

Twitter Sentiment Analysis- three Best picture Oscar nominated movies

Divya Thomas

March 10, 2019

loading data, converting to DF

```
library(twitteR)
```

```
## Warning: package 'twitteR' was built under R version 3.6.2
```

```
load("SentimentTweets.Rdata")  
StarBornDF = twListToDF(T1)  
BlackPantherDF = twListToDF(T2)  
GreenBookDF = twListToDF(T3)
```

Loading the opinion lexicon

```
pos=scan("positive-words.txt",what="character",comment.char=";")  
neg=scan("negative-words.txt",what="character",comment.char=";")  
length(pos)
```

```
## [1] 2006
```

```
length(neg)
```

```
## [1] 4783
```

sentiment score function

```

getSentimentScore=function(tweet_text,pos,neg)
{
  #removes retweet entities
  sentence=gsub("(RT|via)((?:\\b\\W*@[\\w+)+)", " ", tweet_text)

  #removes all @someone
  sentence=gsub("@\\w+", " ", sentence)

  # remove all the punctuation except apostrophe
  sentence=gsub("[[:punct:]]", "", sentence)

  # remove all the control chracters, Like \\n or \\r
  sentence=gsub("[[:cntrl:]]", "", sentence)

  # remove numbers, we need only text for analytics
  sentence=gsub("[[:digit:]]", "", sentence)

  #remove html links
  sentence=gsub("http\\w+", "", sentence)

  # remove unnecessary spaces (white spaces, tabs etc)
  sentence=gsub("[ \\t]{2,}", " ", sentence)
  sentence=gsub("^\\s+|\\s+$", "", sentence)

  # convert to lower case
  #sentence=iconv(sentence, "ASCII", "UTF-8", sub="")
  sentence=tolower(sentence)

  # split into words.
  word.list=strsplit(sentence, " ")

  # initialize vector to store score
  score=numeric(length(word.list))

  # Loop through each tweet
  for(i in 1:length(word.list))
  {
    # compare our words to the dictionaries of positive
    # & negative terms
    pos.matches=match(word.list[[i]], pos)
    neg.matches=match(word.list[[i]], neg)

    # 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[i]=sum(pos.matches)-sum(neg.matches)
  }
  return(score)
}

```

Finding sentiment scores for each movie.

```

StarBorn_text=StarBornDF$text
BlackPanther_text=BlackPantherDF$text
GreenBook_text=GreenBookDF$text

#calling function for each movie to get sent score
StarBorn_score<-getSentimentScore(StarBorn_text,pos,neg)
BlackPanther_score<-getSentimentScore(BlackPanther_text,pos,neg)
GreenBook_score<-getSentimentScore(GreenBook_text,pos,neg)

```

Plotting the distributions of sentiment scores. A vertical line is added to indicate the mean sentiment score. The score distributions are compared by placing them all in the same figure.

```

#plot 1-star is born
plot(density(StarBorn_score),main="Sentiment Analysis of 3 Oscar nominated movies", xlab="Sentiment Score",
      ylab="Density of tweets",col=1,lwd=2,lty=1)
abline(v=mean(StarBorn_score),col=1, lwd=2,lty=2) #mean

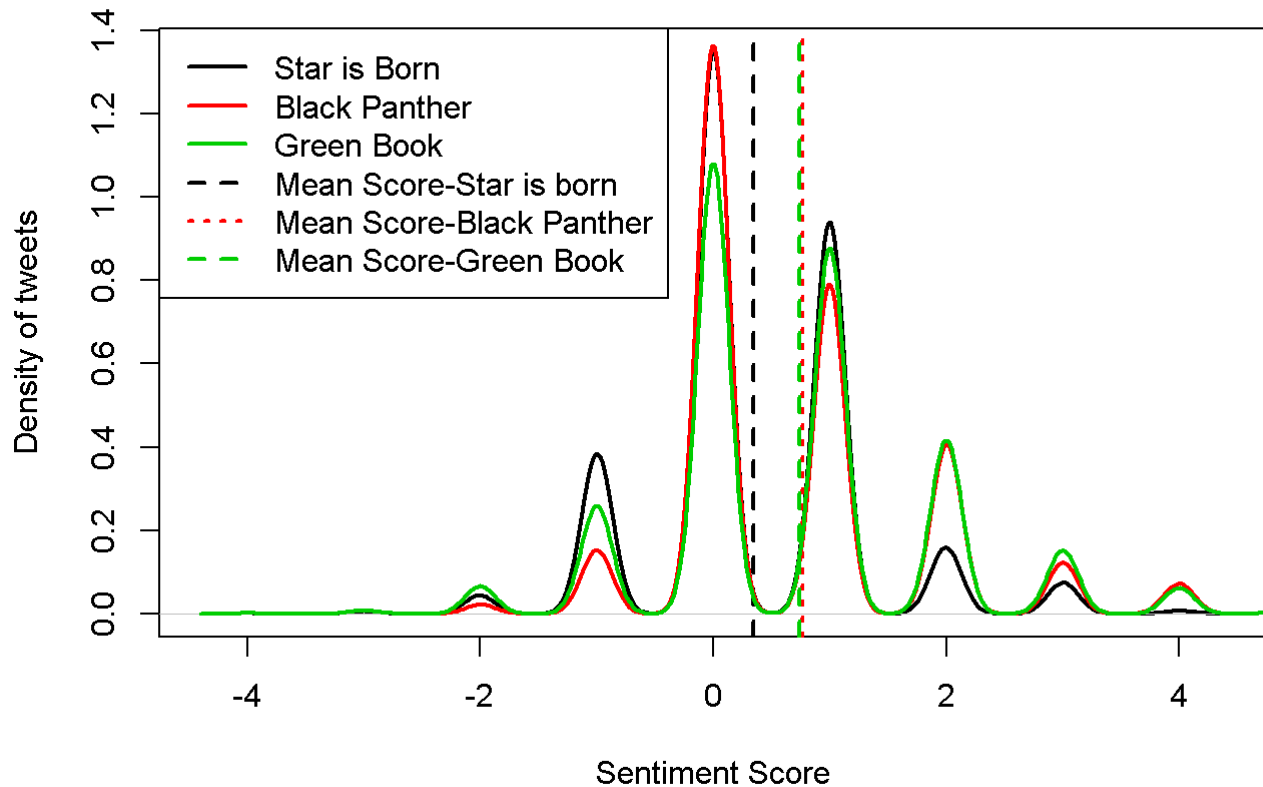
#plot 2
lines(density(BlackPanther_score),col=2, lwd=2)
abline(v=mean(BlackPanther_score),col=2, lwd=2,lty=3)

#plot 3
lines(density(GreenBook_score),col=3,lwd=2)
abline(v=mean(GreenBook_score),col=3, lwd=2,lty=2)

#Legend=d
legend("topleft",col=c(1,2,3,1,2,3),lwd=c(2,2,2,2,2,2),lty=c(1,1,1,2,3,2), c("Star is Born",
"Black Panther","Green Book", "Mean Score-Star is born", "Mean Score-Black Panther","Mean Score
-Green Book"))

```

Sentiment Analysis of 3 Oscar nominated movies



Finding movie with highest number of positive tweets:

```
sum(StarBorn_score>0)/length(StarBorn_score)
```

```
## [1] 0.398125
```

```
## [1] 0.389375
sum(BlackPanther_score>0)/length(BlackPanther_score)
```

```
## [1] 0.4825
```

```
## [1] 0.514375
sum(GreenBook_score>0)/length(GreenBook_score)
```

```
## [1] 0.5221875
```

```
## [1] 0.5234375
# The winner is Green Book
```

Highest proportion of positive tweets are for Green Book. This is because as can be seen from the figure, when we add all the density of tweets and positive sentiment scores(0+) of each, more proportion of tweets have given higher ratings for Black Panther and Green Book than Star is Born. In other words, even though higher proportion of tweets of Star is Born has a sentiment score of around 1, fewer proportion of tweets have given Star is born higher sentiment scores above 2. Black panther and Green book scores are similar but green book has slightly higher proportion.

Generating a word cloud using positive tweets for Green Book

```

#adding sentiment score to dataframe of Green book so that only tweets with positive scores can
be #extracted
GreenBookDF2=cbind(GreenBookDF,GreenBook_score)

#extracting only positive tweets about Green Book
PosTweets=GreenBookDF2[GreenBookDF2[, "GreenBook_score"]>0, "text"]

#cleaning tweets of Green Book

#removes retweet entities
sentence=gsub("(RT|via)((?:\\b\\W*@[\\w+)+)", " ", PosTweets )

#removes all @someone
sentence=gsub("@\\w+", " ", sentence)

# remove all the punctuation except apostrophe
sentence=gsub("[[:punct:]]", "", sentence)

# remove all the control chracters, like \\n or \\r
sentence=gsub("[[:cntrl:]]", "", sentence)

# remove numbers, we need only text for analytics
sentence=gsub("[[:digit:]]", "", sentence)

#remove html links
sentence=gsub("http\\w+", "", sentence)

# remove unnecessary spaces (white spaces, tabs etc)
sentence=gsub("[ \\t]{2,}", " ", sentence)
sentence=gsub("^\\s+|\\s+$", "", sentence)

# convert to lower case
#sentence=iconv(sentence, "ASCII", "UTF-8", sub="")
sentence=tolower(sentence)

# split into words.
word.list=strsplit(sentence, " ")

# convert list to vector
words=unlist(word.list)

#removing stopwords
words=words[!words %in% tm::stopwords(kind="english")]

library(wordcloud)

```

```
## Warning: package 'wordcloud' was built under R version 3.6.2
```

```
## Loading required package: RColorBrewer
```

```
# count the frequency of each word
freq=table(words)

# produce a word cloud

#I'm using minimum freq of 15 so the words are readable.

wordcloud(names(freq),freq,min.freq=15 ,colors=rainbow(12),random.order=FALSE)
title(main="Most frequent Positive words used to tweet about Green Book")
```

Most frequent Positive words used to tweet about Green Book

