Assignment 4

Prasanna_Rao_Sec 52

Monday, May 18, 2015

Assignment tries to measure sentiment analysis by 1) Measuring sentiment analysis of airline passenger tweets 2) Measuring the sentiment of the political leaders in India

1) Twitter feed: Using TwitteR package in R, JetBlue, Delta, United Airlines and American airlines tweets were scrapped, parsed and compared with Hu & Lui bag of positive and negative words. Few other words were also added to this existing list of positive and negative words

2) Sentiment scoring using simple model The sentences were parsed, split and the tweets sentiment was calculated as a simple occurrence of positive and negative words. A numeric score was given based on the difference of positive and negative words. Thus, larger the +ve numeric score, larger the positive sentiment and vice versa. This was then compared with the industry bench mark to measure the customer satisfaction.

```
library(twitteR)
```

```
## Warning: package 'twitteR' was built under R version 3.1.3
```

```
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 3.1.3
```

```
##
## Attaching package: 'plyr'
##
## The following object is masked from 'package:twitteR':
##
##     id
```

```
library(RCurl)
```

```
## Loading required package: bitops
```

```
library(Rcpp)
```

```
## Warning: package 'Rcpp' was built under R version 3.1.3
```

```
library(stringr)
library(ggplot2)
library(tm)
```

```
## Warning: package 'tm' was built under R version 3.1.3
```

```
## Loading required package: NLP
```

```
## Warning: package 'NLP' was built under R version 3.1.3

##
## Attaching package: 'NLP'
##
## The following object is masked from 'package:ggplot2':
##
##     annotate

library(doBy)

## Warning: package 'doBy' was built under R version 3.1.3

## Loading required package: survival
## Loading required package: splines

library(XML)

## Warning: package 'XML' was built under R version 3.1.3
```

3000 Tweets from 1) Delta 2) Jet Blue 3) United Airlines 4) American airlines

```
setup_twitter_oauth()

## [1] "Using direct authentication"

delta.tweets = searchTwitter('@delta', n=3000)
delta.text = laply(delta.tweets, function(t) t$getText())
delta.text=str_replace_all(delta.text,"[^[:graph:]]", " ")

american.tweets = searchTwitter('@AmericanAir', n=3000)
american.text = laply(american.tweets, function(t) t$getText())
american.text=str_replace_all(american.text,"[^[:graph:]]", " ")

jetblue.tweets = searchTwitter('@JetBlue', n=3000)
jetblue.text = laply(jetblue.tweets, function(t) t$getText())
jetblue.text=str_replace_all(jetblue.text,"[^[:graph:]]", " ")

united.tweets = searchTwitter('@united', n=3000)
united.text = laply(united.tweets , function(t) t$getText())
united.text=str_replace_all(united.text,"[^[:graph:]]", " ")
```

Sentiment scoring using bag Hu.Lui bag of words

```
 hu.liu.pos = scan('C:/Prasanna Krishna/Prasanna Krishna/MS/452/Individual As
signment41/R/positive-words.txt',
what='character', comment.char=';')

 hu.liu.neg = scan('C:/Prasanna Krishna/Prasanna Krishna/MS/452/Individual As
signment41/R/negative-words.txt',
what='character', comment.char=';')

pos.words = c(hu.liu.pos, 'upgrade','great',"excited",'thanks','thank')
```

```
neg.words = c(hu.liu.neg, 'wtf', 'wait', 'waiting','delay','mess','scary',
'epicfail', 'mechanical'," don't care",'not','what','cancelled')

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 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("[^[:alnum:] ]", ' ', sentence)
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+')

#print (sentence)
# 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)
```

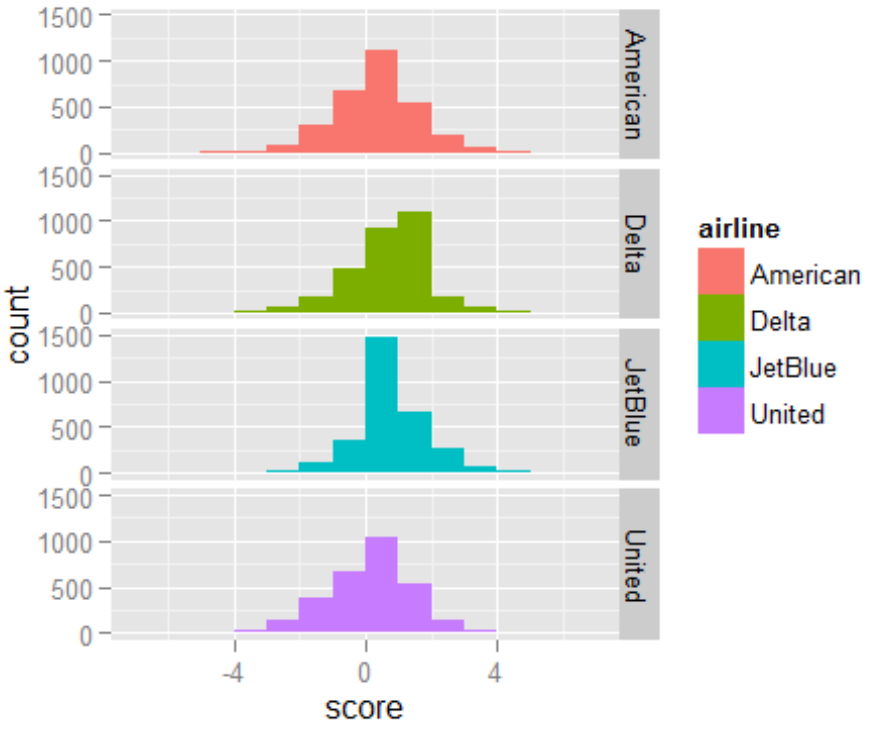
```
}, pos.words, neg.words, .progress=.progress )

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

delta.scores = score.sentiment(delta.text, pos.words,neg.words)
delta.scores$airline="Delta"
delta.scores$code="DL"

american.scores = score.sentiment(american.text, pos.words,neg.words)
american.scores$airline="American"
american.scores$code="AA"

jetblue.scores = score.sentiment(jetblue.text, pos.words,neg.words)
jetblue.scores$airline="JetBlue"
jetblue.scores$code="JB"

united.scores = score.sentiment(united.text, pos.words,neg.words)
united.scores$airline="United"
united.scores$code="UA"

combined_scores = rbind(delta.scores ,american.scores ,jetblue.scores , unite
d.scores)
```

gg plot cmparison of sentiment scores

```
g <-ggplot(data=combined_scores, mapping=aes(x=score, fill=airline) )
g <- g + geom_bar(binwidth=1)
g <- g + facet_grid(airline~.)
g
```

Prepare fr ACSI comparison

```
g <-ggplot(data=combined_scores, mapping=aes(x=score, fill=airline) )
g <- g + geom_bar(binwidth=1)
g <- g + facet_grid(airline~.)
g
```




```
combined_scores$very.pos.bool = combined_scores$score >= 2
combined_scores$very.neg.bool = combined_scores$score <= -2

combined_scores$very.pos = as.numeric( combined_scores$very.pos.bool )
combined_scores$very.neg = as.numeric( combined_scores$very.neg.bool )

twitter.df = ddply(combined_scores, c('airline', 'code'), summarise,
very.pos.count=sum( very.pos ),very.neg.count=sum( very.neg ) )

twitter.df$very.tot = twitter.df$very.pos.count +
twitter.df$very.neg.count

twitter.df$score = round( 100 * twitter.df$very.pos.count /
twitter.df$very.tot )

orderBy(~-score, twitter.df)
```

```
##    airline code very.pos.count very.neg.count very.tot score
## 3  JetBlue   JB            365            137      502    73
## 2    Delta   DL            262            242      504    52
## 1 American   AA            249            409      658    38
## 4   United   UA            204            556      760    27
```

```
#Preparing ASCI portal

acsi.url = 'http://www.theacsi.org/index.php?option=com_content&view=article&
id=147&catid=&Itemid=212&i=Airlines'
```

```
acsi.df = readHTMLTable(acsi.url, header=T, which=1, stringsAsFactors=F)

acsi.df = acsi.df[,c(1,23)]

colnames(acsi.df) = c('airline', 'score')

acsi.df$code = c('JB', NA, NA, NA, NA,
                 'DL', 'AA', 'DL', 'UA',NA,NA,NA,NA,NA)
acsi.df$score = as.numeric(acsi.df$score)

## Warning: NAs introduced by coercion

acsi.df

##               airline score code
## 1             JetBlue    81   JB
## 2           Southwest    78 <NA>
## 3              Alaska    75 <NA>
## 4          All Others    73 <NA>
## 5            Airlines    71 <NA>
## 6               Delta    71   DL
## 7            American    66   AA
## 8           Allegiant    65   DL
## 9              United    60   UA
## 10           Frontier    58 <NA>
## 11             Spirit    54 <NA>
## 12 Northwest Airlines    NA <NA>
## 13        Continental    NA <NA>
## 14         US Airways    NA <NA>

compare.df = merge(twitter.df, acsi.df, by=c('code', 'airline'),
suffixes=c('.twitter', '.acsi'))

orderBy(~-score.acsi, compare.df)

##   code  airline very.pos.count very.neg.count very.tot score.twitter
## 3   JB  JetBlue            365            137      502            73
## 2   DL    Delta            262            242      504            52
## 1   AA American            249            409      658            38
## 4   UA   United            204            556      760            27
##   score.acsi
## 3         81
## 2         71
## 1         66
## 4         60
```