Sharat\_Sripada\_HW10.R

ssharat

2020-03-22

#  
# Course: IST-687  
# Name: Sharat Sripada  
# Homework #10  
# Due Date: 3/22/2020  
# Date Submitted: 3/22/2020  
# Topic: Text-mining  
#  
  
  
# install.packages("tm")  
library(tm)

## Loading required package: NLP

# install.packages("wordcloud")  
library(wordcloud)

## Loading required package: RColorBrewer

get\_sentiment\_score <- function(word\_corpus){  
 tdm <- TermDocumentMatrix(word\_corpus)  
 # convert tdm into a matrix called "m"  
 m <- as.matrix(tdm)  
 wordCounts <- rowSums(m)  
 my\_df <- data.frame(word = names(wordCounts), freq = wordCounts)  
 AFINN <- read.delim("/Users/ssharat/Downloads/HW10AFINN111\_2\_2\_2\_2\_2\_2\_2\_2\_2\_2\_2\_2.txt",sep="\t",header = FALSE)  
 colnames(AFINN) <- c("Word", "Score")  
 # Merge the affinity score data-frame with wordcounts  
 mergedTable <- merge(my\_df, AFINN, by.x="word", by.y="Word")  
 overallScore <- sum(mergedTable$freq \* mergedTable$Score)  
 totalWords <- sum(wordCounts)  
 print(overallScore/totalWords)  
 return(overallScore/totalWords)  
}  
  
get\_pos\_neg\_ratios <- function(corpus\_words, p, n){  
 tdm <- TermDocumentMatrix(corpus\_words)  
 # convert tdm into a matrix called "m"  
 m <- as.matrix(tdm)  
 # create a list of word counts for the first quarter and sort the list  
 wordCounts <- rowSums(m)  
 wordCounts <- sort(wordCounts, decreasing=TRUE)  
 totalWords <- sum(wordCounts)  
 # create a vector that contains all the words in "wordCounts1"  
 words <- names(wordCounts)  
 # locate which words in first quarter were positive (appeared in positive-word list)  
 matchedP <- match(words, p, nomatch = 0)  
 # calculate the number of positive words in first quarter  
 ptotalNumber <- sum(wordCounts[which(matchedP != 0)])  
 # calculate the ratio of positive words  
 ratiop <- ptotalNumber/totalWords  
 # locate which words in first quarter were negative (appeared in negative-word list)  
 matchedN <- match(words, n, nomatch = 0)  
 # calculate the number of negative words   
 ntotalNumber <- sum(wordCounts[which(matchedN != 0)])  
 # calculate the ratio of negative words  
 ration <- ntotalNumber/totalWords  
 return(c(ratiop, ration))  
}  
  
mlk <- readLines("/Users/ssharat/Downloads/MLK\_2\_2\_2\_2\_2\_2\_2\_2\_2\_2.txt")  
mlk <- mlk[which(mlk != "")] # remove all blank lines in the text  
  
# Create a term matrix  
# interprets each element of the "mlk" as a document and create a vector source  
words.vec <- VectorSource(mlk)  
# create a Corpus, a "Bag of Words"  
words.corpus <- Corpus(words.vec)  
# first step transformation: make all of the letters in "words.corpus" lowercase  
words.corpus <- tm\_map(words.corpus, content\_transformer(tolower))

## Warning in tm\_map.SimpleCorpus(words.corpus, content\_transformer(tolower)):  
## transformation drops documents

# second step transformation: remove the punctuation in "words.corpus"  
words.corpus <- tm\_map(words.corpus, removePunctuation)

## Warning in tm\_map.SimpleCorpus(words.corpus, removePunctuation): transformation  
## drops documents

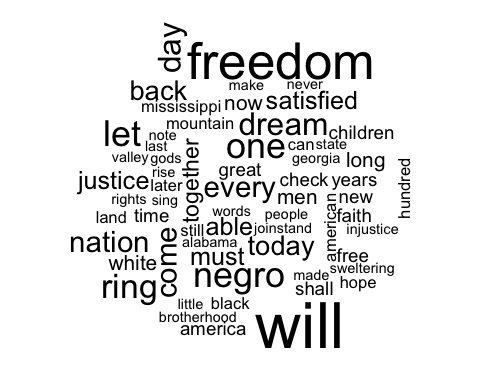
# third step transformation: remove numbers in "words.corpus"  
words.corpus <- tm\_map(words.corpus, removeNumbers)

## Warning in tm\_map.SimpleCorpus(words.corpus, removeNumbers): transformation  
## drops documents

# final step transformation: take out the "stop" words, such as "the", "a" and "at"  
words.corpus <- tm\_map(words.corpus, removeWords, stopwords("english"))

## Warning in tm\_map.SimpleCorpus(words.corpus, removeWords, stopwords("english")):  
## transformation drops documents

# create a term-document matrix "tdm"  
tdm <- TermDocumentMatrix(words.corpus)  
  
# convert tdm into a matrix called "m"  
m <- as.matrix(tdm)  
  
# create a list of counts for each word named "wordCounts"  
wordCounts <- rowSums(m)  
  
# create a vector "words" that contains all the words in "wordCounts"  
words <- names(wordCounts)  
  
# sort words in "wordCounts" by frequency  
wordCounts\_decr <- sort(wordCounts, decreasing=TRUE)  
  
# Build Word Cloud  
cloudFrame <- data.frame(word = names(wordCounts\_decr), freq=wordCounts\_decr)  
wordcloud(cloudFrame$word, cloudFrame$freq)



wordcloud(names(wordCounts\_decr),   
 wordCounts\_decr,   
 min.freq=3,   
 max.words=50,   
 rot.per=.35,   
 colors=brewer.pal(8,"Dark2"))



# Get setiment score  
get\_sentiment\_score(words.corpus)

## [1] 0.1343639

## [1] 0.1343639

# Get sentiment score for 1st quarter  
cutpoint <- round(length(words.corpus)/4)  
words.corpus1 <- words.corpus[1:cutpoint]  
score1 <- get\_sentiment\_score(words.corpus1)

## [1] 0.1128527

# Get sentiment score for 2nd quarter  
words.corpus2 <- words.corpus[(cutpoint+1):(2\*cutpoint)]  
score2 <- get\_sentiment\_score(words.corpus2)

## [1] 0.1061224

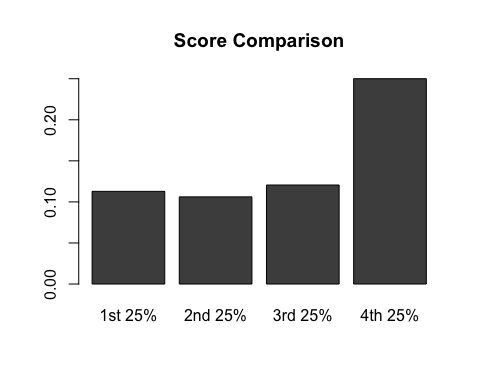
# Get sentiment score for 3rd quarter  
words.corpus3 <- words.corpus[(2\*cutpoint+1):(3\*cutpoint)]  
score3 <- get\_sentiment\_score(words.corpus3)

## [1] 0.1205674

# Get sentiment score for 4th quarter  
words.corpus4 <- words.corpus[(3\*cutpoint+1):length(words.corpus)]  
score4 <- get\_sentiment\_score(words.corpus4)

## [1] 0.25

# Plot a bar-graph for the above  
sentiment\_scores <- cbind(score1, score2, score3, score4)  
barplot(sentiment\_scores, names.arg = c("1st 25%","2nd 25%","3rd 25%","4th 25%"), main = "Score Comparison")



# Comparing the sentiment scores with postive & negative word index  
pos <- "/Users/ssharat/Downloads/positivewords\_2\_2\_2\_2\_2\_2\_2\_2\_2\_2\_2.txt"  
p <- scan(pos, character(0), sep = "\n")   
neg <- "/Users/ssharat/Downloads/negativewords\_2\_2\_2\_2\_2\_2\_2\_2\_2\_2\_2.txt"  
n <- scan(neg, character(0), sep = "\n")  
  
# remove useless rows (row 1 to row 34) of "p"  
p <- p[-c(1:34)]  
# remove useless rows (row 1 to row 34) of "n"  
n <- n[-c(1:34)]  
  
# Get postive & negative ratio for 1st quarter  
cutpoint <- round(length(words.corpus)/4)  
words.corpus1 <- words.corpus[1:cutpoint]  
ratios1 <- c(get\_pos\_neg\_ratios(words.corpus1, p, n))  
print(ratios1)

## [1] 0.09717868 0.10344828

# Get postive & negative ratio for 2nd quarter  
words.corpus2 <- words.corpus[(cutpoint+1):(2\*cutpoint)]  
ratios2 <- c(get\_pos\_neg\_ratios(words.corpus1, p, n))  
print(ratios2)

## [1] 0.09717868 0.10344828

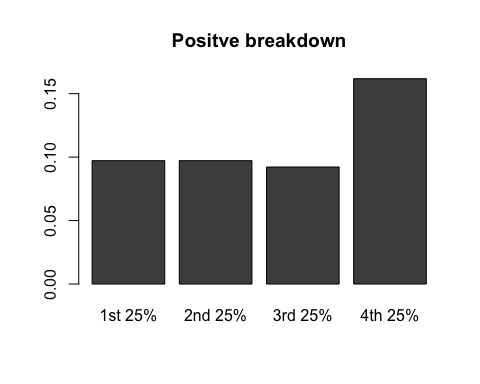
# Get postive & negative ratio for 3rd quarter  
words.corpus3 <- words.corpus[(2\*cutpoint+1):(3\*cutpoint)]  
ratios3 <- c(get\_pos\_neg\_ratios(words.corpus3, p, n))  
print(ratios3)

## [1] 0.09219858 0.06382979

# Get postive & negative ratio for 4th quarter  
words.corpus4 <- words.corpus[(3\*cutpoint+1):length(words.corpus)]  
ratios4 <- c(get\_pos\_neg\_ratios(words.corpus4, p, n))  
print(ratios4)

## [1] 0.161764706 0.007352941

# Plot a bar-graph for postive ratio  
positive\_ratio <- cbind(ratios1[1], ratios2[1], ratios3[1], ratios4[1])  
barplot(positive\_ratio, names.arg = c("1st 25%","2nd 25%","3rd 25%","4th 25%"), main = "Positve breakdown")



# Plot a bar-graph for negative ratio  
positive\_ratio <- cbind(ratios1[2], ratios2[2], ratios3[2], ratios4[2])  
barplot(positive\_ratio, names.arg = c("1st 25%","2nd 25%","3rd 25%","4th 25%"), main = "Negative breakdown")

