BayesianSpamFilter

Robert Tedesco

9/26/2020

```
library(quanteda)
```

```
## Package version: 2.1.2
```

```
## Parallel computing: 2 of 16 threads used.
```

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

```
##
## Attaching package: 'quanteda'
```

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

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(quanteda.textmodels)
```

```
##
## Attaching package: 'quanteda.textmodels'
```

```
## The following object is masked from 'package:quanteda':
##
##     data_dfm_lbgexample
```

```
#spam and ham data pulled from: http://openclassroom.stanford.edu/MainFolder/DocumentPage.php?course=MachineLearning&doc=exe
rcises/ex6/ex6.html

setwd("C:\\Users\\rober\\Documents\\R\\nonspam-train")
nonspamtrain<-list.files()
for (file in nonspamtrain){
  if(!exists("nonspamtrainingset")){
    nonspamtrainingset<-readLines(file,warn=F)
  }
  if (exists("nonspamtrainingset")){
    tempnonspamtrainingset<-readLines(file,warn=F)
    nonspamtrainingset<-rbind(nonspamtrainingset,tempnonspamtrainingset)
    rm(tempnonspamtrainingset)
  }
}
label1<-rep("ham",351)
nonspamtrainingset<-cbind(nonspamtrainingset,label1)
```

```
spamtrain<-list.files()
for (file in spamtrain){
  if(!exists("spamtrainingset")){
    spamtrainingset<-readLines(file,warn=F)
  }
  if (exists("spamtrainingset")){
    tempspamtrainingset<-readLines(file,warn=F)
    spamtrainingset<-rbind(spamtrainingset,tempspamtrainingset)
    rm(tempspamtrainingset)
  }
}
label2<-rep("spam",351)
spamtrainingset<-cbind(spamtrainingset,label2)
```

```
spamtest<-list.files()
for (file in spamtest){
  if(!exists("spamtestingset")){
    spamtestingset<-readLines(file,warn=F)
  }
  if (exists("spamtestingset")){
    tempspamtestingset<-readLines(file,warn=F)
    spamtestingset<-rbind(spamtestingset,tempspamtestingset)
    rm(tempspamtestingset)
  }
}
label3<-rep("spam",131)
spamtestingset<-cbind(spamtestingset,label3)
```

```
nonspamtest<-list.files()
for (file in nonspamtest){
  if(!exists("nonspamtestingset")){
    nonspamtestingset<-readLines(file,warn=F)
  }
  if (exists("nonspamtestingset")){
    tempnonspamtestingset<-readLines(file,warn=F)
    nonspamtestingset<-rbind(nonspamtestingset,tempnonspamtestingset)
    rm(tempnonspamtestingset)
  }
}
label4<-rep("ham",131)
nonspamtestingset<-cbind(nonspamtestingset,label4)
```

```
#Formatting training set with Quanteda: turning vector of emails into list of words with the class attached.
library(readtext)
library(RColorBrewer)
trainingset<-rbind(nonspamtrainingset,spamtrainingset)
trainingset<-as.data.frame(trainingset)
labels<-c(label1,label2)
names(trainingset)<-c("message","type")
table(trainingset$type)
```

```
##
##  ham spam
##  351  351
```

```
msg.corpus<-corpus(trainingset$message)
docvars(msg.corpus,"type")<-trainingset$type


#Wordcloud plot for Ham
spam.plot<-corpus_subset(msg.corpus, type=="spam")
spam.plot<-dfm(spam.plot, tolower = TRUE, remove_punct = TRUE, remove_twitter = TRUE, remove_numbers = TRUE, remove=stopword
s("SMART"))
```

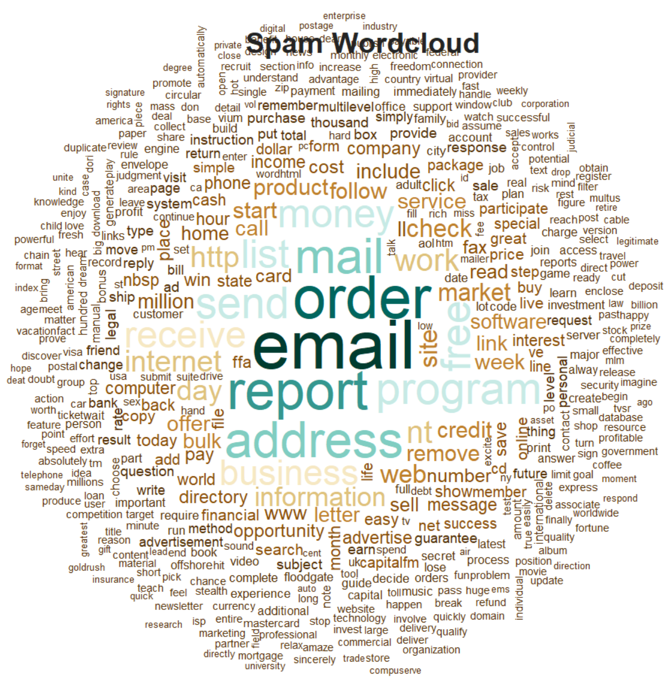
```
## Warning: 'remove_twitter' is defunct; see 'quanteda Tokenizers' in ?tokens
```

```
## Warning: 'stopwords(language = "SMART")' is deprecated.
## Use 'stopwords(source = "smart")' instead.
## See help("Deprecated")
```

```
spam.col <- brewer.pal(10, "BrBG")
textplot_wordcloud(spam.plot, min.freq = 16, color = spam.col)
```

```
## Warning: min.freq is deprecated; use min_count instead
```

```
title("Spam Wordcloud", col.main = "grey14")
```




```
#Wordcloud plot for Ham
ham.plot<-corpus_subset(msg.corpus,type=="ham")
ham.plot<-dfm(ham.plot,tolower = TRUE, remove_punct = TRUE, remove_twitter = TRUE, remove_numbers = TRUE,remove=c("gt", "lt"
, stopwords("SMART")))
```

```
## Warning: 'remove_twitter' is defunct; see 'quanteda Tokenizers' in ?tokens

## Warning: 'stopwords(language = "SMART")' is deprecated.
## Use 'stopwords(source = "smart")' instead.
## See help("Deprecated")
```

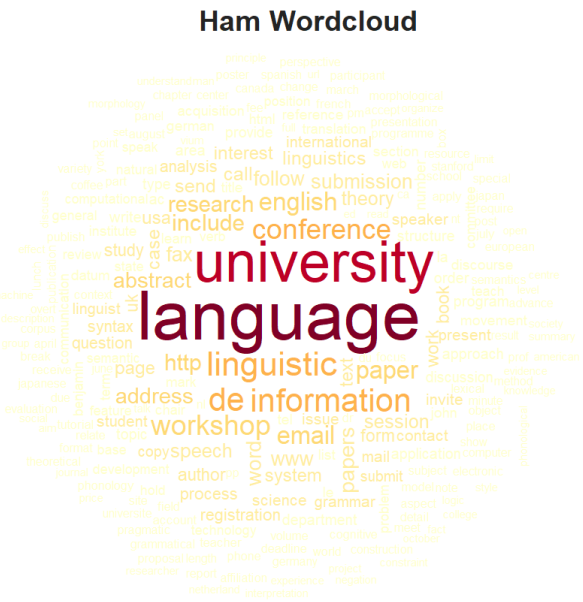
```
ham.col=brewer.pal(10, "YlOrRd")
```

```
## Warning in brewer.pal(10, "YlOrRd"): n too large, allowed maximum for palette YlOrRd is 9
## Returning the palette you asked for with that many colors
```

```
textplot_wordcloud(ham.plot,min.freq=50,colors=ham.col,fixed.asp=TRUE)
```

```
## Warning: min.freqcolorsfixed.asp is deprecated; use min_countcolorfixed_aspect
## instead
```

```
title("Ham Wordcloud",col.main = "grey14")
```




```
#Formatting testing set with Quanteda: combining vector emails into list of words
testingset<-rbind(spamtestingset,nonspamtestingset)
testingset<-as.data.frame(testingset)
labels<-c(label3,label4)
names(testingset)<-c("message","type")
table(testingset$type)
```

```
##
##  ham spam
##  131  131
```

```
test.corpus<-corpus(testingset$message)
docvars(test.corpus,"type")<-testingset$type
msg.dfm <- dfm(test.corpus, tolower = TRUE)

#generating document freq matrix for testing set
msg.dfm <- dfm_trim(msg.dfm, min_termfreq = 5, min_docfreq = 3)
msg.dfm <- dfm_weight(msg.dfm)
msg.dfm.test<-msg.dfm


#generating document freq matrix for training set
msg.dfm <- dfm(msg.corpus, tolower = TRUE)
msg.dfm <- dfm_trim(msg.dfm, min_termfreq = 5, min_docfreq = 3)
msg.dfm <- dfm_weight(msg.dfm)
head(msg.dfm)
```

```
## Document-feature matrix of: 6 documents, 4,296 features (99.0% sparse) and 1 docvar.
##        features
## docs    posting hi m work phonetics project modern irish hard source
##   text1       1  1 2    2         1       1      1     1    1      1
##   text2       1  1 2    2         1       1      1     1    1      1
##   text3       0  0 0    0         0       0      0     0    0      0
##   text4       0  0 0    2         0       2      0     0    0      0
##   text5       0  0 1    0         0       0      0     0    0      0
##   text6       0  0 0    0         0       0      0     0    0      0
## [ reached max_nfeat ... 4,286 more features ]
```

```
msg.dfm.train<-msg.dfm

#Naive Bayes Spam Filter!
nb.classifier<-textmodel_nb(msg.dfm.train,trainingset[,2])
nb.classifier
```

```
##
## Call:
## textmodel_nb.dfm(x = msg.dfm.train, y = trainingset[, 2])
##
##  Distribution: multinomial ; priors: 0.5 0.5 ; smoothing value: 1 ; 702 training documents;  fitted features.
```

```
pred<-predict(nb.classifier,msg.dfm.test,force=T)
```

```
## Warning: 95 features in newdata not used in prediction.
```

```
table(predicted=pred,actual=testingset[,2])
```

```
##          actual
## predicted ham spam
##      ham  126    2
##      spam   5  129
```

```
accuracy<-(126+129)/(262)
accuracy*100
```

```
## [1] 97.32824
```

```
#random forest, everything else,SVM,NN,Gam,Deep Learning, Log Reg, gradient boosting
```