Using Statistics to Identify Spam

Anatomy of an email Message

Spam Data

```
head(list.files(path = file.path(data.dir, "easy_ham")))
```

```
[1] "00001.7c53336b37003a9286aba55d2945844c"
[2] "00002.9c4069e25e1ef370c078db7ee85ff9ac"
[3] "00003.860e3c3cee1b42ead714c5c874fe25f7"
[4] "00004.864220c5b6930b209cc287c361c99af1"
[5] "00005.bf27cdeaf0b8c4647ecd61b1d09da613"
[6] "00006.253ea2f9a9cc36fa0b1129b04b806608"
```

```
head(list.files(path = file.path(data.dir, "spam_2")))
```

```
[1] "00001.317e78fa8ee2f54cd4890fdc09ba8176"
[2] "00002.9438920e9a55591b18e60d1ed37d992b"
[3] "00003.590eff932f8704d8b0fcbe69d023b54d"
[4] "00004.bdcc075fa4beb5157b5dd6cd41d8887b"
[5] "00005.ed0aba4d386c5e62bc737cf3f0ed9589"
[6] "00006.3ca1f399ccda5d897fecb8c57669a283"
```

```
directories <- paste(data.dir, list.files(data.dir), sep = .Platform$file.sep)

file_counts <- sapply(directories, function(dir) length(list.files(dir)))

total_files <- sum(file_counts)
total_files
```

```
[1] 9353
```

```
file_counts
```

```
  D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham
                                                                  5052
D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham_2
```

```
                                                                    1401
  D:/Projects/Statistical-Computing/Case Studies/datasets/spam/hard_ham
                                                                     501
      D:/Projects/Statistical-Computing/Case Studies/datasets/spam/spam
                                                                    1001
    D:/Projects/Statistical-Computing/Case Studies/datasets/spam/spam_2
                                                                    1398
```

```
idx <- c(1:5, 15, 27, 68, 69, 329, 404, 427, 516, 852, 971)

fn <- list.files(directories[1], full.names = T)[idx]

sampleEmail <- sapply(fn, readLines)
```

Text Mining and Naive Bayes Classification

```
msg <- sampleEmail[[1]]
which(msg == "")[1]
```

```
[1] 63
```

```
match("", msg)
```

```
[1] 63
```

```
splitPoint <- match("", msg)

msg[ (splitPoint - 2):(splitPoint + 6)]
```

```
[1] "List-Archive: <https://listman.spamassassin.taint.org/mailman/private/exmh-workers/>"
[2] "Date: Thu, 22 Aug 2002 18:26:25 +0700"
[3] ""
[4] "    Date:        Wed, 21 Aug 2002 10:54:46 -0500"
[5] "    From:        Chris Garrigues <cwg-dated-1030377287.06fa6d@DeepEddy.Com>"
[6] "    Message-ID:  <1029945287.4797.TMDA@deepeddy.vircio.com>"
[7] ""
[8] ""
[9] "  | I can't reproduce this error."
```

```
header <- msg[1:(splitPoint - 1)]
body <- msg[ -(1:splitPoint) ]
```

```
splitMessage <- function(msg) {
   splitPoint <- match("", msg)

   header <- msg[ 1:(splitPoint - 1)]
   body <- msg[ -(1:splitPoint)]
```

```
   return(list(header = header, body = body))
}

sampleSplit <- lapply(sampleEmail, splitMessage)

header <- sampleSplit[[1]]$header
grep("Content-Type", header)
```

```
[1] 46
```

```
grep("multi", tolower(header))
```

```
integer(0)
```

```
header[46]
```

```
[1] "Content-Type: text/plain; charset=us-ascii"
```

```
headerList <- lapply(sampleSplit, function(msg) msg$header)

CTloc <- sapply(headerList, grep, pattern = "Content-Type")
CTloc
```

```
$`D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00001.7c53336b37003a928
[1] 46

$`D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00002.9c4069e25e1ef370c
[1] 45

$`D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00003.860e3c3cee1b42ead
[1] 42

$`D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00004.864220c5b6930b209
[1] 30

$`D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00005.bf27cdeaf0b8c4647
[1] 44

$`D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00014.cb20e10b2bfcb8210
[1] 54

$`D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00025.d685245bdc4444f44
integer(0)

$`D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00062.009f5a1a8fa88f0b3
[1] 21
```

```
$`D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00063.0acbc484a73f0e0b7
[1] 17

$`D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/0030.77828e31de08ebb58b
[1] 52

$`D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00368.f86324a03e7ae7070
[1] 31

$`D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00389.8606961eaeef7b921
[1] 52

$`D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/0047.5c3e049737a2813d4a
[1] 52

$`D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00775.0e012f37346784651
[1] 27

$`D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00883.c44a035e7589e8307
[1] 31
```

```
sapply(headerList, function(header) {
   CTloc <- grep("Content-Type", header)
   if( length(CTloc) == 0) return(NA)
   CTloc
})
```

```
D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00001.7c53336b37003a9286a

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00002.9c4069e25e1ef370c07

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00003.860e3c3cee1b42ead71

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00004.864220c5b6930b209cc

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00005.bf27cdeaf0b8c4647ec

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00014.cb20e10b2bfcb8210a1

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00025.d685245bdc4444f44fa

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00062.009f5a1a8fa88f0b382

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00063.0acbc484a73f0e0b727

 D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/0030.77828e31de08ebb58b5
```

```
D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00368.f86324a03e7ae7070cc

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00389.8606961eaeef7b921ce

 D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/0047.5c3e049737a2813d4ac

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00775.0e012f373467846510d

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00883.c44a035e7589e83076b
```

```
hasAttach <- sapply(headerList, function(header) {
   CTloc <- grep("Content-Type", header)

   if(length(CTloc) == 0) return(F)
   grepl("multi", tolower(header[CTloc]))
})

hasAttach
```

```
D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00001.7c53336b37003a9286a

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00002.9c4069e25e1ef370c07

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00003.860e3c3cee1b42ead71

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00004.864220c5b6930b209cc

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00005.bf27cdeaf0b8c4647ec

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00014.cb20e10b2bfcb8210a1

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00025.d685245bdc4444f44fa

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00062.009f5a1a8fa88f0b382

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00063.0acbc484a73f0e0b727

 D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/0030.77828e31de08ebb58b5

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00368.f86324a03e7ae7070cc

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00389.8606961eaeef7b921ce

 D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/0047.5c3e049737a2813d4ac
```

```
D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00775.0e012f373467846510d

D:/Projects/Statistical-Computing/Case Studies/datasets/spam/easy_ham/00883.c44a035e7589e83076b
```

```
header <- sampleSplit[[6]]$header
boundaryIdx <- grep("boundary=", header)
header[boundaryIdx]
```

```
[1] "    boundary=\"==_Exmh_-1317289252P\";"
```

```
sub(".*boundary=\"(.*)\";.*", "\\1", header[boundaryIdx])
```

```
[1] "==_Exmh_-1317289252P"
```

```
header2 <- headerList[[9]]
boundaryIdx2 <- grep("boundary=", header2)
header2[boundaryIdx2]
```

```
[1] "Content-Type: multipart/alternative; boundary=Apple-Mail-2-874629474"
```

```
sub('.*boundary="(.*)";.*', "\\1", header2[boundaryIdx2])
```

```
[1] "Content-Type: multipart/alternative; boundary=Apple-Mail-2-874629474"
```

```
boundary2 <- gsub('"', "", header2[boundaryIdx2])

sub(".*boundary= *(.*);?.*", "\\1", boundary2)
```

```
[1] "Apple-Mail-2-874629474"
```

```
boundary <- gsub('"', "", header[boundaryIdx])
sub(".*boundary= *(.*);?.*", "\\1", boundary)
```

```
[1] "==_Exmh_-1317289252P;"
```

```
getBoundary <- function(header) {
   boundaryIdx <- grep("boundary=", header)
   boundary = gsub('"', "", header[boundaryIdx])
   gsub(".*boundary= *([^;]*);?.*", "\\1", boundary)
}
```

```
boundary <- getBoundary(headerList[[15]])
body <- sampleSplit[[15]]$body

bString <- paste("--", boundary, sep = "")
bStringLocs <- which(bString == body)
bStringLocs
```

```
[1]  2 35
```

```
eString <- paste("--", boundary, "--", sep = "")
eStringLoc <- which(eString == body)
eStringLoc
```

```
[1] 77
```

```
msg <- body[ (bStringLocs[1] + 1) : (bStringLocs[2] - 1)]
tail(msg)
```

```
[1] ">"     ">Yuck" ">  "   ">"     ""      ""
```

```
msg <- c(msg, body[ (eStringLoc + 1) : length(body) ])
tail(msg)
```

```
[1] ">  " ">"   ""    ""    ""    ""
```

Handle Attachments

Extracting Words from the Message Body

```
head(sampleSplit[[1]]$body)
```

```
[1] "    Date:        Wed, 21 Aug 2002 10:54:46 -0500"
[2] "    From:        Chris Garrigues <cwg-dated-1030377287.06fa6d@DeepEddy.Com>"
[3] "    Message-ID:  <1029945287.4797.TMDA@deepeddy.vircio.com>"
[4] ""
[5] ""
[6] "  | I can't reproduce this error."
```

```
msg <- sampleSplit[[3]]$body
head(msg)
```

```
[1] "Man Threatens Explosion In Moscow "
[2] ""
[3] "Thursday August 22, 2002 1:40 PM"
[4] "MOSCOW (AP) - Security officers on Thursday seized an unidentified man who"
[5] "said he was armed with explosives and threatened to blow up his truck in"
[6] "front of Russia's Federal Security Services headquarters in Moscow, NTV"
```

Stemming

```
exclude_word_list <- stopwords(kind = "en")
```

Convert To Wordlist

```
tolower(gsub("[[:punct:]0-9[:blank:]]+", " ", msg))
```

```
 [1] "man threatens explosion in moscow "
 [2] ""
 [3] "thursday august pm"
 [4] "moscow ap security officers on thursday seized an unidentified man who"
 [5] "said he was armed with explosives and threatened to blow up his truck in"
 [6] "front of russia s federal security services headquarters in moscow ntv"
 [7] "television reported "
 [8] "the officers seized an automatic rifle the man was carrying then the man"
 [9] "got out of the truck and was taken into custody ntv said no other details"
[10] "were immediately available "
[11] "the man had demanded talks with high government officials the interfax and"
[12] "itar tass news agencies said ekho moskvy radio reported that he wanted to"
[13] "talk with russian president vladimir putin "
[14] "police and security forces rushed to the security service building within"
[15] "blocks of the kremlin red square and the bolshoi ballet and surrounded the"
[16] "man who claimed to have one and a half tons of explosives the news"
[17] "agencies said negotiations continued for about one and a half hours outside"
[18] "the building itar tass and interfax reported citing witnesses "
[19] "the man later drove away from the building under police escort and drove"
[20] "to a street near moscow s olympic penta hotel where authorities held"
[21] "further negotiations with him the moscow police press service said the"
[22] "move appeared to be an attempt by security services to get him to a more"
[23] "secure location "
[24] ""
[25] " yahoo groups sponsor "
[26] " dvds free s p join now"
[27] "http us click yahoo com pt ybb nxieaa mg haa gsolb tm"
[28] " "
[29] ""
[30] "to unsubscribe from this group send an email to "
[31] "forteana unsubscribe egroups com"
[32] ""
[33] " "
[34] ""
[35] "your use of yahoo groups is subject to http docs yahoo com info terms "
[36] ""
[37] ""
[38] ""
```

```
msg[ c(1, 3, 26, 27) ]
```

```
[1] "Man Threatens Explosion In Moscow "
```

```
[2] "Thursday August 22, 2002 1:40 PM"
[3] "4 DVDs Free +s&p Join Now"
[4] "http://us.click.yahoo.com/pt6YBB/NXiEAA/mG3HAA/7gSolB/TM"
```

```
cleanMsg <- tolower(gsub("[[:punct:]0-9[:blank:]]+", " ", msg))
cleanMsg[ c(1, 3, 26, 27) ]
```

```
[1] "man threatens explosion in moscow "
[2] "thursday august pm"
[3] " dvds free s p join now"
[4] "http us click yahoo com pt ybb nxieaa mg haa gsolb tm"
```

```
words <- unlist(strsplit(cleanMsg, "[[:blank:]]+"))

words <- words[ nchar(words) > 1 ]

words <- words[ ! (words %in% exclude_word_list) ]

head(words)
```

```
[1] "man"       "threatens" "explosion" "moscow"    "thursday"  "august"
```

```
findMsgWords <- function(msg, exclude) {

   cleanMsg <- tolower(gsub("[[:punct:]0-9[:blank:]]+", " ", msg))

   words <- unlist(strsplit(cleanMsg, "[[:blank:]]+"))

   keep <- sapply(words, function(word) return(!(word %in% exclude)))


   return(words[ keep ])
}
```

Prep Wrap-Up

```
dropAttach <- function(body, boundary) {

   if(is.null(body)) {
      return("")
   }

   bString <- paste("--", boundary, sep = "")
   bStringLocs <- which(bString == body)

   eString <- paste("--", boundary, "--", sep = "")
```

```
   eStringLoc <- which(eString == body)

   if(length(bStringLocs) == 2) {
      msg <- body[ (bStringLocs[1] + 1) : (bStringLocs[2] - 1)]
   }

   if(length(eStringLoc) > 0) {
      msg <- c(msg, body[ (eStringLoc + 1) : length(body) ])
   }

   return(msg)
}

processAllWords <- function(dirName, stopWords) {
   # read all files in the directory
   fileNames <- list.files(dirName, full.names = T)

   # drop files that are not email, i.e., cmds
   notEmail <- grep("cmds$", fileNames)

   if( length(notEmail) > 0) fileNames <- fileNames[ -notEmail ]

   messages <- lapply(fileNames, readLines, encoding = "latin1")

   # split header and body
   emailSplit <- lapply(messages, splitMessage)

   # put body and header in own lists
   bodyList <- lapply(emailSplit, function(msg) msg$body)
   headerList <- lapply(emailSplit, function(msg) msg$header)
   rm(emailSplit)

   # determine which messages have attachments
   hasAttach <- sapply(headerList, function(header) {

      CTloc <- grep("Content-Type", header)

      if( length(CTloc) == 0) return(0)

      multi <- grep("multi", tolower(header[CTloc]))

      if( length(multi) == 0 ) return(0)

      multi
```

```
  })

  hasAttach <- which(hasAttach > 0)

  # find boundary string for messages with attachments
  boundaries <- sapply(headerList[hasAttach], getBoundary)

  # drop attachments from message body
  bodyList[hasAttach] <- mapply(dropAttach, bodyList[hasAttach],
                                boundaries, SIMPLIFY = F)

  # extract words from body
  msgWordsList <- lapply(bodyList, findMsgWords, stopWords)

  invisible(msgWordsList)
}
```

Build Email Database

```
msgWordList <- lapply(directories, processAllWords, stopWords = exclude_word_list)
```

```
Warning in FUN(X[[i]], ...): incomplete final line found on 'D:/
Projects/Statistical-Computing/Case Studies/datasets/spam/hard_ham/
00228.0eaef7857bbbf3ebf5edbbdae2b30493'

Warning in FUN(X[[i]], ...): incomplete final line found on 'D:/
Projects/Statistical-Computing/Case Studies/datasets/spam/hard_ham/
0231.7c6cc716ce3f3bfad7130dd3c8d7b072'

Warning in FUN(X[[i]], ...): incomplete final line found on 'D:/
Projects/Statistical-Computing/Case Studies/datasets/spam/hard_ham/
0250.7c6cc716ce3f3bfad7130dd3c8d7b072'

Warning in FUN(X[[i]], ...): incomplete final line found on 'D:/
Projects/Statistical-Computing/Case Studies/datasets/spam/spam/
00136.faa39d8e816c70f23b4bb8758d8a74f0'

Warning in FUN(X[[i]], ...): incomplete final line found on 'D:/
Projects/Statistical-Computing/Case Studies/datasets/spam/spam/
0143.260a940290dcb61f9327b224a368d4af'
```

```
numMsgs <- sapply(msgWordList, length)
numMsgs
```

```
[1] 5051 1400  500 1000 1397
```

```
isSpam <- rep(c(FALSE, FALSE, FALSE, TRUE, TRUE), numMsgs)

msgWordsList <- unlist(msgWordList, recursive = F)
```

Naive Bayes Classifier Implementation

Train / Test Split

```
numEmail <- length(isSpam)

numSpam <- sum(isSpam)
numHam <- numEmail - numSpam

set.seed(418910)

testSpamIdx <- sample(numSpam, size = floor(numSpam/3))
testHamIdx <- sample(numHam, size = floor(numHam/3))

testMsgWords <- c((msgWordsList[isSpam])[testSpamIdx],
                  (msgWordsList[!isSpam])[testHamIdx])

trainMsgWords <- c((msgWordsList[isSpam])[ - testSpamIdx ],
                   (msgWordsList[!isSpam])[ - testHamIdx])

testIsSpam <- rep(c(T, F),
                  c(length(testSpamIdx), length(testHamIdx)))

trainIsSpam <- rep(c(T, F),
                   c(numSpam - length(testSpamIdx),
                     numHam - length(testHamIdx)))
```

Probability Estimates from Training Sample

```
bow <- unique(unlist(trainMsgWords))

length(bow)
```

```
[1] 69502
```

```
spamWordCounts <- rep(0, length(bow))

names(spamWordCounts) = bow
```