R implementation

Sleiman Bassim, PhD

October 1, 2014

1 Explore Twitter using R

Load packages for data mining of tweets. *twitteR* for accessing twitter, *tm* for semantic mining, and *wordcloud* to visualize the results.

```
library(ROAuth);
library(dplyr);
setwd("/media/Data/Dropbox/R/twitter")
```

1.1 Update to latest version of twitteR

Install an updated version of *twitteR* from github. Run all this chunk ONCE! then RESTART R session!

```
#install.packages(c("devtools","rjson","bit64","httr"));
#library(devtools);
#install_github("twitteR",username="geoffjentry");
library(twitteR);
```

Installation from CRAN did not result in an updated version of *twitteR*. Github installs a dev version.

1.2 Authentication settings from Twitter

From the application settings twitter I will get the access keys to authenticate with twitter. Follow this tutorial under section 3 for additional settings. Get keys after setting an app in Twitter **hint** for a valid URL use http://test.de/

```
source("keys.R")
setup_twitter_oauth(api_key, api_secret, access_token, access_secret)
```

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

1.3 Testing authentication

Getting some of the trends that are popular. But first convert to dplyr wrapper dataframe.

```
current_trends <- availableTrendLocations()
```

```
head(current_trends)
```

```
       name country woeid
1 Worldwide            1
2  Winnipeg  Canada  2972
3    Ottawa  Canada  3369
4    Quebec  Canada  3444
5  Montreal  Canada  3534
6   Toronto  Canada  4118
```

```
montreal_trends <- getTrends(3534)
mt_df <- tbl_df(montreal_trends)
mt_df
```

```
Source: local data frame [10 x 4]

              name
1           Québec
2         Montréal
3         #Salvail
4       Windows 10
5           Canada
6          Toronto
7            Texas
8      #1DProposal
9  #VerifiedNordan
10       Halloween
Variables not shown: url (chr), query (chr),
  woeid (chr)
```

1.4 Transforming data

Retrieve a user current timeline

```
keywords <- userTimeline("genomicsedu", n=100)
```

The list of tweets is transformed into a data frame. You can use `tbl_df` to wrapp the dataframe. Manipulate the text by removing generic and non relevant strings and clarifying the sentences and to keep important keywords from the tweets. `Do.call` works fine in R but not in knitr, so run the chunk in R first, then in knitr, then load the text mining (tm) package.

```
require(plyr);
ask.api <- function(hash,n){
    a <- searchTwitter(hash, n=n)
    b <- tbl_df(do.call("rbind.fill",lapply(a, as.data.frame)))
}
data.tweets <- ask.api(hash="#genomics",n=1000)
```

```
Warning:  1000 tweets were requested but the API can only return 100
Error:  cannot coerce class "structure("status", package = "twitteR")" to a data.frame
```

```
dim(data.tweets)
```

```
[1] 100  16
```

Build a text mining wrapper function then inspect the extracted dataframe.

```
library(tm);
```

```
rm.generic <- function(x,language,words){
    a <- Corpus(VectorSource(x[,1]))
    a <- tm_map(a, removePunctuation)
    a <- tm_map(a, removeNumbers)
    a <- tm_map(a, tolower)
    b <- c(stopwords(language),words)
#    a <- tm_map(a, removeWords, b)
    a <- tm_map(a,PlainTextDocument)
}
testing <- rm.generic(data.tweets, lang='english', words=c("the","tvtag","but"))
#inspect(testing)
```

1.5 Build a document-term matrix to setup a cloud-word

Additional text mining can be used to find the origin of the used words and meging them together. It is called stemming. Finding the source of every word. First, build the document-term matrix then inspects it and the most popular words to find associated terms and their score to the related hashtag (up).

```
my.dt <- DocumentTermMatrix(testing)
my.dt

<<DocumentTermMatrix (documents: 100, terms: 370)>>
Non-/sparse entries: 1030/35970
Sparsity           : 97%
Maximal term length: 24
Weighting          : term frequency (tf)

findFreqTerms(my.dt, lowfreq = 30)

[1] "salvail"

findAssocs(my.dt, 'genomics', .2)

$genomics
numeric(0)
```

Store words relatively to their frequency scores then finally use `wordcloud` to plot the text mining results.

```
r.matrix <- as.matrix(my.dt)
r.freq <- colSums(r.matrix)
r.freq.df <- data.frame(term=names(r.freq), r.freq=r.freq, stringsAsFactors = FALSE)
r.freq.df <- r.freq.df%>%arrange(desc(r.freq))

library(RColorBrewer);
pal <- brewer.pal(5, "Dark2")
library(wordcloud);
wordcloud(r.freq.df$term, r.freq.df$r.freq, min.freq=20,random.color=FALSE,colors=pal)
```

ericsalvail
salvail
bruno

2 Related material

- Text Mining
- TwitteR client for R tutorial
- Using the R twitteR package
- Create Twitter Sentiment Word Cloud in R
- Text Data Mining with Twitter and R

3 Session Information

The version number of R and packages loaded for generating the vignette were:

```
R version 3.1.1 (2014-07-10)
Platform: x86_64-pc-linux-gnu (64-bit)

locale:
 [1] LC_CTYPE=en_CA.UTF-8
 [2] LC_NUMERIC=C
 [3] LC_TIME=en_CA.UTF-8
 [4] LC_COLLATE=en_CA.UTF-8
 [5] LC_MONETARY=en_CA.UTF-8
 [6] LC_MESSAGES=en_CA.UTF-8
 [7] LC_PAPER=en_CA.UTF-8
 [8] LC_NAME=C
 [9] LC_ADDRESS=C
[10] LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_CA.UTF-8
[12] LC_IDENTIFICATION=C

attached base packages:
[1] stats     graphics  grDevices utils
[5] datasets  methods   base

other attached packages:
 [1] wordcloud_2.5      RColorBrewer_1.0-5
 [3] tm_0.6             NLP_0.1-5
 [5] plyr_1.8.1         twitteR_1.1.8
 [7] dplyr_0.2          ROAuth_0.9.3
 [9] digest_0.6.4       RCurl_1.95-4.3
[11] bitops_1.0-6       knitr_1.6

loaded via a namespace (and not attached):
 [1] assertthat_0.1  bit_1.1-12
 [3] bit64_0.9-4     codetools_0.2-9
 [5] compiler_3.1.1  evaluate_0.5.5
 [7] formatR_1.0     highr_0.3
 [9] httr_0.5        magrittr_1.0.1
[11] parallel_3.1.1  Rcpp_0.11.2
[13] rjson_0.2.14    slam_0.1-32
[15] stringr_0.6.2   tools_3.1.1
```