Text_Mining:

Comparing Political Hashtags

Brennan VandenHoek

Project Overview:

I used Pattern to pull information from twitter and then compare the bulk associations of the tweets using Pattern's built in sentiment analysis. With the increased importance of twitter and the almost hostile nature of the current political environment I was interested in looking to get a greater analytical understanding of how different hashtags are being used.

Implementation:

My code is broken down into three core functions:

The lowest level function takes in a hashtag and outputs a list containing a number of tweets that are using that hashtag. There is also an input that allow one to control the number of tweets that are in the list. This function utilizes the pattern.web search tool as a wrapper for the Twitter API so that I didn't have to deal with the OAuth stuff.

The next function takes in a hashtag and output an average sentiment for a set of tweets associated with that hashtag. It does this by calling the previous function for a list of tweets and then traverses the list of tweets calculating their individual sentiments and calculating an average. It also filters out sentiments that come back null like in the case where a tweet is just a link and a hashtag.

The final function is takes in two different hashtags and using the previous functions calculates the average sentiment for each hashtag. It then using a set of if statements compares the two sentiments and prints out a string that describes the comparison.

Results:

The results are my program are pretty straightforward to look at as the output is simple a string. I wanted my output to be very simple and text bot esque. Given my intention of looking into politics the first thing I compared was general Trump and Hillary hashtag. When comparing *#trump2016* and *#hillary2016* the program outputs: *"Tweets that use #hillary2016 are currently more positive in sentiment then tweets that use #trump2016 and tweets that use #trump2016 are currently more subjective in sentiment then tweets that use #hillary2016."*

Another classic example is to compare #win vs #fail. When doing so the result is: *"Tweets that use #win are currently more positive in sentiment then tweets that use #fail and tweets that use #win are currently more subjective in sentiment then tweets that use #fail."* While the output of my program is pretty basic, it is only scraping the surface of what is capable.

Reflection:

I was pretty happy with how this project went. In general I was able to do exactly what I set out to do. If anything it left me wanting to expand on and spend more time creating a more advanced program. The biggest problem I face was limits set by pattern for number of queries. I spent sometime digging into the twitter API and other twitter python wrappers like tweepy and twython. These would have allowed me to get past some of the calling limits imposed by pattern and would have let me be able to get a broader set of data. This was ultimately deemed outside the scope of this project. I found it generally pretty shocking how available and easy it is to scrape data from the web. There is so much possibility for what you could quantitatively be able to do it is incredible.