Text to SQL approaches

Security consideration

Virtual Environment

python3 -m venv .venv

source .venv/bin/activate

Langchain/LlamaIndex

Environment variables

Save sensitive information like database connection credentials in a .env files and exclude file from git using .gitignore

Postgres SQL connection string example

DATABASE\_URL=postgresql+psycopg2://username:password@host:port/database

from dotenv import load\_dotenv

Use load\_dotenv() from dotenv to extract details

load\_dotenv()

connection\_string=os.getenv(‘DATABASE\_URL’)

Database connection

Langchain

from langchain\_community.utilities import SQLDatabase

db=SQLDatabase.from\_uri(connection\_string)

data=db.run(query)

LlamaIndex

from llama\_index.readers.database import DatabaseReader

db=DatabaseReader(connection\_string)

data=db.load\_data(query)

Prompt Engineering

Langchain - FewShotPromptTemplate

Zero shot - no examples, just a basic query

One - Using one example in prompt

from langchain\_core.prompts.few\_shot import FewShotPromptTemplate

from langchain\_core.prompts import PromptTemplate

prompt\_template = “””

You are a {dialect} expert. Generate SQL statements using the table information provided below:

{table\_info}

Generate SQL statements based on the examples provided.

Use the following format:

Question: {question}

SQL Query: {sql\_query}

“””

example\_prompt=PromptTemplate.from\_template(prompt\_template, input\_variables[

“dialect”,

“table\_info”,

“question”,

“sql\_query”

])

examples = [

{

“question”: “How many artists are there?”,

”query”: “SELECT COUNT(\*) artists from table”

}

]

one\_shot\_prompt = FewShotPromptTemplate(

examples=examples,

example\_prompt=example\_prompt,

input\_variables[“dialect”,“table\_info”,“question”,“sql\_query”]

)