KickStat: Uncovering Rare Soccer Events Through Knowledge Graph Analytics

Kishore Vanapalli Sam Serdah Alex Leslie
Carleton University
Ottawa ON Canada
{KishoreVanapalli, SamSerdah, AlexSLeslie}@cmail.carleton.ca

ABSTRACT

With the advent of availability of comprehensive and huge soccer match data, the demand for deep analytical studies on the data to gain critical insights into match strategies is increasing. This presents new challenges for researchers and industry to uncover rare insights from the available data. In this work we present a novel comprehensive analytical framework which is based on knowledge graphs (KG) to help to identify interesting facts emerging out of match data. We also present several internal details of the proposed framework, including adopted data model, design decisions, query formatting, performance evaluations done to measure the efficiency of the adopted approach. Finally, we highlight our experiments to integrate the knowledge graph with a Large Language Model (LLM) to automate complex query generation to automatically generate rare match events from the knowledge graph.

1 INTRODUCTION

1.1 Problem Statement

In the realm of soccer (football) analytics, there is a critical need for a comprehensive system capable of identifying and analyzing rare, extraordinary events during matches. These rare events, which often hold key insights into player performance and match dynamics, remain hidden within vast datasets. Traditional analytics fall short in uncovering them. To address this challenge, our project aims to develop 'KickStat,' a sophisticated knowledge graph-based analytics system. KickStat will enable sports analysts to timely and effortlessly pinpoint and study rare occurrences in the world of soccer, enhancing our understanding of the game and offering valuable insights for sports analysts.

1.2 Importance

The problem of uncovering rare soccer events through Knowledge Graph Analytics is both interesting and important for several reasons: Enhancing Soccer Understanding, Fan Engagement and Data-Driven Decision-Making. It not only enriches our understanding of the sport but also offers practical benefits to sports analysts and fans. It is a multidisciplinary challenge that combines data science, sports analytics, and technology, making it a compelling and impactful area of exploration.

1.3 Difficulty

Uncovering rare soccer events through Knowledge Graph Analytics is a challenging problem to solve due to several intricacies and complexities such as unavailability of relevant labeled data for both training and validation, rareness of events, semantic complexity of rare soccer events, Knowledge Graph inherent complexities and missing benchmarks. The difficulty in solving this problem lies in the need to manage, analyze, and interpret large and diverse datasets while addressing the specific challenges of identifying rare events in the dynamic and multifaceted domain of soccer. It requires a multidisciplinary approach that combines domain expertise, data science, and technology to overcome these complexities.

1.4 Solution

In the developed sophisticated knowledge graph-based analytics system, KickStat, we aimed to build a new knowledge Graph system from scratch based on the StatsBomb open data [1] and explored different methods to build capability to identify potential rare events. Our contributions are as follows:

i. Design a data model for StatsBomb open data to build a knowledge graph
ii. Preprocessing of the data to split it into designed knowledge graph entities and relationships
iii. Building a KG based on the preprocessed data and designed data model
iv. Defining sample template rare events
v. Developing Cypher queries to extract template rare events
vi. Developing a Graphical User Interface (GUI) to guide the users to perform querying
vii. Performance evaluation of queries particularly on complex queries
viii. Exploring various ways to integrate the knowledge graph with a Large Language Models (LLMs) to automate complex query generation to automatically generate rare match events from the knowledge graph

ix. All the source code including preprocessing, building and populating KG, Cypher queries, GUI and LLM fine-tuning are shared in the following GitHub page.

2 KICKSTAT ARCHITECTURE

In this section, we first provide an overview of KickStat's architecture. Then, we dive into the main design decisions in the framework.

2.1 Overview

KickStat framework is based on the popular graph database Kùzu [2] [3], a highly scalable, extremely fast and easy-to-use embeddable database which allows graph-based modeling and querying, graph-optimized storage and graph-optimized query execution. As an extension to the database and querying module, we built a GUI for user input and querying.

2.2 Data Model

The data model is designed for the StatsBomb open data which is certain league matches data openly available for public use for research projects in football analytics. The data consists of detailed information about several key entities of football matches like Competitions, Season, Matches, Players, Teams, Managers, Referees, Stadiums and real-time events captured on a very high frequency period. The following are some of the key design decisions we made while developing an efficient schema for the knowledge graph.

As Statsbomb open data is centered around matches, to efficiently retrieve rare events the following guidelines are followed to build a Knowledge Graph.

Rule 1. "Match" is modeled as a core entity and all simple properties specific to a match like match_id, match_date etc., are modeled as entity properties.

Rule 2. Composite properties of match entity like home_team, competition, season etc., are modeled as separate entities, called supported entities and relations are created to connect these entities with the hosting Match core entity.

Rule 3. Properties which are related to both Match entity and supported entities will be added to the relationship, for example, if a player receives a card in a match, the card type for the player for that match is captured as a relation property and added to the relationship between the player and the match.

Rule 4. Match event data is modeled as Event entities connected to corresponding match and supported entities, for example, possession_team property is connected to Team entity, player property is connected to corresponding Player entity. Also, as event data consists of a large number of properties related to various event types, we had two options to capture the data in the knowledge graph: capturing as properties of event node itself or capturing as subevent entities each specific to a type of event like Bad behavior, Pass etc. The first option comes with the disadvantage that there will be a lot of sparse entries in each event entity node, while the second option comes with the disadvantage that there will be an increased hops in retrieving relevant data. We went with the first option as we are not considering all the event types due to limited scope.

Figure 1. Design of core entity, Match, and relation with supported entities

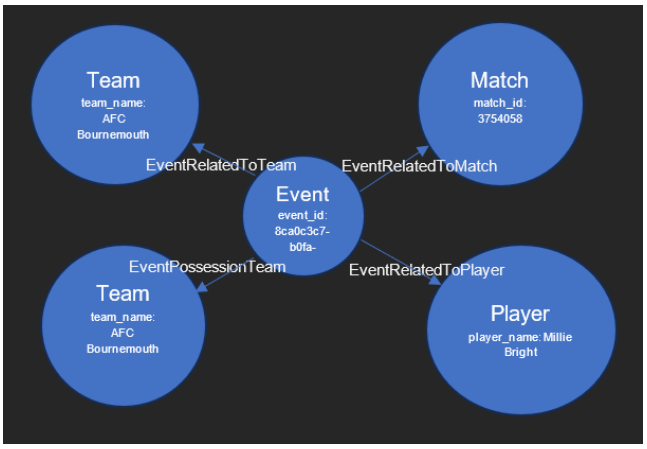

Figure 2. Design of Event entity and its related entities

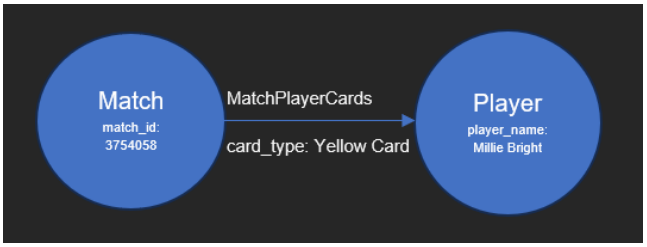

Figure 3. Design highlighting properties added to relations

Project GitHub page:
https://github.com/Kishorevb/kickstat/tree/main

2.3 Data Preparation

StatsBomb data consists of all match data in various JSON format files combining data of all entities and relationships. Whereas Kùzu expects the entities and relationships data in dedicated CSV formatted files. As part of the data preparation step, we designed reusable generic scripts to map StatsBomb data format to KickStat data model format. The scripts are shared in the GitHub repository.

2.4 Building a Knowledge Graph

Building a knowledge graph in Kùzu from the prepared StatsBomb open data consists of two primary steps: Creating schema with the designed entities and relationships as Tables and populating tables with prepared CSV data files. As part of the schema design, we designed 11 entities and 19 relationships among these entities. The complete details of the schema including entities and their properties can be found in the GitHub documentation. The preprocessed files of all events and relationships are also shared as part of GitHub. Once the schema is designed and populated, the KG is ready for querying.

2.5 Define Rare Events

Rare events, which often hold key insights into player performance and match dynamics, remain hidden within vast datasets which when uncovered enhance our understanding of the game and offer valuable insights for sports analysts. For the interest of our work, we have defined some sample templates of rare events which are both generic and exciting.

1. Given a competition name, season name and a player's name, where does he rank with respect to how many matches he is involved in?
2. Given a competition name, season name and a team name, where does the team rank with respect to how many goals the team conceded?
3. Given a competition name, season name, a team name and a stadium name, where does the team rank with respect to win-loss ratio?
4. Given a competition name, season name and a player's name, find out how many referees referred in the matches he is involved in.
5. Given a country name, find how many teams have managers from the given country?
6. Given a player's name, find out how many managers he worked with in all available matches.
7. Given a competition name, season name and a player name, where does he rank with respect to how many events he is involved in? (Table: EventRelatedToPlayer)
8. Given a competition name, season name and a player name, where does he rank with respect to how many goals he scored ?
9. Given a competition name and season name, find the player with the most cards received?
10. Given a competition name and season name, find if any players involved in a self-goal.

Of course, the sample queries can be extended for several other cases, and in a later section we have shown our experiments to automatically generate rare events.

2.6 Cypher Queries

Cypher [4] is Kùzu 's graph query language that lets you retrieve data from the graph. It is like SQL for graphs and was inspired by SQL, so it lets you focus on what data you want out of the graph (not how to go get it). It is the easiest graph language to learn by far because of its similarity to other languages, and intuitiveness. Given a query objective, like SQL, Cypher also provisions several ways to perform queries to retrieve desired outcome using several languages constructs like query and subquery clauses [4].

One of our objectives in designing both the rare events and Cypher queries to add enough variety so that the LLM can learn the schema in an exhaustive way to efficiently generate complex queries as Cypher queries is an important prompt LLM processes to understand the schema.

The following Cypher queries highlight both intuitiveness provided by Cypher as well as the variety in our defined rare events.

```
MATCH (a:MatchNode)-[e:MatchPlayers]->(b:Player) WHERE b.player_name = "Glenn Murray" AND EXISTS \
{MATCH (a)-[MatchHeldInSeason]->(s:Season) WHERE s.season_name="2015/2016"} AND EXISTS \
{MATCH (a)-[MatchPartOfCompetition]->(c:Competition) WHERE c.competition_name = "Premier League"} RETURN count(a.match_id)
```

Query 1. Query to retrieve number of matches played by a chosen player in a given (Competition, Season)

```
MATCH (a:Player)<-[EventRelatedToPlayer]-(e:Event) WHERE e.type_name = "Own Goal For" AND EXISTS \
{MATCH (a)<-[MatchPlayers]-(MatchNode) WHERE a.player_name = "So-Yun Ji"} AND EXISTS \
{MATCH (a)-[MatchHeldInSeason]->(s:Season) WHERE s.season_name="2015/2016"} AND EXISTS \
{MATCH (a)-[MatchPartOfCompetition]->(c:Competition) WHERE c.competition_name = "Premier League"} RETURN count(e.event_id)
```

Query 2. Query to retrieve number of goals scored by a chosen player in a given (Competition, Season)

```
MATCH (a:Player)<-[EventRelatedToPlayer]-(e:Event) WHERE \ e.bad_behaviour_card_name = "Yellow Card" OR e.bad_behaviour_card_name = "Second Yellow" OR e.bad_behaviour_card_name = "Red Card" AND EXISTS \
{MATCH (a)<-[MatchPlayers]-(MatchNode) WHERE a.player_name = "So-Yun Ji"} AND EXISTS \
```

{MATCH (a)-[MatchHeldInSeason]->(s:Season) WHERE s.season_name="2015/2016"} AND EXISTS \
{MATCH (a)-[MatchPartOfCompetition]->(c:Competition) WHERE c.competition_name = "Premier League"} RETURN count(e.event_id)

Query 3. Query to retrieve the player with the most cards received in a given (Competition, Season)

The complete list of working queries is part of the GitHub repository.

2.7 Graphical User Interface

For efficient user input and querying, we developed an extension Graphical User Interface to the Kùzu storage and database engine. This module prompts users to choose a query type, gathers relevant inputs like player name, team name etc. and finally displays the query output. The source code of the GUI is shared in the GitHub repository.

2.8 Large Model Language Fine-tuning

Popular LLM like ChatGPT has shown impressive capabilities in processing and generating human-like text. However, it is not without its imperfections. A primary concern is the model's propensity to produce either inaccurate or obsolete answers, often called "hallucinations." To address this issue, one popular approach that focuses on augmenting ChatGPT using a knowledge graph. This method aims to provide a structured context, ensuring the model outputs are accurate but also relevant and up to date. By bridging the gap between the unstructured textual world of ChatGPT and the structured clarity of knowledge graphs, it strives to enhance the effectiveness and reliability of AI language models.
With the objective of enabling a LLM like ChatGPT with the automatic generation of accurate complex Cypher queries to identify potential rare events, we followed the below process:

As the GPT-3 endpoint has no concept of context, we need to send the training examples along with every user input as shown in Figure 4.

3 EXPERIMENTAL EVALUATION

In this section, we experimentally demonstrate KickStat's efficiency in complex query retrievals and performance metrics collected during the process. In particular, we seek to answer the following questions:

1. How KickStat perform end-to-end over complex queries?
2. What query shapes benefit the most from KickStat?
3. How is the performance of complex queries with respect to CPU times?
4. How effective is to train an LLM with KB data to automatically generate queries for complex insights?

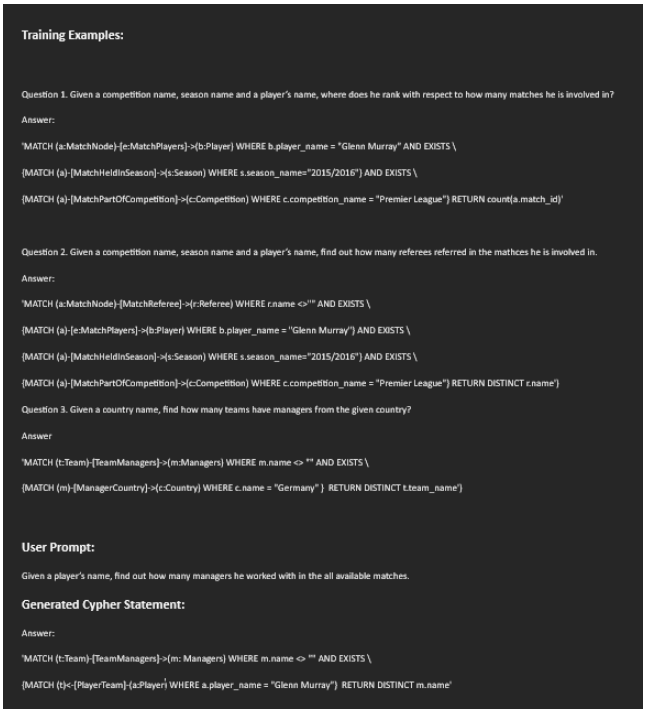

Figure 4. Sample input for ChatGPT Fine-tuning

3.1 KickStat Query Performance

Kùzu Cypher queries, like any graph database queries, are defined as n-hop where n is [0, inf). A hop in a graph database is defined as accessing another entity from source entity over a single relationship link. Hops allow to combine information from several entities which might result in extracting critical insights. The higher the value of n, the higher the chances of combining information from several entities over connecting relationships and thereby uncovering potential rare events. Of course, the higher number of hops for a query result in poor performance particularly for ranking queries where the query needs to be run iteratively over several entities.
In our study, we examined hops of cardinals 0-3 over complex queries and here are our observations:

1. Kùzu is optimized for running expensive, multi-hop queries on very large graphs. Particularly, the CPU time is sequentially proportional to the increased hops. On a standard CPU, each hop takes a few milliseconds, and a large multi-hop, which is rare for a majority of knowledge graphs will still be executed within a second.
2. Compared to other popular graph databases like Neo4j, Kùzu supports both vectorized and multi-threaded execution. Kùzu executes queries in a vectorized fashion, enabled by the fact that data is stored natively in a column-oriented manner and accessed in batches.

Query execution is multi-threaded via "morsel-driven parallelism" – in this approach, the query workload is divided into several small parts (morsels) that are dynamically distributed between threads during runtime. Each thread takes its own morsel as far through the pipeline as possible, only stopping & synchronizing with other threads when absolutely needed. This efficiency is clearly evident when a ranking query is executed over a large number of entities.

3.2 Which Queries will KickStat Benefit?

As mentioned in the previous section, as Kùzu is designed to be highly scalable and efficient, running multi-hop queries seems to be quite efficient. The efficiency is highlighted particularly over multi-hop queries for several hundreds of entities like for ranking queries due to vectorized and multi-threaded execution support.

3.3 Fine-tuning LLMs with Knowledge Base

From our experiments with integrating LLMs with Knowledge Base queries to automatically generate Cypher queries for complex queries, we have the following observations:

1. A significant number of sample queries both with variety and complexity are required as training samples to fine-tune LLMs to understand the schema of a complex knowledge graph to retrieve accurate Cypher queries for complex queries.
2. Also, the time required to generate Cypher queries for complex use cases is not trivial.

3.4 Limitations

The following are some of the limitations of the KickStat, all primarily resulting due to the limited scope of the project:

1. Not all available data is leveraged for building the knowledge graph, particularly the data related to Match360 and some of the event's data. Incorporating all of the data might lead to additional insights.
2. Also, the design choice to define Event schema as a large entity with several sparse properties over multi-hop subevent design which might be more memory efficient.
3. Even though we did some experiments on the integration of Knowledge Base with LLMs, the results are not encouraging, primarily because there is not enough prompt data in terms of number of sample queries.
4. The developed GUI is very minimal and lacks several usability features like auto completion of text and auto prompting.
5. Lack of in-depth performance analysis of the developed KickStat framework.

3.5 Future Work

As part of the work, we made significant progress on developing a sophisticated knowledge graph-based analytics system which will enable sports analysts to timely and effortlessly pinpoint and study rare occurrences in the world of soccer, enhancing our understanding of the game and offering valuable insights for sports analysts. We would like to take our effort to build more enhancements in the future:

1. To leverage all the available StatsBomb open data to build a comprehensive knowledge graph, particularly the data related to Match360 and some of the event's data which is missing. Incorporating all of the data certainly will lead to additional insights. Also, we can automatically incorporate new data into the KG as and when the data is released by the vendor.
2. We want to consider both design approaches, sparse entities and multi-hop subevents, and compare performance on both.
3. We would like to experiment with multiple LLMs, and multiple prompts coming from huge number of queries to improve and assess integration of LLMs with KG to automatically generate complex queries to identify rare events.
4. We would like to develop a sophisticated GUI with usability features like auto completion of text and auto prompting to improve user experience.
5. In-depth performance analysis of the developed KickStat framework over multiple query types.

4 RELATED WORK

4.1 Sports Analytics

Big data and data analytics are becoming increasingly common in sports. The study [5] demonstrates the changes caused by the use of technologies in the context of big data and sports analytics on the basis of a systematic literature review (SLR) in professional football and analyzes to what extent their use has changed and will continue to change the strategies of professional football clubs and their stakeholders. The showed results show that big data and sports analytics have become important tools in professional football and can increase the competitiveness of professional football clubs.

Using machine learning, several papers like [6], [7], [8] highlight sports Analytics as an emerging research area with several applications in a variety of fields. These could be, for example, the prediction of an athlete's or a team's performance, the estimation of an athlete's talent and market value, as well as the prediction of a possible injury. Teams and coaches are increasingly willing to embed such "tools" in their training, in order to improve their tactics. These papers review the literature

on Sports Analytics and propose new approaches for prediction. The conducted experiments using suitable algorithms mainly on football-related data, in order to predict a player's position in the field. They also accumulated data from past years, to estimate a player's goal scoring performance in the next season, as well as the number of a player's shots during each match, known to be correlated with goal scoring probability. Results are very promising, showcasing high accuracy, particularly as the predicted number of goals was very close to the actual one.

4.2 Graph Databases

The paper [9] explains Graph database models can be defined as those in which data structures for the schema and instances are modeled as graphs or generalizations of them, and data manipulation is expressed by graph-oriented operations and type constructors. These models took off in the eighties and early nineties alongside object-oriented models. Their influence gradually died out with the emergence of other database models, in particular geographical, spatial, semi structured, and XML. Recently, the need to manage information with graph-like nature has reestablished the relevance of this area. The main objective of this survey is to present the work that has been conducted in the area of graph database modeling, concentrating on data structures, query languages, and integrity constraints.

Social networking has become the Internet's most common activity and college students are some of its most fervent users. Unfortunately, using social networks places college students at risk because of the content that they often post. Previous research has shown that students commonly post content on social networking sites that could damage their chances for employment. Although students know the risks that they are taking, many continue to post content that is inappropriate. Researchers have dubbed this behavior the "posting paradox". In order to better understand the paradox, this paper reports the results of a field study which compares student uses of two of the most popular social networking sites, Facebook and Twitter. The results of the study indicate that students post inappropriate content on both sites, with the paradox being more pronounced on Twitter. The paper [10] discusses the implications of these results and proposes areas for future research.

4.3 Knowledge Bases

This paper [12] explores new types of knowledge, and new ways of organizing the production of it, may emerge as knowledge producers respond to the challenges posed by a changing society. This paper focuses on the core knowledge of one such emerging field, namely, innovation studies. To explore the knowledge base of the field, a database of references in scholarly surveys of various aspects of innovation, published in "handbooks", is assembled and a new methodology for analyzing the knowledge base of a field with the help of such data is developed. The paper identifies the core contributions to the literature in this area, the most central scholars and important research environments, and analyses – with the help of citations in scholarly journals – how the core literature is used by researchers in different scientific disciplines and cross-disciplinary fields. Based on this information a cluster analysis is used to draw inferences about the structure of the knowledge base on innovation. Finally, the changing character of the field over time is analyzed, and possible challenges for its continuing development are discussed.

In this paper [11], a knowledge base graph embedding module is constructed to extend the versatility of knowledge based VQA (Visual Question Answering) models. The knowledge base graph embedding module constructed in this paper extracts core entities from images and text, and maps them as knowledge base entities, then extracts the sub-graphs closely related to the core entities and converts the sub-graphs into low-dimensional vectors to realize sub-graph embedding. In order to achieve good subgraph embedding, we first extracted two experimental knowledge bases with rich semantics from DBpedia: DBV and DBA. Based on these two knowledge bases, this paper selects several excellent models in knowledge base embedding as test models, including SE (structured embedding),SME(semantic matching energy function), and TransE model to produce link prediction. The results show that there is a clear correspondence between the entities of the DBV, which can achieve excellent node embedding. And the TransE model can achieve a good knowledge base embedding, so we built the knowledge base graph embedding module based on TransE. And then we construct a VQA model (KBSN) based on the knowledge base graph embedding. Experimental results on VQA2.0 and KB-VQA data sets prove that the knowledge base graph embedding module improves accuracy.

4.4 LLMs and Knowledge Bases

This paper [13] explores recent progress in pretraining language models on large textual corpora led to a surge of improvements for downstream NLP tasks. Whilst learning linguistic knowledge, these models may also be storing relational knowledge present in the training data and may be able to answer queries structured as "fill-in-the-blank" cloze statements. Language models have many advantages over structured knowledge bases: they require no schema engineering, allow practitioners to query about an open class of relations, are easy to extend to more data, and require no human supervision to train. We present an in-depth analysis of the relational knowledge already present (without fine-tuning) in a wide range of state-of-the-art pretrained language models. We find that (i) without fine-tuning, BERT contains relational knowledge competitive with traditional NLP methods that have some access to oracle knowledge, (ii) BERT also does remarkably well on open-domain question answering against a supervised baseline, and (iii) certain types of factual knowledge are learned much more readily than others by standard language model pretraining approaches. The surprisingly strong ability of these models to recall factual knowledge without any fine-tuning

demonstrates their potential as unsupervised open-domain QA systems.

In the paper [14], pretrained language models have been suggested as a possible alternative or complement to structured knowledge bases. However, this emerging LM-as-KB paradigm has so far only been considered in a very limited setting, which only allows handling 21k entities whose single-token name is found in common LM vocabularies. Furthermore, the main benefit of this paradigm, namely querying the KB using a variety of natural language paraphrases, is underexplored so far. Here, we formulate two basic requirements for treating LMs as KBs: (i) the ability to store a large number of facts involving a large number of entities and (ii) the ability to query stored facts. We explore three entity representations that allow LMs to represent millions of entities and present a detailed case study on paraphrased querying of world knowledge in LMs, thereby providing a proof-of-concept that language models can indeed serve as knowledge bases.

5 CONCLUSIONS

With the rise in the accessibility of extensive soccer match data, there is a growing demand for in-depth analytical studies aimed at extracting crucial insights into match strategies. This trend poses new challenges for both researchers and the industry in the quest to unveil unique perspectives from the available data. In this study, we introduce an innovative and comprehensive analytical framework centered around Knowledge Graphs to facilitate the identification of noteworthy patterns within match data. Additionally, we disclose various internal aspects of the proposed framework, such as the chosen data model, design considerations, query formatting, and performance evaluations conducted to gauge the effectiveness of the adopted approach. Lastly, we showcase our experiments integrating the knowledge graph with a LLMs to automate the generation of complex queries, thereby enabling the automatic extraction of rare match events from the knowledge graph. We also highlighted the exiting limitations of the developed framework and proposed future work to address the limitations.

REFERENCES

[1] Statsbomb. (2019). Statsbomb/open-data: Free football data from StatsBomb. GitHub. https://github.com/statsbomb/open-data

[2] Semih Salihoglu Kùzu: A Database Management System For "Beyond Relational" Workloads

[3] Python: Kùzu. (2023). https://kuzudb.com/

[4]Cypher: Kùzu. Kzu RSS. (2023). https://kuzudb.com/docusaurus/cypher/

[5] Tim A. Herberger, Christoph Litke The Impact of Big Data and Sports Analytics on Professional Football: A Systematic Literature Review

[6] Konstantinos Apostolou, Christos Tjortjis Sports Analytics algorithms for performance prediction

[7] Victor Chazan Pantzalis, Christos Tjortjis Sports Analytics for Football League Table and Player Performance Prediction

[8] Ahmet Talha Yiğit, Barış Samak, Tolga Kaya Football Player Value Assessment Using Machine Learning Techniques

[9] Renzo Angles, Claudio Gutierrez Survey of graph database models

[10] Justin J. Miller Graph Database Applications and Concepts with Neo4j

[11] Wenfeng Zheng a, Lirong Yin b, Xiaobing Chen a, Zhiyang Ma a, Shan Liu a, Bo Yang Knowledge base graph embedding module design for Visual question answering model

[12] Jan Fagerberg a b c, Morten Fosaas a, Koson Sapprasert Innovation: Exploring the knowledge base

[13] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, Sebastian Riedel Language Models as Knowledge Bases?

[14] Benjamin Heinzerling, Kentaro Inui Language Models as Knowledge Bases: On Entity Representations, Storage Capacity, and Paraphrased Queries