703804: Programming Lab: Innovative Interaktion, Visualisierung und Analyse Final Report

David Kofler David Westreicher Matej Stanic

June 23, 2015

1 Motivation

Over the past decades football has become the world's most popular sport with over 250 million players in 200 countries. Including over 3 billion fans worldwide, it has also become a multi-million dollar business. One of the main income sources of football clubs are transfers. Huge amounts of money are spent for transfer fees. For football fans the transfer circus is always of big interest, but with approximately 12000 transfers per year it is difficult to keep track of every single transfer. Also, football fans are interested in transfer rumors and statistics. This is the starting point of our project. Basically, the goal of the project is to visualize data concerning football transfers including twitter data, which is interesting for football fans.

2 Problem Definition

The single components of the project can be summarized as following:

- **Visualization of the football transfer graph.** Using data from `www.soccerbase.com` the goal is to construct a graph of player transfers between clubs. Intuitively, the graph is a directed graph, where the nodes are the teams and the edges have the players and the transfer fees as attributes. Our visualization idea builds up upon the graph from `www.transferwindow.info`. Initially, the graph is visualized in a geographical world map. The user then can click on a league/team/player to show only transfers of the league/team/player. Also, clicking on one of those three would provide additional statistics. Furthermore, an extended filter can be applied, e.g to show only transfers of players which are under 21, or players which are from Spain.
- **Analysis and visualization of team/player statistics from the football transfer graph.** The transfer graph can be used to analyze

certain behaviors of players and teams using graph analysis algorithms. For example, depth search can be used to find players which left a team and then came back after some years. Or, using the *PageRank* algorithm[1], teams can be found which are of high importance when it comes to transfers. [5] give an overview of graph algorithms, that are also interesting for this project.

- **Calculation of popularity of teams using twitter.** Each football team has an official twitter account where they post news, events, etc. When clicking on a team in the football transfer graph, statistics of their twitter account would be presented, including the number of followers. Unfortunately, the number of followers doesn't reflect the true number of fans. [9] propose a method to identify user interests from tweet times, which are compared to the tweet times of events concerning the team. This algorithm can therefore identify true fans.

- **Inclusion and analysis of transfer rumors from different sources.** Using twitter and/or news sites we will include transfer rumors. [8] have used different machine learning methods to find tweets that refer to scheduled or unscheduled events like transfer rumors. We can directly use their approach in our project, but for unscheduled events the performance is poor.

- **Classification and visualization of fan opinions.** The project will also include user opinions about transfers. [10] have proposed a model for classifying user tweets as positive, negative or neutral using Support Vector Machines. This model can also be applied to transfer opinions.

All in all the project should offer football fans a whole new level of transfer investigation. The inclusion and analysis of twitter data combined with a intuitive visualization is the key innovation compared to other projects.

3 Survey

This section gives an more detailed survey of the single sources of the project. Every group member has chosen at least three sources.

3.1 David Kofler

- [6] use Neo4J to store data about social interactions which was collected during a conference by using wearable proximity sensors. They present some example queries and discover that querying over densely-connected nodes makes it difficult to achieve high performance. This is something we have to keep in mind when we create our data model.

[1] `http://en.wikipedia.org/wiki/PageRank`

- [7] contains an interesting algorithm to find communities of clubs between which players often circulate. Many important real-world networks have the scale-free property (we assume that the football transfer network too), but detecting communities within them is not always easy because sub-communities tend to be classified as communities on their own. This paper presents an algorithm specialized for scale-free networks.

- `D3.js` (`http://d3js.org`) is a JavaScript framework to create animations on webpages. It is similar in usage to other popular JavaScript libraries (like jQuery and Prototype), but contains special features to create animations. We will use it to create our visualizations.

3.2 David Westreicher

- `http://eyeseedata.com/football-player-transfers/` is a blogpost describing the visualization of global football transfers. It contains a video which shows a map of the world in which every transfer is shown as a edge between two countries. The video progresses through the years 1900 to 2013. Then there is also a interactive map where you can click on a country and see the ingoing and outgoing transfers of the selected country. The last interactive part is a statistic of the number of transfers and their fees, grouped by countries. These visualizations give a good idea of how such a map would look, but the difference to our project is that we don't group transfers to countries but to clubs and we will combine the transfers with social data.

- `www.transferwindow.info` is a interactive visualization of football transfers of popular leagues. The main idea here is to represent the transfers by a graph between leagues or teams. On the right side of the site there is also information of the top transfers in the currently selected view. The site is visually appealing and gives a good glimpse of how a beautiful style for graphs is achieved. Our project will however lay out the graph on top of a world map.

- [9] is a interesting paper about extracting interests from twitter users solely by the time they write their tweets. The idea is to connect the time of the tweets to the time of external events. We could use such an approach to find out if a user is a fan of a certain team. For example if team A currently has a match with team B and the user T tweets while they are playing. Then the user T is likely a fan of A or B.

3.3 Matej Stanic

- `www.soccerbase.com` is a football news site that not only contains news but also many statistics about leagues, teams and players. This includes also a detailed transfer history for every team and player. It's a site where football fans can keep up to date. For our project, detailed transfer

information is inevitable in order to build a correct graph of transfers. Soccerbase.com offers exactly that type of transfer data that is needed to build such a graph. By scraping we can directly get the data that we need.

- [10] have evaluated a method for sentiment analysis for Twitter posts. They implemented a crawler to obtain twitter posts, which then where classified as positive, neutral or negative using a standard Support Vector Machine (SVM). The SVM was evaluated on a predefined dataset using cross validation. The average accuracy of their method was 70.592%. For our project, the same approach can be used for transfers and/or transfer rumors. After extracting football fan opinions on transfers from Twitter they could be classified as positive, negative or neutral. This would somehow be a measure of acceptance between the fans and the owners and managers of the club. There may be one problem, which is assigning tweets to transfers. This could be solved by only taking into account tweets which are referencing the transfer tweet.

- [8] evaluate methods which identify tweets referring to scheduled or unscheduled events. Their case study consists of two scenarios: The first one are football matches which are scheduled. On the other hand they also used an unscheduled scenario, football transfers. They have tested several state-of-the-art machine learning methods for their scenarios. The results are twofold: For the scheduled event scenario, the accuracy for some methods was above 80% while for the unscheduled scenario the accuracy was near-baseline. This means that even though the machine learning methods can directly be used for our project, it would not be productive because of the poor performance.

4 Proposed Method

Our solution consists of five parts: the crawler, the geocoder, the transfer rumour crawler, the sentiment analysis module and the website. The crawler is responsible for collecting the football transfer data and storing it into the database. The geocoder researches where the home ground of each team is, and adds that information to the database. The transfer rumour crawler collects transfer rumour data and corresponding opinions from various sources which are then taken for sentiment analysis. The website finally renders the data on a 3D globe. Unfortunately, we were not able to finish all of the proposed features from Section 2. Unfinished and future work is described in Section 6.

4.1 Crawler

The crawler which pulls the data from the website is written in Java. It uses the `jsoup` library [1] to fetch HTML websites and to parse them. CSS Selectors can then be used to extract the content from the HTML DOM tree. By using Dependency Injection and the Observer model the crawlers for the entities Team,

Player and Tournament can be composed, which eases testing the components a lot. Also, since there is no monolithic component which does everything, the crawlers can be used independently from each other, for example to update the data of only a particular entity.

Implementing the crawler was a bit tricky because not every web page contains all the data. Also, it was not obvious how to store various data, for example the transfers. The web page only contains the contracts a player has with clubs. The transfers have to be calculated, because there are overlaps in the contract periods.

The crawler takes a lot of time to run since there should be a short time delay (at least 10 seconds) between each access to the website, else the website administrator might take measures against a perceived attack.

The space needed to store all the data is vanishingly small (a few MB). The largest amount of it is data about the players and their career history. Neo4j is well-suited to our data model and is easy to use.

4.2 Geocoder

The Geocoder is a Python script and uses the `py2neo` library to access the database. It then uses the OpenCage geocoder [3] to look up the coordinates of each team's home ground and to store that information in the database. If no home ground information is available (which is the case for many teams) the team name is used as input for geocoding. Since there were a few bugs in the OpenCage geocoder bindings for Python the script has to use Python's built-in `urllib` library to access the web service. The correctness of the OpenCage geocoder is evaluated in Section 5.1.

4.3 Website

We use NodeJS [2] for implementing the server-side part. We tried to use Java web frameworks, but compared with NodeJS they are very complicated to set up and work with. Also, nowadays it is easier to find webhosting providers for NodeJS than for Java web applications.

The website uses the ThreeJS library [4] to render a globe and dotted lines connecting points on its surface. This visualization will be used to render player transfers. There is also a search field with IntelliSearch, which means that it searches for possible search results while the user types. When the user selects a result then the view is narrowed to that particular result, for example only the transfers of a particular player are displayed.

4.4 Inclusion of Transfer Rumours

For the inclusion of transfer rumors we decided to crawl data from Twitter and `www.transfermarkt.co.uk` which we then visualized separately. For Twitter we crawled static sources including newspapers and accounts which directly address transfer rumours. The posts are visualized in the *Twitter* tab of our website and are not investigated further. In the case of Transfermarkt we crawled their *rumour mill* which is a forum containing the latest information about rumours. These are analyzed concerning their sentiment (Section 4.5) and visualized in the *Rumours* tab of the website.

The only sort of problems that needed intervention were ambiguities when matching the filtered rumours with the data in the database. Often, teams are abbreviated which means that they cannot be matched directly. To solve this we used a modified algorithm based on the Levenshtein distance (also called edit distance) when matching filtered team names to those in the database. The modified Levenshtein distance between two strings $a = a_1...a_n$ and $b = b_1...b_m$ is given by $d_{m,n}$, which is defined recursively:

$$d_{0,0} = 0 \tag{1}$$

$$d_{i,0} = i,\ 1 \leq i \leq m \tag{2}$$

$$d_{0,j} = j,\ 1 \leq j \leq n \tag{3}$$

$$d_{i,j} = min \begin{cases} d_{i-1,j-1} & +0, \text{ if } a_i = b_j \\ d_{i-1,j-1} & +1 \text{ (substitution)} \\ d_{i,j-1} & +0 \text{ (insertion)} \\ d_{i-1,j} & +1 \text{ (deletion)} \end{cases} \quad 1 \leq i \leq m, 1 \leq j \leq n \tag{4}$$

The obvious modification compared to the original Levenshtein distance is the missing penalization of the insertion operation. This suits our case as team names often come abbreviated which should not be penalized by higher distance. An example would be matching *Tottenham* with *Tottenham Hotspur*. Originally, the edit distance is 8, as 8 insertions are needed. In our case the edit distance is 0 as insertions are not penalized. Our observations have shown that the matching works correctly but issues cannot be excluded.

4.5 Sentiment Analysis

For the sentiment analysis we use the posts that are associated with the transfer rumours from Transfermarkt's rumour mill. The sentiment analysis module is based on the AFINN-111 wordlist[2]. The wordlist contains 2477 English words and phrases, each manually rated with values between -5 (negative) and 5 (positive) that reflect the sentiment.

[2] `http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010`

Each post is then analyzed word by word. It is checked whether a word is contained in the AFINN-111 wordlist. For each such word the sentiment values are summed up and an overall score is obtained that represents the sentiment of the whole post. Finally, the mean score of all transfer posts is calculated which is then used as final transfer score.

The results are visualized using different colors depending on the obtained score: green for positive, white for neutral and red for negative. The correctness of the sentiment analysis module is evaluated in Section 5.2.

5 Evaluation

5.1 Geocoder

The correctness of the geocoder module is essential for our project as wrong team locations falsify the visualization. Therefore we evaluated the geocoder regarding wrong or missing results. Missing results can be directly measured: For 1165 out of 5356 teams of the database the geocoder couldn't find a valid location, which is 21.75% of all teams. This is due to missing ground info, and also due to the fact that the home cities cannot always be inferred from the team name (e.g. Schalke 04 is from Gelsenkirchen).

On the other hand evaluating wrong locations requires manual verification. So we chose a sample size of 250, reaching a confidence level of 90%. Out of those 250 teams, 147 (58.8%) were labeled correctly, 44 (17.6%) were assigned wrong and 59 (23.6%) were missing. The main reason for wrong locations were ambiguities concerning town names. Especially English teams often had been assigned a same-named town in the USA. The same problems occurred for other English-speaking countries as Scotland, Ireland, Australia, New Zealand and Canada. All in all geocoding doesn't seem to be suitable when it comes to football teams because it requires manual verification and for 40% of all teams the location has to be set manually.

5.2 Sentiment Analysis

Sentiment analysis is performed on the posts from Transfermarkt's rumour mill. Unfortunately, all the posts are purely informative and don't reflect the users' opinions. Considering a random sample size of 100 posts, 100% of the posts were informative posts with citations of newspapers with absolutely no opinion at all.

However, it is also interesting to analyze these posts. One would expect that the sentiment analysis of such posts should deliver neutral results (score near to 0) due to the neutrality of the newspaper articles. However, many of the transfers are labeled as positive, some even as negative. We tried to investigate the reasons of this phenomenon.

For our tests we chose a random sample size of 64 transfers. Of those 64 transfers 42 transfers (65.63%) have a score which is equal to 0. Furthermore, 8 transfers (12.5%) have a score greater than -1 and smaller than 1, which we labeled as *near neutral*. The rest correspond to positive (13 or 20.31%) and negative labeled transfers (only 1 or 1.56%). Taking into account the first two classes (neutral and near neutral) for measuring accuracy, the sentiment analysis classifier thus reaches an accuracy of 78.13% (50 out of 64 labeled correctly).

Trying to find the reason for the relatively high amount of positively labeled transfers we decided to investigate the content of these and to compare them to the other ones. In all 13 cases the reason is obvious: The posts refer to players' good performances in the past. We think that our observation opens new opportunities to use sentiment analysis for labeling past performances of players.

In order to improve the quality of the results, more opinion-based posts should be considered. One possibility is the inclusion of Twitter posts that contain the corresponding hashtags of a transfer. However, again, our observations have shown that there is a high percentage of information-only tweets that cannot be seen as opinions.

6 Conclusion

Even though we could not fulfill all of our targeted goals, we think that we have developed a nice tool for interested football fans who are now able to investigate the history of transfers in a new way. Also, we provide a new way of delivering transfer rumours which includes sentiment analysis based on informative posts.

Our evaluations reveal two main findings. First, geocoding is not suitable for our case. Even though it delivers many good results all of them need to be checked manually which is overhead compared to doing it directly. Second, sentiment analysis shows that even though informative posts should be neutral, many of them are labeled positive because of praising words towards players. This can be directly used to analyze past performances of players.

6.1 Outline/Future Work

This is an outline of done and unfinished work from the problem definition (Section 2):

- **Visualization of the football transfer graph:** Done. Could be extended with filter to only show special kinds of transfers, e.g. only transfers from France etc.
- **Analysis and visualization of team/player statistics from the football transfer graph:** Not done. Will be part of future work.

- **Calculation of popularity of teams using twitter:** Not done. Will be part of future work.
- **Inclusion and analysis of transfer rumors from different sources:** Done. Could be extended with a classifier that classifies whether transfer rumours are going to be realized.
- **Classification and visualization of fan opinions:** Done using sentiment analysis. The inclusion of Twitter posts would increase the quality of sentiment analysis as more opinions are available.
- **Other Possible Improvements & Future Work:**
  - Using data which is not focused on England.
  - Add missing locations and fix wrong ones.

References

[1] jsoup: Java HTML Parser. `http://jsoup.org/`. Accessed: 2015-05-03.

[2] Node.js. `https://nodejs.org/`. Accessed: 2015-05-03.

[3] OpenCage Geocoder. `http://geocoder.opencagedata.com/`. Accessed: 2015-05-03.

[4] three.js. `http://threejs.org/`. Accessed: 2015-05-03.

[5] Richard Brath and David Jonker. *Graph Analysis and Visualization: Discovering Business Opportunity in Linked Data*. Wiley, 2015.

[6] Ciro Cattuto, Marco Quaggiotto, André Panisson, and Alex Averbuch. Time-varying Social Networks in a Graph Database: A Neo4J Use Case. In *First International Workshop on Graph Data Management Experiences and Systems*, GRADES '13, pages 11:1–11:6, New York, NY, USA, 2013. ACM.

[7] Sorn Jarukasemratana, Tsuyoshi Murata, and Xin Liu. Community detection algorithm based on centrality and node distance in scale-free networks. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, HT '13, pages 258–262, New York, NY, USA, 2013. ACM.

[8] F. Kunneman and A. Van den Bosch. Leveraging unscheduled event prediction through mining scheduled event tweets. In N. Roos, M. Winands, and J. Uiterwijk, editors, *Proceedings of the 24th Benelux Conference on Artficial Intelligence*, Maastricht, The Netherlands, 2012.

[9] Dinesh Ramasamy, Sriram Venkateswaran, and Upamanyu Madhow. Inferring user interests from tweet times. In *Proceedings of the First ACM Conference on Online Social Networks*, COSN '13, pages 235–240, New York, NY, USA, 2013. ACM.

[10] Akash Shrivatava, Shweta Mayor, and Bhasker Pant. Opinion mining of real time twitter tweets. In *International Journal of Computer Applications*, 2014.