11/30/20

Which tasks have been completed?

Pretty much all the source code has all been finished. The remaining coding work will be if I want to make any additional improvements. I have also implemented all the code into a Jupyter Notebook which I will use for a demo. Most of the tutorial and description in that has been finished. I have broken the code down into two parts: basic sentiment analysis and implementation of BM25Okapi to give some more context of the contents of score.

The first part of the code uses Tweepy to source tweets and then Textblob's Sentiment Analysis to classify. Every starting player from each team will receive an average sentiment score based on the classification of these tweets. There is also code that outputs the results visually. The second part of the code uses BM25Okapi ranking to find specific tweets that might have contributed to the sentiment analysis. These findings are also visualized.

Which tasks are pending?

The two main tasks that are pending are the demo run in video and documentation. If I have some more time, perhaps more can be done to improve the overall implementation / clean up of the code for efficiency in the future. I am currently working through the documentation on my GitHub page which is part of the README. The documentation will provide further detail as to what exactly each part of the code does as well as my thought process.

I have also chosen to use a Jupyter Notebook to provide a step by step on how to run through my code. I have written most of the instructions and tutorial in that but will likely refine it some more.

Given the limitations of the Twitter API access (more below), I will likely wait to run the demo along with the video closer to the due date. I will likely provide one notebook with the results of my run through so others can follow along with the results and then a notebook with a blank implementation for free use.

Are you facing any challenges?

There are a few challenges that either fall into the category of text retrieval regarding the API or further improvement of the algorithm.

The Twitter API access is an issue as the free version only retrieves a limited number of tweets over only the past 7 days. This makes it a bit difficult to set a demo of the code, but I will likely just have to set a game that I want to use for the demo run in the coming week to not cut it too close to the deadline. The other challenge is that Tweepy is slow in retrieving many tweets, especially given the Twitter API limitations of how many times I can call it. For speed purposes, I will likely have to set the number of tweets per player at 100 for my demo.

As for the algorithm, there are some areas that could be improved beyond the basic implementation. Teams and players can go by a handful of names so it would be useful to be able to count all occasions. However, for the most part, there is usually a more popular name that each team or player is referred to. As for most NLP, more work can be done in finding deeper meaning in words especially when the tweets are more contextually complex. If multiple players are mentioned in a tweet or previous references are made, it can be difficult to classify if a tweet is positive or negative for a particular player.