Final Project Report

CSE 592.01- Social Networks

Global Sentiment Analysis On Twitter Data

By:-

Rohan Mehta (108648007)

Ruchi Khandelwal (109596950)

Sindhuri Mamidi (109596303)

ABSTRACT

In today's world, Social media is the platform where we as users presents our views, opinions etc. Common examples are of tweets on twitter and posts on Facebook etc. All these are textual information.

Textual Information can be broadly categorized into two main types: facts and opinions. Facts are objective expressions about entities, events and their properties. Opinions are usually subjective expressions that describe people's sentiments, appraisals or feelings toward entities, events and their properties. The concept of opinion is very broad. In this project, we only focus on the expressions that convey people's negative or positive sentiments and perform the sentimental analysis on the twitter data and try to find Correlation between the sentiments of the tweets and various other factors like how many number of retweets , geographic location, number of followers and friends count. We also see how the personality or social status of a user effects the retweet count through our analysis of data.

We have classified the polarity of the text in a review or sentence on the feature/aspect level. This real time sentimental analysis of twitter data is performed using python code and by also an existing API. Our python code automatically classify the textual tweets into positive, neutral or negative depending on the word list for the same. Once the scores are computed we take the score as a metric and perform analysis on how the sentiments play a part in the social network aspect of twitter data.

DATA SET

We have made use of twitter data for the analysis. In order to get the data in large amount and perform the analysis, we scrap the twitter data for all the users.

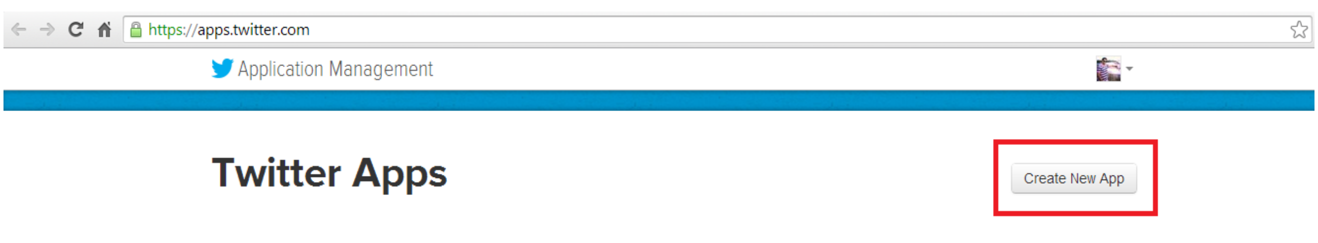
- We have scrapped twitter data for 15 days.
- Each day contains data of around 30,000-40,000 tweets, totaling to around 4, 00,000-6, 00,000 tweets for the analysis and coming to the conclusion.

How did we scrap the data?

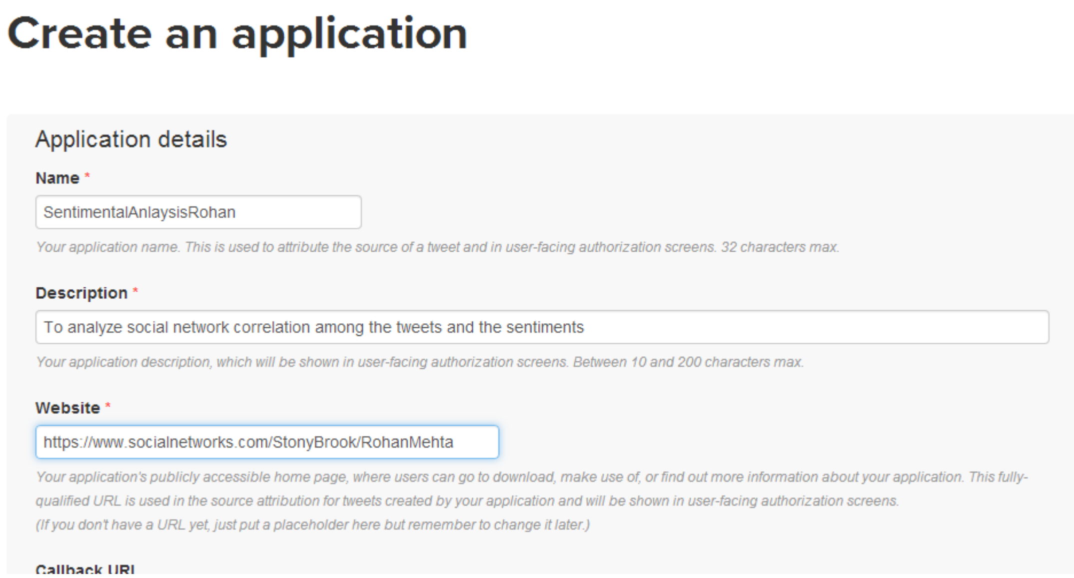
- First of all, we need to install oauth2 on the system in order to run the twitter API. In order to carry out that, we did run the command below on our system where python is installed.

Pip install –U oauth2

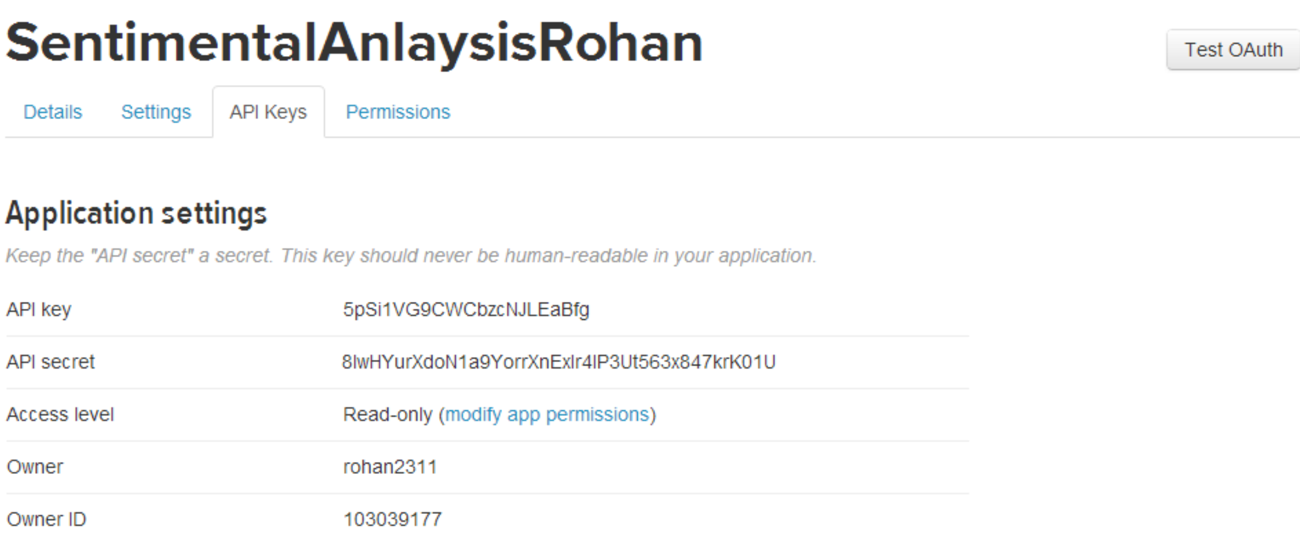
- To access the Twitter API, we had to setup a Twitter Developer account.
- Following steps were carried out :-
    1. Twitter account has to be created. In our case, Rohan had a twitter account.
    2. We went to https://dev.twitter.com/apps and logged in with twitter credentials.
    3. Click 'Create New Application'.



4. Once you fill out the form and click on the rules and regulation part, we can get our self an application created.



5. Once the application is created, on tab "API Keys", Click "Create my access token." Once created, twitter generates the access_token_key , access_secret_key , consumer_key, consumer_secret which needs to be added inside the python script to fetch the tweets.



6. Now once it is created, now we have to stream twitter API with pour credentials provided by twitter.

Twitter API

To access the current Twitter stream, we had to send a GET request to https://stream.twitter.com/1.1/statuses/sample.json. All this was performed through the python script named twitterStream.py

So script construct, sign, and open a twitter request using the hard-coded credentials above and fetch the samples from twitter.

```
import oauth2 as oauth
import urllib2 as urllib

# Set up by making a developer account on twitter
access_token_key = "103039177-NOnGei3vjUcz0jBVs6qUygf7hnY9xZrPlubWJiLJ"
access_token_secret = "MTx7xaZHfcQOmO3dpPzaEvHCf3JFCW8oWvQMEA6Iewq0l"

consumer_key = "5pSi1VG9CWCbzcNJLEaBfg"
consumer_secret = "8lwHYurXdoN1a9YorrXnExIr4IP3Ut563x847krK01U"

_debug = 0

oauth_token    = oauth.Token(key=access_token_key, secret=access_token_secret)
oauth_consumer = oauth.Consumer(key=consumer_key, secret=consumer_secret)

signature_method_hmac_sha1 = oauth.SignatureMethod_HMAC_SHA1()

http_method = "GET"


http_handler  = urllib.HTTPHandler(debuglevel=_debug)
https_handler = urllib.HTTPSHandler(debuglevel=_debug)

...
```

After, we add the authorization code, one can simply fetch the data. The data twitter allows to fetch is in the JSON format.

We run the script for 15 consecutively day to get the sample.json for each day. We provide the name for the JSON files with the dates we fetch the data and it lets us distinguish between the each individual file.

```
def twitterreq(url, method, parameters):
  req = oauth.Request.from_consumer_and_token(oauth_consumer,
                                              token=oauth_token,
                                              http_method=http_method,
                                              http_url=url,
                                              parameters=parameters)

  req.sign_request(signature_method_hmac_sha1, oauth_consumer, oauth_token)

  headers = req.to_header()

  if http_method == "POST":
    encoded_post_data = req.to_postdata()
  else:
    encoded_post_data = None
    url = req.to_url()

  opener = urllib.OpenerDirector()
  opener.add_handler(http_handler)
  opener.add_handler(https_handler)

  response = opener.open(url, encoded_post_data)

  return response

def fetchsamples():
  url = "https://stream.twitter.com/1/statuses/sample.json"
  parameters = []
  response = twitterreq(url, "GET", parameters)
  for line in response:
```

Results of Fetching Data from Twitter

After running the script we got the files for each day in JSON format.

w20140324-184101.json	3/27/2014 8:22 AM	JSON File	63,424 KB
w20140324-232828.json	3/27/2014 8:23 AM	JSON File	44,945 KB
w20140325-195730.json	3/27/2014 8:23 AM	JSON File	0 KB
w20140325-200119.json	3/27/2014 8:23 AM	JSON File	5,171 KB

The challenge with this data was some inconsistency as sometimes data was null for specific fields in it and made it difficult to analyze but on a whole results were good and easily understandable.

The JSON format looks like the one below:-

{"created_at":"Fri Mar 21 04:29:44 +0000 2014","id":441792791656402944,"id_str":"441792791656402944","text":"We be flash mobbing tomorrow.....in our school cafe. But hey it still counts! #cougar # USF","source":"\u003ca href=\"http:\/\/twitter.com\/download\/android\" rel=\"nofollow\"\u003eTwitter for Android\u003c\/a\u003e","truncated":false,"in_reply_to_status_id":null,"in_reply_to_status_id_str":null,"in_reply_to_user_id":null,"in_reply_to_user_id_str":null,"in_reply_to_screen_name":null,"user":{"id":395955792,"id_str":"395955792","name":"MacKenzie Ball","screen_name":"MacKenzieBall2","location":"","url":null,"description":null,"protected":false,"followers_count":31,"friends_count":47,"listed_count":0,"created_at":"Sat Oct 22 14:35:14 +0000 2011","favourites_count":10,"utc_offset":-36000,"time_zone":"Hawaii","geo_enabled":false,"verified":false,"statuses_count":75,"lang":"en","contributors_enabled":false,"is_translator":false,"is_translation_enabled":false,"profile_background_color":"1A1B1F","profile_background_image_url":"http:\/\/abs.twimg.com\/images\/themes\/theme9\/bg.gif","profile_background_image_url_https":"https:\/\/abs.twimg.com\/images\/themes\/theme9\/bg.gif","profile_background_tile":false,"profile_image_url":"http:\/\/pbs.twimg.com\/profile_images\/3327020737\/3d3a7b8fca6b6921c986bf1aa0def130_normal.jpeg","profile_image_url_https":"https:\/\/pbs.twimg.com\/profile_images\/3327020737\/3d3a7b8fca6b6921c986bf1aa0def130_normal.jpeg","profile_banner_url":"https:\/\/pbs.twimg.com\/profile_banners\/395955792\/1364516892","profile_link_color":"2FC2EF","profile_sidebar_border_color":"181A1E","profile_sidebar_fill_color":"252429","profile_text_color":"666666","profile_use_background_image":true,"default_profile":false,"default_profile_image":false,"following":null,"follow_request_sent":null,"notifications":null},"geo":null,"coordinates":null,"place":null,"contributors":null,"retweet_count":0,"favorite_count":0,"entities":{"hashtags":[{"text":"cougar","indices":[78,85]}],"symbols":[],"urls":[],"user_mentions":[]},"favorited":false,"retweeted":false,"filter_level":"medium","lang":"en"}

As we can see, each tweet has various fields associated with it. So once, we got that we could easily classify which fields are necessary for analyzing the social network aspects in twitter data.

Some of the fields which we make use in our analysis are:-

1) Text – Actual tweet text used for computing the sentiment score.

2) Name – username which helps identify the social status of a person

3) Followers count – number of followers for the user

4) Friends count – number of friends for the user.

5) Time zone – zone in which tweets were posted helps us include geographical correlation with sentiments.

6) Retweet counts – it helps us see the retweet network correlation with sentiment score, social status of the user etc.

Now there were several challenges in handling the JSON format files and perform analysis, there were lot of fields which were not required in our analysis and hence we decided to convert the JSON format into an excel format which makes the analysis easy and represents information in a tabular and manageable format.

CONVERSION OF JSON TO EXCEL

In order to convert the JSON format into excel format, we have used to code it using python. The sample code is as below:

```
## Store the regexps for matching
created_pat   = re.compile(r'{"created_at":', re.VERBOSE)
id_pat        = re.compile(r'^\s*"id":', re.VERBOSE)
name_pat      = re.compile(r'^\s*"name":', re.VERBOSE)
followers_pat = re.compile(r'^\s*"followers_count":', re.VERBOSE)
friends_pat   = re.compile(r'^\s*"friends_count":', re.VERBOSE)
timezone_pat  = re.compile(r'^\s*"time_zone":', re.VERBOSE)
retweet_pat   = re.compile(r'"retweet_count":', re.VERBOSE)
lang_pat      = re.compile(r'^\s*"lang":"[a-z]*"}', re.VERBOSE)

#---------------------------------------------
## Host 1

loadInpFile = "".join([cmdArgs.inp, ".json"])
lines      = open(loadInpFile, 'r')

outfile    = open('./tweets_text.txt', 'a')

#update excel sheet
book = xlwt.Workbook()
sheet1 = book.add_sheet("Sindhuri_1")

sheet1.write(0, 0, "ID Number")
sheet1.write(0, 1, "User Name")
sheet1.write(0, 2, "Followers Count")
sheet1.write(0, 3, "Friends Count")
sheet1.write(0, 4, "User TimeZone")
sheet1.write(0, 5, "Retweet Count")
row = 0
#col = 0 #id_pat takes care of this
```

After converting it to excel, the file looks like:

	A	B	C	D	E	F	G	H	I
1	ID Number	Text and h	User Name	Followers (	Friends Cc	User Time.	Retweet Count		
2	443580873	"We took s	#CLOUTG:	28	110	null	0		
3	443580874	"RT @Aye	\u2764\ufe	1013	954	"Pacific Time (US & Canada)"			
4	443580094	"LONG LIV	\u2728Tre(	1498	586	"Central Ti	5		
5	443580885	"Tomorrow	Kevin Soda	199	367	null	0		
6	443580891	"@richrich	P.T. MoOL	831	1923	"Pacific Tir	0	Life im just Roll N	
7	2377917798								
8	443580894	"@obfuscu	Tom Sante	969	518	null	New York	0	Jason Dixon
9	66432490								
10	443580907	"58% OFF	puump	36	0	"Bangkok"	0		
11	443580915	"@LanaPa	NeVeR Lo:	571	986	null	Fort Hood	0	Lana Parrilla
12	129400817								
13	443580920	"@KingJua	Pride's RO	1482	2001	null	0		
14	53604430								
15	443580925	"RT @3Rd	PrettyKei\u	2541	1266	"Central Time (US & Canada)"			

We are interested only on the User ID number, the tweet and their corresponding hashtags, the username, followers, friend count and the user time zone as well as the retweet count. We have only taken those specific fields from the huge JSON files and converted to human readable excel format.

SENTIMENTAL ANALYSIS USING PYTHON CODE

The basic task of sentiment analysis is to classify the emotional degree of a given word in a document or sentence--whether the expressed opinion is positive, negative, or neutral. Beyond the basic emotional degree of a statement, sentiment classification looks, for instance, at emotional states such as "angry," "sad," and "happy."

We prepared the sentiment lexicon from various sources and gathered up to 3500 words. Apart from this we have also added an emoticon dictionary and an acronym dictionary. For example ':)' is denoted as positive and ':(' as negative. Apart from this we have also added acronyms such as LOL for laughing out loud,gr8 as great, rofl as roll on the floor laughing etc.

Emoticon	Polarity
:-) :) :o) :] :3 :c)	Positive
:D C:	Extremely-Positive
:-(:(:c :[	Negative
D8 D; D= DX v.v	Extremely-Negative
: |	Neutral

Acronym	English expansion
gr8, gr8t	great
lol	laughing out loud
rotf	rolling on the floor
bff	best friend forever

The maximum length of Twitter message is 140 characters. So we have broadly classified each word to fall under three categories: Positive, Negative and Neutral. Each word has ranges between a sentiment score between -5 to 5.

The lexicon looks as follows:

```
aggressive	-2
aghast	-2
agog	2
agonise	-3
agonised	-3
agonises	-3
agonising	-3
agonize	-3
agonized	-3
agonizes	-3
agonizing	-3
agree	1
agreeable	2
agreed	1
agreement	1
agrees	1
alarm	-2
alarmed	-2
alarmist	-2
alarmists	-2
alas	-1
alert	-1
alienation	-2
alive	1
allergic	-2
allow	1
alone	-2
```

We have then calculated the sentiment score for each tweet as follows:

Sentiment ratio = total count of positive words/total count of negative words.

- if negative>positive
  total score=-1*(total negative/total positive)
- if positive>negative
  total score=(total positive/total negative)
- else
  total score=neutral

The sample code is as given below:

```
total_score = 0
total_neg = 0
total_pos = 0
for each_word in split_words:
    #print("each_word is %s" %each_word)
    if each_word in self.scores:
        this_wordscore = self.scores[each_word]
        if this_wordscore < 0:
            total_neg +=this_wordscore
        elif this_wordscore > 0:
            total_pos +=this_wordscore
        #print("each word and score %s & %s" %(each_word,this_wordscore))
        #total_score += this_wordscore
total_neg = -1*total_neg
if total_neg > total_pos:
    if total_pos == 0:
        total_score = -1*total_neg
    elif total_pos > 0:
        total_score = float(-1*(total_neg/total_pos))
    else:
        print("ERROR!! Total positive cannot be negative")

elif total_pos > total_neg:
    if total_neg == 0:
        total_score = total_pos
    elif total_neg > 0:
        total_score = float(total_pos/total_neg)
    else:
        print("ERROR total negative is less than zero!")

elif total_neg == total_pos:
        total_score = 0

if total_score > 0:
    sentiment = "positive"
elif total_score < 0:
    sentiment = "negative"
else:
    sentiment = "neutral"
```

The output of the above code is:

	A	B	C	D	E	F	G	H	I	J
1	ID Number	Text and h	Sentiment	Sentiment	User Name	Followers	Friends Cc	User Time:	Retweet Count	
2	441792791	"We be fla	0	neutral	MacKenzie Ball	31	47	"Hawaii"	0	
3	441792796	"Had some	0	neutral	Cedric Dozier	753	460	null	0	
4	441792800	"Incredible	0	neutral	Christopher Somers	2521	2372	"Eastern T	Paradise	0
5	462983953									
6	441792806	"#YallFrigg	0	neutral	Joy Collins	10816	7271	"Central Ti	0	
7	441792810	"#YallFrigg	0	neutral	Joy Collins	10816	7271	"Central Ti	0	
8	441792812	"#YallFrigg	0	neutral	Joy Collins	10816	7271	"Central Ti	0	
9	441792824	"RT @iTra	0	neutral	Jennayy	147	156	null		
10	441791906	"Bout to pu	0	neutral	OG PAPA D	807	791	"Eastern T	0	
11	441792826	"\"# Miche	3.5	positive	Michelle (\u2299\u25e1\u2299\u273f)	619	267	"Atlantic T	0	ahjunnie 2
12	1627870345									
13	441792840	"# Last # V	0	neutral	TunisianPhotography	9	19	"Atlantic T	0	
14	441792850	"RT @_sm	0	neutral	axcita	1045	630	"Eastern Time (US & Canada)"		
15	441767028	"your # ca	3	positive	shayla \u262a	1197	1000	"Pacific Ti	3	
16	441792852	"RT @dee	0	neutral	TracyOeltjenbruns	160	228	"Central Time (US & Canada)"		
17	441792677	"@T_Oeltj	0	neutral	Desiree Ponce	122	373	"Pacific Ti	1	TracyOelt
18	349884478	Justine Petersen								
19	384631575									
20	441792871	"thought i	0	neutral	Hunter	153	181	"Eastern T	0	
21	441792882	"# TWELV	0	neutral	MaFe KaRDeNaS	33	18	null	0	
22	441792886	"@derbyol	6	positive	Bill Torre	178	592	"Central Ti	0	CraigJ
23	1024858321									
24	441792886	"RT @Joy	0	neutral	A New Freedom	483	749	"Pacific Time (US & Canada)"		
25	441792806	"#YallFrigg	0	neutral	Joy Collins	10816	7271	"Central Ti	1	
26	441792892	"RT @pure	0	neutral	Juan Mercado	35	119	null		
27	411555044	"@moorep	0	neutral	\u263e	472	165	"Atlantic T	2	b moore

Each tweet now has a sentiment score and the sentiment of it.

SENTENCE LEVEL SENTIMENT CLASSIFICATION

Consider each sentence as a separate unit.

Assumption: Sentence contain only one opinion.

•Task 1: identify if sentence is subjective or objective

•Task 2: identify polarity of sentence.

FEATURE LEVEL SENTIMENT CLASSIFICATION

•Task 1: identify and extract object features

•Task 2: determine polarity of opinions on features

•Task 3: group same features

•Task 4: summarization

Challenges and Drawbacks:

1) Data was in JSON format and it needed to be converted into an easy readable format on which we can perform the sentiment analysis and then the further analysis.

2) To gather relevant data, detect and summarize the overall sentiment on a topic. There can be typos, acronyms, emoticons.

3) To manually annotate each tweet is a big task! Training data needs to be balanced and normalized.

4) The frequency of misspellings and slang in tweets is much higher than other domains.

5) The data can in many other languages apart from English.

<u>Sentiment Analysis using API</u>

We explored several APIs for sentiment analysis like ViralHeat etc. We choose Semantria as it offered the most extensive sentiment analysis for the tweets.

The purpose of running sentiment analysis via an API was to compare the results we achieved with our Python code versus the one that the API generated.

Algorithm Used by Semantria

1. A document is broken in its basic parts of speech, called POS tags, which identify the structural elements of a document, paragraph, or sentence (i.e. Nouns, adjectives, verbs, and adverbs).

2. Sentiment-bearing phrases, such as “terrible service”, are identified through the use of specifically designed algorithms.

3. Each sentiment-bearing phrase in a document is given a score based on a logarithmic scale that ranges between -10 and 10.

4. Finally, the scores are combined to determine the overall sentiment of the document or sentence. Document scores range between -2 and 2.

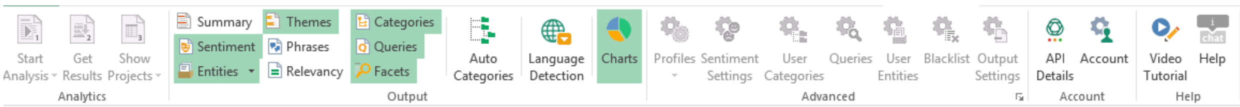
To calculate the sentiment of a phrase such as “terrible service”, Semantria uses search engine queries similar to the following:

“(Terrible service) *near* (good, wonderful, spectacular)”
“(Terrible service) *near* (bad, horrible, awful)”

Each result is added to a hit count, which are then combined using a mathematical operation called “log odds ratio” to determine the final score of a given phrase. In this case, Semantria gave “terrible service” a score of 0.57.

Process Followed for Sentiment Analysis in Excel

We started with creating a developer account with Semantria and downloading their excel plugin.



On registering with Semantria, we get a API Key and a Secret Key which then helps us run analysis on 15000 tweets.

After running the analysis the data looks like this:

	B	C	D	E	F	G	H	I	J
1	Source text	Document Sentiment	Document Sentiment Polarity	Entity	Entity Type	Entity Sentiment	Entity Sentiment Polarity	Theme	Theme Sentiment
2	tomorrow.....in our school cafe. But	0	neutral	tomorrow.... .in our	Quote	0	neutral	mobbing tomorrow	0
3	tomorrow.....in our school cafe. But	0	neutral	#cougar	Pattern	0	neutral	school cafe	0
4	# 37 on my Football Jersey	0	neutral	me why I wear # 37 on	Quote	0	neutral		
5	# 37 on my Football Jersey	0	neutral	#Grind	Pattern	0	neutral		
6	"Incredible atmosphere !	1.5	positive					Incredible atmosphere	0.75
7	"Incredible atmosphere !	1.5	positive						
8	"Incredible atmosphere !	1.5	positive						
9	"Incredible atmosphere !	1.5	positive						
10	"Incredible atmosphere !	1.5	positive						
11	"Incredible atmosphere !	1.5	positive						
12	"Incredible atmosphere !	1.5	positive						
13	"Incredible atmosphere !	1.5	positive						
14	"Incredible atmosphere !	1.5	positive						
15	"Incredible atmosphere !	1.5	positive						
16	"Incredible atmosphere !	1.5	positive						
17	"Incredible atmosphere !	1.5	positive						
18	"Incredible atmosphere !	1.5	positive						
19	"Incredible atmosphere !	1.5	positive						
20	"Incredible atmosphere !	1.5	positive						
21	#ReverbNation #Countrymusic	0	neutral	#ReverbNation	Pattern	0	neutral		
22	#ReverbNation #Countrymusic	0	neutral	#Countrymusic	Pattern	0	neutral		
23	#ReverbNation #Countrymusic	0	neutral	#Nashville	Pattern	0	neutral		

Theme Sentiment Polarity	Auto Category	Auto Sub Category	User Category	User Category Sentiment	User Category Sentiment Polarity	Query	Query Sentiment	Query Sentiment Polarity
neutral						Education	0	neutral
neutral								
						Sports	0	neutral
positive	Climate							
	Climate	Climate_forcing						
	Climate	Climate_feedbacks						
	Space							
	Space	Cosmic_rays						
	Space	Spacecraft_instruments						
	Atmosphere							
	Atmosphere	Planetary_atmospheres						
	Atmosphere	Atmosphere						
	Atmosphere	Atmosphere_of_Earth						
	Atmosphere	namics						
	Atmosphere	Atmospheric_radiation						
	Chemistry							
	Sensors							
	Sensors	and remote sensing						

Thus we can see that Semantria gives us in depth analysis like Document Sentiment, Document Sentiment Polarity, Entity and Theme Sentiment, Category of our tweet, Query Sentiment etc.

Entity and Theme Sentiment

Here the API has found the themes from the tweet and calculated their sentiments. Sentiment analysis done on a just an overall tweet is usually not very useful. Consider this – “The movie was good but the actor was bad”. Now this has a positive sentiment towards the movie, but a negative one towards the actor. And thus the overall sentiment of the tweet is neutral. And hence we need to find sentiment of particular themes or entities.

Consider the following tweet,

"Girls voluntarily elope w\/lovers. On return fabricate story of kidnap & rape 2 escape harsh treatment 4m parents #StopMisuseOfIndianLaws"

Theme	Theme Sentiment	Theme Sentiment Polarity
voluntarily elope	0.099583298	neutral
fabricate story	-0.275729239	neutral
harsh treatment	-0.600000024	negative
4m parents	-0.463385493	negative

"@SVPandRussillo I think Cleveland wants Lebron back so bad that is part of the reason big Z is a GM so it plays into the #retirement"

Entity	Entity Type
Cleveland	Place
Lebron	Person
General Motors	Company
think Cleveland wants	Quote

VLookUp

Semantria generates the results of the analysis in a new document which only contains the source text. But we also need the information from fields C,D,E,F,G. To do this, we do a join operation (VLookUp) on the original file and semantria generated file to merge the two based on the source text.

Original File:

	A	B	C	D	E	F	G	H
1	ID Number	Text and hashtag	User Name	Followers (	Friends Cc	User Time:	Retweet Count	
2	441951315	"RT @niallhoran	vivien	2641	1579	"Eastern Time (US & Canada)"		
3	441687806	"Gareth Gates is an annoying little shit! #	Niall Horar	16205	1013	"Dublin"	Dublin City	1
4	441951317	"# DnBHeaven Radio - Now Playing	DnBHeave	1137	7	"London"	0	
5	441951320	"@Symthic We want to rent servers on c	ps4	3	26	null	0	Symthic
6	1514309444							
7	441951324	"Creative Art! loovee http	LOOVEE	73	47	null	0	

Semantria Generated File:

	A	B	C	D	E	
1	ID	Source text	Document Sentiment	Document Sentiment Polarity	Entity	
2	4d58-981b-	"RT @niallhoran	0	neutral	@niallhoran	Pe
3	4d59-8c6c-	shit! #BigReunion2014 #"	-1.434999943	negative	Gareth Gates	Pe
4	4d59-8c6c-	shit! #BigReunion2014 #"	-1.434999943	negative	#BigReunion2014	Pa
5	4d59-8c6c-	shit! #BigReunion2014 #"	-1.434999943	negative	#"	Pa
6	4067-b518-	"# DnBHeaven Radio - Now Playing	0	neutral		
7	b264-4ba2-	servers on console #	0	neutral	rent servers on console	Qu
8	b264-4ba2-	servers on console #	0	neutral		

Operation Performed:

```
=VLOOKUP(B2,Part1,2,FALSE)
```

Challenges in Semantria

1. A study from the University of Pittsburgh shows that humans can only agree on whether or not a sentence has the correct sentiment, 80% of the time. So any natural language processing engine that can score around 80% is doing a great job with accuracy.
2. One of the biggest issues is that it has trouble understanding irony.
3. Other problems are when words have multiple definitions.
4. Duplicate Tweets (But some might be Retweets): As the API extracts theme and entity from the tweets, the tweets get repeated in the final file.
5. Data Duplication(Excel does not allow one to many mapping)

B	C	D
Source text	Document Sentiment	Theme
Bad experience. Very rude and ine	-0.605383694	bad experience
Bad experience. Very rude and ine	-0.605383694	unconsistent policies
Bad experience. Very rude and ine	-0.605383694	rude staff
Bad experience. Very rude and ine	-0.605383694	inefficient customer se

Comparison of API and Python Script

1. API has trained 7 TetraBytes of data from Wikipedia
2. Python script , we have used only 3500 words
3. API- gives more detailed analysis like entity and theme
4. Python script, gives overall document analysis
5. API, values are more precise

ANALYSIS AND RESULTS

We carried out the analysis on 15 days of twitter data and our analysis span over few metrics and how there is a correlation between those metrics. As we have adopted two methods for sentiment computation, we also compare the results from our own python code and Existing API. We answer following questions from our data sets and its analysis: -

1) How does the sentiments span over the geographical location? Which area exhibits positive sentiment over a course of period or negative and vice versa?

2) How does the polarity of sentiment effects the retweet counts for a particular user? How it depends on whether tweet was positive, negative or neutral?

3) Comparison for the retweet vs sentiment for API and python code over the course of few days?

4) How does the number of followers of the particular user effect the number of retweets?

5) What is the correlation between the sentiments and retweet irrespective of the number of followers, friends count and also with considering those factors too.

6) How social status of a user plays a role in the retweets count irrespective of the sentiment of the tweet? Example: Celebrity, known personality etc.

So let's see the answers to these questions one by one:-

1) SENTIMENTS VS GEOGRAPHICAL

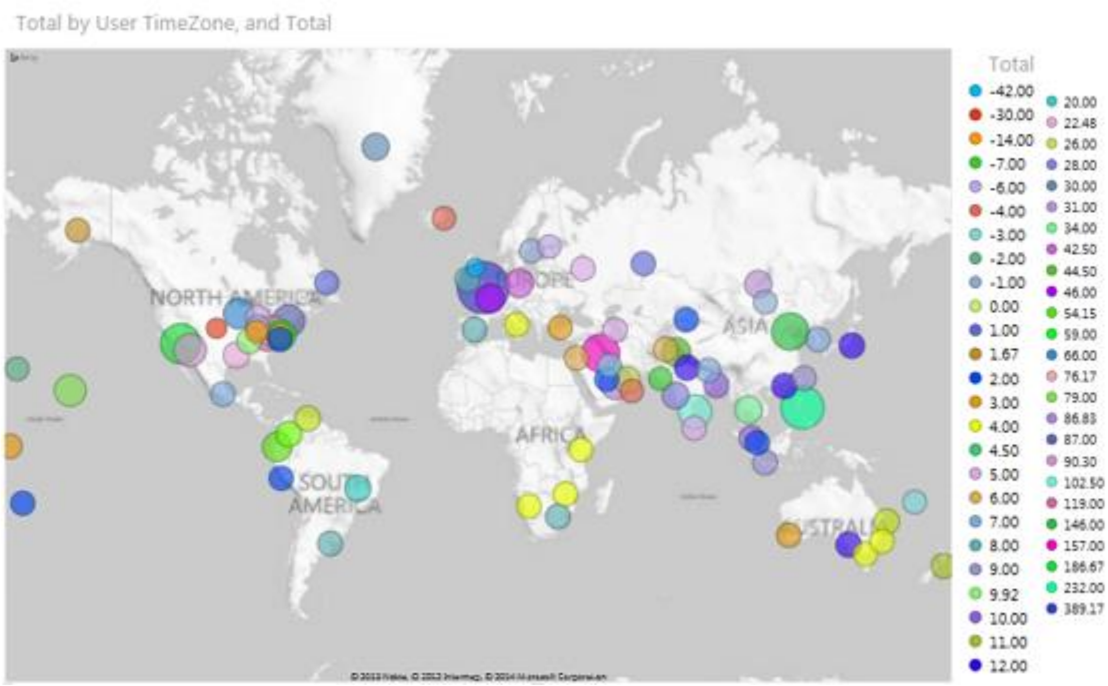
We made use of Power view feature of excel to create graphical representations. Power View is an interactive data exploration, visualization, and presentation experience that encourages intuitive ad-hoc reporting. So we make use of it in our analysis to represent world map, the distribution across locations, graph etc.

Let's see the results geographic location wise for sentiment score and time zone in consideration. First, we will see python code results. Excel snapshots are part of the file as file data is large so to depict we show a subset.

DAY 1

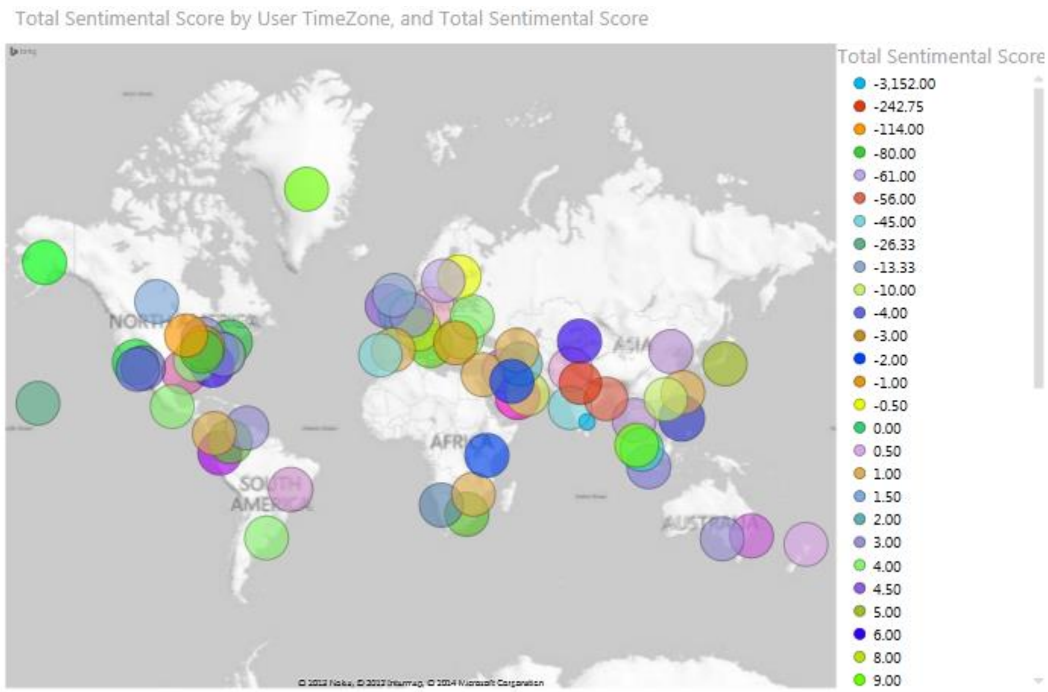
1	User TimeZone	Total Sentimental Score
2	"Abu Dhabi"	26
3	"Adelaide"	12
4	"Africa\/Accra"	2
5	"Alaska"	1.666666667
6	"Almaty"	2
7	"America\/Bahia_Banderas"	4.5
8	"America\/Chicago"	0
9	"America\/Los_Angeles"	5
10	"America\/New_York"	0
11	"Amsterdam"	87
12	"Arizona"	90.3
13	"Asia\/Riyadh"	0
14	"Athens"	119
15	"Atlantic Time (Canada)"	76.16666667
16	"Auckland"	0
17	"Azores"	0
18	"Baghdad"	157
19	"Baku"	5
20	"Bangkok"	34
21	"Beijing"	146
22	"Belgrade"	0
23	"Berlin"	42.5

This table represents the user time zone and the accumulated sentimental score for that location for a particular day (Day 1)

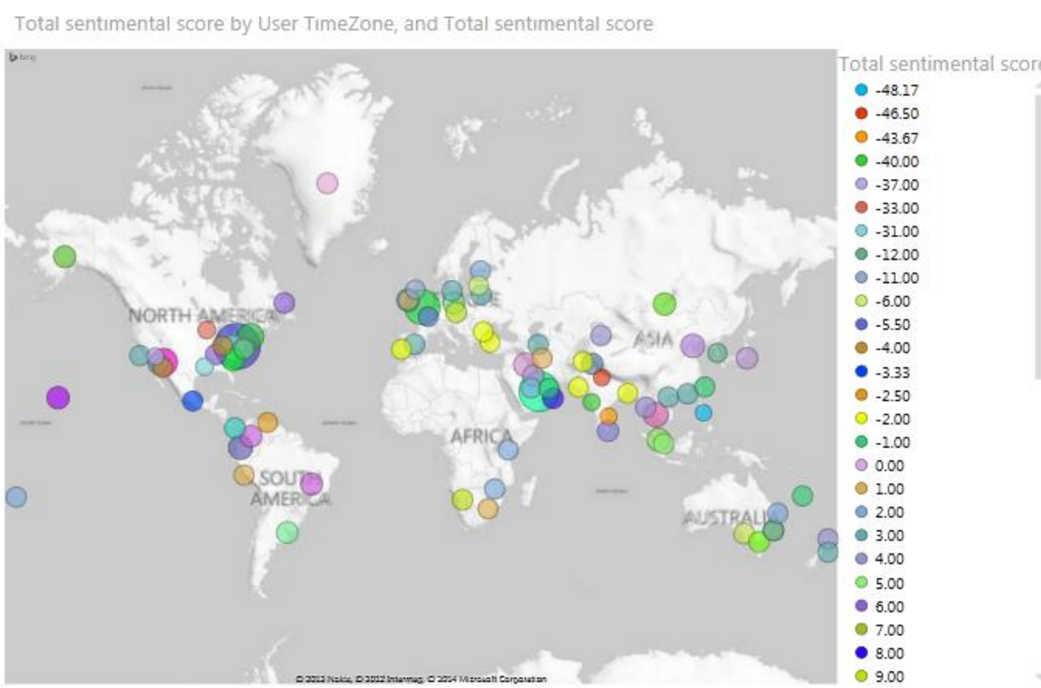


Here, we see the London has a bigger circle representing the tweets are much more positive in that location than any other. Smaller the circle, the negative sentiments of tweets in that location. Now if we observe it over few days, we noticed an observation. Let's see

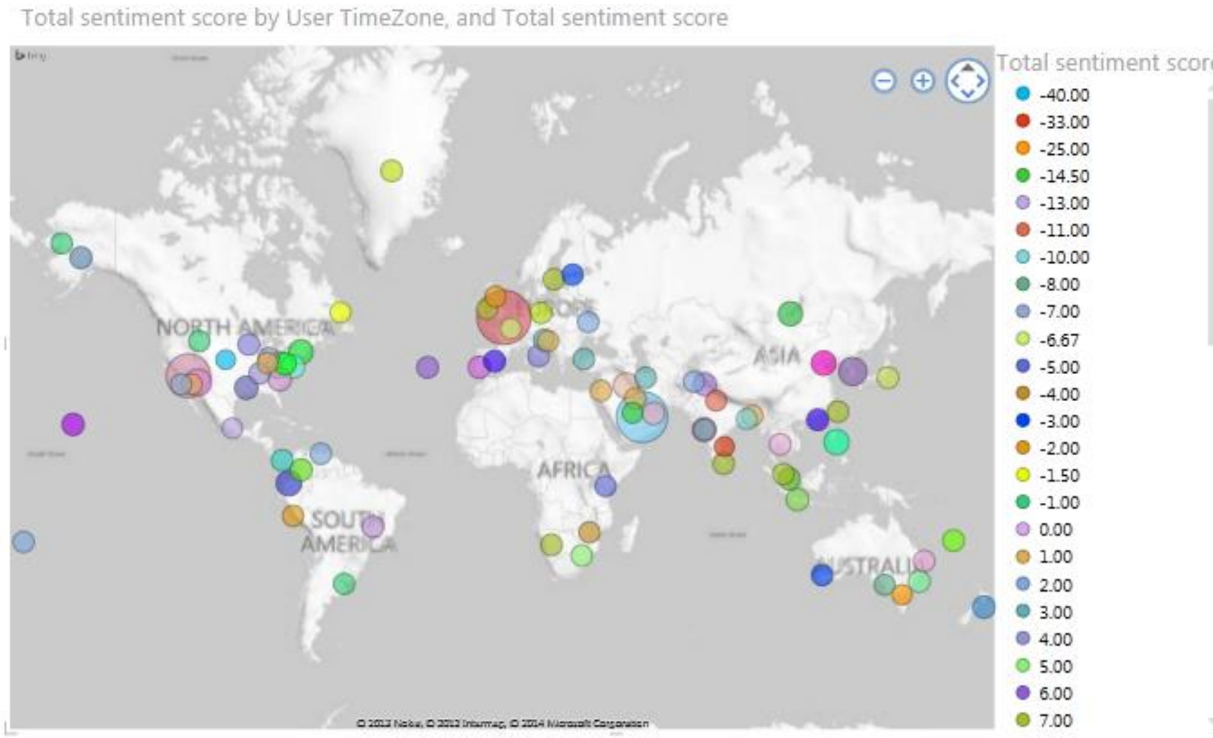
Day 2



Day 3



Day 4



We compared the results for 15 Days over the geographical location and found out that the results differ each time. We dig into the data more and analyzed the tweets for some days where there was some drastic change. We found out: -

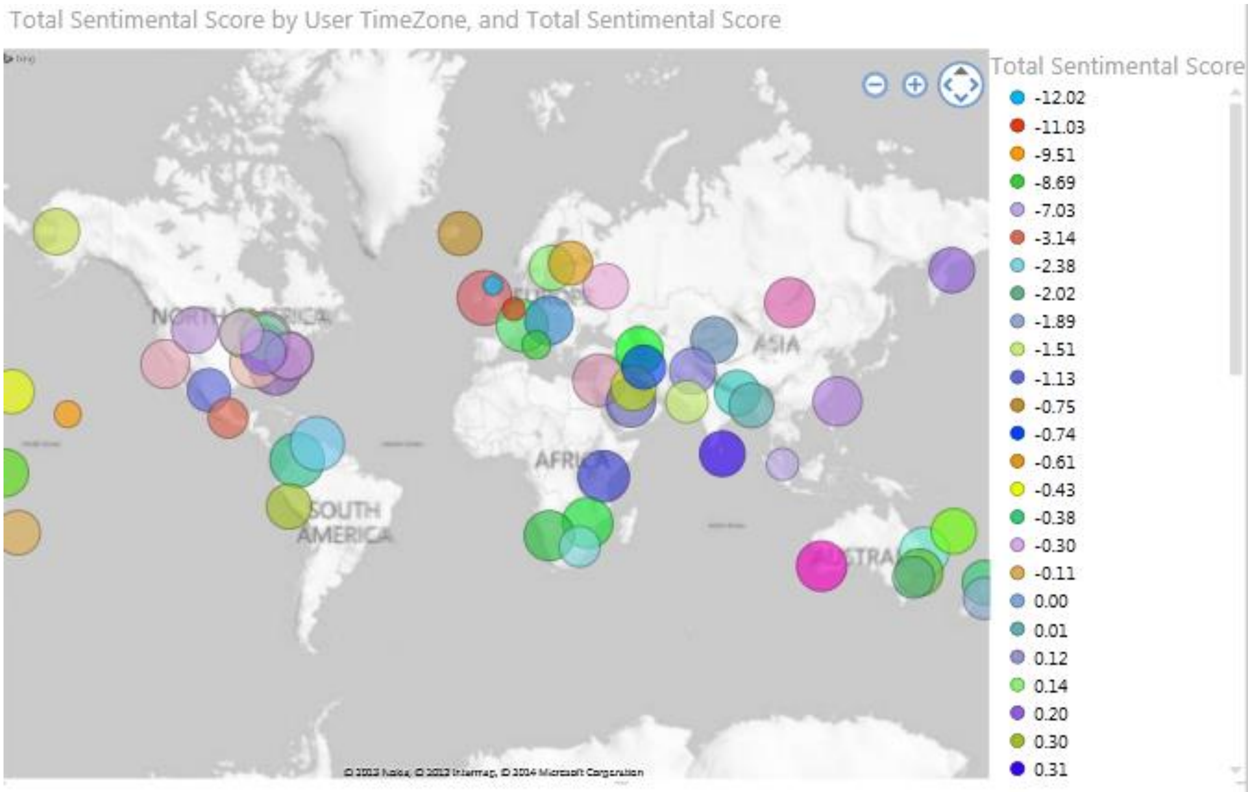
1) The sentiments score of the particular location depends **on what is the current state of that country or place. The state means, what type of news is flowing through that region, is there any event that has happened over the past few days etc.**

2) For instance, we noticed on Day 1, In India the cumulative score was of positive sentiment, but due to some Rape case on the subsequent day, the news flowed like a roar in nation and there were lot of negative comments on twitter. Hence, the pattern changed within a day.

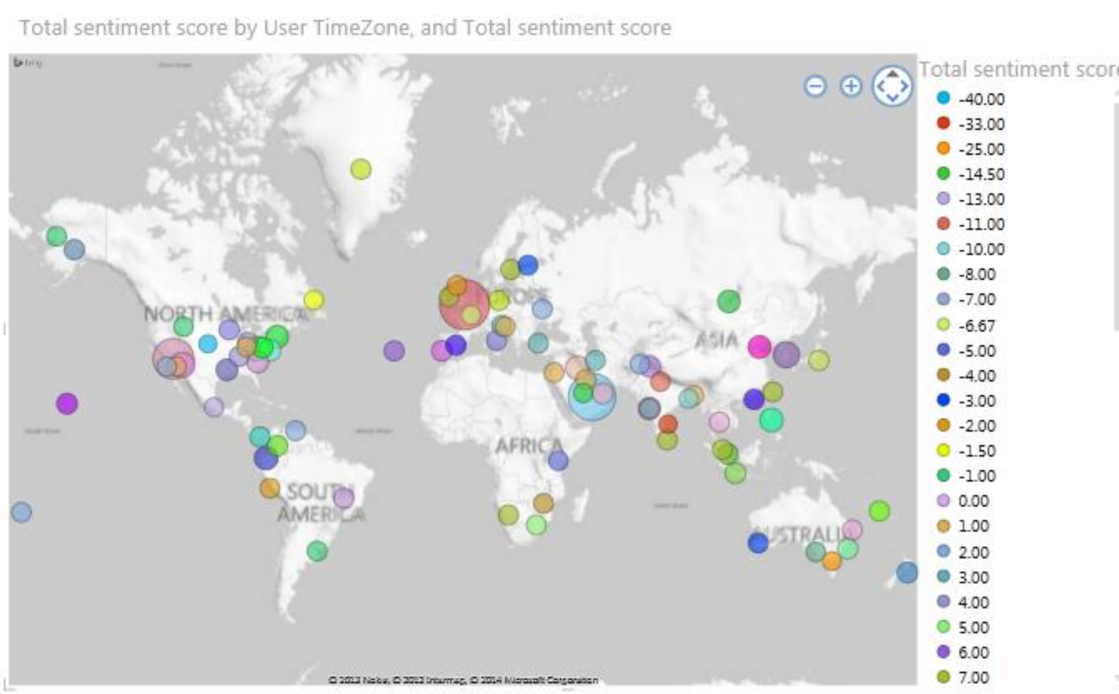
3) So we concluded that sentiments do span over different geographic locations and it entirely depends upon the current happenings around / in that region.

4) We also compared the data from our python script and the existing API and found out that it does depict almost the same results, just some values differ. But overall the depiction is logically evident.

COMPARISON BETWEEN TWO METHODS



EXISITNG API GEOGRAPHIC REPRESENTATION



OUR OWN PYTHON CODE

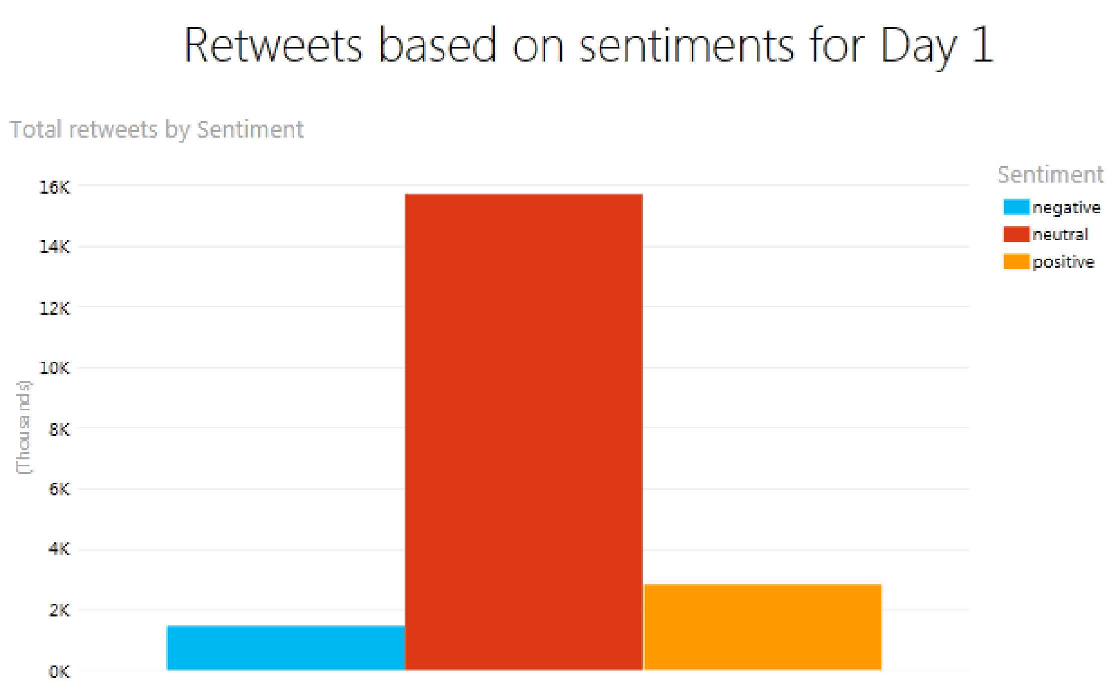
2) SENTIMENT POLARITY Vs RETWEET

Now once we computed the sentiment score and its respective polarity, we analyzed how does the polarity plays the role in retweets in the twitter media.

Let's see what we found out: -

DAY 1

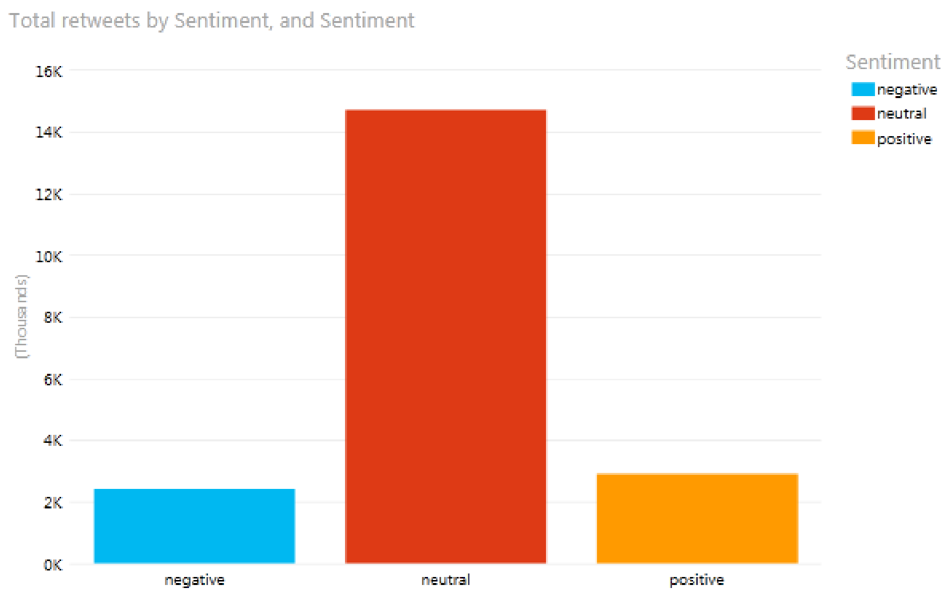
Sentiment	Total retweets
negative	1466
neutral	15712
positive	2821



DAY 2

Sentiment	Total retweets
negative	2415
neutral	14692
positive	2893

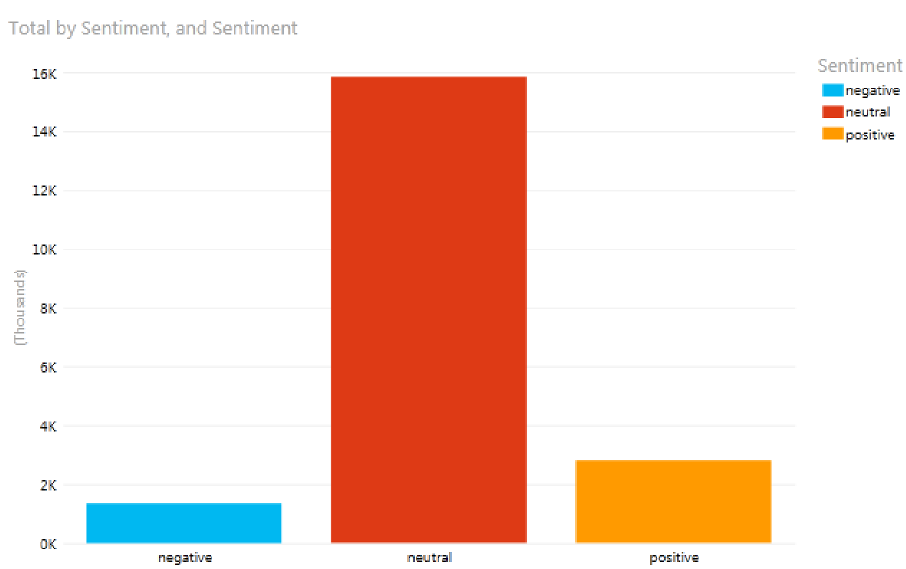
Retweets based on sentiments for Day 2



DAY 3

Sentiment	Total
negative	1344
neutral	15844
positive	2812

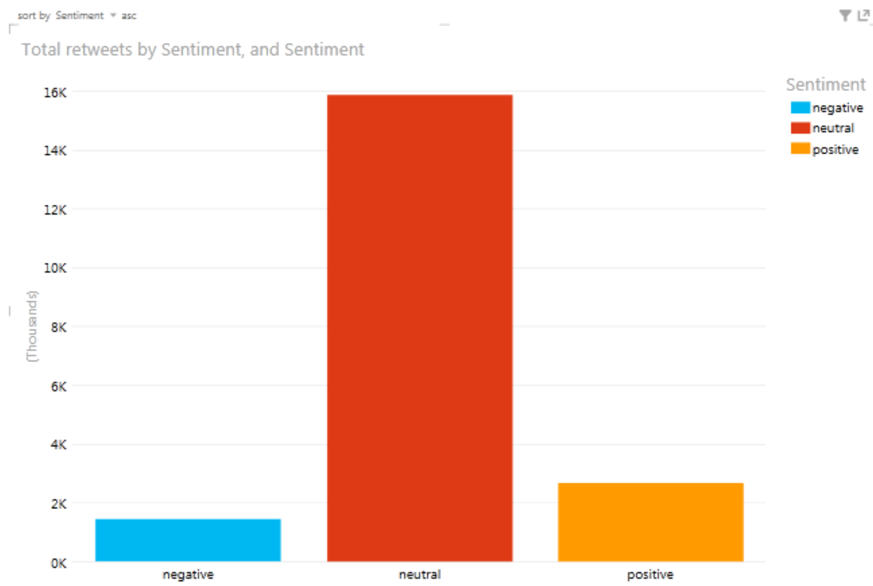
Retweets based on sentiments for Day 3



DAY 4

Sentiment	Total retweets
negative	1450
neutral	15871
positive	2679

Retweets based on sentiments for Day 4



We compared the results for 15 Days over the sentiment score and the retweet count and found out that **the results are consistent over the time**. Here we show only for first four days to remove the same redundant figures. We found out:

1) The polarity of sentiment of the particular tweet do play a major role in retweeting. As we can see**, more the tweet is towards positive/neutral it is retweeted more and more.**

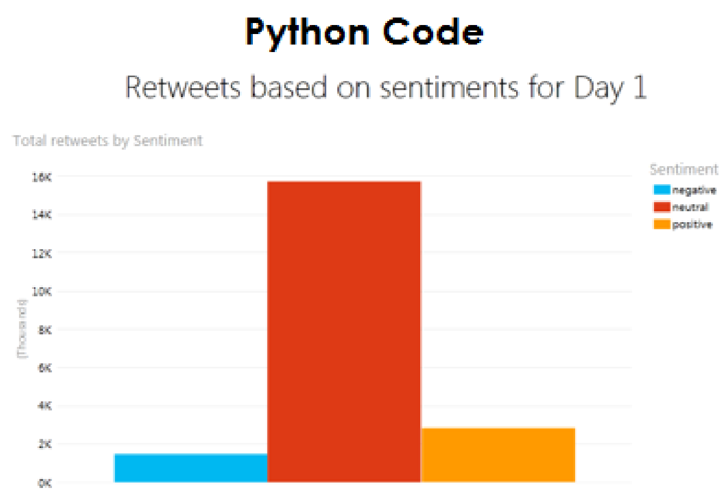
2) For instance, **if the tweets have positive nature it is more likely to be re-tweeted by the followers/ friends than the negative** and that is shown by our study.

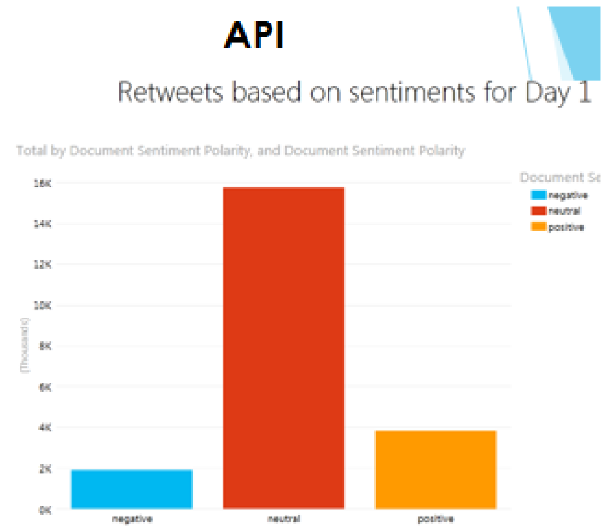
3) So we concluded that polarity plays a role in retweet. We also observe negative tweets are also retweeted, but in small numbers as compared to positive/neutral. Negative are retweeted as they are some political comment or comment due to

some negative event happening at that moment and has created a spurge among people. Example: Malaysian airlines incident, Rape cases etc.

4) We also compared the data from our python script and the existing API and found out that it does depict almost the same results, just some values differ. But overall the depiction is logically evident.

COMPARISON BETWEEN TWO METHODS





We can see as evident from above, the results are almost the same with our python code and API. It's just API has more robust values.

3) NUMBER OF FOLLOWERS Vs RETWEET

We also found out a correlation between how the number of followers effect the retweet count. If we see our analysis on the subset of large amount of data which spanned over 15 days, we find a logical reference of **more the number of followers, more the retweet counts.**

Let's see the data:

DAY 1

Followers Count	Friends Count	Retweet Count
1553846	289	1008
29	0	1
16989	298	10988
772	217	44
70	50	7
23423885	170	673

More followers more retweet.

DAY 2

Followers Count	Friends Count	Retweet Count
50138454	124564	1084
17051	298	11050
7	182	1
6	66	1
39515	375	79
1089	1912	79
831	1152	79
23980	3534	79

Here we see if number of followers is 7, 6 consequently the retweet count is less. It is shown in most of the days and for most of the users.

DAY 3

Followers Count	Friends Count	User TimeZone	Retweet Coun
1554110	289	"Eastern Time (US & Canada)"	1012
50170314	124652	"Eastern Time (US & Canada)"	1099
243032	9222	"Alaska"	393
17527700	4948	"London"	78519
15	60	null	1
8	26	null	1
8	26	null	1
260	234	null	101
31	0	"Bangkok"	0
3	17	null	0
17	91	null	0
5	55	"Central Time (US & Canada)"	0

We can conclude by looking at this data, **number of retweets is proportional to number of followers.** Again, there are some cases where it do not follow this, but it is for minimal amount of data. So we can conclude the general trend on twitter.

This is what we observe in daily life, if someone has large number of followers, it is logical he would tend to have more number of retweets than the person who has comparatively less number of followers.

DAY 4

1554126	289	"Eastern Time (US & Canada)"	1026
23453061	173	"Central Time (US & Canada)"	676
243032	9222	"Alaska"	402
17527700	4948	"London"	78565
15	60	null	2
8	26	null	1
8	26	null	2
260	234	null	106
31	0	"Bangkok"	0
3	17	null	0
17	91	null	0

We compared the results for 15 Days for number of followers and retweets and found out that the number of followers do increase over the time which increases the number of retweets for that particular user. We found out: -

Retweet count do depend on the number of followers and in full dependence.

4) NUMBER OF FOLLOWERS Vs RETWEET VS SENTIMENTS

We also observed and saw what happens when we consider the sentiment into the picture of retweet and the followers. Let's see what we observe from the data.

DAY 1

Sentiment scor	Sentimen	Followers Cour	Friends Coun	User TimeZon	Retweet Coun
-1	negative	106491	66393	"Bangkok"	0
-3	negative	213	207	"Central Time (US	0
-2	negative	298	358	"Eastern Time (US	0
-2	negative	21	26	"Central Time (US	0
3	positive	94811	5	"Pacific Time (US	185
4	positive	17495228	4959	"London"	78376
3	positive	21	26	"Central Time (US	0
3	positive	117	15	"Beijing"	0
4	positive	8238	2139	"Eastern Time (US	14
0	neutral	2185	2033	"Quito"	0
0	neutral	949	1227	"Caracas"	0
0	neutral	1422	18	"London"	10
0	neutral	2784651	12559	"London"	65

Here we observe, **if we have large number of followers and still we tweet something negative, retweets are very less. So we can say sentiments play a big role in the social network in daily life.**

DAY 2

Sentiment	Sentiment	User Name	Followers Count	Friends Count	User TimeZone	Retweet Co	
0	neutral	Justin Bieber	50138454	124564	"Eastern Time (US & Canada)"	1084	
5	positive	joe jonas	6935510	606	"Mountain Time (US & Canada)"	1359	
6	positive	lili_london_	2188	748	"London"	297	
6	positive	lili_london_	2188	748	"London"	106	
-1	negative	Kathy Crowley	1833	342	"Jakarta"	0	
-1	negative	FREDERICA	532	1145	"Quito"	0	
-2.5	negative	Laila Jazayeri	601	1138	"Casablanca"	0	
-2.5	negative	Laila Jazayeri	601	1138	"Casablanca"	0	
-3	negative	Peyton Flemi	477	427	"Atlantic Time (Canada)"	0	
-4	negative	Alok Gupta	1555	54	"Chennai"	508	"
-2	negative	NCRIWomen's	1072	162	"Paris"	102	"

DAY 3

Sentiment scor	Sentimen	Followers Cour	Friends Coun	User TimeZone	Retweet Coun
0	neutral	250712	229040	"Central Time (US & Canada)"	7
0	neutral	6311	4992	"Eastern Time (US & Canada)"	7
-7	negative	1083	98	"Abu Dhabi"	7
0	neutral	4052	3335	"India"	7
-1	negative	13811	12325	"Arizona"	7
0	neutral	417	699	"India"	7
-1	negative	13811	12325	"Arizona"	7
4	positive	17502949	4957	"London"	78435
0	neutral	1553846	289	"Eastern Time (US & Canada)"	1008
0	neutral	17009	298	"Central Time (US & Canada)"	11000
3	positive	2823012	10050	"Pacific Time (US & Canada)"	2845
3	positive	1309	677	"Eastern Time (US & Canada)"	11
2	positive	220	1291	"Central Time (US & Canada)"	27

Day 4

Sentiment score	Sentiment	User Name	Followers Count	Friends Count	User TimeZone	Retweet Count
1.5	positive	Barb	6343	29	"Central Time (US & Canada)"	57
4	positive	Niall Horan	17523975	4948	"London"	78503
6	positive	Jason Cross	59	109	"Pacific Time (US & Canada)"	3
9	positive	mai	20668	21225	"Athens"	32
1.5	positive	Barb	6341	29	"Central Time (US & Canada)"	58
2	positive	Liow Tiong L	35732	77	"Kuala Lumpur"	12
0	neutral	Samraat Dh	260	234	null	101
0	neutral	Samraat Dh	260	234	null	103
0	neutral	freddy casa:	362	635	"Pacific Time (US & Canada)"	0
0	neutral	arun shourie	5274	147	null	71
0	neutral	Yahoo Makt	25522	68	"Abu Dhabi"	8
-5	negative	DiCkiN_U_N	8767	6177	"Central Time (US & Canada)"	Kaw
-7	negative	\u091a\u092	841	8	"Chennai"	0
-2	negative	jacob rieden	33	72	null	0
-2	negative	nshaqirarosl	507	284	"Kuala Lumpur"	0
-7	negative	pratibhakasl	714	575	null	3
-2	negative	ashish	155	194	"New Delhi"	0
-2	negative	\u96fb\u821	2	2	"Irkutsk"	0
-5	negative	k - kizzle	85	173	null	0
-7	negative	Dharmendra	32	54	"Chennai"	0
-5	negative	melissa	2387	377	"Jakarta"	0

We compared the results for 15 Days considering the sentiments, retweet and the number of followers all three at one time and we found out: -

1) **The sentiments score of the tweet is the major factor in the retweet behavior than the number of followers. As we see, if the number of followers for a particular user is more, still it has less number of retweets in the case where the sentiment computed is negative for a tweet.**

2) For instance, we noticed on Day 1, we see

Sentiment scor	Sentimen	Followers Cour	Friends Coun	User TimeZon	Retweet Coun
-1	negative	106491	66393	"Bangkok"	0
-3	negative	213	207	"Central Time (US	0
-2	negative	298	358	"Eastern Time (US	0
-2	negative	21	26	"Central Time (US	0

The sentiments are negative, followers are large in number but retweet count is 0. So sentiments play a bigger role in retweet than number of followers if we have to do comparison between two.

5) SOCIAL STATUS VS RETWEETS (CELEBRITY, KNOWN PERSONALITY)

We observed an interesting fact during the analysis, Social status and the celebrity status of a user also plays a role in a retweet. When we consider that, the sentiment do not play much of role.

Let's see what we observed: -

Sentiment	Sentiment	User Name	Followers Count	Friends Count	User TimeZone	Retweet Co
0	neutral	Justin Bieber	50138454	124564	"Eastern Time (US & Canada)"	1084
5	positive	joe jonas	6935510	606	"Mountain Time (US & Canada)"	1359

As we can see the two famous celebrity Justin Bieber and Joe Jonas do have large number of followers due to their status and famous personality, which let them have a large number of retweets on a neutral tweet.

For instance, Justin Bieber just posted "I am in LA". This was retweeted by more than 1000 people and similarly Joe posted an information regarding the concert he was entitled to go, again number of retweets rises. Similar observation were observed for Indian Television star and celebrity.

So we conclude that social status do play a role in getting more retweet due to the popularity and the belief people have in their stars.

Conclusion

We have successfully scrapped the twitter data for 15 Days, totaling to around 5, 00, 000 tweets in total to perform the global sentiment analysis on it. We have implemented our own python code and also made use of the existing API to compute the sentiment score and do the further analysis. We have compared our own result with the results from the API and drawn the necessary conclusion. After careful observations and graphical representation we conclude the findings:

1) The sentiments score of the particular location depends on what is the current state of that country or place. The state means, **what type of news is flowing through that region, is there any event that has happened over the past few days** etc.

2) The polarity of sentiment of the particular tweet do play a major role in retweeting. **More the tweet is towards positive/neutral it is retweeted more and more.** For instance, **if the tweets have positive nature it is more likely to be re-tweeted by the followers/ friends than the negative** and that is shown by our study.

3) Also, **Number of retweets is** proportional **to number of followers. More the number of followers, more the number of retweets.**

4) **The sentiments score of the tweet is the major factor in the retweet behavior than the number of followers**. As we see, if the number of followers for a particular user is more, still it has less number of retweets in the case where the sentiment computed is negative for a tweet.

5) Also, **social status do play a role in getting more retweet due to the popularity and the belief people have in their stars.**

Resources and Citation:-

- http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html
- Bing Liu. "Sentiment Analysis and Subjectivity." Invited Chapter for the *Handbook of Natural Language Processing*, Second Edition. March, 2010.
- How to get Twitter Data Set: - https://class.coursera.org/datasci-001/lecture/55
- http://www.slideshare.net/mukherjeesubhabrata/twisent-a-multistage-system-for-analyzing-sentiment-in-twitter.
- Lei Zhang and Bing Liu. "Identifying Noun Product Features that Imply Opinions." *ACL-2011* (short paper), Portland, Oregon, USA, June 19-24, 2011.
- Minqing Hu and Bing Liu. "Mining Opinion Features in Customer Reviews." Proceedings of Nineteenth National Conference on Artificial Intelligence (AAAI-2004), San Jose, USA, July 2004
- https://semantria.com/features/sentiment-analysis
- http://scriptogr.am/richie/post/using-viralheats-sentiment-analysis-api-through-excel-2013
- http://www.daniweb.com/software-development/python/threads/141128/read-multiple-files
- http://blog.gopivotal.com/pivotal/products/analyzing-raw-twitter-data-using-hawq-and-pxf