Modern Information Management: Continuous Assessment

Twitter Sentiment Analysis

By: Navdeep Sharma and Sourbh Gaur

M.Sc. in Computer Science (Data Analytics)

Our thoughts and Prayers with Paris and the victims of Paris Attack.

Introduction:

Sentiment Analysis (also known as opinion mining) refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source material.

The purpose of text mining is to process unstructured (textual) information, extract meaningful numeric indices from the text and thus make the information containing in the text accessible to the various data mining algorithms. Hence, we can analyze words, clusters of words used in documents or we could analyze documents and can determine similarity between them.

Twitter is an amazing micro blogging tool and an extraordinary communication medium. In addition, twitter can also be an amazing open mine for text and social web analysis. In this real world, many real world events are commented on by many fans through twitter. Every comment have an associated opinion and sentiments either positive or negative.

The goal of this project is to summarize the events by extracting and aggregating tweets and to develop a model to analyze the sentiments/emotions expressed by the users. The history of the tweets are important as the emotions can vary with time of the events. Considering all the factors, we are developing an approach using classical search model, all the stages of the text processing and machine learning algorithm to analyze the sentiments and opinion of the people. For our project, we are considering the most recent tragic event of Paris Attack (#ParisAttack) and would try to design a ground of prediction using R and implement our approach and at last, will evaluate our approach.

Tools and Packages used in R:

In this project “Twitter Sentiment Analysis Using R”, we have used RStudio GUI and following packages:

- **twitteR** : Provides an interface to the Twitter web API.
- **ROAuth** : This package provides an interface to the OAuth 1.0 specification, allowing users to authenticate via OAuth to the server of their choice.
- **plyr** : This package is a set of tools that solves a common set of problems: you need to break a big problem down into manageable pieces, operate on each pieces and then put all the pieces back together.
- **stringr** : stringr is a set of simple wrappers that make R's string functions more consistent, simpler and easier to use. It does this by ensuring that: function and argument names (and positions) are consistent, all functions deal with NA's and zero length character appropriately, and the output data structures from each function matches the input data structures of other functions.
- **ggplot2** : An implementation of the grammar of graphics in R. It combines the advantages of both base and lattice graphics: conditioning and shared axes are handled automatically, and you can still build up a plot step by step from multiple data sources.
- **RColorBrewer** : The packages provides palettes for drawing nice maps shaded according to a variable.
- **tm** : A framework for text mining applications within R.
- **wordcloud** : This package helps in creating pretty looking word clouds in Text Mining.

High Level Steps of Design and Implementation:

1. Creating a twitter authentication Application
2. Working with R studio – Building the corpus (extraction of tweets).
3. Extraction and pre-processing of text from the tweets.
4. Sentimental Analysis
5. Evaluation of the design.

Detailed Description of all the steps:

1. Creating a twitter authentication application:

First step to perform Twitter Analysis is to create a twitter application. The steps for creating your twitter applications are:

- Go to https://dev.twitter.com and login by using twitter account.
- Then go to My Applications → Create a new application.
- Follow the necessary steps such as provide application name, website URLs and fill the mandatory fields. The important is to make note of the Consumer key and Consumer Secret numbers as they will be used in RStudio later.

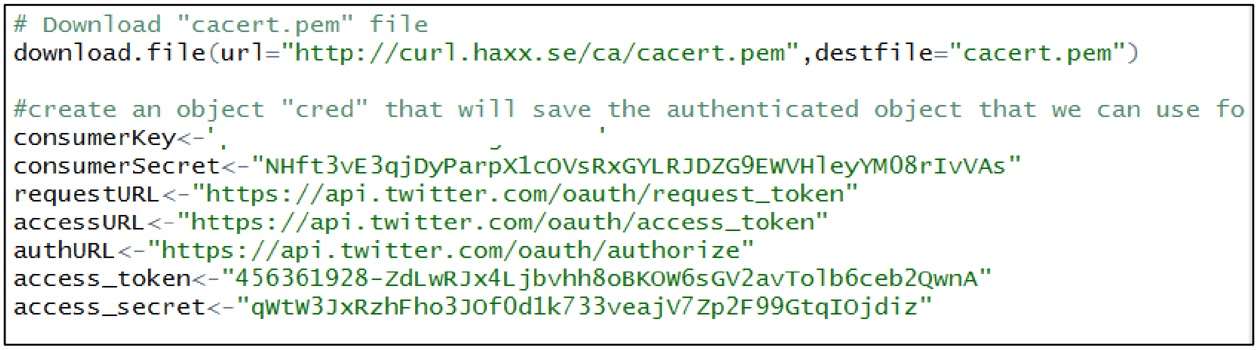
2. Working with Rstudio- building the corpus (extraction of tweets):

I. In this section, we will first use some packages in R. These are twitter, ROAuth, plyr, stringr and ggplot2. You can install these packages using install.packages("package name") function in r.

II. We can load the downloaded packages in the Rstudio session using library("package name")

III. Windows users need to download a small file by using download.file(url, destfile) function where url = "*http://curl.haxx.se/ca/cacert.pem*" and destfile="cacert.pem"

IV. Now once file is downloaded, we are now moving on to accessing the twitter API. This step include the script code to perform handshake using the Consumer Key and Consumer Secret number of our application.



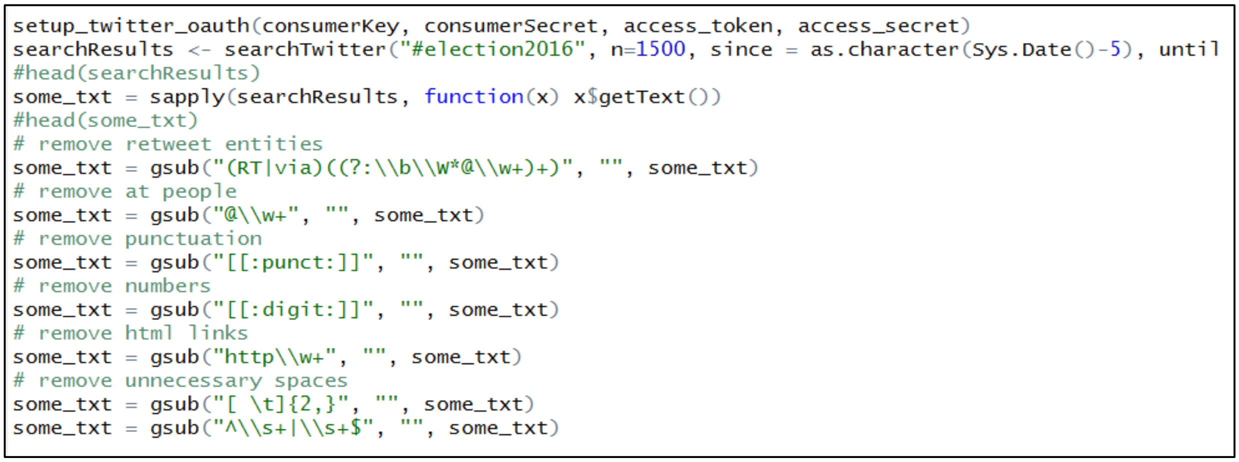
```
# Download "cacert.pem" file
download.file(url="http://curl.haxx.se/ca/cacert.pem",destfile="cacert.pem")

#create an object "cred" that will save the authenticated object that we can use fo
consumerKey<-',
consumerSecret<-"NHft3vE3qjDyParpX1cOVsRxGYLRJDZG9EWVHleyYM08rIvVAs"
requestURL<-"https://api.twitter.com/oauth/request_token"
accessURL<-"https://api.twitter.com/oauth/access_token"
authURL<-"https://api.twitter.com/oauth/authorize"
access_token<-"456361928-ZdLwRJx4LjbvhhhoBKOW6sGV2avTolb6ceb2QwnA"
access_secret<-"qWtW3JxRzhFho3JOf0d1k733veajV7Zp2F99GtqIOjdiz"
```

Figure 1 : Providing Twitter authentication details

3. Extraction and pre-processing of text:

Once we have the tweets we just need to apply some functions to convert these tweets into some useful information. The main working principle of sentiment analysis is to find the words in the tweets that represent positive sentiments and find the words in the tweets that represent negative sentiments.

We will use preprocessing techniques such as word removal (removal of section such as retweets, URLs, spaces, punctuations, tagged people etc.). So we remain with text to perform our analysis.

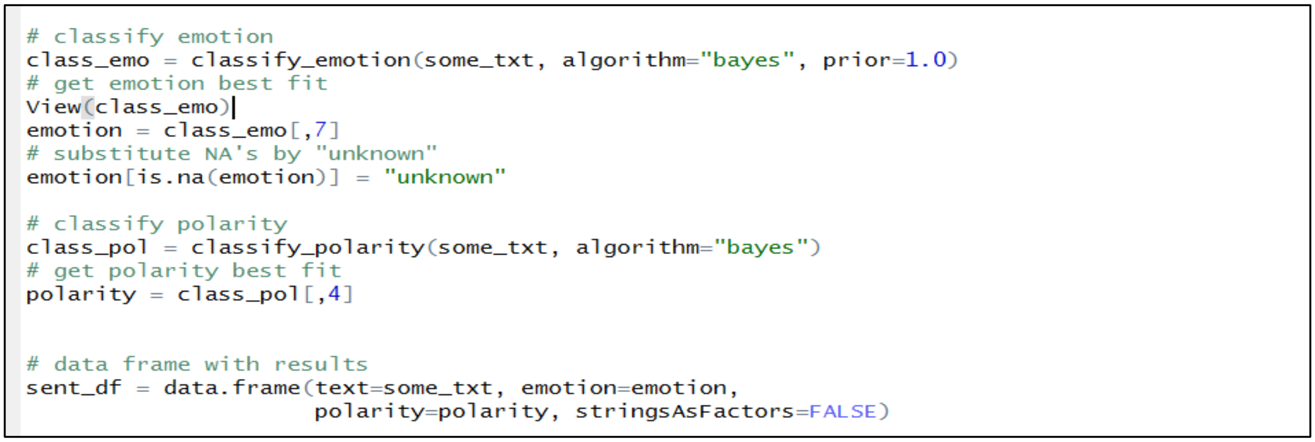


```
setup_twitter_oauth(consumerKey, consumerSecret, access_token, access_secret)
searchResults <- searchTwitter("#election2016", n=1500, since = as.character(Sys.Date()-5), until
#head(searchResults)
some_txt = sapply(searchResults, function(x) x$getText())
#head(some_txt)
# remove retweet entities
some_txt = gsub("(RT|via)((?:\\b\\W*@\\w+)+)", "", some_txt)
# remove at people
some_txt = gsub("@\\w+", "", some_txt)
# remove punctuation
some_txt = gsub("[[:punct:]]", "", some_txt)
# remove numbers
some_txt = gsub("[[:digit:]]", "", some_txt)
# remove html links
some_txt = gsub("http\\w+", "", some_txt)
# remove unnecessary spaces
some_txt = gsub("[ \t]{2,}", "", some_txt)
some_txt = gsub("^\\s+|\\s+$", "", some_txt)
```

Figure 2: Extracting tweets and removing noise

4. Perform Sentimental Analysis:

Sentiment analysis is a classification task to classify words into positive words and negative words. The algorithm we chose for this task is "Bayes". We can implement this algorithm using classify_emotion() function from twitter package. This implementation results to classification on the basis of sentiments.



```
# classify emotion
class_emo = classify_emotion(some_txt, algorithm="bayes", prior=1.0)
# get emotion best fit
View(class_emo)
emotion = class_emo[,7]
# substitute NA's by "unknown"
emotion[is.na(emotion)] = "unknown"

# classify polarity
class_pol = classify_polarity(some_txt, algorithm="bayes")
# get polarity best fit
polarity = class_pol[,4]


# data frame with results
sent_df = data.frame(text=some_txt, emotion=emotion,
                     polarity=polarity, stringsAsFactors=FALSE)
```

Figure 3 : Performing classification on tweets with bayes algorithm

5. Results, General statistics and Evaluation:

Now we have the clear classification of emotions. We can judge the emotions of people over this event and visualize their sentiments by plotting graphs over polarity and emotions.

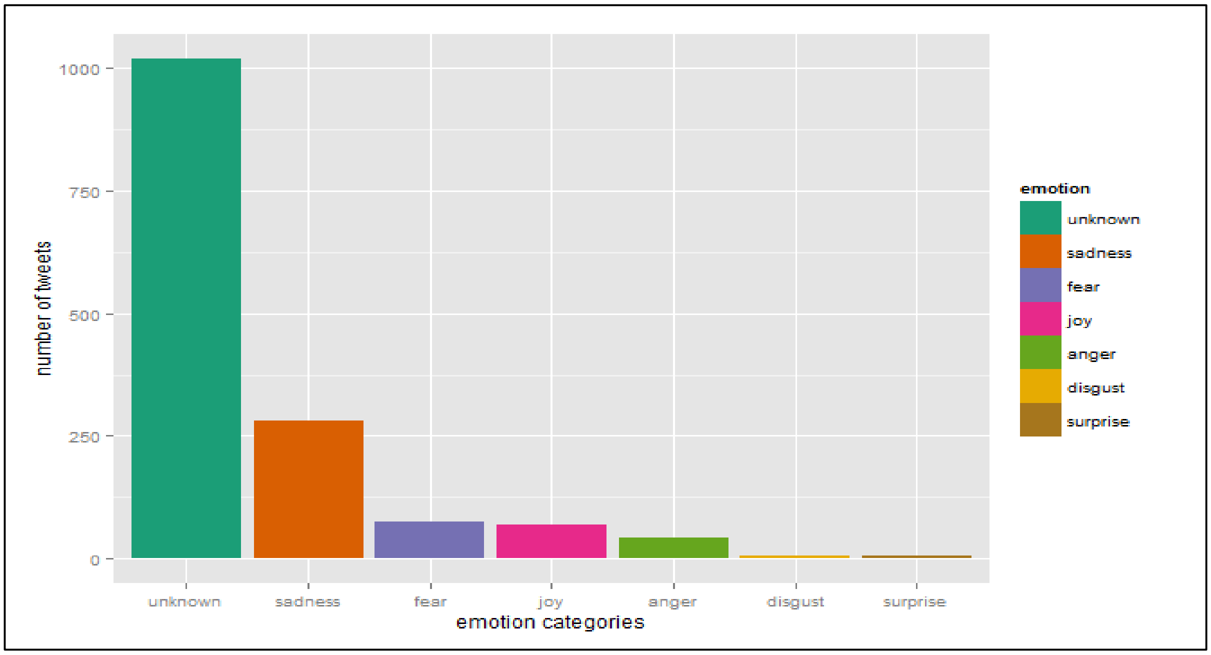


Figure 4 : Classification of tweets under different emotions

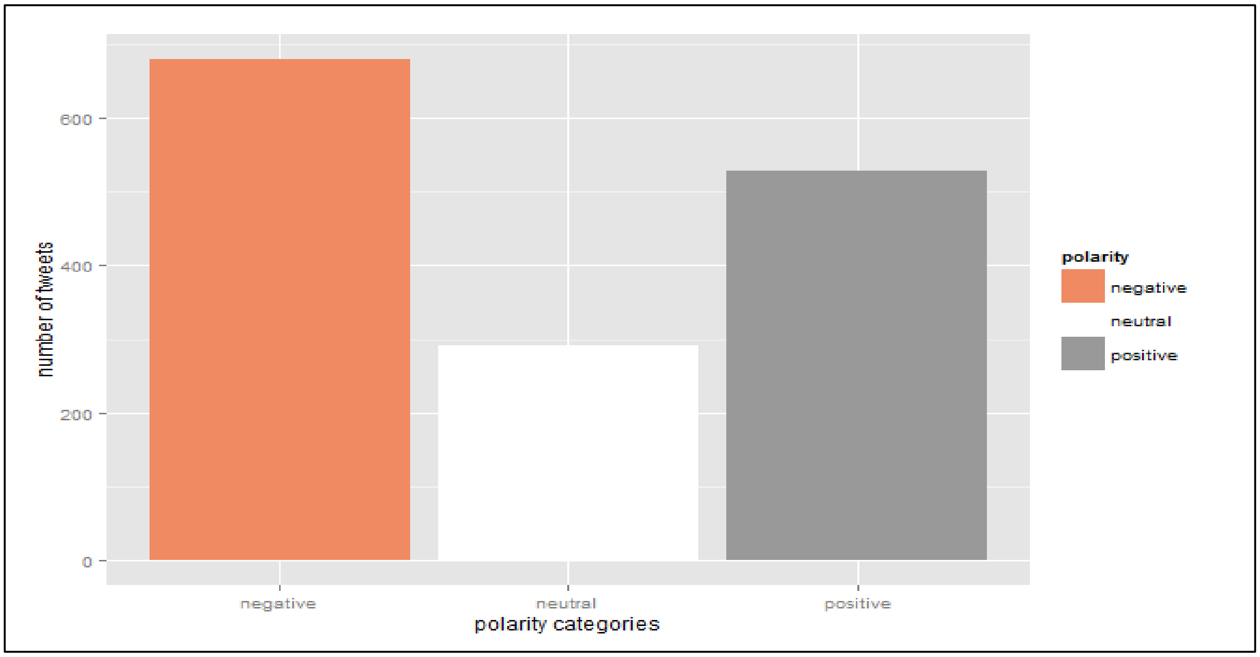


Figure 5 : Polarity categorization

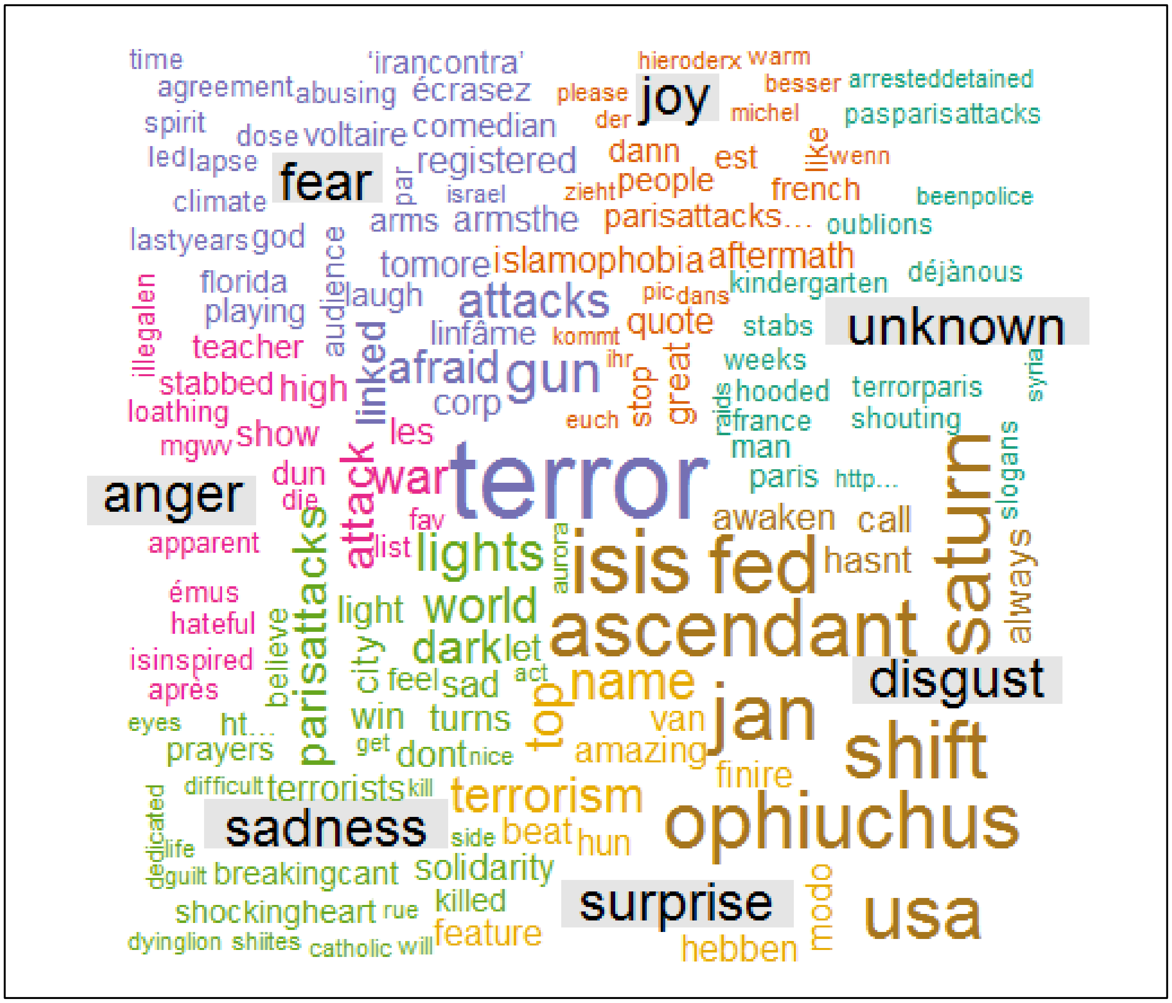


Figure 6 Word cloud, displaying more neighborhood words used in tweets

The Sentimental analysis comes quite as expected. This result is completely based on factors such as duration of tweets, amount of tweets fetched, algorithm used to classify emotions. We can try few things such as performing classification on higher number of tweets, changing the classification algorithm, performing cross validations, in order to get even better results. We finally reach to the conclusion that algorithm is classifying tweets as per their emotion very well but there is a major amount of tweets that are still under unknown category and that motivates to take this analysis even forward.

References:

www.google.com
Tutorial on Sentimental Analysis – Fabio Benedetti
Twitter Analysis (Tutorial Using R) – Kaify Rias