#setup the working directory

setwd("")

install.packages("twitteR")

library(twitteR)

#retreat the saved tweets from Lab Session 1

nk.tweets<-readRDS("data.nk")

#extract the tweet text

nk.text=lapply(nk.tweets,function(t) t$getText())

#remove some non-english character

nk.text <- sapply(nk.text,function(row) iconv(row, "latin1", "ASCII", sub=""))

#loading Hu Liu's opinion lexicon

hu.liu.pos <- scan("opinion\_lexicon\_english/positive-words.txt",what="character", comment.char="");

hu.liu.neg <- scan("opinion\_lexicon\_english/negative-words.txt",what="character", comment.char="");

#loading some industry-specific and/or especially emphatic terms

pos.words <- c(hu.liu.pos, 'prize')

neg.words <- c(hu.liu.neg, 'late')

#function for the score.sentiment

score.sentiment = function(sentences, pos.words, neg.words, .progress='none')

{

require(plyr)

require(stringr)

# we got a vector of sentences. plyr will handle a list

# or a vector as an "l" for us

# we want a simple array ("a") of scores back, so we use

# "l" + "a" + "ply" = "laply":

scores = laply(sentences, function(sentence, pos.words, neg.words) {

# clean up sentences with R's regex-driven global substitute, gsub():

sentence = gsub('[[:punct:]]', '', sentence)

sentence = gsub('[[:cntrl:]]', '', sentence)

sentence = gsub('\\d+', '', sentence)

# and convert to lower case:

sentence = tolower(sentence)

# split into words. str\_split is in the stringr package

word.list = str\_split(sentence, '\\s+')

# sometimes a list() is one level of hierarchy too much

words = unlist(word.list)

# compare our words to the dictionaries of positive & negative terms

pos.matches = match(words, pos.words)

neg.matches = match(words, neg.words)

# match() returns the position of the matched term or NA

# we just want a TRUE/FALSE:

pos.matches = !is.na(pos.matches)

neg.matches = !is.na(neg.matches)

# and conveniently enough, TRUE/FALSE will be treated as 1/0 by sum():

score = sum(pos.matches) - sum(neg.matches)

return(score)

}, pos.words, neg.words, .progress=.progress )

scores.df = data.frame(score=scores, text=sentences)

return(scores.df)

}

#testing the score.function by some tweets

sample=c("You're awesome and I love you", "i hate and hate and hate. so angry. die!")

result=score.sentiment(sample, pos.words, neg.words)

class(result)

result$score

#score the tweets

nk.scores=score.sentiment(nk.text,pos.words,neg.words,.progress='text')

class(nk.scores)

#save the score in a csv file

write.csv(nk.scores, "nk\_scores.csv")