Twitter Data Analysis with R

Yanchang Zhao
RDataMining.com

Making Data Analysis Easier – Workshop Organised by the Monash Business Analytics Team (WOMBAT 2016), Monash University, Melbourne

19 February 2016
Outline

Introduction

Tweets Analysis
  Extracting Tweets
  Text Cleaning
  Frequent Words and Word Cloud
  Word Associations
  Topic Modelling
  Sentiment Analysis

Followers and Retweeting Analysis
  Follower Analysis
  Retweeting Analysis

R Packages

References and Online Resources
Twitter

- An online social networking service that enables users to send and read short 140-character messages called “tweets” (Wikipedia)
- Over 300 million monthly active users (as of 2015)
- Creating over 500 million tweets per day
RDataMining Twitter Account

- @RDataMining: focuses on R and Data Mining
- 580+ tweets/retweets (as of February 2016)
- 2,300+ followers
Techniques and Tools

- Techniques
  - Text mining
  - Topic modelling
  - Sentiment analysis
  - Social network analysis

- Tools
  - Twitter API
  - R and its packages:
    - twitteR
    - tm
    - topicmodels
    - sentiment140
    - igraph
Process

- Extract tweets and followers from the Twitter website with R and the `twitteR` package
- With the `tm` package, clean text by removing punctuations, numbers, hyperlinks and stop words, followed by stemming and stem completion
- Build a term-document matrix
- Analyse topics with the `topicmodels` package
- Analyse sentiment with the `sentiment140` package
- Analyse following/followed and retweeting relationships with the `igraph` package
Outline

Introduction

Tweets Analysis
  Extracting Tweets
  Text Cleaning
  Frequent Words and Word Cloud
  Word Associations
  Topic Modelling
  Sentiment Analysis

Followers and Retweeting Analysis
  Follower Analysis
  Retweeting Analysis

R Packages

References and Online Resources
Retrieve Tweets

## Option 1: retrieve tweets from Twitter

```r
library(twitteR)
library(ROAuth)

## Twitter authentication
setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)

## 3200 is the maximum to retrieve
tweets <- userTimeline("RDataMining", n = 3200)
```

## Option 2: download @RDataMining tweets from RDataMining.com

```r
url <- "http://www.rdatamining.com/data/RDataMining-Tweets-20160212.rds"
download.file(url, destfile = "/data/RDataMining-Tweets-20160212.rds")

## load tweets into R
tweets <- readRDS("./data/RDataMining-Tweets-20160212.rds")
```

Twitter Authentication with OAuth:
Section 3 of http://geoffjentry.hexdump.org/twitteR.pdf
(n.tweet <- length(tweets))

## [1] 448

# convert tweets to a data frame
tweets.df <- twListToDF(tweets)
# tweet #190
tweets.df[190, c("id", "created", "screenName", "replyToSN",
    "favoriteCount", "retweetCount", "longitude", "latitude", "text")]

## id created screenName re...
## 190 362866933894352898 2013-08-01 09:26:33 RDataMining ...
## favoriteCount retweetCount longitude latitude
## 190 9 9 NA NA

## 190 The R Reference Card for Data Mining now provides lin...

# print tweet #190 and make text fit for slide width
writeLines(strwrap(tweets.df$text[190], 60))

## The R Reference Card for Data Mining now provides links to
## packages on CRAN. Packages for MapReduce and Hadoop added.
## http://t.co/RrFypol8kw
library(tm)
# build a corpus, and specify the source to be character vectors
myCorpus <- Corpus(VectorSource(tweets.df$text))
# convert to lower case
myCorpus <- tm_map(myCorpus, content_transformer(tolower))
# remove URLs
removeURL <- function(x) gsub("http[^[:space:]]*", "", x)
myCorpus <- tm_map(myCorpus, content_transformer(removeURL))
# remove anything other than English letters or space
removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]*", "", x)
myCorpus <- tm_map(myCorpus, content_transformer(removeNumPunct))
# remove stopwords
myStopwords <- c(setdiff(stopwords('english'), c("r", "big")),
                 "use", "see", "used", "via", "amp")
myCorpus <- tm_map(myCorpus, removeWords, myStopwords)
# remove extra whitespace
myCorpus <- tm_map(myCorpus, stripWhitespace)

# keep a copy for stem completion later
myCorpusCopy <- myCorpus
Stemming and Stem Completion

```r
myCorpus <- tm_map(myCorpus, stemDocument)  # stem words
writeLines(strwrap(myCorpus[[190]]$content, 60))

## r refer card data mine now provid link packag cran packag
## mapreduc hadoop ad

stemCompletion2 <- function(x, dictionary) {
  x <- unlist(strsplit(as.character(x), " "))
  x <- x[x != ""]
  x <- stemCompletion(x, dictionary=dictionary)
  x <- paste(x, sep="", collapse=" ")
 PlainTextDocument(stripWhitespace(x))
}
myCorpus <- lapply(myCorpus, stemCompletion2, dictionary=myCorpusCopy)
myCorpus <- Corpus(VectorSource(myCorpus))
writeLines(strwrap(myCorpus[[190]]$content, 60))

## r reference card data miner now provided link package cran
## package mapreduce hadoop ad
```

Issues in Stem Completion: “Miner” vs “Mining”

```r
# count word frequency
wordFreq <- function(corpus, word) {
  results <- lapply(corpus,
    function(x) { grep(as.character(x), pattern=paste0("\\<",word)) }
  )
  sum(unlist(results))
}

n.miner <- wordFreq(myCorpusCopy, "miner")
n.mining <- wordFreq(myCorpusCopy, "mining")
cat(n.miner, n.mining)

## 9 104

# replace oldword with newword
replaceWord <- function(corpus, oldword, newword) {
  tm_map(corpus, content_transformer(gsub),
    pattern=oldword, replacement=newword)
}

myCorpus <- replaceWord(myCorpus, "miner", "mining")
myCorpus <- replaceWord(myCorpus, "universidad", "university")
myCorpus <- replaceWord(myCorpus, "scienc", "science")
```
Build Term Document Matrix

tdm <- TermDocumentMatrix(myCorpus,
                           control = list(wordLengths = c(1, Inf)))
tdm

## <<TermDocumentMatrix (terms: 1073, documents: 448)>>
## Non-/sparse entries: 3594/477110
## Sparsity : 99%
## Maximal term length: 23
## Weighting : term frequency (tf)

idx <- which(dimnames(tdm)$Terms %in% c("r", "data", "mining"))
as.matrix(tdm[idx, 21:30])

## Docs
## Terms 21 22 23 24 25 26 27 28 29 30
## data 0 1 0 0 1 0 0 0 0 1
## mining 0 0 0 0 1 0 0 0 0 1
## r 1 1 1 1 0 1 0 1 1 1
Top Frequent Terms

```r
# inspect frequent words
(freq.terms <- findFreqTerms(tdm, lowfreq = 20))

## [1] "analysing"   "analytics"   "australia"   "big"
## [5] "canberra"    "course"     "data"       "example"
## [9] "group"       "introduction" "learn"      "mining"
## [13] "network"     "package"     "position"   "r"
## [17] "rdatamining" "research"    "science"    "slide"
## [21] "talk"        "text"       "tutorial"   "university"

term.freq <- rowSums(as.matrix(tdm))
term.freq <- subset(term.freq, term.freq >= 20)
df <- data.frame(term = names(term.freq), freq = term.freq)
```
library(ggplot2)
ggplot(df, aes(x=term, y=freq)) + geom_bar(stat="identity") +
xlab("Terms") + ylab("Count") + coord_flip() +
theme(axis.text=element_text(size=7))
```r
m <- as.matrix(tdm)
# calculate the frequency of words and sort it by frequency
word.freq <- sort(rowSums(m), decreasing = T)
# colors
pal <- brewer.pal(9, "BuGn")[-(1:4)]

# plot word cloud
library(wordcloud)
wordcloud(words = names(word.freq), freq = word.freq, min.freq = 3, 
          random.order = F, colors = pal)
```
Associations

# which words are associated with 'r'?
findAssocs(tdm, "r", 0.2)

##
## r
## code 0.27
## example 0.21
## series 0.21
## markdown 0.20
## user 0.20

# which words are associated with 'data'?
findAssocs(tdm, "data", 0.2)

##
## data
## mining 0.48
## big 0.44
## analytics 0.31
## science 0.29
## poll 0.24
Network of Terms

library(graph)
library(Rgraphviz)
plot(tdm, term = freq.terms, corThreshold = 0.1, weighting = T)
```r
# Topic Modelling

dtm <- as.DocumentTermMatrix(tdm)
library(topicmodels)
lda <- LDA(dtm, k = 8)  # find 8 topics
term <- terms(lda, 7)  # first 7 terms of every topic
(term <- apply(term, MARGIN = 2, paste, collapse = ", "))

## Topic 1
"data, mining, big, r, analysing, scientist, group"

## Topic 2
"r, mining, data, analysing, university, slide, network"

## Topic 3
"r, data, book, package, mining, cluster, tutorial"

## Topic 4
"data, r, big, mining, rstudio, tutorial, slide"

## Topic 5
"data, mining, research, slide, free, course, position"

## Topic 6
"data, group, package, r, computational, canberra, machine"

## Topic 7
"mining, slide, r, analytics, example, talk, will"

## Topic 8
"r, data, mining, example, pdf, series, available"
```
topics <- topics(lda)  # 1st topic identified for every document (tweet)
topics <- data.frame(date=as.IDate(tweets.df$created), topic=topics)
qplot(date, ..count.., data=topics, geom="density",
      fill=term[topic], position="stack")

Another way to plot steam graph:

# install package sentiment140
require(devtools)
install_github("sentiment140", "okugami79")

# sentiment analysis
library(sentiment)
sentiments <- sentiment(tweets.df$text)
table(sentiments$polarity)

##
## neutral positive
## 428 20

# sentiment plot
sentiments$score <- 0
sentiments$score[sentiments$polarity == "positive"] <- 1
sentiments$score[sentiments$polarity == "negative"] <- -1
sentiments$date <- as.IDate(tweets.df$created)
result <- aggregate(score ~ date, data = sentiments, sum)
plot(result, type = "l")
Outline

Introduction

Tweets Analysis
  Extracting Tweets
  Text Cleaning
  Frequent Words and Word Cloud
  Word Associations
  Topic Modelling
  Sentiment Analysis

Followers and Retweeting Analysis
  Follower Analysis
  Retweeting Analysis

R Packages

References and Online Resources
Retrieve User Info and Followers

```r
user <- getUser("RDataMining")
user$toDataFrame()
friends <- user$getFriends()  # who this user follows
followers <- user$getFollowers()  # this user's followers
followers2 <- followers[[1]]$getFollowers()  # a follower's followers

# [,1] ...
# description "R and Data Mining. Group on LinkedIn: ...
# statusesCount "583" ...
# followersCount "2376" ...
# favoritesCount "6" ...
# friendsCount "72" ...
# url "http://t.co/LwL50uRmPd" ...
# name "Yanchang Zhao" ...
# created "2011-04-04 09:15:43" ...
# protected "FALSE" ...
# verified "FALSE" ...
# screenName "RDataMining" ...
# location "Australia" ...
# lang "en" ...
# id "276895537" ...
# listedCount "157" ...
```
Based on Jeff Leek's twitterMap function at http://biostat.jhsph.edu/~jleek/code/twitterMap.R
# select top retweeted tweets

table(tweets.df$retweetCount)
selected <- which(tweets.df$retweetCount >= 9)

# plot them

dates <- strptime(tweets.df$created, format="%Y-%m-%d")
plot(x=dates, y=tweets.df$retweetCount, type="l", col="grey",
     xlab="Date", ylab="Times retweeted")

colors <- rainbow(10)[1:length(selected)]
points(dates[selected], tweets.df$retweetCount[selected],
       pch=19, col=colors)
text(dates[selected], tweets.df$retweetCount[selected],
       tweets.df$text[selected], col=colors, cex=.9)
Handling and Processing Strings in R — an ebook in PDF format, 105 pages. http://t.co/UXnetU7k87

Lecture videos of natural language processing course at Stanford University: 18 videos, with each of over 1 hr length http://t.co/VKKdA9T ykm

The R Reference Card for Data Mining now provides links to packages on CRAN. Packages for MapReduce and Hadoop added. http://t.co/RrFypol8kw

Slides in 8 PDF files on Getting Data from the Web with R http://t.co/epT4Jv07WD

Handling and Processing Strings in R — an ebook in PDF format, 105 pages. http://t.co/UXnetU7k87

A Twitter dataset for text mining: @RDataMining Tweets extracted on 3 February 2016. Download it at https://t.co/lQp94lvfPf

Free online course on Computing for Data Analysis (with R), to start on 24 Sept 2012  https://t.co/Y617n30y

The R Reference Card for Data Mining now provides links to packages on CRAN. Packages for MapReduce and Hadoop added. http://t.co/RrFypol8kw

Handling and Processing Strings in R — an ebook in PDF format, 105 pages. http://t.co/UXnetU7k87

Lecture videos of natural language processing course at Stanford University: 18 videos, with each of over 1 hr length http://t.co/VKKdA9T ykm

The R Reference Card for Data Mining now provides links to packages on CRAN. Packages for MapReduce and Hadoop added. http://t.co/RrFypol8kw
The tweet potentially reached around 120,000 users.
Outline

Introduction

Tweets Analysis
   Extracting Tweets
   Text Cleaning
   Frequent Words and Word Cloud
   Word Associations
   Topic Modelling
   Sentiment Analysis

Followers and Retweeting Analysis
   Follower Analysis
   Retweeting Analysis

R Packages

References and Online Resources
R Packages

- Twitter data extraction: `twitteR`
- Text cleaning and mining: `tm`
- Word cloud: `wordcloud`
- Topic modelling: `topicmodels, lda`
- Sentiment analysis: `sentiment140`
- Social network analysis: `igraph, sna`
- Visualisation: `wordcloud, Rgraphviz, ggplot2`
Twitter Data Extraction – Package *twitteR* \(^3\)

- `userTimeline`, `homeTimeline`, `mentions`, `retweetsOfMe`: retrieve various timelines
- `getUser`, `lookupUsers`: get information of Twitter user(s)
- `getFollowers`, `getFollowerIDs`: retrieve followers (or their IDs)
- `getFriends`, `getFriendIDs`: return a list of Twitter users (or user IDs) that a user follows
- `retweets`, `retweeters`: return retweets or users who retweeted a tweet
- `searchTwitter`: issue a search of Twitter
- `getCurRateLimitInfo`: retrieve current rate limit information
- `twListToDF`: convert into data.frame

\(^3\)https://cran.r-project.org/package=twitteR
Text Mining – Package \textit{tm} \(^4\)

- \texttt{removeNumbers, removePunctuation, removeWords, removeSparseTerms, stripWhitespace}: remove numbers, punctuations, words or extra whitespaces
- \texttt{removeSparseTerms}: remove sparse terms from a term-document matrix
- \texttt{stopwords}: various kinds of stopwords
- \texttt{stemDocument, stemCompletion}: stem words and complete stems
- \texttt{TermDocumentMatrix, DocumentTermMatrix}: build a term-document matrix or a document-term matrix
- \texttt{termFreq}: generate a term frequency vector
- \texttt{findFreqTerms, findAssocs}: find frequent terms or associations of terms
- \texttt{weightBin, weightTf, weightTfIdf, weightSMART, WeightFunction}: various ways to weight a term-document matrix

\(^4\)https://cran.r-project.org/package=tm
Topic Modelling and Sentiment Analysis – Packages

*topicmodels & sentiment140*

Package *topicmodels* 5

- **LDA**: build a Latent Dirichlet Allocation (LDA) model
- **CTM**: build a Correlated Topic Model (CTM) model
- **terms**: extract the most likely terms for each topic
- **topics**: extract the most likely topics for each document

Package *sentiment140* 6

- **sentiment**: sentiment analysis with the sentiment140 API, tune to Twitter text analysis

---

5 https://cran.r-project.org/package=topicmodels

6 https://github.com/okugami79/sentiment140
Social Network Analysis and Visualization – Package igraph

- degree, betweenness, closeness, transitivity: various centrality scores
- neighborhood: neighborhood of graph vertices
- cliques, largest.cliques, maximal.cliques, clique.number: find cliques, ie. complete subgraphs
- clusters, no.clusters: maximal connected components of a graph and the number of them
- fastgreedy.community, spinglass.community: community detection
- cohesive.blocks: calculate cohesive blocks
- induced.subgraph: create a subgraph of a graph (igraph)
- read.graph, write.graph: read and write graphs from and to files of various formats

7https://cran.r-project.org/package=igraph
Outline

Introduction

Tweets Analysis
  Extracting Tweets
  Text Cleaning
  Frequent Words and Word Cloud
  Word Associations
  Topic Modelling
  Sentiment Analysis

Followers and Retweeting Analysis
  Follower Analysis
  Retweeting Analysis

R Packages

References and Online Resources


Online Resources

- RDataMining Reference Card

- Online documents, books and tutorials
  http://www.rdatamining.com/resources/onlinedocs

- Free online courses
  http://www.rdatamining.com/resources/courses

- RDataMining Group on LinkedIn (18,000+ members)
  http://group.rdatamining.com

- RDataMining on Twitter (2,300+ followers)
  @RDataMining
The End

Thanks!

Email: yanchang(at)RDataMining.com
Twitter: @RDataMining