Homework 5 Text Mining and Topic Modelling

Hamed

4/13/2020

# Load the libraries  
library(twitteR)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:twitteR':  
##   
## id, location

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(splitstackshape)  
library(tidytext)  
library(purrr)

# Load the blackboard dataset   
df\_tweet = read.csv("trump\_tweets3.csv",header = T,stringsAsFactors = FALSE)  
TrumpTweets = df\_tweet  
head(TrumpTweets)

## index  
## 1 1  
## 2 2  
## 3 3  
## 4 4  
## 5 5  
## 6 6  
## text  
## 1 I am asking the chairs of the House and Senate committees to investigate top secret intelligence shared with NBC prior to me seeing it.  
## 2 Anna Wintour came to my office at Trump Tower to ask me to meet with the editors of Conde Nast &amp; Steven Newhouse, a friend. Will go this AM.  
## 3 being a movie star-and that was season 1 compared to season 14. Now compare him to my season 1. But who cares, he supported Kasich &amp; Hillary  
## 4 Wow, the ratings are in and Arnold Schwarzenegger got "swamped" (or destroyed) by comparison to the ratings machine, DJT. So much for....  
## 5 Hopefully, all supporters, and those who want to MAKE AMERICA GREAT AGAIN, will go to D.C. on January 20th. It will be a GREAT SHOW!  
## 6 and knew they were in big trouble - which is why they cancelled their big fireworks at the last minute.THEY SAW A MOVEMENT LIKE NEVER BEFORE  
## favorited favoriteCount replyToSN created truncated replyToSID  
## 1 FALSE 12424 <NA> 1/6/2017 16:51 FALSE NA  
## 2 FALSE 15873 <NA> 1/6/2017 14:18 FALSE NA  
## 3 FALSE 22599 <NA> 1/6/2017 12:42 FALSE NA  
## 4 FALSE 21196 <NA> 1/6/2017 12:34 FALSE NA  
## 5 FALSE 37136 <NA> 1/6/2017 12:05 FALSE NA  
## 6 FALSE 25111 <NA> 1/6/2017 11:45 FALSE NA  
## id replyToUID  
## 1 8.17413e+17 NA  
## 2 8.17375e+17 NA  
## 3 8.17351e+17 NA  
## 4 8.17349e+17 NA  
## 5 8.17341e+17 NA  
## 6 8.17336e+17 NA  
## statusSource  
## 1 <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>  
## 2 <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>  
## 3 <a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>  
## 4 <a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>  
## 5 <a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>  
## 6 <a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>  
## screenName retweetCount isRetweet retweeted longitude latitude  
## 1 realDonaldTrump 4003 FALSE FALSE NA NA  
## 2 realDonaldTrump 3439 FALSE FALSE NA NA  
## 3 realDonaldTrump 5521 FALSE FALSE NA NA  
## 4 realDonaldTrump 5778 FALSE FALSE NA NA  
## 5 realDonaldTrump 8645 FALSE FALSE NA NA  
## 6 realDonaldTrump 5490 FALSE FALSE NA NA

# Let's get rid of links and format dataframe in a way when only one word is in line.  
  
TrumpTweets <- TrumpTweets[-(grep('t.co', TrumpTweets$'text')),]  
TrumpTweets$tweet <- 'tweet'  
TrumpTweets <- TrumpTweets[ , c('text', 'tweet')]  
TrumpTweets <- unnest\_tokens(TrumpTweets, word, text)  
tail(TrumpTweets)

## tweet word  
## 571.15 tweet that  
## 571.16 tweet issue  
## 571.17 tweet has  
## 571.18 tweet only  
## 571.19 tweet gotten  
## 571.20 tweet bigger

head(TrumpTweets)

## tweet word  
## 1 tweet i  
## 1.1 tweet am  
## 1.2 tweet asking  
## 1.3 tweet the  
## 1.4 tweet chairs  
## 1.5 tweet of

# It's obvious that dataframe also contains various words without useful content.  
# So it's a good idea to get rid of them.  
TrumpTweets <- anti\_join(TrumpTweets, stop\_words, by = c('word' = 'word'))  
  
tail(TrumpTweets)

## tweet word  
## 2602 tweet hillary  
## 2603 tweet dem  
## 2604 tweet nomination  
## 2605 tweet mails  
## 2606 tweet issue  
## 2607 tweet bigger

head(TrumpTweets)

## tweet word  
## 1 tweet chairs  
## 2 tweet house  
## 3 tweet senate  
## 4 tweet committees  
## 5 tweet investigate  
## 6 tweet top

# Let's see how many times each word appears in Donald Trump's tweets.  
word\_count <- dplyr::count(TrumpTweets, word, sort = TRUE)  
head(word\_count)

## # A tibble: 6 x 2  
## word n  
## <chr> <int>  
## 1 hillary 39  
## 2 election 29  
## 3 debate 27  
## 4 people 22  
## 5 media 20  
## 6 win 20

# Using the BING Lexicons  
# --------------------------  
# Now it's time to create some dataframe with sentiments that will be used for tweets  
# classification. We will use bing dictionary although you can easily use any other source.  
sentiments\_bing <-get\_sentiments("bing")  
head(sentiments\_bing)

## # A tibble: 6 x 2  
## word sentiment  
## <chr> <chr>   
## 1 2-faces negative   
## 2 abnormal negative   
## 3 abolish negative   
## 4 abominable negative   
## 5 abominably negative   
## 6 abominate negative

sentiments\_bing <- dplyr::select(sentiments\_bing, word, sentiment)  
TrumpTweets\_sentiments\_bing <- merge(word\_count, sentiments\_bing, by.x = c('word'), by.y = c('word'))  
head(TrumpTweets\_sentiments\_bing)

## word n sentiment  
## 1 abusive 1 negative  
## 2 accurately 1 positive  
## 3 affordable 4 positive  
## 4 appalled 1 negative  
## 5 attack 4 negative  
## 6 attacks 1 negative

# Above we did a simple classification of Trump's tweets words using our sentiment  
# bag of words. And this is how the result looks:  
# Let's look at the number of occurrences per sentiment in tweets.  
sentiments\_count\_bing <- dplyr::count(TrumpTweets\_sentiments\_bing, sentiment, sort = TRUE)  
sentiments\_count\_bing

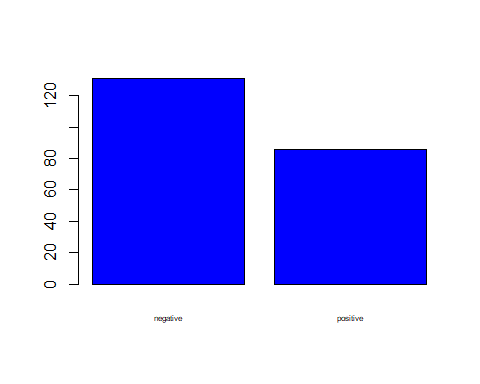
## # A tibble: 2 x 2  
## sentiment n  
## <chr> <int>  
## 1 negative 131  
## 2 positive 86

# We also may want to know the total count and percentage of all the sentiments.  
sentiments\_sum <- sum(sentiments\_count\_bing$'n')  
sentiments\_count\_bing$'percentage' <- sentiments\_count\_bing$'n' / sentiments\_sum

# Let's now create an ordered dataframe for plotting counts of sentiments.  
sentiments\_count\_bing <- rbind(sentiments\_count\_bing)  
  
# sentiment\_count <- sentiment\_count[order(sentiment\_count$sentiment), ]  
sentiments\_count\_bing

## # A tibble: 2 x 3  
## sentiment n percentage  
## <chr> <int> <dbl>  
## 1 negative 131 0.604  
## 2 positive 86 0.396

# And now it's time for the visualization. We will plot the results of our classifier.  
sentiments\_count\_bing$'colour' <- as.integer(4)  
barplot(sentiments\_count\_bing$'n', names.arg = sentiments\_count\_bing$'sentiment', col = sentiments\_count\_bing$'colour', cex.names = .5)



barplot(sentiments\_count\_bing$'percentage', names.arg = sentiments\_count\_bing$'sentiment', col = sentiments\_count\_bing$'colour', cex.names = .5)

