Data Mining in Twitter - Proof of Concept for AFRICOM

Kevin A. Ryan (JHUAPL)

Monday, September 28, 2015

Setup code to enable access to Twitter API. Explain the process here. WHat is involved.

library(devtools)  
library(httr)  
library(rjson)  
library(bit64)  
library(twitteR)  
download.file(url="http://curl.haxx.se/ca/cacert.pem", destfile="cacert.pem")  
 consumer\_key <- 'cD5uXJMuyJfLkodDvYYchEUuM'  
 consumer\_secret <- 'FLVzv3HhHnsSr0qRfnq7ZmCdCWrQS24NUbnREUvHNFhEc89bVg'  
 access\_token <- '740674741-PPuLbrOHGQCYqrh9C9E9AG8I57st4Akj1NBmB2V5'  
 access\_secret <- 'oxsCu5UE9Q6QVowgd98Ds7qFeJIesI9Z3zGk5qDdP3S4m'  
setup\_twitter\_oauth(consumer\_key, consumer\_secret, access\_token, access\_secret)

library(twitteR) ## retrieve tweets from Twitter  
  
#TRUMP  
Trump <-userTimeline('realdonaldTrump' , n=1500, maxID=NULL, sinceID=NULL, includeRts=TRUE,excludeReplies=TRUE)  
tweets.dt <- twListToDF(Trump)# convert tweets to a data frame  
 for (i in c(1:2, 1500)) { #clean up wrap text feature  
 cat(paste0("[", i, "] "))  
 writeLines(strwrap(tweets.dt$text[i], 60))}  
write.csv(tweets.dt, file = "trump")#shortcut for saving tweets to local maching

Build a corpus and clean up

library(tm)

## Loading required package: NLP

library(NLP)  
  
#CLINTON  
trump <- read.csv("~/\_data/TwitterAnalysis/trump")#load the file saved locally  
myCorpus <- Corpus(VectorSource(trump$text))# build a corpus, and specify the source to be character vectors  
myCorpus <- tm\_map(myCorpus, content\_transformer(tolower))# convert to lower case  
removeURL <- function(x) gsub("http[^[:space:]]\*", "", x)# remove URLs  
myCorpus <- tm\_map(myCorpus, content\_transformer(removeURL))  
removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]\*", "", x)# leave only English letters or spaces  
myCorpus <- tm\_map(myCorpus, content\_transformer(removeNumPunct))  
myCorpus <- tm\_map(myCorpus, removePunctuation)#remove punctuation  
myCorpus <- tm\_map(myCorpus, removeNumbers)#remove numbers  
myCorpus <- tm\_map(myCorpus, removeWords, myStopwords)# remove stopwords from corpus  
myCorpus <- tm\_map(myCorpus, stripWhitespace)# remove extra whitespace  
myCorpusCopy <- myCorpus# keep a copy of corpus to use later as a dictionary for stem completion  
myCorpus <- tm\_map(myCorpus, stemDocument)# stem words

Stem Completion step

library(tm)  
myCorpus <- Corpus(VectorSource(myCorpus))  
tdm <- TermDocumentMatrix(myCorpus, control = list(wordLengths = c(1, Inf)))

Frequent Words and Associations

library(tm)  
library(topicmodels)  
library(gridExtra)  
(freq.terms <- findFreqTerms(tdm, lowfreq = 35, highfreq = 90))

FALSE [1] "american" "back" "big" "border"   
FALSE [5] "bush" "can" "candid" "cant"   
FALSE [9] "cnn" "countri" "danscavino" "debat"   
FALSE [13] "dont" "foxandfriend" "foxnew" "get"   
FALSE [17] "give" "go" "gop" "illeg"   
FALSE [21] "im" "iowa" "jeb" "job"   
FALSE [25] "know" "lead" "let" "like"   
FALSE [29] "love" "make" "megynkelli" "mr"   
FALSE [33] "need" "never" "new" "now"   
FALSE [37] "one" "peopl" "presid" "realli"   
FALSE [41] "say" "show" "support" "talk"   
FALSE [45] "time" "tonight" "veteran" "via"   
FALSE [49] "vote" "want" "watch" "win"

term.freq <- rowSums(as.matrix(tdm))  
term.freq <- subset(term.freq, term.freq >= 35)  
term.freq <- subset(term.freq, term.freq <= 90)  
df <- data.frame(term = names(term.freq), freq = term.freq)  
df$term <-factor(df$term, levels=df[order(df$freq), "term"])  
  
library(ggplot2)

FALSE   
FALSE Attaching package: 'ggplot2'  
FALSE   
FALSE The following object is masked from 'package:NLP':  
FALSE   
FALSE annotate

x<-ggplot(df, aes(x = term, y = freq)) + geom\_bar(stat = "identity") +  
xlab("Terms") + ylab("Count") + coord\_flip()  
  
# which words are associated with the term "tax" ?  
(findAssocs(tdm, "tax", 0.2))

FALSE $tax  
FALSE gal hedg practic pay asap   
FALSE 0.50 0.50 0.50 0.41 0.35   
FALSE reduc rais plan middl danhenning   
FALSE 0.35 0.33 0.32 0.30 0.25   
FALSE economist erinburnett larri larrykudlow lip   
FALSE 0.25 0.25 0.25 0.25 0.25   
FALSE lower miriamrosemc outfrontcnn planth planthank   
FALSE 0.25 0.25 0.25 0.25 0.25   
FALSE polina reduct taxesjust topic various   
FALSE 0.25 0.25 0.25 0.25 0.25   
FALSE class fund higher   
FALSE 0.24 0.22 0.22

WOrd cloud mutherfucker!

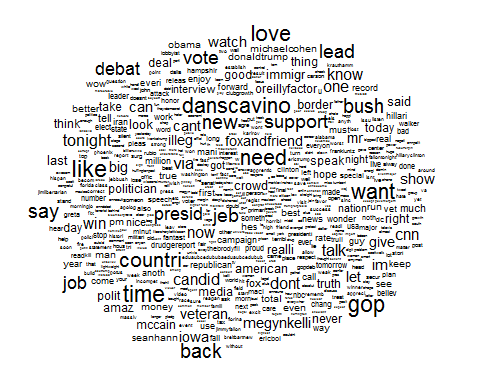
# plot word cloud  
library(wordcloud)

## Warning: package 'wordcloud' was built under R version 3.2.2

## Loading required package: RColorBrewer

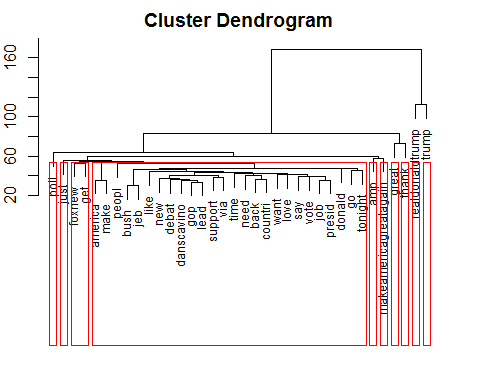
## Warning: package 'RColorBrewer' was built under R version 3.2.1

par(mar=c(.5,.5,.5,.5))  
 m <- as.matrix(tdm)  
# calculate the frequency of words and sort it descendingly by frequency  
wordFreq <- sort(rowSums(m), decreasing=TRUE)  
wordFreq <- subset(wordFreq, wordFreq >= 5)  
wordFreq <- subset(wordFreq, wordFreq <= 75)  
df2 <- data.frame(term = names(wordFreq), freq = wordFreq)  
  
# word cloud  
set.seed(375) # to make it reproducible  
  
wordcloud(words=df2$term, freq=df2$freq, scale = c(1.5 ,.1),min.freq=5, random.order=T, fixed.asp=F, rot.per= 0, use.r.layout =T)



clustering....In the above dendrogram, we can see the topics in the tweets. Words ", " and " are clustered into one group, because there are a couple of tweets on social network analysis. The second cluster from left comprises ", " and ", and they are clustered into one group because of tweets on vacancies of research and postdoctoral positions. We can also see cluster on time series, R packages, parallel computing, R codes and examples, and tutorial and slides. The rightmost three clusters consists of ", "and", which are the keywords of @RDataMining tweets.

library(gridExtra)  
myTdm2 <- removeSparseTerms(tdm, sparse=0.97)#remove sparse terms  
m2 <- as.matrix(myTdm2)  
distMatrix <- dist(scale(m2))# cluster terms  
fit <- hclust(distMatrix, method="ward.D")  
par(mar=c(2,2,2,2))  
plot(fit, asp =1, cex=.75)  
rect.hclust(fit, k=10)# cut tree into 10 clusters



topic modeling2

library(tm)  
library(topicmodels)  
library(gridExtra)  
library(ggplot2)  
dtm <- as.DocumentTermMatrix(tdm) #1437 documents  
rowTotals <- apply(dtm , 1, sum) #Find the sum of words in each Document  
dtm.new <- dtm[rowTotals> 0, ] #remove all docs without words  
trump2 <- trump[1:1484,]   
lda <- LDA(dtm.new, k = 8)#Latent Dirichlet allocation (LDA)  
(term <- terms(lda, 6))

## Topic 1 Topic 2 Topic 3   
## [1,] "trump" "america" "trump"   
## [2,] "realdonaldtrump" "realdonaldtrump" "realdonaldtrump"   
## [3,] "america" "go" "poll"   
## [4,] "just" "thank" "donald"   
## [5,] "poll" "get" "makeamericagreatagain"  
## [6,] "say" "trump" "go"   
## Topic 4 Topic 5 Topic 6   
## [1,] "trump" "realdonaldtrump" "trump"   
## [2,] "realdonaldtrump" "great" "realdonaldtrump"  
## [3,] "america" "thank" "great"   
## [4,] "foxnew" "poll" "thank"   
## [5,] "just" "trump" "amp"   
## [6,] "debat" "amp" "peopl"   
## Topic 7 Topic 8   
## [1,] "realdonaldtrump" "realdonaldtrump"  
## [2,] "trump" "amp"   
## [3,] "poll" "great"   
## [4,] "gop" "new"   
## [5,] "great" "thank"   
## [6,] "just" "foxnew"

term <- apply(term, MARGIN = 2, paste, collapse = ", ")  
topic <- topics(lda, 1)  
topics <- data.frame(date=as.Date(trump2$created), topic)  
qplot(date, ..count.., data=topics, geom="density",fill=term[topic], position="stack")

