Matthew Michalke

12 November 2018

MIS3640 – Professor Zi

Assignment 2 – Text Mining and Analysis

Project Overview

The goal of this Project was to be able to make a program that would pull Tweets for a list of strings and create a table and bar graph in order to compare sentiments for each string. This could be used to compare relative attitudes towards different items. For my examples I wanted to see how people felt about different sports teams and players.

Implementation

In order to implement this program I needed to use a couple of different packages. The first is Twython which I used to pull tweets from Twitter. I then used nltk for its Sentiment Analyzer and pandas as a way to store the results from the Sentiment Analyzer. Matplotlib was used to graph the data and pprint was used to print the dataframe in a nice format as well as interpreting the json from Sentiment Analyzer when building the program.

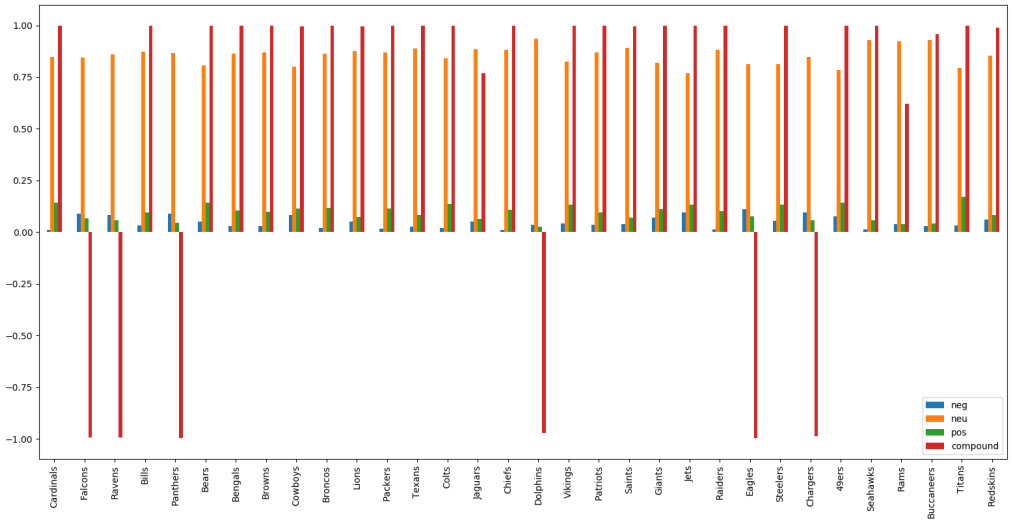
The program works by using three different functions while only having the user use one of the functions. For the user to get results all they have to do is insert a list of strings into the print_sentiment_results() function. The function has another attribute which can be used to search the strings with or without a hashtag. This function works by calling the sentiment_table() function which is used to create the dataframe of sentiment results. This function works by looping through the list, gathering the tweets using the pull_tweets() function and adding these results to the dataframe. Once the dataframe is completed, print_sentiment_results() then uses matplotlib to create a bar graph of the results which allows for easy analysis.

Results

For this program I decided to run the sentiment analyzer using lists of professional sports teams and one list of popular players. When running the analyzer for NFL teams, I expected to see that teams that had lost the week before would have a negative sentiment compared to teams that won. The dataframe results and graph output are below. The results feature four components. 'neg', ' neu', and 'pos' represent the percentage of Negative words, neutral words and positive words respectively. Compound is the score based off of the words in the tweets. A positive compound score means the tweets were more positive and a negative score means there was more negativity. It is possible for a team to have a high negative percentage of words and still have a positive compound score if the positive words out weight the negative words. As expected, there were teams that suffered losses from the week before had a negative compound score but his wasn't always the case though as some teams that lost had positive sentiment scores. I believe that this is because these teams either expected to lose or the loss wasn't as shocking. In the case of an upset, blowout or troubling news is where we see the negative sentiment scores. The Falcons were upset by the Browns in what many experts had as their "lock of the week". The Ravens lost and rumors have been circling that they would be parting ways with their head coach at the end of the year. The Eagles lost a heartbreaker in the final minutes of their

game in which they should of won as well. The Rams score also seemed to take a hit due to some bad news. Even though they won their matchup this weekend and are in first place, the team lost their star Wide Out to an ACL tear. Overall, I feel that this program is a good way to test if a team is in panic mode.

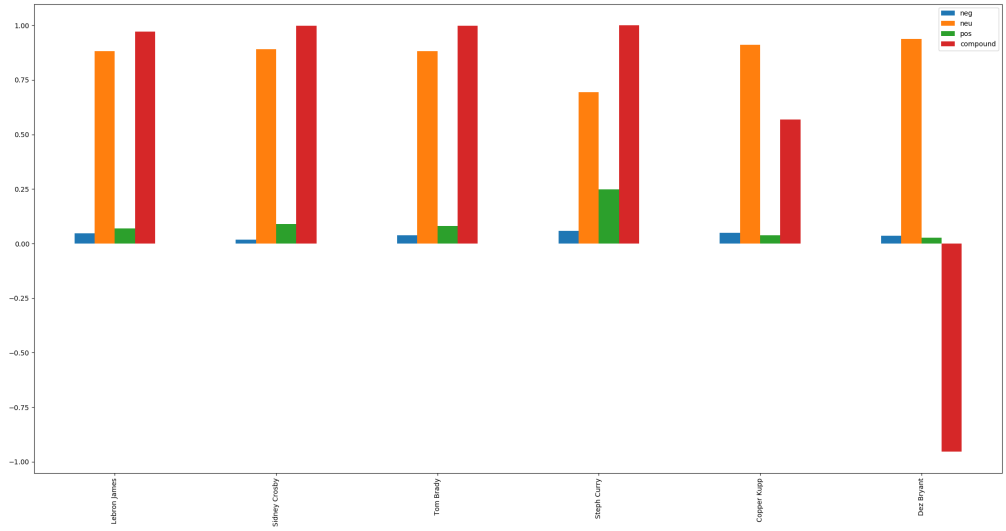
	neg	neu	pos	compound
Cardinals	0.011	0.848	0.141	0.9998
Falcons	0.09	0.843	0.067	-0.9932
Ravens	0.082	0.859	0.059	-0.995
Bills	0.031	0.872	0.096	0.9986
Panthers	0.088	0.867	0.045	-0.9973
Bears	0.052	0.805	0.143	0.9997
Bengals	0.03	0.864	0.105	0.999
Browns	0.03	0.87	0.1	0.9991
Cowboys	0.084	0.8	0.115	0.9959
Broncos	0.02	0.864	0.116	0.999
Lions	0.052	0.875	0.073	0.9945
Packers	0.018	0.868	0.114	0.9994
Texans	0.026	0.889	0.084	0.9987
Colts	0.02	0.842	0.137	0.9997
Jaguars	0.051	0.886	0.063	0.7684
Chiefs	0.011	0.883	0.107	0.9995
Dolphins	0.037	0.937	0.026	-0.9727
Vikings	0.042	0.825	0.133	0.9996
Patriots	0.036	0.871	0.094	0.998
Saints	0.04	0.89	0.07	0.9938
Giants	0.07	0.82	0.11	0.9977
Jets	0.096	0.77	0.134	0.9978
Raiders	0.015	0.881	0.103	0.9992
Eagles	0.111	0.812	0.076	-0.9981
Steelers	0.054	0.814	0.132	0.9992
Chargers	0.095	0.846	0.059	-0.9856
49ers	0.075	0.783	0.142	0.9993
Seahawks	0.015	0.928	0.057	0.998
Rams	0.039	0.923	0.038	0.6218
Buccaneers	0.03	0.929	0.041	0.9582
Titans	0.034	0.794	0.171	0.9998
Redskins	0.062	0.854	0.084	0.9881



The next group of data I ran was for individual players. For this list, I removed the hashtag from the search since I wanted to search for the player's full name and not just a hashtag associated with them. For the most part, the list of players I included features players are leaders in their respective league. Lebron, Crosby, Brady and Curry all had positive scores. All are playing well for the most part of the season. The one place we do see low or negative compound scores are when players get hurt. Copper Kupp was the Wide Out for the Rams who tore his ACL. He has had a productive season so far which may explain why he still had a positive score. Dez Bryant on the other hand tore his ACL in his

second practice after holding out in free agency all year. His lack of performace this season may be able to explain the negative score.

	neg	neu	pos	compound
Lebron James	0.048	0.882	0.07	0.972
Sidney Crosby	0.019	0.891	0.09	0.999
Tom Brady	0.038	0.881	0.081	0.998
Steph Curry	0.058	0.693	0.249	0.9999
Copper Kupp	0.05	0.912	0.039	0.5687
Dez Bryant	0.037	0.937	0.026	-0.9536



Reflection

Overall, I think the program works pretty well. It's fairly quick and I feel that it is accurate. One thing I think I could improve on is somehow limiting or removing retweets. This was something I thought about pursuing but felt that ultimately when someone retweets they are more or less agreeing with that person's sentiment. It might have been a good idea to try this and compare results. I think what I learned going forward is going to help me. Being able to use multiple programs in different ways but combining them in the end is useful when tackling a problem like text analysis.