Mining Your Qualitative Text

Jeanne Spicer

December 2, 2012

Abstract

This article supplements the RQDA tutorial demonstrating a few additional text-processing features available in R for those conducting qualitative research.

Data

We will assume that you are starting with a data frame containing your plain text with one subject per row. This could be the file you export from RQDA with either the full-text or the codings for each subject. Install and load the `tm` package.

Create Corpus

Create a corpus data object so that you can utilize `tm` package functions and transformations. You can change case, remove punctuation or cluster words into their root stems. Use the `getTransformations()` command to view your options. Here we start with a dataframe extracted from RQDA – "`myData`" containing responses to an open-ended survey question in the variable "`file`".

```
mydata.corpus <- Corpus(VectorSource(myData$file), control = list(minWordLength = 1))

# make each letter lowercase
mydata.corpus <- tm_map(mydata.corpus, tolower)

# remove punctuation
mydata.corpus <- tm_map(mydata.corpus, removePunctuation)
```

Examine Terms

The `tm` package has some functions to help you examine your data. The `TermDocumentMatrix` function prepares a matrix of word counts. You can use pre-defined lists of stopwords or create your own.

```
# build a term-document matrix
mydata.dtm <- TermDocumentMatrix(mydata.corpus, control = list(stopwords = TRUE,
    wordLengths = c(1, 30)))

# inspect the document-term matrix for the occurrance of the word 'tv' in
# the first 10 documents
inspect(mydata.dtm["tv", 1:10, ])
```

```
## A term-document matrix (1 terms, 10 documents)
##
## Non-/sparse entries: 3/7
## Sparsity           : 70%
## Maximal term length: 2
## Weighting          : term frequency (tf)
##
##      Docs
## Terms 1 2 3 4 5 6 7 8 9 10
##    tv 0 0 1 0 0 1 0 1 0  0
```

```
# inspect most popular words
findFreqTerms(mydata.dtm, lowfreq = 30)

##  [1] "basketball" "games"      "hang"       "joke"       "laugh"
##  [6] "movies"     "outside"    "play"       "school"     "stuff"
## [11] "talk"       "tv"         "watch"


# associations of word 'talk' with other terms
findAssocs(mydata.dtm, "talk", 0.2)

##    talk  school    boys      bf    carl karaoke   staff    cant  theres
##    1.00    0.29    0.25    0.24    0.24    0.24    0.24    0.22    0.21


# pull counts for top 30 words
freqwrds <- sort(rowSums(as.matrix(mydata.dtm)), decreasing = TRUE)
freqwrds[1:30]

##       talk       play      watch      games basketball         tv
##        308        276        121         88         85         73
##      stuff     movies       hang      laugh    outside       joke
##         65         50         44         40         40         34
##     school  sometimes      music      video     listen        sit
##         30         29         27         27         26         26
##   football        fun       game      board        eat videogames
##         25         22         20         19         19         19
##      chill     sports      cards       dont        try      girls
##         18         18         17         17         16         15
```

After examining your terms, you may want to combine terms, remove sparse terms or perform additional transformations in preparation for further processing. Here we remove sparsely used terms from the term-document matrix so that we can create some plots.

```
# Note: tweak the sparse parameter to determine the number of words.
# About 10-30 words is good.
mydata.dtm2 <- removeSparseTerms(mydata.dtm, sparse = 0.95)
```

```
# convert the sparse term-document matrix to a standard data frame
mydata.df <- as.data.frame(inspect(mydata.dtm2))
```

```
# inspect dimensions of the data frame
nrow(mydata.df)

## [1] 19

ncol(mydata.df)

## [1] 471
```

Hierarchical Cluster Analysis

You can use the R's cluster analysis functions and packages to identify groups of terms.

```
mydata.df.scale <- scale(mydata.df)
d <- dist(mydata.df.scale, method = "euclidean")  # distance matrix
fit <- hclust(d, method = "ward")
```

```
plot(fit)  # display dendogram
# Show clusters
groups <- cutree(fit, k = 5)  # cut tree into 5 clusters
# draw dendogram with red borders around the 5 clusters
rect.hclust(fit, k = 5, border = "red")
```

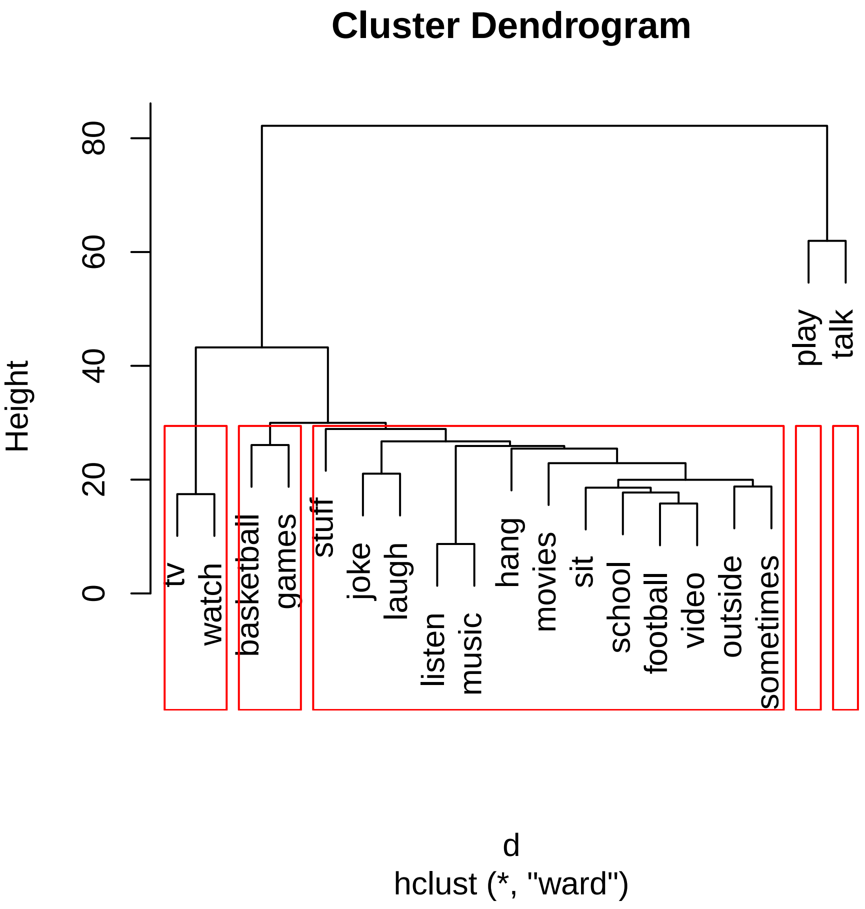

Figure 1: 5-Clusters

Fun with Word Clouds

Word clouds are a popular method of visualizing the most frequently used terms.

```
library(wordcloud)

## Loading required package:  Rcpp
## Loading required package:  RColorBrewer

# calculate the frequency of words using sparse-term reduced matrix
cts <- sort(rowSums(as.matrix(mydata.dtm2)), decreasing = TRUE)
myNames <- names(cts)
wcdata <- data.frame(word = myNames, freq = cts)
```

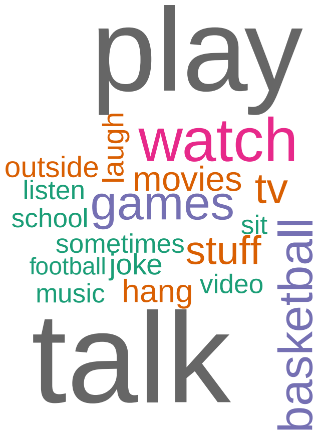

Figure 2: Word Cloud

References

[1] For more information on text-mining with R, see the Natural Language Processing task view `http://cran.r-project.org/web/views/NaturalLanguageProcessing.html`