Jeffrey Lee – U166445
March 14, 2019
Final Project Report

Introduction

For this project, we focused on collecting publicly available tweets from Twitter, analyzing them, and visualizing the results. To do this, we used a series of Java applications with Twitter4J to connect to Twitter's API and build the maps. We also used GATE and the TwitIE plugin for GATE to analyze and extract information from the text. We also used a gazetteer to judge the sentiment of each tweet. For my project, I was given the area of New York City to focus on.

Data Collection

To collect data for this project, I used two different methods from the Twitter4J package. The first was a simple query that acquired recent tweets within a radius from a certain point. For my purposes, I fetched up to 100 tweets at a time within a 3-mile radius of Midtown, Manhattan (40.7549 ,-73.9840). This method was effective if I wanted to fetch tweets at a certain time of day, and it can be seen in the Java class SearchAroundLocation.

The next method I used was the bounding box program we saw in class. I chose the boundaries of (-74,40) and (-73,41) to enclose New York City based on the website we saw in class. This program was a tweet listener and can run continuously until stopped, saving every tweet from within the boundary that occurred since it began running. This method was useful if I wanted to extract data over a longer period of time and not have to manually collect it every 5-10 minutes. The functionality can be seen in the classes SearchByLocation and MyStatusListener.

One issue I found with data collection was the lack of geolocation on many tweets. After some research, I found that if a public tweet was not geotagged, it would set the location of the tweet to the user's profile location and be included in the data collection. However, this data is not useful to us, as we cannot map it without a geolocation. This led to a need for more data, since a large portion was not useful for my project. In the end, I collected 2,340 tweets for analysis.

Information Extraction

Data extraction in this project was done entirely in GATE and preprocessed before running any Java application. We used the TwitIE GATE plugin to extract information and annotate each individual tweet. This package is specialized for information extraction of social media posts, and accounts for abbreviated words, emojis, and more.

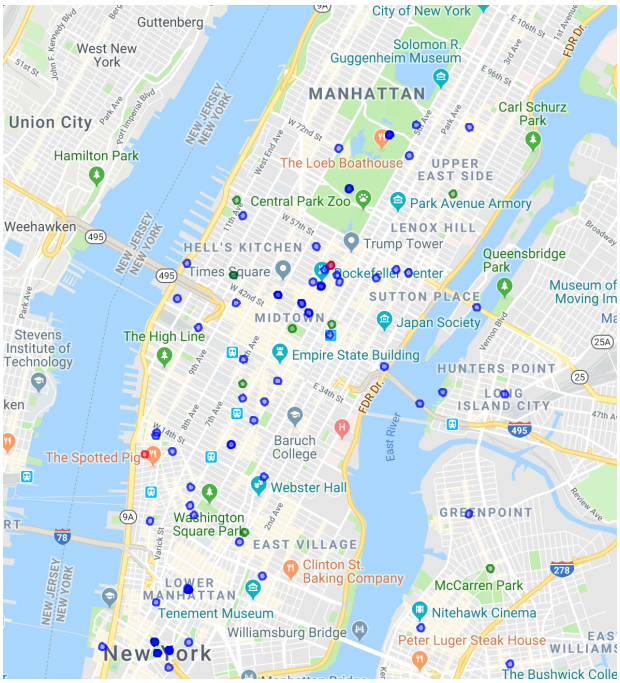
After running TwitIE on my corpus, I needed to use an ANNIE Gazetteer to judge the sentiment of key words in the tweets. To do this, I wrote the Java utility PrepGazetteer to process the XML dictionaries that were provided to us in the lab. The most complicated part of this process was only including positive words with polarity greater than 0.5 and negative words with polarity less than -0.5. However, after testing different methods of document extraction, I was able to compile four positive and negative word lists to supply our gazetteer with.

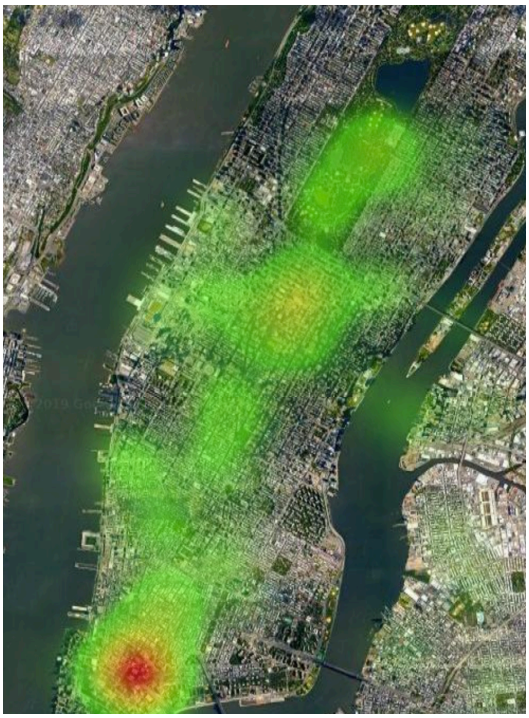
Sentiment Computation

Computing the sentiment of a tweet was fairly easy after the gazetteer annotated each file. For each tweet, I would keep a count of positive and negative sentiment words. I also set a filter that would only count a positive word if it was in the same language as the tweet, and the same for negative. This helps us avoid miscounting words in different languages. The tweet language was taken from the original markups, while the sentiment language came from our gazetteer. At the end of the text, the final sentiment would be marked positive, negative, or neutral depending on the count of each annotation.

Map Creation

Creating the map was a straightforward task given the outline provided by the instructor. I discovered that the supplied Java program added JSON objects to the map template, so to include extra fields such as the number of organizations, people, locations, or links used, all I had to do was add to the Java object being written to the file. Then, in the HTML template, I edited the callback function to display a box with more fields to meet the requirements. I decided on using green to denote positive sentiments, blue to denote neutral sentiments, and red to denote negative sentiments. These colors are all easily distinguishable and already hold similar meaning in many user's minds. For the heat map, not much adjustment was needed to replicate the functionality of the example. Both maps needed to be centered on Midtown, Manhattan, to properly show the data.





Conclusion

This project was an interesting example of data collection, information extraction and data analysis. In the end, we produced a webpage that displays our data. This could be connected to an automated system that live updates the past day or two of tweets in a certain area to show the sentiment of users in a given geographical area.

References

- https://docs.oracle.com/javase/7/docs/api/org/w3c/dom/Document.html
- https://gate.ac.uk/wiki/twitie.html
- http://twitter4j.org/en/

Jeffrey Lee
Seminar 4

1. Twitter is a social media network that allows users to post short statements, as well as media such as photos and video. It is best for updating large crowds very quickly.

2. An interesting application of using Twitter for Opinion Mining is in the stock market. In 2013, DCM Capital launched an application called the "DCM Dealer" which allows stock trades to analyze Twitter's opinions on certain products or companies to assist them in stock trading. The application was a follow up to DCM's Twitter Fund, a mutual fund that invested money based on the same Twitter data. While the application is no longer live, it is a cool use of the same technology we are using in class today. DCM's research found that positive tweets general coincided with a company's stock rising, and therefore was a strong argument to purchase stock in a given company.

3. Twitter4J is very useful for accessing the Twitter API from Java. It simplifies the connection between the program and Twitter with zero extra add-ons needed. Its main features allow you to perform actions you can on the Twitter website, such as posting tweets, getting your timeline, and more. Further, you can get search results and even stream from a certain user.

4. The objective is to pull the last 10 tweets that reference a given keyword. The data is stored in JSON format in data/keyword/*QUERY*/.

5. Tweets
    a. Keyword: lacrosse, Tweet: RT @Abigail_Smith22: First night of the spring season with @FriscoFury in Dallas.
    b. Keyword: bbc, Tweet: RT @bbcmundo: El petróleo domina la economía de Venezuela y representa prácticamente la totalidad de sus ingresos de exportación
    c. Keyword: Barcelona, Tweet: RT @relymp: En un mismo día se juega Real Madrid - Barcelona y se reúnen Trump con Kim Jong-Un

6. SearchByUser pulls the timeline of latest tweets from a provided User ID.

7. After searching Justin Bieber's tweets and comparing to the tweets retrieved by his user ID, it is clear that this function works as it is supposed to.

8. SearchByLocation live streams the Tweets posted from a certain bounding box of latitude/longitude coordinates.

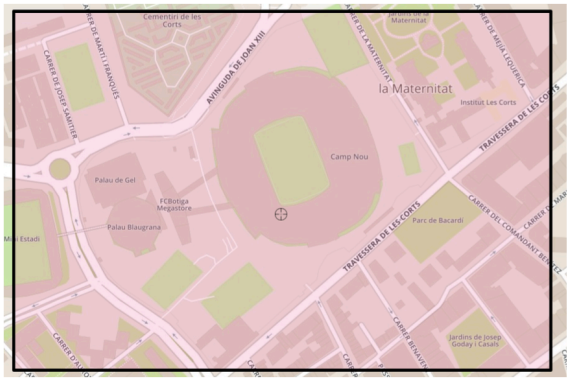
9. Tweets found

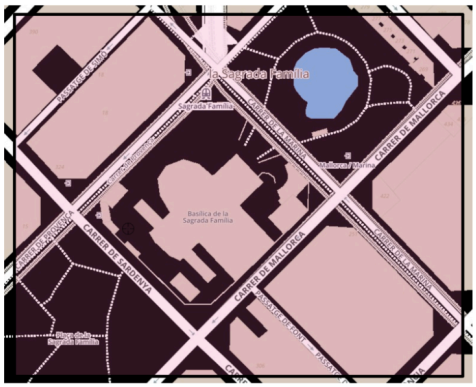
    a. Placa de Catalunya:
    ID: 1105114421649649664
    User: La Salle Manlleu
    Text: Un grup d'alumnes de #4ESO i de #1bat visiten el centre del Banc de sang i teixits de Barcelona #formació… https://t.co/sgs4Mlrr4w

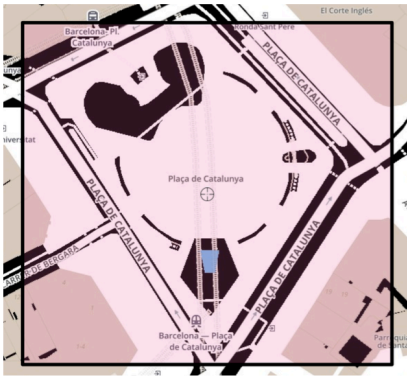
b. Camp Nou:
   ID: 1105115120634601472
   User: SN59
   Text: Le réal vainqueur de la ligue des champions #Zidane

c. Sagrada Familia:
   ID: 1105113497774493697
   User: Hvannadals-hnúkur /@CatalanCouncil
   Text: @DeeJaY_DarE @bombers_man @KRLS Ser investit i declarar la independència de catalunya. Més clar aigua.

Boxes used:







10. TwitIE is Information Extraction and NLP that is meant for brief social media posts, such as Twitter. This GATE plugin can identify the language of the tweets, tags emoji's, usernames, links, and normalizes text.

11. Information Found
    a. UserID = NHSMilton, Organization = NHS, Date/Time = Sat Nov 18 09:02:14 +0000 2017
    b. UserID = Shaker_aphra, URL = https://t.co/wB15ZZV15W, Locations = Turkey, US
    c. BBC, CNN, SKY
    d. Language = es, according to Twitter as found in the Tweet annotation