

Text Mining with R – an Analysis of Twitter Data¹

Yanchang Zhao

<http://www.RDataMining.com>

R and Data Mining Workshop @ AusDM 2014
27 November 2014

¹Presented at Amazon and UJAT

Outline

Introduction

Extracting Tweets

Text Cleaning

Frequent Words and Associations

Word Cloud

Clustering

Topic Modelling

Online Resources

Text Mining

- ▶ unstructured text data
- ▶ text categorization
- ▶ text clustering
- ▶ entity extraction
- ▶ sentiment analysis
- ▶ document summarization
- ▶ ...

Text mining of Twitter data with R ²

1. extract data from Twitter
2. clean extracted data and build a document-term matrix
3. find frequent words and associations
4. create a word cloud to visualize important words
5. text clustering
6. topic modelling

²Chapter 10: Text Mining, *R and Data Mining: Examples and Case Studies*.

Outline

Introduction

Extracting Tweets

Text Cleaning

Frequent Words and Associations

Word Cloud

Clustering

Topic Modelling

Online Resources

Retrieve Tweets

Retrieve recent tweets by @RDataMining

```
## Option 1: retrieve tweets from Twitter  
library(twitteR)  
tweets <- userTimeline("RDataMining", n = 3200)
```

```
## Option 2: download @RDataMining tweets from RDataMining.com  
url <- "http://www.rdatamining.com/data/rdmTweets.RData"  
download.file(url, destfile = "./data/rdmTweets.RData")
```

```
## load tweets into R  
load(file = "./data/rdmTweets.RData")
```

```
(n.tweet <- length(tweets))

## [1] 320

tweets[1:5]

## [[1]]
## [1] "RDataMining: Examples on calling Java code from R \nht...
##
## [[2]]
## [1] "RDataMining: Simulating Map-Reduce in R for Big Data A...
##
## [[3]]
## [1] "RDataMining: Job opportunity: Senior Analyst - Big Dat...
##
## [[4]]
## [1] "RDataMining: CLAVIN: an open source software package f...
##
## [[5]]
## [1] "RDataMining: An online book on Natural Language Proces...
```

Outline

Introduction

Extracting Tweets

Text Cleaning

Frequent Words and Associations

Word Cloud

Clustering

Topic Modelling

Online Resources


```
# convert tweets to a data frame
# tweets.df <- do.call("rbind", lapply(tweets, as.data.frame))
tweets.df <- twListToDF(tweets)
dim(tweets.df)

## [1] 320 14

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, tolower)
```

Package tm v0.5-10 was used in this example. With tm v0.6, “content_transformer” needs to be used to wrap around normal functions.

```
# tm v0.6
myCorpus <- tm_map(myCorpus, content_transformer(tolower))
```

```
# remove punctuation
myCorpus <- tm_map(myCorpus, removePunctuation)
# remove numbers
myCorpus <- tm_map(myCorpus, removeNumbers)
# remove URLs
removeURL <- function(x) gsub("http[[:alnum:]]*", "", x)
myCorpus <- tm_map(myCorpus, removeURL)
# add two extra stop words: 'available' and 'via'
myStopwords <- c(stopwords("english"), "available", "via")
# remove 'r' and 'big' from stopwords
myStopwords <- setdiff(myStopwords, c("r", "big"))
# remove stopwords from corpus
myCorpus <- tm_map(myCorpus, removeWords, myStopwords)
```

```
# keep a copy of corpus to use later as a dictionary for stem completion
myCorpusCopy <- myCorpus
# stem words
myCorpus <- tm_map(myCorpus, stemDocument)
```

```
# inspect the first 5 documents (tweets) inspect(myCorpus[1:5])
# The code below is used for to make text fit for paper width
for (i in 1:5) {
  cat(paste("[", i, "] ", sep = ""))
  writeLines(myCorpus[[i]])
}
```

```
## [[1]] exampl call java code r
##
## [[2]] simul mapreduc r big data analysi use flight data ...
## [[3]] job opportun senior analyst big data wesfarm indust...
## [[4]] clavin open sourc softwar packag document geotag g...
## [[5]] onlin book natur languag process python
```

```
# stem completion  
myCorpus <- tm_map(myCorpus, stemCompletion,  
                    dictionary = myCorpusCopy)
```

```
## [[1]] examples call java code r  
## [[2]] simulating mapreduce r big data analysis used flights...  
## [[3]] job opportunity senior analyst big data wesfarmers in...  
## [[4]] clavin open source software package document geotaggi...  
## [[5]] online book natural language processing python
```

```
# count frequency of "mining"
miningCases <- tm_map(myCorpusCopy, grep, pattern = "\\<mining")
sum(unlist(miningCases))

## [1] 82

# count frequency of "miners"
minerCases <- tm_map(myCorpusCopy, grep, pattern = "\\<miners")
sum(unlist(minerCases))

## [1] 4

# replace "miners" with "mining"
myCorpus <- tm_map(myCorpus, gsub, pattern = "miners",
                    replacement = "mining")
```

```
tdm <- TermDocumentMatrix(myCorpus,  
                           control = list(wordLengths = c(1, Inf)))  
tdm  
  
## A term-document matrix (790 terms, 320 documents)  
##  
## Non-/sparse entries: 2449/250351  
## Sparsity           : 99%  
## Maximal term length: 27  
## Weighting          : term frequency (tf)
```

Outline

Introduction

Extracting Tweets

Text Cleaning

Frequent Words and Associations

Word Cloud

Clustering

Topic Modelling

Online Resources

```

idx <- which(dimnames(tdm)$Terms == "r")
inspect(tdm[idx + (0:5), 101:110])

## A term-document matrix (6 terms, 10 documents)
##
## Non-/sparse entries: 4/56
## Sparsity           : 93%
## Maximal term length: 12
## Weighting           : term frequency (tf)
##
##
##           Docs
## Terms      101 102 103 104 105 106 107 108 109 110
##   r         0   1   1   0   0   0   0   0   1   1
## ramachandran 0   0   0   0   0   0   0   0   0   0
## random       0   0   0   0   0   0   0   0   0   0
## ranked      0   0   0   0   0   0   0   0   0   0
## rann         0   0   0   0   0   0   0   0   0   0
## rapidminer   0   0   0   0   0   0   0   0   0   0

```



```
# inspect frequent words
```

```
(freq.terms <- findFreqTerms(tdm, lowfreq = 15))
```

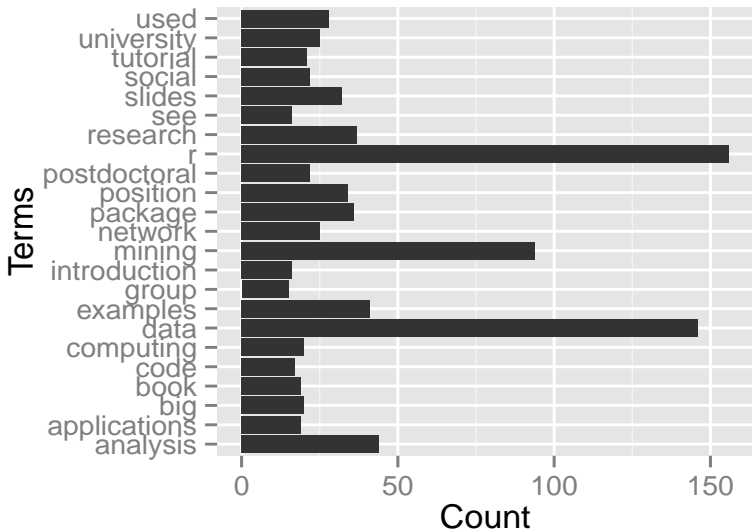
```
## [1] "analysis"      "applications"  "big"           "book"
## [5] "code"          "computing"    "data"          "examples"
## [9] "group"         "introduction" "mining"        "network"
## [13] "package"       "position"     "postdoctoral" "r"
## [17] "research"      "see"          "slides"        "social"
## [21] "tutorial"      "university"   "used"
```

```
term.freq <- rowSums(as.matrix(tdm))
```

```
term.freq <- subset(term.freq, term.freq >= 15)
```

```
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()
```



which words are associated with 'r'?

```
findAssocs(tdm, "r", 0.2)
```

```
##           r
```

```
## examples 0.32
```

```
## code      0.29
```

```
## package   0.20
```

which words are associated with 'mining'?

```
findAssocs(tdm, "mining", 0.25)
```

```
##           mining
```

```
## data          0.47
```

```
## mahout         0.30
```

```
## recommendation 0.30
```

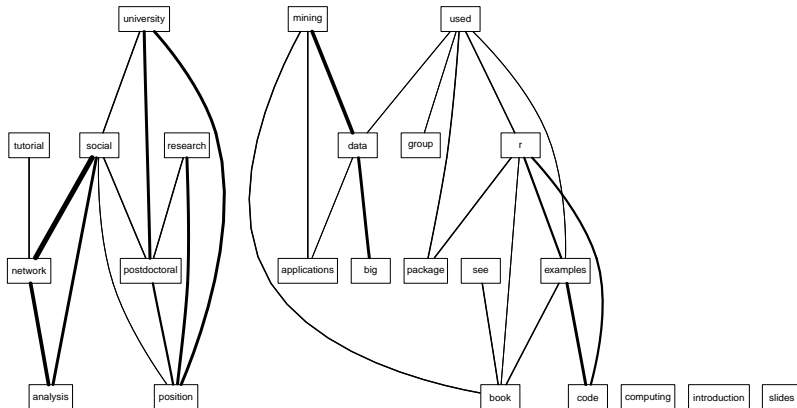
```
## sets           0.30
```

```
## supports       0.30
```

```
## frequent       0.26
```

```
## itemset        0.26
```

```
library(graph)
library(Rgraphviz)
plot(tdm, term = freq.terms, corThreshold = 0.12, weighting = T)
```



Outline

Introduction

Extracting Tweets

Text Cleaning

Frequent Words and Associations

Word Cloud

Clustering

Topic Modelling

Online Resources

Outline

Introduction

Extracting Tweets

Text Cleaning

Frequent Words and Associations

Word Cloud

Clustering

Topic Modelling

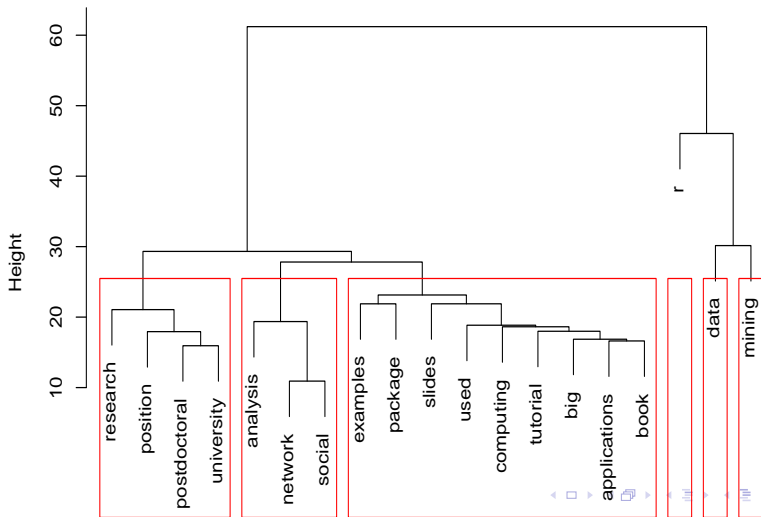
Online Resources

```
# remove sparse terms
tdm2 <- removeSparseTerms(tdm, sparse = 0.95)
m2 <- as.matrix(tdm2)
# cluster terms
distMatrix <- dist(scale(m2))
fit <- hclust(distMatrix, method = "ward")
```



```
plot(fit)
rect.hclust(fit, k = 6) # cut tree into 6 clusters
```

Cluster Dendrogram



```

m3 <- t(m2) # transpose the matrix to cluster documents (tweets)
set.seed(122) # set a fixed random seed
k <- 6 # number of clusters
kmeansResult <- kmeans(m3, k)
round(kmeansResult$centers, digits = 3) # cluster centers

```

```

##      analysis applications      big  book computing  data examples
## 1      0.147              0.088 0.147 0.015          0.059 1.015      0.088
## 2      0.028              0.167 0.167 0.250          0.028 1.556      0.194
## 3      0.810              0.000 0.000 0.000          0.000 0.048      0.095
## 4      0.080              0.036 0.007 0.058          0.087 0.000      0.181
## 5      0.000              0.000 0.000 0.067          0.067 0.333      0.067
## 6      0.119              0.048 0.071 0.000          0.048 0.357      0.000
##      mining network package position postdoctoral      r research
## 1  0.338    0.015    0.015    0.059          0.074 0.235      0.074
## 2  1.056    0.000    0.222    0.000          0.000 1.000      0.028
## 3  0.048    1.000    0.095    0.143          0.095 0.286      0.048
## 4  0.065    0.022    0.174    0.000          0.007 0.703      0.000
## 5  1.200    0.000    0.000    0.000          0.067 0.067      0.000
## 6  0.119    0.000    0.024    0.643          0.310 0.000      0.714
##      slides social tutorial university  used
## 1  0.074    0.000    0.015          0.015 0.029
## 2  0.056    0.000    0.000          0.000 0.250
## 3  0.095    0.762    0.190          0.000 0.095

```

```

for (i in 1:k) {
  cat(paste("cluster ", i, ": ", sep = ""))
  s <- sort(kmeansResult$centers[i, ], decreasing = T)
  cat(names(s)[1:5], "\n")
  # print the tweets of every cluster
  # print(tweets[which(kmeansResult$cluster==i)])
}

## cluster 1:  data mining r analysis big
## cluster 2:  data mining r book used
## cluster 3:  network analysis social r tutorial
## cluster 4:  r examples package slides used
## cluster 5:  mining tutorial slides data book
## cluster 6:  research position university data postdoctoral

```

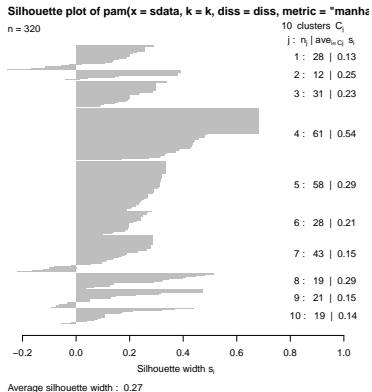
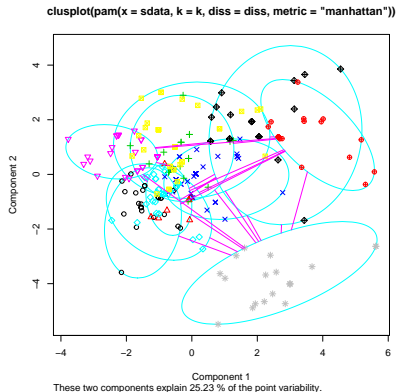
```

library(fpc)
# partitioning around medoids with estimation of number of clusters
pamResult <- pamk(m3, metric="manhattan")
k <- pamResult$nc # number of clusters identified
pamResult <- pamResult$pamobject
# print cluster medoids
for (i in 1:k) {
  cat("cluster", i, ":",
      colnames(pamResult$medoids)[which(pamResult$medoids[i,]==1)], "\n")
}

## cluster 1 :   examples r
## cluster 2 :   analysis data r
## cluster 3 :   data
## cluster 4 :
## cluster 5 :   r
## cluster 6 :   data mining r
## cluster 7 :   data mining
## cluster 8 :   analysis network social
## cluster 9 :   data position research
## cluster 10 :   position postdoctoral university

```

```
# plot clustering result
layout(matrix(c(1, 2), 1, 2)) # set to two graphs per page
plot(pamResult, col.p = pamResult$clustering)
```



```
layout(matrix(1)) # change back to one graph per page
```

Outline

Introduction

Extracting Tweets

Text Cleaning

Frequent Words and Associations

Word Cloud

Clustering

Topic Modelling

Online Resources

Topic Modelling

```
dtm <- as.DocumentTermMatrix(tdm)
library(topicmodels)
lda <- LDA(dtm, k = 8) # find 8 topics
term <- terms(lda, 4) # first 4 terms of every topic
term

##      Topic 1  Topic 2    Topic 3    Topic 4    Topic 5  ...
## [1,] "data"    "r"        "data"      "data"      "data"      ...
## [2,] "group"   "examples" "job"       "r"         "mining"     ...
## [3,] "mining"  "code"     "lecture"   "used"      "r"         ...
## [4,] "ausdm"   "mining"   "australia" "clustering" "applicat...

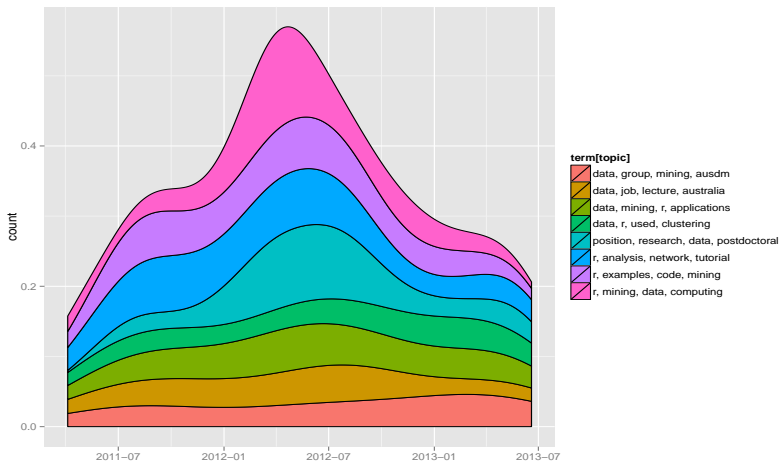
##      Topic 6    Topic 7    Topic 8
## [1,] "position"  "r"        "r"
## [2,] "research"  "mining"   "analysis"
## [3,] "data"      "data"     "network"
## [4,] "postdoctoral" "computing" "tutorial"
```



```
term <- apply(term, MARGIN = 2, paste, collapse = ", ")
```

Topic Modelling

```
# first topic identified for every document (tweet)
topic <- topics(lda, 1)
topics <- data.frame(date=as.IDate(tweets.df$created), topic)
qplot(date, ..count.., data=topics, geom="density",
      fill=term[topic], position="stack")
```



Outline

Introduction

Extracting Tweets

Text Cleaning

Frequent Words and Associations

Word Cloud

Clustering

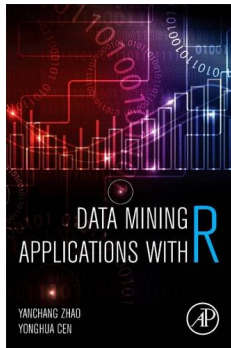
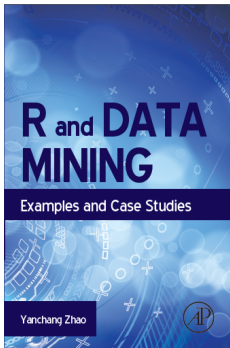
Topic Modelling

Online Resources

Online Resources

- ▶ Chapter 10: Text Mining, in book
R and Data Mining: Examples and Case Studies
<http://www.rdatamining.com/docs/RDataMining.pdf>
- ▶ R Reference Card for Data Mining
<http://www.rdatamining.com/docs/R-refcard-data-mining.pdf>
- ▶ Free online courses and documents
<http://www.rdatamining.com/resources/>
- ▶ RDataMining Group on LinkedIn (8,000+ members)
<http://group.rdatamining.com>
- ▶ RDataMining on Twitter (1,800+ followers)
[@RDataMining](#)

The End



Thanks!

Email: [yanchang\(at\)rdatamining.com](mailto:yanchang(at)rdatamining.com)