R Code for Text-Mining

################# ***Installing required packages to work with HTML files*** ################

install.packages("RTidyHTML")

library(RCurl)

library(XML)

############### ***Storing the URL*** ####################################################

u <- "<http://www.sec.gov/Archives/edgar/data/24741/000119312511064828/ddef14a.htm>"

############## ***Storing the raw data from HTML*** ######################################

doc.raw <- getURL(u)

############## ***Parsing the HTML data using HTMLTreeParse function*** #####################

html <- htmlTreeParse(doc.raw, useInternal = TRUE)

txt <- xpathApply(html, "//body//text()[not(ancestor::script)][not(ancestor::style)][not(ancestor::noscript)]", xmlValue)

############ ***Un-listing the data***####################################################

txt <- unlist(txt)

########### ***Finding the start and end string of the QA-section*** ###########################

qa\_sentence\_start <- "About the Meeting"

grep(qa\_sentence\_start, txt, ignore.case = FALSE) # Use grep function to search the start text

qa\_sentence\_end <- "proxy statement of Annual Report if you are receiving"

grep(qa\_sentence\_end,txt, ignore.case = TRUE)

######## ***Assign the gathered QA text to text* *vector*** #####################################

txt <- txt[328:817]

######## ***Convert the Data Vector to String***##############################################

string <- toString(txt)

####### ***Using gsub function to remove punctuations and special characters*** ###########

finalstring<-gsub("[[:punct:]]|\n|Â|â˜|â—|\t","",string)

print(finalstring)

####### ***Export the cleansed data to text file*** ##########################################

write.table(finalstring,"E:\\ITM\\Data Analytics (ITMD-527)\\week14\\corning2013.txt")

############ ***Installing necessary packages*** ###########################################

install.packages("class")

install.packages("plyr")

install.packages("tm")

library(tm)

library(plyr)

library(class)

################### ***Path to fetch all DEF-14 Files*** #######################################

path<-"C:\\Users\\Jash\\Documents\\R\\MyPrograms\\Corning"

options(stringsAsFactors = FALSE)

##### ***Using the corpus function to collect the required text from all cleaned DEF-14A files*** ########

s.corp<-Corpus(DirSource(directory = path,encoding = "utf-8"))

############# ***Processing the data to remove anomalies*** ##################################

s.corp.test<-tm\_map(s.corp,removePunctuation)

s.corp.test<-tm\_map(s.corp.test, stripWhitespace)

s.corp.test<-tm\_map(s.corp.test, content\_transformer(tolower))

s.corp.test<-tm\_map(s.corp.test,removeWords,stopwords("english"))

####### ***Creating Term Document Matrix*** ###############################################

s.tdm<-TermDocumentMatrix(s.corp.test)

result<-inspect(s.tdm)

################### ***Taking* *Transpose of result*** #######################################

result1<-t(inspect(s.tdm))

########## ***Installing "xlsx" package to write into Excel File*** ###############################

install.packages("xlsx")

library(xlsx)

write.xlsx(result,"C:\\Users\\Jash\\Documents\\R\\MyPrograms\\Corning\_Word\_Freq\\frequecyWords1.xlsx")

####### ***Frequency of words for all fillings*** #############################################

freq<-rowSums(as.matrix(result))

###### ***Matrix for word frequency and sorting them in descending order as per occurrence*** #######

a<-as.matrix(result)

ab<-sort(rowSums(a),decreasing=TRUE)

d <- data.frame(word = names(ab),freq=ab)

set.seed(1234)

########## **Installing SnowballC and wordcloud package to create word cloud** ##############

install("SnowballC")

install.packages("wordcloud")

library(wordcloud)

library(snowballC)

########## ***Creating word cloud*** ###################################################

wordcloud(words = d$word, freq = d$freq, min.freq = 1,max.words=100, random.order=FALSE, rot.per=0.35,colors=brewer.pal(8, "Dark2"))