Tweet for Taft: An international look at international affairs.
Software Design HW5 Write-up
Neal

Overview

The U.S. can at times be quite insular when it comes to world affairs, with the tendency of our media to focus on celebrity gossip, freak accidents, trials and crime. At times our knowledge of how our foreign policy is perceived by the rest of the world can be completely backwards, if for no other reason than we are too close to the decisions being made to be able to take an objective stance.

In this project I sought to embiggen and enhance my knowledge of global political perceptions by amalgamating sets of data from Twitter users in various countries to see what trends might emerge. Looking at a select few leading Western powers and analyzing the sentiments of tweets from people in their capital cities, I unearthed some curious insights into how affairs are perceived by different nations.

Implementation

My program was implemented using Tom de Smedt's Pattern package, the Pickle package, and Matplotlib. Creating a few basic functions around its Pattern.web Twitter-crawling and geo-locating functionality, and its Pattern.en sentiment analysis I was able to scrap and analyze text with relative ease.

My first step in approaching this problem was to make a function that would search Twitter for the information I sought, taking form as

```
search_with_language_in_region(lang, capital_city, search_terms, file_name).
```

This function takes in `lang` and `capital_city`, `search_terms` and `file_name`. lang is passed to the `Twitter()` constructor, in this case telling it to only search for tweets in English; `capital_city` is an argument to Pattern.web's `geocode()` function, telling it to search in the vicinity of the given capital city, and is likewise used by the constructed twitter object when it performs searches. `search_terms` is self-explanatory, it is a term or list of terms you want to search for, and `file_name` is what you want the file saved as.

When the function is run it gathers the tweets and saves them in a dictionary whose keys are the unique id of each tweet. When the search is finished the dictionary is pickled and saved to disk with the given filename.

The next function does semantic analysis of the gathered data using Pattern.en, and is called

```
semantic_analysis(pickled_tweet_data_filename)
```

It takes in the pickled dictionary of tweet data, unpacks it, and runs sentiment analysis on it via Pattern.en’s `sentiment()` function, thresholding the data appropriately and returning a list of the positive tweets it counted and the negative tweets it counted.

Next are two functions that make one’s life much easier: `automate_data_collection()` and `automate_sentiment_analysis():`

The first of these two just runs through a list of cities, and for each city runs through a list of terms, and calls the search function appropriately:

```
cities = ['Paris', 'Brussels', 'Berlin', 'Kiev', 'Moscow']
search_terms = ['obama', 'putin', ['obama', 'crimea'], ['putin',
'crimea']]
for city in cities:
      for term in search_terms:
            search_with_language_in_region('en', city, term, city + '_'
      + term)
```

thus automating the collection and storage of data.

The second of the two automation functions does a similar thing, running through a list of cities and terms and calling `semantic_analysis()` appropriately. It then returns the count of positive and negative tweets, sorted into a dictionary by city and term:

```
cities = ['Paris', 'Brussels', 'Berlin', 'Kiev', 'Moscow']
search_terms = ['obama', 'putin', ['obama', 'crimea'], ['putin',
'crimea']]
city_tweet_data = {}
for city in cities:
      for term in search_terms:
            city_tweet_data[city + '_' + term] = semantic_analysis(city
      + '_' + term)
return city_tweet_data
```

The remaining functions generate the plots for this assignment. Their implementation is not described here because it is unambiguous, based on provided examples, and is self-explanatory.

Results

The results for this project are interesting. While at times they are somewhat ambiguous, they do exhibit interesting salients that are neat to be seen proven in practice. They are elaborated below.

To begin, I figured a cool current event to get an international perspective on would be the situation in Crimea. To see what the international community thought I searched for a combination of the terms “Obama,” “Putin,” “Crimea,” “Obama Crimea,” and “Putin Crimea.” I looked at the capital cities Berlin, Brussels, Paris, Moscow and Kiev, due to their wide geographic spacing, importance as NATO members, and their importance as key players in the Crimean situation.

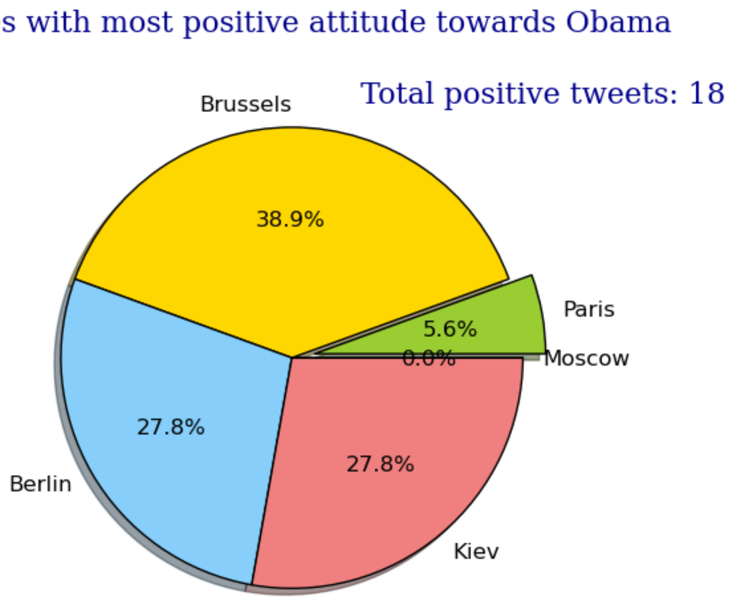

fig. 1: Percentage of positive tweets for search term “Obama” sorted by city.

Figure 1 shows which cities love Obama the best, based on the ratio of positive tweets regarding Obama to all tweets regarding Obama. The chart shows a pretty even split, as expected because Germany, Brussels and the Ukrainian capital of Kiev are all anti-Russia and hence pro-Obama (probama). Paris is the outlier in this regard, but this makes sense: given America’s shaky relationship with France during our stint in the Middle East, it is no surprise that that Parisians would have much good things to say about possible U.S. involvement in Europe, which brings us to Figure 2, which is the same idea but with Putin instead.

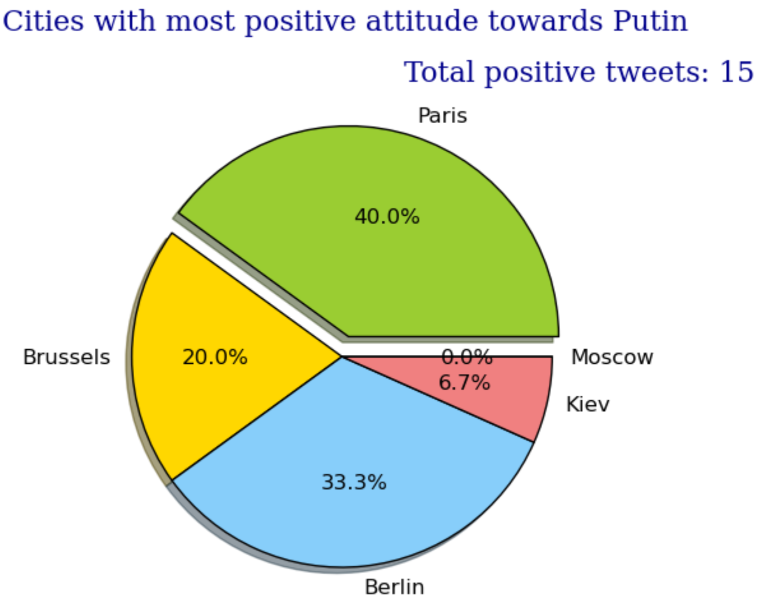

fig. 2: Percentage of positive tweets for search term “Putin” sorted by city.

Figure 2 is an interesting corollary to fig. 1; it shows data for a search term that is not competing with the term Obama (and hence won’t take away tweets), but still seems to accurately display an inverse relationship with it. Specifically I refer to the Paris section of the pie chart: whereas in the first figure Paris represented the smallest percentage of positive tweets towards Obama, in this second figure it represents the largest percentage of positive tweets towards Putin, Obama’s rival in the Crimean situation. These two figures combined hence appear to show a distinct French distaste for U.S. foreign policy at all costs--even if that entails supporting Russia. It is worth noting that Kiev has a quite small percentage of love for Putin, which is a good gut check that the sentiment analysis is at least in the right ballpark.

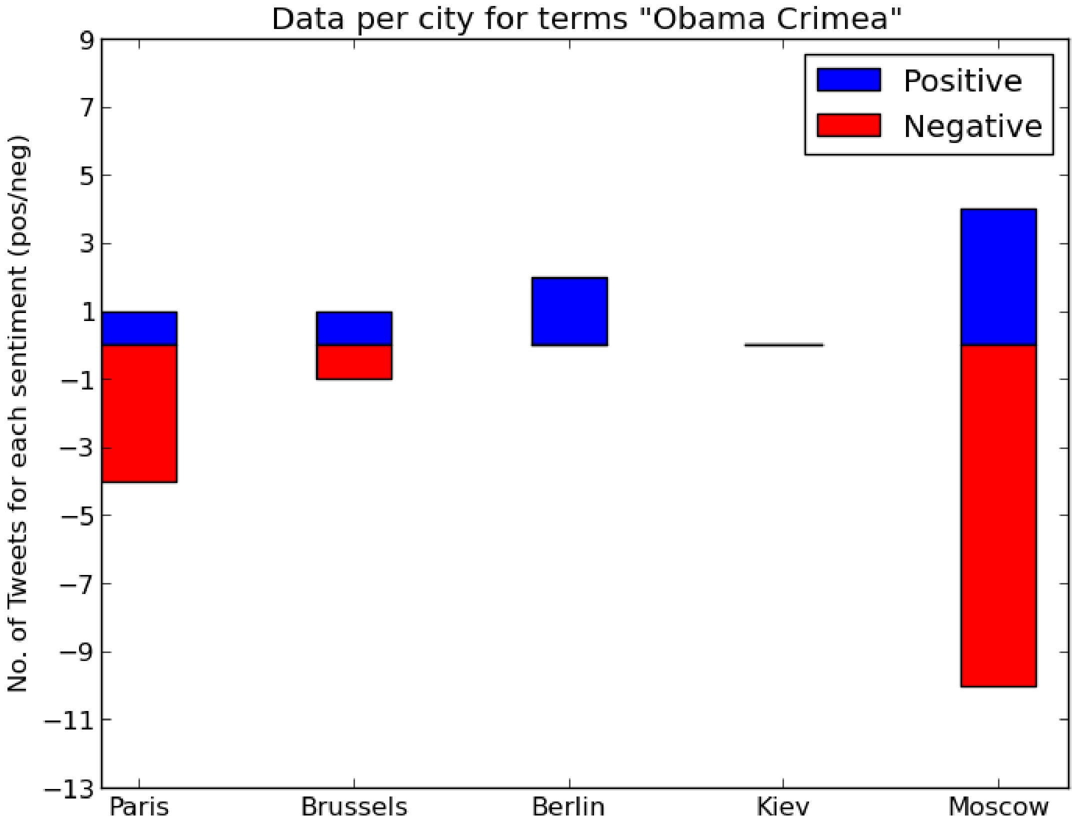

fig. 3: Bar plots showing the data for each city for the more specific search of “Obama Crimea”

In figure 3 we have a plot showing the positive and negative tweets for each city. The Kiev data is skewed to be zero because it was flooded with retweet spam that didn’t pass the thresholding boundary of the sentiment analyzer, so it was counted as neither positive nor negative. The Paris data reflects the above-noted anti U.S. sentiment of the French, and the Moscow data shows strong anti-U.S. leanings, and these are to be expected; it is the few positive tweets for Moscow which are interesting: while our inherent dispositions as citizens of this great nation make the negative responses seem obvious, it is the goal of this project to see through these prejudices and see if there is broader sentiment than we might believe. Thus, the positive tweets for Moscow might convey a greater tension within Russia than we might otherwise be able to perceive.

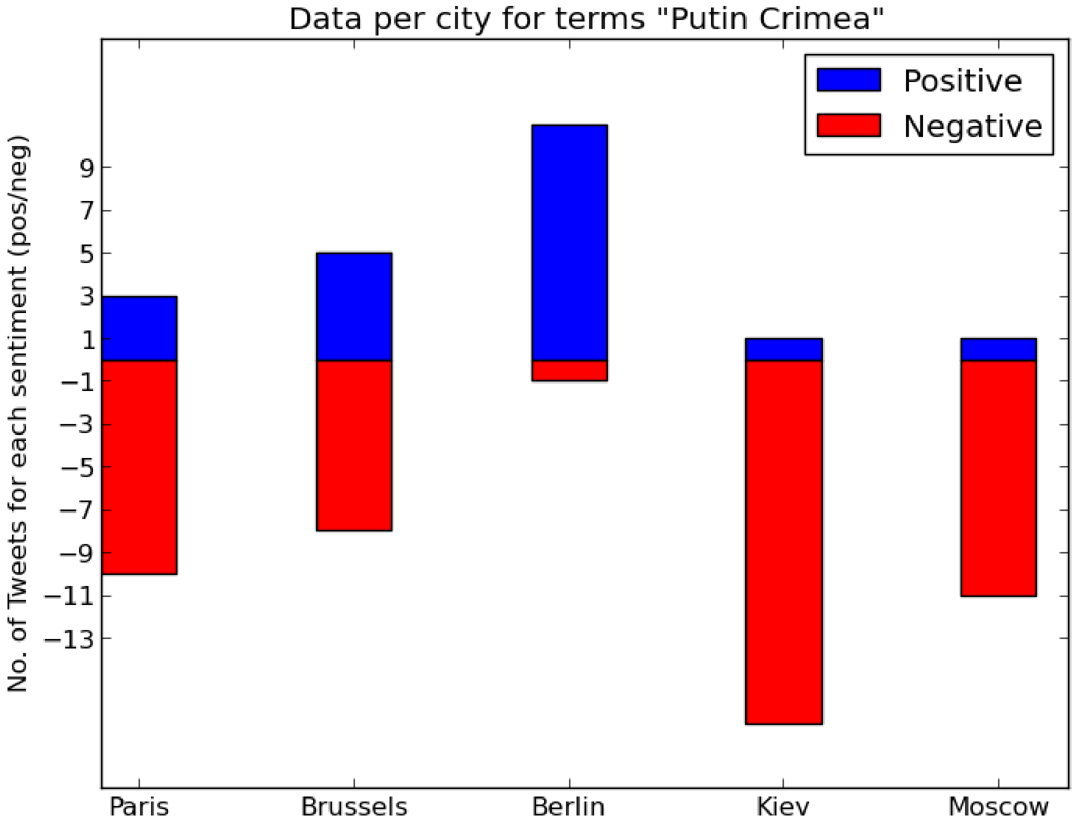

fig. 4: Bar plots showing the data for each city for the more specific search of “Putin Crimea.”

While it may not have the most startling revelations, figure 4 is perhaps the most telling plot: it seems to convey that the U.S. is not alone in its dislike of Putin’s grabbing the Crimea. Nearly all capital cities--except, strangely, Berlin--disapprove quite strongly of the invasion. This gives us insight that we are not just trying to dictate European affairs without their mandate: they appear equally upset at the situation as we are.

In conclusion, while this program may not be the most accurate, it provides an interesting look at world sentiment of international affairs. It would be quite cool to run the program without the limits imposed by Twitter’s free text-mining license.

Reflection

This was quite a fun project. I think it went well, but in retrospect I can see a lot of improvements that could have been made.

Pattern made writing the program a joy. It is quite well documented and incredibly extensible; at first I thought I would have to write a parser to have it only analyze tweets in English, but this turned out not to be the case much to my happiness.

I realized that using objects to store my data, rather than fumbling around with dictionaries, would have made things a million times easier just as I finished the project . This is especially true when trying to plot things, as each function to generate a plot was written in a very naive, hard-coded manner in order to access the specific data I wanted to show. Otherwise, I think the project was adequately scoped since I was working by myself; it was not a trivial project but I was able to get it done in a reasonable amount of time. Building the program bit by bit (i.e. starting with a search function, then an analysis function, then automating them) made unit testing very straightforward.