**#-------------------AUTHENTICATION----------------**

getwd()

setwd("C:/Users/Bindu/SentiTwitterR")

getwd()

library(twitteR)

library(plyr)

library(ROAuth)

library(stringr)

library(ggplot2)

library(ggplot2)

library(RCurl)

library(wordcloud)

library(base64enc)

library(RColorBrewer)

library(e1071)

consumer\_key <- "P7uGaiQFBGUXxKfT9xlQxKGy4"

consumer\_secret <- "vDfiySuheFdFakWspRhkOaoVroOujmeMK7OsttsMfAe51V6NLK"

access\_token <- "851486451598491650-yarCnAzcA6tHwt2w0kpeTlHCxiry8PN"

access\_token\_secret <- "j3t818L1tEd99OpcTMh9Ngj6x1Jw9o2L4iL0965Js8C2C"

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

setup\_twitter\_oauth(consumer\_key,consumer\_secret,access\_token,access\_token\_secret)

cred = OAuthFactory$new(consumerKey=consumer\_key,

consumerSecret=consumer\_secret,

requestURL='https://api.twitter.com/oauth/request\_token',

accessURL='https://api.twitter.com/oauth/access\_token',

authURL='https://api.twitter.com/oauth/authorize')

cred$handshake(cainfo="cacert.pem")

tweets=searchTwitter("#modi",n=1000)

**#------------CLEANING\_TWEET----------------**

#Extract tweets. Example-

modi.tweets = searchTwitter("modi", n=150)

#converts to data frame

df <- do.call("rbind", lapply(modi.tweets, as.data.frame))

#remove odd characters

df$text <- sapply(df$text,function(row) iconv(row, "latin1", "ASCII", sub="")) #remove emoticon

df$text = gsub("(f|ht)tp(s?)://(.\*)[.][a-z]+", "", df$text) #remove URL

sample <- df$text

**#----------WORD\_DATABASE-------------**

#After downloading the hui.lui.pos and hui.lui.neg, mention below the location of the file

pos.words = scan('C:/Users/Bindu/SentiTwitterR/positive-words.txt', what='character', comment.char=';')

neg.words = scan('C:/Users/Bindu/SentiTwitterR/negative-words.txt', what='character', comment.char=';')

#Adding words to positive and negative databases

pos.words=c(pos.words, 'Congrats', 'prizes', 'prize', 'thanks', 'thnx', 'Grt', 'gr8', 'plz', 'trending', 'recovering', 'brainstorm', 'leader')

neg.words = c(neg.words, 'Fight', 'fighting', 'wtf', 'arrest', 'no', 'not')

**#-----------SCORE\_SENTIMENT-------------**

score.sentiment = function(sentences, pos.words, neg.words, .progress='none')

{

require(plyr)

require(stringr)

list=lapply(sentences, function(sentence, pos.words, neg.words)

{

sentence = gsub('[[:punct:]]',' ',sentence)

sentence = gsub('[[:cntrl:]]','',sentence)

sentence = gsub('\\d+','',sentence) #removes decimal number

sentence = gsub('\n','',sentence) #removes new lines

sentence = tolower(sentence)

word.list = str\_split(sentence, '\\s+')

words = unlist(word.list) #changes a list to character vector

pos.matches = match(words, pos.words)

neg.matches = match(words, neg.words)

pos.matches = !is.na(pos.matches)

neg.matches = !is.na(neg.matches)

pp = sum(pos.matches)

nn = sum(neg.matches)

score = sum(pos.matches) - sum(neg.matches)

list1 = c(score, pp, nn)

return (list1)

}, pos.words, neg.words)

score\_new = lapply(list, `[[`, 1)

pp1 = lapply(list, `[[`, 2)

nn1 = lapply(list, `[[`, 3)

scores.df = data.frame(score = score\_new, text=sentences)

positive.df = data.frame(Positive = pp1, text=sentences)

negative.df = data.frame(Negative = nn1, text=sentences)

list\_df = list(scores.df, positive.df, negative.df)

return(list\_df)

}

**#---------FUNC\_ON\_TWEET------------**

# Clean the tweets and returns merged data frame

result = score.sentiment(sample, pos.words, neg.words)

library(reshape)

test1=result[[1]]

test2=result[[2]]

test3=result[[3]]

#Creating three different data frames for Score, Positive and Negative

#Removing text column from data frame

test1$text=NULL

test2$text=NULL

test3$text=NULL

#Storing the first row(Containing the sentiment scores) in variable q

q1=test1[1,]

q2=test2[1,]

q3=test3[1,]

qq1=melt(q1, ,var='Score')

qq2=melt(q2, ,var='Positive')

qq3=melt(q3, ,var='Negative')

qq1['Score'] = NULL

qq2['Positive'] = NULL

qq3['Negative'] = NULL

#Creating data frame

table1 = data.frame(Text=result[[1]]$text, Score=qq1)

table2 = data.frame(Text=result[[2]]$text, Score=qq2)

table3 = data.frame(Text=result[[3]]$text, Score=qq3)

#Merging three data frames into one

table\_final=data.frame(Text=table1$Text, Score=table1$value, Positive=table2$value, Negative=table3$value)

**#------------PERCENTAGE---------------**

#Positive Percentage

#Renaming

posSc=table\_final$Positive

negSc=table\_final$Negative

#Adding column

table\_final$PosPercent = posSc/ (posSc+negSc)

#Replacing Nan with zero

pp = table\_final$PosPercent

pp[is.nan(pp)] <- 0

table\_final$PosPercent = pp

#Negative Percentage

#Adding column

table\_final$NegPercent = negSc/ (posSc+negSc)

#Replacing Nan with zero

nn = table\_final$NegPercent

nn[is.nan(nn)] <- 0

table\_final$NegPercent = nn

**#-----------GRAPHS---------------**

#Histogram

hist(table\_final$Positive, col=rainbow(10))

hist(table\_final$Negative, col=rainbow(10))

hist(table\_final$Score, col=rainbow(10))

#Pie

slices <- c(sum(table\_final$Positive), sum(table\_final$Negative))

labels <- c("Positive", "Negative")

library(plotrix)

#pie(slices, labels = labels, col=rainbow(length(labels)), main="Sentiment Analysis")

pie3D(slices, labels = labels, col=rainbow(length(labels)),explode=0.00, main="Sentiment Analysis")

**#--------------TOP\_10\_HASHTAGS---------------**

library(twitteR)

tw = userTimeline("BarackObama", n = 3200)

tw = twListToDF(tw)

vec1 = tw$text

#Extract the hashtags:

hash.pattern = "#[[:alpha:]]+"

have.hash = grep(x = vec1, pattern = hash.pattern) #stores the indices of the tweets which have hashes

hash.matches = gregexpr(pattern = hash.pattern,

text = vec1[have.hash])

extracted.hash = regmatches(x = vec1[have.hash], m = hash.matches) #the actual hashtags are stored here

df = data.frame(table(tolower(unlist(extracted.hash)))) #dataframe formed with var1(hashtag), freq of hashtag

colnames(df) = c("tag","freq")

df = df[order(df$freq,decreasing = TRUE),]

dat = head(df,50)

dat2 = transform(dat,tag = reorder(tag,freq)) #reorder it so that highest freq is at the top

library(ggplot2)

p = ggplot(dat2, aes(x = tag, y = freq)) + geom\_bar(stat="identity", fill = "blue")

p + coord\_flip() + labs(title = "Hashtag frequencies in the tweets of the Obama team (@BarackObama)")

**#--------------TOP\_TRENDS-----------------**

#assuming input = Ottawa

a\_trends = availableTrendLocations()

woeid = a\_trends[which(a\_trends$name=="Ottawa"),3]

canada\_trend = getTrends(woeid)

trends = canada\_trend[1:2]

#To clean data and remove Non English words:

dat <- cbind(trends$name)

dat2 <- unlist(strsplit(dat, split=", "))

dat3 <- grep("dat2", iconv(dat2, "latin1", "ASCII", sub="dat2"))

dat4 <- dat2[-dat3]

dat4

**#------------WORDCLOUD---------------**

modi\_text = sapply(modi.tweets, function(x) x$getText()) #sapply returns a vector

df <- do.call("rbind", lapply(modi.tweets, as.data.frame)) #lapply returns a list

modi\_text <- sapply(df$text,function(row) iconv(row, "latin1", "ASCII", sub=""))

str(modi\_text) #gives the summary/internal structure of an R object

library(tm) #tm: text mining

modi\_corpus <- Corpus(VectorSource(modi\_text)) #corpus is a collection of text documents

modi\_corpus

inspect(modi\_corpus[1])

#clean text

library(wordcloud)

modi\_clean <- tm\_map(modi\_corpus, removePunctuation)

modi\_clean <- tm\_map(modi\_clean, removeWords, stopwords("english"))

modi\_clean <- tm\_map(modi\_clean, removeNumbers)

modi\_clean <- tm\_map(modi\_clean, stripWhitespace)

wordcloud(modi\_clean, random.order=F,max.words=80, col=rainbow(50), scale=c(3.5,1))