BMKG Assignment 3

Emery Karambiri, Tarik El-Khoury, Damian Poştaru, Alexandros Triantafyllou

March 17, 2024

Contents

1 Introduction

In the dynamic world of football, the ability to extract detailed insights and analyze complex player trajectories, team compositions, and tournament outcomes is invaluable. The rich history of the sport, coupled with its global appeal, presents unique challenges and opportunities for data analysis.

2 Knowledge Graph

For our project, we have selected Wikidata as the primary Knowledge Graph (KG) due to its comprehensive and well-structured dataset on various subjects, with a particular focus on football. Wikidata's extensive coverage includes detailed information about athletes, teams, tournaments, and specific matches, making it an invaluable resource for sports analytics and research.

2.1 Data Acquisition

We sourced our data directly from Wikidata through the SPARQL endpoint. Using a series of eight SPARQL queries, we were able to extract detailed information about individual players, tournaments, teams, and specific matches.

First, we got different football players. They are the main foundation of our KG. However, we had to limit how many we requested given that there are a lot of players and getting all of them would require a lot of computational power. Once we have the players, we get all the teams they are, or once were, a part of. Given this one-to-many relationship, we also attribute a start and end date to each of them.

For all the relevant teams we found, we get team information such as which league they're part of, the country they are based in, and the tournaments they've won.

Following the teams, we get information about the tournaments. This includes the number of participants, the start and end dates, where it took place, who the organizer is and what teams participated in it.

To get more detailed information about players, we also fetched data about the matches they participated in. For each match, we get the organizer, its location, the date, the winner, the tournament it was part of, the participants (teams and players), and the how many goals were scored by each player (only if they scored).

These queries allowed us to target precisely the data relevant to our project while ensuring accuracy and consistency.

2.2 Structure

Our Knowledge Graph is structured using several key namespaces, instead of only the wikidate one, to make the graph human-readable and to facilitate easier data manipulation. These additionally namespaces are Schema, FOAF and a custom namespace. We utilized these namespaces to define the relations and properties of entities within our graph. A more detailed view of the relations and properties can be seen in Appendix A.

2.3 Justification

The decision to use Wikidata as our KG foundation is due to the vastness of Wikidata's structured data format and its extensive coverage of football-related entities. This wide range of data enables complex multi-hop queries.

Additionally, by directly querying Wikidata, we ensure our data is up-to-date and accurate. The structured nature of the data allows for precise queries, crucial for answering complex questions such as identifying Brazilian players who have played in the Premier League and won trophies, or players who played in clubs in at least two different countries who have won the UEFA Champions League.

3 Multi-hop Question Answering System Implementation

3.1 Embedding and Neural Network based

Our multi-hop reasoning system is embedding and LLM based. It operates by embedding segments of the Knowledge Graph, following a method similar to the one explored in our RAG lab. Specifically, we divide the graph data into 1000-character chunks, ensuring a 200-character overlap between adjacent segments. These segments are then embedded using the 'bge-small-en-v1.5' model, made by the Beijing Academy of Artificial Intelligence.

Method When a question is asked, the system first reformulates it for clarity and specificity. It then selects the 20 most relevant graph segments based on the embedded representations. We decided to use 20 due to the additional information required to make the complex connections. This additional context, alongside the reformulated question, is provided to a Large Language Model (using the OpenAI key given) who is tasked with generating an answer solely based on the given context.

Reformulating Question A lot of effort went into trying to optimize the reformulating of the question, either by trying to make it explicitly mention types and relations, or by trying to make it aware of the structure of the knowledge graph. Surprisingly, simply reformulating the question to be clearer and more explicit performed better. This is likely due to the fact that the embeddings don't have direct knowledge of the knowledge graph structure and mainly try to understand the content.

Data Processing Given the nature of the system and the pre-processing done during the graph's construction, missing and inconsistent values are relatively low concerns. The embedding and selection process ensures that only the most relevant and accurate data segments are utilized for answering queries, minimizing the impact of any irregularities in the data.

Explainability Embeddings and neural networks are inherently difficult to explain at the most basic level. However, we can understand the answer given if we look at the context, the 20 most similar embeddings to the question according to the model used to embed them. This gives us a look inside the rationale of the answer, however it does not tell us why a specific embedding is considered to be closely related to the given question.

3.2 Evaluation Metrics

There are a few metrics that we planned on using in order to evaluate the Multi-Hop answering questions. Firstly, we need to measure the "accuracy" of the response, it is important that when asked for Brazilians football players, the response should contain only Brazilian players. And secondly, we will also use "precision" in order to measure our response, we need our model to return the Brazilian Football players and is able to filter out any irrelevant elements.

Due to difficulties, we could not implement these programatically.

However, we did do some hollistic tests. These included asking questions, and checking how they match with a SPARQL query we wrote.

Question 1: How many goals were scored by each argentinian player, individually? For this question, the multi-hop system was not able to give an answer as can be seen in fig.1. This was unexpected as goal information is available in the graph. The expected answer can be seen in table 1.

```
Reformulated question:  Can you provide a breakdown of the number of goals scored by
each individual Argentinian player?

No, this information is not provided in the given context.
```

Figure 1: The multi-hop system's answer to question 1.

Table 1: Players and Number of Goals

Player	Goals
Sergio Agüero	1
Javier Zanetti	1
Esteban Cambiasso	1
Mario Kempes	3
Pablo Aimar	1
Roberto Ayala	1
David Trezeguet	11
Maxi Rodriguez	2
Mauro Camoranesi	2
Omar Sívori	1

Question 2: List goalkeepers who have played both in Spain and in Italy This question is relatively simple to answer with a SPARQL query however, it is difficult from an embedding and semantic standpoint. The embedding would have to semantically know, on top of the directly available information (the match, the country it took place in, the player that participated in), that the player is a goalkeeper. The answer given, shown in fig. 2, reflects this difficulty. No answer was given by the system even though there were 5 solutions to this question according to the SPARQL query: Andoni Zubizarreta, Oleksandr Shovkovskyi, Peter Schmeichel, Dino Zoff, and Toni Schumacher.

```
Reformulated question:  Which goalkeepers have played for teams in both Spain and
Italy?

    Answer: I cannot answer this question with the given information.
```

Figure 2: The multi-hop system's answer to question 3.

Overall, it seems that the embedding-based multi-hop system was not the correct choice for this graph structure. The number of hops required to make certain inferences are too numerous which makes the embeddings not truly capture the deeper semantics of a triple.

4 Challenges and Solutions

There are a few challenges with our implementation, notably the certainty in the data quality given the provenance from Wikidata, the scalability and its ability to handle unknown and difficult to parse questions complex questions, especially questions containing more than one "and".

4.1 Data Quality and consistency

Ensuring the quality and consistency of data imported from Wikidata posed a challenge. Given the open nature of Wikidata, variations in data entry and occasional inaccuracies required us to implement data cleaning and validation mechanisms.

We did this with checks during querying of the data from the Wikidata end point. We had assertion checks to identify and rectify common inconsistencies, such as varying formats for dates, missing or invalid information.

We kept the same Wikidata URIs throughout the graph to ensure consistency in the entries. For the triples that we had to created URIs for, we made a string with the entities that make the triple unique and applied sha256 hashing to it.

4.2 Scalability

Another significant challenge faced in implementing our knowledge graph-based system was ensuring scalability. As the volume of data within the football domain is vast and continuously expanding, designing a system capable of efficiently processing large datasets without compromising response times or accuracy is important. We were not able to solve this challenge given that it is mainly a computational issue.

4.3 Complex Questions

By limiting the answers of the system to only consider the context, it is not able to make highly complex connections such as those present in the graph. For example, to determine if a player won a tournament, we need to look at all tournaments that all the teams that the player played for won. And then filter for teams that won during the tenure of the player. This alone is already difficult to say verbally. This is one of the reasons we decided to use embeddings. Through embeddings, we thought that it might be possible to condense all this information, however, looking at the results it was not successful.

4.4 Zero-shot Questions

Given the use of LLMs to parse and answer the questions, unseen or unanticipated questions are not necessarily an issue for this system.

5 Conclusion

In conclusion, the main challenge of this project was getting a graph that was reasonably parse-able. This would simplify the creation of SPARQL queries, but could also considerably enhance the semantics captured by the embeddings.

Looking back on the implementation and the results, a couple of things could have been done better. The construction of the graph itself could have been made clearer by adding more triples to make certain relations explicit.

In regards to the multi-hop system, a rule-based system would have possibly performed better. The relations between entities would have been hard coded and it would not need to understand the them. The main drawback and limitation of a rule-based systems is the necessity of parsing natural language into a rule-like structure. This was one of the reasons we decided on an embedding-based approach.

To explore the implementation further, the code repository can be found at https://github.com/EmeryK-1/BMKG-Assignment-3.

A Skeleton of the Knowledge Graph in Turtle Format

```
@prefix wd: <http://www.wikidata.org/entity/> .
@prefix schema: <http://schema.org/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix ex: <http://example.org/> .

<player_uri>
    a schema:Person ;
    schema:name "Player Name" ;
    schema:nationality <country_uri> ;
    ex:position <position_uri>;
    schema:memberOf <team_uri> ;
    ex:hasTenure <tenure_uri> . # Multiple tenures

<position_uri>
    a ex:PlayerPosition ;
    schema:name "Position Name" .

<country_uri>
    a schema:Country ;
    schema:name "Country Name" .

<team_uri>
    a schema:SportsTeam ;
    schema:name "Team Name" ;
    schema:location <country_uri> ;
    schema:partOf <league_uri> .

<tenure_uri>
    a schema:Role ;
    ex:atTeam <team_uri> ;
    schema:startDate "YYYY-MM-DD"^^xsd:date ;
    schema:endDate "YYYY-MM-DD"^^xsd:date. # Optional

<league_uri>
    a schema:SportsOrganization ;
    schema:name "League Name" .

<tournament_uri>
    a schema:SportsSeason ;
    schema:name "Tournament Name" ;
    schema:numParticipants  X^^xsd:integer;
    schema:location <country_uri> ;
    schema:winner <team_uri> ;
    schema:partOf <league_uri> ;
    schema:organizer <organizer_uri> ;
    schema:startDate "YYYY-MM-DD"^^xsd:date ;
    schema:endDate "YYYY-MM-DD"^^xsd:date ;
    schema:participant <team_uri> . # Multiple participants

<organizer_uri>
    a schema:Organization ;
    schema:name "Organizer Name" .

<match_uri>
    a ex:Match ;
    schema:name "Match Name" ;
```

```
    schema:organizer <organizer_uri> ;
    schema:location <country_uri> ;
    schema:startDate "YYYY-MM-DD"^^xsd:date ;
    schema:winner <team_uri> ;
    schema:partOf <tournament_uri> ;
    schema:participant <team_uri> ; # Multiple participants
    schema:participant <player_uri> . # Multiple participants

<goals_uri>
    a ex:Score ;
    ex:player <player_uri> ;
    ex:goals X^^xsd:integer;
    ex:match <match_uri> .
```

B Challenges and Reflection

This project was full of challenges, however there were two main challenges. The initial challenge was about getting the data and the way to store it. Given the intricacies of the data we worked with, it was not easy to simply connect everything together. Some links required a lot of context, such as a player being part of a team requires not only the player and the team but also a start and end date. Additionally, given the vast amount of data available, we were not able to have a complete dataset.

As a consequence of this complexity, the next challenge arose. Making SPARQL queries to query the graph was difficult and required paying careful attention to many different parts and details. This difficulty also made it hard to evaluate the answer given by the multi-hop system as we had nothing to check it against.

C Work Distribution

Emery worked on the structure of the graph and getting the data from the Wikidata SPARQL endpoint. He also worked on the multi-hop system presented, and the report. Alexandros worked on making SPARQL queries to answer multi-hop questions.

D LLM Disclosure and Reflection

During this project, multiple AI products were used:

- ChatGPT 3.5 was used to help with writing this paper. Mainly to reformulate when a sentence was difficult to write.
- GitHub CoPilot was used for code generation, especially the code used to make the graph as a lot of it was repetitive.

Using LLMs increased productivity tremendously, especially when it comes to writing repetitive code.