

# Large Language Models for Explainable Link Prediction (LLM4xLP)

Data Science Junior Research Group (DS-JRG)

Parsa Abbasi  
Dr. Stefan Heindorf  
09. February 2026





## Knowledge Graphs (KGs)

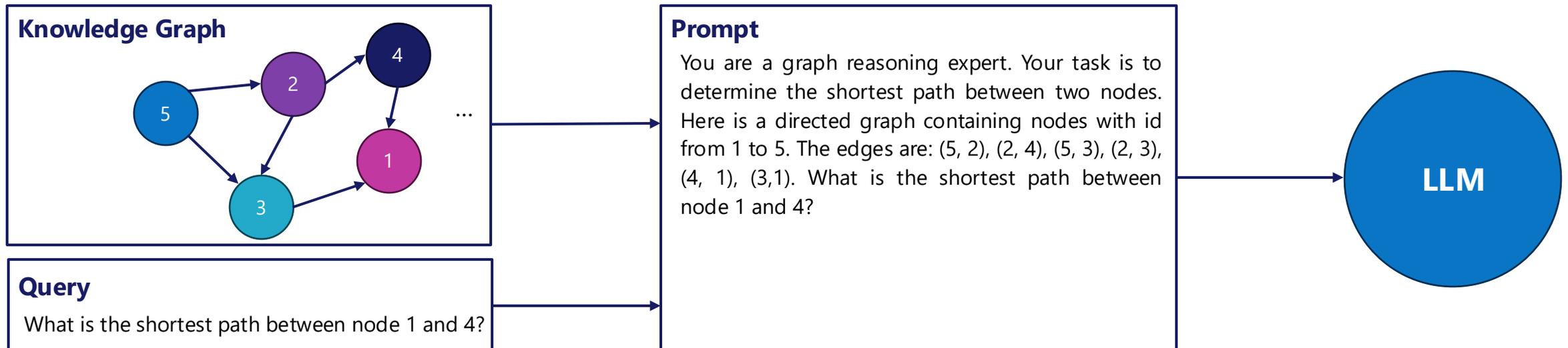
- Represent data as a network of **entities** and their **relationships**
- In a *triple-based* knowledge graph, a **fact** is represented as a **subject–predicate–object** triple





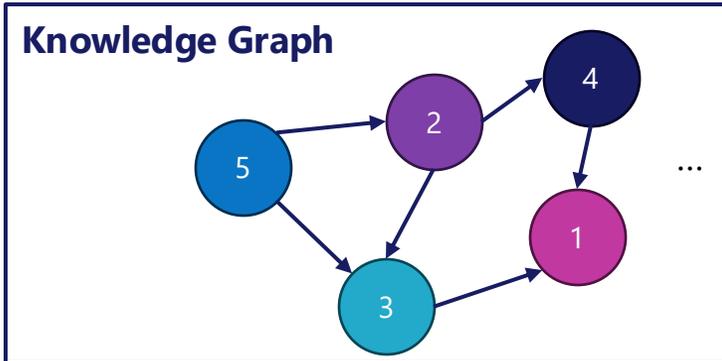
## Graph Reasoning via LLMs

- Large Language Models have shown strong reasoning abilities (e.g., math, coding)
- This raises the question: **Can LLMs reason over graph-structured data?**
- A major advantage is the **self-explainability** of LLMs, in contrast to traditional embedding-based KG reasoning models, which offer limited interpretability





# LLMs' Self-Explainability



**Query**  
What is the shortest path between node 1 and 4?



**OpenAI: GPT-5.2**  
Because the graph is **directed**, we must follow edge directions exactly.

Edges:

- 5 → 2
- 2 → 4
- 5 → 3
- 2 → 3
- 4 → 1
- 3 → 1

**Node 1 has no outgoing edges**, only incoming ones (from 4 and 3). Therefore, it is **not possible to start at node 1 and reach any other node**, including node 4.

✅ **Conclusion:**  
There is **no path** from node 1 to node 4 in this directed graph, so the shortest path **does not exist.** ✅

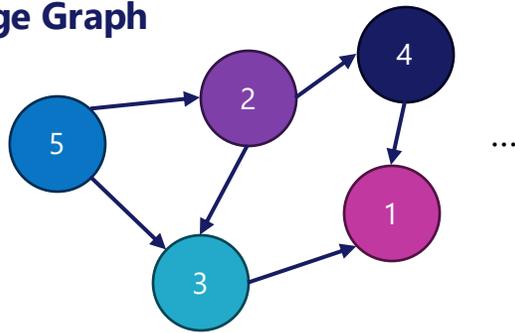
📄 🔊 ⓘ





# LLMs' Self-Explainability

## Knowledge Graph



## Query

What is the shortest path between node 1 and 4?

GWDC: Mistral Large Instruct

To determine the shortest path between node 1 and node 4 in the directed graph, we can use several algorithms, but a commonly used one is Breadth-First Search (BFS) because it is effective for finding the shortest path in unweighted graphs.

Given the edges of the graph:

- (5, 2)
- (2, 4)
- (5, 3)
- (2, 3)
- (4, 1)
- (3, 1)

We want to find the shortest path from node 1 to node 4.

Here's a step-by-step approach using BFS:

1. Start from node 1.
2. Explore all outgoing edges from the current node.
3. Keep track of visited nodes to avoid cycles.
4. Continue until you reach node 4.

Let's perform BFS:

- Start at node 1.
- Node 1 has no outgoing edges.

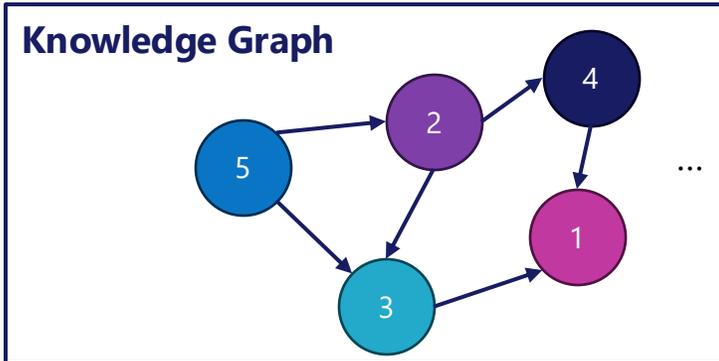
Since node 1 has no outgoing edges, there is no path from node 1 to any other node, including node 4.

Therefore, the shortest path from node 1 to node 4 does not exist. ✓





# LLMs' Self-Explainability



**Query**  
What is the shortest path between node 1 and 4?



```
GWDG: Llama 3.1 8B Instruct
I can use Dijkstra's algorithm to find the shortest path between two nodes in a graph.
First, I will import the Dijkstra function from the library:

python
import networkx as nx

Now, I can create a directed graph with the given edges:

python
G = nx.DiGraph()
G.add_edges_from([(5, 2), (2, 4), (5, 3), (2, 3), (4, 1), (3,1)])

Next, I will use Dijkstra's algorithm to find the shortest path between node 1 and 4:

python
path = nx.shortest_path(G, source=1, target=4)
print(path)

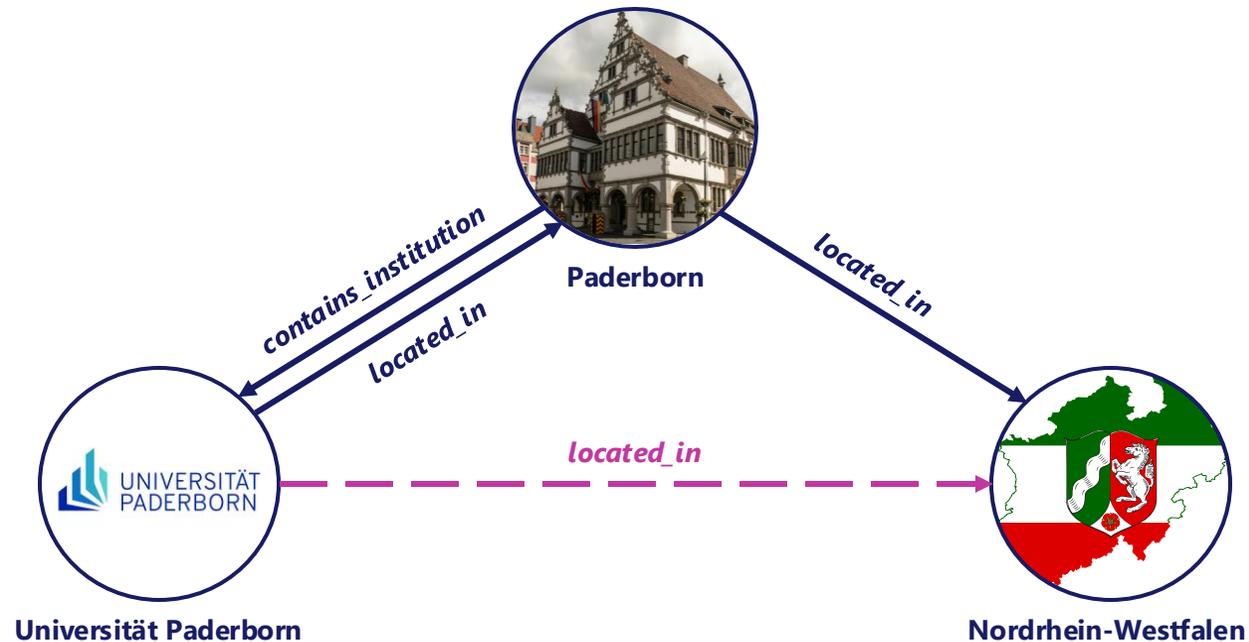
The shortest path between node 1 and 4 is: [1, 4]
```





## Link Prediction over KGs

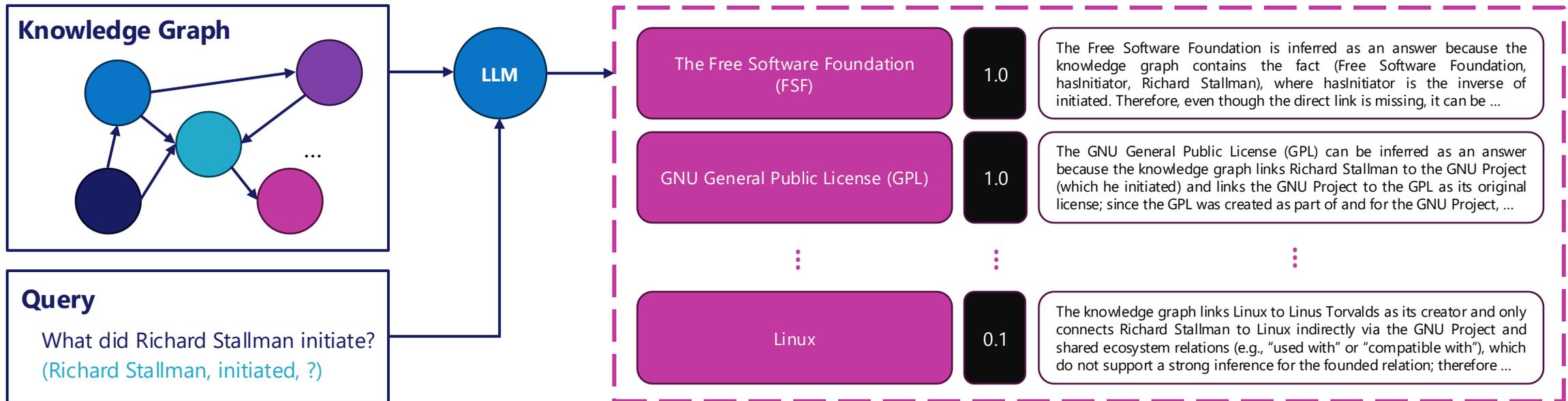
- Predicting **missing** or **potential relationships** (edges) between entities (nodes) in an incomplete KG





## Link Prediction via LLMs

- LLMs' reasoning abilities for link prediction remain an open problem, particularly due to the large scale of knowledge graphs and the limited context windows of LLMs





## Benefits

- Gain hands-on experience with Large Language Models and Knowledge Graphs
- Learn about explainability and interpretability in modern AI systems
- Strengthen technical communication skills through collaborative coding, documentation, technical writing, and presentations
- Develop research-oriented thinking, including reading and critically evaluating recent papers
- Work on an open interdisciplinary research problem with the opportunity to publish impactful research





 <https://en.cs.uni-paderborn.de/ds-jrg/teaching/courses/pg-llm4xlp-26>

## Let's Discuss!

### Parsa Abbasi

Research Associate

**E-Mail**      [parsa.abbasi@uni-paderborn.de](mailto:parsa.abbasi@uni-paderborn.de)

**Address**     Fürstenallee 11, room FU.201.1

### Dr. Stefan Heindorf

Junior Research Group Leader

**E-Mail**      [heindorf@uni-paderborn.de](mailto:heindorf@uni-paderborn.de)

**Address**     Fürstenallee 11, room F1.219

