**Classification of NBA Fans with Twitter Data**

Jean Chakmakas and Lauren Kemperman

I. INTRODUCTION

*Background*

Characterizing sports fans of different teams and different levels of devotion is an interesting phenomenon that has not received a lot of attention in the past. Throughout a sports season as a particular team record flushes out, we expect that there are changes in who is supporting what team depending on several factors. In the NBA, we are interested in collecting data via tweets from Twitter to classify teams and fans. The data we collect is from top teams and bottom teams, and we are interested in the comparison of fans from these two divisions.

*Related Work*

There is some similar work done on interpreting Twitter data in sports applications. One paper focused on the analysis of sentiments during the 2014 World Cup [8]. Twitter data was used to analyze the emotions of U.S. soccer fans in conjunction with goals scored for them and against them throughout the world cup [8]. They found that the U.S. showed more positive emotions than negative ones, but were not able to conclude any patterns between emotion and responses to goals and losses [8]. Ultimately, they found the Twitter sentiment to be consistent with what was expected of fans [8].

Another paper focused on predictions in the NCAAB and the NBA [9]. The goal of this paper was to predict the outcome of basketball games. One algorithm used was naïve Bayes, which was found to be a good predictor of the NBA playoff series [9]. There were difficulties in generalizing between the two leagues, and each league required different classification biases [9].

Both papers reveal interesting topics and methods that we hope to combine in classifying different types of fans.

*Project Goals*

There are a few goals for this project. First, we are interested in characterizing fans of different teams. The following questions are important to explore:

1. Are the most popular teams the most liked?

2. Are the teams that are doing the best associated with positive sentiment, and the worst teams negative sentiment?

II. METHODS AND ALGORITHMS

*Data Collection*

Data was collected in the form of tweets by using the Twitter API [1]. A python script was created to collect tweets based on NBA keywords [2]. At the time of collection, we were interested in collecting data on two top teams and two bottom teams for comparison. The two top teams we chose to use were the Cleveland Cavaliers and the Toronto Raptors. The bottom two teams we chose to use were the Dallas Mavericks and the Philadelphia 76ers. When collecting data on the Cleveland Cavaliers, the keywords used were “Cleveland Cavaliers”, “Cavaliers”, “Cavs”, “LeBron James”, “Kevin Love”, “Tristan Thompson”, “J.R. Smith”, and “Kyrie Irving”. For consistency, the keywords used for each team were common ways to identify the team as well as the team’s starters.

*Naïve Bayes Classification*

Naïve Bayes classification was used in sentiment analysis to determine the sentiment of the collected tweets. This sentiment is defined as either positive or negative. Naïve Bayes uses conditional probability from a training set to predict the outcome of the test set. Bayes rule is illustrated below [3]:

This equation shows how the probability is calculated when there is a given tweet and a given classification. For example, the probability of a tweet being positive given that tweet is calculated with Bayes theorem [3]:

The probability of the tweet is constant, so it does not factor in to the equation [3]. The probability of the class, in this case, the probability of positive, is always going to be 0.5, because there are equal probabilities of a tweet being positive or negative. The neutral sentiment was not considered in our analysis. If a probability was found to be at least 0.5, it was classified as positive.

A training set was used to train the classifier so that we could find the probability of the tweet given the class (positive or negative). This training set contained a list of sentences that were already classified as positive or negative [10]. Each word in the classified sentences was considered, so when the test set of tweets was read in, a probability could be computed. With this probability of the tweet given the class, the conditional probability of the class given the tweet could be found.

The dataset we used to train the naïve bayes classifier was based on movie reviews. To ensure that there was not a domain bias being introduced, vaderSentiment [11] was applied to each tweet. VaderSentiment returned 4 values for each tweet: probability of positive, negative, neutral and combined. Our analysis focuses on text and twitter user clustering, and the results of this comparison were not further explored. However, it is evident from the following examples that this classifier has the ability to analyze complex social media sentiment data that may not be accurately predicted using a naïve bayes classifier.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tweet | Positive | Negative | Neutral | Compound |
| RT @dloadinghive: I feel like LeBron be losing on purpose just for a reason to act like a leader or sum https://t.co/NQuLnmqXyf | 0.212 | 0.11 | 0.678 | 0.34 |
| LeBron James honors his World Series wager with Dwyane Wade. (via @uninterrupted) https://t.co/RaUMEvPAO9 | 0.0 | 0.0 | 1.0 | 0.0 |
| DeMar DeRozan - Dual Game Worn Swatch - 2014/15 Immaculate #75/99!! https://t.co/hG2tYTVdtA https://t.co/5z5ES8BSBG | 0.0 | 0.218 | 0.782 | -0.4184 |
| DeMar DeRozan - Dual Game Worn Swatch - 2014/15 Immaculate #75/99!! https://t.co/hG2tYTVdtA https://t.co/5z5ES8BSBG | 0.0 | 0.218 | 0.811 | -0.2732 |

*Figure 1.* Example tweets and sentiment analysis output from vaderSentiment

*K-means Clustering*

The other algorithm we chose to use for gaining insight on the type of fans was k-means clustering. K-means clustering is a clustering technique that first assigns the number of centroids, k, to a space that sets each centroid ideally as far apart from each other as possible [4]. Next, each object is added to the cluster that has the closest centroid value to its value [4]. After each object has been assigned to a cluster, the positions of the centroids should be recalculated [4]. The objects will be reassigned to the centroid they are most similar to, and when the centroids no longer move, the algorithm is complete [4].

K-means clustering was used in two contexts. The first was for assessing the friends and followers of each team’s twitter account. Each team’s account names are represented in figure 1 below [7].

|  |  |
| --- | --- |
| **Team** | **Twitter Account** |
| Cleveland Cavaliers | @cavs |
| Toronto Raptors | @Raptors |
| Philadelphia 76ers | @Sixers |
| Dallas Mavericks | @dallasmavs |

*Figure 2.* Twitter accounts associated with each team.

This clustering was done in the R programming language [11], and a script was used to retrieve 500 friends of each account and 500 followers of each account [5] [6]. These friends and followers were grouped together, and the friend and follower count was retrieved for each user in this group. The logarithm was calculated for each of these values, because there can be very high friend and follower counts. Before taking the logarithm, counts of 0 were changed to 1, because the logarithm of 0 is undefined. To determine the number of clusters, a plot of sum of squares within groups versus the number of clusters was created. This allowed us to visualize the smallest number of clusters that still has a relatively small mean squared error. This is referred to as the “elbow” of the plot. For each of the four teams, three clusters were used. Lastly, the k-means clustering algorithm was run and a visual was produced for further analysis.

The second application of clustering was to see if words in NBA tweets suggested a grouping of topics. This analysis may be able convey what fans are interested in and what they are talking about the most. This will also be a good way to examine similarities and differences between the NBA teams that we are looking at.

K-means clustering was applied to 4 sets of team tweets separately, and then to all of the tweets merged together with labels corresponding to the datasets. The purpose of clustering for all of the teams separately is to get a topic breakdown for each NBA team. The purpose of clustering for all of the teams together is to see if there are teams that get clustered together, and look back at the original clusters to see if the breakdown makes sense.

The first step that was necessary before clustering was to remove words from the list of tweets that do not convey significant meaning. NLTK stopwords [13] was used to remove words such as “a”, “the”, or “in”. It was also important to remove hyperlinks and emojis from the data, because we did not focus on deciphering what those mean for our analysis.

Then, the tweets were split up into lists of words, which were used to create a dictionary. The dictionary was looked at and repetitive terms were removed for each separate clustering. We also went back and added more words and terms that were not interpretable to the list of stop words.

We then got a term frequency-inverse document frequency (tf-idf) matrix [14] by counting word occurrences by document. This evaluated the importance of the words in every tweet we looked at. Cosine similarity [15] was then used to approximate the similarity between each tweet.

Before applying k-means clustering, we took 1 minus the cosine similarity to plot our data points on a Euclidean plane. This was a good way to visualize how many natural clusters would fit the data. This was then chosen as the value of k, and the top terms in every cluster were examined and consecutively labeled with an over-arching topic. For the merged clustering with the combined tweets, the clusters were labeled by the teams that were assigned to that cluster. This proved to be more useful, as we have difference frequencies of tweets for each team. This was causing the frequency of terms from those datasets to dominate the clusters.

III. ANALYSIS AND DISCUSSION

WordClouds [16] were generated to visualize frequent words in tweets that appeared in the data. This was helpful because it verified that the data were relevant, and gave a sense for what people tweet about the NBA and how tweet content differs between teams. This information will be useful when the clusters of tweets are analyzed later on based on their content.

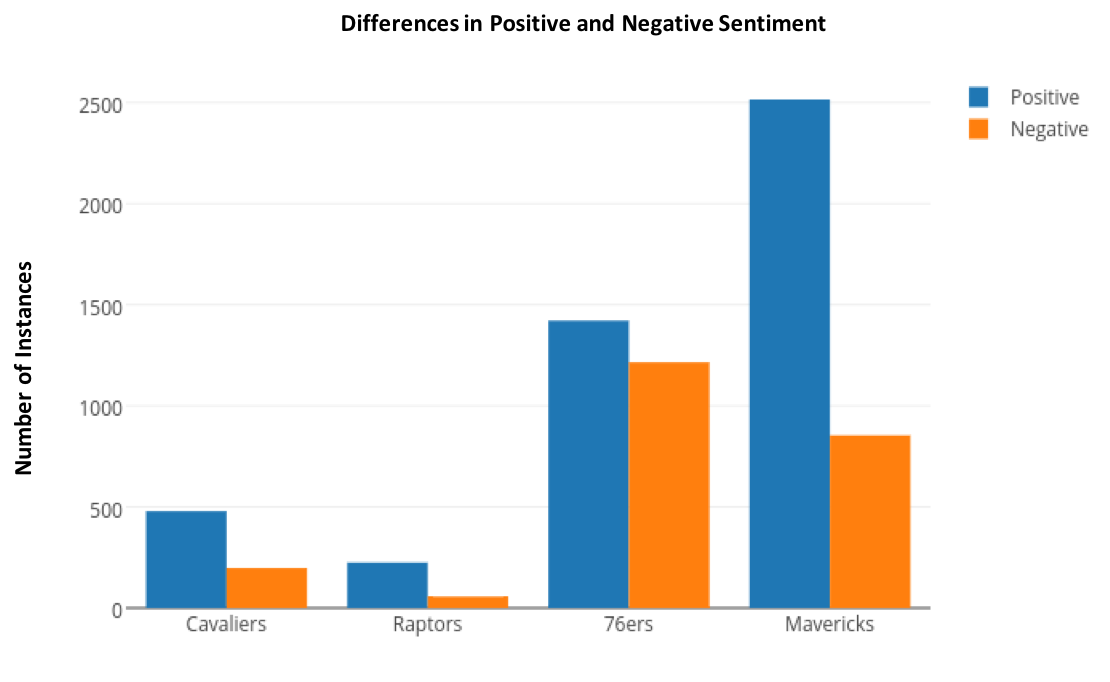


*Figure 3.* WordCloud from tweets about 76ers



*Figure 4.* WordCloud from tweets about Mavericks

When naïve Bayes classification was used to predict the sentiment of tweets between teams, there were not any direct trends that were evident. Figure 2 shows a graphical representation of how this sentiment data was distributed.



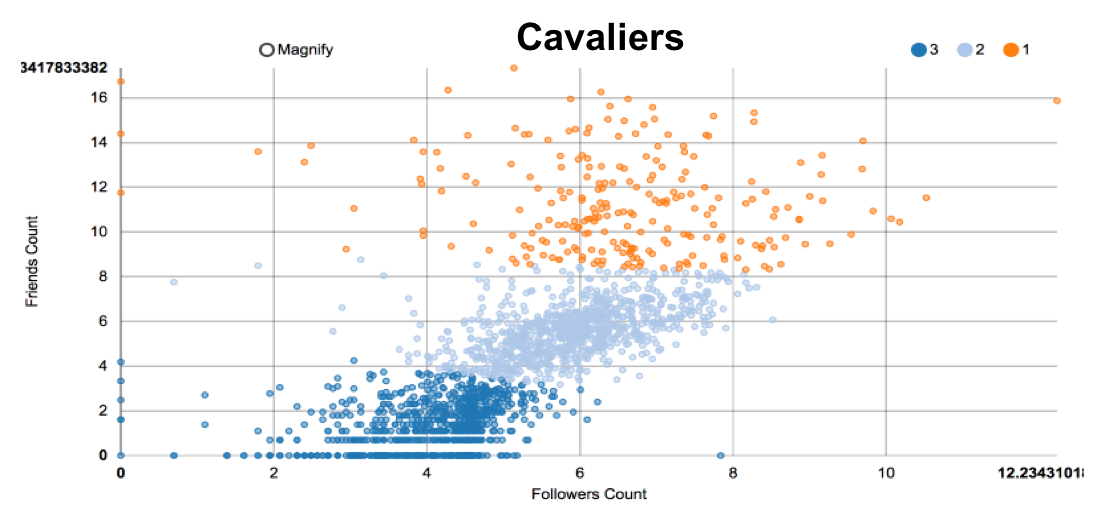
*Figure 5.* Differences in positive and negative sentiment by team (naïve bayes)

The percentages of positive and negative observations are illustrated in figure 3. We anticipated that the top teams would have a much higher percentage of positive tweets than negative tweets. However, this is not the case. First, there should be an equal number of observations to begin. This may not have affected the outcome, but we should be consistent in the number of tweets being collected. Second, the timing of when tweets are collected can greatly affect the sentiment. For example, if the 76ers were playing and doing well during the time of collection, that would explain why their percentage of positive sentiment is so high. One pitfall is that we were not consistent in the time of collection.

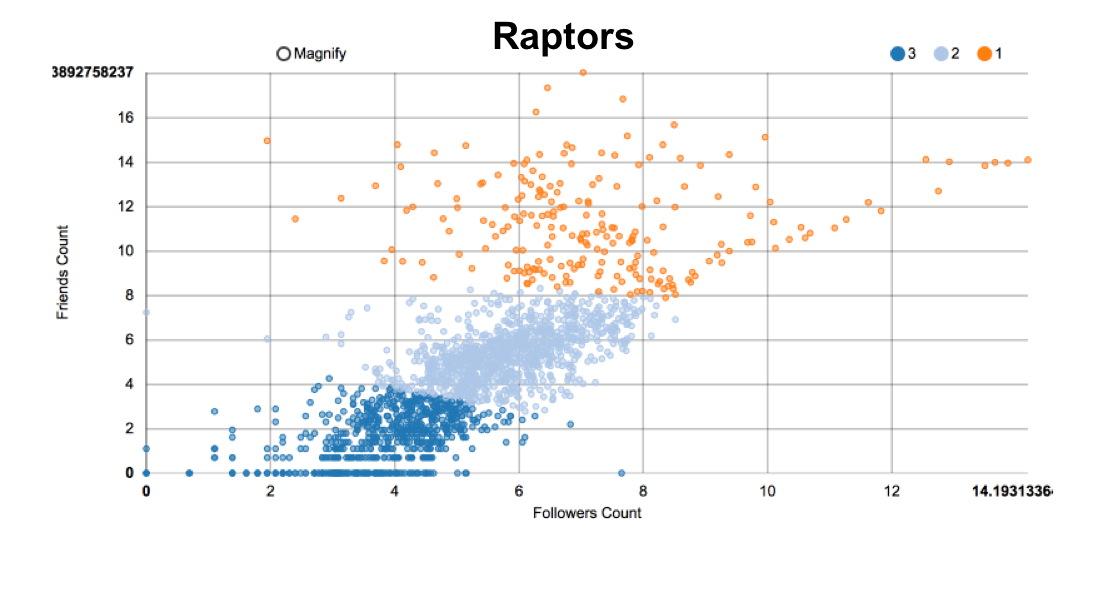
|  |  |  |
| --- | --- | --- |
| **Team** | **% Positive** | **% Negative** |
| Cleveland Cavaliers | 82.3% | 17.7% |
| Toronto Raptors | 75.21% | 24.79% |
| Philadelphia 76ers | 67.3% | 32.7% |
| Dallas Mavericks | 70.12% | 29.88% |

*Figure 6.* The percentages of positive and negative tweets per team.

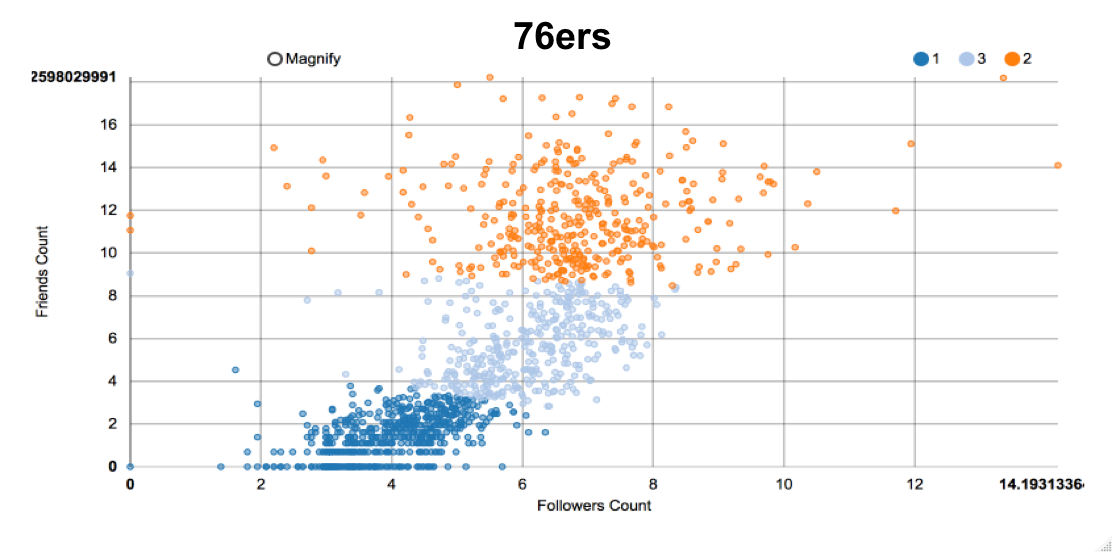
Clustering of the friend and follower counts of 500 friends and 500 followers of each team produced more meaningful data. Graphical representations of the friend count versus the follower count in top teams (Cavaliers and Raptors) are show in figures 4 and 5, respectively. Likewise, graphical representations of the friend count versus the follower count in bottom teams (76ers and Mavericks) are show in figures 6 and 7, respectively



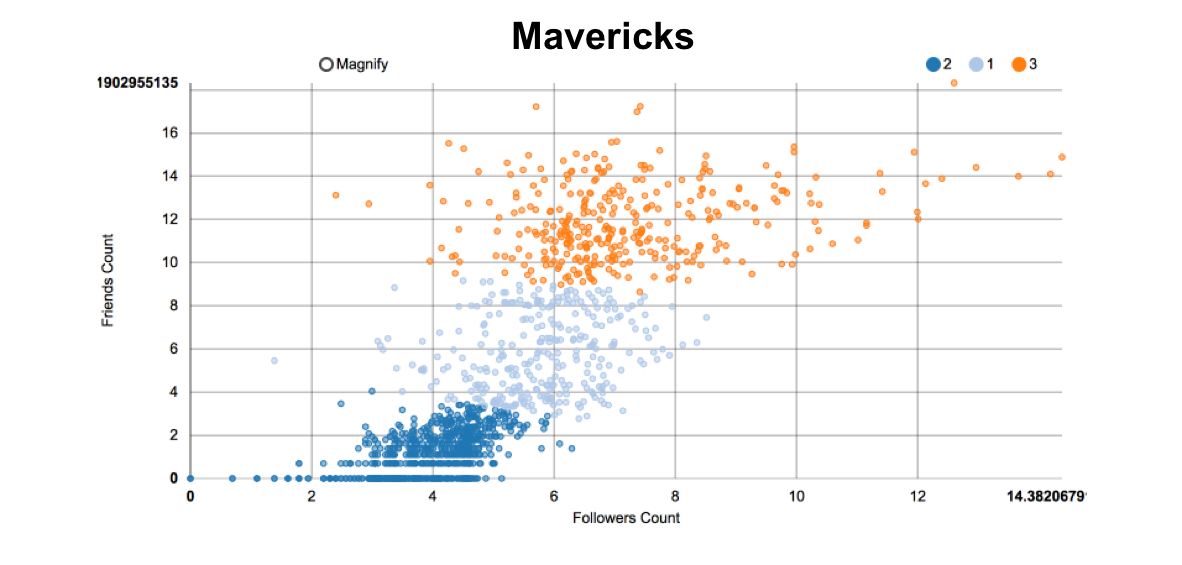
*Figure 7.* K-means cluster analysis of the friends count versus the followers count in the Cleveland Cavaliers.



*Figure 8.* K-means cluster analysis of the friends count versus the followers count in the Toronto Raptors.



*Figure 9.* K-means cluster analysis of the friends count versus the followers count in the Philadelphia 76ers.

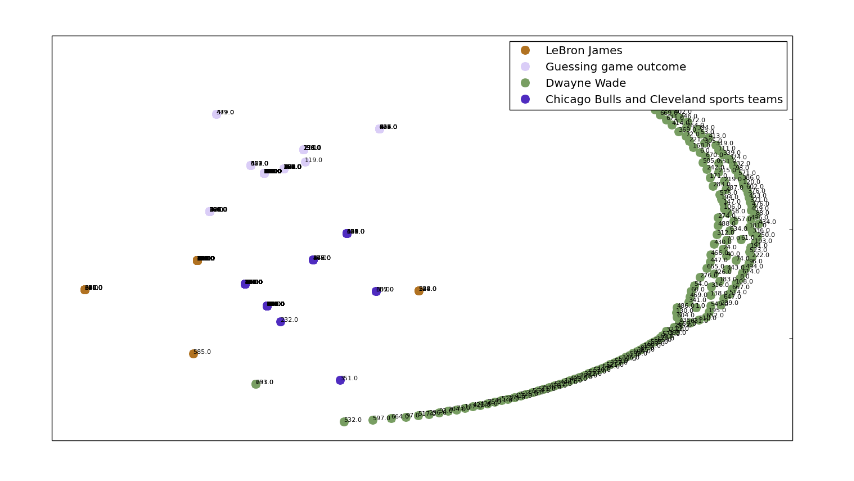


*Figure 10.* K-means cluster analysis of the friends count versus the followers count in the Dallas Mavericks.

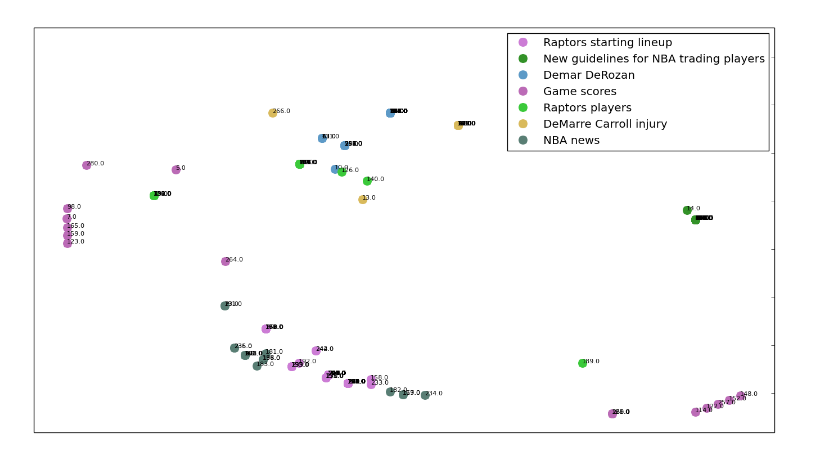
The most important observation to be discussed is the difference in the light blue, or middle cluster between the top teams and bottom teams. In the top teams, this cluster is much more concentrated with data points than in the bottom teams. This cluster represents more of the typical twitter user, with a moderate amount of friends and followers. This observation suggests that the top teams have more of these users because they are more popular at the time, and people tend to follow teams that are popular and on the rise. In other words, these users may be bandwagon fans that are only following the team due to their popularity.

On the other hand, the 76ers and the Mavericks show generally higher follower counts in the orange, or top cluster. There also appear to be more of these data points present in this cluster. This could be attributed to the fact that these type of friends and followers are more loyal, and follow the team because they have done so for a long time. These could be followers from that area, have a personal connection to the team, or have simply enjoyed that team over time.

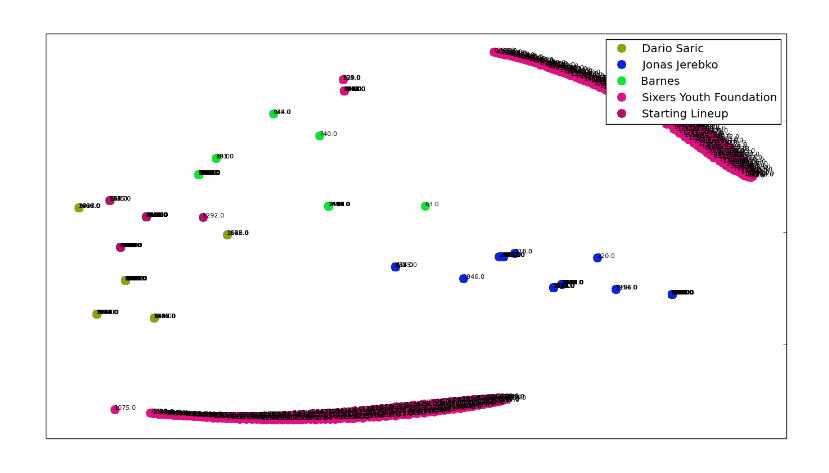
Clustering of text data also produced meaningful results. This analysis showed us that fans are very interested in individual players, and talk about players, specifically very iconic players or players that were injured. This was more frequent than teams being mentioned as a whole in our data set. Fans also like to tweet about game scores, and who is going to win for match-ups with teams ranked closely to each other. This was especially apparent for the Cavaliers and Mavericks tweet clusters. Fans were also interested and actively tweeting about events that take place in the NBA, such as new guidelines that were introduced for trading players. Lastly, it was common for tweets to contain information about team foundations and charities.



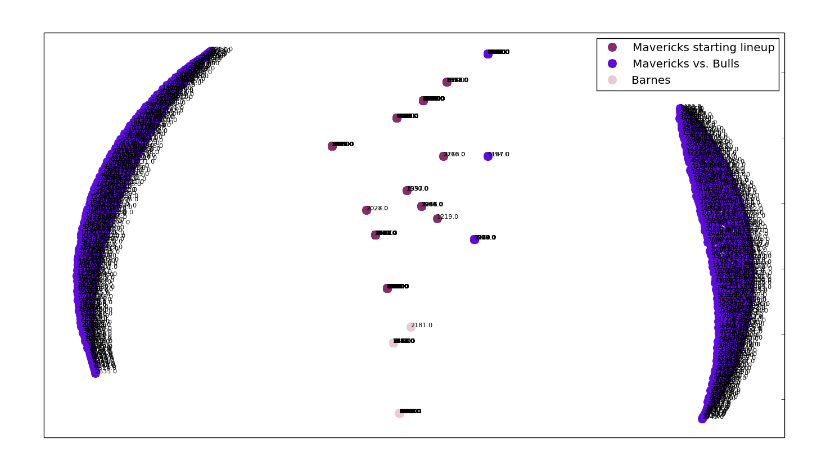
*Figure 11.* Cavaliers tweet clusters by text

**

*Figure 12.* Raptors tweet clusters by text

**

*Figure 13.* 76ers tweet clusters by text

**

*Figure 14.* Mavericks tweet clusters by text

IV. CONCLUSION

Our analysis suggests that there are differences between the types of followers for different caliper teams in the NBA. Popular NBA teams tend to have a larger volume of followers represented by typical twitter users with moderate amounts of friends and followers. Our analysis also suggests that basketball fans are interested in iconic players, injuries, game scores and other local sports teams. These overarching interests suggested by cluster centroids and popular terms are shared by several teams in the NBA. This information may be useful for fan targeting efforts, and in the future these analyses could be merged together to predict how different types of followers will differ in their interests related to the NBA. Future analysis may also determine that there are differences between the sentiments of teams that are doing well versus teams that are not at any given time in the NBA season.

**References:**

[1] “Twitter developer documentation – Twitter developers,” in *Twitter*, Twitter, 2016. [Online]. Available: https://dev.twitter.com/docs. Accessed: Dec. 15, 2016.

[2] "Welcome to python.org," Python.org, 2016. [Online]. Available: https://www.python.org. Accessed: Dec. 15, 2016.

[3] J. du Toit, "A first step into machine learning: Building a Bayes Classifier," in *Machine Learning*, Cloud Academy Blog, 2015. [Online]. Available: http://cloudacademy.com/blog/naive-bayes-classifier/. Accessed: Dec. 15, 2016.

[4] "K-Means Clustering,". [Online]. Available: https://home.deib.polimi.it/matteucc/Clustering/tutorial\_html/kmeans.html. Accessed: Dec. 15, 2016.

[5] "Home," RStudio, 2016. [Online]. Available: https://www.rstudio.com. Accessed: Dec. 16, 2016.

[6] J. Hill, "JulianHill/r-tutorials," GitHub, 2016. [Online]. Available: https://github.com/JulianHill/R-Tutorials/blob/master/r\_twitter\_cluster.r. Accessed: Dec. 16, 2016.

[7] "Twitter," in Twitter, Twitter. [Online]. Available: https://twitter.com. Accessed: Dec. 16, 2016.

[8] Y. Yu and X. Wang, "World cup 2014 in the Twitter world: A big data analysis of sentiments in U.S. Sports fans’ tweets," Computers in Human Behavior, vol. 48, pp. 392–400, Jul. 2015.

[9] A. Zimmermann, "Basketball predictions in the NCAAB and NBA: Similarities and differences," Statistical Analysis and Data Mining: The ASA Data Science Journal, vol. 9, no. 5, pp. 350–364, Jun. 2016.

[10] K. Inc, "Data - sentiment analysis on movie reviews," 2016. [Online]. Available: https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews/data. Accessed: Dec. 16, 2016.

[11] T. R. Foundation, "R: The R project for statistical computing," 2015. [Online]. Available: https://www.r-project.org. Accessed: Dec. 16, 2016.

[12] 2016, "VaderSentiment 2.5: Python package index," 1990. [Online]. Available: https://pypi.python.org/pypi/vaderSentiment. Accessed: Dec. 16, 2016.

[13] "2. Accessing text Corpora and lexical resources," 2014. [Online]. Available: http://www.nltk.org/book/ch02.html. Accessed: Dec. 16, 2016.

[14] "Tf-idf: A single-page Tutorial - information retrieval and text mining,". [Online]. Available: http://www.tfidf.com. Accessed: Dec. 16, 2016.

[15] "Cosine similarity," in *Wikipedia*, Wikimedia Foundation, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Cosine\_similarity. Accessed: Dec. 16, 2016.

[16] "Word clouds in python — wordcloud 0.1 documentation," 2013. [Online]. Available: https://amueller.github.io/word\_cloud/. Accessed: Dec. 16, 2016.