|  |
| --- |
| library(data.table) |
|  | library(NLP) |
|  | library(openNLP) |
|  |  |
|  | tagPOS <- function(x, ...) { |
|  | s <- as.String(x) |
|  | word\_token\_annotator <- Maxent\_Word\_Token\_Annotator() |
|  | a2 <- Annotation(1L, "sentence", 1L, nchar(s)) |
|  | a2 <- annotate(s, word\_token\_annotator, a2) |
|  | a3 <- annotate(s, Maxent\_POS\_Tag\_Annotator(), a2) |
|  | a3w <- a3[a3$type == "word"] |
|  | POStags <- unlist(lapply(a3w$features, `[[`, "POS")) |
|  | POStagged <- paste(sprintf("%s/%s", s[a3w], POStags), collapse = " ") |
|  | list(POStagged = POStagged, POStags = POStags) |
|  | } |
|  |  |
|  | data<-fread("Reviews.csv") |
|  | dim(data) |
|  | data<-data[,c(-1,-4)] |
|  |  |
|  | helpful\_denom\_0\_ind<-which(data$HelpfulnessDenominator==0) |
|  | helpful\_denom\_0<-data[helpful\_denom\_0\_ind] |
|  |  |
|  | non\_helpful\_denom\_0<-data[-helpful\_denom\_0\_ind] |
|  | helpful\_ratio\_gt\_50<-non\_helpful\_denom\_0[non\_helpful\_denom\_0$HelpfulnessNumerator>=0.5\*non\_helpful\_denom\_0$HelpfulnessDenominator] |
|  | ###################################################################### |
|  |  |
|  | #preprocessing |
|  |  |
|  | ######################################################################## |
|  | data<-rbind(helpful\_ratio\_gt\_50,helpful\_denom\_0) |
|  | write.csv(data,"preprocessed\_reviews.csv",row.names = FALSE) |
|  | data<-fread("preprocessed\_reviews.csv") |
|  | head(data) |
|  | nrow(data) |
|  | train\_ind <- sample(1:nrow(data), size = .7\*nrow(data)) |
|  | train<-data[train\_ind,] |
|  | nrow(train) |
|  | test<-data[-train\_ind] |
|  | nrow(test) |
|  | write.csv(train,"train.csv",row.names = FALSE) |
|  | write.csv(test,"test.csv",row.names = FALSE) |
|  | head(train) |
|  | good\_bad<-train[,c(5,7)] |
|  | head(good\_bad) |
|  | dim(data) |
|  | data[1,6] |
|  |  |
|  |  |
|  | ####################################################################### |
|  |  |
|  | #Extract words |
|  |  |
|  | ###################################################################### |
|  | goodwords<-c() |
|  | count<-0 |
|  | for (i in (good\_bad[good\_bad$Score>3]$Summary)) |
|  | { |
|  | words<-strsplit(i,'[ /",\\().!?:;-]') |
|  | goodwords<-c(goodwords,words[[1]]) |
|  | print(count) |
|  | count=count+1 |
|  | } |
|  | goodwords |
|  | badwords<-c() |
|  | for (i in good\_bad[good\_bad$Score<3]$Summary) |
|  | { |
|  | words<-strsplit(i,'[ /",\\().!?:;-]') |
|  | badwords<-c(badwords,words[[1]]) |
|  | } |
|  | modwords<-c() |
|  | modwords |
|  | for (i in good\_bad[good\_bad$Score==3]$Summary) |
|  | { |
|  | print(i) |
|  | words<-strsplit(i,'[ /",\\().!?:;-]') |
|  | modwords<-c(modwords,words[[1]]) |
|  |  |
|  | } |
|  |  |
|  | ############################################### |
|  | #LOWER-CASE |
|  | ############################################ |
|  | goodwords\_case<-sapply(goodwords,tolower) |
|  | modwords\_case<-sapply(modwords,tolower) |
|  | badwords\_case<-sapply(badwords,tolower) |
|  | ############################################## |
|  | #UNIQUE |
|  | ############################################### |
|  | goodwords<-unique(goodwords\_case) |
|  | badwords<-unique(badwords\_case) |
|  | modwords<-unique(modwords\_case) |
|  | ############################################# |
|  | #INTERSECT AND UNION AND DIFF |
|  | ########################################## |
|  | inter\_gb<-intersect(goodwords,badwords) |
|  | inter\_gm<-intersect(goodwords,modwords) |
|  | inter\_bm<-intersect(badwords,modwords) |
|  | u\_gb\_bm<-union(inter\_gb,inter\_gm) |
|  | u\_gb\_bm<-union(u\_gb\_bm,inter\_bm) |
|  | final\_goodwords<-setdiff(goodwords,u\_gb\_bm) |
|  | final\_badwords<-setdiff(badwords,u\_gb\_bm) |
|  | final\_modwords<-setdiff(modwords,u\_gb\_bm) |
|  | final\_goodwords |
|  | final\_modwords |
|  | final\_badwords |
|  | class(final\_modwords) |
|  | final\_goodwords1<-data.frame() |
|  | ######################################################### |
|  | #GREATER THAN 3 |
|  | ########################################################## |
|  | for( i in (1:length(final\_goodwords))) |
|  | { |
|  | if(nchar(final\_goodwords[i])>=3) |
|  | { |
|  |  |
|  | final\_goodwords1<-c(final\_goodwords1,final\_goodwords[i]) |
|  |  |
|  | } |
|  | } |
|  | final\_goodwords1 |
|  | final\_badwords1<-data.frame() |
|  | for( i in (1:length(final\_badwords))) |
|  | { |
|  | if(nchar(final\_badwords[i])>=3) |
|  | { |
|  |  |
|  | final\_badwords1<-c(final\_badwords1,final\_badwords[i]) |
|  |  |
|  | } |
|  | } |
|  | #run from here |
|  | final\_modwords1<-data.frame() |
|  | for( i in (1:length(final\_modwords))) |
|  | { |
|  | if(nchar(final\_modwords[i])>=3) |
|  | { |
|  |  |
|  | final\_modwords1<-c(final\_modwords1,final\_modwords[i]) |
|  |  |
|  | } |
|  | } |
|  | final\_badwords1 |
|  | final\_goodwords<-final\_goodwords1 |
|  | final\_badwords<-final\_badwords1 |
|  | final\_modwords<-final\_modwords1 |
|  | ########################################################################## |
|  | #3-LETTERS |
|  | ######################################################################### |
|  | final\_modwords1 |
|  | for(i in (1:length(final\_modwords))){ |
|  | final\_modwords[i]<-gsub('([[:alpha:]])\\1\\1+',"\\1\\1\\1",final\_modwords[i]) |
|  | } |
|  |  |
|  | final\_modwords[1] |
|  |  |
|  | final\_badwords |
|  | for(i in (1:length(final\_badwords))){ |
|  | final\_badwords[i]<-gsub('([[:alpha:]])\\1\\1+',"\\1\\1\\1",final\_badwords[i]) |
|  | } |
|  | final\_goodwords |
|  | for(i in (1:length(final\_goodwords))){ |
|  | final\_goodwords[i]<-gsub('([[:alpha:]])\\1\\1+',"\\1\\1\\1",final\_goodwords[i]) |
|  | } |
|  | #################################################################################### |
|  | #PRONOUN |
|  | #################################################################################### |
|  | final\_modwords1<-data.frame() |
|  | final\_modwords1 |
|  | final\_goodwords1<-data.frame() |
|  | final\_badwords1<-data.frame() |
|  | tagged\_str <- tagPOS(final\_goodwords) |
|  | s<-tagged\_str$POStags |
|  | for(i in 1:length(final\_goodwords)) |
|  | { |
|  | if(s[i]!="DT" && s[i]!="CD" && s[i]!="EX" && s[i]!="IN" && s[i]!="PDT" && s[i]!="POS"&& s[i]!="TO" ) |
|  | final\_goodwords1<-c(final\_goodwords1,final\_goodwords[i]) |
|  | } |
|  | tagged\_str <- tagPOS(final\_modwords) |
|  | s<-tagged\_str$POStags |
|  | for(i in 1:length(final\_modwords)) |
|  | { |
|  | if(s[i]!="DT" && s[i]!="CD" && s[i]!="EX" && s[i]!="IN" && s[i]!="PDT" && s[i]!="POS"&& s[i]!="TO" ) |
|  | final\_modwords1<-c(final\_modwords1,final\_modwords[i]) |
|  | } |
|  | tagged\_str <- tagPOS(final\_badwords) |
|  | s<-tagged\_str$POStags |
|  | for(i in 1:length(final\_badwords)) |
|  | { |
|  | if(s[i]!="DT" && s[i]!="CD" && s[i]!="EX" && s[i]!="IN" && s[i]!="PDT" && s[i]!="POS"&& s[i]!="TO" ) |
|  | final\_badwords1<-c(final\_badwords1,final\_badwords[i]) |
|  | } |
|  | final\_badwords1 |
|  | final\_badwords |
|  | final\_goodwords<-final\_goodwords1 |
|  | final\_badwords<-final\_badwords1 |
|  | final\_modwords<-final\_modwords1 |
|  | length(final\_goodwords) |
|  | length(final\_modwords) |
|  | length(final\_badwords) |
|  | data\_pred<-c() |
|  | dim(test)[1] |
|  | ################################################################################################ |
|  | #ACCURACY |
|  | ############################################################################################# |
|  | for( i in 1:dim(test)[1]) |
|  | { |
|  | print (i) |
|  | #SPLIT |
|  | two<-strsplit(test[i,7][[1]],'[ /",\\().!?:;-]') |
|  | #LOWER |
|  | two<-sapply(two,tolower) |
|  | #UNIQUE |
|  | two<-unique(two) |
|  |  |
|  | two1<-data.frame() |
|  | #GREATER THAN 3 |
|  | for( j in (1:length(two))) |
|  | { |
|  | if(nchar(two[j])>=3) |
|  | { |
|  |  |
|  | two1<-c(two1,two[j]) |
|  | } |
|  | } |
|  | #3-LETTER |
|  | if(length(two1)>0){ |
|  | for(k in (1:length(two1))){ |
|  | #print(two1[k]) |
|  | two1[k]<-gsub('([[:alpha:]])\\1\\1+',"\\1\\1\\1",two1[k]) |
|  | # print(two1[k]) |
|  | } |
|  | } |
|  | final\_badwords |
|  | two<-two1 |
|  |  |
|  |  |
|  | #COMPARE WITH GOOD BAD |
|  | if((length(intersect(two1,final\_goodwords))%%2)>=(length(intersect(two1,final\_badwords))%%2)) |
|  | { |
|  | if((length(intersect(two1,final\_goodwords))%%2)>=(length(intersect(two1,final\_modwords))%%2)) |
|  | data\_pred[i]="good" |
|  | else |
|  | data\_pred[i]="mod" |
|  |  |
|  | } |
|  | else if((length(intersect(two1,final\_goodwords))%%2)<=(length(intersect(two1,final\_badwords))%%2)) |
|  | { |
|  | if((length(intersect(two1,final\_modwords))%%2)<=(length(intersect(two1,final\_badwords))%%2)) |
|  | data\_pred[i]="bad" |
|  |  |
|  | } |
|  | if(length(intersect(two1,final\_badwords))%%2>0) |
|  | data\_pred[i]="bad" |
|  |  |
|  |  |
|  | } |
|  | check<-test[,5] |
|  | check |
|  | ############################################################################ |
|  |  |
|  | #POST ACCURACY |
|  |  |
|  | ############################################################################### |
|  | unique(data\_pred) |
|  | check[check==1]<-10 |
|  | check[check==2]<-10 |
|  | check[check==3]<-20 |
|  | check[check==5]<-30 |
|  | check[check==4]<-30 |
|  | check |
|  | check\_pred<-data\_pred |
|  | check\_pred |
|  | data\_pred[data\_pred=="good"]<-30 |
|  | data\_pred[data\_pred=="mod"]<-20 |
|  | data\_pred[data\_pred=="bad"]<-10 |
|  | length(data\_pred) |
|  | data\_pred |
|  | length(check$Score) |
|  | #ACCURACY |
|  | accuracy<-table(pred=data\_pred,true=check$Score) |
|  | accuracy |
|  | print (((accuracy[1]+accuracy[5]+accuracy[9])/dim(test)[1])\*100) |
|  |  |
|  | #WRITE TO FILE |
|  | textfile<-file("good.txt") |
|  | for(j in final\_goodwords) |
|  | { |
|  |  |
|  | cat(j, file="good.txt", append=TRUE, sep = "\n") |
|  | } |
|  | close(textfile) |
|  | textfile<-file("bad.txt") |
|  | for(j in final\_badwords) |
|  | { |
|  | cat(j, file="bad.txt", append=TRUE, sep = "\n") |
|  | } |
|  | close(textfile) |
|  | textfile<-file("mod.txt") |
|  | for(j in final\_modwords) |
|  | { |
|  | if(length(j)>=3) |
|  | cat(j, file="mod.txt", append=TRUE, sep = "\n") |
|  | } |
|  | close(textfile) |