Data Science with R
Text Mining

Graham Williams and Anthony Nolan

30th January 2014

Visit http://onepager.togaware.com/ for more OnePageR's.

Text Mining or Text Analytics applies analytic tools to learn from collections of text documents like books, newspapers, emails, etc. The goal is similar to humans learning by reading books. Using automated algorithms we can learn from massive amounts of text, much more than a human can. The material could be consist of millions of newspaper articles to perhaps summarise the main themes and to identify those that are of most interest to particular people.

The required packages for this module include:

```
library(tm)             # Framework for text mining.
library(SnowballC)      # Provides wordStem() for stemming.
```

As we work through this module, new R commands will be introduced. Be sure to review the command's documentation and understand what the command does. You can ask for help using the `?` command as in:

```
?read.csv
```

We can obtain documentation on a particular package using the *help=* option of `library()`:

```
library(help=rattle)
```

This present module is intended to be hands on. To learn effectively, you are encouraged to have R running (e.g., RStudio) and to run all the commands as they appear here. Check that you get the same output, and you understand the output. Try some variations. Explore.

1 Loading a Corpus

A corpus is a collection of texts, usually stored electronically, and from which we perform our analysis. A corpus might be a collection of news articles from Reuters or the published works of Shakespeare. Within each corpus we will have separate articles, stories, volumes, each treated as a separate entity or record.

Documents which we wish to analyse come in many different formats. Quite a few formats are supported by `tm` (Feinerer and Hornik, 2014), the package we will illustrate text mining with in this module. The supported formats include text, PDF, Microsoft Word, and XML.

A number of open source tools are also available to convert most document formats to text files. For our corpus used initially in this module, a collection of PDF documents were converted to text using `pdftotext`.

```
$ for f in *.pdf; do pdftotext -nopgbrk $f; done
```

2 Loading a Corpus: Sources and Readers

There are a variety of sources supported by `tm`. We can use `getSources()` to list those that are supported.

```
getSources()

## [1] "DataframeSource" "DirSource"       "ReutersSource"   "URISource"
## [5] "VectorSource"
```

Exercise: Generate a table in R that extracts the Description from the help page for each of the listed data sources.

In addition to different kinds of sources of documents, our documents for text analysis will come in many different formats. A variety are supported by `tm`:

```
getReaders()

## [1] "readDOC"                 "readPDF"
## [3] "readReut21578XML"        "readReut21578XMLasPlain"
## [5] "readPlain"               "readRCV1"
## [7] "readRCV1asPlain"         "readTabular"
## [9] "readXML"
```

Exercise: Generate a table in R that extracts the Description from the help page for each of the listed data readers.

3 Loading a Corpus: Text Documents

We load a corpus of text documents which is a collection of research papers all stored in the folder we identify below. To work along with us in this module, you can create your own folder called `corpus/txt` and place into that folder a collection of text documents. It does not need to be as many as we use here but a reasonable number makes it more interesting.

```
cname <- file.path(".", "corpus", "txt")
cname

## [1] "./corpus/txt"
```

We can list some of the file names.

```
dir(cname)

##  [1] "acnn96.txt"
##  [2] "adm02.txt"
##  [3] "ai02.txt"
##  [4] "ai03.txt"
....
```

There are 46 documents in this particular corpus.

After loading the `tm` (Feinerer and Hornik, 2014) package into the R library we are ready to load the files from the directory as the source of the files making up the corpus, using `DirSource()`. The source object is passed on to `Corpus()` which loads the documents. We save the resulting collection of documents in memory, stored in a vairalbe called `docs`.

```
library(tm)
docs <- Corpus(DirSource(cname))
docs

## A corpus with 46 text documents

class(docs)

## [1] "VCorpus" "Corpus"  "list"

class(docs[[1]])

## [1] "PlainTextDocument" "TextDocument"      "character"
```

4 Loading a Corpus: PDF Documents

Exercise: Repeat the process but load PDF documents directly rather than our collection of converted text documents.

5 Exploring the Corpus

The `summary()` function provides quite basic information about the corpus.

```
summary(docs)

## A corpus with 46 text documents
##
## The metadata consists of 2 tag-value pairs and a data frame
## Available tags are:
##   create_date creator
## Available variables in the data frame are:
##   MetaID
```

The metadata is just data about the data. This is the description of the types of variables, their functions, permissible values, and so on. Some formats including html and xml contain tags and other data structures that provide more metadata. For our simple PDF text here there is little descriptive information about the data available.

We can (and should) inspect the documents using `inspect()`. This will assure us that data has been loaded properly and as we expect.

```
inspect(docs[16])

## A corpus with 1 text document
##
## The metadata consists of 2 tag-value pairs and a data frame
## Available tags are:
##   create_date creator
## Available variables in the data frame are:
##   MetaID
##
## $hwrf12.txt
## Hybrid weighted random forests for
## classifying very high-dimensional data
## Baoxun Xu1 , Joshua Zhexue Huang2 , Graham Williams2 and
## Yunming Ye1
## 1
##
## Department of Computer Science, Harbin Institute of Technology Shenzhen Gr...
## School, Shenzhen 518055, China
## 2
## Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, S...
## 518055, China
## Email: amusing002@gmail.com
## Random forests are a popular classication method based on an ensemble of a
## single type of decision trees from subspaces of data. In the literature, t...
## are many dierent types of decision tree algorithms, including C4.5, CART,...
## CHAID. Each type of decision tree algorithm may capture dierent information
## and structure. This paper proposes a hybrid weighted random forest algorithm,
....
```

6 Preparing the Corpus

We generally need to perform some pre-processing of the text data to prepare for the text analysis. Example transformations include converting the text to lower case, removing numbers and punctuation, removing stop words, stemming and identifying synonyms. The basic transforms are all available within `tm`.

```
getTransformations()

## [1] "as.PlainTextDocument" "removeNumbers"        "removePunctuation"
## [4] "removeWords"          "stemDocument"         "stripWhitespace"
```

Exercise: Generate a table in R that extracts the Description from the help page for each of the listed transforms.

The function `tm_map()` is used to apply the transformations. We will apply the transformations sequentially to remove unwanted characters from the test. The following pages illustrate these transformations.

7 Preparing the Corpus: Simple Transforms

We start with some manual special transforms we may want to do. For example, we might want to replace "/", used sometimes to separate alternative words, with a space. This will avoid the two words being run into one string of characters through the transformations. We might also replace "@" with a space, for the same reason.

```
for (j in seq(docs))
{
  docs[[j]] <- gsub("/", " ", docs[[j]])
  docs[[j]] <- gsub("@", " ", docs[[j]])
}
```

Check the email address in the following.

```
inspect(docs[16])

## A corpus with 1 text document
##
## The metadata consists of 2 tag-value pairs and a data frame
## Available tags are:
##   create_date creator
## Available variables in the data frame are:
##   MetaID
##
## $hwrf12.txt
## Hybrid weighted random forests for
## classifying very high-dimensional data
## Baoxun Xu1 , Joshua Zhexue Huang2 , Graham Williams2 and
## Yunming Ye1
## 1
##
## Department of Computer Science, Harbin Institute of Technology Shenzhen Gr...
## School, Shenzhen 518055, China
## 2
## Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, S...
## 518055, China
## Email: amusing002 gmail.com
## Random forests are a popular classication method based on an ensemble of a
## single type of decision trees from subspaces of data. In the literature, t...
## are many dierent types of decision tree algorithms, including C4.5, CART,...
## CHAID. Each type of decision tree algorithm may capture dierent information
## and structure. This paper proposes a hybrid weighted random forest algorithm,
....
```

8 Preparing the Corpus: Conversion to Lower Case

```
docs <- tm_map(docs, tolower)
```

```
inspect(docs[16])

## A corpus with 1 text document
##
## The metadata consists of 2 tag-value pairs and a data frame
## Available tags are:
##   create_date creator
## Available variables in the data frame are:
##   MetaID
##
## $hwrf12.txt
## hybrid weighted random forests for
## classifying very high-dimensional data
## baoxun xu1 , joshua zhexue huang2 , graham williams2 and
## yunming ye1
## 1
##
## department of computer science, harbin institute of technology shenzhen gr...
## school, shenzhen 518055, china
## 2
## shenzhen institutes of advanced technology, chinese academy of sciences, s...
## 518055, china
## email: amusing002 gmail.com
## random forests are a popular classication method based on an ensemble of a
## single type of decision trees from subspaces of data. in the literature, t...
## are many dierent types of decision tree algorithms, including c4.5, cart,...
## chaid. each type of decision tree algorithm may capture dierent information
## and structure. this paper proposes a hybrid weighted random forest algorithm,
....
```

We often want to convert to lower case to not distinguish between words simply on case.

9 Preparing the Corpus: Remove Numbers

```
docs <- tm_map(docs, removeNumbers)
```

```
inspect(docs[16])

## A corpus with 1 text document
##
## The metadata consists of 2 tag-value pairs and a data frame
## Available tags are:
##   create_date creator
## Available variables in the data frame are:
##   MetaID
##
## $hwrf12.txt
## hybrid weighted random forests for
## classifying very high-dimensional data
## baoxun xu , joshua zhexue huang , graham williams and
## yunming ye
##
##
## department of computer science, harbin institute of technology shenzhen gr...
## school, shenzhen , china
##
## shenzhen institutes of advanced technology, chinese academy of sciences, s...
## , china
## email: amusing gmail.com
## random forests are a popular classication method based on an ensemble of a
## single type of decision trees from subspaces of data. in the literature, t...
## are many dierent types of decision tree algorithms, including c., cart, and
## chaid. each type of decision tree algorithm may capture dierent information
## and structure. this paper proposes a hybrid weighted random forest algorithm,
....
```

Numbers may or may not be relevant to our analyses. This transform can remove numbers simply.

10 Preparing the Corpus: Remove Punctuation

```
docs <- tm_map(docs, removePunctuation)
```

```
inspect(docs[16])

## A corpus with 1 text document
##
## The metadata consists of 2 tag-value pairs and a data frame
## Available tags are:
##   create_date creator
## Available variables in the data frame are:
##   MetaID
##
## $hwrf12.txt
## hybrid weighted random forests for
## classifying very highdimensional data
## baoxun xu  joshua zhexue huang  graham williams and
## yunming ye
##
##
## department of computer science harbin institute of technology shenzhen gra...
## school shenzhen  china
##
## shenzhen institutes of advanced technology chinese academy of sciences she...
##  china
## email amusing gmailcom
## random forests are a popular classication method based on an ensemble of a
## single type of decision trees from subspaces of data in the literature there
## are many dierent types of decision tree algorithms including c cart and
## chaid each type of decision tree algorithm may capture dierent information
## and structure this paper proposes a hybrid weighted random forest algorithm
....
```

Punctuation can provide gramatical context which supports understanding. Often for initial analyses we ignore the punctuation. Later we will use punctuation to support the extraction of meaning.

11 Preparing the Corpus: Remove English Stop Words

```
docs <- tm_map(docs, removeWords, stopwords("english"))
```

```
inspect(docs[16])
## A corpus with 1 text document
##
## The metadata consists of 2 tag-value pairs and a data frame
## Available tags are:
##   create_date creator
## Available variables in the data frame are:
##   MetaID
##
## $hwrf12.txt
## hybrid weighted random forests
## classifying  highdimensional data
## baoxun xu  joshua zhexue huang  graham williams
## yunming ye
##
##
## department  computer science harbin institute  technology shenzhen graduate
## school shenzhen  china
##
## shenzhen institutes  advanced technology chinese academy  sciences shenzhen
##  china
## email amusing gmailcom
## random forests   popular classication method based   ensemble
## single type  decision trees  subspaces  data   literature
##  many dierent types  decision tree algorithms including c cart
## chaid  type  decision tree algorithm may capture dierent information
##  structure  paper proposes  hybrid weighted random forest algorithm
....
```

Stop words are common words found in a language. Words like *for*, *very*, *and*, *of*, *are*, etc, are common stop words. Notice they have been removed from the above text.

12 Preparing the Corpus: Remove Own Stop Words

```
docs <- tm_map(docs, removeWords, c("department", "email"))
```

```
inspect(docs[16])

## A corpus with 1 text document
##
## The metadata consists of 2 tag-value pairs and a data frame
## Available tags are:
##   create_date creator
## Available variables in the data frame are:
##   MetaID
##
## $hwrf12.txt
## hybrid weighted random forests
## classifying  highdimensional data
## baoxun xu  joshua zhexue huang  graham williams
## yunming ye
##
##
##   computer science harbin institute  technology shenzhen graduate
## school shenzhen  china
##
## shenzhen institutes  advanced technology chinese academy  sciences shenzhen
##  china
##  amusing gmailcom
## random forests   popular classication method based   ensemble
## single type  decision trees  subspaces  data   literature
##  many dierent types  decision tree algorithms including c cart
## chaid  type  decision tree algorithm may capture dierent information
##  structure  paper proposes  hybrid weighted random forest algorithm
....
```

Previously we used the English stopwords provided by tm. We could instead or in addition remove our own stop words as we have done above. We have chosen here two words, simply for illustration. The choice might depend on the domain of discourse, and might not become apparent until we've done some analysis.

13 Preparing the Corpus: Strip Whitespace

```
docs <- tm_map(docs, stripWhitespace)
```

```
inspect(docs[16])

## A corpus with 1 text document
##
## The metadata consists of 2 tag-value pairs and a data frame
## Available tags are:
##   create_date creator
## Available variables in the data frame are:
##   MetaID
##
## $hwrf12.txt
## hybrid weighted random forests
## classifying highdimensional data
## baoxun xu joshua zhexue huang graham williams
## yunming ye
##
##
##  computer science harbin institute technology shenzhen graduate
## school shenzhen china
##
## shenzhen institutes advanced technology chinese academy sciences shenzhen
##  china
##  amusing gmailcom
## random forests popular classication method based ensemble
## single type decision trees subspaces data literature
##  many dierent types decision tree algorithms including c cart
## chaid type decision tree algorithm may capture dierent information
##  structure paper proposes hybrid weighted random forest algorithm
....
```

14 Preparing the Corpus: Specific Transformations

We might also have some specific transformations we would like to perform. The examples here may or may not be useful, depending on how we want to analyse the documents. This is really for illustration using the part of the document we are looking at here, rather than suggesting this specific transform adds value.

```
for (j in seq(docs))
{

  docs[[j]] <- gsub("harbin institute technology", "HIT", docs[[j]])
  docs[[j]] <- gsub("shenzhen institutes advanced technology", "SIAT", docs[[j]])
  docs[[j]] <- gsub("chinese academy sciences", "CAS", docs[[j]])
}
```

```
inspect(docs[16])

## A corpus with 1 text document
##
## The metadata consists of 2 tag-value pairs and a data frame
## Available tags are:
##   create_date creator
## Available variables in the data frame are:
##   MetaID
##
## $hwrf12.txt
## hybrid weighted random forests
## classifying highdimensional data
## baoxun xu joshua zhexue huang graham williams
## yunming ye
##
##
##  computer science HIT shenzhen graduate
## school shenzhen china
##
## SIAT CAS shenzhen
##  china
##  amusing gmailcom
## random forests popular classication method based ensemble
## single type decision trees subspaces data literature
##  many dierent types decision tree algorithms including c cart
## chaid type decision tree algorithm may capture dierent information
##  structure paper proposes hybrid weighted random forest algorithm
....
```

15 Stemming

```
docs <- tm_map(docs, stemDocument)
```

```
inspect(docs[16])

## A corpus with 1 text document
##
## The metadata consists of 2 tag-value pairs and a data frame
## Available tags are:
##   create_date creator
## Available variables in the data frame are:
##   MetaID
##
## $hwrf12.txt
## hybrid weight random forest
## classifi highdimension data
## baoxun xu joshua zhexu huang graham william
## yunm ye
##
##
##  comput scienc HIT shenzhen graduat
## school shenzhen china
##
## SIAT CAS shenzhen
##  china
##  amus gmailcom
## random forest popular classic method base ensembl
## singl type decis tree subspac data literatur
##  mani dier type decis tree algorithm includ c cart
## chaid type decis tree algorithm may captur dier inform
##  structur paper propos hybrid weight random forest algorithm
....
```

Stemming uses an algorithm that removes common word endings for English words, such as "es", "ed" and "'s". The functionality for stemming is provided by `wordStem()` from SnowballC (Bouchet-Valat, 2013).

16 Creating Document Term Matrix

A document term matrix is simply a matrix with documents as the rows and terms as the columns and a count of the frequency of words as the cells of the matrix. We us `DocumentTermMatrix()` to create the matrix. The transpose is created using `TermDocumentMatrix()`.

```
dtm <- DocumentTermMatrix(docs)
dtm

## A document-term matrix (46 documents, 6662 terms)
##
## Non-/sparse entries: 30194/276258
## Sparsity           : 90%
## Maximal term length: 65
## Weighting          : term frequency (tf)
```

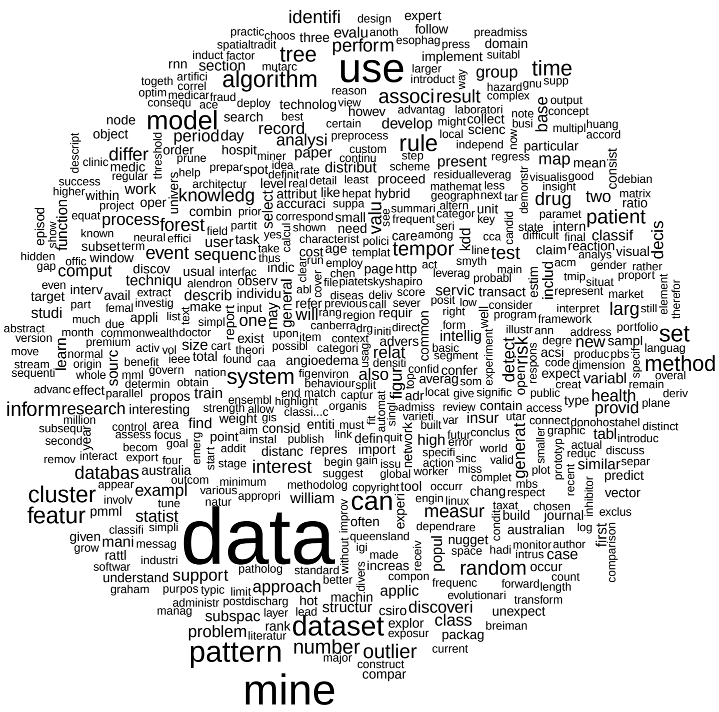
17 Word Clouds

From http://www.rdatamining.com/examples/text-mining

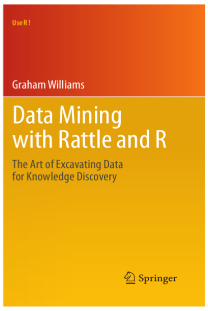
```
library(wordcloud)
m <- as.matrix(dtm)
# calculate the frequency of words
v <- sort(colSums(m), decreasing=TRUE)
head(v, 14)

##      data      mine       use   pattern   dataset       can     model
##      3100      1446      1366       887       776       709       703
##   cluster algorithm      rule    featur       set      tree    method
##       616       611       609       578       555       547       544
....

words <- names(v)
d <- data.frame(word=words, freq=v)
wordcloud(d$word, d$freq, min.freq=40)
```



18 Further Reading



The Rattle Book, published by Springer, provides a comprehensive introduction data mining and analytics using Rattle and R. It is available from Amazon. Other documentation on a broader selection of R topics of relevance to the data scientist is freely available from `http://datamining.togaware.com`, including the Datamining Desktop Survival Guide.

This module is one of many OnePageR modules available from `http://onepager.togaware.com`. In particular follow the links on the website with a * which indicates the generally more developed OnePageR modules.

Other resources include:

- The Journal of Statistical Software article, *Text Mining Infrastructure in R* is a good start `http://www.jstatsoft.org/v25/i05/paper`
- Bilisoly (2008) presents methods and algorithms for text mining using Perl.

19 References

Bilisoly R (2008). *Practical Text Mining with Perl*. Wiley Series on Methods and Applications in Data Mining. Wiley. ISBN 9780470382851. URL http://books.google.com.au/books?id=YkMFVbsrdzkC.

Bouchet-Valat M (2013). *SnowballC: Snowball stemmers based on the C libstemmer UTF-8 library*. R package version 0.5, URL http://CRAN.R-project.org/package=SnowballC.

Feinerer I, Hornik K (2014). *tm: Text Mining Package*. R package version 0.5-10, URL http://CRAN.R-project.org/package=tm.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Williams GJ (2009). "Rattle: A Data Mining GUI for R." *The R Journal*, **1**(2), 45–55. URL http://journal.r-project.org/archive/2009-2/RJournal_2009-2_Williams.pdf.

Williams GJ (2011). *Data Mining with Rattle and R: The art of excavating data for knowledge discovery*. Use R! Springer, New York. URL http://www.amazon.com/gp/product/1441998896/ref=as_li_qf_sp_asin_tl?ie=UTF8&tag=togaware-20&linkCode=as2&camp=217145&creative=399373&creativeASIN=1441998896.

This document, sourced from TextMiningO.Rnw revision 282, was processed by KnitR version 1.5 of 2013-09-28 and took 30.3 seconds to process. It was generated by gjw on nyx running Ubuntu 13.10 with Intel(R) Xeon(R) CPU W3520 @ 2.67GHz having 4 cores and 12.3GB of RAM. It completed the processing 2014-01-30 06:19:37.