Soccer-GraphRAG: Applications of GraphRAG in Soccer

Zahra Sepasdar[1,2][0000−0003−2535−6508], Sushant Gautam[1,3][0000−0001−9232−2661], Cise Midoglu[1][0000−0003−0991−4418], Michael A. Riegler[1,3][0000−0002−3153−2064], and Pål Halvorsen[1,2,3][0000−0003−2073−7029]

[1] SimulaMet, Norway
[2] Forzasys, Norway
[3] OsloMet, Norway

Abstract. In the realm of soccer analytics, the need for efficient and accurate information retrieval is crucial. In this paper, we introduce Soccer-GraphRAG, a framework designed to facilitate the retrieval of soccer-related information through natural language queries. This system leverages knowledge graphs, created from the recently released SoccerNet-Echoes dataset which includes transcriptions of soccer game audio commentaries. Soccer-GraphRAG aims to streamline the retrieval, access, and analysis of soccer data, providing insights with precision and contextual relevance. This framework is ideally suited for analyzing player performance, as well as for engaging in question answering (Q&A) and summarizing tasks.

Keywords: Association Football · Knowledge Graphs · GraphRAG · Automatic Speech Recognition (ASR) · Large Language Models (LLM)

1 Introduction

In recent years, Large Language Models (LLMs) have made significant strides in understanding and generating human-like text [1,14]. Despite these advances, LLMs face critical limitations that hinder their reliability and applicability in practical scenarios. One major issue is the phenomenon known as 'hallucinations', where models generate plausible but factually incorrect or nonsensical information [1,3,4]. Additionally, LLMs are constrained by the static nature of their training data, limiting their ability to incorporate the most recent information or answer queries outside their training corpus [3]. To address these limitations, the Retrieval Augmented Generation (RAG) approach offers a compelling solution. By integrating external data dynamically at query time, RAG enhances the ability of LLMs to produce more accurate and contextually relevant responses [3,9]. However, traditional RAG methods relying on vector search cannot capture the full complexity of information [7]. Incorporating a dynamic knowledge base with RAG not only improves the quality of the answers provided by LLMs, but also significantly expands their knowledge base, allowing

them to reference up-to-date and specific information that is not available in their initial training set [11]. GraphRAG is a potent enhancement to traditional RAG which is based on vector search [15]. This approach utilizes the organized structure of graph databases, where data is categorized into nodes and interconnected relationships, to enrich the quality of information retrieval [10]. By doing so, GraphRAG significantly enhances the depth and contextuality of the information gathered, providing a more nuanced and comprehensive understanding compared to standard methods [12,13].

Knowledge graphs, with their ability to represent entities (like players, teams, tournaments) and their relationships (team memberships, competition history) [2], offer a promising solution for efficiently retrieving information from sports datasets. This structured format allows for more nuanced queries and facilitates the retrieval of comprehensive information compared to traditional methods that rely on keyword matching in unstructured data [8]. Despite their potential, existing approaches for sports data retrieval might struggle with the sheer volume and complexity of sports data. Additionally, capturing the rich context and relationships within sports datasets can be challenging with traditional methods.

Our research aims to tackle the complexities of extracting information from extensive sports datasets using the power of knowledge graphs. We propose Soccer-GraphRAG, a framework designed for retrieving information from soccer knowledge graphs through natural language queries. Soccer-GraphRAG utilizes knowledge graphs created from textual soccer datasets, such as the recently released SoccerNet-Echoes dataset [5] which includes automated transcriptions of audio commentaries from soccer game broadcasts. By implementing an approach based on knowledge graphs, we enhance user interaction and improve the accessibility of sports datasets. The contributions of our work are the following:

- Soccer-GraphRAG is specifically designed to leverage Automatic Speech Recognition (ASR) data, which has traditionally been challenging due to its unstructured nature and the presence of transcription errors. Our approach enhances the accuracy and utility of information extracted from ASR sources by effectively integrating these data into our knowledge graph.
- Our research addresses the issue of hallucinations in ASR outputs where erroneous or misleading information might be generated. Through the structured organization of data in the knowledge graph and the contextual checks provided by the Soccer-GraphRAG framework, we minimize these inaccuracies.
- Our Soccer-GraphRAG framework is specifically designed for the soccer domain, enabling precise information retrieval from extensive sports datasets through natural language queries.

The rest of the article is organized as follows: Section 2 details our Soccer-GraphRAG method and strategy. Section 3 describes the experimental setup, including the source of the dataset, pre-processing data phase, and knowledge graph construction. Section 4 presents the preliminary results. Finally, Section 5 concludes the paper by summarizing the main outcomes and suggesting potential directions for future research.

2 Methodology

This section outlines the methodology of our framework for retrieving information from a knowledge graph in response to user queries. When a user asks a question about soccer, the system leverages an LLM to create an appropriate Cypher query. This Cypher query is designed to extract relevant data from the knowledge graph, which is constructed from our dataset to provide accurate and detailed information. Once the Cypher query is created, the smart search tool utilizes it to navigate through the knowledge graph and retrieve the relevant information. The retrieved information, along with the user's original question, is then fed back into the LLM. The LLM processes this input to generate a comprehensive response. This complete process, from query generation to the final response, is illustrated in Figure 1.

The soccer knowledge graph illustrated in Figure 1 is constructed through a process that begins with a pre-processing phase. In this initial stage, entities and their interconnections are identified and extracted from the data. These entities and relationships are then used to create nodes and edges of the soccer knowledge graph. This graph is then stored in the Neo4j graph database. For this research, we employed the Neo4j Python driver to establish and manage connections with the Neo4j database, enabling efficient query execution and manipulation of graph data. This integration is part of our Python-based analytical framework.

In summary, our system operates through the following steps:

- **Data Pre-processing:** Entities and relations are extracted from our dataset.
- **Graph Construction:** The pre-processed data is used to structure soccer knowledge graphs.
- **Query Translation:** User query is translated into Cypher queries.
- **Information Retrieval:** Cypher queries retrieve relevant soccer graph data from Neo4j.
- **Answer Generation:** Based on retrieved graph data, LLM generates a response.

This methodology integrates advanced LLMs and graph technology to create a dynamic and responsive system for answering user queries about soccer. By utilizing Cypher queries to interact with a well-structured knowledge graph, and leveraging the processing power of LLMs for intelligent response generation, this framework ensures that users receive accurate, relevant, and detailed answers.

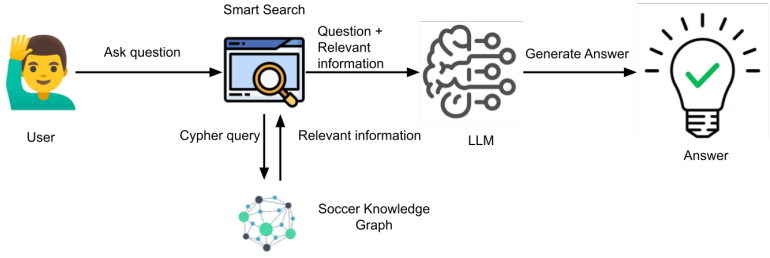


Fig. 1: Soccer-GraphRAG framework overview.

Listing 1.1: ASR data structure.

```
"segments": {
    "segment index": [
        "start time": < >
        "end time":   < >
        "text":       < >
                      ],
    ...
            }
```

3 Experimental Setup

3.1 Dataset

SoccerNet [6] is an expansive dataset designed for the analysis and understanding of soccer videos, collected from 550 soccer games from 6 different European leagues and 4 different seasons, narrated live by commentators speaking one of 10 different languages. Gautam et al. [5] have extended this dataset with audio commentary transcriptions using ASR, and have released the transcriptions for each game half as an open dataset called SoccerNet-Echoes. Our data source is ASR dataset from SoccerNet-Echoes dataset. The data in ASR has the structure as described in Listing 1.1.

Figure 2 shows a sample of ASR data for an English Premier League (EPL) game played between Chelsea and Burnley on February 21st, 2015 [5]. As illustrated in Figure 2 and Listing 1.1, each segment in an ASR file consists of a start timestamp, end timestamp, and corresponding text.

3.2 Data Pre-processing

The aim of the pre-processing phase is to extract 3-tuples from the 'text' in Listing 1.1, in the format of `(entity1, relation, entity2)`. These 3-tuples are used for the construction of a graph in Section 3.3

To extract 3-tuples, we create two different lists: one is used to extract entity1 and entity2, and the other is used for the relation. For entity1 and entity2, we

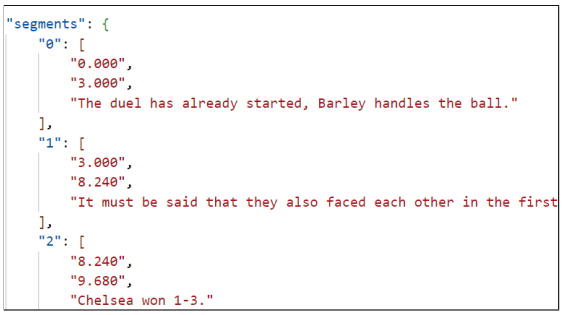


Fig. 2: Sample ASR file for an English Premier League (EPL) game played between Chelsea and Burnley on February 21st, 2015.

create a list containing the names of players, teams, referees, and events in a soccer game, such as goals, yellow cards, corners, etc. To extract the relation elements, we create a list containing the different types of actions in a soccer game such as score, won, etc. Subsequently, we utilize these lists and employ GPT-3 to extract 3-tuples from 'text'. The extracted 3-tuples are collected in an array named 'entities'. Now, by adding the timestamp and game half, we create a new ASR dataset, which has the structure as shown in Listing 1.2. Figure 3 presents the corresponding 'new' ASR file for the original file shown in Figure 2.

Listing 1.2: New ASR Data Structure.

```
"time": [
    "start time",
    "end time"
    ]
"description": <"text">
"half": " "
"entities": {(entity1, relation, entity2)}
```

3.3 Graph Construction

For each game in our dataset, we create one graph. Below, we outline the detailed structure of these graphs.

Game Node & Team Node For each game, we instantiate a node in Neo4j labelled `Game` with these attributes: `away_team, home_team, awaya_score, home_score, coach_home_team, coach_away_team, date, venue, referee, winner`. Each team is represented by a single node with label `Team` without any attributes.


```
{
    "time": [
        "0.000",
        "3.000"
    ],
    "description": "The duel has already started, Barley handles the ball.",
    "half": "1",
    "entities": "{(BARLEY, handles, ball)}"
},
{
    "time": [
        "3.000",
        "8.240"
    ],
    "description": "It must be said that they also faced each other in the first
    "half": "1",
    "entities": ""
},
{
    "time": [
        "8.240",
        "9.680"
    ],
    "description": "Chelsea won 1-3.",
    "half": "1",
    "entities": "{(Chelsea, won, 1-3)}"
},
```


Fig. 3: Sample 'new' ASR file, updated from the original file depicted in Figure 2.

Game & Team Nodes Connections We establish connections between each team and the corresponding games using an edge labelled `PARTICIPATED_IN`. Additionally, teams are connected to their respective games with edges labelled `HOME_TEAM` or `AWAY_TEAM`, indicating the home or away status of the team. If the game result is not a draw, two additional edges, `WINNER` and `LOSER`, are created to link the game node with the winning and losing teams, respectively. Figure 4 demonstrates the connections between game and team nodes for an English Premier League (EPL) match between Crystal Palace and Arsenal on February 21st, 2015.

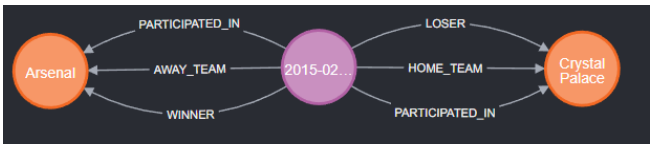


Fig. 4: Sample Game and Team nodes connections.

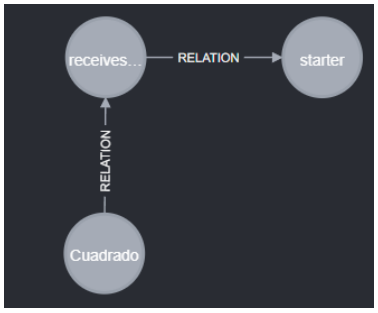
ASR Nodes As discussed in the previous section, the new ASR dataset is utilized to construct the graph structure. To elucidate the methodology, we provide a detailed explanation using the example depicted in Figure 5.


```
{
    "time": [
        "44.440",
        "46.800"
    ],
    "description": "Cuadrado who once again receives an opportunity as a starter.",
    "half": "1",
    "entities": "{(Cuadrado, receives_opportunity, starter)}"
},
```

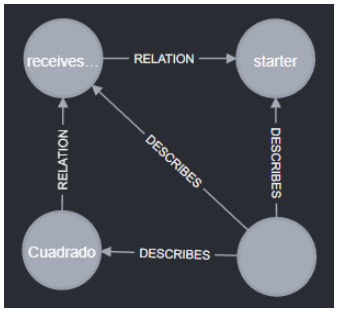

Fig. 5: An example of ASR Data.

For the text in 'description', we create one node labelled `Description`, which includes the attributes `time` and `half`. Each element in 'entities' is represented as a triplet `(entity1, relation, entity2)`. For each triplet, we generate three nodes: two nodes labelled `Entity` corresponding to `entity1` and `entity2`, and one node labelled `RELATION` corresponding to the `relation` component.

Entity & Relation Nodes Connections We connect the `entity1` node to the `relation` node with an edge labelled `RELATION`, and similarly, connect the `relation` node to the `entity2` node with another `RELATION` edge. In Figure 6a, the connections in the triple (`Cuadrado`, `receives_opportunity`, `starter`) are shown.



(a) Connections between Entity and Relation nodes.



(b) Connections between Entity, Relation, Description nodes.

Fig. 6: Visual representations of node connections.

The `Description` node is connected to these three nodes via an edge labelled `DESCRIBES`. Figure 6b illustrates this connection. Each `Description`, `Relation`, and `Entity` node is further connected to the corresponding game node via an edge labelled `BELONGS_TO`. Figure 7 demonstrates the connection between the text and the corresponding game, relation, and entity nodes.

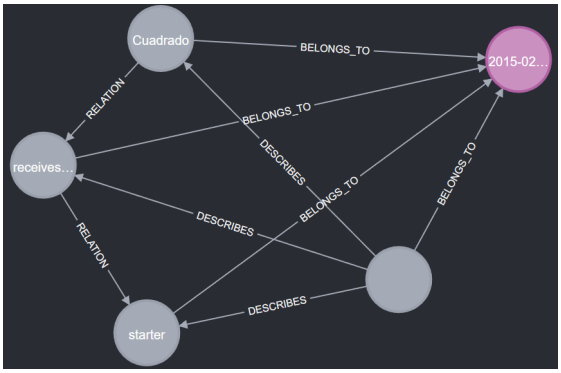


Fig. 7: An example of all connections.

4 Preliminary Results

In this section, we demonstrate the applications of our soccer knowledge graph within the context of LLM. Our soccer knowledge graph facilitates the summarizing and analysis of teams' and players' performances within a single game, across an entire league, or throughout a season. Furthermore, this graph is adept at supporting question-answering tasks, showcasing its versatility and utility in processing complex sports data. ASR data are unstructured, and their direct utilization in question-answering systems, such as LLMs, can lead to inaccuracies and hallucinations. In our project, unstructured ASR data is converted into

structured information through the Soccer-GraphRAG framework, leveraging a comprehensive knowledge graph. Our approach not only mitigates the typical inaccuracies found in direct analysis of ASR data but also significantly enhances the reliability and precision of the output. The following examples illustrate how our Soccer-GraphRAG framework leverages this enriched knowledge graph to deliver more accurate, reliable, and context-aware responses in a sports analytics setting.

Example 1: This example illustrates the application of our method for handling Q&A tasks. The query was, 'Tell me who scored the goal.' The LLM first converted this inquiry into a Cypher query, as depicted in Figures 8 (Generated Cypher). This query was then used to interrogate the soccer knowledge graph, identifying pertinent nodes and edges. The execution details retrieved from the knowledge graph are detailed in the 'Full Context' section of Figures 8. Utilizing this context, the LLM derived the answer, which is displayed in the 'Result' section of Figures 8. Another example of Q&A task is shown in Fig9.

Example 2: This example illustrates the application of our method for handling summarization tasks. The query was, 'Tell me about Ivanovic and Cuadrado.' LLM first converted this inquiry into a Cypher query, as depicted in Figures 10 (Generated Cypher). This query was then used to interrogate the soccer knowledge graph, identifying pertinent nodes and edges. The execution details retrieved from the knowledge graph are detailed in the 'Full Context' section of Figures 10. Utilizing this context, the LLM derived the answer, which is displayed in the 'Result' section of Figures 10.

```
> Entering new GraphCypherQAChain chain...
Generated Cypher:
MATCH (entity1:Entity)-[:RELATION]->(r:Relation)-[:RELATION]->(entity2:Entity)
WHERE r.name = 'scored_by'
RETURN entity1.name, entity2.name
Full Context:
[{'entity1.name': 'Felipe Luis', 'entity2.name': 'Chelsea'}, {'entity1.name': 'Felipe Luis', 'entity2.name': 'Ivanovic'}, {'e
ntity1.name': 'Felipe Luis', 'entity2.name': 'first'}, {'entity1.name': 'Felipe Luis', 'entity2.name': 'Felipe Luis'}, {'enti
ty1.name': 'Felipe Luis', 'entity2.name': "Ivanovic's fourth goal"}, {'entity1.name': 'Felipe Luis', 'entity2.name': 'goal'},
{'entity1.name': 'Ivanovic', 'entity2.name': 'Chelsea'}, {'entity1.name': 'Ivanovic', 'entity2.name': 'Ivanovic'}, {'entity1.
name': 'Ivanovic', 'entity2.name': 'first'}, {'entity1.name': 'Ivanovic', 'entity2.name': 'Felipe Luis'}]

> Finished chain.
 {'query': ' tell me who scored the goal',
  'result': 'Felipe Luis scored the goal.'}
```

Fig. 8: Sample Q&A application.

The examples provided demonstrate how the application of our methodology can lead to a more accurate and nuanced understanding of game dynamics and player performances. Furthermore, the use of a knowledge graph in LLM-based systems minimizes the risk of inaccuracies and hallucinations often associated with direct ASR data usage, thereby increasing confidence in the outputs generated by such systems. Looking forward, the refinement and expansion of our framework could significantly improve the efficiency of sports analytics, potentially influencing other fields where issues of unstructured data present similar challenges. The next phase of our project will focus on expanding the application

```
MATCH (e1:Entity)-[:RELATION]->(r:Relation)-[:RELATION]->(e2:Entity)
WHERE r.name = 'injured_by'
RETURN e1.name, e2.name
Full Context:
[{'e1.name': 'OBI MIKEL', 'e2.name': 'OBI MIKEL'}]
> Finished chain.
{'query': 'tell me who injuried in the game',
 'result': 'OBI MIKEL injuried in the game.'}
```

Fig. 9: Sample Q&A application.

```
Generated Cypher:
MATCH (entity1:Entity)-[:RELATION]->(r:Relation)-[:RELATION]->(entity2:Entity), (d:Description)-[:DESCRIBES]->(r)
WHERE entity1.name = 'Ivanovic' AND entity2.name = 'Cuadrado'
RETURN r.name, d.name
Full Context:
[{'r.name': 'offers', 'd.name': 'Ivanovic offers Cuadrado in that unmarking to win the baseline.'}, {'r.name': 'unmarking_off
ers', 'd.name': 'Where Ivanovic is, Cuadrado offers himself in that unmarking to win the baseline.'}]

> Finished chain.
{'query': ' tell me about Ivanovic and Cuadrado',
 'result': 'Ivanovic offers Cuadrado in that unmarking to win the baseline. Cuadrado offers himself in that unmarking to win
the baseline where Ivanovic is.'}
```

Fig. 10: Sample summarization application.

of Soccer-GraphRAG to encompass larger datasets, including both structured and unstructured data, thereby testing the framework's versatility and scalability.

5 Conclusion

We present the Soccer-GraphRAG framework which aims to support the retrieval of soccer-related information via natural language queries. Relying on an experimental knowledge graph structure built upon the SoccerNet-Echoes dataset, the framework has shown promising performance in preliminary experiments. The next steps include the fine-tuning and up-scaling of the graph structure to reflect the entire dataset.

Acknowledgement

This work was partly funded by the Research Council of Norway, project number 346671 (AI-Storyteller), and has benefited from the Experimental Infrastructure for Exploration of Exascale Computing (eX3), which is financially supported by the Research Council of Norway under contract 270053.

References

1. Chang, Y., Wang, X., Wang, J., et al.: A Survey on Evaluation of Large Language Models. ACM Trans. Intell. Syst. Technol. **15**(3), 1–45 (Mar 2024). https://doi.org/10.1145/3641289
2. Chen, Z., Zhang, Y., Fang, Y., et al.: Knowledge Graphs Meet Multi-Modal Learning: A Comprehensive Survey. arXiv (Feb 2024). https://doi.org/10.48550/arXiv.2402.05391
3. Gao, Y., Xiong, Y., Gao, X., et al.: Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv (Dec 2023). https://doi.org/10.48550/arXiv.2312.10997
4. Gautam, S.: FactGenius: Combining Zero-Shot Prompting and Fuzzy Relation Mining to Improve Fact Verification with Knowledge Graphs. arXiv (2024)
5. Gautam, S., Sarkhoosh, M.H., Held, J., Midoglu, C., Cioppa, A., Giancola, S., et al.: SoccerNet-Echoes: A Soccer Game Audio Commentary Dataset. arXiv (May 2024). https://doi.org/10.48550/arXiv.2405.07354
6. Giancola, S., Amine, M., Dghaily, T., Ghanem, B.: SoccerNet: A Scalable Dataset for Action Spotting in Soccer Videos. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 18–22. IEEE (2018). https://doi.org/10.1109/CVPRW.2018.00223
7. Jeong, C.: A Study on the Implementation of Generative AI Services Using an Enterprise Data-Based LLM Application Architecture. arXiv (Sep 2023). https://doi.org/10.54364/AAIML.2023.1191
8. Jia, R., Zhang, B., Méndez, S.J.R., et al.: Leveraging Large Language Models for Semantic Query Processing in a Scholarly Knowledge Graph. arXiv (May 2024). https://doi.org/10.48550/arXiv.2405.15374
9. Jiawei, Chen, and Lin, Hongyu, and Han, Xianpei, and Sun Le : Benchmarking Large Language Models in Retrieval-Augmented Generation. The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24) (Jan 2024), https://arxiv.org/pdf/2309.01431
10. Pan, J.Z., Vetere, G., Gomez-Perez, J.M., et al.: Exploiting linked data and knowledge graphs in large organisations. Springer International Publishing AG, Cham, Switzerland (Jan 2017). https://doi.org/10.1007/978-3-319-45654-6
11. Siriwardhana, S., Weerasekera, R., Wen, E., et al.: Improving the Domain Adaptation of Retrieval Augmented Generation (RAG) Models for Open Domain Question Answering. Transactions of the Association for Computational Linguistics **11**, 1–17 (Dec 2023). https://doi.org/10.1162/tacl_a_00530
12. Wei, L., Xinyan, X., Jiachen, L., Hua, W., Haifeng, W., Junping, D.: Leveraging Graph to Improve Abstractive Multi-Document Summarization. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (July 2020). https://doi.org/10.18653/v1/2020.acl-main.555
13. Xu, W., Fang, M., Yang, L., et al.: Enabling Language Representation with Knowledge Graph and Structured Semantic Information. In: International Conference on Computer Communication and Artificial Intelligence (CCAI). IEEE (2021). https://doi.org/10.1109/CCAI50917.2021.9447453
14. Yang, J., Jin, H., Tang, R., Han, X., Feng, Q., Jiang, H., et al.: Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond. ACM Trans. Knowl. Discovery Data (Jun 2023). https://doi.org/10.1145/3649506
15. Ye, X., Yavuz, S., Hashimoto, K., et al.: RnG-KBQA: Generation Augmented Iterative Ranking for Knowledge Base Question Answering. arXiv (Sep 2021). https://doi.org/10.48550/arXiv.2109.08678