knowledgegraphdb

August 24, 2024

[1]:
```
!pip install --upgrade --quiet  langchain langchain-community langchain-groq␣
↪neo4j
```

```
                                           50.4/50.4 kB
3.1 MB/s eta 0:00:00
                                         997.8/997.8 kB
18.6 MB/s eta 0:00:00
                                         2.3/2.3 MB
55.5 MB/s eta 0:00:00
                                         293.6/293.6 kB
16.4 MB/s eta 0:00:00
                                         103.5/103.5 kB
6.2 MB/s eta 0:00:00
                                         393.9/393.9 kB
22.5 MB/s eta 0:00:00
                                         149.1/149.1 kB
9.0 MB/s eta 0:00:00
                                         75.6/75.6 kB
4.1 MB/s eta 0:00:00
                                         77.9/77.9 kB
5.2 MB/s eta 0:00:00
                                         49.3/49.3 kB
2.9 MB/s eta 0:00:00
                                         141.9/141.9 kB
8.7 MB/s eta 0:00:00
                                         58.3/58.3 kB
3.7 MB/s eta 0:00:00
```

[3]:
```python
from google.colab import userdata

#Graphdb configuration
NEO4J_URI=userdata.get('NEO4J_URI')
NEO4J_USERNAME=userdata.get('NEO4J_USERNAME')
NEO4J_PASSWORD= userdata.get('NEO4J_PASSWORD')
```

[5]:
```
from langchain_community.graphs import Neo4jGraph
graph=Neo4jGraph(
    url=NEO4J_URI,
    username=NEO4J_USERNAME,
    password=NEO4J_PASSWORD,
)
```

[6]:
```
graph
```

[6]:
```
<langchain_community.graphs.neo4j_graph.Neo4jGraph at 0x7ceb2b6c67d0>
```

[7]:
```
groq_api_key=userdata.get("GROQ_API")
```

[9]:
```
from langchain_groq import ChatGroq

llm=ChatGroq(groq_api_key=groq_api_key,model_name="Gemma2-9b-It")
llm
```

[9]:
```
ChatGroq(client=<groq.resources.chat.completions.Completions object at
0x7ceaf901add0>, async_client=<groq.resources.chat.completions.AsyncCompletions
object at 0x7ceaf901bac0>, model_name='Gemma2-9b-It',
groq_api_key=SecretStr('**********'))
```

[15]:
```
from langchain_core.documents import Document
text="""
Elon Reeve Musk (born June 28, 1971) is a businessman and investor known for␣
  ↪his key roles in space
company SpaceX and automotive company Tesla, Inc. Other involvements include␣
  ↪ownership of X Corp.,
formerly Twitter, and his role in the founding of The Boring Company, xAI,␣
  ↪Neuralink and OpenAI.
He is one of the wealthiest people in the world; as of July 2024, Forbes␣
  ↪estimates his net worth to be
US$221 billion.Musk was born in Pretoria to Maye and engineer Errol Musk, and␣
  ↪briefly attended
the University of Pretoria before immigrating to Canada at age 18, acquiring␣
  ↪citizenship through
his Canadian-born mother. Two years later, he matriculated at Queen's␣
  ↪University at Kingston in Canada.
Musk later transferred to the University of Pennsylvania and received␣
  ↪bachelor's degrees in economics
 and physics. He moved to California in 1995 to attend Stanford University, but␣
  ↪dropped out after
  two days and, with his brother Kimbal, co-founded online city guide software␣
  ↪company Zip2.
 """
```

```
documents=[Document(page_content=text)]
print(documents)
```

```
[Document(page_content="\nElon Reeve Musk (born June 28, 1971) is a businessman
and investor known for his key roles in space\ncompany SpaceX and automotive
company Tesla, Inc. Other involvements include ownership of X Corp.,\nformerly
Twitter, and his role in the founding of The Boring Company, xAI, Neuralink and
OpenAI.\nHe is one of the wealthiest people in the world; as of July 2024,
Forbes estimates his net worth to be\nUS$221 billion.Musk was born in Pretoria
to Maye and engineer Errol Musk, and briefly attended\nthe University of
Pretoria before immigrating to Canada at age 18, acquiring citizenship
through\nhis Canadian-born mother. Two years later, he matriculated at Queen's
University at Kingston in Canada.\nMusk later transferred to the University of
Pennsylvania and received bachelor's degrees in economics\n and physics. He
moved to California in 1995 to attend Stanford University, but dropped out
after\n  two days and, with his brother Kimbal, co-founded online city guide
software company Zip2.\n ")]
```

[16]:
```
!pip install --upgrade --quiet langchain_experimental
```

```
                                204.3/204.3 kB
3.3 MB/s eta 0:00:00
```

[22]:
```
from langchain_experimental.graph_transformers import LLMGraphTransformer
llm_transformer=LLMGraphTransformer(llm=llm)
# Convert the document text into graph
```

[23]:
```
graph_documents=llm_transformer.convert_to_graph_documents(documents)
graph_documents
```

[23]:
```
[GraphDocument(nodes=[Node(id='Elon Reeve Musk', type='Person'), Node(id='Maye',
type='Person'), Node(id='Errol Musk', type='Person'), Node(id='University Of
Pretoria', type='University'), Node(id='Canada', type='Country'),
Node(id="Queen'S University At Kingston", type='University'),
Node(id='University Of Pennsylvania', type='University'), Node(id='California',
type='State'), Node(id='Stanford University', type='University'),
Node(id='Kimbal', type='Person'), Node(id='Zip2', type='Company'),
Node(id='Spacex', type='Company'), Node(id='Tesla, Inc.', type='Company'),
Node(id='X Corp.', type='Company'), Node(id='The Boring Company',
type='Company'), Node(id='Xai', type='Company'), Node(id='Neuralink',
type='Company'), Node(id='Openai', type='Company'), Node(id='Forbes',
type='Organization')], relationships=[Relationship(source=Node(id='Elon Reeve
Musk', type='Person'), target=Node(id='Maye', type='Person'), type='PARENT'),
Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='Errol Musk', type='Person'), type='PARENT'),
Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='University Of Pretoria', type='University'), type='ATTENDED'),
Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='Canada', type='Country'), type='IMMIGRATION'),
Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id="Queen'S University At Kingston", type='University'),
type='ATTENDED'), Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='University Of Pennsylvania', type='University'),
type='ATTENDED'), Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='California', type='State'), type='RELOCATION'),
Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='Stanford University', type='University'), type='ATTENDED'),
Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='Zip2', type='Company'), type='FOUNDED'),
Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='Spacex', type='Company'), type='INVOLVED'),
Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='Tesla, Inc.', type='Company'), type='INVOLVED'),
Relationship(source=Node(id='Elon Reeve Musk', type='Person'), target=Node(id='X
Corp.', type='Company'), type='INVOLVED'), Relationship(source=Node(id='Elon
Reeve Musk', type='Person'), target=Node(id='The Boring Company',
type='Company'), type='INVOLVED'), Relationship(source=Node(id='Elon Reeve
Musk', type='Person'), target=Node(id='Xai', type='Company'), type='INVOLVED'),
Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='Neuralink', type='Company'), type='INVOLVED'),
Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='Openai', type='Company'), type='INVOLVED'),
Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='Forbes', type='Organization'), type='NET_WORTH_ESTIMATE')],
source=Document(page_content="\nElon Reeve Musk (born June 28, 1971) is a
businessman and investor known for his key roles in space\ncompany SpaceX and
automotive company Tesla, Inc. Other involvements include ownership of X
Corp.,\nformerly Twitter, and his role in the founding of The Boring Company,
xAI, Neuralink and OpenAI.\nHe is one of the wealthiest people in the world; as
of July 2024, Forbes estimates his net worth to be\nUS$221 billion.Musk was born
in Pretoria to Maye and engineer Errol Musk, and briefly attended\nthe
University of Pretoria before immigrating to Canada at age 18, acquiring
citizenship through\nhis Canadian-born mother. Two years later, he matriculated
at Queen's University at Kingston in Canada.\nMusk later transferred to the
University of Pennsylvania and received bachelor's degrees in economics\n and
physics. He moved to California in 1995 to attend Stanford University, but
dropped out after\n  two days and, with his brother Kimbal, co-founded online
city guide software company Zip2.\n "))]
```

[24]:
```
graph_documents[0].nodes
```

[24]:
```
[Node(id='Elon Reeve Musk', type='Person'),
 Node(id='Maye', type='Person'),
 Node(id='Errol Musk', type='Person'),
 Node(id='University Of Pretoria', type='University'),
 Node(id='Canada', type='Country'),
 Node(id="Queen'S University At Kingston", type='University'),
 Node(id='University Of Pennsylvania', type='University'),
 Node(id='California', type='State'),
 Node(id='Stanford University', type='University'),
 Node(id='Kimbal', type='Person'),
 Node(id='Zip2', type='Company'),
 Node(id='Spacex', type='Company'),
 Node(id='Tesla, Inc.', type='Company'),
 Node(id='X Corp.', type='Company'),
 Node(id='The Boring Company', type='Company'),
 Node(id='Xai', type='Company'),
 Node(id='Neuralink', type='Company'),
 Node(id='Openai', type='Company'),
 Node(id='Forbes', type='Organization')]
```

[25]:
```
graph_documents[0].relationships
```

[25]:
```
[Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='Maye', type='Person'), type='PARENT'),
 Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='Errol Musk', type='Person'), type='PARENT'),
 Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='University Of Pretoria', type='University'), type='ATTENDED'),
 Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='Canada', type='Country'), type='IMMIGRATION'),
 Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id="Queen'S University At Kingston", type='University'),
type='ATTENDED'),
 Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='University Of Pennsylvania', type='University'),
type='ATTENDED'),
 Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='California', type='State'), type='RELOCATION'),
 Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='Stanford University', type='University'), type='ATTENDED'),
 Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='Zip2', type='Company'), type='FOUNDED'),
 Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='Spacex', type='Company'), type='INVOLVED'),
 Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='Tesla, Inc.', type='Company'), type='INVOLVED'),
 Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='X Corp.', type='Company'), type='INVOLVED'),
 Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='The Boring Company', type='Company'), type='INVOLVED'),
 Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='Xai', type='Company'), type='INVOLVED'),
 Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='Neuralink', type='Company'), type='INVOLVED'),
 Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='Openai', type='Company'), type='INVOLVED'),
 Relationship(source=Node(id='Elon Reeve Musk', type='Person'),
target=Node(id='Forbes', type='Organization'), type='NET_WORTH_ESTIMATE')]
```

[26]:
```
### Load the dataset of movie

movie_query="""
LOAD CSV WITH HEADERS FROM
'https://raw.githubusercontent.com/tomasonjo/blog-datasets/main/movies/
 ↪movies_small.csv' as row

MERGE(m:Movie{id:row.movieId})
SET m.released = date(row.released),
    m.title = row.title,
    m.imdbRating = toFloat(row.imdbRating)
FOREACH (director in split(row.director, '|') |
    MERGE (p:Person {name:trim(director)})
    MERGE (p)-[:DIRECTED]->(m))
FOREACH (actor in split(row.actors, '|') |
    MERGE (p:Person {name:trim(actor)})
    MERGE (p)-[:ACTED_IN]->(m))
FOREACH (genre in split(row.genres, '|') |
    MERGE (g:Genre {name:trim(genre)})
    MERGE (m)-[:IN_GENRE]->(g))
"""
```

[28]:
```
graph.query(movie_query)
```

[28]:
```
[]
```

[29]:
```
graph.refresh_schema()
print(graph.schema)
```

```
Node properties:
Person {born: INTEGER, name: STRING}
Movie {title: STRING, released: INTEGER, id: STRING, imdbRating: FLOAT}
Genre {name: STRING}
Relationship properties:

The relationships:
(:Person)-[:ACTED_IN]->(:Movie)
```

```
(:Person)-[:DIRECTED]->(:Movie)
(:Movie)-[:IN_GENRE]->(:Genre)
```

[30]:
```python
from langchain.chains import GraphCypherQAChain
chain=GraphCypherQAChain.from_llm(llm=llm,graph=graph,verbose=True)
chain
```

[30]:
```
GraphCypherQAChain(verbose=True,
graph=<langchain_community.graphs.neo4j_graph.Neo4jGraph object at
0x7ceb2b6c67d0>, cypher_generation_chain=LLMChain(prompt=PromptTemplate(input_va
riables=['question', 'schema'], template='Task:Generate Cypher statement to
query a graph database.\nInstructions:\nUse only the provided relationship types
and properties in the schema.\nDo not use any other relationship types or
properties that are not provided.\nSchema:\n{schema}\nNote: Do not include any
explanations or apologies in your responses.\nDo not respond to any questions
that might ask anything else than for you to construct a Cypher statement.\nDo
not include any text except the generated Cypher statement.\n\nThe question
is:\n{question}'),
llm=ChatGroq(client=<groq.resources.chat.completions.Completions object at
0x7ceaf901add0>, async_client=<groq.resources.chat.completions.AsyncCompletions
object at 0x7ceaf901bac0>, model_name='Gemma2-9b-It',
groq_api_key=SecretStr('**********'))),
qa_chain=LLMChain(prompt=PromptTemplate(input_variables=['context', 'question'],
template="You are an assistant that helps to form nice and human understandable
answers.\nThe information part contains the provided information that you must
use to construct an answer.\nThe provided information is authoritative, you must
never doubt it or try to use your internal knowledge to correct it.\nMake the
answer sound as a response to the question. Do not mention that you based the
result on the given information.\nHere is an example:\n\nQuestion: Which
managers own Neo4j stocks?\nContext:[manager:CTL LLC, manager:JANE STREET GROUP
LLC]\nHelpful Answer: CTL LLC, JANE STREET GROUP LLC owns Neo4j
stocks.\n\nFollow this example when generating answers.\nIf the provided
information is empty, say that you don't know the
answer.\nInformation:\n{context}\n\nQuestion: {question}\nHelpful Answer:"),
llm=ChatGroq(client=<groq.resources.chat.completions.Completions object at
0x7ceaf901add0>, async_client=<groq.resources.chat.completions.AsyncCompletions
object at 0x7ceaf901bac0>, model_name='Gemma2-9b-It',
groq_api_key=SecretStr('**********'))), graph_schema='Node properties are the
following:\nPerson {born: INTEGER, name: STRING},Movie {title: STRING, released:
INTEGER, id: STRING, imdbRating: FLOAT},Genre {name: STRING}\nRelationship
properties are the following:\n\nThe relationships are the following:\n(:Person)
-[:ACTED_IN]->(:Movie),(:Person)-[:DIRECTED]->(:Movie),(:Movie)-[:IN_GENRE]->(:G
enre)')
```

[32]:
```python
chain.invoke({"query":"Who was the director of the movie Restoration"})
# Restoration
```

```
> Entering new GraphCypherQAChain chain...
Generated Cypher:
MATCH (p:Person)-[:DIRECTED]->(m:Movie {title: "Restoration"})

RETURN p.name


Full Context:
[{'p.name': 'Michael Hoffman'}]

> Finished chain.
```

```
[32]: {'query': 'Who was the director of the movie Restoration',
       'result': 'Michael Hoffman  \n'}
```

```
[34]: response = chain.invoke({"query":"Who was the actor of the movie Restoration"})
```

```
> Entering new GraphCypherQAChain chain...
Generated Cypher:
MATCH (m:Movie {title: "Restoration"})<-[:ACTED_IN]-(p:Person)

RETURN p.name


Full Context:
[{'p.name': 'Robert Downey Jr.'}, {'p.name': 'Sam Neill'},

{'p.name': 'David Thewlis'}, {'p.name': 'Polly Walker'}]

> Finished chain.
```

```
[35]: response=chain.invoke({"query":"Give me the list of movie having imdb rating␣
        ↪more than 8"})
      response
```

```
> Entering new GraphCypherQAChain chain...
Generated Cypher:
MATCH (m:Movie) WHERE m.imdbRating > 8 RETURN m.title


Full Context:
```

```
[{'m.title': 'Toy Story'}, {'m.title': 'Heat'}, {'m.title':
'Casino'}, {'m.title': 'Twelve Monkeys (a.k.a. 12 Monkeys)'}, {'m.title': 'Seven
(a.k.a. Se7en)'}, {'m.title': 'Usual Suspects, The'}, {'m.title': 'Hate (Haine,
La)'}, {'m.title': 'Braveheart'}, {'m.title': 'Taxi Driver'}, {'m.title': 'Anne
Frank Remembered'}]

> Finished chain.
```

```
[35]: {'query': 'Give me the list of movie having imdb rating more than 8',
       'result': "I don't know the answer. \n"}
```

```
[ ]:
```