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.

```r
library(twitteR)
```

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

```r
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
```

```r
library(RCurl)
```

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

```r
library(Rcpp)
```

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

```r
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
```

```r
library(doBy)
```

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

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

```r
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

```r
setup_twitter_oauth('yHTaKavjc9ZlK1f6DHy1ub4hu','v5ikU72GdsOrntl8JDrd8DcWxrN1jgBQuUnPutTcZZHJvO5kYo','2
```

```
## [1] "Using direct authentication"
```

Sentiment scoring using bag Hu.Lui bag of words

```r
 hu.liu.pos = scan('C:/Prasanna Krishna/Prasanna Krishna/MS/452/Individual Assignment41/R/positive-word
what='character', comment.char=';')

 hu.liu.neg = scan('C:/Prasanna Krishna/Prasanna Krishna/MS/452/Individual Assignment41/R/negative-word
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():
```

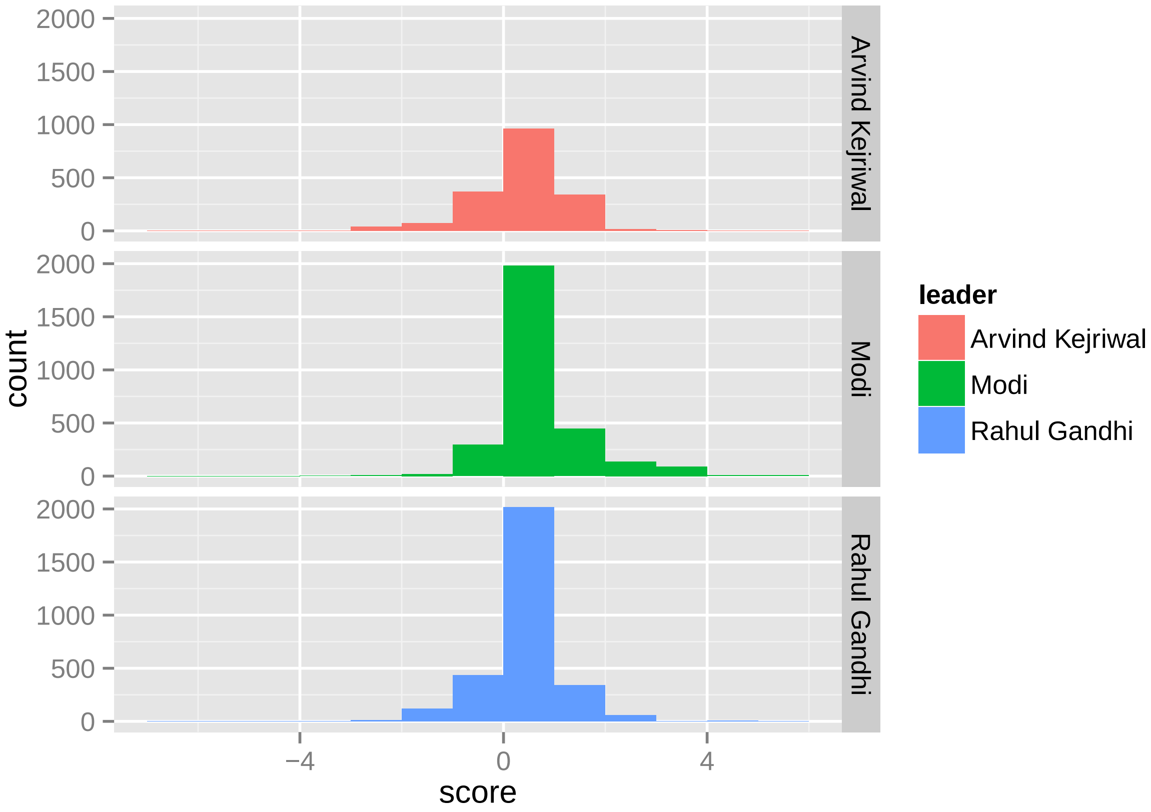
```
#sentence = gsub("[^[:alnum:] ]", ' ', sentence)
sentence = gsub('[[:punct:]]', '', sentence)
sentence = gsub('[[:cntrl:]]', '', sentence)
sentence = gsub('\\d+', '', sentence)
# and convert to lower case:
sentence = tolower(sentence)

# 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)
}
```

Prepare fr ACSI comparison

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


modi.scores = score.sentiment(modi.text, pos.words,neg.words)
modi.scores$leader="Modi"
modi.scores$code="MD"

ak.tweets = searchTwitter('#ArvindKejriwal', n=3000)
```

```
## Warning in doRppAPICall("search/tweets", n, params = params,
## retryOnRateLimit = retryOnRateLimit, : 3000 tweets were requested but the
## API can only return 1815
```

```
ak.text = laply(ak.tweets, function(t) t$getText())
ak.text=str_replace_all(ak.text,"[^[:graph:]]", " ")


ak.scores = score.sentiment(ak.text, pos.words,neg.words)
ak.scores$leader="Arvind Kejriwal"
ak.scores$code="AK"

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

rg.scores = score.sentiment(rg.text, pos.words,neg.words)
rg.scores$leader="Rahul Gandhi"
rg.scores$code="RG"


combined_score = rbind(modi.scores ,ak.scores,rg.scores)


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


2000
1500
1000
500
0
count
Arvind Kejriwal
Modi
Rahul Gandhi
−4
0
4
score
leader
🟥 Arvind Kejriwal
🟩 Modi
🟦 Rahul Gandhi