

Mary Keenan
22 February 2016

Twitter Polarity

Project Overview:

I searched Twitter using the names of the different presidential candidates and collected a number of recent tweets about each candidate. I analyzed the polarity of each tweet using the Pattern package and found the average polarity of the tweets about each candidate. Then, I graphed the results in a bar chart, candidates vs. Twitter's polarity towards them. I thought it would be interesting to learn which candidates were viewed positively on Twitter at any given time. The bar chart is a simple/clear way to show the results.

Implementation:

I use Pattern.web to search Twitter with a search term and collect a certain number of tweets. I find the average polarity of the tweets using Pattern.en and use a bar graph to show Twitter's polarity at that moment in time towards each of the candidates.

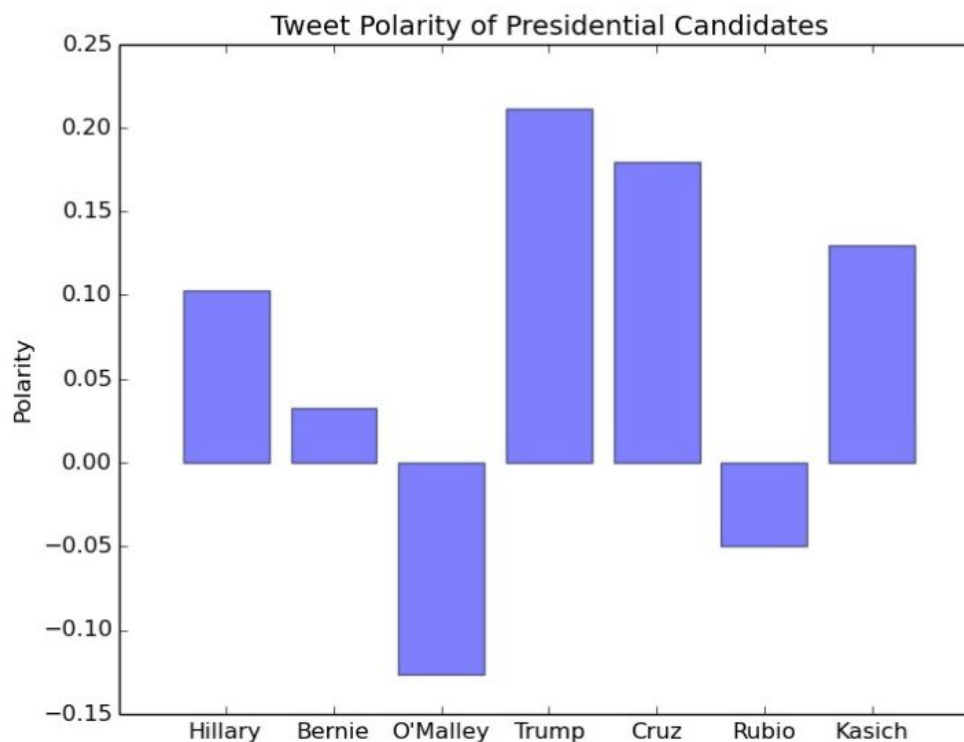
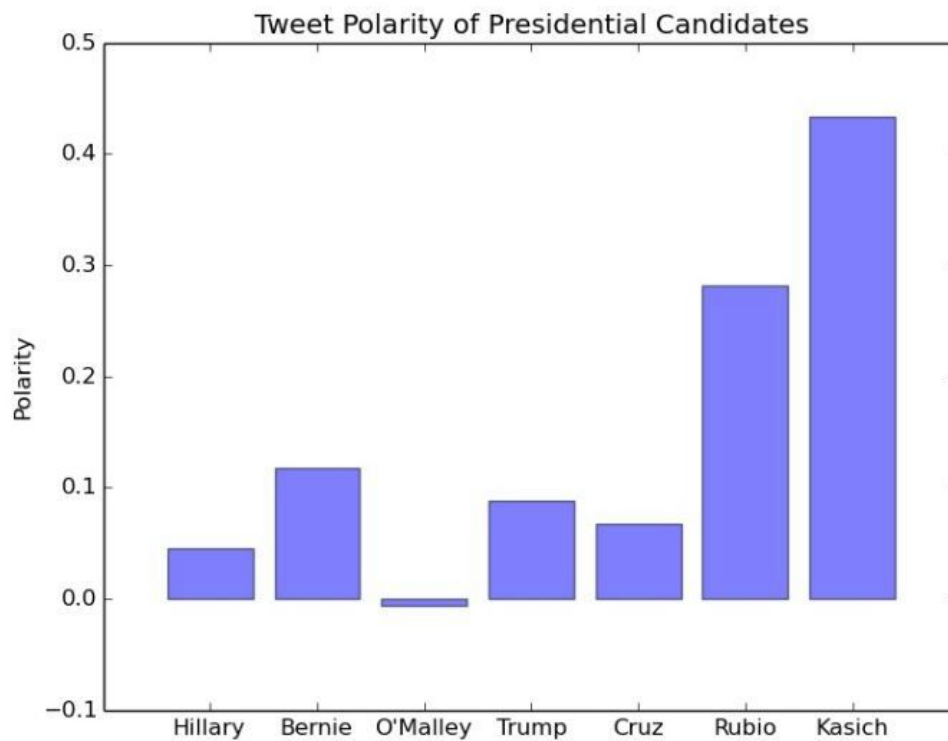
The biggest design decision I had to make was how to change my code so that it could take more than two search terms and return more than two average polarities -- which is what I had before I decided to look at tweet polarities of presidential candidates. I was running two separate for-loops to compare the two search terms and the tweet-collecting and tweet-polarity functions were separate, adding another layer of complexity. I decided that rather than create a separate for-loop for each candidate (lazy, but easy) or run the entire function seven times for the seven candidates in a massive and redundant for-loop, I would integrate my tweet-collecting and tweet-polarity functions into one function that could take any number of search terms and return the same number of polarities. This probably sounds like a really obvious solution, but I wasn't really sure how to do that, and I was worried I'd spend a lot of time trying and get nowhere. I ended up choosing to do that because I know shorter is often better when it comes to coding. You shouldn't do something in 100 lines of code if it's possible to do it in 20.

Results:

I've run my project a number of times now over a couple of hours, and my main finding is this: not only is the polarity of the tweets about the candidates highly variable, it's also generally positive. From what I understand, more than half of the country dislikes each candidate, so it would make more sense for the candidates to mostly have negative polarities. However, I'm going to hazard the guess that although it takes a relatively small amount of effort to write a tweet, it's more effort than most people are willing to expend. For the most part, only the people who feel strongly about the candidates tweet about them, and the people who feel most strongly about them are, based on my project, their fans.

Here are two graphs. The one on the top was produced at 9 pm Monday night and the one on the bottom was created at 5 pm Tuesday. Although the polarity of the tweets about each candidate have changed, you'll notice they are almost entirely positive. Also, I noticed that the biggest change from Monday night to Tuesday afternoon was for Marco Rubio -- I looked at

recent news and the only thing I saw that might have caused people to post more negative tweets after Monday night and before Tuesday at 5 pm is the fact that the Koch brothers are now endorsing him. So, I would say that usually the people who tweet about candidates are their fans, but when the candidates have major news, others are prompted to voice their more negative opinions. That's pretty obvious, I know. I think it speaks more to the efficacy of my project at measuring sentiment towards candidates on Twitter and less to my analytical skills.



Reflection:

I started with code that compared the polarity of two different search terms, and when I decided to expand to comparing the polarity of multiple search terms (about seven -- I wanted to look at presidential candidates), I had a bit of a challenge, as I mentioned earlier. I think I condensed and adapted my code well. It's very short, but it gets the job done better than it did before.

I would love to be able to look at the polarity of tweets about the candidates over an extended period of time (a couple of months) and match the result up to significant moments in their campaigns, but I'd have to download more than a hundred thousand tweets and I think that's a little much for this project. Still, that's an interesting extension.

Also, Pattern is incredibly useful, but it's not the best sentiment-analysis tool out there, so I could probably have improved my accuracy if I had used Indico, but I had trouble installing that because of my virtual machine (I think my machine is tied to Olin somehow and can only install certain things in some Olin database? I don't remember what the problem was exactly, but I need to figure out how to change that).

P.S. I'm really happy that I learned how to use matplotlib. It's really useful and I didn't even know it existed prior to this project:)