Project Report

Twitter Sentiment Analysis Using R

(in RStudio)

TABLE OF CONTENTS

Abstract

Twitter! Who hasn't heard of this micro-blogging site? All of us have. It is a medium which facilitates users to exchange short messages more popularly known as "tweets". People use twitter to express their individual views and opinions on various topics. Needless to say, a variety of emotions are observed. We have three basic categories – a positive response/opinion, a negative response/opinion or a negative response/opinion. Be it reviews about movies, the elections, football, and so on, everybody has their own say.

In the scope of this project, with help of Twitter and a good data science tool (R Language in our case), we are going to find out what the world is saying about a particular topic. We are going to find out how many people are for, against or maintain a neutral stance on Manchester City (a popular English soccer club). Technically, we are doing a sentiment analysis on the views and opinions expressed about Manchester City. We are hoping to fetch the data from Twitter, mine them for the adjectives and then categorize them as for, against or neutral based on the adjective being a positive or negative adjective. Performing this exercise helps us to understand the challenges in mining social media and also enlighten us on how to integrate an Application Programmer Interface (API) in our scripts to access the information available on social media sites (in our case, Twitter).

Real-time tweets are obtained from the micro blogging website and then we separate the individual words and compare them with previously stored positive and negative word lists. Based on this matching, the number of positive words and negative words is obtained. After this, a sentiment score is calculated by calculating the difference of positive words and negative words. And finally, we analyse if the tweet is positive, negative or neutral based on this score.

Introduction

Social media platforms are one of the major sources of data in today's modern world. The information that is found here is very varied. Different kinds of people have different takes on a particular topic and behind the shelter of their computer screens, a lot of people are more comfortable about sharing their views and opinions. People post real time views about current issues, their likes and dislikes about a particular topic, sentiments for products they use in daily life and so on.

Twitter is a rapidly expanding service with over 200 million registered users, out of which 100 million are active users and half of them log on to Twitter on a daily basis - generating nearly 250 million tweets per day. Users can post short, real-time messages on Twitter called tweets. Tweets are short messages limited to 140 characters in length. Users generally use hashtags (#) to mark topics. The primary use of this is to increase the visibility of the tweets. The nature of this site is quick and short messages, and thus people sometimes resort to using smiley faces or acronyms to convey their emotions by using lesser number of characters. Smiley faces are generally called emoticons that pictorially represent the various facial expressions of humans. ☺ - indicates happy, ☹ - indicates sad and so on. We can make an attempt at judging the mood of the user on the basis of this. Acronyms are short, colloquial forms of regular words. For example, gr8 for great, lol for laughing out loud and so on. There are tons of acronyms popular on social media sites. For the scope of our project, we have focussed only on the text part of the tweet and have not analysed emoticons.

According to Wikipedia, Sentiment Analysis refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials. Sentiment Analysis is also known as or referred to as Opinion Mining which simply means extracting opinions, emotions and sentiments in text. As one can imagine, one of the most common applications of Sentiment Analysis is to track attitudes, feelings and emotions on the web, especially for products, brans, movie reviews, or even people. The main idea is to determine if the audience has a positive or negative view on a particular topic.

Why R for our project?

Among the different softwares that can be used to analyse Twitter, R offers a wide variety of options and we can do a lot of interesting things and innovative things. We have also used RStudio for our project, as it is very convenient to work on R scripts on RStudio.

One of the most powerful aspects of using R is that you can download free packages for so many tools and types of analysis.

R is a language and environment for statistical computing and graphics and can perform a wide variety of computing tasks by giving them access to various commands.

With R, we can quickly and easily clean, and explore the information that we want to analyse. R allows the business to explore new information with the help customised visualizations.

For our project R came in handy by providing visual graphics and statistics that showed the neutral, positive and negative tweets extracted from twitter about a Manchester City soccer club.

Features of R:

On The Functional Side:

- has first class functions
- has lazy evaluation of arguments

On The Object Oriented Side:

- It has built in Object Oriented paradigms: S3 and S4, which are immutable and support the more common message-passing style Object Oriented.

In contrast to languages such as Java and C++ where objects are distinct from more primitive data types, every reference in R is to an object. **Object orientated**.

- Can function many calculations on vectors (c( ))
- Can process statistics data
- Open source
- Is available for different types of hardware and software

Cons of R

- Memory Management
- Speed
- Efficiency
- Implemented from S, so its basic principles are derived from an older language.

Motivation

We have chosen the “Twitter” social network for our project because it is the only place where both brands and consumers have an even playing field with unrestricted lines of clear, concise communication. Twitter not only attracts a unique audience, but it is also easy to pinpoint a particular group of audience. For instance, for our project we have attempted an analysis of sentiments regarding a popular soccer team, and the above mentioned feature made it easy for us to target that group of audience that vests a lot of their time on following the game of soccer, watching soccer matches on a regular basis and also share their opinions on Twitter. One of the most interesting things that you could ever come across on the internet is two ardent followers of rival football clubs, going at each other full throttle in an enthralling argument, showcasing both their abilities of being able to recall the most complicated statistics and being humorous at the same time, in an amicable manner! It is easier to discern interests on Twitter because one does not have to go through the incredibly long individual profiles, but can simply search the tweets, retweets and hashtags of the particular person.

The analysis of sentiments is a big thing these days and there are several companies who are pouring in lots of money for the research of technology that can help them get a sense of general sentiment for their product. Separate wings of operations are being opened and analysts are being hired in a good number to study user reactions on micro blogs and also provide replies to the user, if required. The primary challenge of course is to build a state of the art technological product that can detect and summarise an overall sentiment. Sentiment analysis of public is highly critical in macro-scale socioeconomic phenomena like predicting the stock market rate of a particular firm. Firms can also estimate how well their product is responding in the market, which areas of the market is it having a favourable response and in which a negative response (since twitter allows us to download stream of geo-tagged tweets for particular locations).

Limitations

There are a few limitations while doing Twitter Analysis using R that we have to keep in mind before embarking upon the project.

First, while getting status of the user timeline, the method can only return a fixed maximum number of tweets which is limited by the Twitter API.

Second, while requesting tweets for a particular keyword, it happens frequently that the number of retrieved tweets are less than the number of required tweets.

Tools and Packages Used

In our project we have used RStudio for implementing R Scripts and the following packages:

twitter – Provides an interface to the Twitter Web API.

plyr – This package is a set of tools that solves a common set of problems. It is essential when we have to break down a big problem into manageable, operational pieces, perform functionalities on the pieces and then put all the pieces back together. In our project we use this package to clean data.

stringr – This package consists of 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 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. We use this package in our project to perform manipulation on strings.

ggplot2 – This package serves the purpose of implementation of the grammar 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.

dplyr - This package simplifies how we think commonly about data manipulation tasks by constraining our options. It provides simple "verbs", functions that correspond to the most common data manipulation tasks, to help you translate those thoughts into code. It uses efficient data storage backends, so you spend less time waiting for the computer. We use it in our project to implement the summarise function.

plotrix - Lots of plots, various labeling, axis and color scaling functions. We use it in our project to view a 3D piechart.

RCurl - A wrapper for 'libcurl'. Provides functions to allow one to compose general HTTP requests and provides convenient functions to fetch URIs, get & post forms, etc. and process the results returned by the Web server. This provides a great deal of control over the HTTP/FTP/... connection and the form of the request while providing a higher-level interface than is available just using R socket connections. Additionally, the underlying implementation is robust and extensive, supporting FTP/FTPS/TFTP (uploads and downloads), SSL/HTTPS, telnet, dict, ldap, and also supports cookies, redirects, authentication, etc.

RStudio - RStudio is an integrated development environment (IDE) for the R programming language. Some of its features include:

- Customizable workbench with all of the tools required to work with R in one place (console, source, plots, workspace, help, history, etc.).
- Syntax highlighting editor with code completion.
- Execute code directly from the source editor (line, selection, or file).
- Runs on all major platforms (Windows, Mac, and Linux) and can also be run as a server, enabling multiple users to access the RStudio IDE using a web browser.

Domain Introduction

This project of analyzing sentiments of tweets comes under the domain of “Pattern Classification” and “Data Mining”. Both of these terms are very closely related and intertwined, and they can be formally defined as the process of discovering “useful” patterns in large set of data, either automatically (unsupervised) or semi-automatically (supervised).

Unsupervised method - In unsupervised methods, no target variable is identified as such. Instead, the data mining algorithm searches for patterns and structure among all the variables. The most common unsupervised data mining method is clustering.

Supervised method – In supervised methods, there is a particular pre-specified target variable, and the algorithm is given many examples where the value of the target variable is provided, so that the algorithm may learn which values of the target variable are associated with which values of the predictor variables.

The features that can be used for modelling patterns and classification can be divided into two main groups: formal language based and informal blogging based. Language based features are those that deal with formal linguistics and include prior sentiment polarity of individual words and phrases Prior sentiment polarity means that some words and phrases have a natural innate tendency for expressing particular and specific sentiments in general. For example, the word “excellent” has a strong positive connotation while the word “evil” possesses a strong negative connotation. So whenever a word with positive connotation is used in a sentence, chances are that the entire sentence would be expressing a positive sentiment and the same applies for negative connotation as well.

Twitter based features are more informal and relate with how people express themselves on online social platforms and compress their sentiments in the limited space of 140 characters offered by twitter. They include twitter hashtags, retweets, word capitalization, lengthening, question marks, presence of url in tweets, exclamation marks, internet emoticons, internet shorthand/slangs and so on.

There are two types of adaptive techniques: Passive and active. Passive techniques are the ones which use the feedback only to learn about the environment but not using this improved learning in our current classification algorithm, while the active approach continuously keeps changing its classification algorithm according to what it learns at real-time.

Sentiment analysis of in the domain of micro-blogging is a relatively new research topic so there is still a lot of room for further research in this area. Decent amount of related prior work has been done on sentiment analysis of user reviews, documents, web blogs/articles and general phrase level sentiment analysis. These differ from twitter mainly because of the limit of 140 characters per tweet which forces the user to express opinion compressed in very short text. The best results reached in sentiment classification use supervised learning techniques.

Functionality and Design

Data Acquisition - Data in the form of raw tweets is acquired by using the R library “twitteR” which provides a function for simple twitter streaming API named searchTwitter(). This function takes in arguments such as the topic for which tweets have to be fetched preferably prefixed with a hashtag (#), number of tweets to be fetched, the time period during of the tweets to be fetched, the language of the tweets and so on.

Since we wanted to increase the generality of our data, we acquired it in portions at different points of time instead of acquiring all of it at one go. If we used the latter approach then the generality of the tweets might have been compromised since a significant portion of the tweets would be referring to some certain trending topic and would thus have more or less of the same general mood or sentiment.

A tweet acquired by this method has a lot of raw information in it which we may or may not find useful for our particular application. It comes in the form of various key-value pairs. A list of some key-value pairs are given below:

• Whether a tweet has been favourite

• Screen name of the user

• Original Text of the tweet

• Presence of hashtags

• Whether it is a re-tweet

• Date and time when the tweet was created

This is a lot of information. While performing the project we only filter out what we need and then proceed accordingly. Generally speaking, the date and time creation of the tweet and the actual text of the tweet is of utmost importance and is very crucial from the point of view of what we are seeking to achieve in our project.

Since human labelling is an expensive process, we further filter out the tweets to be labelled so that we have the greatest amount of variation in tweets without the loss of generality. The filtering criteria applied are stated below:

• Replace special symbols like @, via, RT, etc with “”

• Replace white spaces with “”

• Replace digits with “”

• Replace punctuation marks with “”

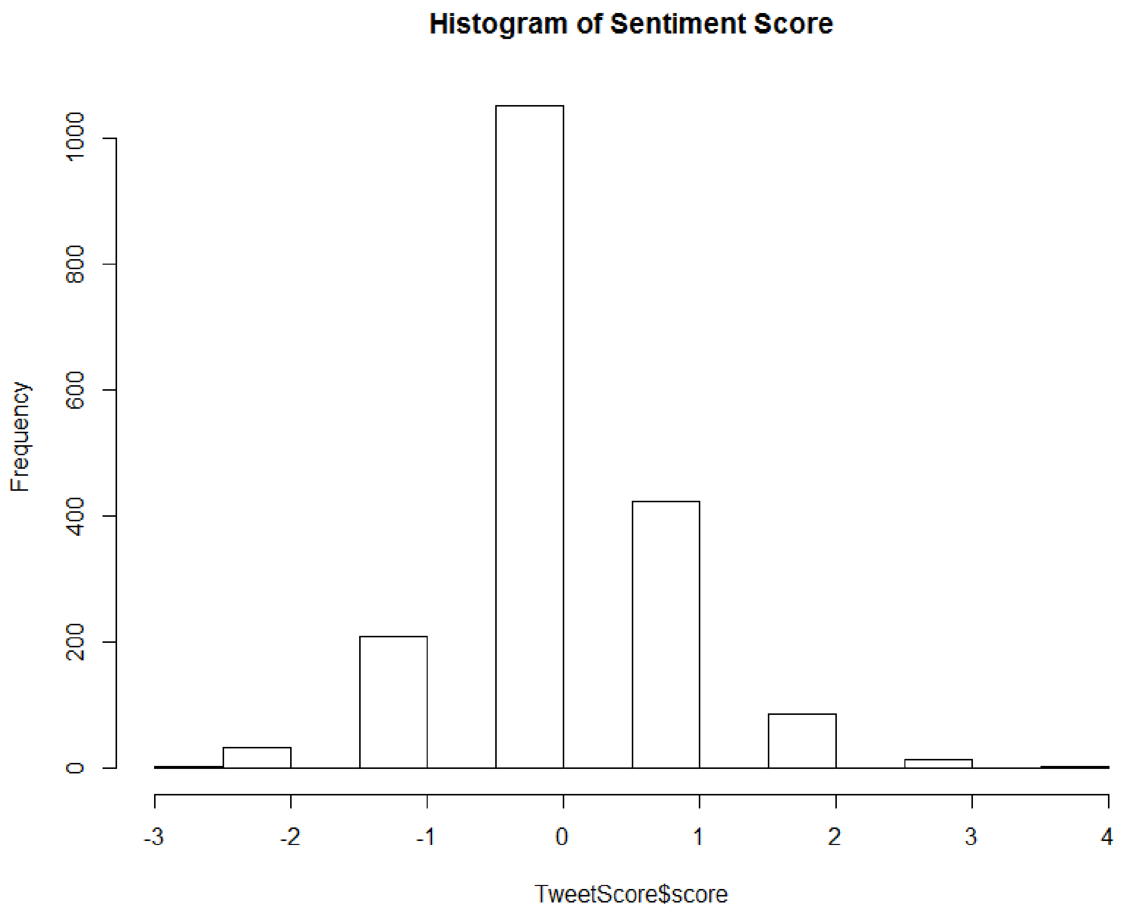
Classes of Tweets

We labelled the tweets in three classes according to sentiments expressed/observed in the tweets: positive, negative, neutral.

We arrived at the following statistics for each class:

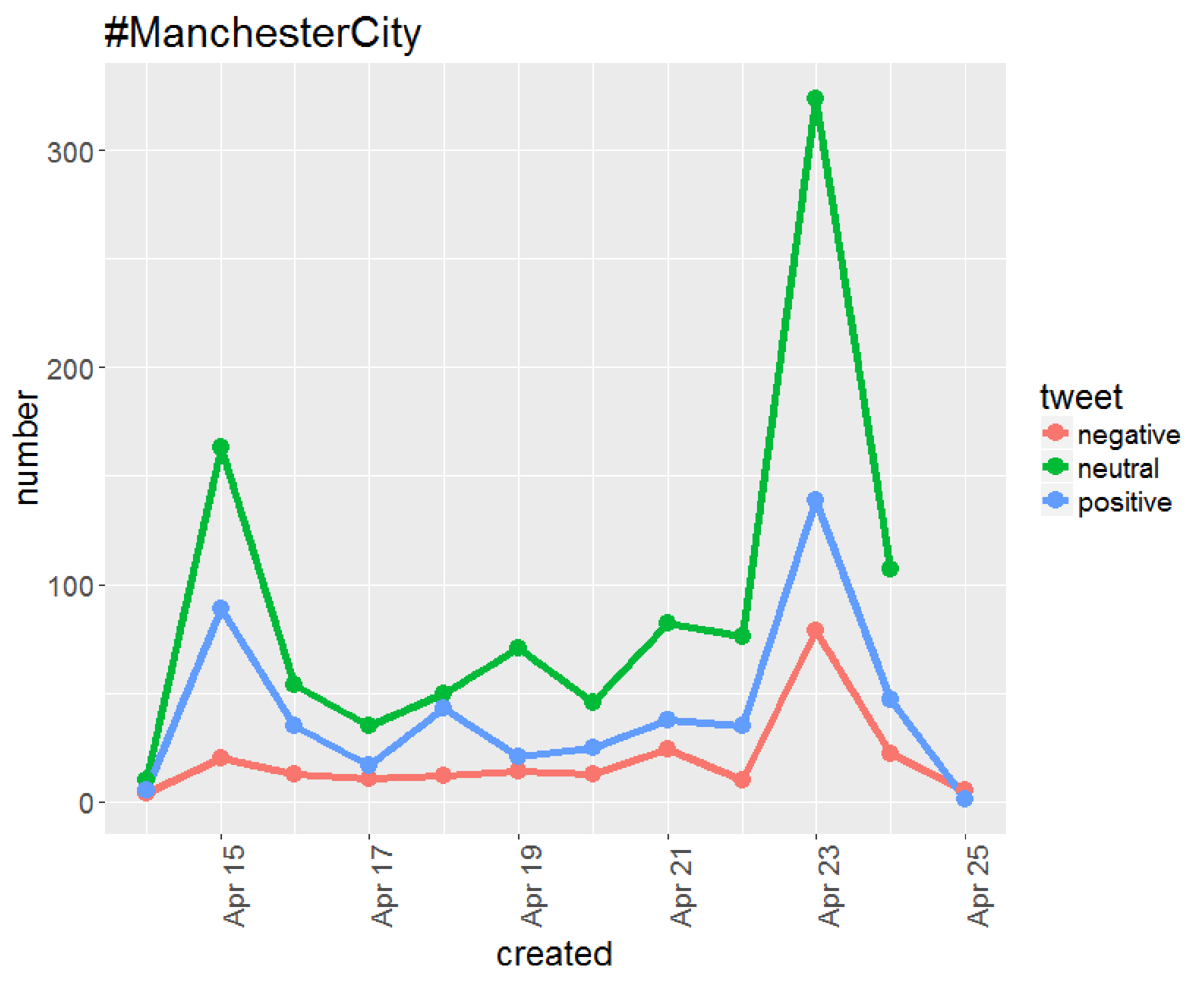
Negative tweets	227 tweets
Positive tweets	1018 tweets
Neutral tweets	495 tweets
Total tweets	1740 tweets

Graphical representation of analysis

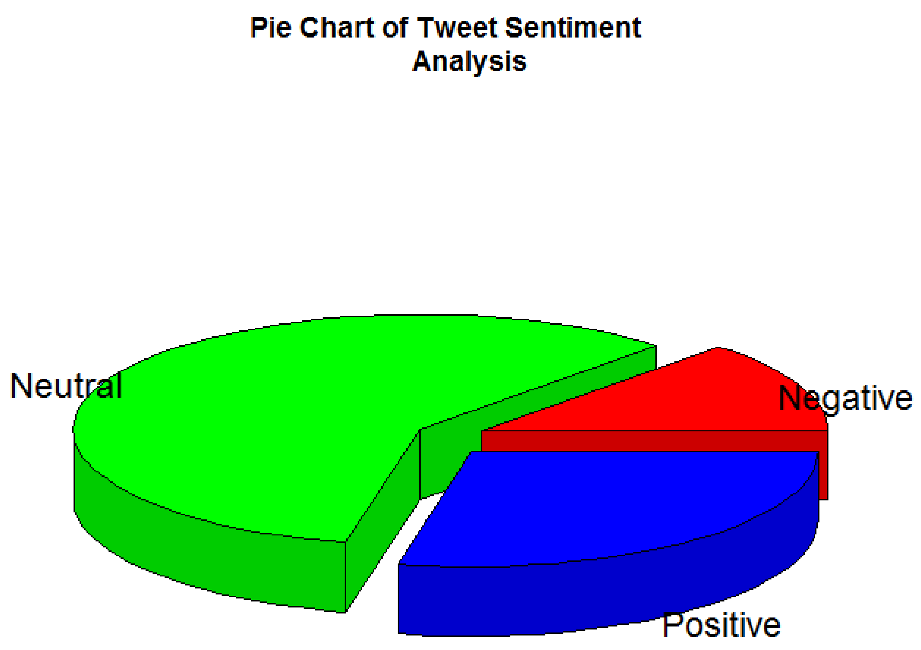


Sentiment score calculation:
greater than 1 indicates a positive tweet, less than 1 indicates a negative tweet,
and 0 indicates a neutral tweet. This is implemented using the if-else function in our project.

PLOT FOR #MANCHESTERCITY



This graph shows the sentiment of people on the particular dates based on the sentimental score functionality described above. For example, on April 15th, there were about 20 negative tweets, about 80 positive tweets and about 170 neutral tweets about #ManchesterCity. In general, it can be inferred that neutral tweets are greater than positive and negative tweets about #ManchesterCity between April 15th and April 25th.



The above is a Pie chart based on the count of negative, positive and neutral tweets between April 15th and April 25th for #ManchesterCity.

Code

#Team members :

#Lohitaksh Yogi

#Tin Le

#Ahmad Salah

#Jorge Palacios

#Nasif Majeed

#Aabir Kumar Datta

```
#Provides functions for general HTTP requests, GET and POST forms
install.packages('RCurl')

#To create an interface with the Twitter web API
install.packages('twitteR')

#Needed for the summarise function. Here we summarise the tweets and number them
install.packages('dplyr')

#Used to split, add functionality and combine data. Here we use to clean our data
install.packages('plyr')

#Provides creative data visualisation
install.packages('ggplot2')

#To manipulate strings
install.packages('stringr')

#For 3D graph visualisation
install.packages('plotrix')


#making the packagaes for the particular session as they are not included by default
require(twitteR)
require(RCurl)
require(dplyr)
require(ggplot2)
require(plyr)
require(stringr)
require(plotrix)
```

```
library(twitteR)

library(RCurl)

library(dplyr)

library(ggplot2)

library(plyr)

library(stringr)

library(plotrix)


#The details of the twitter application created at dev.twitter.com via a twitter account

consumer_key <- 'HqB7aGKH1vLuPQ7gX0lWcQuc2'

consumer_secret <- 'DZtrxs7eLq8DNCASXsCu8jwGwUtwUsP4kLbnHAXuUn3Nz0JU2z'

access_token <- '849363124780138496-Kcb11TXTFK3aLCRaqYXevZOKu0KVaBV'

access_secret <- 'bkZXT8GvNQcdwQJBlZh14OLMTegvR3TsVLIl2Uo51GG5J'



#Accessing the twitter API using the above details

#Select option 2, that is NO, when prompted after executing this command.

#We are NOT using a local file for authorisation of credentials and hence No.

setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)



#Obtaining n tweets using the searchTwitter function for a particular time period

ManchesterCityTweets <- searchTwitter("#ManchesterCity", n = 2000, since = '2007-10-30',
resultType = 'recent',

                         lang = "en")

ManchesterCityTweets



#Converting the list of tweets to a dataframe

mcdf <- twListToDF(ManchesterCityTweets)
```

```
#Cleaning the data and by removing blank spaces, punctuation marks, symbols and so on.
#We do this by replacing the unwanted characters by a "".
#gsub function is used for replacement.
ManchesterCityTweets$text = gsub("[:blank:]", "", ManchesterCityTweets$text)
ManchesterCityTweets$text = gsub("[[:punct:]]", "", ManchesterCityTweets$text)
ManchesterCityTweets$text = gsub("[:cntrl:]", "", ManchesterCityTweets$text)
ManchesterCityTweets$text = gsub("[[:digit:]]", "", ManchesterCityTweets$text)
ManchesterCityTweets$text = gsub("[:blank:]", "", ManchesterCityTweets$text)
ManchesterCityTweets$text = gsub("(RT|via)((?:\\b\\W*@\\w+)+)", " ", ManchesterCityTweets$text)
ManchesterCityTweets$text = gsub("@\\w+", "", ManchesterCityTweets$text)
ManchesterCityTweets$text = gsub("http\\w+", "", ManchesterCityTweets$text)


#When the tweets are obtained, all the details related to the tweet like user, number of retweets,
#and so on are also obtained.
#Here we create a subset that contains only the text content of the tweets.
mcdf_subset = subset(mcdf, select = c(text))
mcdf_subset


#The column names are arranged in alphabetical order.
#order () is used for the ordered arrangemnt.
#names () contains the column names
# , indicates all columns.
names(mcdf)
mcdf <- mcdf[, order(names(mcdf))]
```

```
#The date the dataframe is created in converted into YYYY-MM-DD format using the strftime
function.

mcdf$created <- strftime(mcdf$created, '%Y-%m-%d')



#If file doesn't exist, then create a file with a particular name.

#paste() combines two strings

#write.csv function converts the dataframe into a csv file.

if (file.exists(paste('#ManchesterCity', 'Proj.csv'))==FALSE)

  write.csv(df, file=paste('#ManchesterCity', 'Proj.csv'), row.names=F)



#Read the previously created CSV file and remove duplicate text.

#read.csv() is used to read the CSV file.

#rbind () is used to combine the dataframe and CSV file.

ReadTweetFile <- read.csv(file=paste('#ManchesterCity', 'Proj.csv'))

ReadTweetFile <- rbind(ReadTweetFile, mcdf)

ReadTweetFile <- subset(ReadTweetFile, !duplicated(ReadTweetFile$text))

write.csv(ReadTweetFile, file=paste('#ManchesterCity', 'Proj.csv'), row.names=F)



#function to calculate sentiment score

#lapply() is used to apply a function to a dataframe

#str_split() is used to split the individual words in the sentence

#unlist() creates a vector using the individual components of str_split

#This vector is compared with our stock of positive and negative words and a score is
calucalated

#based on number of positive words - number of negative words

SentimentScoreFunction <- function(OurSentence, PositiveWordList, NegativeWordList,
.progress='none')

{
  require(plyr)
  require(stringr)
  TweetScore <- laply(OurSentence, function(sentence, PositiveWordList, NegativeWordList){
  sentence <- gsub('[[:punct:]]', "", sentence)
  sentence <- gsub('[[:cntrl:]]', "", sentence)
  sentence <- gsub('\\d+', "", sentence)
  sentence <- tolower(sentence)
  OurWordList <- str_split(sentence, '\\s+')
  Words <- unlist(OurWordList)
  PositiveWordMatches <- match(Words, PositiveWordList)
  NegativeWordMatches <- match(Words, NegativeWordList)
  PositiveWordMatches <- !is.na(PositiveWordMatches)
  NegativeWordMatches <- !is.na(NegativeWordMatches)
  SentimentScore <- sum(PositiveWordMatches) - sum(NegativeWordMatches)
  return(SentimentScore)
}, PositiveWordList, NegativeWordList, .progress=.progress)
TweetScoredf <- data.frame(score=TweetScore, text=OurSentence)
return(TweetScoredf)
}


#Loading the positive words wordlist using the scan() function
pos <- scan('E:/UNCC/Survey of Programming Languages/Project/positive-words.txt',
           what='character', comment.char=';')


#Loading the negative words wordlist using the scan() function
neg <- scan('E:/UNCC/Survey of Programming Languages/Project/negative-words.txt',
           what='character', comment.char=';')
```

```
#as.factor convert the vector into factor
#the text along with the positive and negative word list is sent to score.sentiment() function
#as arguments
ReadFile <- read.csv("#ManchesterCity Proj.csv")
ReadFile$text <- as.factor(ReadFile$text)
TweetScore <- SentimentScoreFunction(ReadFile$text, pos, neg, .progress='text')
write.csv(TweetScore, file=
          paste("#ManchesterCity", 'scores.csv'),
        row.names=TRUE)


#as.date() converts the date of creation into the date format.
#mutate() adds new variables while preserving the old variables.
#In our case, the tweet score variable is added preserving the existing creation date variable.
#tweet score -> greater than 1 indicates a positive tweet, less than 1 indicates a negative tweet,
#and 0 indicates a neutral tweet. This is implemented using the if-else function.
Statistic <- TweetScore
Statistic$created <- ReadTweetFile$created
Statistic$created <- as.Date(Statistic$created)
Statistic <- mutate(Statistic, tweet=ifelse(Statistic$score > 0, 'positive',
                             ifelse(Statistic$score < 0, 'negative', 'neutral')))


#group_by() function combines the arguments into a dataframe.
#dplyr::summarise collapses the dataframe created into a single row.
#This information is exported to a CSV file using the write.csv function.
Tweet <- group_by(Statistic, tweet, created)
Tweet <- dplyr::summarise(Tweet, number=n())
write.csv(Tweet, file=paste("#ManchesterCity", 'opin.csv'), row.names=TRUE)
```

```
#A plot graph based on the tweet sentiment score
ggplot(Tweet, aes(created, number)) + geom_line(aes(group=tweet, color=tweet), size=2) +
  geom_point(aes(group=tweet, color=tweet), size=4) +
  theme(text = element_text(size=18), axis.text.x = element_text(angle=90, vjust=1)) +
  ggtitle("#ManchesterCity")


#A qplot graph based on the tweet sentiment score
qplot(TweetScore$score, xlab = "Score")


#A histogram based on the tweet sentiment score
hist(TweetScore$score, main = 'Histogram of Sentiment Score')


#Calculating the number of negative, positive and neutral tweets
opin <- read.csv(file = "#ManchesterCity opin.csv")

i <- 0

total_negative <- 0
total_positive <- 0
total_neutral <- 0

for(i in 1:length(opin$tweet))
{
  if(opin$tweet[i] == "negative")
  {
    total_negative <- total_negative + opin$number[i]
  }
  else if(opin$tweet[i] == "positive")
  {
    total_positive <- total_positive + opin$number[i]
  }
  else
  {
    total_neutral <- total_neutral + opin$number[i]
  }

}



total_negative
total_positive
total_neutral



#A 3D piechart based on the tweet sentiment score
library(plotrix)
slices <- c(total_negative, total_neutral, total_positive)
lbls <- c("Negative", "Neutral", "Positive")
pie3D(slices, labels = lbls,  explode=0.1,
      main="Pie Chart of Tweet Sentiment
      Analysis")
```

Conclusion

The task of sentiment analysis is still in the developing stage in general and is nowhere near completion. This is especially so in the sphere analysis from micro-blogging sites. For this project, we have worked with the simple algorithms and have compared the adjectives in the tweet text with a pre-defined positive and negative word list datasets and have indicated if the tweet is a positive tweet, negative tweet or a neutral tweet.

We presented results for sentiment analysis on Twitter. We conclude from the analysis of our tweets that in general, the majority of the tweeters have taken a neutral stance about the English soccer club Manchester City (we have used the search term #ManchesterCity), since statistically the number of neutral tweets are almost double of the negative and positive tweets as evident from the table above. We also tentatively conclude that sentiment analysis for Twitter data is not that different from sentiment analysis for other genres.

We could take this a notch higher in the future by analysing sarcasm, since we cannot conclude hastily that a tweet is a positive tweet, just because the majority of the words are positive, since they could have been used in a sarcastic sense as well. Another aspect that could be dealt with would be the analysis of emoticons or smileys for better understanding of the tweet. Understanding of machine learning algorithms and richer linguistic algorithms would be a core requirement for this.

References

Research papers
[1] Sentiment Analysis of Twitter Data
by Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow and Rebecca Passonneau
from Department of Computer Science Columbia University New York, NY 10027 USA

Online resources
[2] Twitter Sentiment Analysis By Afroze Ibrahim Baqapuri
https://arxiv.org/ftp/arxiv/papers/1509/1509.04219.pdf

[3] Twitter analysis by Kaify Rais
https://www.slideshare.net/ajayohri/twitter-analysis-by-kaify-rais

[4] R-bloggers – Twitter Sentiment Analysis
https://www.r-bloggers.com/twitter-sentiment-analysis-with-r/

[5] GitHub – Twitter Sentiment Analysis
https://github.com/Twitter-Sentiment-Analysis/R/

[6] Wordpress – Twitter Analysis
https://bigdataenthusiast.wordpress.com/category/twitter-data-analytics-using-r/

[7] Wikipedia – Sentimental Analysis
https://en.wikipedia.org/wiki/Text_mining#Sentiment_analysis

[8] YouTube
https://www.youtube.com/watch?v=QETCjkQ3CBw
https://www.youtube.com/watch?v=lT4Kosc_ers