[**Getting Started with Latent Dirichlet Allocation using RTextTools + topicmodels**](http://www.rtexttools.com/blog/getting-started-with-latent-dirichlet-allocation-using-rtexttools-topicmodels)

8/30/2011

[20 Comments](http://www.rtexttools.com/blog/getting-started-with-latent-dirichlet-allocation-using-rtexttools-topicmodels#comments)

RTextTools bundles a host of functions for performing supervised learning on your data, but what about other methods like [latent Dirichlet allocation](http://en.wikipedia.org/wiki/Latent_Dirichlet_allocation)? With some help from the [topicmodels](http://cran.r-project.org/web/packages/topicmodels/index.html) package, we can get started with LDA in just five steps. Text in green can be executed within R.  
  
**Step 1: Install RTextTools + topicmodels**  
We begin by installing and loading RTextTools and the topicmodels package into our R workspace.  
  
install.packages(c("RTextTools","topicmodels"))  
library(RTextTools)  
library(topicmodels)  
  
**Step 2: Load the Data**  
In this example, we will be using the bundled NYTimes dataset compiled by [Amber E. Boydstun](http://www.amberboydstun.com/). This dataset contains headlines from front-page NYTimes articles. We will take a random sample of 1000 articles for the purposes of this tutorial.  
  
data(NYTimes)  
data <- NYTimes[sample(1:3100,size=1000,replace=FALSE),]  
  
**Step 3: Create a DocumentTermMatrix**  
Using the *create\_matrix()* function in RTextTools, we'll create a DocumentTermMatrix for use in the*LDA()* function from package topicmodels. Our text data consists of the *Title* and *Subject* columns of the NYTimes data. We will be removing numbers, stemming words, and weighting the DocumentTermMatrix by term frequency.  
  
matrix <- create\_matrix(cbind(as.vector(data$Title),as.vector(data$Subject)), language="english", removeNumbers=TRUE, stemWords=TRUE, weighting=weightTf)  
  
**Step 4: Perform Latent Dirichlet Allocation**  
First we want to determine the number of topics in our data. In the case of the NYTimes dataset, the data have already been classified as a training set for supervised learning algorithms. Therefore, we can use the *unique()* function to determine the number of unique topic categories (*k*) in our data. Next, we use our matrix and this *k* value to generate the LDA model.  
  
k <- length(unique(data$Topic.Code))  
lda <- LDA(matrix, k)  
  
**Step 5: View the Results**  
Last, we can view the results by most likely term per topic, or most likely topic per document.  
  
terms(lda)  
Topic 1  "campaign"  Topic 2  "kill"      Topic 3  "elect"     Topic 4  "china"     Topic 5  "govern"    Topic 6  "fight" Topic 7  "leader"    Topic 8  "york"      Topic 9  "isra"      Topic 10 "win"       Topic 11 "report"    Topic 12 "plan"  
Topic 13 "republican"Topic 14 "aid"       Topic 15 "set"       Topic 16 "clinton"   Topic 17 "nation"    Topic 18 "hous"  
Topic 19 "iraq"      Topic 20 "bush"      Topic 21 "citi"      Topic 22 "rais"      Topic 23 "overview"  Topic 24 "money"  
Topic 25 "basebal"   Topic 26 "court"     Topic 27 "war"  
  
topics(lda)  
Output too long to display here. Try it out for yourself to see what it looks like