Topic Modeling with Mallet

Sowmya

3/20/2018

Purpose of this tutorial is to expand on what the author describes in Chapter 13. You have to first read Chapters 10 and 13 (and make notes), before trying this out.

Software Requirements:

Java needs to be installed on your computer. Go to this link for installation: https://java.com/en/download/help/index_installing.xml

After that: R library mallet needs to be installed in Rstudio (which inturn installs rJava). On MacOS, R may also prompt you to install some other some legacy java version - if it does, install that.

R libraries you need to install: mallet, wordcloud, XML

Data pre-processing:

The author again uses the XML corpus from Text classification example - this is provided as supplementary material in the electronic version of the textbook. I am providing it in the data folder inside this folder.

He first splits each file into 1000 chunks (instead of 10 in the classification example) using the function below and does these pre-processing: Extract text from xml, lowercase it, replace anything that is not an alpha-numeric character or a space or an apostrophe with a space.

Then, the next part is about splitting the text into multiple parts. Two methods are suggested for that. First one is - split the text into X parts based on the length of the text i.e., split after every 1000 words (for example). Second method is: splitting all texts into equal number of words. If you do that, sometimes, there may be a few leftover words at the end. This function addresses that too (how?).

```
makeFlexTextChunks <- function(doc.object, chunk.size=1000, percentage=TRUE){
  paras <- getNodeSet(doc.object,
                      "/d:TEI/d:text/d:body//d:p",
                      c(d = "http://www.tei-c.org/ns/1.0"))
  words <- paste(sapply(paras,xmlValue), collapse=" ")
  words.lower <- tolower(words)
  words.lower <- gsub("[^[:alnum:][:space:]']", " ", words.lower)
  words.l <- strsplit(words.lower, "\\s+")
  word.v <- unlist(words.l)
  x <- seq_along(word.v)
  if(percentage){
    max.length <- length(word.v)/chunk.size
    chunks.l <- split(word.v, ceiling(x/max.length))
  } else {
    chunks.l <- split(word.v, ceiling(x/chunk.size))
    #deal with small chunks at the end
    if(length(chunks.l[[length(chunks.l)]]) <=
       length(chunks.l[[length(chunks.l)]])/2){
      chunks.l[[length(chunks.l)-1]] <-
        c(chunks.l[[length(chunks.l)-1]],
          chunks.l[[length(chunks.l)]])
      chunks.l[[length(chunks.l)]] <- NULL
    }
  }
  chunks.l <- lapply(chunks.l, paste, collapse=" ")
  chunks.df <- do.call(rbind, chunks.l)
  return(chunks.df)
}
```

Once you have that function, the next step is to use this function to create such chunks for each file, combine them into a two column data frame where one column represents the text and chunk id and the other column shows the words in that chunk.

```
library(XML)
wd <- "~/Dropbox/ClassroomSlides-BothCourses/LING410X/Week11Mats/21Mar2018/"
inputDir <- paste(wd,"data/XMLAuthorCorpus",sep="")
stopwords_path <- paste(wd, "data/stoplist.csv",sep="")
files.v <- dir(path=inputDir , pattern=".*xml")
chunk.size <- 1000
topic.m <- NULL
for(i in 1:length(files.v)){
  doc.object <- xmlTreeParse(file.path(inputDir, files.v[i]),
                             useInternalNodes=TRUE)
  chunk.m <- makeFlexTextChunks(doc.object, chunk.size,
                                percentage=FALSE)
  textname <- gsub("\\..*","", files.v[i])
  segments.m <- cbind(paste(textname,
                            segment=1:nrow(chunk.m), sep="_"), chunk.m)
  topic.m <- rbind(topic.m, segments.m)
}
documents <- as.data.frame(topic.m, stringsAsFactors=F)
colnames(documents) <- c("id", "text")
```

This is all the pre-processing needed before you start training a topic model. mallet.import takes 5 arguments - ids for documents, text of the documents, stopwords list (csv file), preserve case (true or false), and a regular expression to split the text into tokens/words.

Once you have these, creating a topic model is a simple procedure. You need to specify the number of topics.

```
library(mallet)
```

```
## Loading required package: rJava
```

```
mallet.instances <- mallet.import(documents$id,documents$text,stopwords_path,FALSE,
                                  token.regexp="[\\p{L}']+")
topic.model <- MalletLDA(num.topics=43)
topic.model$loadDocuments(mallet.instances)

#The following line is optional.
#Mallet takes default parameters if you don't specify anything here.
topic.model$setAlphaOptimization(40, 80)

#This starts the training process
topic.model$train(400)
```

We can use explore this topic model to analysse topics and words in this data:

```r
#browse the vocabulary of this dataset and get some stats.
vocabulary <- topic.model$getVocabulary()
length(vocabulary)
```

```
## [1] 55444
```

```r
head(vocabulary)
```

```
## [1] "summer"   "topsail"  "schooner" "slipped"  "cove"     "trinidad"
```

```r
vocabulary[1:50]
```

```
##  [1] "summer"      "topsail"     "schooner"    "slipped"     "cove"
##  [6] "trinidad"    "head"        "dropped"     "anchor"      "edge"
## [11] "kelp"        "fields"      "fifteen"     "minutes"     "small"
## [16] "boat"        "deposited"   "beach"       "man"         "armed"
## [21] "long"        "squirrel"    "rifle"       "axe"         "carrying"
## [26] "food"        "clothing"    "brown"       "canvas"      "pack"
## [31] "watched"     "return"      "weigh"       "stand"       "sea"
## [36] "northwest"   "trades"      "disappeared" "ken"         "swung"
## [41] "broad"       "powerful"    "back"        "strode"      "resolutely"
## [46] "timber"      "mouth"       "river"       "john"        "cardigan"
```

```r
#word freqs has 3 columns. word, its frequency in entire corpus, its frequency per document.
word.freqs <- mallet.word.freqs(topic.model)

#this below line returns a matrix where each row is a topic,
#each column is a word in the dataset.
topic.words.m <- mallet.topic.words(topic.model,smoothed=TRUE,normalized=TRUE)

dim(topic.words.m)
```

```
## [1]    43 55444
```

```r
rowSums(topic.words.m)
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [36] 1 1 1 1 1 1 1 1
```

```r
topic.words.m[1:3, 1:3]
```

```
##              [,1]         [,2]         [,3]
## [1,] 3.440089e-07 3.440089e-07 3.440089e-07
## [2,] 8.585185e-07 8.585185e-07 8.585185e-07
## [3,] 7.240594e-07 7.240594e-07 7.240594e-07
```

```r
colnames(topic.words.m) <- vocabulary
keywords <- c("california", "ireland")
topic.words.m[, keywords]
```

```
##         california      ireland
##   [1,] 3.440089e-07 3.440089e-07
##   [2,] 8.585185e-07 8.585185e-07
##   [3,] 7.240594e-07 7.240594e-07
##   [4,] 5.918609e-07 5.918609e-07
##   [5,] 2.315544e-07 2.315544e-07
##   [6,] 1.337125e-07 1.337125e-07
##   [7,] 3.354310e-07 3.354310e-07
##  [8,] 1.937672e-07 1.937672e-07
##  [9,] 1.517795e-07 1.517795e-07
## [10,] 1.632222e-07 1.632222e-07
## [11,] 3.363235e-03 5.582788e-07
## [12,] 5.113770e-04 7.441480e-07
## [13,] 8.708564e-04 8.150889e-07
## [14,] 1.669692e-06 1.669692e-06
## [15,] 5.532627e-07 5.532627e-07
## [16,] 1.340916e-06 1.340916e-06
## [17,] 6.742904e-07 6.742904e-07
## [18,] 9.597928e-07 1.171818e-03
## [19,] 1.738793e-07 1.738793e-07
## [20,] 4.354327e-07 4.354327e-07
## [21,] 1.918428e-07 1.918428e-07
## [22,] 8.544914e-07 8.544914e-07
## [23,] 2.621855e-03 3.800916e-03
## [24,] 3.747765e-07 3.747765e-07
## [25,] 1.326942e-06 1.326942e-06
## [26,] 1.868290e-06 1.141439e-03
## [27,] 3.153644e-07 3.153644e-07
## [28,] 1.221412e-06 1.221412e-06
## [29,] 2.484411e-07 2.484411e-07
## [30,] 1.470912e-04 1.100059e-03
## [31,] 3.644561e-07 3.644561e-07
## [32,] 2.880987e-07 4.591153e-03
## [33,] 1.570525e-06 1.570525e-06
## [34,] 3.203872e-07 3.203872e-07
## [35,] 8.723949e-07 8.723949e-07
## [36,] 9.387479e-04 4.922371e-07
## [37,] 8.205965e-07 8.205965e-07
## [38,] 2.280596e-07 2.280596e-07
## [39,] 6.784761e-07 6.784761e-07
## [40,] 3.130012e-07 5.491972e-04
## [41,] 7.903345e-07 1.458333e-02
## [42,] 7.749249e-07 7.749249e-07
## [43,] 1.871218e-02 7.597964e-07
```

```r
#The topic which has the highest count of numbers in the row (indicating weights for words)
#can be thought of as an important topic.
imp.row <- which(rowSums(topic.words.m[, keywords]) ==max(rowSums(topic.words.m[, keywords])))

#10 most frequent words in the Topic given by imp.row
mallet.top.words(topic.model, topic.words.m[imp.row,], 10)
```

```
##                  words     weights
## san                san 0.020392147
## california california 0.018712175
## men                men 0.015989461
## mr                  mr 0.013846049
## land              land 0.013150888
## francisco    francisco 0.013035027
## city              city 0.012687447
## state            state 0.010022663
## states          states 0.007821320
## gold              gold 0.006199278
```

```r
#5 most frequent words in the Topic 10
mallet.top.words(topic.model, topic.words.m[10,], 5)
```

```
##         words    weights
## god       god 0.01987442
## heart   heart 0.01932686
## father father 0.01884151
## mother mother 0.01492141
## poor     poor 0.01273114
```

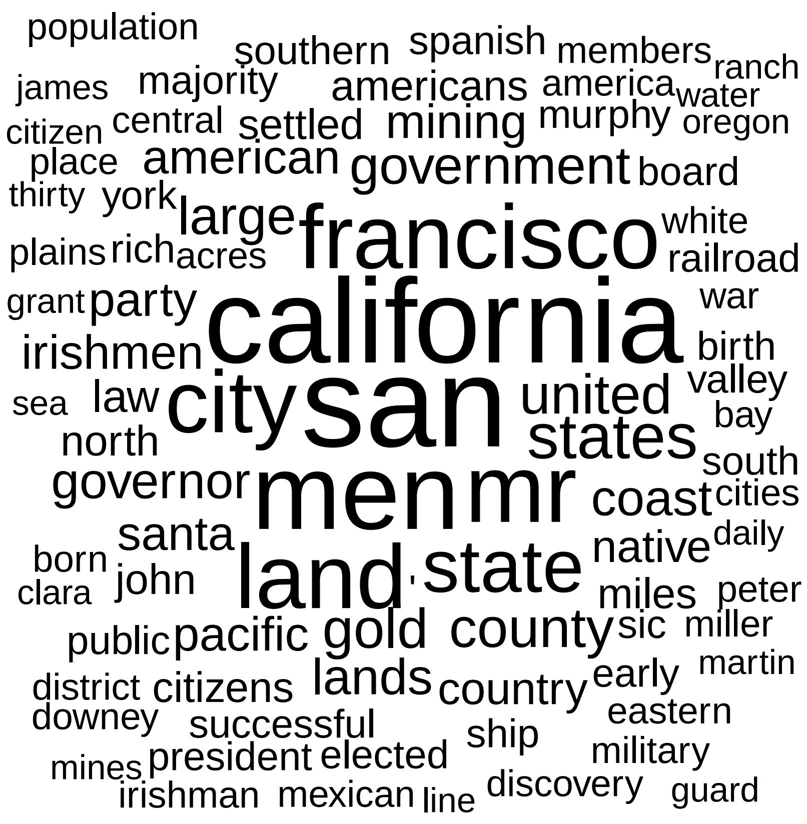
Little bit of visualization

This needs the library wordcloud.

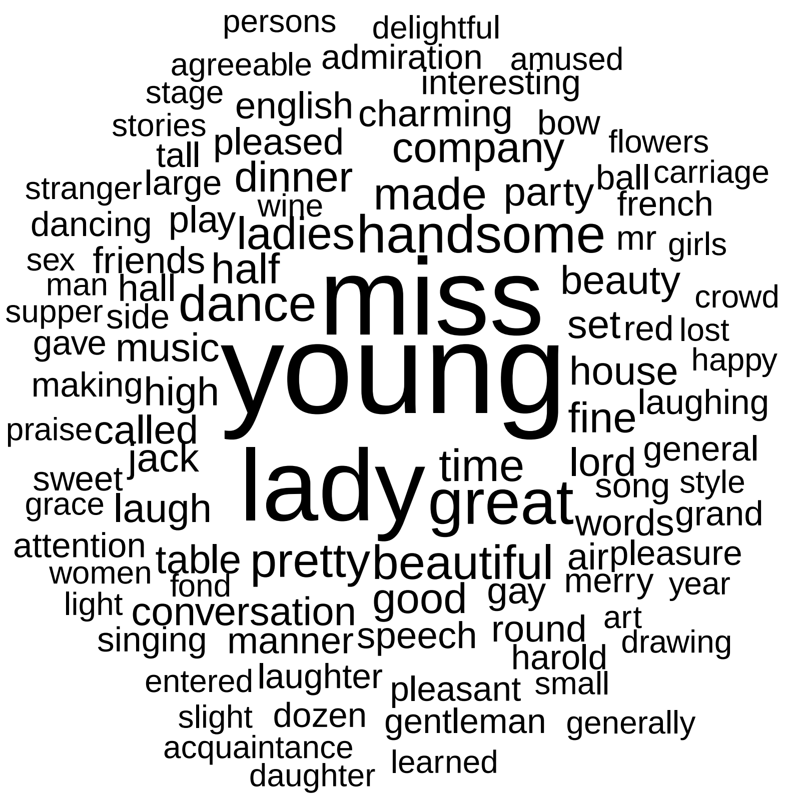
```r
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```r
#plotting a word cloud for the imp.row
topic.top.words <- mallet.top.words(topic.model,topic.words.m[imp.row,], 100)
wordcloud(topic.top.words$words,topic.top.words$weights,c(4,.8), rot.per=0, random.order=F)
```



```r
#plotting a word cloud for topic 20
topic.top.words <- mallet.top.words(topic.model,topic.words.m[20,], 100)
wordcloud(topic.top.words$words,topic.top.words$weights,c(4,.8), rot.per=0, random.order=F)
```


persons
delightful
agreeable
admiration
amused
interesting
stage
english
charming
stories
bow
pleased
company
flowers
tall
dinner
ball
carriage
stranger
large
made
party
wine
french
dancing
play
ladies
handsome
mr
girls
sex
friends
half
man
hall
miss
beauty
crowd
supper
side
dance
set
red
lost
gave
music
young
happy
making
high
house
laughing
praise
called
fine
jack
time
lord
general
sweet
lady
song
style
grace
laugh
great
words
grand
attention
air
pleasure
women
table
pretty
beautiful
merry
year
fond
light
conversation
good
gay
art
singing
manner
speech
round
drawing
harold
entered
laughter
pleasant
small
slight
dozen
gentleman
generally
acquaintance
learned
daughter