Maastricht University

Department of Advanced Computing Sciences

Building and Mining Knowledge Graphs
Group 6

# Assignment 3 – Multi-Hop Reasoning System on a Knowledge Graph



| *Authors:* | *Student numbers:* |
|---|---|
| Aurélien Bertrand | i6256590 |
| Johannes Coppeneur | i6361640 |
| Tiago Ferreira | i6368709 |
| Teun Hermans | i6219876 |



March 27, 2024

# Contents

# 1 Introduction

In this assignment, our main objective was to develop a sophisticated system capable of seamlessly processing user input containing a comprehensive array of symptoms. Following this, the system would proceed to formulate and execute intricate queries against a robust medical Knowledge Graph (KG). The goal was to extract diseases associated with the input symptoms, taking into account relevant risk factors and associated conditions.

Furthermore, our system, driven by the Language Model (LLM), employed advanced multi-hop reasoning techniques. Leveraging the LLM's capabilities, the system generated SPARQL queries while considering the schema of the KG. These queries not only retrieved diseases linked to the input symptoms but also explored related risk factors and conditions.

Drawing upon the wealth of information gathered through these sophisticated algorithms, our system generated detailed and comprehensive explanations. These explanations not only outlined potential diagnoses but also illuminated the intricate connections between symptoms, diseases, and associated risk factors. Additionally, they provided insightful suggestions for the next steps or potential treatments, thereby offering invaluable guidance to users navigating complex medical scenarios.

# 2 Knowledge Graph selection

For this assignment, we decided to use a graph generated from the data available from Wikidata[1]. A query was written to retrieve information about drugs, which disease the cure and which possible side effects come with the drug. To make the graph a bit more extensive, we added whether diseases were genetically transferable and which complaints come with certain diseases. With this graph, we are looking to infer relationships between the causes of diseases and potential complaints that come with them. To show a simple example, Lung cancer would be caused by smoking, lung cancer also comes with complaints like shortness of breath. Here, a new link could be inferred from smoking to shortness of breath.

# 3 System design and Implementation

## 3.1 System design

The system we have devised operates in the following manner: upon receiving two input questions, it formulates SPARQL queries to address them and integrates the resulting data into multi-hop reasoning in order to discover relations/links between both questions.

[1] `https://www.wikidata.org/`

### 3.1.1 Query generation

This section details the process of SPARQL query generation using an LLM. When a user inputs a question they seek to answer using their KG, the system undergoes the following steps:

The initial step involves refining the input question to enhance clarity and precision. Subsequently, the question is dissected to discern the known and unknown pieces of information. For example, the question "What drug can be used to treat the disease vomiting?" is broken down into:

- **Known variables**: Disease → vomiting;
- **Target**: Drug to treat vomiting.

This breakdown assists the LLM in creating the SPARQL query effectively. Before this refinement, syntax errors were common, and occasionally, known facts were mistaken as unknowns. By delineating knowns and unknowns, the LLM gains a clearer understanding of which facts to utilize and which to retrieve through the query.

Given that the LLM lacks prior knowledge of the KG, a schema is essential. Since no suitable schema was available online for our KG, we developed our own. This schema provides the LLM with insights into the structure of triples, aiding it in selecting appropriate predicates.

Consequently, the LLM can generate a valid SPARQL query for most input questions, leveraging the provided schema for guidance.

### 3.1.2 Multi-hop reasoning

Our approach to multi-hop reasoning relies on a rule-based system to infer complex relationships within a knowledge graph. This approach leverages logical rules to navigate through the graph and deduce indirect relationships.

To traverse the knowledge graph and infer these relationships, we employ a Breadth-First Search (BFS) strategy. BFS systematically explores the graph, starting from a given entity and gradually expanding its search to neighbouring entities. This exploration proceeds level by level, ensuring that closer entities are explored before moving to more distant ones.

By utilizing BFS, our system efficiently explores the graph structure, uncovering connections that may span multiple hops or intermediary entities. Each traversal step considers all possible paths from the current entity, allowing us to uncover complex relationships that may not be immediately apparent.

### 3.2 Data processing

Considering our research interests, we opted to use a specific subset of Wikidata, focusing on drug-disease data. A significant drawback of the Wikidata KG lies in its unintuitive URIs, which predominantly employ identifiers. Consequently, we resolved to construct our own vocabulary for the KG. This decision proved invaluable for our approach, as it facilitated enhanced comprehension for the LLM regarding the properties to incorporate within the queries.

### 3.3 Evaluation metrics

For this assignment, we use accuracy and human evaluation (i.e., we assess the quality and relevance of the generated queries compared to the original questions). In order to employ this, we:

- Prepare a dataset with annotated questions and corresponding ground truth SPARQL queries.
- Use the conversational system to generate SPARQL queries for the questions in the dataset.
- Evaluate the generated queries against the ground truth using the defined metrics.

## 4 Theoretical analysis

Given the LLM's proficiency in generating quality SPARQL queries, even in rare cases, it's poised to detect them effectively. Nonetheless, the considerable computational resources needed for executing complex queries and providing detailed explanations might impose constraints on the system's efficiency, particularly in real-time or resource-limited settings.

### 4.1 Strengths

LLMs exhibit exceptional efficacy across various scenarios. Leveraging this capability enhances the potency and versatility of our approach, making it applicable to a wide range of questions.

### 4.2 Weaknesses

A significant drawback of our methodology is its heavy reliance on the LLM. We have minimal control over its functioning, which poses a remarkable limitation.

Furthermore, during our system's experimentation phase, it became evident that the LLM's comprehension of our directives was not always clear. Additionally, it occasionally struggled to discern relationships within the graph.

### 4.3 Zero-shot Learning

When confronted with unseen or rare associations, the outcome hinges on both the quality of the data and the effectiveness of the inherent LLM.

## 5 Results and Discussion

Due to challenges in sourcing an appropriate KG, a significant portion of our time was dedicated to searching for a suitable one for this project. Consequently, implementing the multi-hop reasoning algorithm was delayed, leading to a loss of valuable time. While we have a basic BFS approach operational, integrating it with the rest of the system remains unresolved.

On a brighter note, the SPARQL query generation process yields satisfactory results. It successfully constructs valid queries in a timely manner in most instances. However, there are occasional failures in query creation, where either a wrong query is generated or unnecessary relations are included, resulting in prolonged query execution times.

Despite these challenges, we believe our approach is inherently innovative. With further refinement of the multi-hop reasoning component, we anticipate that the system's potential can be greatly enhanced.

## 6 Work Distribution

- **Aurélien Bertrand:** SPARQL query generation, report
- **Johannes Coppeneur:** report, multi-hop
- **Tiago Ferreira:** report, rule-based approach
- **Teun Hermans:** report, graph generation, combining the systems

## 7 Disclosure and Reflection

Large Language Models (LLMs) were heavily used in this assignment since the methodology revolves around it. The LLM was very capable of producing SPARQL queries with enough context. Additionally, code completion tools like GitHub Copilot expedited the task completion process, enabling faster progress.

During this assignment, we had a lot of trouble with finding a usable knowledge graph to our interest. Datasets were mostly too big, making it such that generating would already take up way too much time, let alone the multi-hopping through it. It took a lot of effort and time before settling with the Wikidata graph that we are using now, the problem with this graph is that it lacks complication and is dominated by unintuitive URIs.

# A Appendix

## A.1 Practical implementation

The code used for this assignment can be found in our public GitHub repository[2]. The file `assignment-3_multi-hop-reasoning.ipynb` contains the code to run our system, including documentation and dependencies to be installed.

The `src/other` comprises other files which were eventually not used in the final submission.

## A.2 Dataset

For this assignment, we decided to build a KG using a subset of the complete Wikidata KG. The KG as well as its schema can be found in the `data` folder.

[2] `https://github.com/Aurelien-Bertrand/bmkg-assignment3`