Data Integration of Football Players and Teams

Group Project - Web Data Integration HWS18

presented by
Ahmad Altaweel 1581320
Anjeza Gjuzi 1623356
Bora Basha 1622319
Essam Azzam 1584169
Rafik Takieddin 1584090

submitted to the
Data and Web Science Group
Prof. Dr. Christian Bizer
Anna Primpeli
Oliver Lehmberg
University of Mannheim

October 2018

Contents

Chapter 1

Introduction

Over years sports have had either a decrease or an increase in popularity. However, the most time-resistant and favorite one has remained soccer, with over four billion fans worldwide. There are professional leagues, semi-professional leagues, amateur leagues, youth leagues, womens leagues and many more. Even only considering the professional leagues, there are tens of them. For this reason we decided to work on this project on creating an integrated and consolidated data set of players and teams across different professional leagues.

The integration is performed on several data sets that were collected from various sources and in different formats. The data used originally come from Fdration Internationale de Football Association, also known as FIFA, which is recognized as the international governing body of football. Data is also collected in different years, 2017 or 2018, and each data set serves different purposes independently.

The goal of this project is to provide an overall collection of all the important information regarding players and teams for all the football fans worldwide, so that they can easily find correct data for their favorite sport.

Chapter 2

Data Collection and Data Translation

Initially, six data sets of different formats (CSV, JSON, HTML table) were selected to work on for this project. In total the data sets contained more than 300.000 records describing players, teams and coaches. Accordingly, the classes that were selected were players, teams and coaches. In this first attempt in data collection, the goal was to have 16 attributes in the joint data set, 11 of which were contained in more than one data set.

2.1 Data Collection

Not all data available can serve a certain goal. In order to have a good target schema it was necessary to screen through the initial collection of data to decide which of the data sets served the purpose of the project. As a result, the data set about coaches was excluded for its lack of meaningful information to map the records with respective teams and the players of that team.

For the following phases of the project, five data sets were used in total. Two data sets were found on kaggle.com. The first is a SQLite file that contains different tables about players and teams in the European Professional Football. The tables were generated in October 2016. The second one contains data for every player that is part of FIFA 2018, including personal data such as name, nationality, age, salary, and player style statistics such as dribbling, aggression, acceleration etc. This data set was generated in 2017. The third one is a JSON file generated from DBpedia regarding players with attributes such as player name, country of birth, team, country of team etc. The fourth data set is a csv file that contains data about team regarding their API, name, abbreviation etc. The fifth data set is an XLSX file

generated from a table in PDF format of all the players in 2018 FIFA World Cup Russia. The table contains attributes such as player name, national team, position, club name etc.

DataSet	Source (*)	file name	format	class (**)	# of entities	# of attributes	list of attributes (***)
Kaggle.com	Download dataset	Player.csv	CSV	Players	11059	5	player_api_id, player_name, player_fifa_api_id, birthday, height, weight
Kaggle.com	Download dataset	CompleteDataset	CSV	Players	1798	74	Name', 'Age', 'Photo', 'Nationality', 'Flag', 'Overall', 'Potential', 'Club', 'Club Logo', 'Value', 'Wage', 'Special', 'Acceleration', 'Aggression', 'Agility', 'Balance', 'Ball control', 'Composure', 'Crossing', 'Curve', 'Dribbling', 'Finishing', 'Free kick accuracy', 'GK diving', 'GK handling', 'GK kicking', 'GK positioning', 'GK reflexes', 'Heading accuracy', 'Interceptions', 'Jumping', 'Long passing', 'Long shots', 'Marking', 'Penalties', 'Positioning', 'Reactions', 'Short passing', 'Shot power', 'Sliding tackle', 'Sprint speed', 'Stamina', 'Standing tackle', 'Strength', 'Vision', 'Volleys', 'CAM', 'CB', 'CDM', 'CF', 'CM', 'ID', 'LAM', 'LB', 'LCB', 'LCM', 'LDM', 'LF', 'LM', 'LS', 'LW', 'LWB', 'Preferred Positions', 'RAM', 'RB', 'RCB', 'RCM', 'RDM', 'RF', 'RM', 'RS', 'RW', 'RWB', 'ST'
Datahub.io	Download dataset	Team	CSV	Teams	298	7	name'' 'league', 'rank', 'prev_rank', 'off', 'def', 'spi'
DBpedia	dbpedia.org	players from db pedia.json	JSON	Players	3861	4	soccerplayer, CountryOfBirth, Team, Country of Team
FIFA.com	Download dataset	New Player-Team dataset	CSV	Players	735	8	team, position, fifa popular name, birth date, shirt name, club, height, weight

Figure 2.1: Table of Datasets

2.2 Project Requirements

After redefining and reselecting which data sets were useful in the previous stage, in the final five data sets, there is a total of around 34.000 entities. The classes to be represented in the target schema are two: Players and Teams. In order to analyze the overlapping of entities between data sets, a sample was built from two data sets regarding players. The sample included 1000 random records per each data set, more than 200 of which were contained in both data sets giving an estimation of 20% expected overlap across all data set. Statistically speaking, in more than 33.000 records, it is expected to have an overlap of approximately 6.000 records contained in at least two data sets.

Class name	Attribute name	Datasets in which attribute is found
Player	First Name	SourceOne, SourceTwo, SourceThree, SourceFive
Player	Last Name	SourceOne, SourceTwo, SourceThree, SourceFive
Player	Nationality	SourceTwo, SourceFive
Player	Birthday	SourceOne, SourceFive
Player	Club Name	SourceOne, SourceTwo, SourceFive
Player	Height	SourceOne, SourceFive
Player	Weight	SourceOne, SourceFive
Player	Acceleration	SourceOne, SourceTwo
Team	Club Name	SourceFour

Figure 2.2: Attributes contained in different data sets

Before constructing the target schema the data sets were checked for missing values. Overall, there is a range between 0 - 5% of missing values in the files. In all the data sets there is a total of over 150 attributes. There are 19 attributes in the target schema, without including the player_id attribute that was generated automatically, of which 17 are contained in at least two data sets.

2.3 Data Translation using Mapforce

The first step was to consolidate all the data sets into one target schema, having one format (XML). Hence the attributes that are most relevant to the project are as follows:

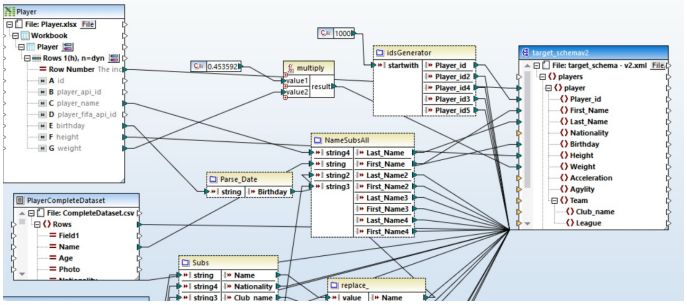

Figure 2.3: Target schema in MapForce

- ID Generation: The aim was to translate 5 data sets into the above-mentioned target schema. The first step was to generate IDs to identify each entity per each data set, to maintain a unique identifier across all data sets. The following format was used for IDs generation: Playerxlsx_1000, where Playerxlsx is the data set name and 1000 is the first player ID for that data set.
- Name Separation: Most of the data sets have the name attribute containing information about full name (Firstname and Lastname). Some records included also middle names, abbreviations for the firstname or only the lastname of the player. To have a more consistent representation of the entities in order to produce better results in the next phase, the substring function is applied to split the Name attribute into Firstname and Lastname attributes in the target schema.
- Date Parsing: Two data sets contained the Birthday attribute. However they were presented in different formats. Parsedate function formats the given

data. In the target schema ”Birthday” was set to yy-mm-dd.
Example: (input: 29.02. 1992 00:00, output:1992-02-29)

- Lb to KG transformation: For the Weight attribute, in the target schema all the values were converted to Kilogram in order to have consistency across data sets. Hence all the weights that were in lb were converted to kg. The multiply function was used to multiply the weights in the Playerxlsx data set with 0.453592 (The lb to kg conversion rule). The results were the players weights in kg, that was then mapped to the target schema.

- Uri to String: The Playersdbpedia data set is a JSON file. Thus all the entries are represented in URIs, having underscores (_) to separate strings and quotation marks ( ) enclosing each string. Several functions had to be applied to extract the relevant data out of each URI, replace the underscores with spaces. Example: Input : {”soccerplayer”: { ”type”: ”uri”, ”value”: ”http://dbpedia.org/resource/Abbas_Ahmed_Khamis”}
Output: (First name: Abbas) and (Last name: Ahmed Khamis) The following is a representation of the final schema mapping:

Chapter 3

Identity Resolution

The players competing in the important professional leagues change every year. However most of the players keep playing football for many years for different teams and clubs. Thus, they appear in at least one of the data sources that were used, and some even appear in all of them. The goal of Identity Resolution is to identify when two different records in separate data sets describe the same real-world entity. For the two classes in this project, Identity Resolution was performed only for the class Player in order to find correspondences across data sets that contain information about the same player.

3.1 Gold Standard

To construct the gold standard, four data sets were combined two by two for each possible combination to find correspondences. The gold standard was built by considering positive examples (roughly at 25%), negative ones (around 50%) and corner cases (around 25%). In the cases where there were less than three attributes in common for two data sets (example: Playerxlsx and PlayerComplete) it wasn't necessary to build a gold standard, as it would be difficult to tell if two records refer to the same player.

For the positive examples, when the records of two players matched in at least three attributes, the ID pairs were manually labeled with true. The main weight in the comparison was carried by attributes such as "Firstname" and "Lastname". To ensure that the two records represented the same player comparison was drawn even between attributes such as "Nationality", "Team", "Height" or "Weight". In the cases where the attributes of two player IDs didnt match for most of the attributes, the pair was manually labeled with false as a negative example.

However, there are records that are categorized under false positives or false

negatives. In the false positives, the attributes of two players would seem to match when comparing names, mainly when the name feature would contain initials, or when the player had a very popular name (with several players having the same first name or last name), also including a match in the nationality attribute. In this case further examination of attributes was required to determine that it wasnt a match and had to be labeled with false. In the false negatives case, attributes such as team (due to the fact that players change teams every few years), height, weight, or misspellings of the name, would make it seems as a non-match. Careful evaluation of the attributes was required to determined that the two records indeed referred to the same player, and the pair was labeled with true.

In total there are three gold standards built, ensuring that each data set that is later going to be used in the data fusion task is covered by at least one gold standard. Two of the gold standard combinations contain around 150 pairs that were manually labeled and one contains more than 200 pairs. In total more than 500 manually annotated pairs cover all the data sets regarding Player class. Since there are more than 100.000 records in total, having a larger gold standard would help to achieve better results. But such task requires a lot of manual labor, thus the gold standard constitutes around 500 pairs.

3.2 Creating a matching rule

Players attributes like ”Club Name”, ”Weight”, etc. may change over time. Therefore more consistent attributes like First Name, Last Name, Height & Nationality have higher chances of yielding better correspondence results. Therefore the comparators used for correspondences were as follows:

- PlayerDataComparator1Year: comparing the birthday year of the players and give a match if the difference less than 1 year, because there are some datasets the month and the day of the birthday.
- PlayerFNameComparatorJaccard: comparing the first name of the players based on Jaccard similarity.
- PlayerFNameComparatorFirstLetter: comparing only the first letter from the first name and give match with exact value.
- PlayerLNameComparatorJaccard: comparing the players last names based on Jaccard similarity.
- PlayerLNameComparatorLevenshtein: comparing the players last names based on Levenshtein similarity.

- PlayerNationaltyComparatorEqual: comparing the players Nationality and give match with exact value.
- PlayerHeightComparator5cm: comparing the height of the player with max 5 cm.
- PlayerWeightComparator10kg:comparing the weight of the player with max 10 kg.

3.3 Blocking strategy

The matching phase is a many-to-many process which includes (n * m) / 2 record comparisons, thus the use of blockers was necessary. There were 2 blocking strategies used: the first one was to block based on the first 2 letters in the last name, the second was to block based on the nationality (have lower variance). The runtime for using a Noblocker strategy took up to 5 minutes or a complete crash. However after applying the blocking strategies, it took on average around 1.001 - 3 seconds between different Identity Resolution runs, while for machine learning runs it took up to 8-9 seconds.

3.4 Identity resolution using Linear Combination

After defining the matching rules & the blockers the identity resolution was run, using manual combinations of different weights and thresholds, till getting the best precision, recall & f1 scores based on the gold standards. In the beginning the gold standards were not read correctly, hence the logger was used to trace the problem. After multiple runs the best results achieved were as follows:

data set Linearcomb	2_5	1_5	2_3
Precision	1,00	0,3846	0
recall	0,9800	0,1042	0
F1	0,9899	0,1639	0

Figure 3.1: reusult matching rules1

Matching Rule	Blocker	P	R	F1	#Correspondences	Time
Rule1: FirstName, LastName,Nationality	PlayerBlockingKeyByTitleGenerator	0	0	0	3760	1s
Rule2: FirstName,LastName,Height	PlayerBlockingKeyByNationality	1	0,9800	0,9899	87093	3,1s
Rule3: FirstName,LastName,Height	PlayerBlockingKeyByTitleGenerator	0,3846	0,1042	0,1639	20484	2,0sc

Figure 3.2: reusult matching rules2

3.5 Identity resolution using Logistic Regression (Machine Learning)

As the results for 2_5 were satisfactory enough, there was no need to run it again using machine learning. However there was potential for improvement with 1_5 as the results were not satisfactory enough. Applying linear regression could help yield better results. This time instead of assigning and tuning the matching rules, the training set from the gold standard was fed into the rule learner, then all the possible comparators are added to the matching rule, after running the Identity resolution, the learner provided the best weights to yield the best results. The new scores were as follows:

```
*
*       Evaluating result
*
Player2 <-> Player5
Precision: 0,5942
Recall: 0,3388
F1: 0,4316
```

Figure 3.3: Machine Learning Results

Chapter 4

Data Fusion

The last phase of the project consisted of resolving the conflicts between the records that represent the same real-world entity but present different values and merging the data. For this phase the output from Identity Resolution served is input. The attributes for which fusion is performed are First name, Last name, Nationality, Height and Weight attributes. First step is building a gold standard with 24 entities taken from the output of Identity Resolution, in order to ensure that these entities have matches. Manually extracted information from external sources such as Google or Wikipedia are used for attributes like First name, Last name, Nationality, Birthday, Club name, Height and Weight.

4.1 Java Template

The first step was making the object model fusible, by extending the AbstractRecord class, which implements both Matchable and Fusible. The Fusible class is used to provide meta data about the model for generating reports. Afterwards, FusibleFactory was implementedto create the fused objects, or in this case players.

Next step was creating the PlayersApp java class in accordance with the template used to load the data using the FusibleDataSet class. Specifically ds1 (Playerxlsx.xml), ds2 (PlayerComplete.xml), ds3 (PlayersDbPedia.xml), ds5 newplayerTeamData.xml) were added. Moreover, the data sets from Identity Resolution were represented as: 2_3_nodup.csv" between ds2, ds3; 2_5_nodup.csv" between ds2, ds5 1_5_nodup.csv" between ds1, ds5.

4.2 Conflict Resolution Strategies

Firstly, DataFusionStrategy class was used to define how each of the attributes is fuse. Next, it was added a Fuser and an EvaluationRule for each attribute. Since Fusers use a conflict resolution function to fuse the values for an attribute and the Evaluation Rules determine whether two values are equal, multiple fusers were created and tried different strategies in order to improve the accuracy step by step. Various challenges were faced during this process when wanting to create fusers for integers or numerical values in general due to restrictions requiring certain types. Also due to lack of examples in the numerical fusers such as Average and moreover for fusers accepting an object, The numerical attributes were represented as string, datatype independent functions and functions that use meta data. At the end, the achieved results were better when using Voting on ”First name”, Voting on ”Last name” and ”Nationality”, FavourSource on ”Height” and MostRecent for ”Weight”.

4.3 Results and Quality of Data Fusion

In the beginning of the data fusion process, the results were really low on accuracy, further analysis was required to understand the issue. In the output files from Identity Resolution, due to low threshold, there were a lot of correspondences. An increase of the threshold for all the matching rules gave better performance. To further improve the results, the apply maximum matching was used to remove the duplicated values and leave only the ones with the highest weights. Moreover,to further improve the accuracy some other strategies were used. At the end better results were achieved by using LongestString on first name, Voting on last name and nationality, FavourSource on height and MostRecent for weight.

Final results for accuracy were : Attribute-specific Accuracy: fName: 0.29; lName: 0.50; nationality: 0.46; height: 0.21; weight: 0.17;
General Accuracy: 0.33

	Option 1	Option 2	Option 3	Option 4				
General accuracy	0.23	0.28	0.32	0.33				
Attributes accuracy					Option 1	Option 2	Option 3	Option 4
First Name	0.25	0.17	0.25	0.29	Longest	Shortest	Longest	Voting
Last Name	0.33	0.5	0.5	0.5	Longest	Voting	Voting	Voting
Nationality	0.46	0.46	0.46	0.46	Voting	Voting	Voting	Voting
Height	0.08	0.17	0.21	0.21	Voting	Source	Source	Source
Weight	0.04	0.17	0.17	0.17	Voting	Voting	Recent	Recent

Figure 4.1: Accuracy of different conflict resolution functions

Chapter 5

Conclusion

The goal of this project was to finalize a data set to serve as a ”one-stop-shop” for all the football fans all over the world to get information quickly, correctly and fully for all their favorite players. It provides insights on the process from gathering these data to the final fusion of the player attributes.

Throughout every stage, it was important to redefine and reanalyze the outcome of the previous phase, in order to get clean results. Understandably so, attributes describing personal information about the player, such as name, nationality or birthday maintained their importance in every stage. This is confirmed during the translation of the data into the target schema, through identity resolution and finally in the data fusion.

Even though in the beginning there were almost 34.000 records describing the players and the teams, in the end there are around 1000+ records in the fused data set. This means that most of the records actually represented the same real-world entities. Also adding more attribute fusers or tailoring the fusers more accordingly to the data sets' needs would provide more and better results. These results would provide a good starting point for further research.