Popularity and Connections Between Overwatch League Players

Craig Bruce & Emerson Moniz

Wentworth Institute of Technology

Abstract:

This paper reviews the usage of Social Network Analysis to find the connectivity of followers in the Overwatch League. The information being analyzed will be limited, due to the time constraint of pulling data from the Twitter API. Therefore, the method chosen to find the quantity of users to analyze, will be determined by the minimum amount of friends and followers on each account; this will ensure an accurate sample dataset. Information will be analyzed as a subset of data about players in the Overwatch League extracted from Twitter. Thereafter, the information that is extracted from Twitter will be stored in a file. Thus, the information being analyzed will be presented in a directed graph where each node represents a person on Twitter.

Keywords: Twitter, Social Network Analysis, Overwatch, connectivity

Introduction:

For this project we chose to analyze the Overwatch League. The information being analyzed will be retrieved from the Twitter platform. Granted, there is a total of 12 teams and in each team, there are 5 players. The information we retrieve from the Twitter platform includes the follower’s username and date/time of follow. Thus, the information will be represented as a directed graph. Each node represents a username/person on Twitter and edges of the directed graph will be weighted. The weight will be based on the date/time of the follow between the follower and the player. Thus, the impact of this project will be to analyze the Overwatch tournament to view trends on the teams and follower accounts. Using the data gathered, the following questions will be answered:

* Who is the most popular player in the league?
* Who is the most popular player in their respective team?
* How many edges are between each teams group?
* What are some common followers between players?
* What are some common followers between teams?
* Who links groups together?
* How are spikes in followers compared to when the player appears in tournament?
* What is the transitivity among the followers who view the players?

The questions being proposed will be answered using the libraries Tweepy, Pandas, NumPy, Seaborn, MatPlot and NetworkX. There will be charts to display our findings, analyzing the trends/patterns and answer the questions.

Relevant Work:

In the beginning of the project, there was a scarce number of articles that were similar to the experiment. Some articles focused on the sentimental and content analysis of the Twitter followers. Although this was an interesting aspect to investigate, the authors of this project chose to analyze the structure of the players’ followers on Twitter. Some of the methodologies used in the retrieval of the data comes from trial and error of pulling the data from the Twitter API and guidance from Professor Othman.

Description of your project:

In order to reproduce this project, the user must request from Twitter a developer account. Furthermore, once approved the user can build an app and gain access to use the Twitter API. After obtaining access of the consumer key, consumer secret, access token and access token secret from the Twitter application. Use the Tweepy library to access the OAuthHandler method and use the users given access token, to retrieve the authentication access and start using the API to retrieve the information desired. In this case, the user should take an object-oriented approach and create a class that has a constructor that serves as its base authentication for the usage of the Twitter API. Retrieving information from Twitter is relatively easy however, the user should be aware of how much data you can extract using the Twitter API in a specific timeframe.

Thus, it’s suggested that before retrieving the number of followers and friends on Twitter for each account have a number of followers and friends you want to analyze on Twitter. For instance, the method used to find the minimum number of followers and friends was created by sorting the followers and friends of each account respectively to know how much data to retrieve. Consequently, after finding the minimum quantity of friends and followers you want to save and acquire the authentication you can start to create methods to retrieve the number of followers using Tweepy’s cursor.



Fig.1: Method used to retrieve the user’s followers returns dataframe

A restriction imposed by the Twitter API is the rate limit error, this error will not allow you to retrieve the data from the specified user. In this project the user may encounter the rate limit error when retrieving data from an account that has over 500 followers and friends [Get Friends Ids], which meant there was too much data being pulled from Twitter in a small timeframe [Rate Limiting]. In order to get around this error, you should implement a sleep method of 15 minutes in between each retrieval of 270 user’s information about following and friendship. Therefore, to retrieve the friends and followers in the user’s account, the user must store all the information in to a Pandas DataFrame with the columns ‘from’, ‘to’ and ‘created at’.

Once retrieving all the data for the specified user, the user should save this information from the Pandas DataFrame in to a text file. After the data collection process is complete, the user needs need to combine all the datasets and produce a master dataset. In order to answer one of the prompted questions in the introduction we need to find the maximum out degree of the entire dataset and display the username.



Fig 2: Method used to store the relationships of each player

After obtaining all the data and converting it into an edge list. Notably, the user can make a graph. The method follower\_timeplot works by looking at the predecessors of the node in question, which would be the player, a list of all the times that the follower relation was created. Using the list, data could be graphed and visualized. The times where followers spiked, and tournament days could now be compared. Popularity among players was found by using a method called common follower that worked by comparing all the indegrees. Common followers where determined by if a follower had more than 2 out-degrees because that means that they follow more than 1 player.

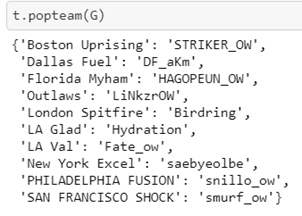


Fig 3: Result of displaying all the popular players in each team

Results and Evaluation



Fig 4: Result of displaying all the common followers and the first 100

When the graph was made most of the hypotheses about our data was proved. As shown in Fig. 4 there are many connections between Teams and players alike. Lots of connections in the graph are between other followers of one player and another player. The fact that the data sample was relatively small (0.01% of total data set) shows that it is high probability that the teams and players are heavily connected with common followers between them.

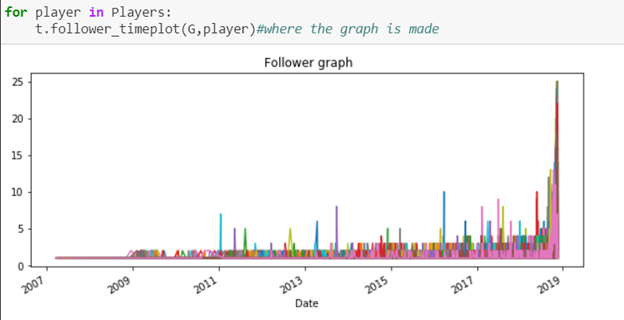


Fig 5: Result of displaying the spikes of common followers

Figure 5 shows the spike in followers over time for each player (each colored line) When looking at the graph and the graphs of independent players and compare them to game times, it is shown that the spikes match. So, a player follower count increases on the days he plays in official broadcasted games.

Conclusion and Future Work:

In this project I learned that the players’ followers are connected by the Overwatch league’s tournament schedule. One of the initial hypotheses in this project was that the players’ followers were connected. However, by using only a small dataset, one would think that there wouldn’t be any connections amongst each players’ followers in the dataset. Thus, by analyzing the reduced dataset to see the connectivity of each player. The analysis presented a common follower amongst the players. There was an increase in the player follower count on the days the players played in official games. Spikes in Fig.5 display the changes where there was an official broadcasted match. In a future experiment, one may investigate separate teams and see the transitivity of each player’s follower in comparison to other teams. In other groups this could be applied and could therefore look at the content of each individual player’s retweets and analyze the reactions (retweets or favorites) would make the follower more prone to follow the person being retweeted or favorited.

References:

API Reference Index - Twitter Developers.” *Twitter*, Twitter, 2018, developer.twitter.com/en/docs/api-reference-index.html

Get Friends Ids - Twitter Developers.” *Twitter*, Twitter, 2018, <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-friends-ids>

Get Followers Ids - Twitter Developers.” *Twitter*, Twitter, 2018, <https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-followers-ids>

Rate Limiting - Twitter Developers.” *Twitter*, Twitter, 2018, <https://developer.twitter.com/en/docs/basics/rate-limiting>