Sentiment analysis

kiyana mirbaghestan

2023-03-12

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

library(caret)

## Warning: package 'caret' was built under R version 4.2.2

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.2.2

## Loading required package: lattice

library(quanteda)

## Warning: package 'quanteda' was built under R version 4.2.2

## Package version: 3.2.4  
## Unicode version: 13.0  
## ICU version: 69.1

## Parallel computing: 4 of 4 threads used.

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

library(quanteda.textmodels)

## Warning: package 'quanteda.textmodels' was built under R version 4.2.2

library(corpus)

## Warning: package 'corpus' was built under R version 4.2.2

library(tidytext)

## Warning: package 'tidytext' was built under R version 4.2.2

library(ggplot2)  
library(janeaustenr)

## Warning: package 'janeaustenr' was built under R version 4.2.2

library(stringr)

## Warning: package 'stringr' was built under R version 4.2.2

library(wordcloud)

## Warning: package 'wordcloud' was built under R version 4.2.2

## Loading required package: RColorBrewer

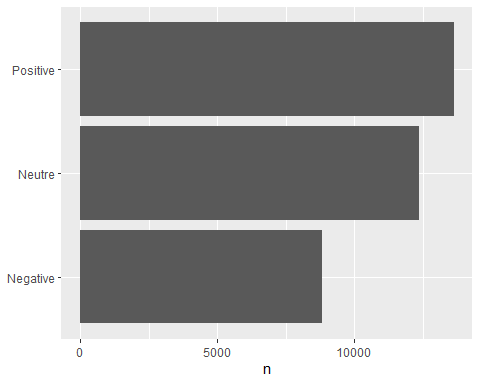
library(reshape2)

## Warning: package 'reshape2' was built under R version 4.2.2

text\_data<-read.csv("C:/Users/asus/Documents/sentiment\_analysis\_data.csv",stringsAsFactors = FALSE)  
length(which(!complete.cases(text\_data)))

## [1] 0

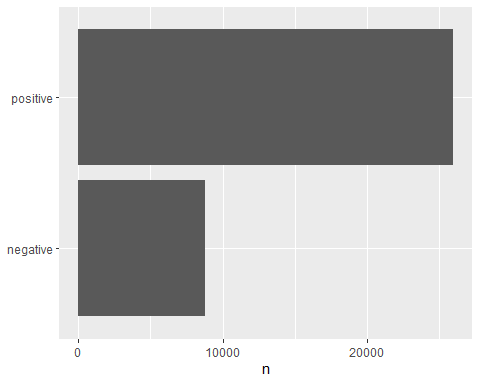
text\_data <- text\_data[,c("Text","sentiment")]  
  
colnames(text\_data) <- c( "text","Category")  
attach(text\_data)  
data <- corpus\_frame(Category,text)  
data2<-data.frame(Category,text)  
  
#Visualization of Titles   
data2%>%  
 count(Category,sort=TRUE)%>%  
 filter(n>10)%>%  
 mutate(word=reorder(Category,n))%>%  
 ggplot(aes(Category,n))+  
 geom\_col()+  
 xlab(NULL)+  
 coord\_flip()



text\_data$Category[text\_data$Category=="Negative"] <- "negative"   
text\_data$Category[text\_data$Category=="Positive"] <- "positive"   
text\_data$Category[text\_data$Category=="Neutre"] <- "positive"   
  
  
  
head(text\_data)

## text  
## 1 Why ?   
## 2 Sage Act upgrade on my to do list for tommorow.  
## 3 ON THE WAY TO MY HOMEGIRL BABY FUNERAL!!! MAN I HATE FUNERALS THIS REALLY SHOWS ME HOW BLESSED I AM   
## 4 Such an eye ! The true hazel eye-and so brilliant ! Regular features , open countenance , with a complexion , Oh ! What a bloom of full health , and such a pretty height and size ; such a firm and upright figure ! There is health , not merely in her bloom , but in her air , her head , her glance . One hears sometimes of a child being ' the picture of health ' ; now , she always gives me the idea of being the complete picture of grown-up health . She is loveliness itself .   
## 5 @Iluvmiasantos ugh babe.. hugggzzz for u .! babe naamazed nga ako e babe e, despite nega's mas pinaramdam at fil ko ang   
## 6 I'm expecting an extremely important phonecall any minute now #terror #opportunity  
## Category  
## 1 positive  
## 2 positive  
## 3 negative  
## 4 positive  
## 5 positive  
## 6 positive

#Visualization of Titles  
text\_data%>%  
 count(Category,sort=TRUE)%>%  
 filter(n>10)%>%  
 mutate(word=reorder(Category,n))%>%  
 ggplot(aes(Category,n))+  
 geom\_col()+  
 xlab(NULL)+  
 coord\_flip()



#Most common words  
data2 %>%  
 group\_by(Category) %>%  
 mutate(linenumber = row\_number(),  
 chapter = cumsum(str\_detect(text, regex("^chapter[\\divxlc]", ignore\_case = TRUE)))) %>%  
 ungroup()

## # A tibble: 34,792 × 4  
## Category text linen…¹ chapter  
## <chr> <chr> <int> <int>  
## 1 Neutre " Why ? " 1 0  
## 2 Neutre "Sage Act upgrade on my to do list for tommorow." 2 0  
## 3 Negative "ON THE WAY TO MY HOMEGIRL BABY FUNERAL!!! MAN I HA… 1 0  
## 4 Positive " Such an eye ! The true hazel eye-and so brilliant… 1 0  
## 5 Neutre "@Iluvmiasantos ugh babe.. hugggzzz for u .! babe … 3 0  
## 6 Positive "I'm expecting an extremely important phonecall any… 2 0  
## 7 Negative " .Couldnt wait to see them live. If missing them i… 2 0  
## 8 Neutre "maken Tip 2: Stop op een moment dat je het hele pr… 4 0  
## 9 Neutre "En dan krijg je ff een cadeautje van een tweep #me… 5 0  
## 10 Neutre " @1116am Drummer Boy bij op verzoek van @BiemOoste… 6 0  
## # … with 34,782 more rows, and abbreviated variable name ¹​linenumber

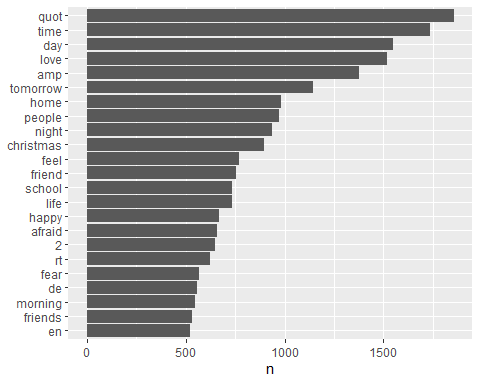
tidy<-data2 %>%  
 unnest\_tokens(word,text)  
data("stop\_words")  
tidy<-tidy%>%  
 anti\_join(stop\_words)

## Joining, by = "word"

##Most common words  
a<-tidy%>%  
 count(word,sort=TRUE)  
head(a)

## word n  
## 1 quot 1856  
## 2 time 1733  
## 3 day 1546  
## 4 love 1516  
## 5 amp 1378  
## 6 tomorrow 1146

tidy%>%  
 count(word,sort=TRUE)%>%  
 filter(n>500)%>%  
 mutate(word=reorder(word,n))%>%  
 ggplot(aes(word,n))+  
 geom\_col()+  
 xlab(NULL)+  
 coord\_flip()



tidy %>%  
 anti\_join(stop\_words) %>%  
 count(word) %>%  
 with(wordcloud(word, n, max.words = 100))

## Joining, by = "word"

## Warning in wordcloud(word, n, max.words = 100): quot could not be fit on page.  
## It will not be plotted.

## Warning in wordcloud(word, n, max.words = 100): christmas could not be fit on  
## page. It will not be plotted.

## Warning in wordcloud(word, n, max.words = 100): school could not be fit on page.  
## It will not be plotted.

## Warning in wordcloud(word, n, max.words = 100): friends could not be fit on  
## page. It will not be plotted.

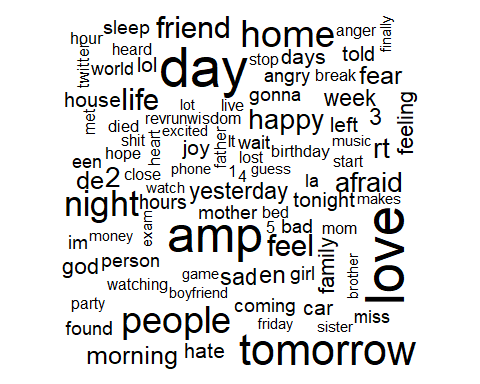
## Warning in wordcloud(word, n, max.words = 100): class could not be fit on page.  
## It will not be plotted.

## Warning in wordcloud(word, n, max.words = 100): moment could not be fit on page.  
## It will not be plotted.

## Warning in wordcloud(word, n, max.words = 100): time could not be fit on page.  
## It will not be plotted.

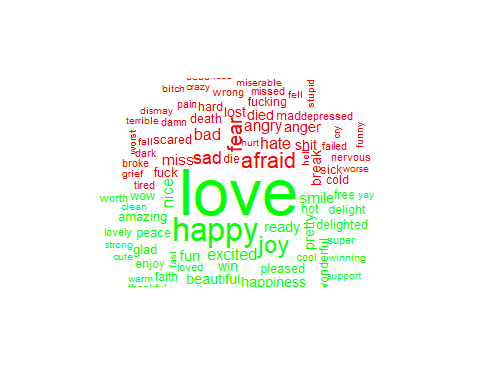
## Warning in wordcloud(word, n, max.words = 100): weekend could not be fit on  
## page. It will not be plotted.

## Warning in wordcloud(word, n, max.words = 100): parents could not be fit on  
## page. It will not be plotted.



tidy %>%  
 inner\_join(get\_sentiments("bing")) %>%  
 count(word, sentiment, sort = TRUE) %>%  
 acast(word ~ sentiment, value.var = "n", fill = 0) %>%  
 comparison.cloud(colors = c("red", "green"),  
 max.words = 100)

## Joining, by = "word"



set.seed(2012)  
text\_data<-text\_data[sample(nrow(text\_data)),]  
text\_corpus <- corpus(text\_data$text)  
# storing the label  
docvars(text\_corpus, "Category") <- text\_data$Category  
  
#separating Train and test data  
text.train<-text\_data[1:as.integer(nrow(text\_data) \* 0.7),]  
text.test<-text\_data[(as.integer(nrow(text\_data) \* 0.7)+1):nrow(text\_data),]  
  
text.dfm <- dfm(text\_corpus,   
 tolower = TRUE, #Lower case conversion  
 dfm\_wordstem = TRUE, # Stemming the words  
 remove\_punct = TRUE, # Remove punctuations   
 remove\_numbers = TRUE, # Remove numbers  
 remove\_symbols = TRUE,  
 remove = stopwords("english")  
) #generating document freq matrix

## Warning: 'dfm.corpus()' is deprecated. Use 'tokens()' first.

## Warning: dfm\_wordstem argument is not used.

## Warning: '...' should not be used for tokens() arguments; use 'tokens()' first.

## Warning: dfm\_wordstem argument is not used.  
  
## Warning: dfm\_wordstem argument is not used.

## Warning: 'remove' is deprecated; use dfm\_remove() instead

text.dfm

## Document-feature matrix of: 34,792 documents, 39,834 features (99.98% sparse) and 1 docvar.  
## features  
## docs just hate sentence end way think octopus apple add hdmi  
## text1 1 1 1 1 1 1 1 0 0 0  
## text2 1 0 0 0 0 0 0 1 1 1  
## text3 0 0 0 0 0 0 0 0 0 0  
## text4 0 0 0 0 0 0 0 0 0 0  
## text5 0 0 0 0 0 0 0 0 0 0  
## text6 0 0 0 0 0 0 0 0 0 0  
## [ reached max\_ndoc ... 34,786 more documents, reached max\_nfeat ... 39,824 more features ]

# Removing the sparse terms  
text.dfm <- dfm\_trim(text.dfm, sparsity = 0.97)  
text.dfm

## Document-feature matrix of: 34,792 documents, 12 features (96.03% sparse) and 1 docvar.  
## features  
## docs just s time tomorrow love today day get one like  
## text1 1 0 0 0 0 0 0 0 0 0  
## text2 1 0 0 0 0 0 0 0 0 0  
## text3 0 0 0 0 0 0 0 0 0 0  
## text4 0 0 0 0 0 0 0 0 0 0  
## text5 0 1 0 0 0 0 0 0 0 0  
## text6 0 0 1 0 0 0 0 0 0 0  
## [ reached max\_ndoc ... 34,786 more documents, reached max\_nfeat ... 2 more features ]

# Setting the term frequency according to its prevelance in the document  
text.dfm <- dfm\_tfidf(text.dfm, base = 2, scheme\_tf = "prop")   
text.dfm

## Document-feature matrix of: 34,792 documents, 12 features (96.03% sparse) and 1 docvar.  
## features  
## docs just s time tomorrow love today day get one like  
## text1 4.175076 0 0 0 0 0 0 0 0 0  
## text2 4.175076 0 0 0 0 0 0 0 0 0  
## text3 0 0 0 0 0 0 0 0 0 0  
## text4 0 0 0 0 0 0 0 0 0 0  
## text5 0 5.012327 0 0 0 0 0 0 0 0  
## text6 0 0 4.410511 0 0 0 0 0 0 0  
## [ reached max\_ndoc ... 34,786 more documents, reached max\_nfeat ... 2 more features ]

text.dfm.train <- text.dfm[1:as.integer(nrow(text.dfm) \* 0.7),]   
text.dfm.test <- text.dfm[(as.integer(nrow(text.dfm) \* 0.7)+1):nrow(text.dfm),]   
  
text.classifier.nb <- textmodel\_nb(text.dfm.train, text.train$Category,prior = "termfreq")   
text.classifier.svm <- textmodel\_svm(text.dfm.train, text.train$Category)  
text.classifier.svmlin <- textmodel\_svmlin(text.dfm.train, text.train$Category)  
  
text.predictions.nb <- data.frame(predict(text.classifier.nb, newdata = text.dfm.test) )  
conf.nb<- table(text.predictions.nb$predict.text.classifier.nb..newdata...text.dfm.test., text.test$Category)  
  
text.predictions.svm <- data.frame(predict(text.classifier.svm, newdata = text.dfm.test) )  
conf.svm<- table(text.predictions.svm$predict.text.classifier.svm..newdata...text.dfm.test., text.test$Category)  
  
text.predictions.svmlin <- data.frame(predict(text.classifier.svmlin, newdata = text.dfm.test) )  
conf.svmlin<- table(text.predictions.svmlin$predict.text.classifier.svmlin..newdata...text.dfm.test., text.test$Category)  
  
library(caret)  
library(e1071)

## Warning: package 'e1071' was built under R version 4.2.2

confusionMatrix(conf.nb,mode = "everything", positive = "positive")

## Confusion Matrix and Statistics  
##   
##   
## negative positive  
## negative 351 762  
## positive 2326 6999  
##   
## Accuracy : 0.7042   
## 95% CI : (0.6953, 0.7129)  
## No Information Rate : 0.7435   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.0407   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.9018   
## Specificity : 0.1311   
## Pos Pred Value : 0.7506   
## Neg Pred Value : 0.3154   
## Precision : 0.7506   
## Recall : 0.9018   
## F1 : 0.8193   
## Prevalence : 0.7435   
## Detection Rate : 0.6705   
## Detection Prevalence : 0.8934   
## Balanced Accuracy : 0.5165   
##   
## 'Positive' Class : positive   
##

confusionMatrix(conf.svm,mode = "everything", positive = "positive")

## Confusion Matrix and Statistics  
##   
##   
## negative positive  
## negative 0 0  
## positive 2677 7761  
##   
## Accuracy : 0.7435   
## 95% CI : (0.735, 0.7519)  
## No Information Rate : 0.7435   
## P-Value [Acc > NIR] : 0.5052   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 1.0000   
## Specificity : 0.0000   
## Pos Pred Value : 0.7435   
## Neg Pred Value : NaN   
## Precision : 0.7435   
## Recall : 1.0000   
## F1 : 0.8529   
## Prevalence : 0.7435   
## Detection Rate : 0.7435   
## Detection Prevalence : 1.0000   
## Balanced Accuracy : 0.5000   
##   
## 'Positive' Class : positive   
##

confusionMatrix(conf.svmlin,mode = "everything", positive = "positive")

## Confusion Matrix and Statistics  
##   
##   
## negative positive  
## negative 2374 6575  
## positive 303 1186  
##   
## Accuracy : 0.3411   
## 95% CI : (0.332, 0.3502)  
## No Information Rate : 0.7435   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.0224   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.1528   
## Specificity : 0.8868   
## Pos Pred Value : 0.7965   
## Neg Pred Value : 0.2653   
## Precision : 0.7965   
## Recall : 0.1528   
## F1 : 0.2564   
## Prevalence : 0.7435   
## Detection Rate : 0.1136   
## Detection Prevalence : 0.1427   
## Balanced Accuracy : 0.5198   
##   
## 'Positive' Class : positive   
##