Getting Sentiment Resources from the Internet

Wouter van Atteveldt

June 3, 2016

This handout describes how to download and parse a sentiment lexicon and collection of reviews from the Internet.

You don't need to run this as the results are saved in the github repository and can be downloaded directly: lexicon; reviews. However, it can be interesting to see how to download files and parse the 'custom' file formats in R.

Pittsburgh Sentiment Lexicon

There are many sentiment dictionaries available for download. For this handout, we use a dictionary developed at the University of Pittsburgh that can be freely downloaded from http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/.

```
url = "http://mpqa.cs.pitt.edu/data/subjectivity_clues_hltemnlp05.zip"
file = "subjectivity_clues_hltemnlp05/subjclueslen1-HLTEMNLP05.tff"
download.file(url, destfile="lexicon.zip")
unzip("lexicon.zip", file=file)
lines = scan(file, what = "", sep="\n")
head(lines)
```

```
## [1] "type=weaksubj len=1 word1=abandoned pos1=adj stemmed1=n priorpolarity=negative"
## [2] "type=weaksubj len=1 word1=abandonment pos1=noun stemmed1=n priorpolarity=negative"
## [3] "type=weaksubj len=1 word1=abandon pos1=verb stemmed1=y priorpolarity=negative"
## [4] "type=strongsubj len=1 word1=abase pos1=verb stemmed1=y priorpolarity=negative"
## [5] "type=strongsubj len=1 word1=abasement pos1=anypos stemmed1=y priorpolarity=negative"
## [6] "type=strongsubj len=1 word1=abash pos1=verb stemmed1=y priorpolarity=negative"
```

The file contained is in a somewhat strange format, with one word per line coded as name=value pairs. So, we make a 'read_pairs' function that we apply to the result of extracting all name=value pairs per line:

```
read_pairs = function(x, fields) {
  x = x[x[,2] %in% fields, ]
  values = x[,3]
  names(values) = x[,2]
  values
}
m = stringr::str_match_all(lines, "(\\w+)=([^=]+)(?= |$)")
fields = m[[1]][,2]
lexicon = plyr::ldply(m, read_pairs, fields=fields)
saveRDS(lexicon, "data/lexicon.rds")
head(lexicon)
```

```
##         type len       word1   pos1 stemmed1 priorpolarity
## 1   weaksubj   1   abandoned    adj        n      negative
## 2   weaksubj   1 abandonment   noun        n      negative
```

```
## 3   weaksubj   1     abandon   verb        y      negative
## 4 strongsubj   1       abase   verb        y      negative
## 5 strongsubj   1   abasement anypos        y      negative
## 6 strongsubj   1       abash   verb        y      negative
```

Amazon Reviews

For this exercise we will download the Amazon reviews in the 'automotive' category as published at http://jmcauley.ucsd.edu/data/amazon/.

These reviews are stored in a gzipped while which contains one json record per line, so we use scan to split the file into lines, and then read each line with a custom functions that converts the line from json and into a data frame. We then use `ldply` to apply this function to each line:

```
url = "http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews_Automotive_5.json.gz"
download.file(url, destfile="reviews.json.gz")
lines = scan(gzfile("reviews.json.gz"), sep = "\n", what="")
readline = function(line) {
  x = rjson::fromJSON(line)
  x$helpful = NULL
  as.data.frame(x)
}
reviews = plyr::ldply(lines, readline)
```

```
## Warning: closing unused connection 5 (reviews.json.gz)
```

```
saveRDS(reviews, "data/reviews.rds")
head(reviews)
```

```
##       reviewerID       asin                reviewerName
## 1 A3F73SC1LY51OO B00002243X              Alan Montgomery
## 2 A2OS66SKYXULG2 B00002243X                    alphonse
## 3 A2I8LFSN2IS5EO B00002243X                       Chris
## 4 A3GT2EWQSO45ZG B00002243X                      DeusEx
## 5 A3ESWJPAVRPWB4 B00002243X                  E. Hernandez
## 6  A1ORODEBRN64C B00002243X James F. Magowan "Jimmy Mac"
##
## 1
## 2
## 3
## 4 I absolutley love Amazon!!!  For the price of a set of cheap Booster/Jumper Cables in a brick and m
## 5
## 6
##   overall                                    summary unixReviewTime
## 1       5 Work Well - Should Have Bought Longer Ones     1313539200
## 2       4                           Okay long cables     1315094400
## 3       5                Looks and feels heavy Duty      1374710400
## 4       5     Excellent choice for Jumper Cables!!!     1292889600
## 5       5    Excellent, High Quality Starter Cables     1341360000
## 6       5                      Compact and Strong !     1258156800
##    reviewTime
## 1 08 17, 2011
```

```
## 2  09 4, 2011
## 3 07 25, 2013
## 4 12 21, 2010
## 5  07 4, 2012
## 6 11 14, 2009
```