Data Mining in Twitter - Proof of Concept for AFRICOM

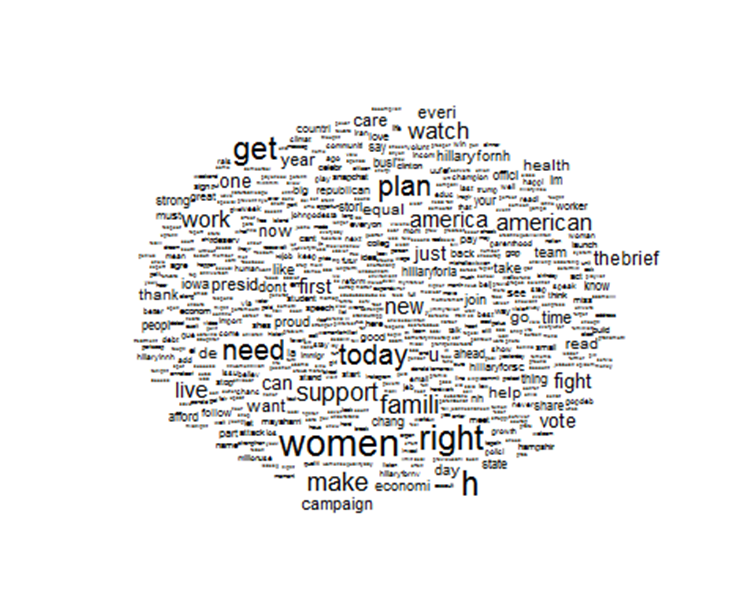
Kevin A. Ryan (JHUAPL)

Monday, September 28, 2015

This document presents an overview of the technical concepts of text mining as applied to a social media data source. The previous 4 months of Twitter data was extracted for two candidates, from different political parties, currently running for president. Analyzing the two candidate’s twitter data will…..

TheIt starts with extracting text from Twitter. The extracted text is then transformed to build a document-term matrix. After that, frequent words and associations are found from the matrix. A word cloud is used to present important words in documents. In the end, words and tweets are clustered to

Comparison of the Trump and the Clinton word

nd groups of words and also groups of tweets. In this

chapter, \tweet" and \document" will be used interchangeably, so are \word" and \term".

Step 1: Initial steps for using the Twitter API using the R programming language and some openly available add on packages: (devtools),(httr),(rjson). This method uses Oauth methods. There are limits on this particular version of the API and only 1500 tweets can be downloaded at a time. This code accesses the Twitter API and downloads the last 1500 tweets from the Twitter user = 'HillaryClinton'.

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  
Clinton <-userTimeline('HillaryClinton' , n=1500, maxID=NULL, sinceID=NULL, includeRts=TRUE,excludeReplies=TRUE)  
tweets.hc <- twListToDF(Clinton)# convert tweets to a data frame  
 for (i in c(1:2, 1500)) { #clean up wrap text feature  
 cat(paste0("[", i, "] "))  
 writeLines(strwrap(tweets.hc$text[i], 60))}  
write.csv(tweets.hc, file = "clinton")#shortcut for saving tweets locally

Once the

library(tm)

library(NLP)  
clinton <- read.csv("~/\_data/TwitterAnalysis/clinton")#load the file saved locally  
myCorpus <- Corpus(VectorSource(clinton$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)

library(ggplot2)  
#(freq.terms <- findFreqTerms(tdm, lowfreq = 35, highfreq = 115))  
term.freq <- rowSums(as.matrix(tdm))  
term.freq <- subset(term.freq, term.freq >= 35)  
term.freq <- subset(term.freq, term.freq <= 115)  
df <- data.frame(term = names(term.freq), freq = term.freq)  
df$term <-factor(df$term, levels=df[order(df$freq), "term"])  
y<-ggplot(df, aes(x = term, y = freq)) + geom\_bar(stat = "identity") +  
xlab("Terms") + ylab("Count") + coord\_flip()  
 grid.arrange(y, y, ncol=2)

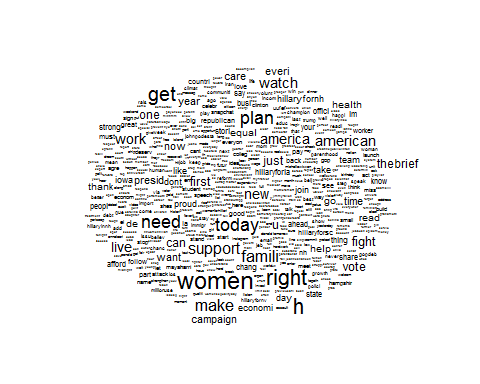
# which words are associated with the term "tax" ?  
(findAssocs(tdm, "tax", 0.2))

FALSE $tax  
FALSE billionair yuuuug credit apprent deport incent   
FALSE 0.71 0.71 0.50 0.35 0.35 0.35   
FALSE releas skill wsj biggest winner break   
FALSE 0.35 0.35 0.35 0.25 0.25 0.22   
FALSE invest   
FALSE 0.22

WOrd cloud mutherfucker!

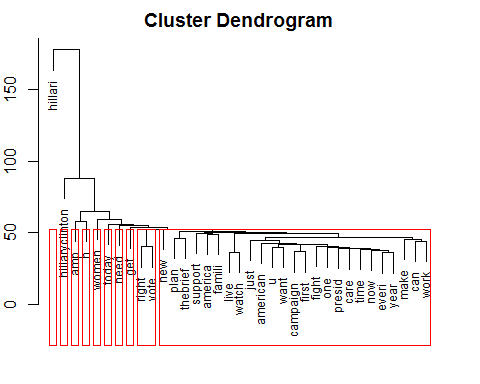
# plot word cloud  
library(wordcloud)

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 <= 115)  
df2 <- data.frame(term = names(wordFreq), freq = wordFreq)  
  
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)  
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  
clinton2 <- clinton[1:1437,]   
lda <- LDA(dtm.new, k = 8)#Latent Dirichlet allocation (LDA)  
(term <- terms(lda, 6))

## Topic 1 Topic 2 Topic 3 Topic 4 Topic 5   
## [1,] "hillari" "hillari" "women" "hillari" "hillari"  
## [2,] "de" "women" "h" "hillaryclinton" "support"  
## [3,] "get" "hillaryclinton" "thebrief" "work" "women"   
## [4,] "watch" "today" "plan" "fight" "one"   
## [5,] "la" "right" "american" "get" "today"   
## [6,] "make" "amp" "amp" "care" "famili"   
## Topic 6 Topic 7 Topic 8   
## [1,] "amp" "hillari" "need"   
## [2,] "hillari" "hillaryclinton" "right"   
## [3,] "get" "h" "hillari"  
## [4,] "women" "plan" "new"   
## [5,] "time" "right" "work"   
## [6,] "america" "america" "now"

term <- apply(term, MARGIN = 2, paste, collapse = ", ")  
topic <- topics(lda, 1)  
topics <- data.frame(date=as.Date(clinton2$created), topic)  
qplot(date, ..count.., data=topics, geom="density",fill=term[topic], position="stack")