



Case Study 4 Web Mining from Twitter Data

1. Twitter Account

This case study requires a twitter account. You can use your own twitter account if you have one. Otherwise, we can use Dr. Donglei Du's public twitter login as a substitute to illustrate Twitter Mining. Dr. Donglei Du is a Professor in Operations Research, from the faculty of Business Administration, University of New Brunswick (bio: <http://www2.unb.ca/~ddu/>). You can either do step 1.1 (own account) or step 1.2 (Dr. Donglei Du).

1.1. Steps to Setup Handshake between Own Twitter Account and R

- 1.1.1. Go to <https://apps.twitter.com/>
- 1.1.2. Sign in with your account.
- 1.1.3. Click on create a new app.
- 1.1.4. Enter the following: Name: "Sentiment Analysis (Your Initials)", Description: "Sentiment Analysis Using R", Website: <http://test.de/>. Agree on the terms and click on Create your Twitter Application.
- 1.1.5. Click on Keys and Access Tokens [Keys and Access Tokens](#) .
- 1.1.6. Take Note of the Keys:
 - Consumer Key (API Key)
 - Consumer Secret (API Secret)
- 1.1.7. Click on "Create My Access Token"
- 1.1.8. Take Note of the Keys:
 - Access Token
 - Access Token Secret

1.2. Steps to Setup Handshake between Dr. Donglei Du's Twitter Account and R

Skip this section if you have done step 1.1.

- 1.2.1. Go to <https://apps.twitter.com/>
- 1.2.2. Sign in with ID: donglei.du@gmail.com, PW: ddl11700
- 1.2.3. Go to <https://apps.twitter.com/>. The interface should look like this:

Twitter Apps



- 1.2.4. Click on the app "donglei du"
- 1.2.5. Click on Keys and Access Tokens [Keys and Access Tokens](#) .
- 1.2.6. Take Note of the Keys:
 - Consumer Key (API Key)
 - Consumer Secret (API Secret)
 - Access Token
 - Access Token Secret



1.3. Steps to Setup Authentication of Twitter and R

1.3.1. Open R Studio

1.3.2. On the file explorer tab click on Files.



1.3.3. Click on Explore

1.3.4. Go to the Desktop Folder -> Module 3 Datasets -> Case 4

1.3.5. Click on More. . Click on Set as Working Directory.

1.3.6. Click on File-> New File -> R Script.

1.3.7. In the new tab script , type the following code:

1.3.8. Type the following lines of code:

```
• download.file(url="http://curl.haxx.se/ca/cacert.pem",  
  destfile="cacert.pem")  
• library("twitterR")  
• library("wordcloud")  
• library("tm")  
• library("plyr")  
• library("stringr")  
• consumer_key = '<paste the consumer key here>'  
• consumer_secret = '<paste the consumer secret key here>'  
• access_token = '<paste access token here>'  
• access_secret = '<paste access secret here>'  
• setup_twitter_oauth(consumer_key, consumer_secret, access_token,  
  , access_secret)
```

1.3.9. Highlight all lines of code and click on Run .

1.3.10. Type Yes if prompted in the Console.

**Take note that due to UPD's proxy internet system, the twitter authentication might fail. You can try this code at home where you have direct connection to the internet. To continue with this case, you can load the pre-downloaded set of tweets from the Case 4 folder (Section 1.4).*

1.4. If Twitter Authentication is Successful

1.4.1. Type the following lines of code:

```
• iphone.tweets = searchTwitter('#iphone6', lang='en', n=1500)  
• galaxys6.tweets = searchTwitter('#SamsungGalaxyS6', lang='en',  
  ,n=1500)
```

1.4.2. Highlight all lines of code and click on Run .

1.5. If Twitter Authentication is Unsuccessful

1.5.1. In the environment tab, click on open

1.5.2. Look for the data in the Case 4 folder named: "TwitterTweetsCase4.Rdata"

1.5.3. Click on Open

1.5.4. Tweets are loaded into the environment.



2. Text Mining on the Tweets


2.1. Text Preprocessing

2.1.1. To transform the tweets into character vectors, type the following lines of code:

```

• iphone.text = laply(iphone.tweets, function(t) t$text())
• galaxys6.tweets.text = laply(galaxys6.tweets, function(t)
  t$text())

```

2.1.2. Highlight all lines of code and click on Run .

2.1.3. Type the following lines of code to remove all non-text characters.

```

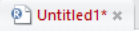
• iphone=str_replace_all(iphone.text,"[^[:graph:]]", " ")
• galaxys6=str_replace_all(galaxys6.tweets.text,"[^[:graph:]]",
  " ")

```

2.2. Load Opinion Lexicon

Hu and Liu's "opinion lexicon" categorizes nearly 6,800 words as positive or negative and can be downloaded from Bing Liu's web site: <http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar>. The lexicon consists of two text files, one containing a list of positive words and the other containing negative words. Each file begins with some documentation, which we need to skip and is denoted by initial semi-colon (";") characters. We also added some words like 'wtf', 'epicfail' etc.


2.2.1. Click on File-> New File -> R Script.

2.2.2. In the new tab script , type the following code:

```

• hu.liu.pos = scan('positive-words.txt', what='character',
  comment.char=';')
• hu.liu.neg = scan('negative-words.txt', what='character',
  comment.char=';')
• pos.words = c(hu.liu.pos, 'upgrade')
• neg.words = c(hu.liu.neg, 'wtf', 'wait',
  'waiting', 'epicfail', 'mechanical')

```

2.2.3. Highlight the four lines and click on Run . As a result, the word sentiment data is loaded in the Environment

2.3. Copying the Sentiment Scorer Function

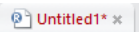
Jeffery Breen provided an R script to score each tweet. The score.sentiment() function uses laply() to iterate through the input text. It strips punctuation and control characters from each line using R's regular expression-powered substitution function, gsub(), and uses match() against each word list to find matches:

2.3.1. Go to the Desktop/Case 4 Folder. And Open the Text File "Sentiment Scorer Function Code.txt" file

2.3.2. Copy all lines of code.

2.3.3. Paste the lines of code after the code from Step 2.2.2 in the R Studio Script tab.

2.4. Scoring the Tweets

2.4.1. In the new tab script , type the following code:

```

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

```



- galaxys6.scores = score.sentiment(galaxys6, pos.words, neg.words, .progress='text')

2.4.1. Highlight the lines of code from step 2.3.3 to the code lines from step 2.4.1 and click on Run



2.5. Creating a Histogram of the Scores

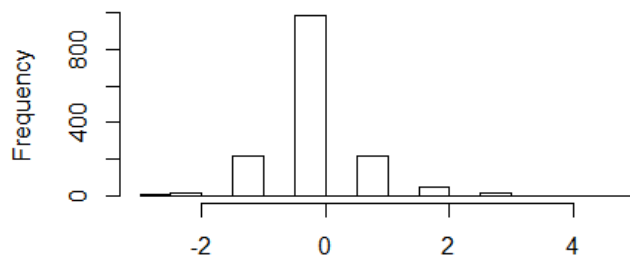
2.5.1. In the new tab script , type the following code:

- par(mfrow = c(1, 2), mar=c(2, 2, 2, 2))
- hist(iphone.scores\$score)
- hist(galaxys6.scores\$score)

2.5.2. Highlight the lines of code and click on Run .

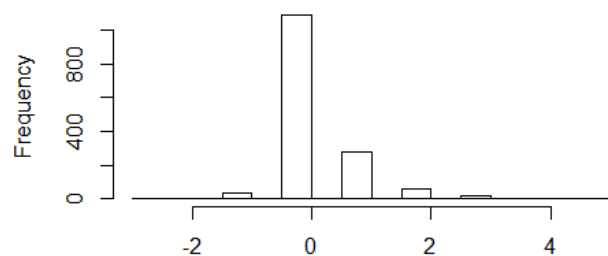
2.5.3. The output of the lines of code should look like this if tweets are loaded from Case 4 folder.

Histogram of iphone.scores\$score



iphone.scores\$score

Histogram of galaxys6.scores\$score



galaxys6.scores\$score

2.5.4. To calculate the average sentiment score of each phone, type the following lines of code:

- AvgiPhoneScore = mean(iphone.scores\$score)
- Avvgalaxys6score = mean(galaxys6.scores\$score)

2.5.5. Highlight the lines of code and click on Run .

2.5.6. The output of the lines of code should be as follows:

- AvgiPhoneScore=0.074
- Avvgalaxys6score=0.283

2.5.7. What analysis can you gather from the sentiment score and histograms?

2.6. Generating a Word Cloud of Text for both Phones.

2.6.1. In the new tab script , type the following code:

- set.seed(4363)
- par(mfrow = c(1, 2), mar=c(2, 2, 2, 2))
- wordcloud(iphone, max.words=30)
- wordcloud(galaxys6, max.words=30)

2.6.2. Highlight the lines of code and click on Run .

2.6.3. The output of the code should look like this:



iphone6cover
loveit ipad sleeve
organic soccer case
shop apple ios amp
game play new edition
strike the sleeve mole sleeve
appstore s phonexd
plus app
iphone6case
leather

claroquetieneminúmero
galaxys6case deerskin
character pretty pitahaya
s6s6 via smartphone
kiwi shop new salzburg xaver
selfie samsung
time felt the autumn phone
galaxys6
edge

Case Adapted and Edited from Jeffery Breen: <http://www.inside-r.org/howto/mining-twitter-airline-consumer-sentiment>