Text Mining

2023-04-16

## Solution 1

setwd("C:\\Users\\user\\Documents\\HW6")  
library(jsonlite)  
library(tm)  
library(SnowballC)  
  
newsgroups <- fromJSON("newsgroups.json")  
# Extract content field  
newsgroups\_text <- unlist(lapply(newsgroups$content, paste, collapse = " "))  
# Create a Corpus  
newsgroups\_corpus <- Corpus(VectorSource(newsgroups\_text))  
# Define custom function for text pre-processing  
my\_preprocessor <- function(text) {  
 # Convert text to lowercase  
 text <- tolower(text)  
 # Remove punctuation  
 text <- removePunctuation(text)  
 # Remove numbers  
 text <- removeNumbers(text)  
 # Remove stopwords  
 text <- removeWords(text, stopwords("english"))  
 # Stem words  
 text <- stemDocument(text)  
 return(text)  
}  
  
## Corpus with pre-processor  
newsgroups\_corpus <- tm\_map (newsgroups\_corpus,my\_preprocessor)  
# Create a document-term matrix  
dtm <- DocumentTermMatrix(newsgroups\_corpus)  
  
# Create a term-document matrix  
tdm <- TermDocumentMatrix(newsgroups\_corpus)

## Solution 2

## Solution 2  
# Load required packages  
library(topicmodels)  
  
# Set number of topics  
num\_topics <- 4  
  
# Set the seed for reproducibility  
set.seed(123)  
  
# Fit the LDA model  
library(topicmodels)  
lda\_model <- LDA(dtm, k = 4)  
  
# Get the top 5 terms for each topic  
top\_terms <- terms(lda\_model, 5)  
top\_terms

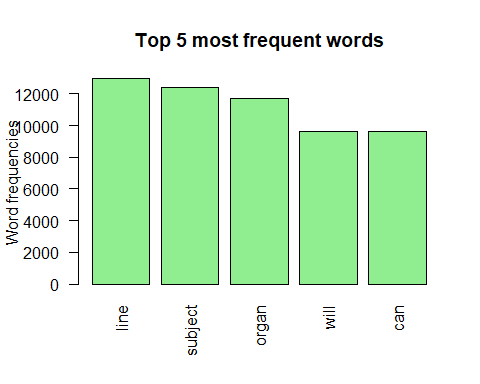
## Topic 1 Topic 2 Topic 3 Topic 4   
## [1,] "use" "line" "god" "maxaxaxaxaxaxaxaxaxaxaxaxaxaxax"  
## [2,] "line" "subject" "one" "peopl"   
## [3,] "subject" "organ" "subject" "will"   
## [4,] "organ" "write" "can" "one"   
## [5,] "can" "will" "line" "write"

## Solutiom 3

## Sentiment Analysis  
dtm\_m<-as.matrix(tdm)  
# Sort by descearing value of frequency  
dtm\_v <- sort(rowSums(dtm\_m),decreasing=TRUE)  
dtm\_d <- data.frame(word = names(dtm\_v),freq=dtm\_v)  
# 5 most frequent words  
head(dtm\_d, 5)

## word freq  
## line line 12960  
## subject subject 12434  
## organ organ 11687  
## will will 9649  
## can can 9619

# Plot the most frequent words  
barplot(dtm\_d[1:5,]$freq, las = 2, names.arg = dtm\_d[1:5,]$word,  
 col ="lightgreen", main ="Top 5 most frequent words",  
 ylab = "Word frequencies")



## Solution 4

library(udpipe)  
library(word2vec)  
library(text2vec)  
library(text)  
  
# Create CBOW model  
cbow\_model <- word2vec(  
 newsgroups\_text,  
 type = "cbow",  
 dim = 20,  
 iter = 20,  
 min\_count = 1  
)  
# Find top 5 terms nearest to "car" and "man"  
near\_can<-predict(cbow\_model,"can",5,type = "nearest");near\_can

## $can  
## term1 term2 similarity rank  
## 1 can could 0.9594413 1  
## 2 can should 0.9566046 2  
## 3 can might 0.9547368 3  
## 4 can won 0.9467642 4  
## 5 can couldn 0.9428374 5

near\_man<-predict(cbow\_model,"man",5,type = "nearest");near\_man

## $man  
## term1 term2 similarity rank  
## 1 man woman 0.9647167 1  
## 2 man neighbor 0.9279403 2  
## 3 man cop 0.9243380 3  
## 4 man heart 0.9210284 4  
## 5 man eyes 0.9205604 5

# Print top 5 terms for each word  
cat("Top 5 terms nearest to 'car': ", paste(near\_can, collapse = ", "), "\n")

## Top 5 terms nearest to 'car': list(term1 = c("can", "can", "can", "can", "can"), term2 = c("could", "should", "might", "won", "couldn"), similarity = c(0.959441304206848, 0.956604599952698, 0.954736769199371, 0.946764230728149, 0.942837417125702), rank = 1:5)

cat("Top 5 terms nearest to 'man': ", paste(near\_man, collapse = ", "), "\n")

## Top 5 terms nearest to 'man': list(term1 = c("man", "man", "man", "man", "man"), term2 = c("woman", "neighbor", "cop", "heart", "eyes"), similarity = c(0.964716672897339, 0.927940309047699, 0.924337983131409, 0.92102837562561, 0.920560359954834), rank = 1:5)

## Solution 5

# Create skip-gram model  
skipgram\_model <- word2vec(  
 newsgroups\_text,  
 type = "skip-gram",  
 dim = 20,  
 iter = 20,  
 min\_count = 1  
)  
# Find top 5 terms nearest to "religion" and "adult"  
top\_n <- 5  
near\_religion <- predict(skipgram\_model, "religion",5,type="nearest")  
near\_adult <- predict(skipgram\_model, "adult",5,type="nearest")  
  
# Print top 5 terms for each word  
cat("Top 5 terms nearest to 'religion': ", paste(near\_religion, collapse = ", "), "\n")

## Top 5 terms nearest to 'religion': list(term1 = c("religion", "religion", "religion", "religion", "religion"), term2 = c("antipathy", "belief", "marriage", "offshoots", "encroaching"), similarity = c(0.96811056137085, 0.964483976364136, 0.962613582611084, 0.962528467178345, 0.96252828836441), rank = 1:5)

cat("Top 5 terms nearest to 'adult': ", paste(near\_adult, collapse = ", "), "\n")

## Top 5 terms nearest to 'adult': list(term1 = c("adult", "adult", "adult", "adult", "adult"), term2 = c("categorization", "erhood", "incarcerating", "avowed", "eliminating"), similarity = c(0.957521319389343, 0.955641269683838, 0.941194355487823, 0.939946413040161, 0.938042640686035), rank = 1:5)