**##################### SENTIMENT ANALYSIS FOR CONGRESS ################################**

**################## using tidytext for sentiment analysis #############################**

**library(tidytext)**

**library(dplyr)**

**library(stringr)**

**library(tidyr)**

**library(ggplot2)**

**library(tm)**

**########################## three sentiment dictionaries ##############################**

**bing<- get\_sentiments("bing")**

**bing**

**bing[1:20,]**

**nrc<- get\_sentiments("nrc")**

**nrc**

**nrc[1:20,]**

**nrc\_joy<- get\_sentiments("nrc") %>%**

**filter(sentiment == "joy")**

**nrc\_joy**

**nrc\_joy[1:20,]**

**afinn<- get\_sentiments("afinn")**

**afinn**

**afinn[1:20,]**

**########################## three sentiment dictionaries ##############################**

**## loading the data (see Chapter 4)**

**load("C:\\Johannes Ledolter\\2020March01Book\\Chapter6WEB\\PrelimData.RData")**

**## using the document term matrix**

**corpus <- VCorpus(VectorSource(data),readerControl = list(reader = readPlain))**

**corpus1 <- tm\_map(corpus, stripWhitespace)**

**corpus2 <- tm\_map(corpus1, content\_transformer(tolower))**

**corpus3 <- tm\_map(corpus2, removePunctuation)**

**corpus4 <- tm\_map(corpus3, removeNumbers)**

**corpus5 <- tm\_map(corpus4, removeWords, stopwords("english"))**

**corp.dtm <- DocumentTermMatrix(corpus5,control=list(stemming=FALSE))**

**dim(corp.dtm)**

**## creating index for documents that include certain terms**

**## specified in comb – such as the word "laughter"**

**## to be used later**

**Bcorp.dtm=weightBin(corp.dtm) ## occurrences only**

**dim(Bcorp.dtm)**

**comb=c("laughter") ## words to select**

**comb**

**xx=dim(dim(Bcorp.dtm)[1])**

**for (i in 1: dim(Bcorp.dtm)[1]) {**

**xx[i]=0**

**}**

**for (i in 1:length(comb)) {**

**ind=labels(Bcorp.dtm)$Terms== comb[i]**

**if (sum(ind)!=0) xx=xx+as.matrix(Bcorp.dtm[,ind])**

**}**

**indexcomb=xx>0**

**table(index)**

**## using tidytext for sentiment analysis**

**speech=tidy(corp.dtm)**

**speech**

**colnames(speech)**

**colnames(speech)[colnames(speech)=="term"]="word"**

**colnames(speech)**

**speech**

**dim(speech)**

**## obtaining (bing) sentiment information for letter**

**out=speech %>%**

**inner\_join(get\_sentiments("bing")) %>%**

**count(document,wt=count,sentiment) %>%**

**spread(sentiment, n, fill = 0) %>%**

**mutate(sentiment = 100\*positive/(positive + negative))**

**out=as.matrix(out)**

**out[1:20,]**

**dim(out)**

**## stripping out meta variables (meta2, len, indexcomb)**

**## for speeches where both nupos=nuneg=0**

**number=as.numeric(out[,1])**

**meta2r=meta2[number]**

**lenr=len[number]**

**indexcombr=indexcomb[number]**

**dim(out)**

**length(meta2r)**

**length(lenr)**

**length(indexcombr)**

**## removing short speeches: considering speeches with length > limit**

**## reason: ratio pos/(pos+neg) is poorly estimated for short speeches**

**limit=1000**

**indlong=lenr>limit**

**outlong=out[indlong,]**

**lenlong=lenr[indlong]**

**meta2long=meta2r[indlong]**

**indexcomblong=indexcombr[indlong]**

**dim(outlong)**

**length(meta2long)**

**length(lenlong)**

**length(indexcomblong)**

**sent=as.numeric(outlong[,4])**

**pos=as.numeric(outlong[,3])**

**neg=as.numeric(outlong[,2])**

**## stratification of sentiment on frequent speaker**

**## consider the kkk most frequent speakers**

**kkk=30**

**frequent=dim(kkk)**

**frequency=dim(kkk)**

**average=dim(kkk)**

**tt=sort(table(meta2long),decreasing=TRUE)**

**for(i in 1:kkk) {**

**frequent[i]=labels(tt[i])**

**frequency[i]=tt[[i]]**

**}**

**frequent**

**frequency**

**for (i in 1:kkk) {**

**pp=sent[meta2long==frequent[i]]**

**average[i]=mean(pp,rm.na=TRUE) ## calculate proportion**

**}**

**average**

**hist(average)**

**table=data.frame(frequent,frequency,average)**

**print(table)**

**table1=table[order(table[,3],decreasing=TRUE),] ## ordered for positivity**

**table1**

**## stratification on the presence of laughter**

**table(indexcomblong)**

**t.test(sent ~ indexcomblong)**

**avesentwith=mean(sent[indexcomblong],rm.na=TRUE)**

**avesentwithout=mean(sent[!indexcomblong],rm.na=TRUE)**

**avesentwith**

**avesentwithout**

**## further analyses that could be considered**

**## select word window around the word laughter (such as +/- 50 words)**

**## save documents that include laughter with words around the word laughter**

**## analyze resulting text**

**##################### SENTIMENT ANALYSIS FOR CONGRESS ################################**

**################## using tidytext for sentiment analysis #############################**