Neeraj Asthana
nasthan2
2/12/2016
Stat 480 Homework 3

*Most of the code was taken and modified from code demonstrated in class

**Question 2**
I changed the dropAttach function to a new function called includeAttach which includes the words inside HTML and plaintext attachments. I also changed the code for processAllWords to use includeAttach instead of dropAttach. Code for includeAttach:

```
includeAttach = function(body, boundary){

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

  # if there are fewer than 2 beginning boundary strings,
  # there is on attachment to drop
  if (length(bStringLocs) <= 1) return(body)

  # do ending string processing
  eString = paste("--", boundary, "--", sep = "")
  eStringLoc = which(eString == body)

  # if no ending boundary string, grab contents between the first
  # two beginning boundary strings as the message body
  n = length(body)
  if (length(eStringLoc) == 0)
    return(body[c( (bStringLocs[1] + 1) : (bStringLocs[2] - 1), (bStringLocs[2] + 1) : n  )])

  # typical case of well-formed email with attachments
  # grab contents between first two beginning boundary strings and
  # add lines after ending boundary string
  if (eStringLoc < n)
    return( body[ c( (bStringLocs[1] + 1) : (bStringLocs[2] - 1), (bStringLocs[2] + 1) :
(eStringLoc - 1),
                  ( (eStringLoc + 1) : n )) ] )

  # fall through case
  # note that the result is the same as the
  # length(eStringLoc) == 0 case, so code could be simplified by
  # dropping that case and modifying the eStringLoc < n check to
  # be 0 < eStringLoc < n
  return( body[ (bStringLocs[1] + 1) : (bStringLocs[2] - 1) ])
}
```

The accuracy of the original classifier (using dropAttach) was .9396662 and the accuracy of the new model (using includeAttach) is .8735558, so we see a significant (about 6%) drop in accuracy when using includeAttach.

**Question 3**

I created a new function called myGetBoundary which functions the same way as getBoundary. I included my code for this function below:

```
myGetBoundary = function(header){
  #split all lines on boundary= and see which ones have a split that is greater than 1
(meaning that the boundary is contained on that line)
  splits <- strsplit(header, "boundary=")
  line = ""
  for(i in splits){
    if(length(i) > 1){
      line = i
    }
  }

  line = line[2]

  #remove all whitespace and quotes
  line = gsub('"', "", line)
  line = gsub(' ', "", line)

  #remove semicolon if it exists
  line = unlist(strsplit(line, ";"))[1]
  return(line)
}
```

**Question 6**

I used the stemDocument() function inside of the “tm” package in order to stem words with similar roots (this function can be seen inside of the findMsgWords function and I also ensured I used dropAttach instead of includeAttach inside of processAllWords). I then ran the same classifier again however my accuracy slightly improved to .9326059 (while using stemming), so the choice to use stemming was not advisable.