SFDC Topic Modeling and Sentiment

Ken Brooks

June 3, 2015

Adapted from Ted Kwartler (Ted@sportsanalytics.org (mailto:Ted@sportsanalytics.org)), Open Data Science Conference Workshop: Intro to Text Mining using R, 5-30-2015, v7.0 Topic Modeling and simple sentiment

```
#Set the working directory and import libraries
#setwd("~/Google Drive KB/Open Source Conf")

#libraries
library(tm)
```

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

```
library(topicmodels)
#install.packages('topicmodels')
library(portfolio)
```

```
## Loading required package: grid
```

```
## Loading required package: lattice
```

```
## Loading required package: nlme
```

```
#install.packages("portfolio")
#library(ggplot2)
#library(ggthemes)
library(plyr)
library(stringr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following object is masked from 'package:nlme':
##
##     collapse
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readxl)
```

Set options and defined functions

```
#options, functions
options(stringsAsFactors = FALSE)
Sys.setlocale('LC_ALL','C')
```

```
## [1] "C/C/C/C/C/en_US.UTF-8"
```

```
#try to lower function
tryTolower <- function(x){
  # return NA when there is an error
  y = NA
  # tryCatch error
  try_error = tryCatch(tolower(x), error = function(e) e)
  # if not an error
  if (!inherits(try_error, 'error'))
    y = tolower(x)
  return(y)
}

clean.corpus<-function(corpus){
  corpus <- tm_map(corpus, removePunctuation)
  corpus <- tm_map(corpus, stripWhitespace)
  corpus <- tm_map(corpus, removeNumbers)
  corpus <- tm_map(corpus, content_transformer(tryTolower))
  corpus <- tm_map(corpus, removeWords, custom.stopwords)
  return(corpus)
}

#Bigram token maker
bigram.tokenizer <-function(x)
  unlist(lapply(ngrams(words(x), 2), paste, collapse = " "), use.names = FALSE)

#Bring in subjective lexicons
pos <- readLines("data/positive_words.txt")
neg <-readLines("data/negative_words.txt")

#Simple sentiment subject word counter function, poached online
score.sentiment = function(sentences, pos.words, neg.words, .progress='none')
{
  scores = laply(sentences, function(sentence, pos.words, neg.words) {
    word.list = str_split(sentence, '\\s+')
    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)
    pos.matches = !is.na(pos.matches)
    neg.matches = !is.na(neg.matches)
    #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)
}
```

Create custom stop words

```
#Create custom stop words

custom.stopwords <- c(stopwords('english'), 'lol', 'smh')
```

Import and clean text, build dtm

```
#bring in some text
# text<-read.csv('data/SFDC_Survey.csv', header=TRUE)
text <-readxl::read_excel('data/LP Spring 2016 Student Survey- 3_Free Responses.xlsx')
col1 = "Comment"

#Create a clean corpus
corpus <- Corpus(DataframeSource(data.frame(text[[col1]])))
corpus <-clean.corpus(corpus)

#Make a DTM
dtm<-DocumentTermMatrix(corpus, control=list(tokenize=bigram.tokenizer))
```

Perform topic modeling

```
#In Topic Modeling, remove any docs with all zeros after removing stopwords
rowTotals <- apply(dtm , 1, sum)
dtm.new   <- dtm[rowTotals> 0, ]

#In Sentiment, to ensure the number of rows in the dtm.new and the sentiment data frame
 equal
text <-cbind(text,rowTotals)
text <- text[rowTotals> 0, ]

#Begin Topic Modeling; can use CTM or LDA
topic.model <- LDA(dtm.new, control = list(alpha = 0.1), k = 5)

#Topic Extraction
topics<-get_terms(topic.model, 5)
colnames(topics)<-c("topic1","topic2","topic3","topic4","topic5")
topics<-as.data.frame(topics)
t1<-paste(topics$topic1,collapse=' ')
t2<-paste(topics$topic2,collapse=' ')
t3<-paste(topics$topic3,collapse=' ')
t4<-paste(topics$topic4,collapse=' ')
t5<-paste(topics$topic5,collapse=' ')
topics
```

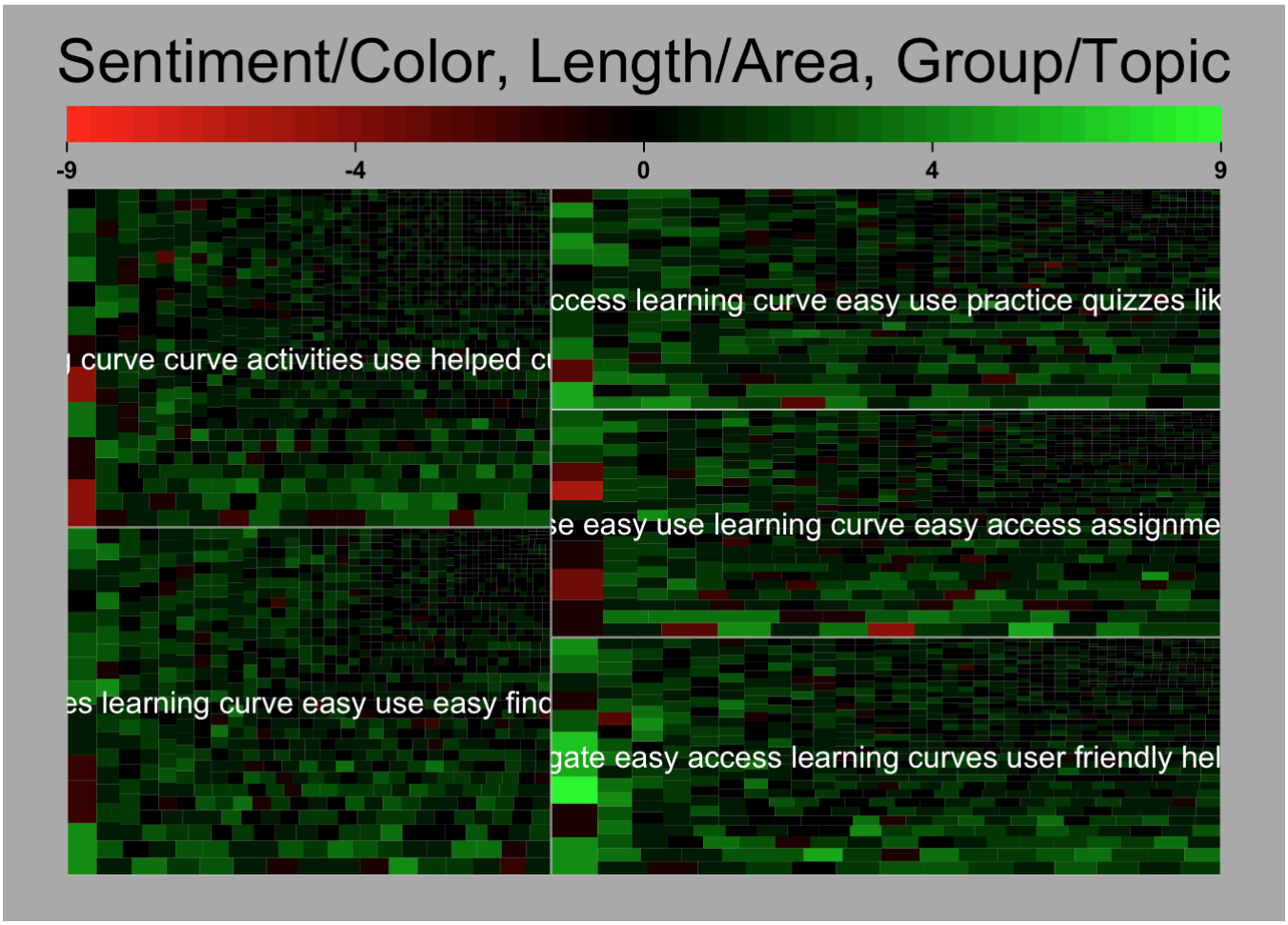
```
##              topic1          topic2           topic3          topic4
## 1          easy use        ease use      easy access learning curves
## 2    learning curve        easy use   learning curve  learning curve
## 3  curve activities  learning curve         easy use        easy use
## 4        use helped     easy access practice quizzes       easy find
## 5 curve assignments assignments due        like easy     easy access
##            topic5
## 1   easy navigate
## 2     easy access
## 3 learning curves
## 4   user friendly
## 5    helped study
```

Assign documents to topics

```
#Score each tweet's probability for the topic models then add the topic words to the df
 as headers
scoring<-posterior(topic.model)
scores<-scoring$topics
scores<-as.data.frame(scores)
colnames(scores)<-c(t1,t2,t3,t4,t5)

#The max probability of each tweet classifies the tweet document
topics.text<-as.data.frame(cbind(row.names(scores),apply(scores,1,function(x) names(scor
es)[which(x==max(x))])))
```

Perform sentiment scoring

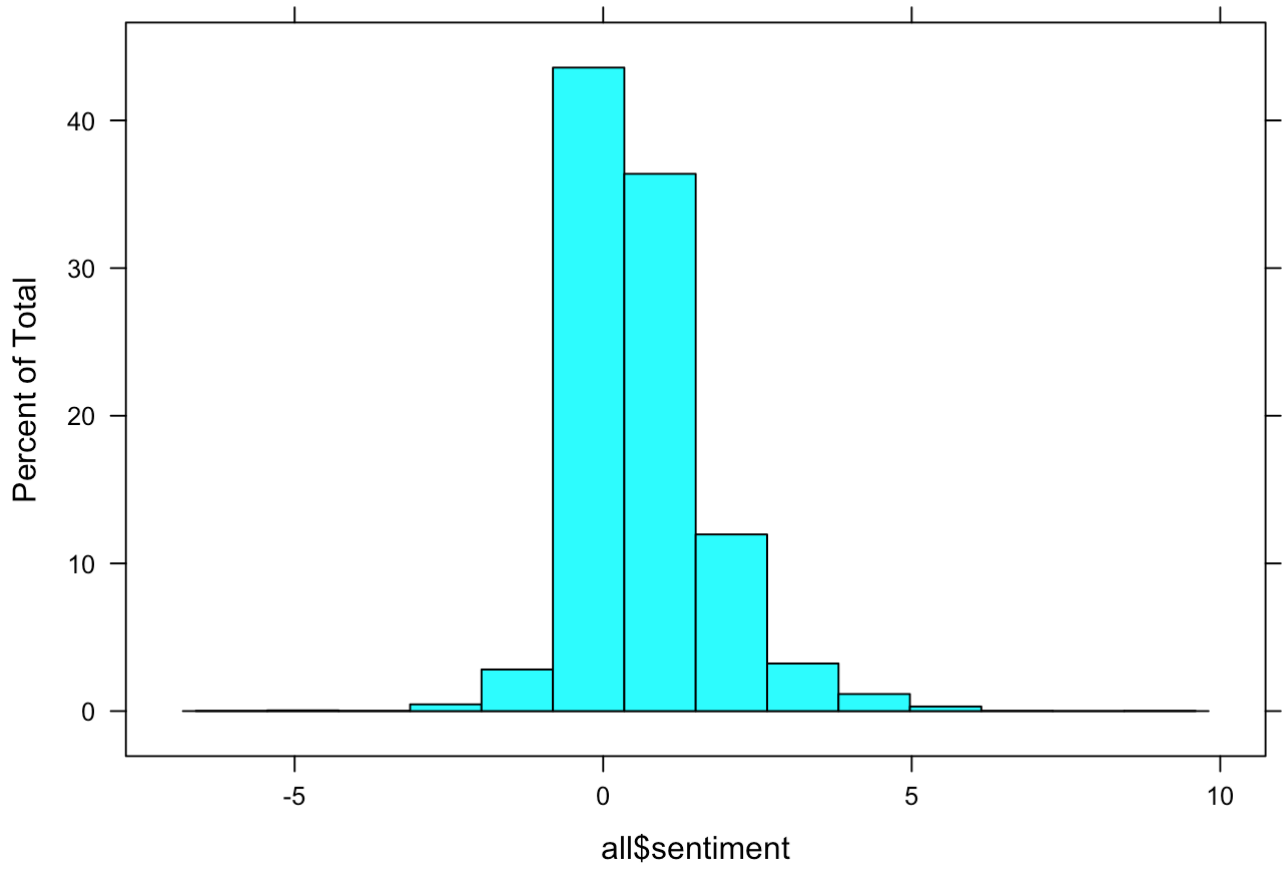
```
#Apply the subjective lexicon scoring function

# sentiment.score<-score.sentiment(text[[col1]], pos, neg, .progress='text')
sentiment.score<-score.sentiment(text[[col1]], pos, neg)

#Get the length of each doc by number of words not characters
doc.length<-rowSums(as.matrix(dtm.new))

#Create a unified data frame
all<-cbind(topics.text,scores,sentiment.score, doc.length)
names(all)[2]<-paste("topic")
names(all)[8]<-paste("sentiment")
names(all)[10]<-paste("length")
all[all == ""] <- NA

#Make the treemap
map.market(id=all$V1, area=all$length, group=all$topic, color=all$sentiment, main="Senti
ment/Color, Length/Area, Group/Topic")
```




```
#End
```

Sort comments with most negative on top and print them

```
histogram(all$sentiment)
```




```
sent.limit = -5

all %>% filter(sentiment <= sent.limit) %>% arrange(desc(sentiment)) %>% select(text)
```

```
##
                                                                                            t
ext
## 1 I don't like anything about this scam. Either pay full price for the book and acces
s, or pay damn near full price for access.  I literally learned nothing from launchpad,
 with exception of how useless and lazy Americans really are.  This product encourages s
tudents to be mediocre. I will drop any classes that require this absolutely useless res
ource.  I learned nothing this semester thanks to an instructor that relied heavily on t
his garbage product.  Thank you for cheating me out of education in the name of convenie
nce.
## 2
                        A complete fucking gimmick used by a stupid old man that thinks
 rascal scooters are cool. The school already has a quiz function built into their websi
te. The internet has study resources. This thing is USELESS, poorly organized, and unnee
ded.
## 3
                                    ebook comes with online activities. most other onlin
e activities that are paired with the books don't actually come with an ebook. so you're
 still stuck with buying a textbook and and having to purchase the ridiculously expensiv
e online modules, which you can't resell access to. having an ebook paired with the onli
ne stuff makes it a little more worthwile to pay for something still exorbitantly expens
ive
## 4
                    Nothing. Launchpad is among the most awful formats for studying/comp
leting homework, that I have ever used. It is unnecessarily difficult to use and a compl
ete step down from statsportal. It's frustrating as hell that my teacher can't do anythi
ng to help either because the settings she can edit are so limit. Whoever is responsible
 for this atrocity of a website should be fired immediately. Scrap this shit website ASA
P.
```

Plot sentiment over time

- Make sure data frame is in date order
- aggregate by week?
- plot time series
- add a loess trend line