Text mining & Sentimental Analysis on Twitter Data (Election 2016 USA)

Dept. of Computer Science

Central Michigan University

By

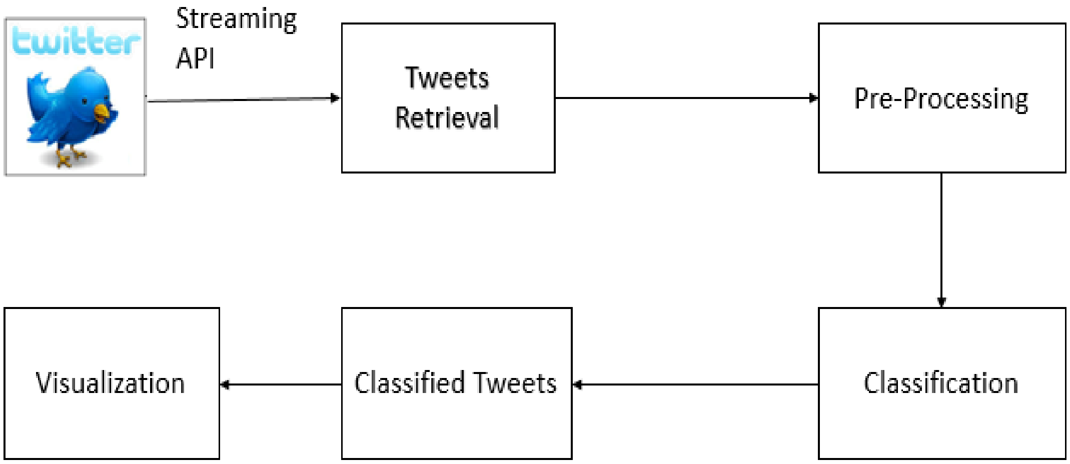
Sandeep Reddy Rakasi

Abstract:

The aim of this project report is to mine the tweets from twitter and find the sentiment in them towards the Presidential Election candidates 2016. It describes the systematic procedure followed to extract useful and important information from twitter. The report consists of several scripts that are useful to extract tweets related to specific tags. The twitter API used is so simple that a person with mere amount of programing background and knowledge will be able to extract and analyze the data.

We already know that twitter is the most used microblogging website. Tweeters use twitter to express their emotion towards a trending or a topic. The aim here is to collect tweets relevant to presidential elections 2016 and find the polarity of tweets to analyze the sentiment in them. Tweets are collected based on specific hashtags related to these presidential candidates.

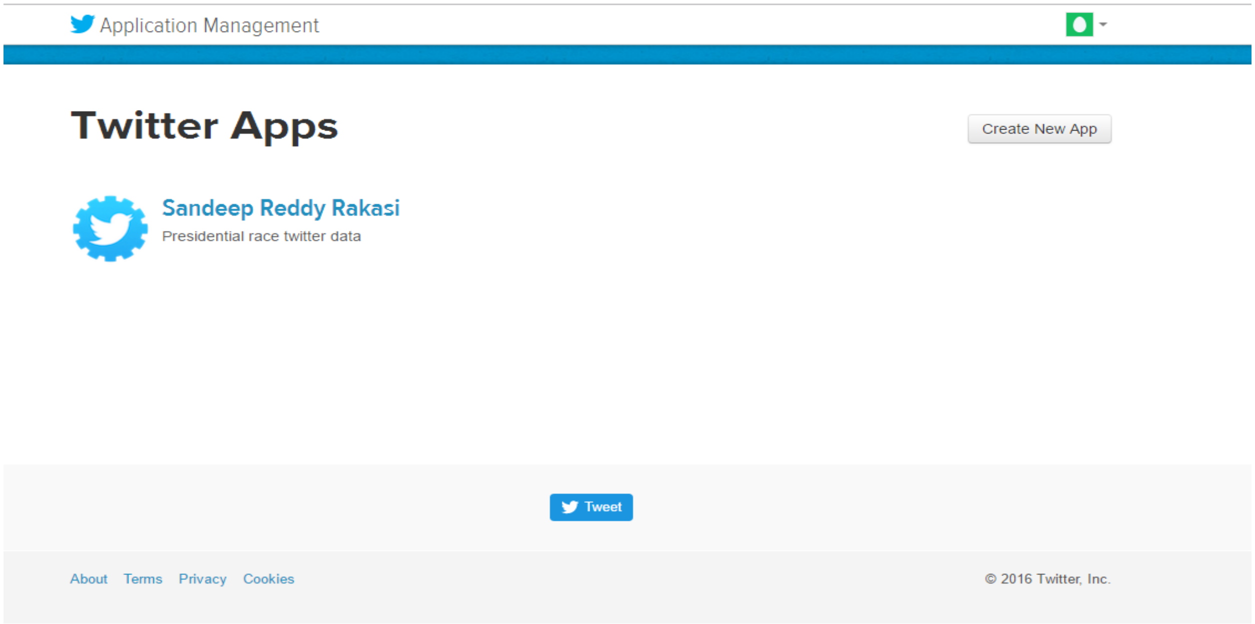
Steps involved in twitter sentiment analysis:



Collection of Tweets:

Create an Application:

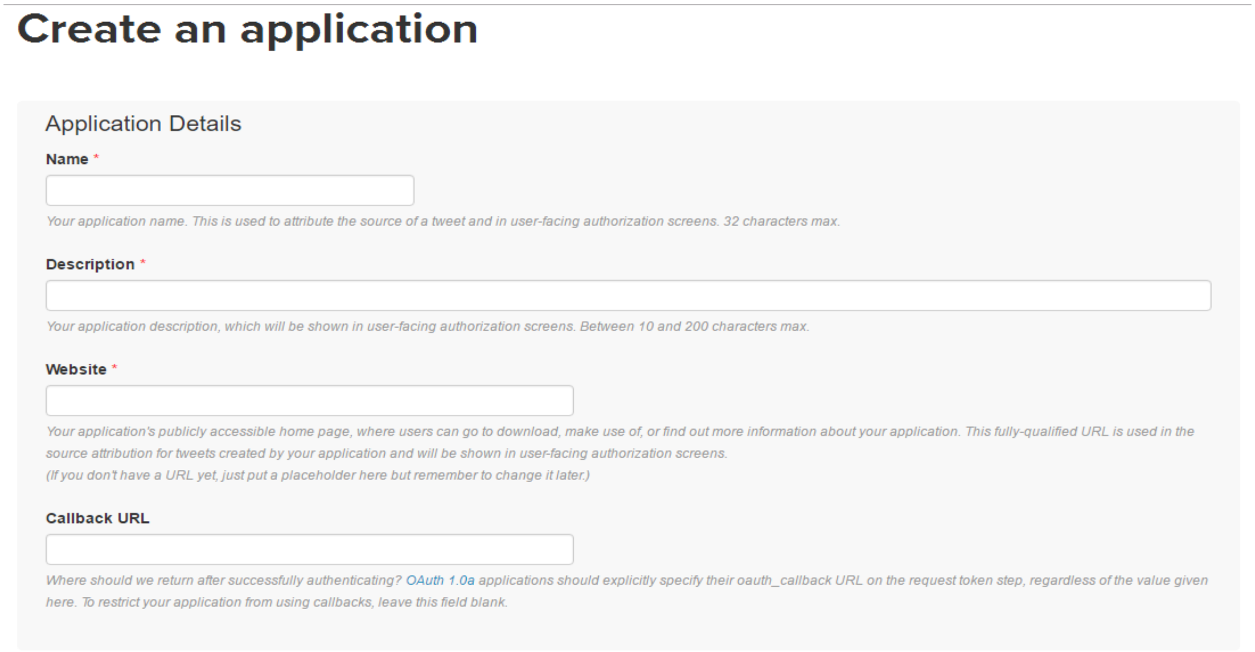
To keep track of the Tweets programmatically, we need to create an Application that interacts with the Twitter API. The first Step in this process is to Create an Application. You can do that by using this http://apps.twitter.com , and logging into you Twitter Account (If you don't have one you can Create one) and Create a new Application.



Twitter Apps

Sandeep Reddy Rakasi
Presidential race twitter data

Now you can click on the Create New App button in the above screen to Create a new Application by filling up the Application details such as Name, Description of the Application, Website, and URL. Fill up the details by giving a proper Website from which others can see and download the work that you have done. Thus, an App can be created through which you can make use of Twitter API to collect Streaming data form Twitter.



Create an application

Application Details

Name *

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

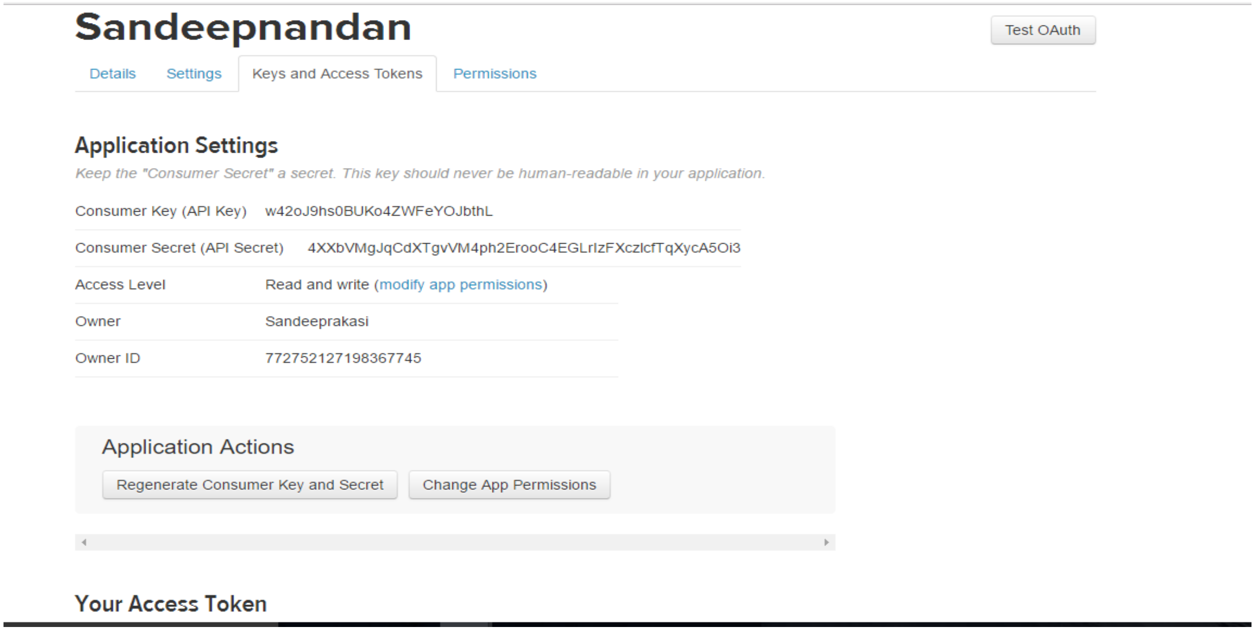
Website *

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

After this Creation of Application, generate keys and Access Tokens which must be kept *Secret*.



Sandeepnandan

Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key)	w42oJ9hs0BUKo4ZWFeYOJbthL
Consumer Secret (API Secret)	4XXbVMgJqCdXTgvVM4ph2ErooC4EGLrIzFXczlcfTqXycA5Oi3
Access Level	Read and write (modify app permissions)
Owner	Sandeeprakasi
Owner ID	772752127198367745

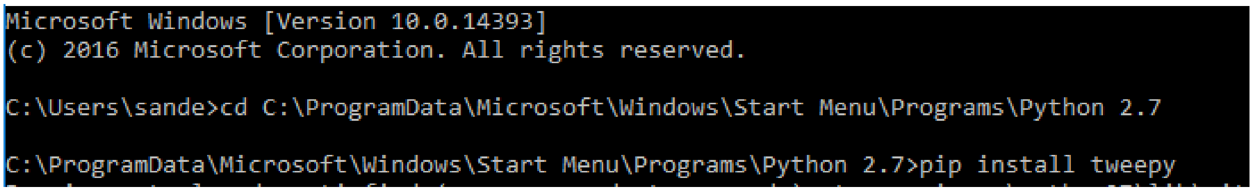
Application Actions

Your Access Token

Similarly, you need to Create *Access key* and *Access Secret* which must be kept Secret like Consumer key and Secret strings as they provide access to twitter on behalf of your account by making use of the API.

Collection of tweets:

We can make use REST API's that are provided by Twitter to interact with their Service. Here I have used Tweepy which is an open-source, which enables python to interact with Twitter and Stream Tweets by making use of their API's. This ***Tweepy*** can be installed by making use of ***pip*** (a package management system to manage and install packages). Tweepy can be installed by using the following command:

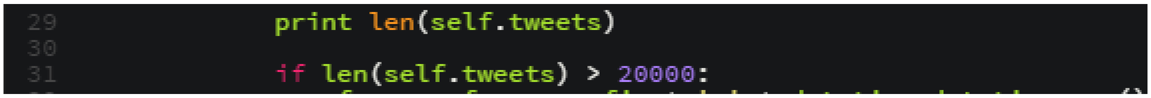


```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\sande>cd C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Python 2.7

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Python 2.7>pip install tweepy
```

Thus, form the Above Tweepy can be installed.

Now we need to make use of Tweepy to gather the Tweets which enables python to interact with Twitter API. The below Python Script fetches the Tweets. Here we use consumer token and key, Access token and Access secret which collects the Tweets from Twitter on behalf of your Account. Here we make use of StreamListener so that we can collect Stream of Tweets related to the given hashtags.

In this way, we can collect Tweets and store them in the form of JSON format which is pretty easy further to transform it into any other data format as many of the newer technologies provide an option of importing the data in bulk (such as NOSQL). Iteration through different types of objects can be achieved through Tweepy by making use of convenient cursor interface.



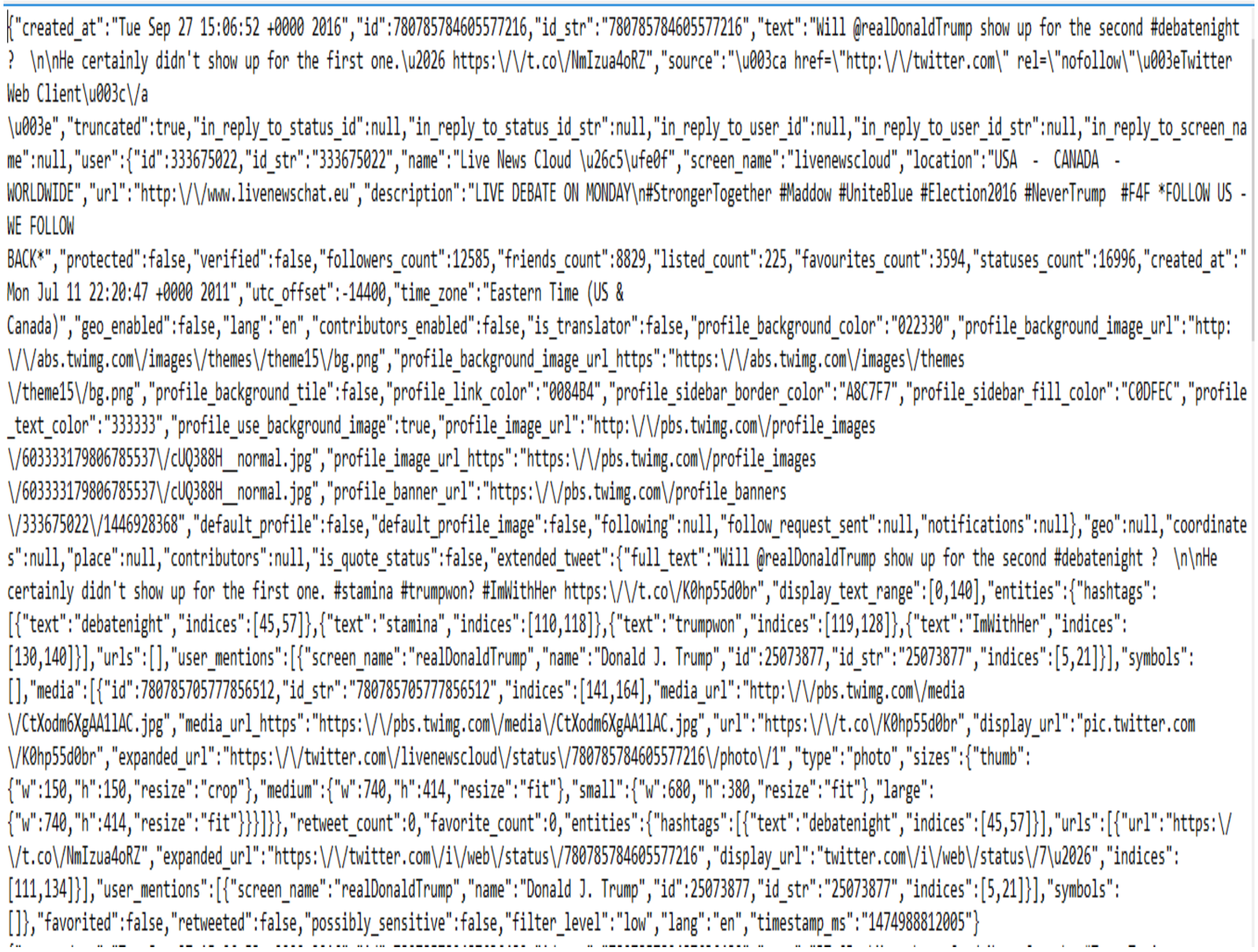
```
                print len(self.tweets)

                if len(self.tweets) > 20000:
```

In the above code, I gave the limit of number of tweets read at once as 20000, which is not fixed. Of course we can access more number of tweets my specifying the limit but officially the Twitter API connection is terminated or released every fifteen minutes.

```
import datetime
import tweepy
from tweepy import OAuthHandler
from tweepy import Stream
from tweepy.streaming import StreamListener

consumer_key = 'sbYay36q0Qjh4Aq402pJ1GUbN'
consumer_secret = 'HShlWoDdZ5MZKQNSPDRhJqNivQ8BV7L35Xl4vCMdBnmM1UpQ6x'
access_token = '772752127198367745-yf8F706LrYDlzgkKLJ7abTw03NNDl1B'
access_secret = 'SeAR88pHouGvMNjbTCnqBCO2FltS1YJHZaARkX1dZ3tCp'



fname_prefix = 'Elections_2016'

auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)



class MyListener(StreamListener):

    def __init__(self):
        self.tweets = []



    def on_data(self, data):
        try:
            self.tweets.append(data);

            print len(self.tweets)
```

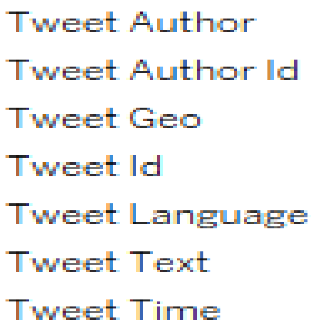
```
# to write Every 20000 tweets onto a new file
            if len(self.tweets) > 20000:
                fname = fname_prefix + '-' + ".json"
                with open(fname, 'a') as f:
                    for tweet in self.tweets:


                        f.write("%s" % tweet)


                self.tweets = []



            #with open('python.json', 'a') as f:
            #    f.write(data)
            return True
        except BaseException as e:
            print("Error on_data: %s" % str(e))
        return True

    def on_error(self, status):
        print(status)
        return True

twitter_stream = Stream(auth, MyListener())
# The track keeps track of the tweets from those hashtags
twitter_stream.filter(track=[' '])
```

After the Tweets are collected we need to preprocess them, as we might not need to work on the whole data. The original data format is

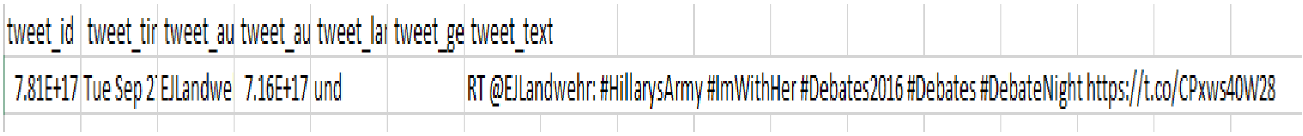


```
{"created_at":"Tue Sep 27 15:06:52 +0000 2016","id":780785784605577216,"id_str":"780785784605577216","text":"Will @realDonaldTrump show up for the second #debatenight
?  \n\nHe certainly didn't show up for the first one.\u2026 https:\/\/t.co\/NmIzua4oRZ","source":"\u003ca href=\"http:\/\/twitter.com\" rel=\"nofollow\"\u003eTwitter
Web Client\u003c\/a
\u003e","truncated":true,"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_na
me":null,"user":{"id":333675022,"id_str":"333675022","name":"Live News Cloud \u26c5\ufe0f","screen_name":"livenewscloud","location":"USA  -  CANADA  -
WORLDWIDE","url":"http:\/\/www.livenewschat.eu","description":"LIVE DEBATE ON MONDAY\n#StrongerTogether #Maddow #UniteBlue #Election2016 #NeverTrump  #F4F *FOLLOW US -
WE FOLLOW
BACK*","protected":false,"verified":false,"followers_count":12585,"friends_count":8829,"listed_count":225,"favourites_count":3594,"statuses_count":16996,"created_at":"
Mon Jul 11 22:20:47 +0000 2011","utc_offset":-14400,"time_zone":"Eastern Time (US &
Canada)","geo_enabled":false,"lang":"en","contributors_enabled":false,"is_translator":false,"profile_background_color":"022330","profile_background_image_url":"http:
\/\/abs.twimg.com\/images\/themes\/theme15\/bg.png","profile_background_image_url_https":"https:\/\/abs.twimg.com\/images\/themes
\/theme15\/bg.png","profile_background_tile":false,"profile_link_color":"0084B4","profile_sidebar_border_color":"A8C7F7","profile_sidebar_fill_color":"C0DFEC","profile
_text_color":"333333","profile_use_background_image":true,"profile_image_url":"http:\/\/pbs.twimg.com\/profile_images
\/603333179806785537\/cUQ388H__normal.jpg","profile_image_url_https":"https:\/\/pbs.twimg.com\/profile_images
\/603333179806785537\/cUQ388H__normal.jpg","profile_banner_url":"https:\/\/pbs.twimg.com\/profile_banners
\/333675022\/1446928368","default_profile":false,"default_profile_image":false,"following":null,"follow_request_sent":null,"notifications":null},"geo":null,"coordinate
s":null,"place":null,"contributors":null,"is_quote_status":false,"extended_tweet":{"full_text":"Will @realDonaldTrump show up for the second #debatenight ?  \n\nHe
certainly didn't show up for the first one. #stamina #trumpwon? #ImWithHer https:\/\/t.co\/K0hp55d0br","display_text_range":[0,140],"entities":{"hashtags":
[{"text":"debatenight","indices":[45,57]},{"text":"stamina","indices":[110,118]},{"text":"trumpwon","indices":[119,128]},{"text":"ImWithHer","indices":
[130,140]}],"urls":[],"user_mentions":[{"screen_name":"realDonaldTrump","name":"Donald J. Trump","id":25073877,"id_str":"25073877","indices":[5,21]}],"symbols":
[],"media":[{"id":780785705777856512,"id_str":"780785705777856512","indices":[141,164],"media_url":"http:\/\/pbs.twimg.com\/media
\/CtXodm6XgAA1lAC.jpg","media_url_https":"https:\/\/pbs.twimg.com\/media\/CtXodm6XgAA1lAC.jpg","url":"https:\/\/t.co\/K0hp55d0br","display_url":"pic.twitter.com
\/K0hp55d0br","expanded_url":"https:\/\/twitter.com\/livenewscloud\/status\/780785784605577216\/photo\/1","type":"photo","sizes":{"thumb":
{"w":150,"h":150,"resize":"crop"},"medium":{"w":740,"h":414,"resize":"fit"},"small":{"w":680,"h":380,"resize":"fit"},"large":
{"w":740,"h":414,"resize":"fit"}}}]}},"retweet_count":0,"favorite_count":0,"entities":{"hashtags":[{"text":"debatenight","indices":[45,57]}],"urls":[{"url":"https:\/
\/t.co\/NmIzua4oRZ","expanded_url":"https:\/\/twitter.com\/i\/web\/status\/780785784605577216","display_url":"twitter.com\/i\/web\/status\/7\u2026","indices":
[111,134]}],"user_mentions":[{"screen_name":"realDonaldTrump","name":"Donald J. Trump","id":25073877,"id_str":"25073877","indices":[5,21]}],"symbols":
[]},"favorited":false,"retweeted":false,"possibly_sensitive":false,"filter_level":"low","lang":"en","timestamp_ms":"1474988812005"}
```

As we can see that, the above format is a bit ambiguous. We preprocess these tweets and only collect the useful field for the analysis and visualization. As the main aim is to analyze the change trend in Tweeters towards the Presidential candidates during a period of time.so, for this I am mainly concentrating on the fields,



Tweet Author
Tweet Author Id
Tweet Geo
Tweet Id
Tweet Language
Tweet Text
Tweet Time



tweet_id	tweet_tir	tweet_au	tweet_au	tweet_lai	tweet_ge	tweet_text
7.81E+17	Tue Sep 2'	EJLandwe	7.16E+17	und		RT @EJLandwehr: #HillarysArmy #ImWithHer #Debates2016 #Debates #DebateNight https://t.co/CPxws40W28

The above is the format of the filtered Tweets. Now, the tweet_text field needs to be filtered as it has many special symbols, hyperlinks numbers which might not be very much useful for the sentiment analysis. For this preprocessing few steps are to be followed. The Script below shows filters the required fields from all of the fields and it also checks the schema for null values and some other formats.

```
import json
import sys
from csv import writer


input=open('in_file', 'r')
output=open('out_file', 'w')
  print >> out_file, 'tweet_id, tweet_time, tweet_author, tweet_author_id, tweet_language, tweet_geo, tweet_text'
  csv = writer(out_file)
  tweet_count = 0
  tweet_filtered = 0


  for line in input:
    tweet_count += 1
      tweet = json.loads(line)


      try:
    # Pull out various data from the tweets
      row = (
            tweet['id'],                    # tweet_id
      tweet['created_at'],           # tweet_time
            tweet['user']['screen_name'],   # tweet_author
            tweet['user']['id_str'],        # tweet_authod_id
            tweet['lang'],                 # tweet_language
            tweet['geo'],                  # tweet_geo
            tweet['text']                 # tweet_text
      )
      except KeyError, e:
            id != "int"
            created_at = "null"
            lang = "null"
            lang != "en"
            text = "null"
    values = [(value.encode('utf8') if hasattr(value, 'encode') else value) for value in row
      tweet_filtered += 1


# print the name of the file and number of tweets imported
print "# Tweets Imported:", tweet_count
print "File Exported:", output.write(values)
print "# Tweets Exported:", tweet_filtered
```

From the above Script, we input a file consisting of Tweets with all the fields. Then we separate specific fields as mentioned above from the file into another file and save it on to the system. The processed Tweets consists of tweet[id], tweet[created_at], tweet[user][screen_name], tweet[user][id_str], tweet[lang], tweet[geo], tweet[text]. The tweet_text consists of the Tweet with some special characters, symbols, punctuations, and hyperlinks which are to be removed and a polarity is to be generated for the Tweet.

As the retrieved tweets consists of many special symbols and characters, hyperlinks, images, and other kinds of multimedia data the tweets data needs to be pre-processed such that all of these can be removed.

Filtering by Hashtag: The data retrieved above consists of tweets related to both presidential candidates. To find polarity of tweets that belong to each of them we need to consider polarity of tweets differently. Below Script separates the tweets depending on the hashtags that belong to these candidates.

```
import re


def test():
    input=open('data.txt','r')
    output=open('hillary.txt','w')
    check=open('hashhillary.txt','r')
    input_l=list()
    check_l=list()
    for word in check :
        word=word.rstrip('\n')
        i=check_l.append(word)
    for line in input:
        line=line.rstrip('\n')
        l=input_l.append(line)
    for w in check_l:
        for li in input_l:
            if w in li:
                #print li
                output.write(li + '\n')
    output.close()
a=test()
```

In the above Script the file ‘hashhillary.txt’ consists of hashtags that are used to support Hillary and oppose Trump.

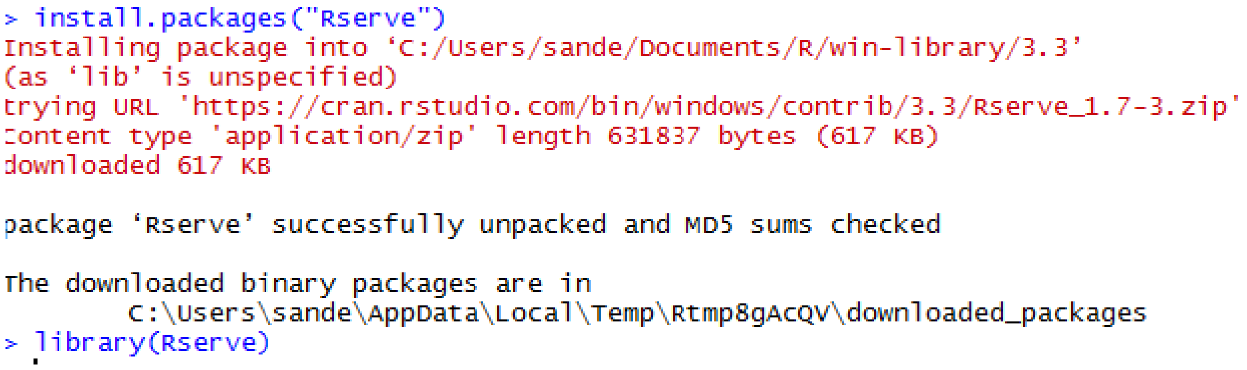
Finding Polarity:

To find the polarity of tweets we use R

- The sentiment of the mined tweets is analyzed using R.

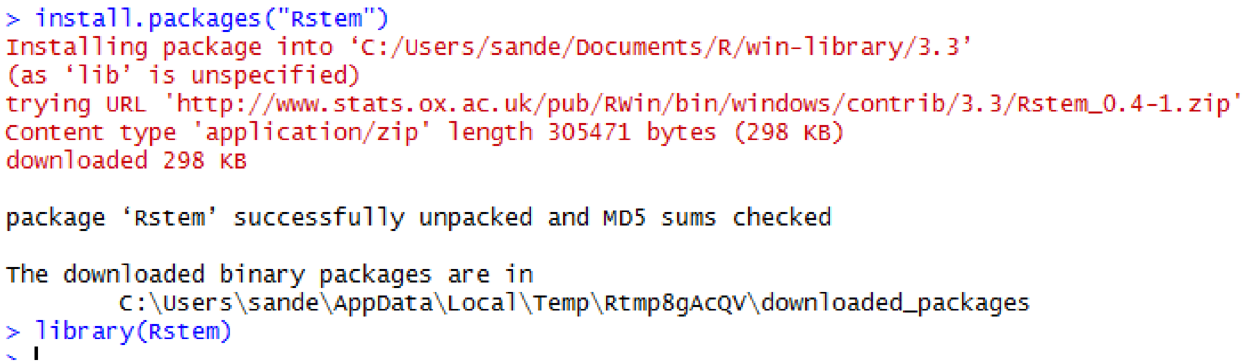
- The results of this are visualized using tableau.

- We need to install several packages for sentiment analysis through R.



```
> install.packages("Rserve")
Installing package into ‘C:/Users/sande/Documents/R/win-library/3.3’
(as ‘lib’ is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.3/Rserve_1.7-3.zip'
Content type 'application/zip' length 631837 bytes (617 KB)
downloaded 617 KB

package ‘Rserve’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\sande\AppData\Local\Temp\Rtmp8gAcQV\downloaded_packages
> library(Rserve)
```

- This Rstem package is used to provide bindings to word stemming in R language.



```
> install.packages("Rstem")
Installing package into ‘C:/Users/sande/Documents/R/win-library/3.3’
(as ‘lib’ is unspecified)
trying URL 'http://www.stats.ox.ac.uk/pub/RWin/bin/windows/contrib/3.3/Rstem_0.4-1.zip'
Content type 'application/zip' length 305471 bytes (298 KB)
downloaded 298 KB

package ‘Rstem’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\sande\AppData\Local\Temp\Rtmp8gAcQV\downloaded_packages
> library(Rstem)
```

- run.Rserve() makes the current Rserve process into an instance.

- Rserve starts a new process where as run.Rserve() turns the current R session into an Rserve.

- Now using the External service connection establish a connection between R and tableau.

- Specify the server name and port number as localhost and 6311 respectively.

Syuzhet in Analyzing Sentiment:

- The sentiment of tweets is evaluated by making use of syuzhet package.

- This package comes with basically four sentiment dictionaries which are “bing”, “afinn”, “nrc”, and “stanford”.

- By making use of the above dictionaries the get_sentiment () function

Will assess the sentiment of words or sentences depending on the given input.

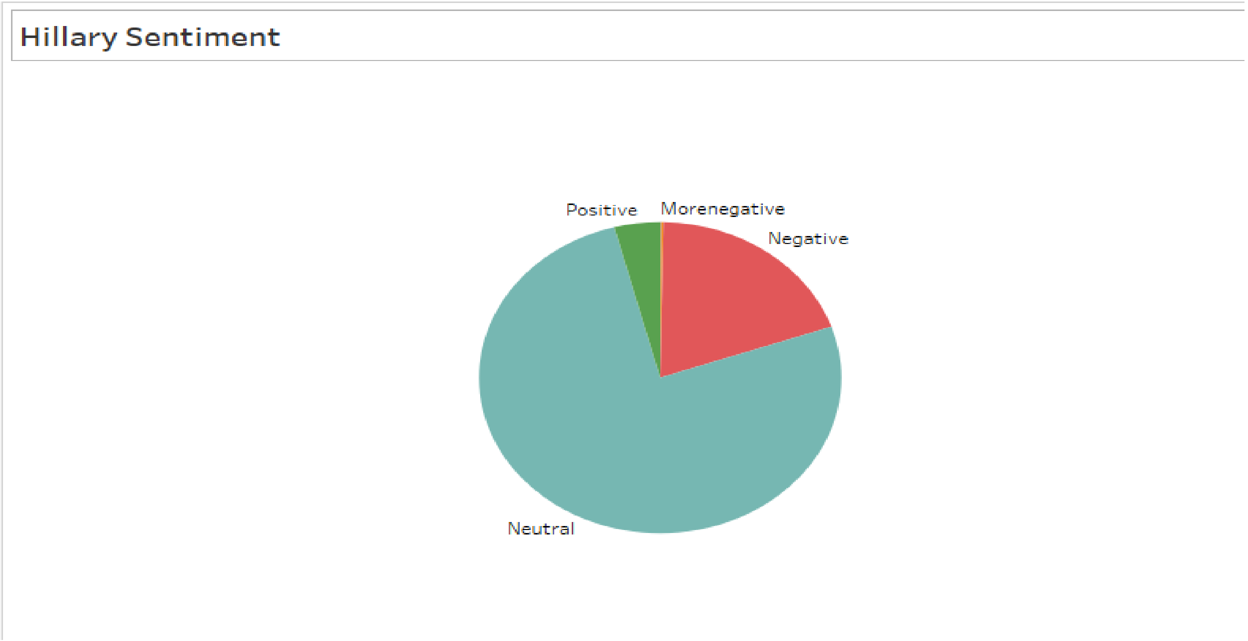
- The get_sentiment () function takes two arguments. A word or a sentence and a method.
- The sentiment is calculated based on the polarity of the sentences.
- The scale is divided into 5 ranges which fall under,
    - “more negative”,
    - “negative”,
    - “neutral”,
    - “positive”,
    - “more positive”.

Formula for Sentiment Analysis in Tableau using R:

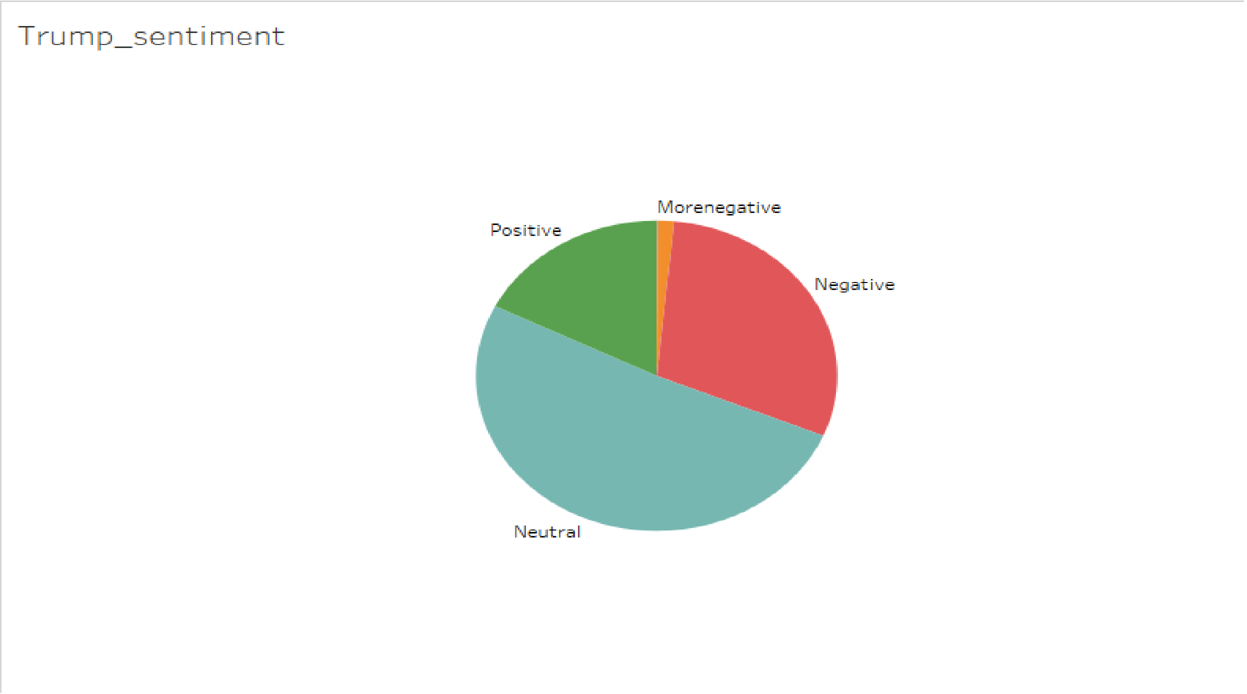
```
SCRIPT_STR('library(syuzhet);

r<-rescale(get_sentiment(.arg1,method = "syuzhet"))

as.character(cut(r,breaks=c(-Inf,-0.5,0,0.2,0.5,Inf),labels=c("morenegative","negative", "neutral", "positive", "more positive")))',

ATTR ([Tweet Text]))
```

The above formula executes the query on R using the established remote connection form Tableau.

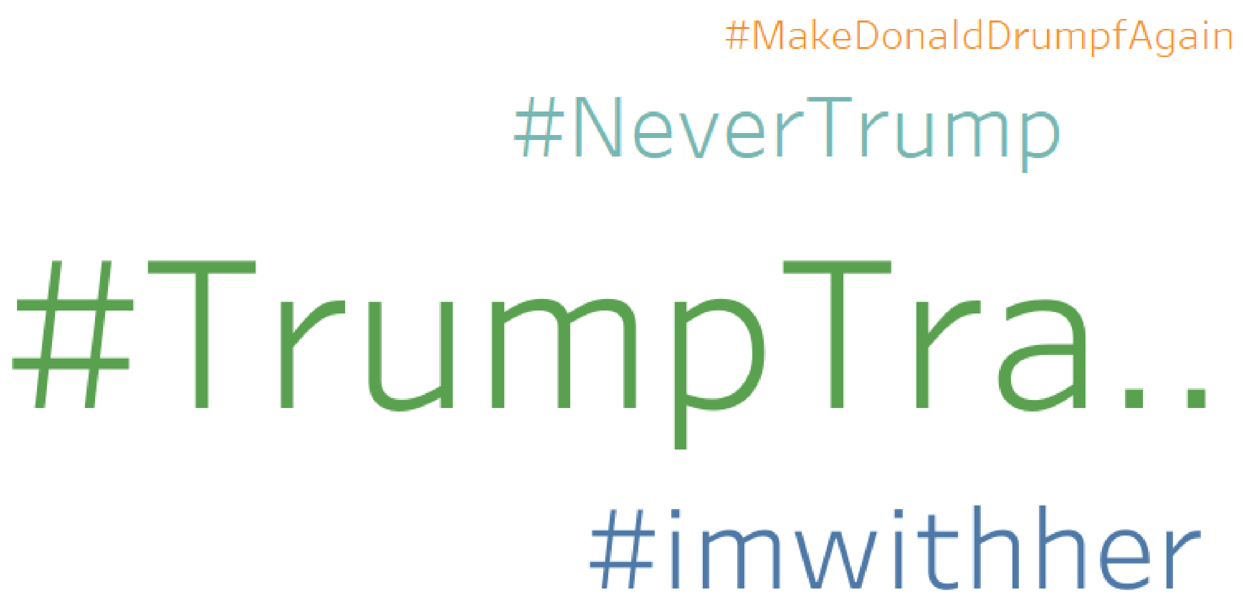
Visualization of Hillary's Sentiment: In this I have only plotted the polarities but as the "More positive" count is very less in number couldn't be seen in the chart.



Visualization of Trump's Sentiment: In this I have only plotted the polarities but as the "More positive" count is very less in number couldn't be seen in the chart.



Word Cloud for #hashtags:



The above hashtags represent the most used hashtags from the collected tweets.

Conclusion:

The above report describes how the text mining and sentiment analysis is done using Python and Twitter API's. The observed results show that polarity of tweets on both the candidates are inclined more towards "negative" and "neutral". This work can be extended such that the trend in polarity changes of tweeters can be identified if collected for a long period. Also, polarity of tweets based on location can be analyzed, if sufficient data is collected.

Twitter API can simply be used to access and collect tweets. There are some limitations on the amount of data users can collect which is a time-consuming task. However, there is an easy way of doing it by using more number of access tokens.

References:

[1] https://docs.python.org/2/whatsnew/2.7.html

[2] http://docs.tweepy.org/en/v3.5.0/

[3] http://casci.umd.edu/wp-content/uploads/2013/12/Tableau-Tutorial.pdf

[4]https://onlinehelp.tableau.com/current/pro/desktop/enus/calculations_calculatedfields_ex1create.html