Twiiter Data Analysis - Lab Project

(Faraz Ali - 172074), (Mohsin Ali - 170409) & (Hamza Nadeem - 170353)

6/25/2021

Introduction:

Social media warfare termed 5th generation warfare describes the use of social media as a kind of weapon intending to cause lasting damage to certain actors such as governments or companies. Various strategies and tactics, as well as technological means, are used to push through a political, economic, social, or cultural agenda.

Pakistan being the victim of this 5th gen war for a long time. Last year EU Disinfo lab busted Indian propaganda media outlets operating across 65 countries the main purpose of them is to discredit Pakistan in any means possible.

Our project is to analyze Twitter trends, extract negative tweets from them and see the location of those persons if they are from Pakistan or not. We applied this on-trend #HafizSaeed which was trending after the recent bombings in Johar Town, Lahore. The results we got were astonishing, more than 90% of negative/Anti-Pakistan tweets are from Indian states.

Loading the packages and connecting with twitter

```
library(twitteR)
library(ROAuth)
library(tm)
library(SnowballC)
library(RColorBrewer)
library(ggplot2)
library(wordcloud)
library(graph) #For Network of terms
library(Rgraphviz) #For Network of terms
library(topicmodels) #Topic Modelling
library(data.table)
library(devtools)
library(sentiment)
setup_twitter_oauth("weOUVCVse5yNZ0RMq21sTHCPh",
                    "WnVpnc1e9GIEcrUc5Gz45Edel49oXbqGqzD8zQyx6eyGB9e6R6",
                    "1100733914291818496-3XskjnyklycO6FxQA02nzthI3wPcvp",
                    "9uf22j37zdiFFYNYa7taFcOOPg8S5I7lI5sBubCTNXP4e")
```

```
[1] “Using direct authentication”
```

Getting Tweets of #hafizsaeed

```
#Getting tweets
#tweets<-searchTwitter("#hafizsaeed", lang = "en", n=3200)
#Getting length of tweets
#(n.tweet<-length(tweets))
#Converting tweets to dataframe
#tweets.df <- twListToDF(tweets)
#write_as_csv(tweets.df, "twitterData.csv", prepend_ids = TRUE, na = "",
fileEncoding = "UTF-8")

tweets.df<-read.csv("twitterData.csv")
tweets.df[1, c("id", "created", "screenName", "replyToSN", "favoriteCount",
"retweetCount", "longitude", "latitude", "text")]

       id         created screenName replyToSN favoriteCount retweetCount
```

1 1.40806e+18 6/24/2021 13:37 emgreat 0 3 longitude latitude 1 NA NA text 1 RT @NaveedKBhatti: Well we all know who was behind the #JoharTown blast. We all know for whom #hafizsaeed is the number one target. I persoâ\200¦

```
#Wrapping the tweet text
writeLines(strwrap(tweets.df$text[1], 06))
```

RT @NaveedKBhatti: Well we all know who was behind the #JoharTown blast. We all know for whom #hafizsaeed is the number one target. I persoâ€¦

Text Cleaning

```
#Getting only the text element from the dataframe that we got
#through twitter API
data<-tweets.df$text
#Creating a copy of data
tweetsCopy <- data
#We are creating corpus to perform some computations
myCorpus <- Corpus(VectorSource(data))
#Converting data to lower case

myCorpus <- tm_map(myCorpus, tolower)
#Creating function which will be used to remove URLs from the data
removeURL <- function(x) gsub("http[^[:space:]]*", "", x)
#Now by using the above function we are removing the URLs and getting
#the data back in myCorpus var
myCorpus <- tm_map(myCorpus, content_transformer(removeURL))

#Creating another function which will remove the letters other than english
#letters or spaces
removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]*", "", x)
#Applying above function on the data
myCorpus <- tm_map(myCorpus, content_transformer(removeNumPunct))
```

```r
#Removing all the numbers
myCorpus <- tm_map(myCorpus,removeNumbers)

#Now applying the above words and remove them from the data
myCorpus <- tm_map(myCorpus, removeWords, stopwords('english'))

#Removing extra spaces from the data
myCorpus <- tm_map(myCorpus, stripWhitespace)
#Removing punctuation from the data
myCorpus <- tm_map(myCorpus, removePunctuation)
#Creating a copy for later use
myCorpusCopy <- myCorpus
#Inspecting first three tweets
inspect(myCorpus[1:3])
```

<> Metadata: corpus specific: 1, document level (indexed): 0 Content: documents: 3

[1] rt naveedkbhatti well know behind johartown blast know hafizsaeed number one target persoâ
[2] rt sheetpak fif falahe insaniat foundation setup hafizsaeed pakarmy person consoling jud chief hafizsaeed cousin jubair afterâ [3] rt mahnag wish bought âtraitorsâ india enough enough btw hafizsaeed home blast yesterday isi work related tâ

Stemming and Stem Completion

```r
#removing the commoner morphological and inflectional endings from
#words in English using stemDocument which works on porter's stemming
#algorithm
myCorpus <- tm_map(myCorpus, stemDocument)
#Getting the first latest resultant tweet after stemming
writeLines(strwrap(myCorpus[[1]]$content, 60))
```

rt naveedkbhatti well know behind johartown blast know hafizsae number one target
persoâ

Fixing issues of Stemming

```r
wordFreq <- function(corpus, word) {
results <- lapply(corpus,
function(x) { grep(as.character(x), pattern=paste0("nn<",word)) })
sum(unlist(results))
}

replaceWord <- function(corpus, oldword, newword) {
tm_map(corpus, content_transformer(gsub),
pattern=oldword, replacement=newword)
}
```

```
myCorpus <- replaceWord(myCorpus, "hafizsae", "Hafiz Saeed")
myCorpus <- replaceWord(myCorpus, "hafizsa", "Hafiz Saeed")
myCorpus <- replaceWord(myCorpus, "hafizs", "Hafiz Saeed")
myCorpus <- replaceWord(myCorpus, "secur", "security")
myCorpus <- replaceWord(myCorpus, "peopl", "people")
myCorpus <- replaceWord(myCorpus, "paxstan", "pakistan")
myCorpus <- replaceWord(myCorpus, "forc", "force")
myCorpus <- replaceWord(myCorpus, "armi", "army")
myCorpus <- replaceWord(myCorpus, "lahor", "lahore")
myCorpus <- replaceWord(myCorpus, "lahr", "lahore")
myCorpus <- replaceWord(myCorpus, "polic", "police")
myCorpus <- replaceWord(myCorpus, "countri", "country")
myCorpus <- tm_map(myCorpus, removeWords, 'hafizsaeed')
```

Building Term Document Matrix

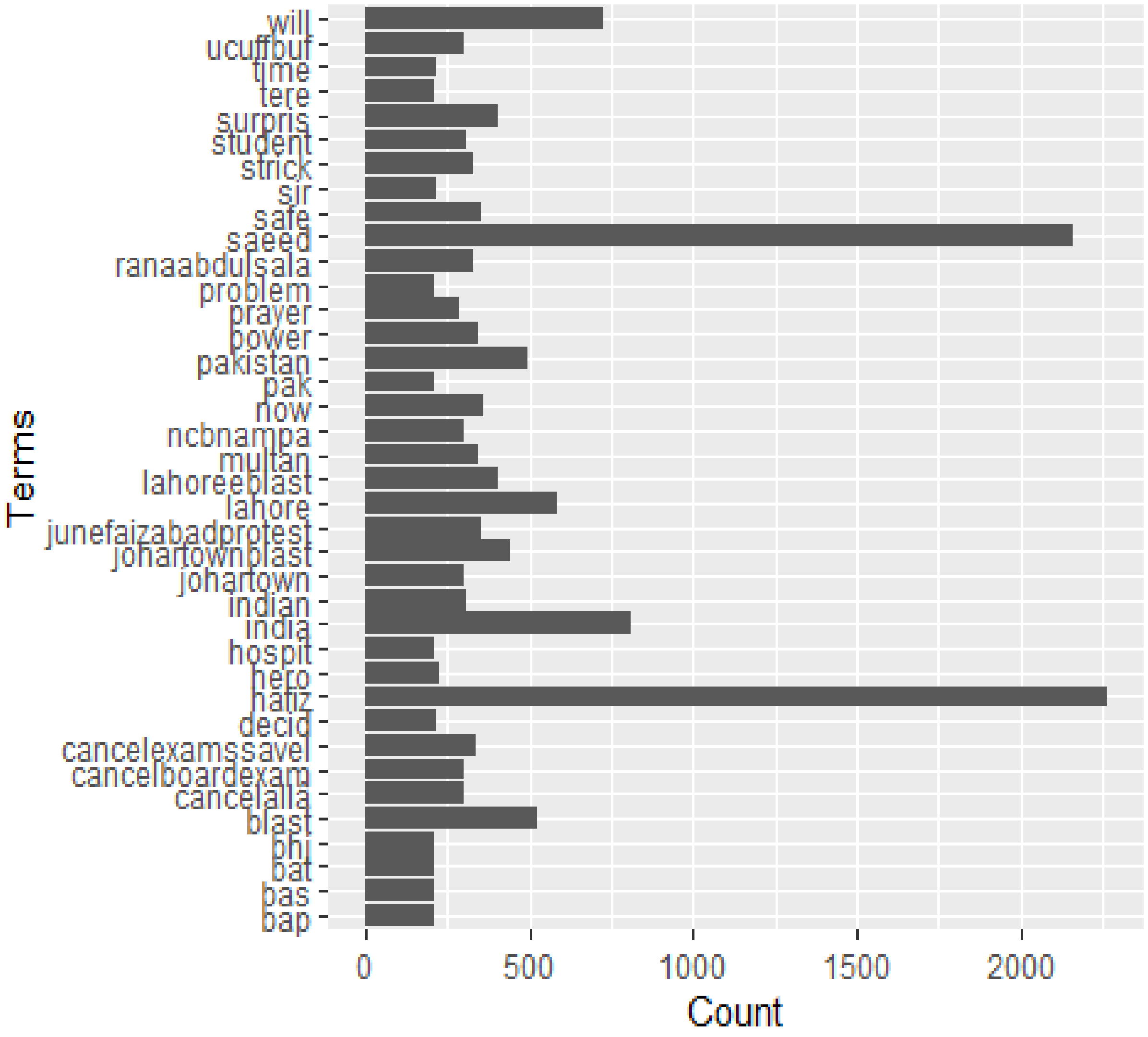
```
#Converting Unstructured data to structured data using TDM
tdm <- TermDocumentMatrix(myCorpus)
tdm
```

```
<<TermDocumentMatrix (terms: 2364, documents: 3199)>> Non-/sparse entries:
36962/7525474 Sparsity : 100% Maximal term length: 41 Weighting : term frequency (tf)
```

```
#The words with frequency more that 150
(freq.terms <- findFreqTerms(tdm, lowfreq = 200))
```

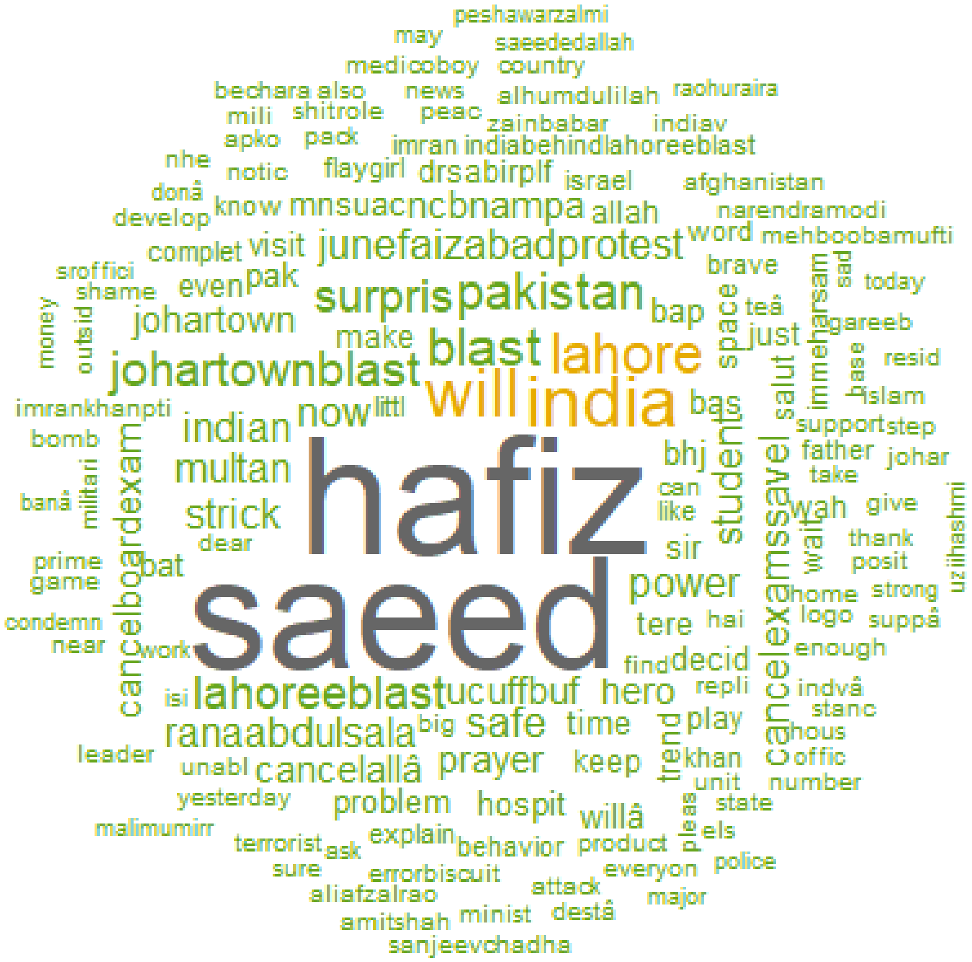
```
[1] “blast” “hafiz” “johartown”
[4] “saeed” “india” “now”
[7] “johartownblast” “lahoreeblast” “hero”
[10] “pakistan” “cancelallâ” “cancelboardexam”
[13] “cancelexamssavel” “junefaizabadprotest” “multan”
[16] “ncbnampa” “power” “ranaabdulsala”
[19] “strick” “student” “ucuffbuf”
[22] “bap” “bas” “bat”
[25] “bhj” “decid” “lahore”
[28] “problem” “surpris” “tere”
[31] “time” “will” “indian”
[34] “safe” “hospit” “prayer”
[37] “pak” “sir”
```

```
#Rowsumming and getting only words having more than 10 frequency
term.freq <- rowSums(as.matrix(tdm))
term.freq <- subset(term.freq, term.freq >= 200)
df <- data.frame(term = names(term.freq), freq = term.freq)
#Now plotting
ggplot(df, aes(x=term, y=freq)) + geom_bar(stat="identity") +
  xlab("Terms") + ylab("Count") + coord_flip()
```



Wordcloud

```
w <- as.matrix(tdm)
#Calculating the frequency of words and sort it by decreasing order
#of frequency
word.freq <- sort(rowSums(w), decreasing = T)
#Creating color for wordcloud
pal <- brewer.pal(8, "Dark2")[-(1:4)]
#Creating word cloud
wordcloud(words = names(word.freq), freq = word.freq,
          min.freq = 50, random.order = F, colors = pal)
```


peshawarzalmi
may
saeededallah
medicoboy
country
bechara
also
news
alhumdulilah
raohuraira
mili
shitrole
peac
zainbabar
indiav
apko
pack
imran
indiabehindlahoreeblast
nhe
notic
flaygirl
drsabirplf
israel
afghanistan
donâ
know
mnsuacncbnampa
allah
narendramodi
develop
complet
visit
junefaizabadprotest
word
mehboobamufti
sroffici
brave
shame
even
pak
surpris
pakistan
harsam
sad
today
money
outsid
johartown
bap
space
teâ
gareeb
make
just
johartownblast
blast
lahore
resid
base
imme
imrankhanpti
will
india
bas
salut
islam
littl
now
indian
bomb
support
step
bhj
father
johar
militari
multan
student
take
banâ
can
savel
give
like
wah
strick
wait
thank
uzilhashmi
dear
hafiz
sir
prime
posit
bat
power
game
home
strong
cancelboardexam
condemn
tere
hai
suppâ
logo
near
saeed
cancelexams
work
find
decid
enough
isi
lahoreeblast
ucuffbuf
hero
repli
indvâ
stanc
ranaabdulsala
big
safe
time
play
trend
hous
leader
keep
khan
offic
unabl
cancelallâ
prayer
unit
number
yesterday
problem
hospit
state
willâ
pleas
malimumirr
els
explain
terrorist
ask
behavior
product
police
sure
errorbiscuit
everyon
aliafzalrao
attack
major
amitshah
minist
destâ
sanjeevchadha

Topic Modelling

```
dtm <- as.DocumentTermMatrix(tdm)
rowTotals <- apply(dtm , 1, sum)
dtm.new   <- dtm[rowTotals> 0, ]
lda <- LDA(dtm.new, k = 6) #Finding 8 topics from the tweets data
term <- terms(lda,5) #And each topic will have first 7 terms
(term <- apply(term, MARGIN = 2, paste, collapse=", "))

                                                           Topic 1
                           "hafiz, saeed, pakistan, khan, support"
                                                           Topic 2
                             "pakistan, play, hafiz, saeed, india"
                                                           Topic 3
                               "will, now, surpris, india, lahore"
                                                           Topic 4
                       "hafiz, saeed, johartownblast, safe, india"
                                                           Topic 5
                          "hafiz, saeed, blast, lahore, yesterday"
                                                           Topic 6
```

“junefaizabadprotest, multan, cancelexamssavel, strick, ranaabdulsala”

```
topics <- topics(lda) # 1st topic identified for every document (tweet)
```

Sentiment Analysis

```
sentiments <- sentiment(tweets.df$text)
#Checking polarity of tweets
table(sentiments$polarity)
```

negative neutral positive 168 2467 564

```
#Plotting Sentiments
sentiments$score <- 0 #Assigning scores
sentiments$score[sentiments$polarity == "positive"] <- 1
sentiments$score[sentiments$polarity == "negative"] <- -1
sentiments$user <- tweets.df$screenName
#Now combining
abc<-tweets.df$screenName
df <- do.call(rbind.data.frame, Map('c', sentiments$polarity,
                                    tweets.df$text,
                                    abc))
colnames(df)[1] <- "A"
colnames(df)[2] <- "B"
colnames(df)[3] <- "C"
fd <- df[df$A != "neutral", ];
fd <- fd[fd$A != "positive", ];
```

Preprocessing of Negative Tweets

```
data<-fd$B
myCorpus <- Corpus(VectorSource(data))
myCorpus <- tm_map(myCorpus, tolower)
removeURL <- function(x) gsub("http[^[:space:]]*", "", x)
myCorpus <- tm_map(myCorpus, content_transformer(removeURL))
removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]*", "", x)
myCorpus <- tm_map(myCorpus, content_transformer(removeNumPunct))
myCorpus <- tm_map(myCorpus,removeNumbers)
myCorpus <- tm_map(myCorpus, removeWords, stopwords('english'))
myCorpus <- tm_map(myCorpus, stripWhitespace)
myCorpus <- tm_map(myCorpus, removePunctuation)

tdm <- TermDocumentMatrix(myCorpus)

dtm <- as.DocumentTermMatrix(tdm)
rowTotals <- apply(dtm , 1, sum)
dtm.new   <- dtm[rowTotals> 0, ]
lda <- LDA(dtm.new, k = 7) #Finding 7 topics from the tweets data
term <- terms(lda,5) #And each topic will have first 5 terms
(term <- apply(term, MARGIN = 2, paste, collapse=", "))

                              Topic 1
   "enough, blast, home, bought, btw"
                              Topic 2
```

```
"hafizsaeed, kill, alive, pig, wants"
                              Topic 3
   "attack, news, blast, hafiz, like"
                              Topic 4
"hafiz, saeed, nightmare, usa, india"
                              Topic 5
```

```
“india, hafizsaeed, ever, fail, still” Topic 6 “pak, home, work, accident, doct” Topic 7
“hafizsaeed, bodyguard, one, people, sad”
```

```
topics <- topics(lda) # 1st topic identified for every document (tweet)

sentiments <- sentiment(fd$B)
fd$C
```

```
[1] “RaviSha82603152” “ADShah30” “Subhash86894697” “chaitanya77W”
[5] “rohit_rls143_” “kaushik_2212” “deepakb14” “rinkugueria”
[9] “SBudhiraju” “MamataNath” “andhagadha” “MmkChitn”
[13] “Narende05395811” “azadbhakt57” “Sarthak50599594” “Krish74357931”
[17] “nyadavooo” “Sanjay_Dixit” “AkshayB93093403” “1frm90Migration” [21] “sree1832”
“LiberalBhakt2” “I_AmJust_Myself” “arifaajakia”
[25] “SHIWANSHUSING19” “neerajtiwariin” “Saidgoharkhan” “EagleEyeXD”
[29] “Jasim06808473” “TrainersofPaki1” “RizwanaRashidT” “Prabhjyot_Madan” [33]
“ankurv46” “BharatPrem4” “ramkumariramsu1” “EmmeyMemon”
[37] “MAHMAL_” “WishuDelhi” “iamzamangul” “NosharwanZahid” [41] “lamees86951110”
“iamburhan_n” “Mareeb" "Trishoolam"
[45] "stitchnfitworld" "AMBQ21" "LADHIT2" "nandy_gautam"
[49] "Solanki65887483" "YusufEducation” “Anjna116” “WhyWhoHow1”
[53] “NationalismBorn” “IndianKangana” “reachind_USACAN” “reachind_uk”
[57] “reachind_bharat” “Mahna5G” “EshaButt2298” “azkazaib”
[61] “bharatpremi68” “Hu77682975” “Dr_Sidraaa” “zohaibrehman143” [65] “arnabnsg”
“wasimmughal_” “abdullabhattii” “Hamnatistic50”
[69] “ali_14572” “srisriyash1” “Mstndi1” “pickooo”
[73] “Sunny_Rjpt” “jun_wala” “RahulMahar18” “New44Nothing”
[77] “Iftikha05790412” “MUKESHN74154668” “IshanikaSingh” “HRA_RT”
[81] “jawad_nadeem450” “MianMeh09376312” “RajaAsim555” “saeedah91538405” [85]
“Im_YIG1” “Gulfam_01” “Rayyan84639122” “NaveedNaushad”
[89] “HARISME01972162” “Bird_Of_Jove” “mfaraz_ulislam” “FarooqHKhan”
[93] “Hanzi87” “Khushikbk” “AAKiani1” “Baba_Yaga77”
[97] “anisha90588196” “ChAdnanAkramGu2” “AltafAhmedTarar” “ChaudryAamir5”
[101] “S_Ali_9” “HainMohsin” “saadraza442” “ihsan_PTI1”
[105] “RaheemDawarr” “bhartiyaoz” "one___4" “WorthyMan6”
[109] “Aworrior888” “HaryanaAbhishek” “BHARATIYASEEKER” “salimabhutto”
[113] “kmadhu019” “AsterixdeGaulle” “sumitrana1” “Md_Ayesha_”
[117] “medhparth” “WisdomHous” “nainitalwali” “BasavaraddiNav1” [121] “MahaveerM_”
“ActivistIk” “Tourist2020” “sanaullah304”
[125] “Uroojsayyami” “nehaparam” “S_HA101” “MubashirAli_PVF” [129] “vaddanki001”
```

```
“MHASSAN90750181” “MHASSAN90750181” “PathanAsmakhan” [133] “YKapase”
“pramodwagh22097” “Aaraam_thambran” “DSI786”
[137] “Meerub02699898” “Rumisa36282198” “HariyaneKaChora” “UsamaAnsari825”
[141] “AEOIjazAhmad” “AbuShafaqat” “neelofer23” “Be_Haad_Masoom” [145] “Pti333”
“Inegol66224658” “Jutt678” "Nazish__Says"
[149] “Pthan_1” “pohton_1” “llillvf” “Kashi_712”
[153] “javidtalk” “SR_Official510” “Ali_Mujahid1” “ForeverZindabad” [157] “aruntechone”
“SweetCookiesme” “MunirAh27264442” “saddamkorai”
[161] “Tourist2020” “sanaKas17897461” “ImMeharSam” “ImAsim777”
[165] “BePakistani6” “smes_india” “TangoCharlie108” “ExposeAntiIndia”
```

```
#Checking polarity of tweets
table(sentiments$polarity)
```

```
negative 168
```

Followers Analysis

```
#uniqueOnly<-unique(fd$C)
#uniqueOnly
#library(twitteR)
#library(foreign)
#library(base64enc)
#library(devtools)
#library(raster)
#library(RCurl)
#users <- uniqueOnly
#locations <- list() #Create an empty list to populate
#k<-1
#  for (i in 1:length(users)){
#    start <- getUser(users[i])
#    friends_object <- lookupUsers(start$getFriendIDs())
#    friends_object <- twListToDF(friends_object)
#    #followers_object <- lookupUsers(start$getFriendIDs())
#    #followers_object
#    if(length(friends_object)>0){
#      #friends_object <- twListToDF(friends_object)
#      locationss <- friends_object[[12]]
#      for (j in length(locationss)){
#        locations[k]<-locationss[j]
#        k<-k+1
#    }
#
#  }
#
#}
#length(locations)
#class(locationss)
#abcd<-data.frame(locationss)
```

```
#write_as_csv(abcd, "locations.csv", prepend_ids = TRUE, na = "",
fileEncoding = "UTF-8")
```

location preprocessing

```
location.df<-read.csv("locations.csv")
data<-location.df
#Creating a copy of data
tweetsCopy <- data
#We are creating corpus to perform some computations
myCorpus <- Corpus(VectorSource(data))
#Converting data to lower case

myCorpus <- tm_map(myCorpus, tolower)
#Creating function which will be used to remove URLs from the data
removeURL <- function(x) gsub("http[^[:space:]]*", "", x)
#Now by using the above function we are removing the URLs and getting
#the data back in myCorpus var
myCorpus <- tm_map(myCorpus, content_transformer(removeURL))

#Creating another function which will remove the letters other than english
#letters or spaces
removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]*", "", x)
#Applying above function on the data
myCorpus <- tm_map(myCorpus, content_transformer(removeNumPunct))

#Removing all the numbers
myCorpus <- tm_map(myCorpus,removeNumbers)
#Removing extra spaces from the data
myCorpus <- tm_map(myCorpus, stripWhitespace)
#Removing punctuation from the data
myCorpus <- tm_map(myCorpus, removePunctuation)
#Creating a copy for later use
myCorpusCopy <- myCorpus
#Inspecting first three tweets
inspect(myCorpus[1:3])
```

<> Metadata: corpus specific: 1, document level (indexed): 0 Content: documents: 3

[1] crajbhavan kolkata india maga country chandigarh india new delhi india jaipur sri anand pur sahib punjab california usa new delhi india new delhi india moscow russian federation himalayas newdelhi new delhi india uu uufuuduuudueuu amravati enclave panchkula new delhi india new delhi india jammu delhi mandi himachal pradesh balochistan india new delhi india uuueud mumbai gujarat india new delhi india delhi noida up bhopal india orange county amritsar india new delhi india chandigarh india delhi kashmir mumbai bhubaneshwar india puri odisha bhaarat delhi jammu jammu jammu and kashmir new delhi srinagar and islamabad new delhi new delhi india union territory of jk ayodhya india delhi new delhiindia india varanasi india uufue uuuu uubuauc uauucueuc new yorknew jersey ct tx ca usa uufuuuuueuudufudu delhi india bahraich lucknow up bihar

india india exiled in my own country india israel ferozepur assam india new delhi india india india india india jalandhar india guwahati india new delhi udueuuue uuuu uuuuuauu ueuufuc himachal pradesh india home bharat india india new delhi india uu uufuuduu udueuu new delhi india india panchkula jaipurrajasthan new delhi india noida india delhi south block new delhi kolkata india haridwar india guwahatiindia spaced out in outer space new delhi uubuuuauu uuuduu uauduuuu delhi noida delhi delhi shirdi delhi india india delhi india toronto ontario bangalore itanagar india jerusalem israel new delhi india israel noida india noida india delhi haryana delhi delhihimachalindia india new delhi india india gurgaon mumbai india countries bangalore new york ny new delhi india maharashtra new delhi varanasi ungma mumbai india new delhi delhi patna nadiad gujarat india calgary alberta canada alberta canada delhiodishaindia india ãt north block new delhi maharashtra stateindia delhi new delhi india india new delhi india india ghaziabad uttar pradesh delhi india delhi india new delhi india india new delhi india mumbai india [2]
[3]

Wordcloud of Locations

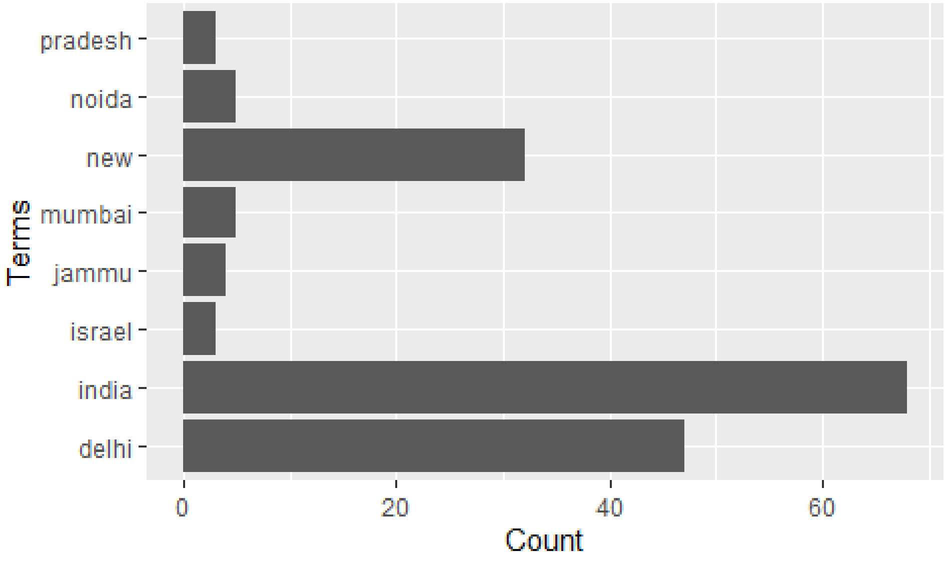
```
tdm <- TermDocumentMatrix(myCorpus)
tdm
```

<<TermDocumentMatrix (terms: 110, documents: 1)>> Non-/sparse entries: 110/0 Sparsity : 0% Maximal term length: 18 Weighting : term frequency (tf)

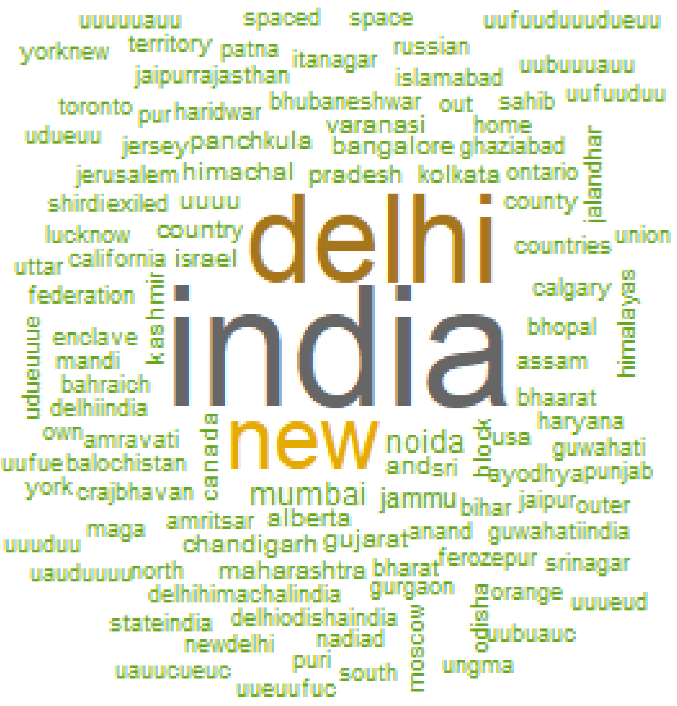
```
#The words with frequency more that 150
(freq.terms <- findFreqTerms(tdm, lowfreq = 3))
```

[1] “delhi” “india” “israel” “jammu” “mumbai” “new” “noida”
[8] “pradesh”

```
#Rowsumming and getting only words having more than 10 frequency
term.freq <- rowSums(as.matrix(tdm))
term.freq <- subset(term.freq, term.freq >= 3)
df <- data.frame(term = names(term.freq), freq = term.freq)
#Now plotting
ggplot(df, aes(x=term, y=freq)) + geom_bar(stat="identity") +
  xlab("Terms") + ylab("Count") + coord_flip()
```

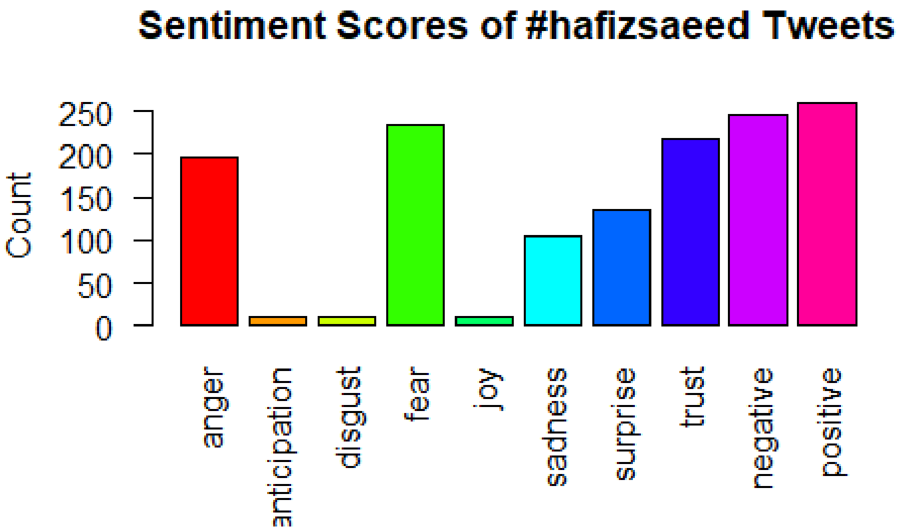


```
w <- as.matrix(tdm)
#Calculating the frequency of words and sort it by decreasing order
#of frequency
word.freq <- sort(rowSums(w), decreasing = T)
#Creating color for wordcloud
pal <- brewer.pal(8, "Dark2")[-(1:4)]
#Creating word cloud
wordcloud(words = names(word.freq), freq = word.freq,
          min.freq = 1, random.order = F, colors = pal)
```



Sentiment Analysis using Syuzhet

```
library(syuzhet)
x<-get_nrc_sentiment(fd$B)
barplot(colSums(x), las=2, col=rainbow(10), ylab = 'Count',
        main = 'Sentiment Scores of #hafizsaeed Tweets')
```



<u>The End</u>