An Automated Evaluation Framework for Graph Database Query Generation Leveraging Large Language Models

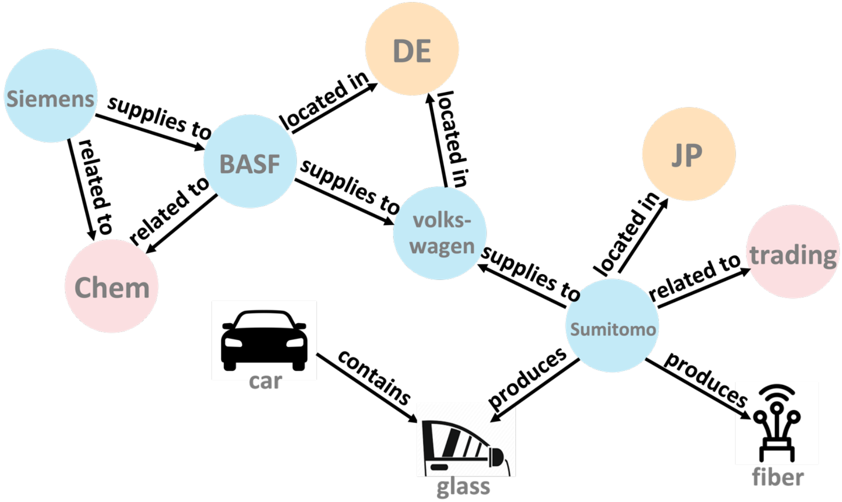
Bailan He, Yushan Liu, Marcel Hildebrandt, Zifeng Ding, Yaomengxi Han and Volker Tresp

Table of contents
Index / Agenda

Introduction

Terminologies

- **Knowledge Graphs**: a collection of triples $\mathcal{G} \subset \mathbf{E} \times \mathbf{R} \times \mathbf{E}$



(Siemens, supplies to, BASF),
(Sumitomo, located in, JP),
……

SIEMENS

Introduction

Terminologies

- **Query Generation:** given a user's natural language request $X = (x_1, x_2, \ldots, x_m)$, generate a corresponding query $Y = (y_1, y_2, \ldots, y_n)$, that can retrieve the answer the user wants from a database.

User:
How many companies are in Germany?

Target Query:
MATCH (n:Supplier) –[:LOCATED_IN] -> (:Country {name:"DE"})
RETURN count(n)

SIEMENS

Framework
Overview

- **Process I – Query Dataset Creation**

  A dataset consisting of diverse NL requests and queries.

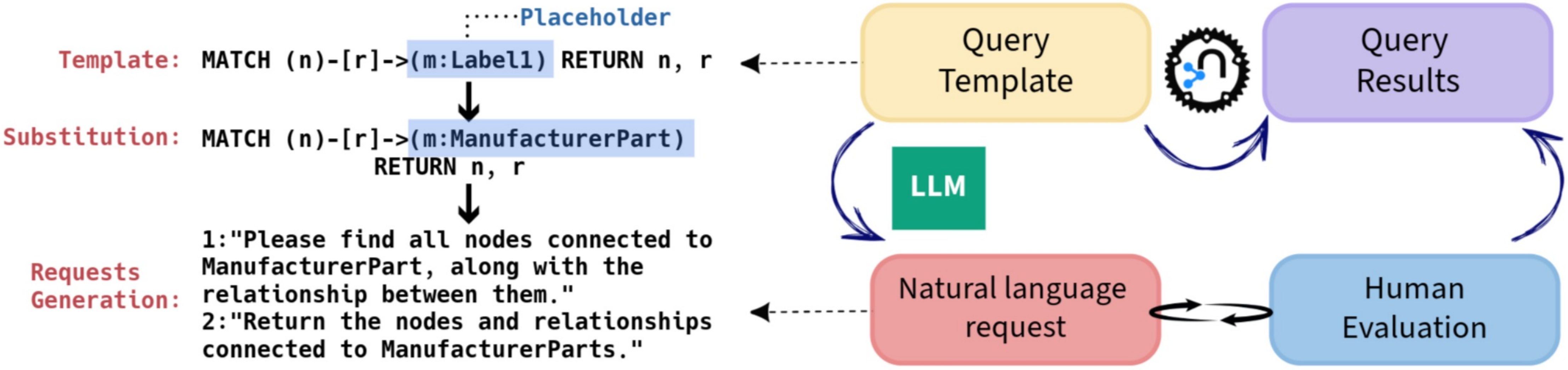
- **Process II – Query Generation**

  Evaluate the model performance of different prompts based on the generated evaluation dataset.

SIEMENS

Query Dataset Generation

Workflow



Initial Query Template -> Placeholder Substitution -> Requests Generation -> Human Evaluation

SIEMENS

Query Dataset Generation

Different Prompting

Simple prompt

````
I want you to act as a Neo4j specialist. I will supply you with a Cypher query that needs to
be explained in a simple manner, suitable for someone without any knowledge of databases.
Please write a user-friendly question for the given query in one sentence, ensuring it is
concise, easily understandable, and avoids technical jargon.
The question should not start the description with "This query..." or "The query..." and do
not mention the database or network.
````

+Schema prompt

````
The schema is delimited with triple dashes.
Schema: ---{schema}---
````

+In-context prompt

````
I will supply you with two examples of questions and answers. Each example contains a [Query]
and a [Question]. In the end you need to supply a question for the given query.
[Query]: ```MATCH (n)-[r]->(m:ManufacturerPart) RETURN n, r```
[Question]: What are the things connected to manufacturer, how are they related to each other?
[Query]: ```MATCH (n:Branch)-[r1]->(m:Branch) RETURN COUNT(r1)```
[Question]: How many connections exist between different branches?
````

Simple prompt

````
[Query]: ```{query}```.
[Question]:
````

Query Generation with LLMs

Different Prompting

Request: What countries are connected to branches, and what are the names of those branches?
Gold query: MATCH (n:Country)-[r1]->(m:Branch) RETURN n, collect(m.bp_name)

Simple Prompt	Prompt with Schema	In-context Prompt
Generate a Cypher query corresponding to the following verbalization delimited with triple backticks, the output should be a Cypher query and nothing else. [no prose]: [Question]: ```{requests}```	Simple Template + Given the Neo4j schema delimited with triple dashes and the property keys delimited with open angle brackets. Schema: ---{schema}--- Property keys: <property>	Schema Template + Several examples will be provided. Example: [Question]: ```What are the names of all the countries?``` [Query]: MATCH (n:Country) RETURN collect(n.name_de)
Output: MATCH (n:countries)-[r1]-> (m:branches) RETURN n, m.names	Output: MATCH (n:Country)-[r1]->(m:Branch) RETURN n, m.bp_name	Output: MATCH (n:Country)-[r1]->(m:Branch) RETURN n, collect (m.bp_name)

SIEMENS

Framework

Metrics

- **Execution Rate (ER)**

$$\mathrm{ER} = \frac{\text{Number of executable queries}}{\text{Total number of output queries}}$$

- **Gold Query Accuracy (GQA)**

$$\mathrm{GQA} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{BERTScore}(O_i, G_i)$$

- **Execution Accuracy (EA)**

$$\mathrm{EA} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{BERTScore}(R_i, R_{G_i})$$

SIEMENS

Experiment

Settings: Supply Chain Knowledge Graph

Entity type	# Nodes	Relation Type	# Edges
Supplier	61,234	supplies_to	138,197
Manufacturer Part	1,650	related_to	59,894
Company Part	1,295	belongs_to	56,663
Smelter	340	located_in	30,107
Substance	321	includes	10,088
Component	233	produces	7,831
Country	172	produced_in	4,381
Business Scope	32	same_as	1,847
		manufactured_by	1,564
		contains	764
		refines	340
Total	**65,277**	**Total**	**311,676**

The dataset [1] is constructed with internal information of the company Siemens.

In total, there are 16,910 tier-1 suppliers, 43,759 tier-2 suppliers, and 49,775 tier-3 suppliers of Siemens.

SIEMENS

Experiment

Settings: Supply Chain Query Dataset

Query Template Type	Number
Node Matching	10
Relationship Matching	10
Aggregation and Analysis	10
Combining Filters and Aggregation	10
Complex Queries	15
Traversal and Paths	5

The dataset includes 825 pairs of query-requests, with a value of 0.72 w.r.t Fleiss' kappa metric, indicating the soundness of the dataset.

SIEMENS

Experiment

Results: Query Generation Results

GPT-3.5				GPT-4			
Method	**GQA**	**ER**	**EA**	**Method**	**GQA**	**ER**	**EA**
simple	0.47	0.76	0.31	simple	0.55	0.81	0.29
schema	0.62	0.74	0.42	schema	0.71	0.89	0.43
ICL-1 shot	0.63	0.87	0.44	ICL-1 shot	0.69	0.89	0.47
ICL-3 shot	0.70	0.90	0.55	ICL-3 shot	0.73	0.91	0.52
ICL-5 shot	0.72	0.91	0.52	ICL-5 shot	0.75	0.93	0.53

Notation:

Simple denotes direct model instruction, **Schema** indicates prompting with schema, and **ICL-k shot** (in-context learning with k examples) involves instructing the model with in-context demonstrations.

- **GPT-4** outperforms GPT-3.5 across prompting all methods;
- Employing **schema** yields better results;
- **In-context learning** consistently outperforms using direct instructions.

SIEMENS

Conclusion

Findings and Limitations

- The proposed **automated QG evaluation framework** tackles domain-specific challenges (SCM).
- The framework involves both **dataset creation** and **model performance evaluation**.
- The work studies how different **prompting** and **models** affect LLMs’ performance of QG.

SIEMENS

Future Directions

- **Experimental Configuration**
  - Include other prominent LLMs like **Gemini** [1] and **Llama** [2]
- **Query Style**
  - Explore **multi-turn dialogue** in the workflow.