

Sentimental Analysis on Twitter Data using R Programming

Pooja Umathe

Master's in Data Science

Saint Peter's University, Jersey City, NJ-07306

Introduction:

Our personal life is highly dependent on the technology that is being developed over the years. Technology has advanced with years and it has changed the way of thinking using the Data Science techniques as well. This project is based on sentimental analysis performed on Twitter data relevant to United Airlines using R programming. Twitter is a popular social platform for expressing our emotions, activities and popular for housing a massive amount of information around the web. In addition, Twitter can also be an amazing open mine for text and social web analysis. The Twitter API is simply a set of URLs that take parameters. Those URLs allows access to many features of Twitter, such as posting a tweet or finding tweets of specific person or specific topic that contain a word, etc. This project utilizes Twitter API to extract Tweets to analyze and filter the data in required form [1].

Sentimental analysis (opinion mining or emotion AI) refers to the use of natural language processing, text analysis to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to voice of the customer materials such as reviews and survey responses, online and social media. This process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude is positive, negative, or neutral towards a topic, product, etc. [2]

Method:

To perform analysis on Twitter data we need to extract data from Twitter using Twitter API. Initial step is to create a Twitter application, this application will allow you to perform analysis by connecting R console to the Twitter using the Twitter API.

To create new Twitter application the user has to login into their respective Twitter account and create a new application. After the Twitter application is created the user will be granted the customer and access keys and tokens. R and Twitter can be connected by saving all these keys and tokens for Twitter authentication to extract the tweets. R is a programming language and software environment intended for deep statistical computing and graphics. It is open source and available across different platforms, e.g., Windows, Mac, Linux. It is now used in a variety of applications including visualizations and data mining. One of the keys to R's explosive growth has been its densely populated collection of extension software libraries, known in R terminology as packages, supplied and maintained by R's extensive user community. Each package extends the functionality of the base R language and core packages, in addition to the functions and data it must include documentation and examples, often in the form of vignettes demonstrating the use of the package [3].

For this project we need some specific packages and libraries. **ROAuth** provides an interface to the OAuth 1.0 specification, allowing users to authenticate via OAuth to the server of their choice. **Twitter** provides an interface to the Twitter web API. The **NLP** package provides a set of classes and functions for NLP which

are used widely by other packages in R. The text mining **package** (tm) and the **word cloud** generator **package** (wordcloud) are available in R for helping us to analyze texts and to quickly visualize the keywords as a **word cloud**.

After installation and loading packages we will need the authentication for Twitter data in R. By using those keys and tokens which we generated during creating application we will get access to the Twitter data by defining in R programming. To look for the data relevant to the United Airlines use hash tag and search Twitter library so that it will search only specific data which we are looking for. Once we get the data we need to extract that data into the data frame so that we can understand the data clearly and can analyze easily. After that we can display or view our data by using view command in R so that we can see the data for our understanding. Raw data won't give good results because generally raw data has some missing values, repeated data, a lot of special characters and unnecessary data which we would not require. Hence it becomes extremely important to pre-process this data or clean this data and then we can continue with our analysis. So, for data preprocessing it must be converted to lower case, replace the blank spaces, missing values, and replace the usernames and removed punctuations, links, tabs. The removal of stop words in the text mining is also important. Stop words are basically a set of commonly used words in any language, not just English. The reason why stop words are critical to many applications is that, if we remove the words that are very commonly used in a given language, we can focus on the important words instead.

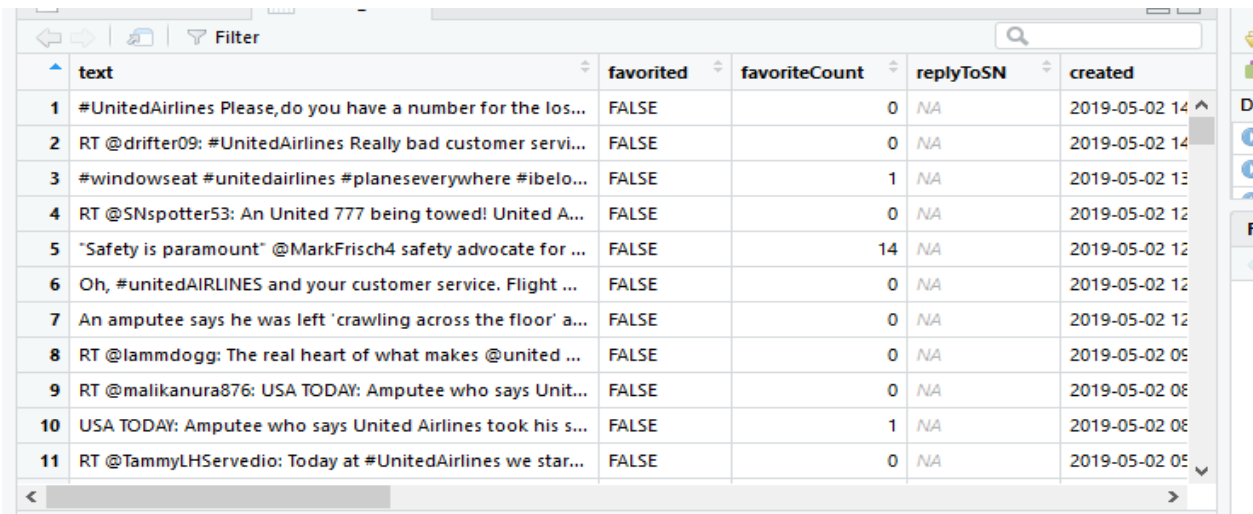
A **text corpus** is a large and unstructured set of texts used to do statistical analysis and hypothesis testing, checking occurrences or validating linguistic rules within a specific language territory. So, for the stop words I have used text.corpus in this project. To build document term matrix and the transpose of that matrix I have used **TM** package which represents the collection of text documents. If we have most frequent words then we can analyze anything very easily. I have used function **FindfreqTerms** to find most frequent 50 words. So that all frequent words related to United Airlines will come out. Data Visualization plays an important role in the data analysis. It is the graphical representation of information and data. By plotting the graphs or plots we can understand our results very clearly and also, we can determine our specific tasks using visualization techniques. I have used **ggplot2** package and library to plot the graphs. To generate most frequent words of United Airlines I have plotted bar plot which shows the graph of all frequent words. Text mining methods allow us to highlight the most frequently used keywords in a paragraph of texts. One can create a word cloud, also referred as text cloud or tag cloud, which is a visual representation of text data. For the Sentimental analysis I have used **get_nrc_sentiment** function which is used to get emotions and valence from NRC in built Dictionary to calculate the presence of eight different emotions and their corresponding valence in a text file. The ten columns are as follows: "anger", "anticipation", "disgust", "fear", "joy", "sadness", "surprise", "trust", "negative", "positive." with sentiment analysis from a text analytics point of view, we are essentially looking to get an understanding of the attitude of a writer with respect to a topic in a piece of text and its polarity; whether it's positive, negative or neutral. So, for our understanding I have calculated score of each sentiment which is very helpful to understand the emotions and also, I have plotted the bar graph with the total scores of each emotions which is shown in the results.

Results:

The first screenshot (Figure 1) below shows the authentication step. To access Twitter data we need authentication by Twitter using Twitter API. The next screenshot (Figure 2) below shows how the data looks like when we get tweets from Twitter. Third screenshot (Figure 3) shows document term matrix and its transpose in which 100% sparsity was received and non-sparse entries were 10608 from 2274392 entries for 1000 documents and 2285 terms. The most frequent 50 words are shown below which belong to United Airlines data and which has helpful for the analysis (Figure 4).

```
> setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)
[1] "Using direct authentication"
> tweets <- searchTwitter("#UnitedAirlines", n=1000, lang = "en")
> tweets_UN <- twListToDF(tweets)
> View(tweets_UN)
```

Figure 1



	text	favorited	favoriteCount	replyToSN	created
1	#UnitedAirlines Please, do you have a number for the los...	FALSE	0	NA	2019-05-02 14
2	RT @drifter09: #UnitedAirlines Really bad customer servi...	FALSE	0	NA	2019-05-02 14
3	#windowseat #unitedairlines #planeseverywhere #ibelo...	FALSE	1	NA	2019-05-02 13
4	RT @SNspotter53: An United 777 being towed! United A...	FALSE	0	NA	2019-05-02 12
5	"Safety is paramount" @MarkFrisch4 safety advocate for ...	FALSE	14	NA	2019-05-02 12
6	Oh, #unitedAIRLINES and your customer service. Flight ...	FALSE	0	NA	2019-05-02 12
7	An amputee says he was left 'crawling across the floor' a...	FALSE	0	NA	2019-05-02 12
8	RT @lammdogg: The real heart of what makes @united ...	FALSE	0	NA	2019-05-02 05
9	RT @malikanura876: USA TODAY: Amputee who says Unit...	FALSE	0	NA	2019-05-02 08
10	USA TODAY: Amputee who says United Airlines took his s...	FALSE	1	NA	2019-05-02 08
11	RT @TammyLHServedio: Today at #UnitedAirlines we star...	FALSE	0	NA	2019-05-02 05

Figure 2

```
> dtm_UN <- DocumentTermMatrix(UN_tweets.text.corpus)
> dtm_UN
<<DocumentTermMatrix (documents: 1000, terms: 2285)>>
Non-/sparse entries: 10608/2274392
Sparsity : 100%
Maximal term length: 60
Weighting : term frequency (tf)
> tdm_UN <- TermDocumentMatrix(UN_tweets.text.corpus)
> tdm_UN
<<TermDocumentMatrix (terms: 2284, documents: 1000)>>
Non-/sparse entries: 10606/2273394
Sparsity : 100%
Maximal term length: 60
Weighting : term frequency (tf)
```

Figure 3

```
> # Most frequent words
> (freq.terms <- findFreqTerms(tdm_UN, lowfreq = 50))
[1] "united" "unitedairlines" "francisco" "international" "san" "comparison"
[7] "new" "united's" "..." "first" "737800" "arriving"
[13] "gate" "houston" "iah" "livery" "n37267" "flight"
[19] "airlines" "avgeek" "boeing..." "taxi" "time" "behind"
[25] "beingunited" "avgeek..." "767" "last" "night" "pushing"
```

Figure 4

The bar graph below (Figure 5) shows all the most frequent words. From this graph we can see which word is repeated how many times. This graph was very helpful to find the most frequent words in numbers so that we can analyze more things related to United Airlines.

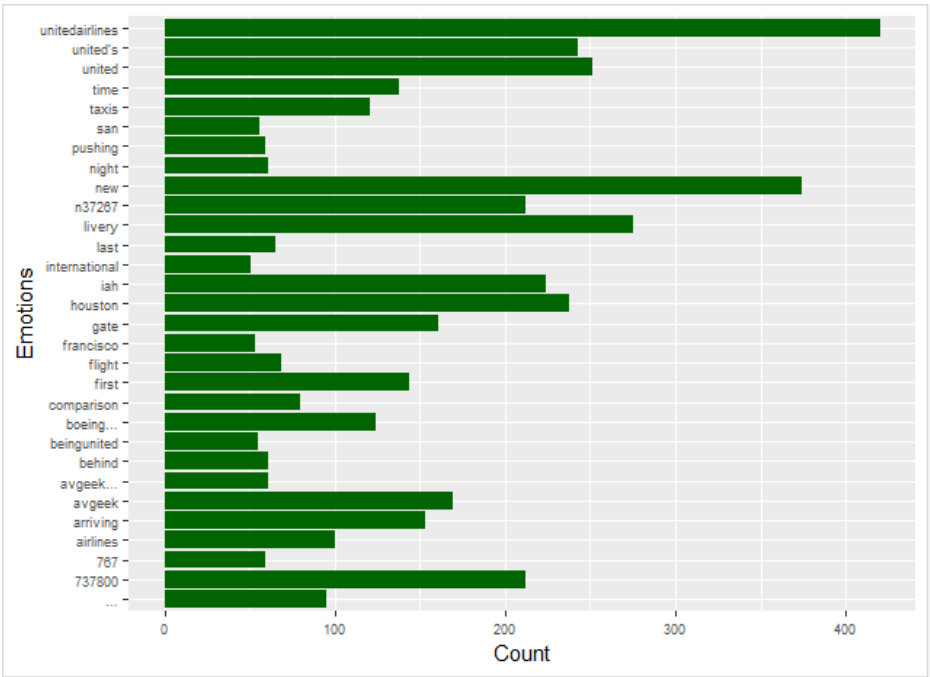


Figure 5

The word cloud below (Figure 6) shows the most frequent words from our data. If the word in the word cloud is bolder and more big then it is considered as most repeated word and vice-versa. As stated in the methods above sentiment analysis is always very helpful to find positive, negative or neutral comments; in our sample data on United Airlines the same is achieved by using sentiments count and get_nrc_sentiment function. By calculating scores of sentiments, we can see (Figure 6) what is the count of each emotion and the graph (Figure 7) using ggplot2 library shows those exact number of emotions.

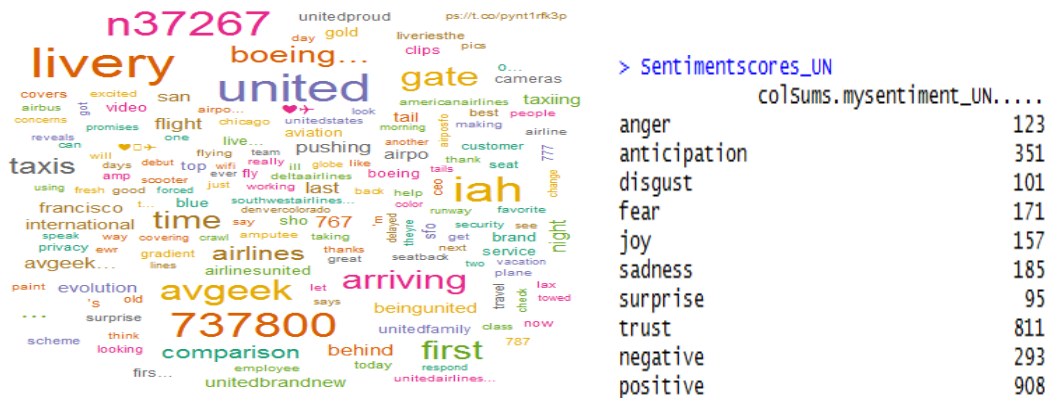


Figure 6

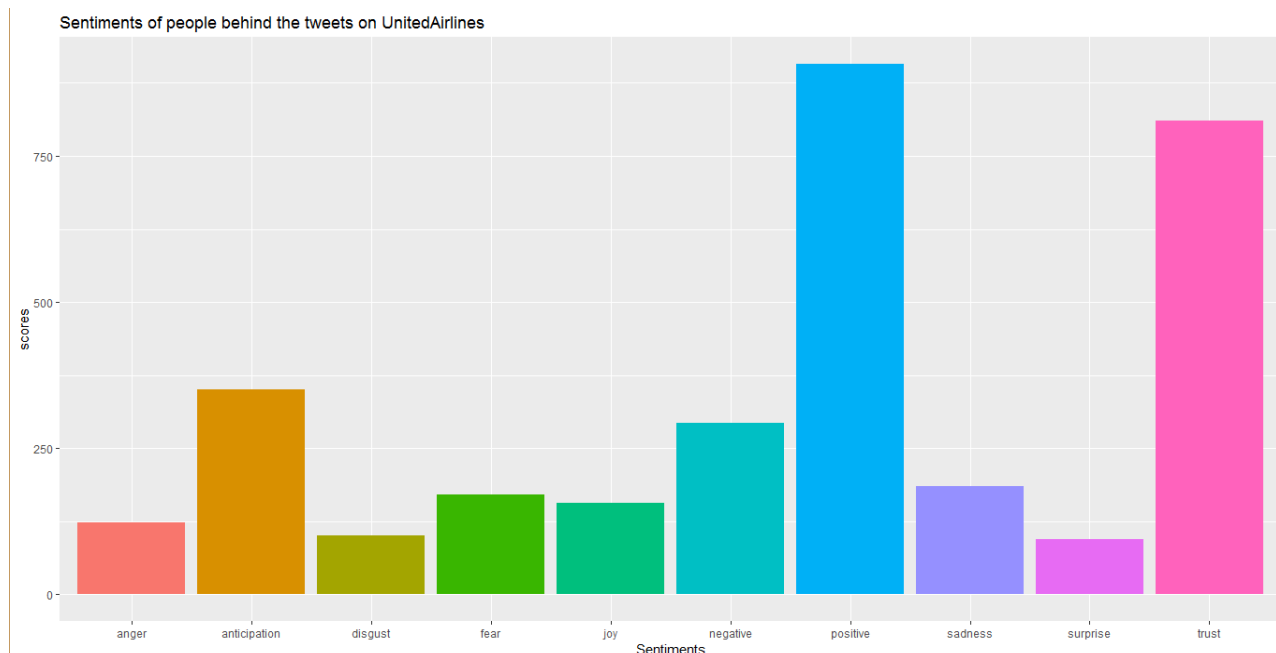


Figure 7

From the above graph we can say that highest number of tweets are positive and then trust, means so many people are happy with United Airlines services. Negative tweets about United Airlines are not more as compare to the positive tweets.

So, we can conclude that most of the people are happy but the anticipation rate is also not good as it shows above 300 so we need to ask people about their expectation by using feedback forms or survey. From these techniques we can improve United Airlines services.

References:

- [1] Amir Hossein Akhavan Rahnama, “*Distributed Real-Time Sentiment Analysis for Big Data Social Streams*” <https://arxiv.org/ftp/arxiv/papers/1612/1612.08543.pdf>
- [2] Ashish Katrekar AVP, Big Data Analytics , “*An introduction to Sentimental Analysis*” <https://globallogic.com/wp-content/uploads/2014/10/Introduction-to-Sentiment-Analysis.pdf>
- [3] W. N. Venables, D. M. Smith and the R Core Team, “*An Introduction to R*” <https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf>

Code:

```
UnitedAirlines.R* x
Source on Save

1 # Install packages
2
3 install.packages("ROAuth")
4
5 install.packages("twitter")
6 library(twitter)
7
8 install.packages("NLP")
9 library("NLP")
10
11 install.packages("tm")
12 library(tm)
13
14 install.packages("tmap")
15 library(tmap)
16
17 install.packages("syuzhet")
18 library("syuzhet")
19
20 install.packages("snowballc")
21 library("snowballc")
22
23 library("stringi")
24
25 install.packages("topicmodels")
26 library("topicmodels")
27
28 install.packages("wordcloud")
29 library("wordcloud")
30
31 install.packages("ggplot2")
32 library("ggplot2")
33
34

# Defining keys to access twitter data

consumer_key <- "xxxxxxx"
consumer_secret <- "xxxx"
access_token <- "xxxx"
access_secret <- "xxxx"

# Getting authentication from twitter to access data

setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)

# Getting Data

tweets <- searchTwitter("#UnitedAirlines", n=1000, lang = "en")

# Converting this extracted data to a dataframe

tweets_UN <- twListToDF(tweets)

#Displaying tweets

view(tweets_UN)

# Data Preprocessing

UN_text<- tweets_UN$text
```

```

#convert all text to lower case
UN_text<- tolower(UN_text)

# Replace blank space ("rt")
UN_text <- gsub("rt", "", UN_text)

# Replace @UserName
UN_text <- gsub("@\\w+", "", UN_text)

# Remove punctuation
UN_text <- gsub("[[:punct:]]", "", UN_text)

# Remove links
UN_text <- gsub("http\\w+", "", UN_text)

# Remove tabs
UN_text <- gsub("[ \\t]{2,}", "", UN_text)

103 # Remove blank spaces at the beginning
104
105 UN_text <- gsub("^ ", "", UN_text)
106
107 # Remove blank spaces at the end
108
109 UN_text <- gsub(" $", "", UN_text)
110
111 #create corpus
112
113 UN_tweets.text.corpus <- Corpus(VectorSource(UN_text))
114
115 #clean up by removing stop words
116
117 UN_tweets.text.corpus <- tm_map(UN_tweets.text.corpus, function(x)removewords(x,stopwords()))
118
119 # The document term matrix
120
121 dtm_UN <- DocumentTermMatrix(UN_tweets.text.corpus)
122 dtm_UN
123
124 # The transpose of the document term matrix
125
126 tdm_UN <- TermDocumentMatrix(UN_tweets.text.corpus)
127 tdm_UN
128
129
130 # Most frequent words
131 (freq.terms <- findFreqTerms(tdm_UN, lowfreq = 50))
132

```

```

# Most frequent words
(freq.terms <- findFreqTerms(tdm_UN, lowfreq = 50))

term.freq <- rowSums(as.matrix(tdm_UN))
term.freq <- subset(term.freq, term.freq >= 50)
df2 <- data.frame(term = names(term.freq), freq = term.freq)
ggplot(df2, aes(x=term, y=freq)) + geom_bar(stat="identity", fill="dark green")
+ xlab("Emotions") + ylab("count") + coord_flip() + theme(axis.text=element_text(size=7))

# generating wordcloud
wordcloud(UN_tweets.text.corpus,min.freq = 10,colors=brewer.pal(8, "Dark2"),
          random.color = TRUE,max.words = 500)

#getting emotions
mysentiment_UN<-get_nrc_sentiment((UN_text))
mysentiment_UN

#calculating total score for each sentiment
Sentimentscores_UN<-data.frame(colSums(mysentiment_UN[,]))
Sentimentscores_UN
names(Sentimentscores_UN)<-"Score"
Sentimentscores_UN<-cbind("sentiment"=rownames(Sentimentscores_UN),Sentimentscores_UN)
rownames(Sentimentscores_UN)<-NULL

#plotting the sentiments with scores
ggplot(data=Sentimentscores_UN,aes(x=sentiment,y=Score))+geom_bar(aes(fill=sentiment),
                                                                    stat = "identity")+
  theme(legend.position="none")+
  xlab("Sentiments")+ylab("scores")+ggtitle("Sentiments of people behind the tweets on UnitedAirlines")

```