain.generate\_sql(query)

#p = db\_chain.run(query)

#print("printing output",p)

print("part -II")

###############################################################

query = "get me details of ticket 'INC011039341'"

from langchain.chains import create\_sql\_query\_chain

#chain = create\_sql\_query\_chain(llm, db)

#response = chain.invoke({"question": query})

#print(response)

#db.run(response)

#######

from langchain.prompts import PromptTemplate

TEMPLATE = """Given an input question, first create a syntactically correct {dialect} query to run, then look at the results of the query and return the answer.

given input is ticket value in the question.

Use the following format:

Question: "Question here"

SQLQuery: "SQL Query to run"

SQLResult: "Result of the SQLQuery"

Answer: "Final answer here"

#

Only use the following tables:

#

{table\_info}.

#

Some examples of SQL queries that correspond to questions are:

#

# {few\_shot\_examples}

#

Question: {input}"""

CUSTOM\_PROMPT = PromptTemplate(

input\_variables=["input", "few\_shot\_examples", "table\_info", "dialect"],

template=TEMPLATE,

)

print(db.table\_info)

db = SQLDatabase.from\_uri(

conuri,

include\_tables=[

"combined\_teams"

] # I included only one table to save tokens in the prompt :)

, sample\_rows\_in\_table\_info=2,

)

print("c2")

db\_chain = SQLDatabaseChain.from\_llm(llm, db, verbose=True)

db\_chain.run(query)