Information Retrieval

STA 325: Lab 3, Fall 2017

Today's agenda: Information Retrieval, document term matrices, queries, text mining, writing functions.

Programming partner's: You should have a programming partner for each lab, and you should switch off who is programming, and use each other for help. We will spend about 30–50 minutes per week on lab exercises and you will be expected to bring you laptops to class to work on these exercises in class. Myself and the TA's will be in class to help you.

Lab Tasks

1. Read the text data (provided) on Shakespeare into R with the load() function. The variable shakespeare contains the complete works of William Shakespeare. Apply the function length() to determine how many documents are present.

```
# Load the Text Mining Package #
require(tm)
```

```
## Loading required package: tm
```

```
## Loading required package: NLP
```

```
require(SnowballC)
```

```
## Loading required package: SnowballC
```

```
# Load the Data into R #
load(file="lab3.Rdata")
length(shakespeare)
```

```
## [1] 182
```

2. Create a corpus of this text using the following command Corpus(VectorSource(shakespeare)) and store this in the variable corp.

```
# Create a Text Corpus #
# Create a vector give the source text
# Creating a corpus of the vector of the
# source text
corp <- Corpus(VectorSource(shakespeare))
```

3. Use the tm_map() command to pre-process the data: remove punctuation, convert to lower case, and remove any numbers present in the data. After the processing, create a document term matrix (DTM) of this corpus and store it in the variable dtm. Apply as.matrix() to the final matrix.

```
# Perform the Data Processing #
# Convert to Lower Case #
corp <- tm_map(corp, content_transformer(tolower))

# Remove Punctuation #
corp <- tm_map(corp, removePunctuation)

# Remove Numbers #
corp <- tm_map(corp, removeNumbers)

# Create a Document Term Matrix for the Corpus #
```

```
dtm <- as.matrix(DocumentTermMatrix(corp))
dim(dtm)
```

```
## [1]   182 27710
```

4. Set the variable myQuery to the following c("something","rotten", "state","denmark")

```
# My Query #
myQuery <- c("something", "rotten", "state", "denmark")
```

5. Write a function called myTextMiner() that accepts as its inputs, a string vector containing keywords (akin to myQuery), and a corpus. The function should then process the corpus to first convert all entries to lower case, then remove all punctuation, then remove any numbers present and finally construct a document term matrix (DTM) that is normalized by the length of each document. Finally based on the query above, compute the Euclidean distance to each document in the shakespeare corpus. Note that the easiest way to do so involves including the query in the DTM. The function should return a subset of the normalized DTM with those columns that are shared with the query, with one additional column that contains the Euclidean distance for each document that has been normalized by document length. Name this column distanceMetric.} **Hint**: Remember to use as.matrix() on the DTM before any calculations.

```
# Inputs: Corpus, Query #
# Outputs: Document Term Matrix with Distance Column #
# Summary: Process the text data and produce DTM with distance metric column #
myTextMiner <- function(query, corpus){
    ## Perform pre-processing ##
    tempCorpus <- corpus
    tempCorpus <- tm_map(tempCorpus, content_transformer(tolower))
    tempCorpus <- tm_map(tempCorpus, removePunctuation)
    tempCorpus <- tm_map(tempCorpus, removeNumbers)
    ## Create a document Term Matrix ##
    myDTM <- as.matrix(DocumentTermMatrix(tempCorpus))
    ## Bind the query to the DTM ##
    ## Query Counts ##
    queryCounts <- as.numeric(table(query))
    ## Add a new row in the DTM that is the query ##
    myDTM <- rbind(myDTM, 0)
    myDTM[dim(myDTM)[1],myQuery] <- queryCounts
    ## Perform Calculation ##
    ## Normalize by Document Length ##
  DTMNorm <- myDTM/rowSums(myDTM)
  ## Calculate Euclidean Distance ##
  eucDistance <- sqrt(rowSums((scale(DTMNorm,center=
      DTMNorm[dim(DTMNorm)[1],],scale=FALSE)^2)))
  finalDTM <- cbind(DTMNorm[,query], eucDistance)
  colnames(finalDTM) <- c(query,"distanceMetric")
  return(finalDTM)
}
# Test Case #
results <- myTextMiner(myQuery, corp)
write.csv(results, 'Shakespeare.csv')
```