Project Summary: Baseball KG

November 2022

Fandel Lin
fandel.lin@usc.edu

Zihao Han
zihaohan@usc.edu

1 PROJECT DOMAIN AND GOAL

The project objective is *building MLB knowledge graph for player-performance and game-result prediction with statistical analysis.*

Baseball is one of the major professional sports in the United States, and can attract tens of thousands of on-site spectators per match. However, instead of simply skimming through match results, there are complex interactive and synergistic relationships among players in baseball.

In this project, we plan to build a knowledge graph about baseball games. The knowledge graph will hold teams of a baseball league, match schedules of teams, game results of matches, players of teams, historical records of players, and statistics of players in matches. For different types (e.g., pitchers and hitters) of players in a team, the statistics here span from pitch types, spin direction, and pitch movement for pitchers; to pitch tracking, plate discipline, and batted-ball profiles for hitters. This brings a total of at least 11 semantic types.

1.1 Project Goal

With visualization, this knowledge graph could help people dig into the interactive and synergistic relationship among players in baseball games. Furthermore, we plan to exploit this knowledge graph to provide a data-centric approach to inferring the performance of players and the results of matches.

To achieve our goal, the architecture is defined and depicted in Fig 1.

1.2 Motivating Example

As a motivating example, spectators for baseball games not only want to know whether the team they support can win a match, but also want to take part in some 'special events' in person! Such events could be intuitive, for instance, a team winning the championship, or a pitcher (a player whose routine is stable) winning a match. However, most of such events are uncertain, for instance, a team playing a 'perfect game' (no opposing player ever reaches base by any means in a match, e.g., there are only 21 perfect games in MLB since 1903), a pitcher getting his shot outs or complete games, or a hitter getting his career home run in round hundreds (e.g., Albert Pujols gets his 698^{th} home run on 9/16, but his 699^{th} and 700^{th} ones both on 9/24). These types of events are difficult to be expected if a spectator only relies on the impression of recent situations of players or teams.

1.3 Research Question

Accordingly, the major quantitative evaluation for this project consists of three parts: one for entity resolution, and two for predictive tasks. To facilitate the extrapolated prediction, we use a sliding window of 200 matches. For instance, we use match number 1 to 200 to construct a sub-KG, and use this sub-KG for predicting the results and performance in match number 201 to 400. The formulations are described as follows:

- **RQ1** Can our approach retain good performance in entity resolution by treating a subset of manually labeled data as the validation set?
- **RQ2** Can our approach perform good extrapolated prediction for match result (either win or lose) under a sliding window?
- **RQ3** Can our approach perform good extrapolated prediction for player performance (how many point one player get or lose in a match) under a sliding window?

2 DATASET

There are 2430 matches and at least 1200 active players per year in MLB. We will crawl the game and schedule info from the first dataset, and the player info from all three datasets.

(1) MLB.com
- The official website of Major League Baseball.
- Contains all season data for 30 total teams.

(2) Baseballsavant.mlb.com
- The official statistics of Major League Baseball.
- Contains statistics of each player.

(3) Wikipedia.org
- Any other information of players beside match statistics.

All three datasets have structured tables, this is expected to bring around 6030 pages and more than 2400 structured records. Screenshots for the three datasets are displayed in Fig 2.

3 TECHNICAL CHALLENGE

The project will solve technical challenges in analyzing knowledge graphs. The difficulties lie in categorizing entities (players) with similar characteristics, understanding the relationship among different types of entities, and exploiting the uncertainty in such relationships for an inference. We conduct the experiment on a real-world dataset from the Major League Baseball (MLB), and evaluate the performance for the inference problem based on the accuracy of game results (either win or lose) and player performance (how many point a player get or lose). To be more precise, we construct a sliding window of 200 matches for all the matches in a season. For each two windows, we treat the first window (200 matches) for constructing the predictive model, and use the next window (200 matches) for evaluation.

Compared to other ball games, MLB has more teams and players, so we have more relationships between players on each team. Plus with more game data, we could face a very large knowledge graph and then may slow down our predictions.

On the other hand, for designing the custom ontologies, we include the statistics and records from multiple aspects in baseball to support further extrapolated prediction.

Next, We use SPARQL queries to extract and build our knowledge graph for players from MLB in various aspects. Precisely, we harvest different pitch types, spin direction, and pitch movement for pitchers; different pitch tracking, plate discipline, and batted-ball profiles for hitters from the baseball savant dataset. Based on these statistics, we can categorize these pitchers and hitters, as well as the teams. We will use these trends to build a probabilistic predictive model among hitters and pitchers based on their categories (embedding).

4 METHODOLOGY

The architecture for our approach is depicted in Fig 1. In this section, we introduce our approach in dealing with the technical challenges for each step in the architecture.

4.1 Data Crawling, Extraction, and Cleaning

The first challenge lies in crawling dynamic web pages and web links especially for parsing data from MLB.com. We utilize *Selenium* to simulate mouse clicks to click on the link of the next game to crawl all game schedules and dates. Meanwhile, since the pages for match information from MLB.com have their links consist of multiple variables (including but not limited to the game id, game type and game stage), we use additional resources such as the MLB official gamelog to make the corresponding web link completed.

For data extraction, since there are thousands of tables needed to be stored, we set each player as a directory and each table as a individual file to read in order to have a more uniform format on subsequent reads. However, it slows down when reading and ends up with more than over 7,000 tables. Besides, when referring to the team to which a player belongs to, Baseballsavant.mlb.com tends to use image (symbol) instead of text in some tables.

Finally, for data cleaning, to deal with the encoding issue and unstructured data, we first re-encode the extracted tables to uniform encoding utf-8. Next, we transform some symbols to text and reconstruct the clawing logic if needed.

4.2 Ontology and Entity Resolution

Another critical challenge for this project lies in the lack of an existing ontology. To solve this, we design custom ontologies that capture the concepts from the structured sources. For instance, the class in the ontology includes player, team, match, statistics in baseball (e.g., hits, HR, RBI,

IP, etc.; all statistics are stored on a basis of "per-match", "per-season", "per-month", "per-type-of-opponent", etc.); on the other hand, the relation in the ontology includes (team, has match against, team), (team, has, player), (player, plays in, match), (player, plays against, player), (pitcher, has, pitch type), (hitter, has, pitch tracking), (player, belongs to, player type), (player, plays against, player type), and (player type, plays against, player type), etc. The overview of our custom ontology is illustrated in Fig 3, with some examples listing the defined properties, domains, and ranges in Fig 4. The statistics for our custom ontology are provided in Table 1, with more than 100 nodes in the custom ontology itself.

For one of the quantitative research questions, we utilize entity resolution to solve the issues of distinct representation of data across various sources. For instance, in Fig 5, in the gamelog from MLB.com, when referring to a specific player, sometimes it uses last name or last name with the first letter from first name; however, sometimes it rely on last name with multiple (more than 2) letters from first name, or even letters from the player's full name that is only provided in another source. Accordingly, we incorporate multiple attributes from each source to perform entity resolution. Precisely, as an example depicted in Fig 6, we use a player's name, tokens extracted from name, and the team information (with transfer record if this player is transferred from one team to another during a seaon) with Jaccard index for similarity matching.

On the other hand, when referring to a specific team, there are also different representations across sources. However, since there are only 30 teams in MLB, we implement a rule-based system for direct replacement. Therefore, the quantitative evaluation for entity resolution only applies to the linking of players.

4.3 Extrapolated Prediction

For two of the quantitative research questions, we target the extrapolated prediction of match results and player performance in future matches.

To predict the match results in the future, we incorporating recent performance with expectations based on teams with similar characteristics. Precisely, we first construct temporal embedding for each team. The vectors here include but are not limit to the recent performance in statistics of the team (e.g., RBI, HR, ER, etc.) and the statistics of players this team has. Based on this temporal embedding, we expect similar match results when two teams under similar embedding play against the same team. For instance, team A defeated team B recently, and team A has similar temporal embedding with team C (considering their recent performance in terms of team and players); we expect team C could also defeat team B in the upcoming match. On the other hand, if team D defeated team B recently, but team D does not have similar temporal embedding with team C; there is no positive feedback from team D to team C for the upcoming match against team B.

Similarly, for predicting the player performance in the future, we construct embedding of players according to their recent performance. So that we can predict the points that one play might get or lose in a future match by accessing the "types" of team that this player face.

It is worth mentioning that, the reason for constructing temporal embedding for teams and players is that, according to the match schedule, the relationship among each pair of teams is extremely uneven. (In fact, *the algorithmic approach for constructing the MLB schedule is somewhat unfair and uneven before 2023 and can be traced back to 20th century due to the complex design in multiple divisions/ leagues and limited inter-division/ inter-league/ rivalry sets of matches*[1]; besides, we use the 2022 season for our project.) For instance, a team will play against specific teams more often (e.g., 19 times per season), face some teams less (e.g., 6 times per season) or even never meet some teams in a season. Therefore, to handle such uneven and possibly sparse relationship for facilitating extrapolated predictions, we need to construct embedding for teams and players to access their performance based on similar players or teams playing against opponents with corresponding characteristics.

4.4 User Interface

For the user interface, we provide multiple querying approaches for users to explore the Baseball KG. As screenshots shown in Fig 7, users can select from a set of pre-defined queries, or select their designated combinations of subjects and predicates.

[1]Example of an official press release talking about the amendments in 2013 MLB schedule: https://web.archive.org/web/20160304091328/http://m.mlb.com/news/article/38287660

5 EVALUATION

5.1 Quantitative Evaluation

The quantitative evaluation consists of three parts.

First, for entity resolution, we create a subset of 100 players as the validation set. The F1 score is selected as the evaluation metric. The ablation study for entity resolution is depicted in Table 2. Where our approach secures an F1 score of 1.000 for the validation set.

For extrapolated prediction, We use a sliding window of 200 matches. For instance, match number 1 to 200 are used to construct a sub-KG, and we perform prediction based on this sub-KG for match number 201 to 400. We use the F1 score and Mean Average Error (MAE) to access the performance for the two prediction tasks respectively. The performance of ablation study for these two tasks are shown in Table 3 and Table 4. Where our approach demonstrates its efficacy of exploiting multiple attributes and holds a performance better than baseline methods.

5.2 Qualitative Evaluation

The qualitative evaluation is five-folded:

(i) **Data Extraction and Cleaning**: Generate error dump file with known patterns, inducing problems by type for further amendment.

(ii) **Data Curation and TTL Triple Generation**: Use SPARQL queries to examine triples and distinguish problems by patterns.

(iii) **Ontology Correctness**: Use online RDF validation tools (e.g., RDF validator) and visualization tools (e.g., webvowl).

(iv) **KG Quality**: Manual check for the quality dimensions (e.g., intrinsic, contextual, and representational, etc.), including (a) data for the latest MLB season is extracted with sufficient breadth and depth in the domain; (b) additional ontologies are defined to be interpreted by human and machines easily without ambiguity; and (c) free of error with a consistent representation in URI and links to trustworthy resources.

(v) **KG Usefulness**: Support both user-defined and pre-defined sets of SPARQL queries.

6 CONCLUSION

This project builds a baseball KG and aims for three major research question including one for entity resolution and two for extrapolated predictions. In general, we build custom ontologies with trustworthy interlinks and apply KG for capturing complex relationships in baseball matches, which supports up-to-date and free-of-error data with consistent representation. Furthermore, we provide an easy-to-use user interface with multiple querying approach, so that user can exploit data with sparse relationships in KG for extrapolated prediction.

To be more precise, we have applied different approaches in solving technical challenges that could be faced when building a knowledge graph based on real-world data. First, for data crawling, extraction, and cleaning, one not only needs to get familiar with the complex and dynamic webpages, but also has clear image of what parts of the contents are required to be extracted. With full knowledge of the targeted domain (e.g., baseball in our case) and clear image of the expected output, one can save a lot of time by focusing on only the data needed.

Next, we believe that the design of ontology can heavily affect the complexity of composing SPARQL queries. Therefore, we have invested a huge amount of effort in designing our own custom ontologies and corresponding TTL generation that can not only capture the complex relationship in our targeted domain, but also facilitate the easiness and intuitiveness of querying through SPARQL. On the other hand, as one of the key qualitative indicators, securing the KG quality is a crucial issue. We perform entity resolution by exploiting relevant attributes, and make sure that all the data are free of error under a consistent representation. Although the exact attributes being used heavily depend on the domain that one is conducting the entity resolution, there are several general techniques that can be adopted (and some of them are also used in our case) such as blocking, which can save valuable computational resources.

Last but not least, we believe that applying KG makes itself somewhat more easy to be navigated and explored compared to relational database. For our targeted domain, this allows users to fully exploit the complex yet sparse relationship. Furthermore, to the best of our knowledge, there is no existing ontologies or systems that provides easy access to such intriguing relations behind this tempting sport. And our KG opens a door for users to dig into it.

A APPENDIX

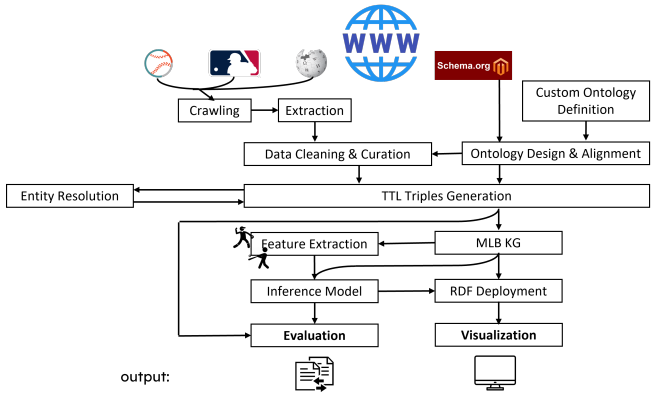

Figure 1: The project architecture.

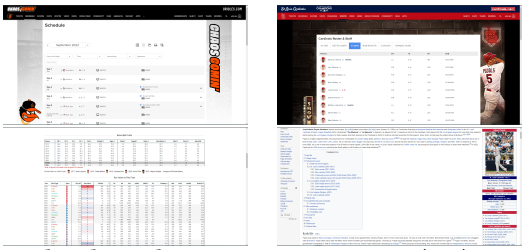

Figure 2: The three data sources: MLB.com (top), Baseballsavant.mlb.com, and Wikipedia.org (bottom, left to right).

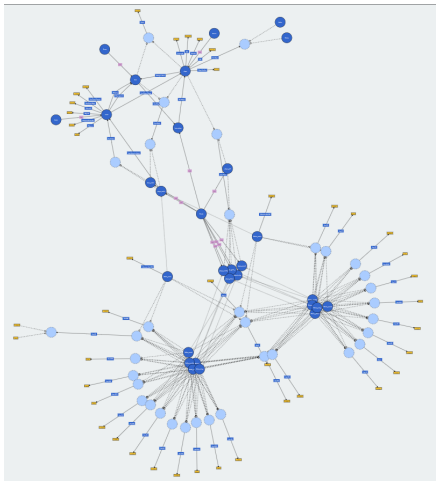

Figure 3: The overview of our custom ontology.

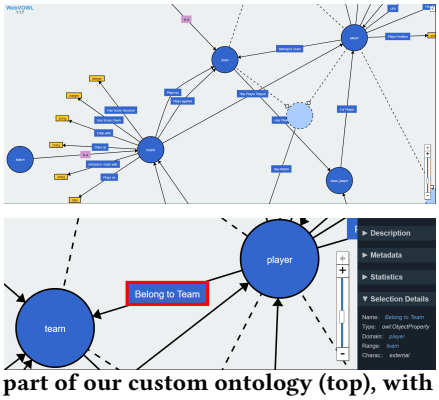

Figure 4: A part of our custom ontology (top), with the defined properties, domains, and ranges (bottom).

Table 1: Statistics of our custom ontology.

Type	Node	Edge
Ontology itself	107	248
Individual imported	11,027	104,690

Table 2: Ablation study for entity resolution.

Approach (Ablation)	F1 Score
Ours	**1.0000**
- team consistency & transfer	0.9597
- string similarity	0.7299

Table 3: Ablation study for match-result prediction.

Approach (Ablation)	F1 Score
Ours	**0.6639**
- team embedding	0.6597
- recent performance	0.5770
- opponent's performance	0.5413
baseline (random value)	0.4951

Table 4: Ablation study for player-performance prediction.

Approach (Ablation)	MAE
Ours	**0.6213**
- recent-performance decay	0.6700
- player embedding	0.7326
- recent performance	0.7441
baseline (average value)	0.8049

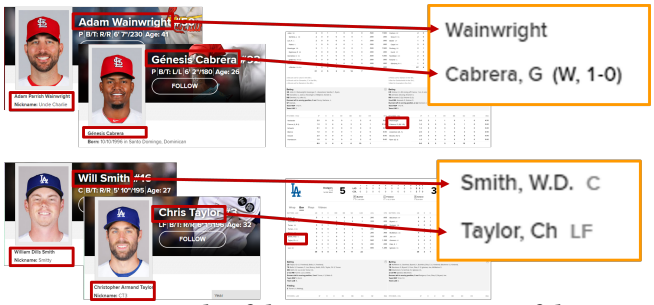

Figure 5: An example of distinct representation of data across sources, where last name or last name with the first letter from first name is used (top); while some rely on last name with multiple letters from first name, or letters from full name (bottom).

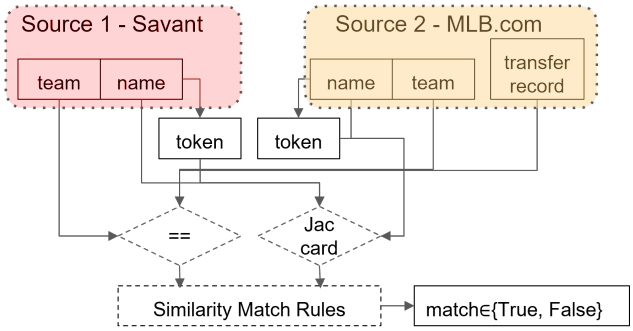

Figure 6: An example of the entity-resolution workflow.

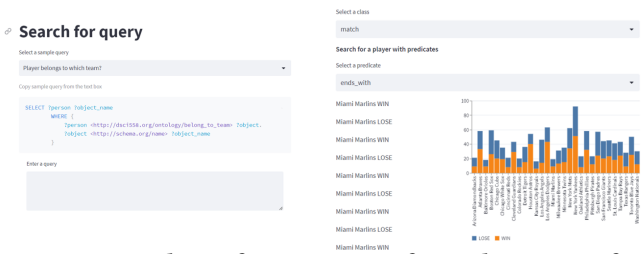

Figure 7: Screenshots of our user interface, where a set of predefined queries (left) and user-defined combinations of subjects and predicates (right) are two of the querying approaches in our system.