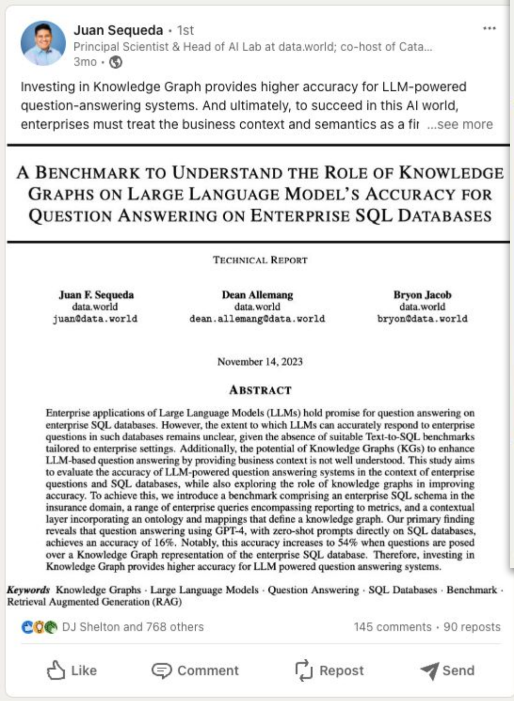
Going Meta #26
Unpicking the data.world benchmark on the role of KGs in LLM QA over enterprise data



Juan Sequeda · 1st
Principal Scientist & Head of AI Lab at data.world; co-host of Cata...
3mo ·

Investing in Knowledge Graph provides higher accuracy for LLM-powered question-answering systems. And ultimately, to succeed in this AI world, enterprises must treat the business context and semantics as a fir ...see more

A Benchmark to Understand the Role of Knowledge Graphs on Large Language Model's Accuracy for Question Answering on Enterprise SQL Databases

Technical Report

Juan F. Sequeda
data.world
juan@data.world

Dean Allemang
data.world
dean.allemang@data.world

Bryon Jacob
data.world
bryon@data.world

November 14, 2023

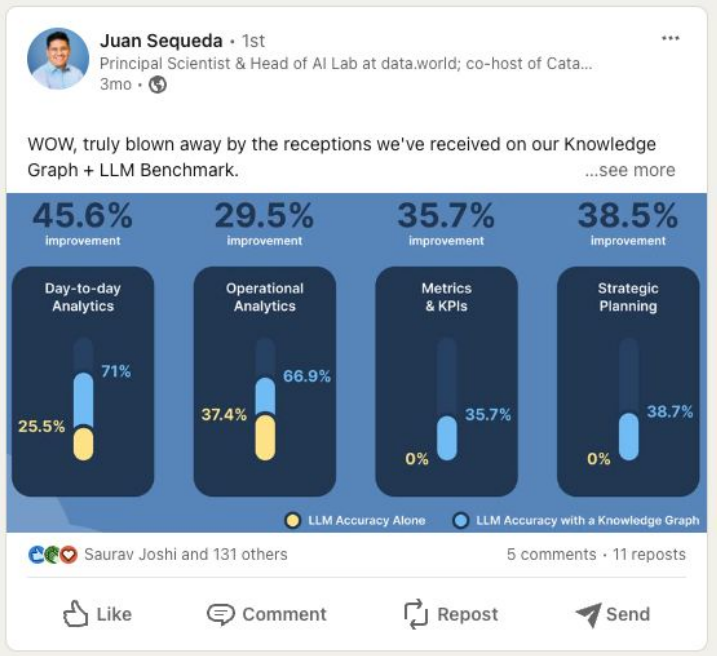
Abstract

Enterprise applications of Large Language Models (LLMs) hold promise for question answering on enterprise SQL databases. However, the extent to which LLMs can accurately respond to enterprise questions in such databases remains unclear, given the absence of suitable Text-to-SQL benchmarks tailored to enterprise settings. Additionally, the potential of Knowledge Graphs (KGs) to enhance LLM-based question answering by providing business context is not well understood. This study aims to evaluate the accuracy of LLM-powered question answering systems in the context of enterprise questions and SQL databases, while also exploring the role of knowledge graphs in improving accuracy. To achieve this, we introduce a benchmark comprising an enterprise SQL schema in the insurance domain, a range of enterprise queries encompassing reporting to metrics, and a contextual layer incorporating an ontology and mappings that define a knowledge graph. Our primary finding reveals that question answering using GPT-4, with zero-shot prompts directly on SQL databases, achieves an accuracy of 16%. Notably, this accuracy increases to 54% when questions are posed over a Knowledge Graph representation of the enterprise SQL database. Therefore, investing in Knowledge Graph provides higher accuracy for LLM powered question answering systems.

Keywords Knowledge Graphs · Large Language Models · Question Answering · SQL Databases · Benchmark · Retrieval Augmented Generation (RAG)

DJ Shelton and 768 others 145 comments · 90 reposts

Like Comment Repost Send

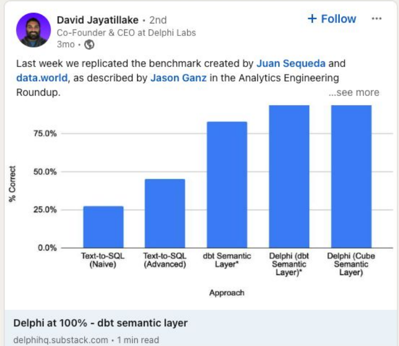


Juan Sequeda · 1st
Principal Scientist & Head of AI Lab at data.world; co-host of Cata...
3mo ·

WOW, truly blown away by the receptions we've received on our Knowledge Graph + LLM Benchmark. ...see more

Saurav Joshi and 131 others 5 comments · 11 reposts

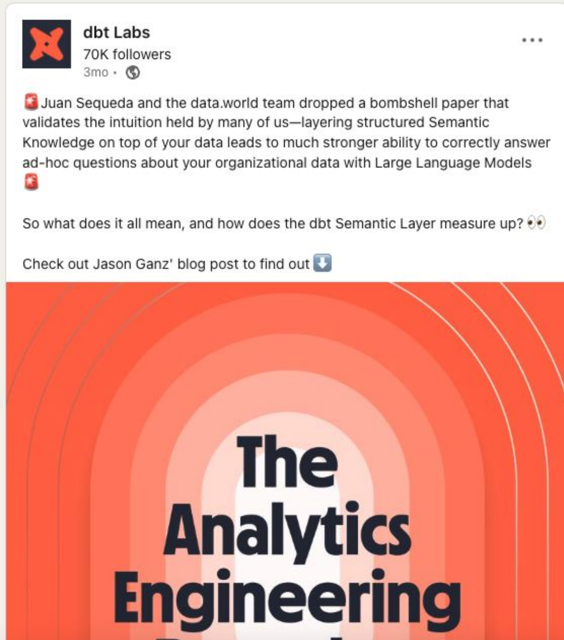
Like Comment Repost Send



David Jayatillake · 2nd + Follow
Co-Founder & CEO at Delphi Labs
3mo ·

Last week we replicated the benchmark created by Juan Sequeda and data.world, as described by Jason Ganz in the Analytics Engineering Roundup. ...see more

Delphi at 100% - dbt semantic layer
delphihq.substack.com · 1 min read

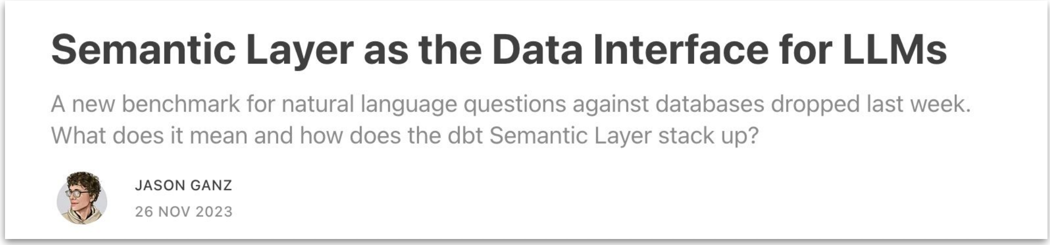


dbt Labs
70K followers
3mo ·

🚨Juan Sequeda and the data.world team dropped a bombshell paper that validates the intuition held by many of us—layering structured Semantic Knowledge on top of your data leads to much stronger ability to correctly answer ad-hoc questions about your organizational data with Large Language Models 🚨

So what does it all mean, and how does the dbt Semantic Layer measure up? 👀

Check out Jason Ganz' blog post to find out ⬇️

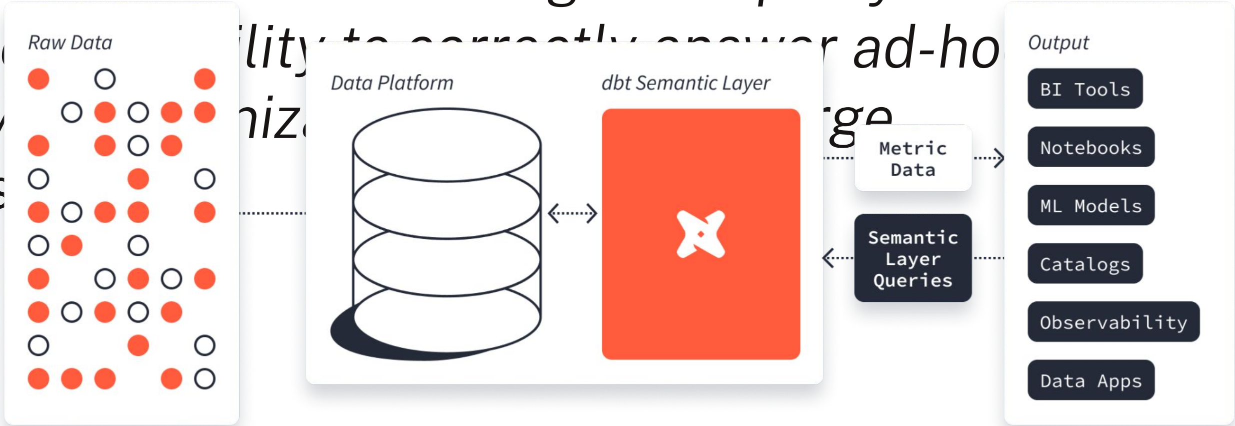


Semantic Layer as the Data Interface for LLMs

A new benchmark for natural language questions against databases dropped last week. What does it mean and how does the dbt Semantic Layer stack up?

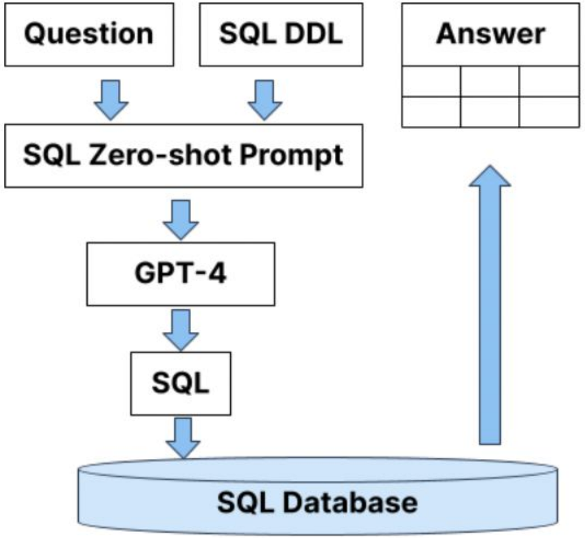
JASON GANZ
26 NOV 2023

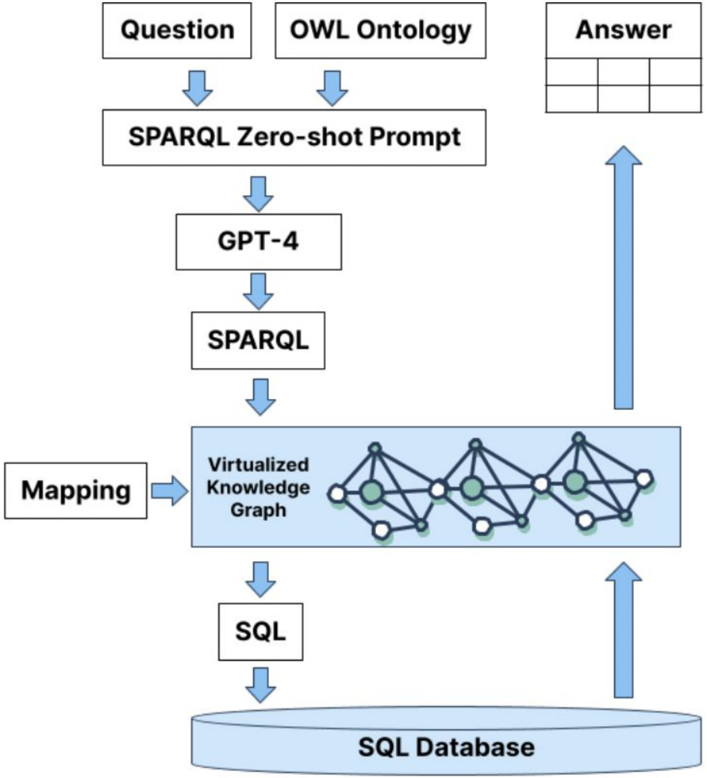
layering structured Semantic Knowledge on top of your data leads to much str ility to correctly answer ad-ho questions about y niz rge Language Models



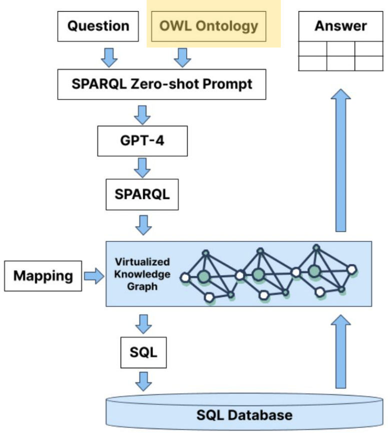
https://roundup.getdbt.com/p/semantic-layer-as-the-data-interface

The two systems

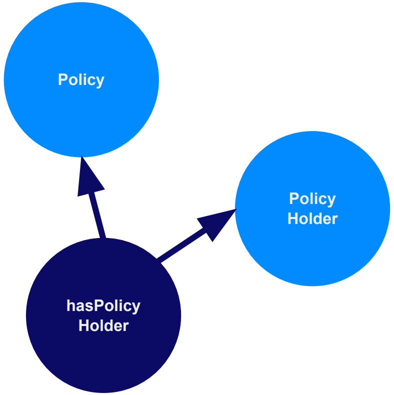




The Unpicking... The ontology



ACME_Insurance/ontology/insurance.ttl

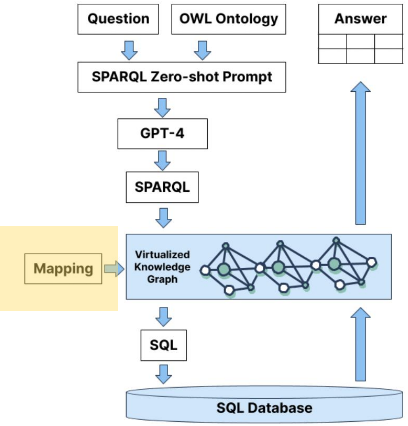


```
###  http://data.world/schema/insurance/PolicyHolder
in:PolicyHolder rdf:type owl:Class ;
                rdfs:isDefinedBy <http://data.world/schema/insurance/> ;
                rdfs:label "Policy Holder" .
```

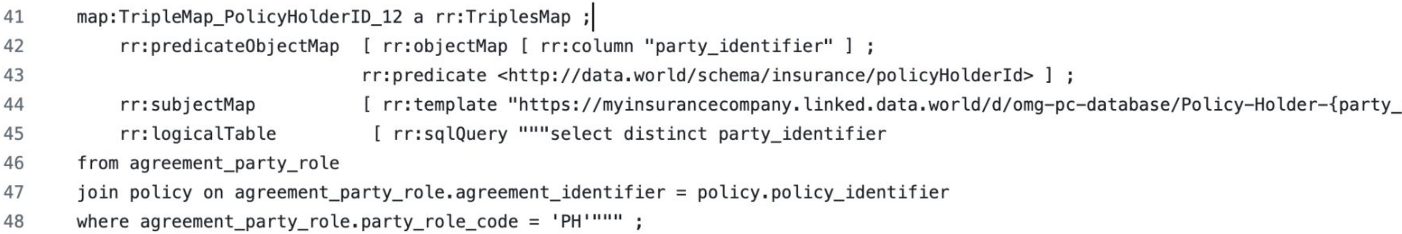
```
###  http://data.world/schema/insurance/hasPolicyHolder
in:hasPolicyHolder rdf:type owl:ObjectProperty ;
                   rdfs:domain in:Policy ;
                   rdfs:range in:PolicyHolder ;
                   rdfs:isDefinedBy <http://data.world/schema/insurance/> ;
                   rdfs:label "has policy holder" .

```

https://github.com/datadotworld/cwd-benchmark-data

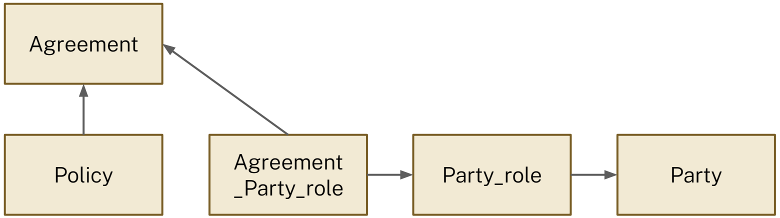
The Unpicking... The mapping onto-source data

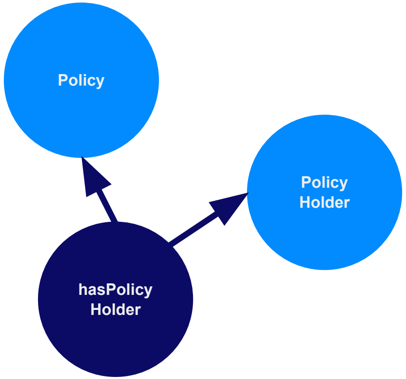


ACME_Insurance/data/data.world_P&C_Insurance_Ontology_V1.r2rml



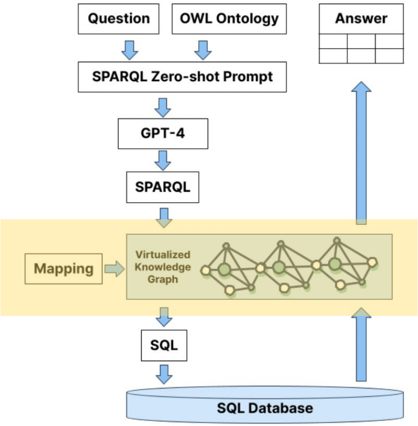
```
map:TripleMap_PolicyHolderID_12 a rr:TriplesMap ;
    rr:predicateObjectMap  [ rr:objectMap [ rr:column "party_identifier" ] ;
                           rr:predicate <http://data.world/schema/insurance/policyHolderId> ] ;
    rr:subjectMap          [ rr:template "https://myinsurancecompany.linked.data.world/d/omg-pc-database/Policy-Holder-{party_
    rr:logicalTable         [ rr:sqlQuery """select distinct party_identifier
from agreement_party_role
join policy on agreement_party_role.agreement_identifier = policy.policy_identifier
where agreement_party_role.party_role_code = 'PH'""" ;
```

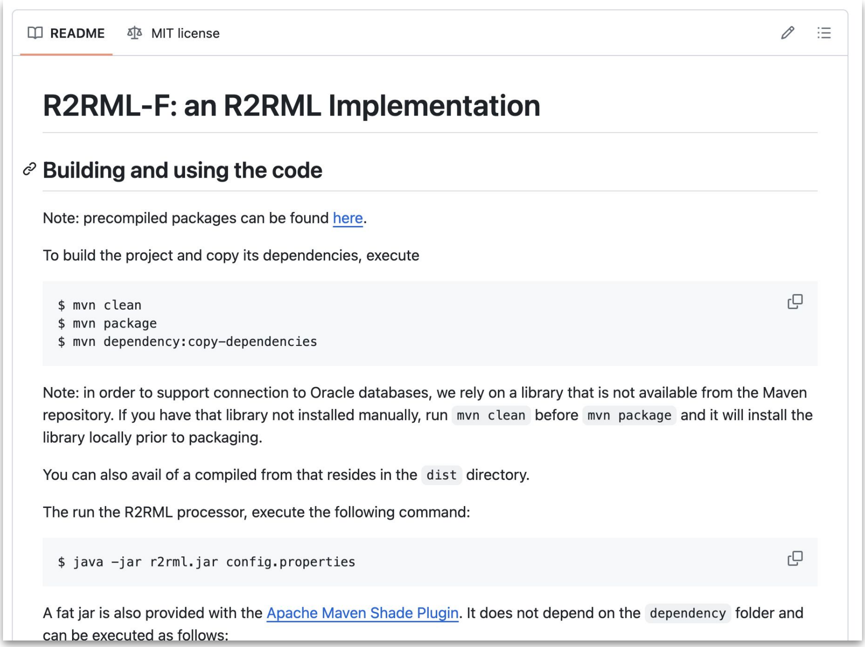




https://github.com/datadotworld/cwd-benchmark-data

The Unpicking… The mapping engine





R2RML-F: an R2RML Implementation

Building and using the code

Note: precompiled packages can be found here.

To build the project and copy its dependencies, execute

```
$ mvn clean
$ mvn package
$ mvn dependency:copy-dependencies
```

Note: in order to support connection to Oracle databases, we rely on a library that is not available from the Maven repository. If you have that library not installed manually, run `mvn clean` before `mvn package` and it will install the library locally prior to packaging.

You can also avail of a compiled from that resides in the `dist` directory.

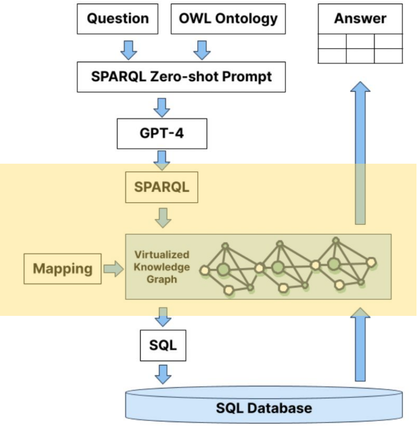
The run the R2RML processor, execute the following command:

```
$ java -jar r2rml.jar config.properties
```

A fat jar is also provided with the Apache Maven Shade Plugin. It does not depend on the `dependency` folder and can be executed as follows:

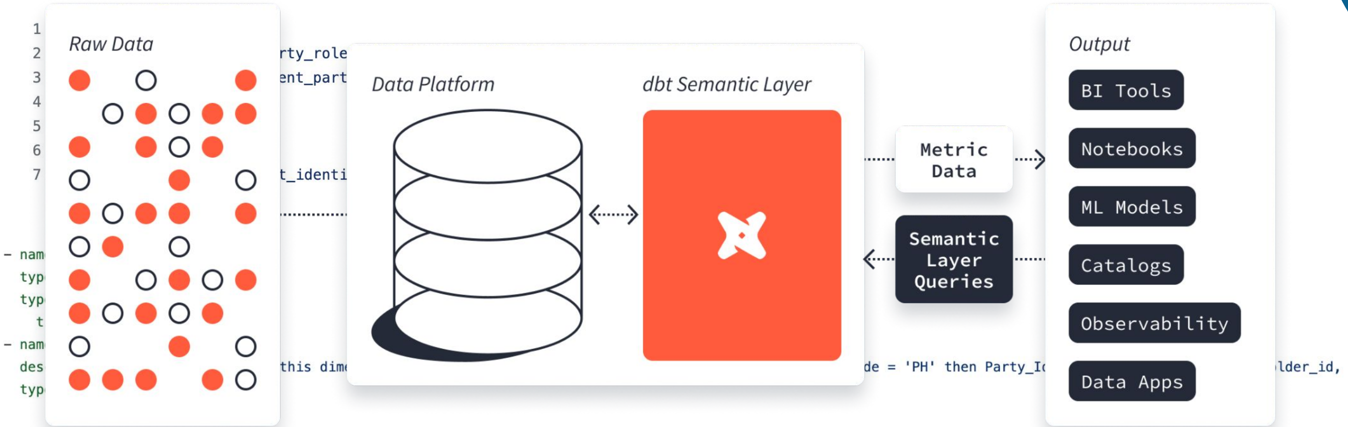
https://github.com/chrdebru/r2rml

The Unpicking... Alternatives to formalise the semantic layer



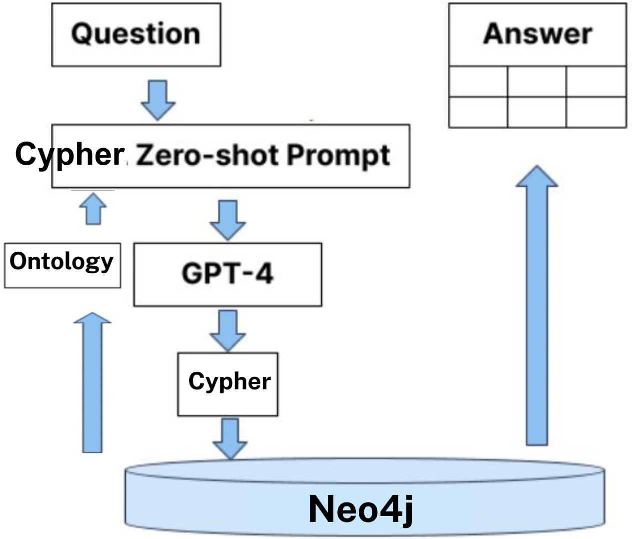
semantic-layer

models/omg_semantics/agreement_party_role.yaml



https://github.com/dbt-labs/semantic-layer-llm-benchmarking

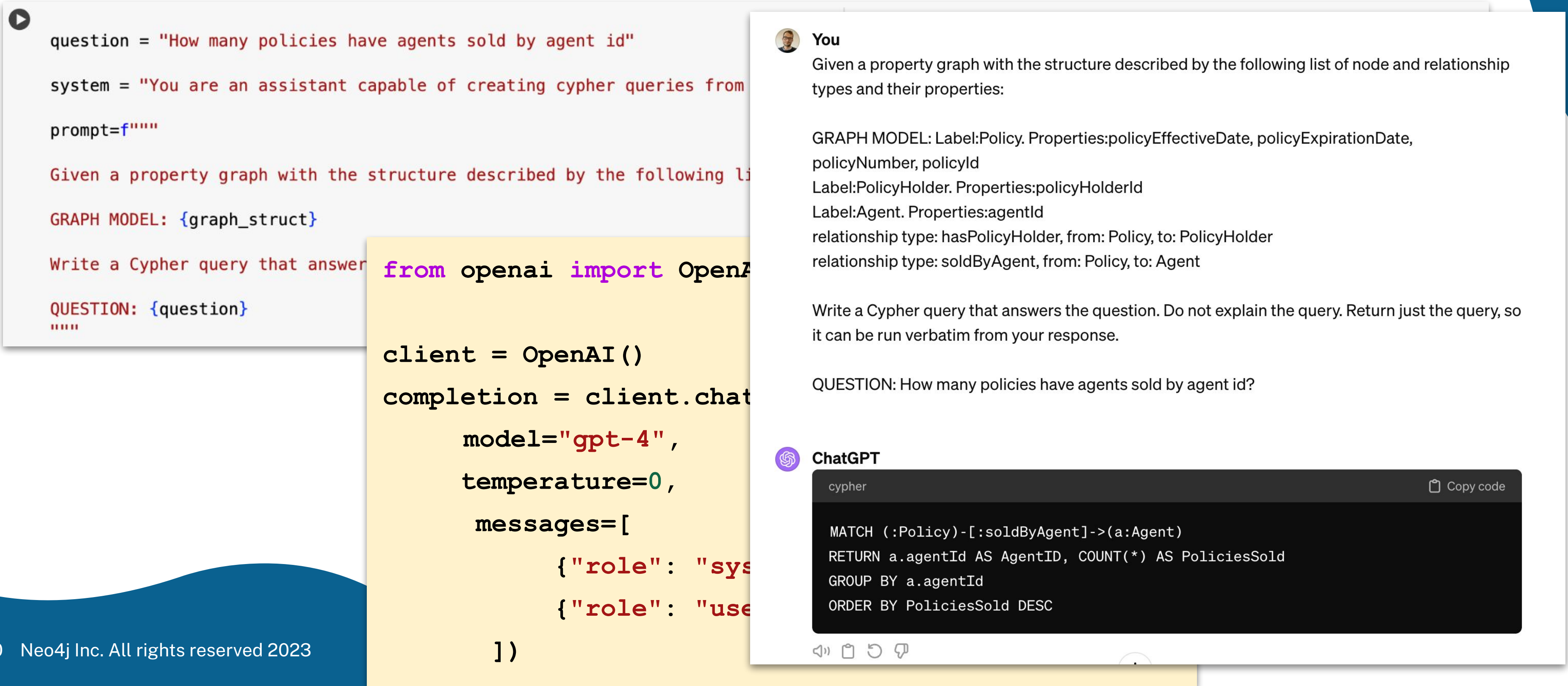
A Neo4j based version



1. Build the property graph building the semantic layer into the data
   a. Use the ontology as target model
   b. Data-Import tool (or code based alternatives) check GM#5
2. Get LLM to retrieve the ontology from the graph in Neo4j and generate the cypher query

https://github.com/jbarrasa/goingmeta

The completions API (or manually in ChatGPT)



```
question = "How many policies have agents sold by agent id"
system = "You are an assistant capable of creating cypher queries from
prompt=f"""
Given a property graph with the structure described by the following li
GRAPH MODEL: {graph_struct}
Write a Cypher query that answer
QUESTION: {question}
"""
```

```
from openai import OpenA

client = OpenAI()
completion = client.chat
      model="gpt-4",
      temperature=0,
       messages=[
             {"role": "sys
             {"role": "use
       ])
```

You

Given a property graph with the structure described by the following list of node and relationship types and their properties:

GRAPH MODEL: Label:Policy. Properties:policyEffectiveDate, policyExpirationDate, policyNumber, policyId
Label:PolicyHolder. Properties:policyHolderId
Label:Agent. Properties:agentId
relationship type: hasPolicyHolder, from: Policy, to: PolicyHolder
relationship type: soldByAgent, from: Policy, to: Agent

Write a Cypher query that answers the question. Do not explain the query. Return just the query, so it can be run verbatim from your response.

QUESTION: How many policies have agents sold by agent id?

ChatGPT

```
MATCH (:Policy)-[:soldByAgent]->(a:Agent)
RETURN a.agentId AS AgentID, COUNT(*) AS PoliciesSold
GROUP BY a.agentId
ORDER BY PoliciesSold DESC
```