

Fraunhofer-Institut für Integrierte Schaltungen IIS

# Graf von Data – A Large Language Model Augmented Agent for KGQA over Integrated Data

Daniel Henselmann, Rene Dorsch, Andreas Harth

# Agenda

1. Introduction

- 2. Architecture
- 3. Summary



## **Introduction** State of the Art

Large Language Models (LLMs) have been used for a variety of approaches for KGQA



[1]: Thamer Mecharnia, Mathieu d'Aquin, Performance and Limitations of Fine-Tuned LLMs in SPARQL Query Generation https://aclanthology.org/2025.genaik-1.8.pdf

[2]: Patrick Lewis et al, Retrieval-Augmented Generation for Knowledge-Intensive NLP-Tasks https://dl.acm.org/doi/abs/10.5555/3495724.3496517

[3]: David Schmidt, Mohammad Elahi, Philipp Cimiano, Lexicalization Is All You Need: Examining the Impact of Lexical Knowledge in a Compositional QALD System https://link.springer.com/chapter/10.1007/978-3-031-77792-9\_7

[4]: Liu, S., Semnani, S.J., Triedman, H., Xu, J., Zhao, I.D., Lam, M.S.: SPINACH: SPARQL-Based Information Navigation for Challenging Real-World Questions. https://aclanthology.org/2024.findings-emnlp.938/



## **Introduction** State of the Art

• Large Language Models (LLMs) have been used for a variety of approaches for KGQA



- State of the Art: Assumes a known knowledge graph representation with customized tools
- > Impractical for integrated data



# Introduction

Graf von Data

- Requirements: A flexible and accessible approach for integrated • data
  - Agent-based
  - No knowledge about the knowledge graph  $\geq$
  - No specialized tools: Access only via SPARQL endpoint with  $\succ$ keyword search extension (e.g., [1, 2])
  - Plug-and-Play Solution  $\geq$





# Architecture Tools

### Search

- Goal: Find URIs for Keywords
- Input: Search term as a string
- Output: A list of <URI, similarity score</li>
   [1]>-tuples

Used think ^

<think> The user is asking for the telephone number of a person named Baldwin Dirksen. I need to find the URI for Baldwin Dirksen and then retrieve the telephone number associated with this person.

Used search ^

The search for Baldwin Dirksen returned the following matches.

- http://ld.company.org/prod-instances/empl-Baldwin.Dirksen%40company.org 7.858535
- http://ld.company.org/prod-instances/empl-Baldwin.Guenther%40company.org 3.7900822

#### Describe

Goal: Traverse knowledge graph Input: URI of an entity, property, or class Output: ~Neighborhood in RDF to the specified URI

#### Used think $\uparrow$

<think> The search returned two URIs, but the first one seems to be the most relevant as it has a<br/>higher score. I will use this URI to find the telephone number of Baldwin Dirksen. <act>describe("http://ld.company.org/prod-instances/empl-Baldwin.Dirksen%40company.org")</a>act>

Used describe  $\land$ 

#### The description of http://ld.company.org/prod-instances/empl-

Baldwin.Dirksen%40company.org is as follows. Triples with frequently occurring properties have been pruned.

Raw code

@prefix ns1: <http://ld.company.org/prod-vocab/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

<http://ld.company.org/prod-instances/hw-A391-9508821> ns1:hasProductManager<http://ld.company.org/prod-instances/empl-Baldwin.Dirksen%40company.org> .

<http://ld.company.org/prod-instances/hw-I409-8215134> ns1:hasProductManager<http://ld.company.org/prod-instances/empl-Baldwin.Dirksen%40company.org> .

#### Query

Goal: Evaluates SPARQL queries Input: SPARQL query Output: Serialized result set

#### Used think $\, \wedge \,$

<think> The description of the URI for Baldwin Dirksen includes the telephone number. I can nowconstruct a query to retrieve this information. <act>query("PREFIX ns1: <acth>http://</a>ld.company.org/prod-vocab/> SELECT ?telephone WHERE { <ath>http://</a>instances/empl-Baldwin.Dirksen%40company.org> ns1:phone ?telephone .}")</act>

Used query ^

ſ'n

The query PREFIX ns1: <http://ld.company.org/prod-vocab/> SELECT ?telephone WHERE { <http://ld.company.org/prod-instances/empl-Baldwin.Dirksen%40company.org> ns1:phone ?telephone . } returned the following solution.

[{'telephone': "+49-6200-33069465"'}]



[1]: Apache Jena: http://loopasam.github.io/jena-doc/documentation/query/text-query.html

# Architecture LLMs

### • Requirements:

- Reliable action calling
- Reliable agent loops
- Focus on < 100bn models

## • Used Large Language Models:

- Llama 3.3 70B Instruct [1]
- Qwen 2.5 72B Instruct [2]
- Observations:
  - <10 bn models do not understand the agent loop
  - > 100bn models perform in general better, but are more expensive

[1]: https://www.llama.com/docs/model-cards-and-prompt-formats/llama3\_3/

[2: https://graphdb.ontotext.com/documentation/11.0/full-text-search.html

[3]: https://informationisbeautiful.net/visualizations/the-rise-of-generative-ai-large-language-models-llms-like-chatgpt/ last accessed: 31.05.2025

Seite 7 01.06.2025 © Fraunhofer IIS



# Architecture Prompt

#### • Requirements:

- Reliable action calling
- Reliable agent loops

### • Prompt Structure:

- Task definition
- Loop definition using ReAct framework [1]
- Tool specification
- Ending conditions
- Stages
- Guidelines

[1]: Shunyu Yao et al, ReAct: Synergizing Reasoning and Acting in Language Models https://arxiv.org/abs/2210.03629



# Conclusion

- We developed Graf von Data
- An LLM-augmented agent for Text2SPARQL
- Plug-and-Play for integrated data



- **Further Work:** •
  - **COMPASS: A Process Mining-based** Methodology for Prompt Optimization of Large Language Model Agents

to be published at HybridAIMS CAISE Workshop 2025 as Short Paper



Impact of Knowledge Graph **Representations on Question Answering** with Language Models

to be published at HybridAIMS CAISE Workshop 2025 as Full Paper



