Text Mining

Varun Dua

01/04/2020

## Part 1:

URL = "http://www.cs.cornell.edu/people/pabo/movie-review-data/rt-polaritydata.tar.gz"  
download.file(URL, "rt-polaritydata.tar.gz")  
untar("rt-polaritydata.tar.gz")  
library(readtext)  
library(quanteda)

## Package version: 2.0.1

## Parallel computing: 2 of 8 threads used.

## See https://quanteda.io for tutorials and examples.

##   
## Attaching package: 'quanteda'

## The following object is masked from 'package:utils':  
##   
## View

df\_neg <- data.frame(sentence = readLines("./rt-polaritydata/rt-polarity.neg"),   
 stringsAsFactors = FALSE)  
df\_neg['sentiment'] <- "neg"  
  
df\_pos <- data.frame(sentence = readLines("./rt-polaritydata/rt-polarity.pos"),   
 stringsAsFactors = FALSE)  
df\_pos['sentiment'] <- "pos"  
  
corp\_movies <- corpus(rbind(df\_neg, df\_pos), text\_field='sentence')  
  
toks\_corp <- tokens(corp\_movies, remove\_punct = TRUE)  
  
nostop\_corp\_movies <- tokens\_select(toks\_corp, pattern = stopwords('en'), selection = 'remove')  
head(nostop\_corp\_movies)

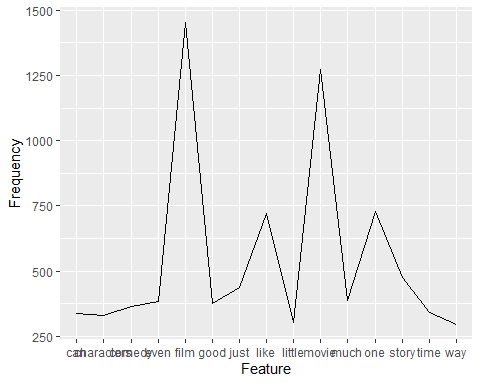
## Tokens consisting of 6 documents and 1 docvar.  
## text1 :  
## [1] "simplistic" "silly" "tedious"   
##   
## text2 :  
## [1] "laddish" "juvenile" "teenage" "boys" "possibly" "find" "funny"   
##   
## text3 :  
## [1] "exploitative" "largely" "devoid" "depth"   
## [5] "sophistication" "make" "watching" "graphic"   
## [9] "treatment" "crimes" "bearable"   
##   
## text4 :  
## [1] "garbus" "discards" "potential" "pathological"   
## [5] "study" "exhuming" "instead" "skewed"   
## [9] "melodrama" "circumstantial" "situation"   
##   
## text5 :  
## [1] "visually" "flashy" "narratively" "opaque"   
## [5] "emotionally" "vapid" "exercise" "style"   
## [9] "mystification"  
##   
## text6 :  
## [1] "story" "also" "unoriginal" "come" "already"   
## [6] "recycled" "times" "care" "count"

### a) Frequency plot of top 15 words

df <- dfm(nostop\_corp\_movies)  
ts <- textstat\_frequency(df)  
  
ts <- head(ts,15)  
library(ggplot2)  
fplot<- ggplot(data = ts, aes(ts$feature, ts$frequency)) + geom\_line(group = 1) + labs(x = "Feature", y = "Frequency")  
print(ts)

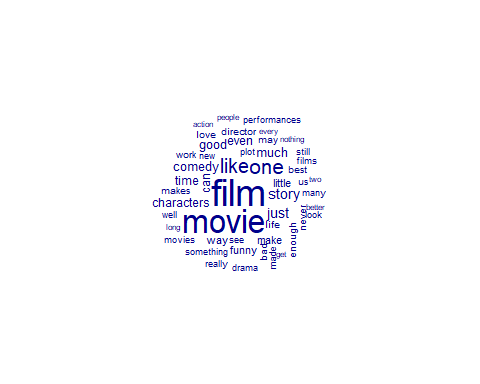
## feature frequency rank docfreq group  
## 1 film 1453 1 1423 all  
## 2 movie 1273 2 1231 all  
## 3 one 727 3 696 all  
## 4 like 721 4 693 all  
## 5 story 477 5 470 all  
## 6 just 439 6 428 all  
## 7 much 386 7 380 all  
## 8 even 382 8 379 all  
## 9 good 377 9 354 all  
## 10 comedy 366 10 361 all  
## 11 time 341 11 331 all  
## 12 can 338 12 328 all  
## 13 characters 330 13 329 all  
## 14 little 302 14 294 all  
## 15 way 296 15 290 all

fplot



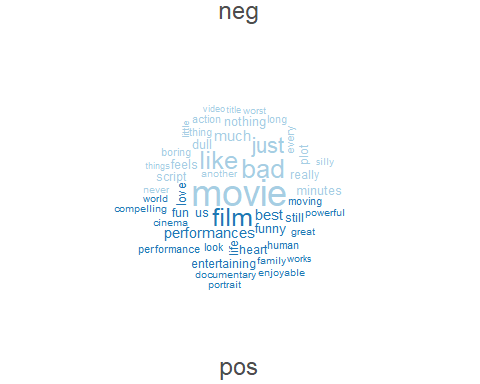
### b) Word Cloud of 50 most common words

set.seed(132)  
textplot\_wordcloud(df, max\_words = 50)



### c) Grouped Word Cloud of 50 most common words in positive or negative sentences

dfmat\_grouped <- dfm(nostop\_corp\_movies, groups = "sentiment")  
set.seed(132)  
textplot\_wordcloud(dfmat\_grouped, comparison = TRUE, max\_words = 50)



### d) Plot of Lexical diversity of 20 random sentences

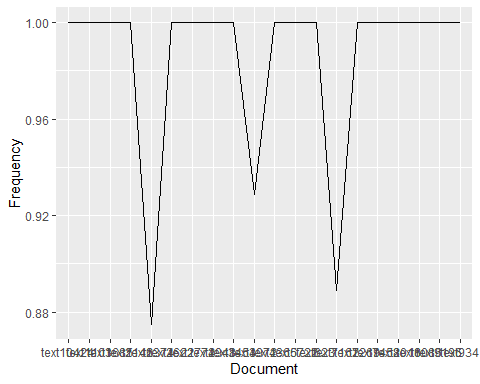
library(dplyr)

##   
## Attaching package: 'dplyr'

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

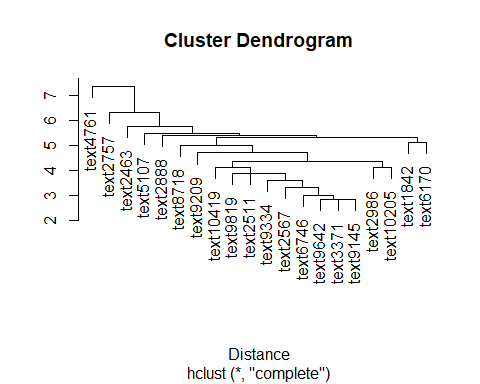
## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

set.seed(1234)  
tstat\_lexdiv <- textstat\_lexdiv(df)  
sampled\_df<- sample\_n(tstat\_lexdiv,20)  
splot<- ggplot(data = sampled\_df, aes(sampled\_df$document, sampled\_df$TTR)) + geom\_line(group = 1) + labs(x = "Document", y = "Frequency")  
splot



### e) Dendogram of hierarchical clustering of randomly selected 20 sentences

set.seed(123)  
cs<- corpus\_sample(corp\_movies, 20)  
toks\_new <- tokens(cs)  
df <- dfm(toks\_new, remove = stopwords('en'))  
dist\_mat <- as.dist(textstat\_dist(df))  
clust<- hclust(dist\_mat)  
plot(clust, xlab = "Distance", ylab = NULL)



## Part 2:

library(quanteda)  
library(quanteda.corpora)  
library(caret)

## Loading required package: lattice

sentiment <- tokens\_lookup(nostop\_corp\_movies, dictionary = data\_dictionary\_LSD2015[1:2],   
 exclusive = FALSE, nested\_scope = "dictionary")  
sent\_mat<-dfm(sentiment)  
sent\_df<- convert(sent\_mat[,c("negative","positive")], to = "data.frame")  
sent\_df<- cbind(sent\_df, docvars(sent\_mat))  
  
library(dplyr)  
sent\_df<- mutate(sent\_df, diff = positive - negative)  
sent\_df$newSent<- with(sent\_df, ifelse(diff>=0, "pos", "neg"))  
  
sent\_df<- select(sent\_df, c(sentiment, newSent))  
  
tab\_class <- table(sent\_df$sentiment, sent\_df$newSent)  
  
tab\_class

##   
## neg pos  
## neg 2279 3052  
## pos 999 4332

confusionMatrix(tab\_class, mode = "everything")

## Confusion Matrix and Statistics  
##   
##   
## neg pos  
## neg 2279 3052  
## pos 999 4332  
##   
## Accuracy : 0.6201   
## 95% CI : (0.6108, 0.6293)  
## No Information Rate : 0.6926   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.2401   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.6952   
## Specificity : 0.5867   
## Pos Pred Value : 0.4275   
## Neg Pred Value : 0.8126   
## Precision : 0.4275   
## Recall : 0.6952   
## F1 : 0.5294   
## Prevalence : 0.3074   
## Detection Rate : 0.2137   
## Detection Prevalence : 0.5000   
## Balanced Accuracy : 0.6410   
##   
## 'Positive' Class : neg   
##

## Part 3:

library(quanteda)  
library(quanteda.textmodels)  
library(quanteda.corpora)  
library(caret)  
  
set.seed(300)  
id\_train <- sample(1:10662, 7463, replace = FALSE)  
head(id\_train, 10)

## [1] 874 3650 789 553 9897 4557 2828 5479 10060 9004

corp\_movies$id\_numeric <- 1:ndoc(corp\_movies)  
  
dfmat\_training <- corpus\_subset(corp\_movies, id\_numeric %in% id\_train) %>%  
 dfm(remove = stopwords("english"), stem = TRUE)  
dfmat\_test <- corpus\_subset(corp\_movies, !id\_numeric %in% id\_train) %>%  
 dfm(remove = stopwords("english"), stem = TRUE)  
tmod\_nb <- textmodel\_nb(dfmat\_training, dfmat\_training$sentiment)  
summary(tmod\_nb)

##   
## Call:  
## textmodel\_nb.dfm(x = dfmat\_training, y = dfmat\_training$sentiment)  
##   
## Class Priors:  
## (showing first 2 elements)  
## neg pos   
## 0.5 0.5   
##   
## Estimated Feature Scores:  
## simplist , silli tedious . laddish juvenil teenag  
## neg 1.662e-04 0.05456 0.0007148 3.823e-04 0.08036 3.325e-05 8.312e-05 0.0001829  
## pos 6.464e-05 0.05947 0.0002424 6.464e-05 0.07885 1.616e-05 1.616e-05 0.0001454  
## boy possibl find funni exploit larg devoid  
## neg 0.0003823 0.0003657 0.0007813 0.00123 3.325e-04 0.0002660 2.494e-04  
## pos 0.0004040 0.0003717 0.0009050 0.00202 8.081e-05 0.0002424 4.848e-05  
## depth sophist make watch graphic treatment crime  
## neg 0.0003158 0.0001164 0.003109 0.001346 0.0001164 0.0001164 0.0004655  
## pos 0.0002424 0.0002424 0.003669 0.001099 0.0001131 0.0001454 0.0003232  
## bearabl [ garbus ] discard potenti patholog  
## neg 3.325e-05 0.001579 3.325e-05 0.001579 3.325e-05 0.0002826 6.649e-05  
## pos 1.616e-05 0.001487 1.616e-05 0.001487 3.232e-05 0.0001778 8.081e-05  
## studi  
## neg 0.0001995  
## pos 0.0004363

dfmat\_matched <- dfm\_match(dfmat\_test, features = featnames(dfmat\_training))  
  
actual\_class <- dfmat\_matched$sentiment  
predicted\_class <- predict(tmod\_nb, newdata = dfmat\_matched)  
tab\_class <- table(actual\_class, predicted\_class)  
  
tab\_class

## predicted\_class  
## actual\_class neg pos  
## neg 1230 390  
## pos 357 1222

confusionMatrix(tab\_class, mode = "everything")

## Confusion Matrix and Statistics  
##   
## predicted\_class  
## actual\_class neg pos  
## neg 1230 390  
## pos 357 1222  
##   
## Accuracy : 0.7665   
## 95% CI : (0.7514, 0.7811)  
## No Information Rate : 0.5039   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.533   
##   
## Mcnemar's Test P-Value : 0.2417   
##   
## Sensitivity : 0.7750   
## Specificity : 0.7581   
## Pos Pred Value : 0.7593   
## Neg Pred Value : 0.7739   
## Precision : 0.7593   
## Recall : 0.7750   
## F1 : 0.7671   
## Prevalence : 0.4961   
## Detection Rate : 0.3845   
## Detection Prevalence : 0.5064   
## Balanced Accuracy : 0.7666   
##   
## 'Positive' Class : neg   
##