Apurva Raman
Text Mining
Software Design Spring 2016

Project Overview

For this project, I chose to use Google as my source and wanted to do sentiment analysis. My data set was the first 100 search results for 'Black Lives Matter'. I wanted to know how subjective the top results were and what the overall polarity of the sentiment was. I also wanted to be able to represent the data graphically and see if there were any trends in sentiment related to the order of the results (for example, are later search results more subjective?).

Implementation

To implement this process, I first used `pattern` to search Google for the top 100 results for Black Lives Matter. Once the search results (the urls and strings) were compiled into two separate lists, I used `pickle` to write each list to a file. The pickled file is then loaded and processed using pattern's sentiment analysis functionality, and the polarity and subjectivity are plotted. The average polarity and sentiment are also found.

I did a lot of refactoring for the code; for example, the code encapsulated in the `plot_list` function used to be in both `plot_polarity` and `plot_subjectivity`. Similarly, `get_sentiment` and `get_polarity_subjectivity` were the same function. I wanted to split the code into more readable segments, so I chose to reorganize it.

Results

I didn't really go into this project with a specific hypothesis in mind: I was just looking to see what results I could find. From the top 100 search results for “Black Lives Matter”, the average polarity of the results was -0.022376901455026445 and the average subjectivity was 0.4281954034391531.

The article with the lowest polarity was this little gem that is quite clearly racist:

http://townhall.com/tipsheet/katiepavlich/2015/09/02/exposing-black-lives-matter-for-what-it-is-promotion-of-cop-jilling-n2046941
(URL 29)

The article with the highest polarity is a very positive feel-good sort of article about Beyonce's connection with the movement:
http://www.rollingstone.com/culture/news/black-lives-matter-co-founder-to-beyonce-welcome-

to-the-movement-20160211

Given that the ends of the spectrum make sense, sentiment analysis with polarity may be a good way to filter this kind of data about a polarizing political movement.

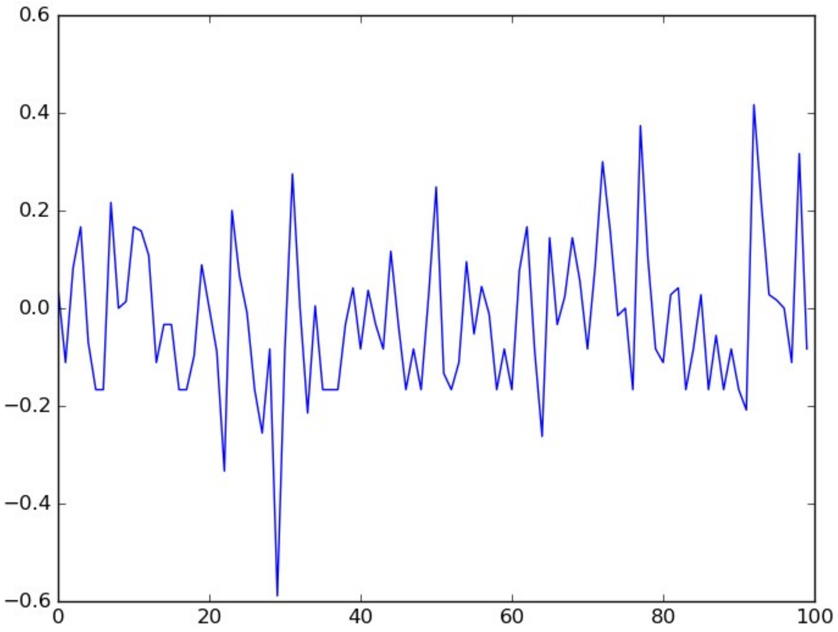

Fig. 1: Polarity of each search result

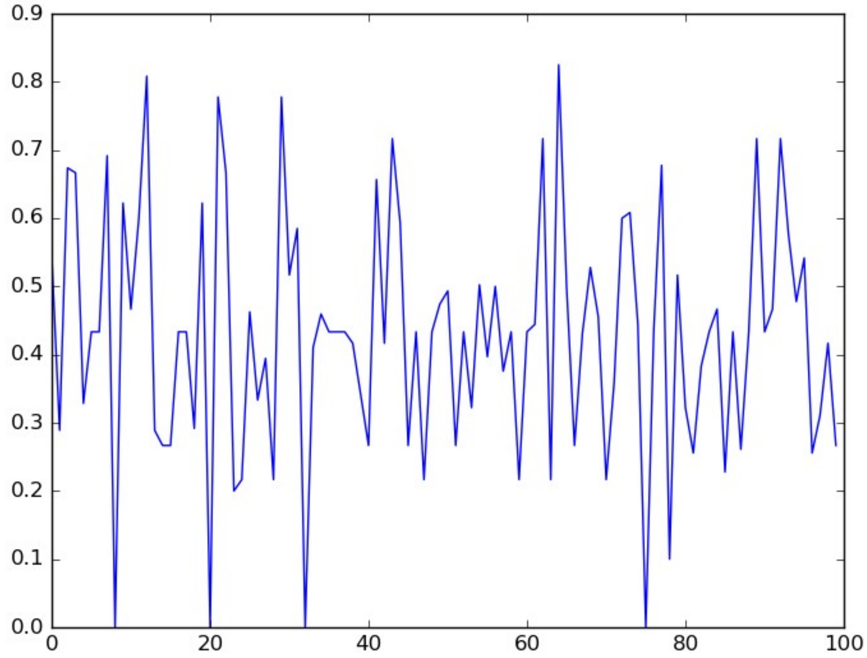

Fig. 2: Subjectivity of each result

The page with the highest subjectivity was this news article:

http://www.oregonlive.com/politics/index.ssf/2015/11/black_lives_matter_oregon_just.html
(URL 64)

This was flagged as highly subjective because of the use of an opinionated quote.

The pages ranked as having no subjectivity were:

http://www.wired.com/2015/10/how-black-lives-matter-uses-social-media-to-fight-the-power/
https://stories.californiasunday.com/2015-03-01/black-lives-matter/
http://www.latimes.com/local/lanow/la-me-ln-garcetti-forum-black-lives-matter-20151019-story.html
http://www.left-bank.com/black-lives-matter
(URL 8, 20, 32, 75)

These are all also results from publications with the exception of the last (which is a booklist), but they all have a neutral metadescription, even if their contents are less neutral.

From a preliminary analysis, it appears that polarity is more easy to get an accurate gauge of from a metadescription from a Google search than for subjectivity.

There was no apparent relationship between subjectivity and how far down the search result was, but as the subjectivity metric does not seem to be very accurate, more testing is needed.

Reflection

There are definitely things I could improve, but most of the limitations for me for this miniproject were largely due to my health in the time frame. I learned that my ability to write code readably is severely impaired when I am unwell. I did not scope the project appropriately to begin with as I wanted to implement too many ways to process data. I quickly re-evaluated my strategy, which was helpful. I did unit testing by using artificially constructed data to see if the results were as I expected. I also printed my results at several points in the execution flow to see if things were working. I would like to find a way to implement doctests (perhaps with an artificial small data set defined in the script) and would like to spend more time figuring out what kind of data I wanted to get. Formulating a hypothesis and testing it would be a good way of approaching this, and I wish I had known that to start with. In addition, I learned a lot about MDS and many of the things pattern does just by playing around, which I will definitely take with me going forward.