@ProjectNYC

Redefining NYC Tweets – Project Report

Maxime Goovaerts & Rohit Bhatia

CSE 891

Spring 2013

Intro

Project Goal

The goal of this project was 1) Categorize and cluster geocoded tweets in and around New York City, and 2) Identify trends within these clusters.

Real World Application

Gathering tweets and categorizing them has an infinite range of possible applications. The Twitter API allows researchers to collect vast amounts of geocoded data in real time. The problem lies in properly categorizing them for your application. Tweets could potentially be used to monitor the diffusion of an epidemic disease (Lampos *et al*., 2010) or to identify sales trends for market research.

Project Development

Data Collection

The project data was gathered directly from Twitter using their API and a geocode delimiter of NYC (-74,40° -73,41°).

Once the code was written to stream tweets into a tab delimited text file the desktop settings were adjusted to stay on for the sixteen day period. We also gathered several practice sets of data before settling on the final formatting.

The final set of data was collected from ~5pm 2/27/2013 to ~8pm 3/15/2013. The total number of gathered tweets was 1,075,833.

Data Preprocessing

The data preprocessing portion was by far the most complex and challenging part of this project. We attempted various different approaches before settling on our final, and most effective sorting method.

Originally we had picked four categories of tweets: entertainment, personal, work, and politics. We decided to create a training data set by sorting two thousand tweets by hand. We also developed a list of words we thought were key to allocating tweets into the different categories. Our target was to have *Weka* find trends in the different categories, such as special words or

phrases. However, this brought no results; *Weka* crashed when attempting to parse through our data.

We then used *Apriori* to do the same task. First the tweets had to be properly formatted: each word was to be separated by a comma. We had to run *Apriori* on the training data to identify stop words. This was done by setting *Apriori* to identify words, or phrases, which appeared in over 50% of the gathered tweets. Once the stop words were removed, *Apriori* analysis on the training data provided greater insight. Experimenting with *Apriori's* settings, min/max confidence and support, led to the identification of different association rules.

At this point we realized that more categories would be useful, to discover a larger variety of trends. Our final set of categories was set as follows: day, entertainment, location, love, sports, weather, work, politics, pop culture, and art. The words identified by *Apriori* were a good start but more rules were required and so a process of scrolling through tweets and identifying words that fit exceptionally well with the category was added. We wrote a program to apply the new rules being developed to our gathered tweets. This program displayed the total number of tweets thrown into each category, which was helpful in tracking our progress as new rules were added.

This sorting program first prepared the tweets for analysis by replacing certain punctuation {# ? ! , . - * = + ~ ^ ( )} with spaces. We picked these symbols by looking at basic tweet styles and composition. Given our sorting method, based on finding substrings, most punctuation would be disruptive to categorization. Replacing them with spaces facilitated the search for specific substrings. Next in the categorization code was the rule section. Once a tweet was successfully matched up to a rule its' category was set. In other words, the rules listed last would only be triggered if the tweet failed to fit any of the previous ones.

Therefore rule ordering was of great importance. For example, if a tweet contains both "Obama" and "baseball" which category should it be sorted into: *politics* or *sports*? Whichever rule was listed first, in our sorting code, had priority.

Rules in each category were given a priority level based on how it pertained to the topic, for example "Obama" has a higher priority than "Washington" in the politics category. The likeliness of someone's tweet being about politics is almost certain when they include the name "Obama" but not necessarily when containing the word "Washington."

Due to this issue our sorting program had two "tiers", going through the higher priority rules first, before moving to the lower priority rules to insure a higher sorting accuracy.

Data Analysis

Tweets were allocated into categories based on their content but also into administrative units (e.g. zip codes here) based on their geographical locations. This required overlaying the tweet locations over a shapefile of NY zip codes (https://www.Google.com/fusiontables/data?docid=1MtC0oa6oq1WPuT9AydZK6IFdLketH8pwf8Zppg#map:id=8) using the program SpaceStat (http://www.biomedware.com/?module=Page&sID=spacestat). However, some tweets could not be assigned to a zip code either because they weren't from NY or their location fell in the water due to geocoding errors or maybe a user was in transit.

The next step was to derive for each ZIP code the predominant category of tweet. Our first analysis was done without adjusting the weight of categories (*Figure 1*). However, we felt this distribution was too uneven and that more interesting results might be obtained after readjusting the weight of each category.

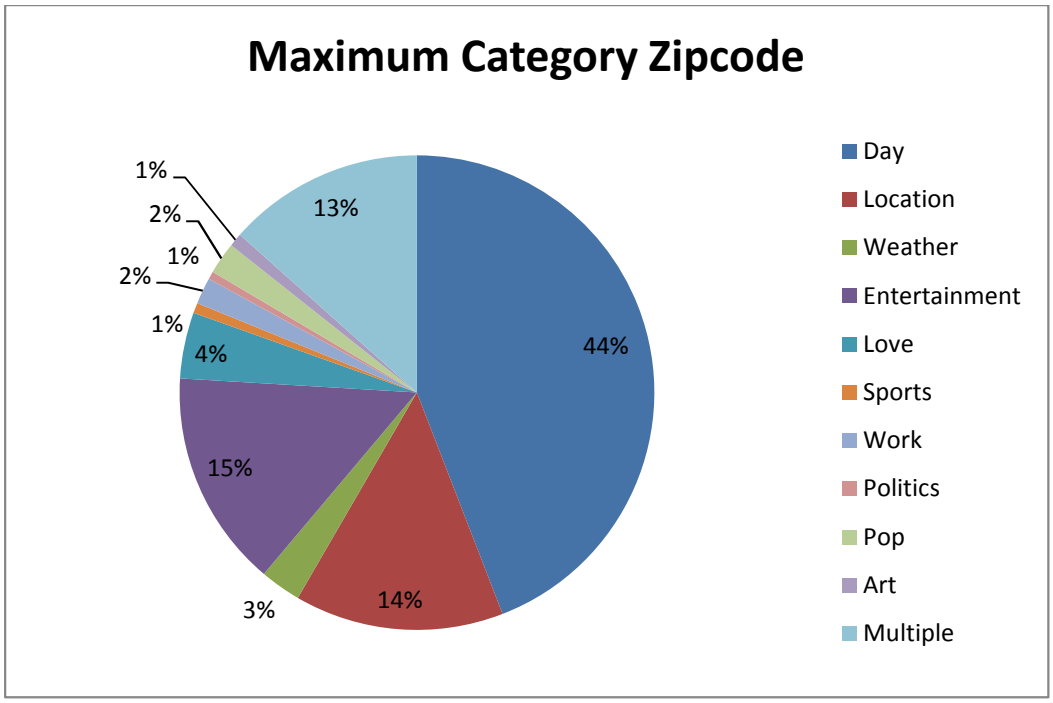

Figure 1: pie chart displaying zip code distribution by category of tweets

The weight of each category was adjusted based on the number of tweets collected in each category. For example: each *political* tweet, which only made up 2% of our data, would be worth ten tweets compared to *weather* (20%). This readjustment of data, which was referred to as "Normalization" (*Figure 2*), led to a much more uniform distribution of prevalent tweet categories among zip codes.

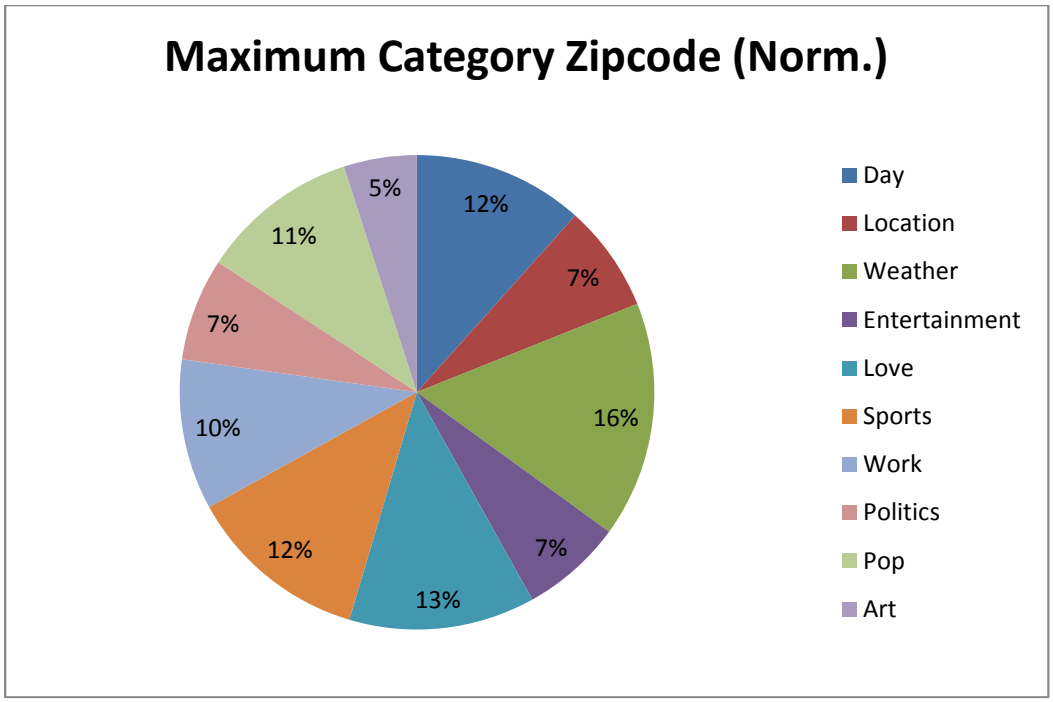

Figure 2: pie chart displaying zip code distribution, after normalization, by category of tweets

Website

http://rohitbhatia338.wix.com/projectnyc

Once categorized and assigned to their respective zip code, the data was added to Google Fusion Tables (*Figure 3*).

1-100 of 1,075,832

Date and Time	Location	ZipCode	Category	Tweet
02/27/2013;17:04-57	40.63318411,-73.94686323	11210	none	You ain't the only one tryna be the only one.
02/27/2013;17:04-59	43.2071309,-77.65555932	14615	pop	When I throw it back I got this nigga speakin Fren...
02/27/2013;17:05-00	40.04016759,-74.86426452	8016	none	Wow , did you really just accuse me of that ?
02/27/2013;17:05-03	40.7190001,-73.8188806	11367	weather	I am leaving this lovely 50 weather for wat looks...
02/27/2013;17:05-06	40.74224026,-73.93537389	11101	none	Bana bir masal anlat baba...
02/27/2013;17:05-07	40.7203238,-73.98383224	10002	location	Paws up @ 176 Stanton St. http://t.co/HutZXd4fkU
02/27/2013;17:05-07	40.650173,-73.959844	11226	location	I'm at fisherman's Cove (Brooklyn, NY) http://t.co...
02/27/2013;17:05-07	40.67542855,-73.95607791	11216	none	IT'S TIME FOR AN ASIAN POPE
02/27/2013;17:05-10	40.7225553,-73.60976097	11530	none	@Ireland_Hudson r u serious ? ur teeth were perfec...
02/27/2013;17:05-10	40.35481379,-74.0086132	7760	none	I'm okay, you're okay, everyone is okay.
02/27/2013;17:05-10	43.0207237,-73.9780241	12020	day	@GoldsteinHannah Life will be as easy or as hard a...
02/27/2013;17:05-10	40.773839,-73.87122	11371	location	My #nailart (post a night of rock climbing in #Bro...
02/27/2013;17:05-11	40.675905,-73.844699	11417	pop	Love. Joy. Faith. by MarinCreations http://t.co/ZD...
02/27/2013;17:05-14	40.75113089,-73.97547609	10017	sports	@RealMichaelKay Fans wrongfully blame Melo for Kni...
02/27/2013;17:05-15	40.39824053,-74.11302933	7748	day	Epic day today! I'm growing up and seeing doors op...
02/27/2013;17:05-15	40.67011463,-73.7149817	11580	none	Watch http://t.co/6Okd3HmXU6
02/27/2013;17:05-16	41.00679983,-73.87128887	10522	sports	@Kost_Nick congrats bro big win

Figure 3: sample of original data uploaded to Google fusion tables

Based on the Latitude and Longitude values, all of the tweets were plotted to Google maps. (*Figure 4*).

Figure 4: pointer map

A few changes were made to this map, to display Google maps and charts, to compare regions, using the tweet categories as the measure (*Figure 5*).

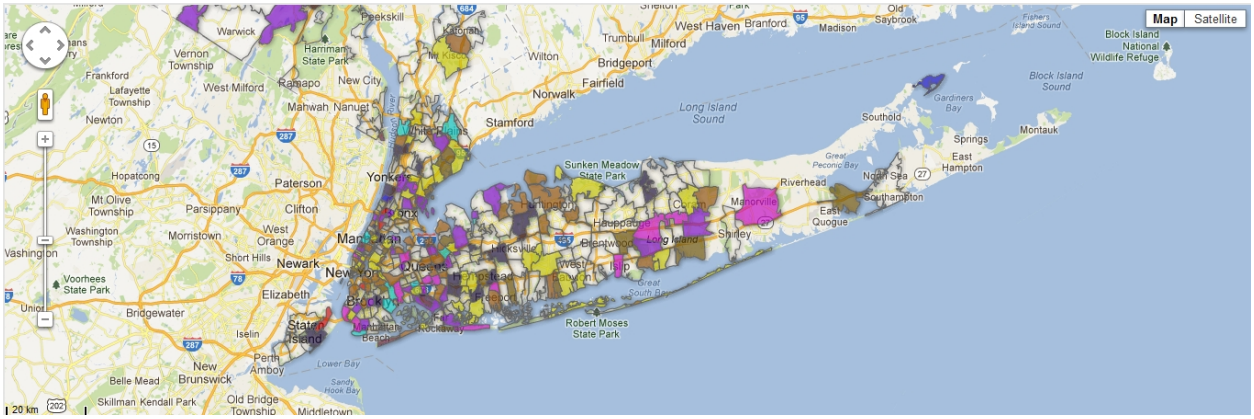

Figure 5: original zip code map

However, this did not fully serve the purpose of the project. Thus, a multi-layer map was built. Based on the zoom level, data would be represented with different levels of resolution. The zoom levels are defined from 1-17. 1 being the furthest (global view) and, as we zoom in, the zoom level increases and it gets as close as to the street view, on zoom level 17. Two separate Google maps were built, and each map is known as a layer. A JavaScript was written, to change the layer depending on what the zoom level was. In the end we chose to not to use this map.

Different Google charts were created to represent the trends in our data (*Figures 6-9*).

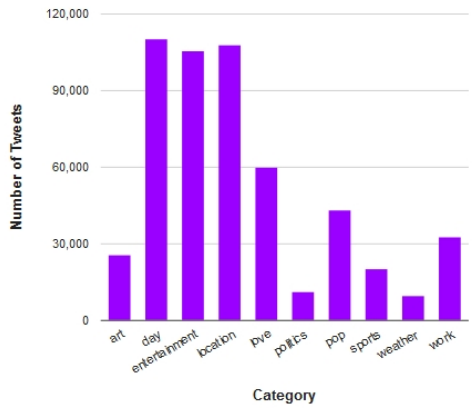

Figure 6: bar chart of category distribution

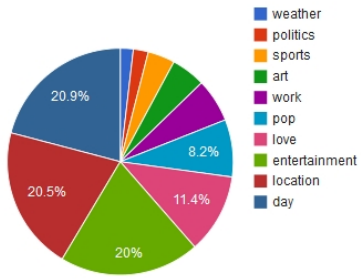

Figure 7: pie chart of category distribution

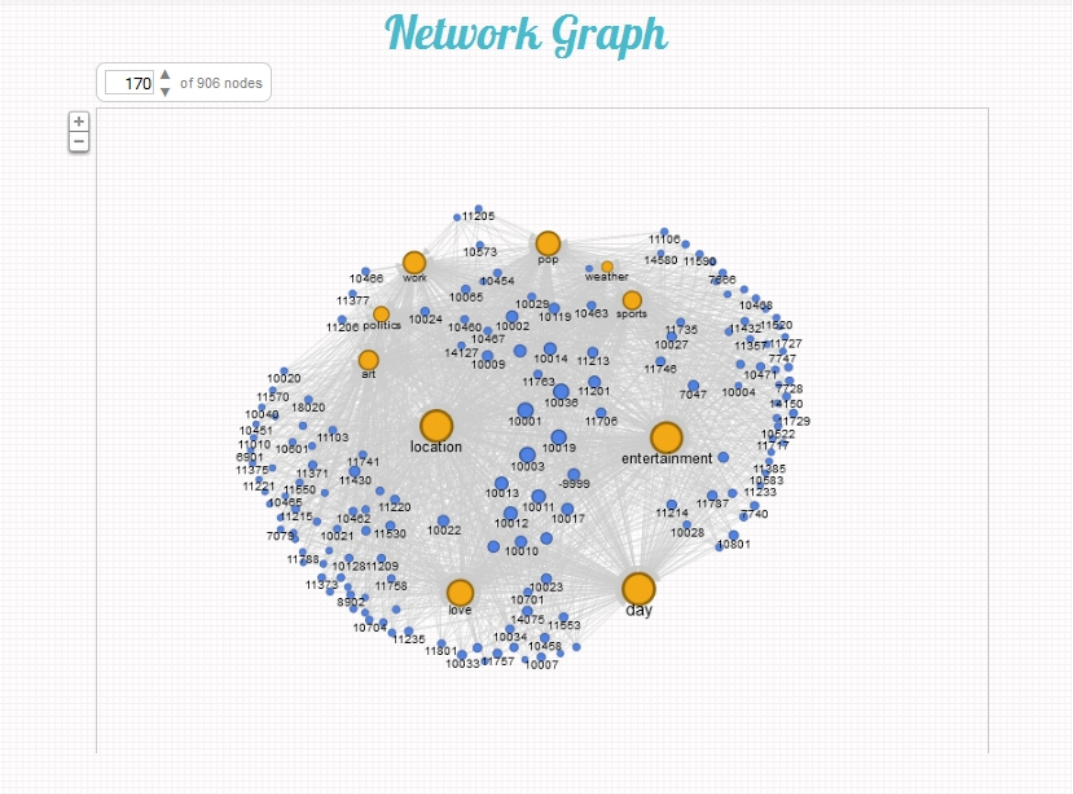

Figure 9: network graph

Once the data representation was completed, our next challenge was to put the data on the website.

At first, a basic website was built to check whether all the features of Google maps and charts were represented or not. However, there were issues with both Google maps and charts. With Google maps, the pointers were used to represent each tweet but none of the details were coming

up. This same issue was also observed with the charts. The database had to be made publicly available, so that any user could use it in other web sessions.

We were having issues with this website builder so we decided to try using another online site building tool, called “Wix”. Originally this new website only had one map with markers or pointer displaying the tweets. However, this was not the only requirement for the project. The tweets were also supposed to be grouped based on their zip code, and each zip code had to be displayed as a polygon (a term used to display a region on Google maps.

At first, the data uploaded on fusion tables, did not contain a dimension or column for zip code. We tried to append the data set with a new column, but this led to many errors. Thus a new data set was uploaded to fusion tables and then, two different maps were created, one for the individual pointers and another one for the zip code polygons.

When this data was uploaded on the map, the data and the maps were overlapping, as a result of some internal restriction from Google maps. At different zoom level, the map was overlaying the pointers and zip code.

The team then decided to have map-specific data sets: one for the pointers which uses the geocoded location gathered from Twitter, and the other one using the zip code polygon which was extracted from an online data set about NYC zip code. This would allow the zip code map, with fewer data, to run smoothly. Once these new data sets were uploaded we no longer had issues.

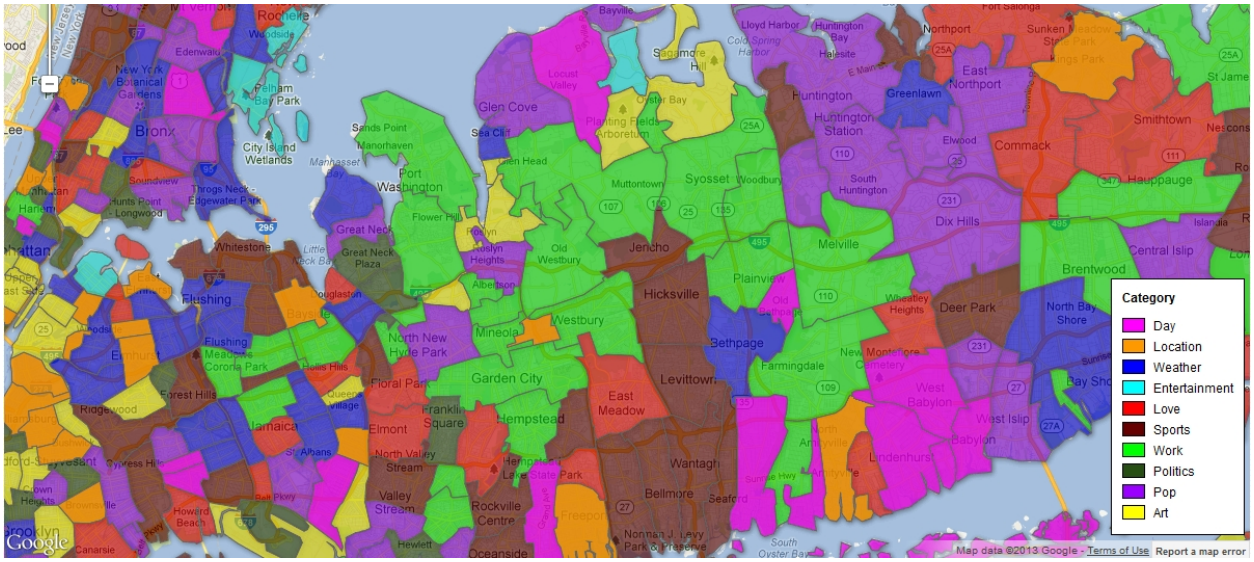

Figure 10: normalized zip code map

Another issue was that some zip codes had multiple categories as their majority category. All the zip codes with multiple majority categories were marked as “multi” and displayed with a separate color (*Figure 10 & 11*).

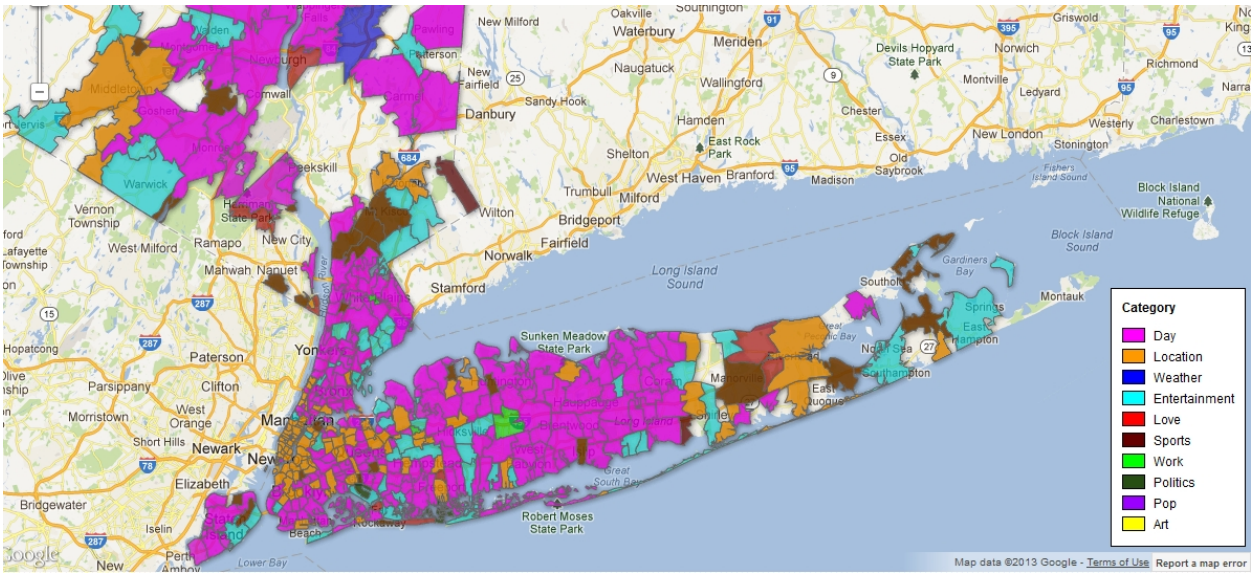

Figure 11: zip code map

Another analysis was to look at geographic clusters of tweets. We found that most of them were located near famous places in New York. Thus, a multilayer map was added to the website: a layer of the polygon maps and a layer of the new key areas. These key areas were represented with different types of pointers on the Google map (*Figure 12*).

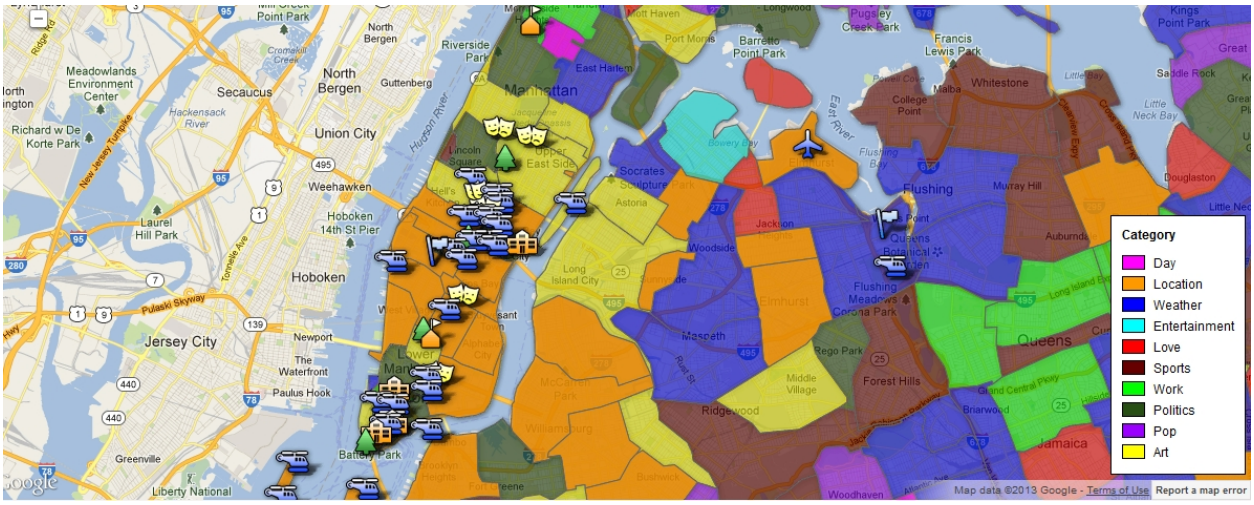

Figure 12: map displaying key areas and normalized zip codes

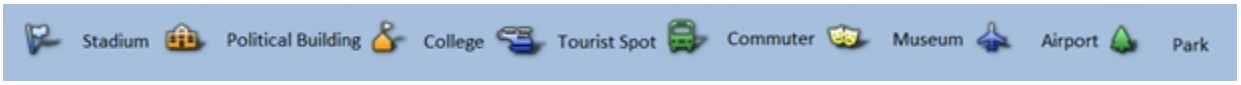

Figure 13: legend for key areas in the map above

Results

Categorizing

Around half of all gathered tweets (half a million) were categorized. Of the ten categories *day*, *entertainment*, and *location* have the clear majority with ~20% each of the categorized tweets (*Figure 14*).

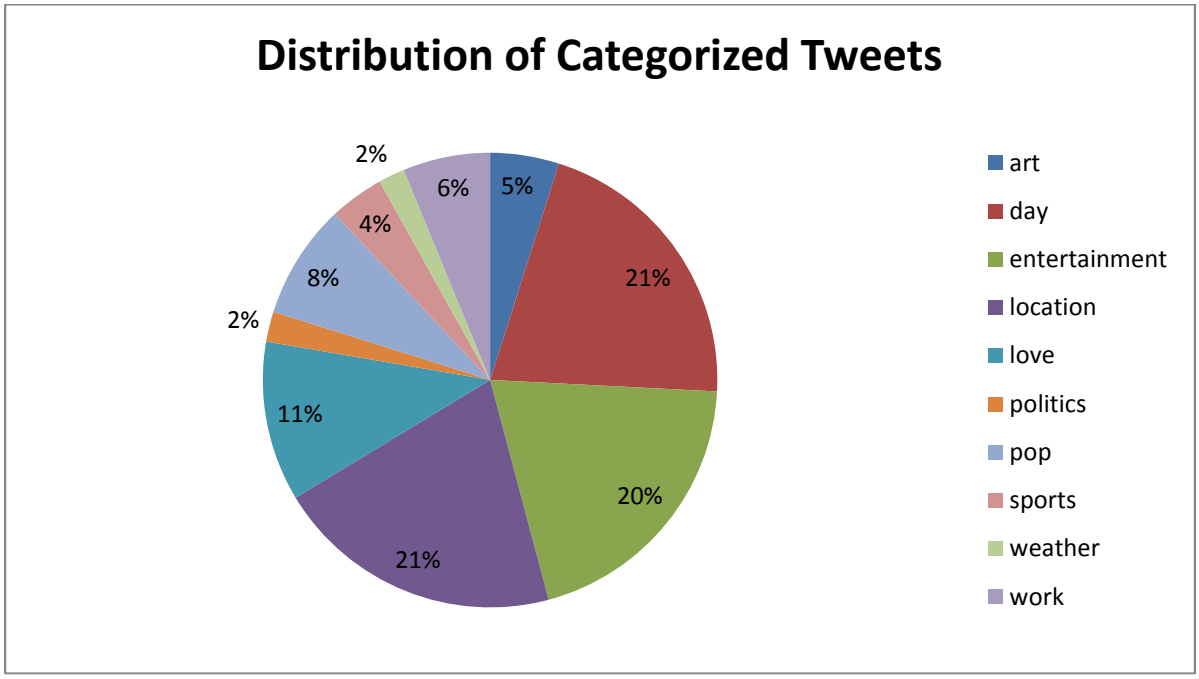

Figure 14: categorization distribution not including the "none" category.

Although the categorization code was only able to sort around half of all gathered tweets its accuracy was much more impressive. From a random sample of 2000 tweets, 80.6% of them were correctly sorted (*Figure 15*).

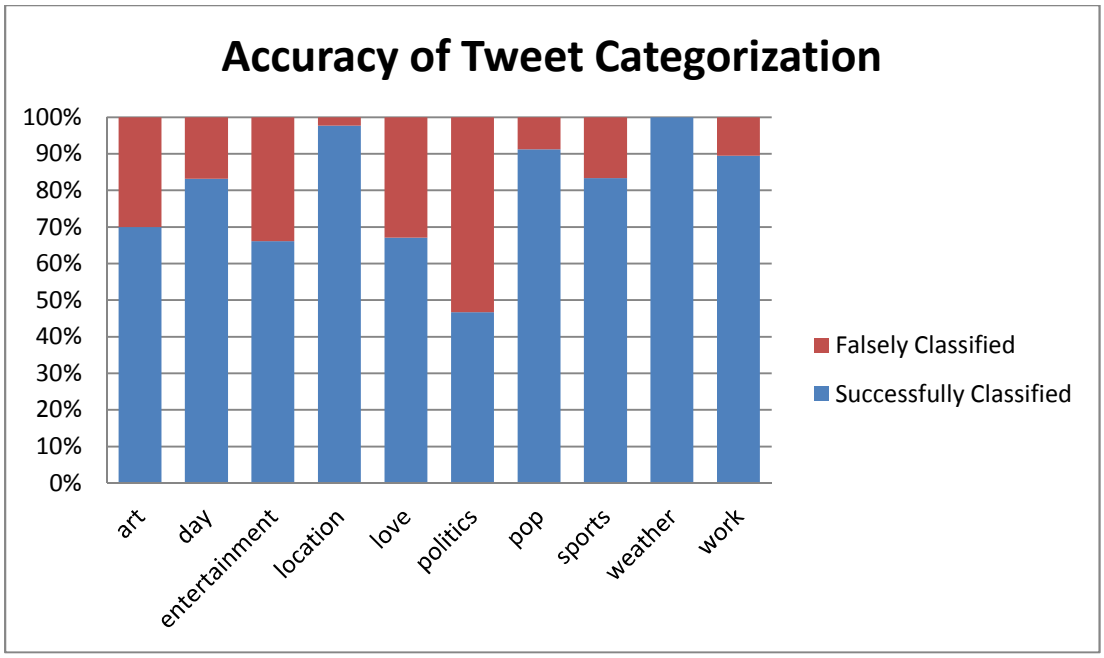

Figure 15: accuracy of sorting split by categories (based on a random sample of 2000 tweets)

Interestingly enough, the sorting accuracy varied a lot among categories. The lowest accuracy was observed for *politics* (47%) and *entertainment* (66%) while *location* (98%) and *weather* (100%) tweets were the most accurate. The *location* category had such a strong accuracy because we identified Foursquares' tweet format. Foursquare is a program that users use to "check in" to locations with tweets.

Map

We picked out a variety of key locations among NYC; many of which had a correlation with our gathered data. For example, Yankee Stadium and Madison Square Garden lie in a zip code region that includes a majority of s*port* tweets. Touristy spots such as the Statue of Liberty and Time Square lie in a *location* region. However, not- all results could be easily explained. Citi Field, a fairly popular baseball field in Queens, didn't fall into the *sport* category that would be expected at that location. Possible reasons are that during the period of tweets gathering no games were played or because the prevalent category of that region, *weather*, was just too noisy.

Some unexpected trends were found, such as the high volume of *work* tweets emerging from commuting neighborhoods outside of Manhattan. It turns out that a good amount of people tweet on their way to work; this trend closely follows highway 495, coming in from East of NYC (*Figure 16*).

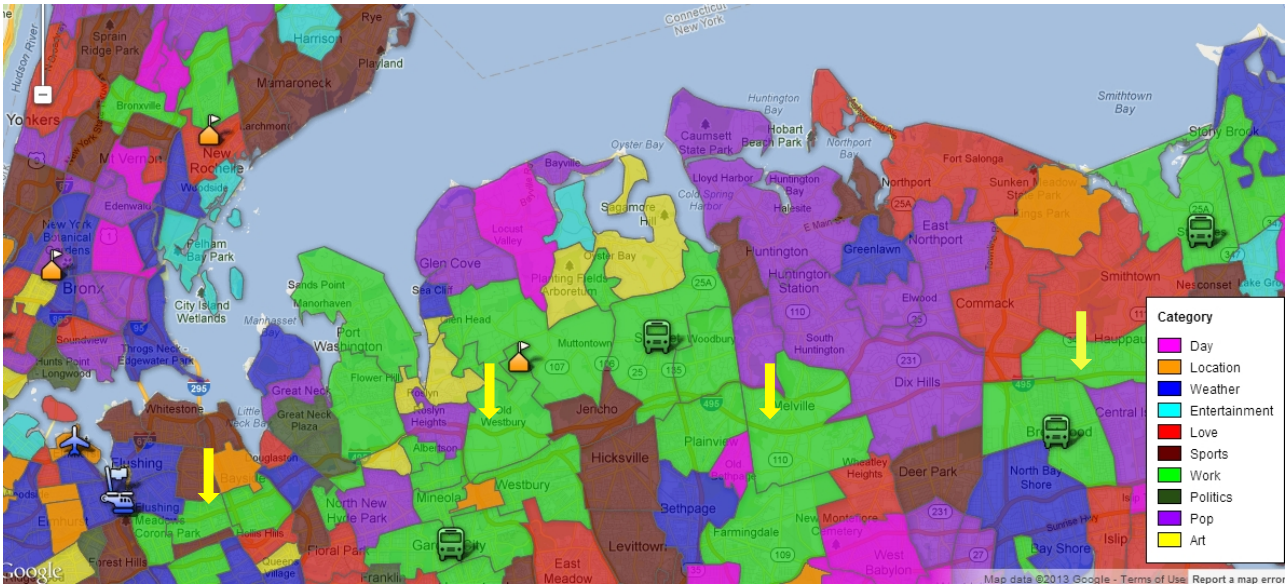

Figure 16: highway 495 displayed by the yellow pointers

Improvements

Data

The geographic delimiter we used to gather tweets was not specific enough and we ended up with tweets as far away from NYC as Niagara Falls. If this were adjusted it would reduce the amount of tweets that needed to be sorted, allowing a stronger focus on our intended region.

Categorization could also be improved. Further analysis of rules, especially in the low accuracy categories, should help identifying which rules are causing the highest percentages of miscategorization. While testing for accuracy I noticed a few inaccurate rules; for example using the keyword "Kennedy" led to a *political* classification. This is especially problematic in NYC where one of the airports is named after Kennedy. We also noticed that foreign language tweets were almost always misunderstood by our categorization code. This could be solved by adjusting our tweet gathering code to only gather English tweets.

Website

The pointer map portion of the website could be improved by having a more extensive tweet selection option. For example, the user should be able to select displayed pointers by time range, category and zip code. The zip code map could be improved by having a sidebar of statistical data. This data would change when the user clicks on different zip codes.

A very large but possible improvement to this website would be making it update itself. Rather than using static data a script would gather tweets, categorize them, and upload them to fusion tables for immediate deployment on the site.

References

Lampos, V., De Bie, T., & Cristianini, N. 2010. Flu detector-Tracking epidemics on Twitter. Available at http://link.springer.com/chapter/10.1007/978-3-642-15939-8_42.