Data607_Project4_Rajan

Krishna Rajan

4/15/2018

```
##PROJECT 4: Document Classification
##It can be useful to be able to classify new "test" documents using already classified "training" docu

##Install Tools
require(stringr)
```

```
## Loading required package: stringr
```

```
require(tm)
```

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

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

```
require(RTextTools)
```

```
## Loading required package: RTextTools
```

```
## Loading required package: SparseM
```

```
##
## Attaching package: 'SparseM'
```

```
## The following object is masked from 'package:base':
##
##     backsolve
```

```
## Error: package or namespace load failed for 'RTextTools' in loadNamespace(i, c(lib.loc, .libPaths())
##  there is no package called 'glmnet'
```

```
require(SnowballC)
```

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

```
require(knitr)
```

```
## Loading required package: knitr
```

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:NLP':
##
##     annotate
```

```
require(dplyr)
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
##the following functions are helpful to wrap the functions together
toVCorpus <- function(file_path) {
  corpus <- file_path %>%
    paste(., list.files(.), sep = "/") %>%          # Create a vector of file paths
    lapply(readLines) %>%                           # Read the text in each file
    VectorSource() %>%                              # Turn into VectorSource
    VCorpus()                                       # Turn into VCorpus
  return(corpus)
}
docClean <- function(corpus) {
  corpus <- corpus %>%
    tm_map(removeNumbers) %>%                                  # Remove numbers
    tm_map(str_replace_all, "[[:punct:]]", " ") %>% # Remove punctuations
    tm_map(tolower) %>%                                          # Remove upper cases
    tm_map(PlainTextDocument) %>%                              # Transform back to PlainTextDocument
    tm_map(removeWords, stopwords("en")) %>%        # Remove stop words
    tm_map(stemDocument)                                         # Reduce to stems
  return(corpus)
}
addTag <- function(corpus, tag, value){
  for (i in 1:length(corpus)){
    meta(corpus[[i]], tag) <- value                    # Add the value to the specified tag
  }
  return(corpus)
}

##File Path for HAM & SPAM files
ham_paths <- "/Users/rajans/Desktop/CUNY/Data Acquition & Management/Project 4/HAM"
spam_paths <- "/Users/rajans/Desktop/CUNY/Data Acquition & Management/Project 4/SPAM"

# Create ham corpus
ham_corpus <- ham_paths %>%
  toVCorpus %>%
  docClean %>%
  addTag(tag = "ham_spam", value = "ham")
```

```
## Warning in FUN(X[[i]], ...): incomplete final line found on '/
## Users/rajans/Desktop/CUNY/Data Acquition & Management/Project 4/HAM/
## 00228.0eaef7857bbbf3ebf5edbbdae2b30493'
```

```
## Warning in FUN(X[[i]], ...): incomplete final line found on '/
## Users/rajans/Desktop/CUNY/Data Acquition & Management/Project 4/HAM/
## 0231.7c6cc716ce3f3bfad7130dd3c8d7b072'
```

```
## Warning in FUN(X[[i]], ...): incomplete final line found on '/
## Users/rajans/Desktop/CUNY/Data Acquition & Management/Project 4/HAM/
## 0250.7c6cc716ce3f3bfad7130dd3c8d7b072'
```

```r
# Create spam corpus
spam_corpus <- spam_paths %>%
  toVCorpus %>%
  docClean %>%
  addTag(tag = "ham_spam", value = "spam")


spamassassin_corpus <- c(ham_corpus, spam_corpus)

spamassassin_corpus <- spamassassin_corpus[sample(c(1:length(spamassassin_corpus)))]

# Check ham/spam proportion
spamassassin_corpus_prop <- spamassassin_corpus %>%
  meta(tag = "ham_spam") %>%
  unlist() %>%
  table()
spamassassin_corpus_prop
```

```
## .
##  ham spam
## 6952 2398
```

```r
spamassassin_dtm <- spamassassin_corpus %>%
  DocumentTermMatrix() %>%
  removeSparseTerms(1-(10/length(spamassassin_corpus)))
spamassassin_labels <- unlist(meta(spamassassin_corpus, "ham_spam"))

##N <- length(spamassassin_labels)
##split <- round(0.8*N)
##container <- create_container(spamassassin_dtm, labels = spamassassin_labels,trainSize = 1:split,
##testSize = (split+1):N,
##virgin = FALSE
##)
## unfortunately I am struck here as I am not able to create a container (getting an error message) and

##Training the Module
##svm_model_spamassassin <- train_model(container, "SVM")
##tree_model_spamassassin <- train_model(container, "TREE")
##maxent_model_spamassassin <- train_model(container, "MAXENT")
```