# Library to connect with twitter & pulling Tweets & converting into db

library(twitteR)

library(syuzhet)

library(tm)

library(SnowballC)

library(ROAuth)

library(ggplot2)

require(devtools)

library(sentiment)

library(wordcloud)

#Creating access token to connect with twitter rest api

consumer\_key = "6auEJxBtDGJL0FqWJFXh23vLP"

consumer\_secret = "26rBF7WQcSx8aqiJwOyukW39uYmQovyB7th4HG4fABR1AEl9Bx"

access\_token = "1011597163011477507-9Ksmm0q61xYNuUfuRnpr2UvVZiHC1J"

access\_secret = "HDxRMCgRQWpJbJG0kkYY65zduJSKYar2tS2ZvjjVpH5Sz"

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

tk = twitteR::searchTwitter('#MotoE5Plus', n = 10000, retryOnRateLimit = 1e3)

d = twitteR::twListToDF(tk)

n.tweet<-length(d)

View(d)

head(d$text)

#Pre-processing, removing hashtags, hyperlinks etc

buffer<-Corpus(VectorSource(d$text)) # Building a Corpus, specifying it to be a character vector

#Converting to Lower Case

buffer<-tm\_map(buffer, content\_transformer(tolower))

#Removing URLs

removeURL <- function(x) gsub("http[^[:space:]]\*", "", x)

buffer <- tm\_map(buffer, content\_transformer(removeURL))

# remove anything other than English letters or space

removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]\*", "", x)

buffer <- tm\_map(buffer, content\_transformer(removeNumPunct))

# remove stopwords

myStopwords <- c(setdiff(stopwords('english'), c("r", "big")),

"use", "see", "used", "via", "amp")

buffer <- tm\_map(buffer, removeWords, myStopwords)

# remove extra whitespace

buffer <- tm\_map(buffer, stripWhitespace)

# keep a copy for stem completion later

bufferCopy <- buffer

# Stem & Stemming

buffer <- tm\_map(buffer, stemDocument) # stem words

writeLines(strwrap(buffer[[190]]$content, 60))

stC2 <- function(x, dictionary) {

x <- unlist(strsplit(as.character(x), " "))

x <- x[x != ""]

x <- stemCompletion(x, dictionary=dictionary)

x <- paste(x, sep="", collapse=" ")

PlainTextDocument(stripWhitespace(x))

}

buffer <- lapply(buffer, stC2, dictionary=bufferCopy)

buffer <- Corpus(VectorSource(buffer))

writeLines(strwrap(buffer[[190]]$content, 60))

#Counting the Frequency of words repeating

wordFreq <- function(corpus, word) {

results <- lapply(corpus,

function(x) { grep(as.character(x), pattern=paste0("\\<",word)) }

)

sum(unlist(results))

}

n.miner <- wordFreq(bufferCopy, "miner")

n.mining <- wordFreq(bufferCopy, "mining")

cat(n.miner, n.mining)

replaceWord <- function(corpus, oldword, newword) {

tm\_map(corpus, content\_transformer(gsub),

pattern=oldword, replacement=newword)

}

buffer <- replaceWord(buffer, "miner", "mining")

buffer <- replaceWord(buffer, "universidad", "university")

buffer <- replaceWord(buffer, "scienc", "science")

##Build Term Document Matrix for building a matrix that describes the repeating words

tdm <- TermDocumentMatrix(buffer,

control = list(wordLengths = c(1, Inf)))

tdm

idx <- which(dimnames(tdm)$Terms %in% c("r", "data", "mining"))

as.matrix(tdm[idx, 21:30])

# Inspecting frequent words

(freq.terms <- findFreqTerms(tdm, lowfreq = 20))

term.freq <- rowSums(as.matrix(tdm))

term.freq <- subset(term.freq, term.freq >= 1000)

df <- data.frame(term = names(term.freq), freq = term.freq)

#Plotting the data

ggplot(df, aes(x=term, y=freq)) + geom\_bar(stat="identity") +

xlab("Terms") + ylab("Count") + coord\_flip() +

theme(axis.text=element\_text(size=7))

#Sentiment Analysis

sentiments <- sentiment(d$text)

table(sentiments$polarity)

#WordCloud

m <- as.matrix(tdm)

word.freq <- sort(rowSums(m), decreasing = T)

pal <- brewer.pal(9, "BuGn")[-(1:4)]

wordcloud(words = names(word.freq), freq = word.freq, min.freq = 3,

random.order = F, colors = pal)