Quantitative Text Analysis
Exercise 9: Topic Models

31st July 2014, Essex Summer School

Kenneth Benoit and Paul Nulty

In today's lab we will use the R package `topicmodels` to discover topics in a set of UK newspaper articles that contain terms relating to immigration.

1. We will begin by loading the packages and data necessary and converting our corpus into the necessary format.

   (a) The corpus of newspaper articles is built in to `quanteda`. Load it and examine the attributes.

   ```
   library(quanteda)
   data(newsCorpus)
   ```

   (b) We will need to install the `topicmodels` package on the lab machines. Use the command:

   ```
   install.packages('topicmodels')
   ```

   (c) Building the topic model on large corpora with many topics can take a long time. We will select only the two most common newspapers from the corpus.

   ```
   paperCount <- table(newsCorpus$attribs$paperName)
   topPapers<- names(sort(paperCount, decreasing = TRUE)[1:2])
   reducedCorpus <- subset(newsCorpus, paperName %in% topPapers)
   ```

   (d) The next step is to make a dfm — again, we will trim low frequency words and remove stopwords to limit the size of the matrix. We use a list of custom stopwords (again built into quanteda) to exclude terms particular to this corpus, such as the names of the newspapers and copyright notices.

   ```
   byDocDfm <- dfm(reducedCorpus)
   byDocDfmTrim <- dfmTrim(byDocDfm, minCount=50, minDoc=20)
   data(custom_stopwords
   finalDfmByDoc <- stopwordsRemove(byDocDfmTrim, custom_stopwords)
   ```

   (e) The `topicmodels` package needs to work with a corpus in the triplet matrix format used by `tm`. We can convert the quanteda dfm to this format, after removing any empty rows:

   ```
   finalDfmByDoc <- finalDfmByDoc[which(rowSums(finalDfmByDoc) > 0),]
   finalTripletByDoc<- dfm2tmformat(finalDfmByDoc)
   ```

2. Now, with the correctly formatted triplet matrix, we are ready to fit and examine the model.

   (a) This code will fit a model with twenty topics using expectation-maximization. This command might take a few minutes.

```
newsLdaModel <- LDA(finalTripletByDoc, method="VEM", k = 20)
```

(b) If the lab machines aren't up to running the above command, the fitted model is available as a data object in quanteda: `data(newsLdaModel)`

(c) We can examine the words that contribute the most to each topic, and the topics that contribute the most to each document:

```
# which terms contribute most to each topic
get_terms(newsLdaModel, k=10)

# which is the dominant topic for each document
get_topics(newsLdaModel)

# the topic contribution of each topic to each document
postTopics <- data.frame(posterior(newsLdaModel)$topics)
```

(d) Finally, we can find the average contribution of a topic to an article from a particular newspaper, and compare newspapers on particular topics:

```
# get the newspaper names from the rownames of the topic contribution matrix
x <- sapply(rownames(postTopics),strsplit,'_')
paperNames <- sapply(x, head, n=1)
postTopics["paper"]<- paperNames
# Mean contribution of topic $X to document for each newspaper
#install.packages('plyr')
library(plyr)
byPaperTopics <-ddply(postTopics, "paper", numcolwise(mean))
barplot(byPaperTopics$X1, names.arg=byPaperTopics$paper, beside=TRUE)
```