STA 141 Homework3

Shuxin Li

```
setwd("..")
setwd('E:/U course/STA 141/SpamAssassinTraining')
dir=list.files()
trainMessages=vector("list",5)
names(trainMessages)=dir

#------------  function -------------------

attachment<-function(x,i)
{ # function to get attachments out

    ath=tt2[x[i]:x[i+1]]
    spaceath=which(ath=="")
    if(length(names(table(grepl('[C|c]ontent-[T|t]ype',ath))))==2)
    {

        if(grep('[C|c]ontent-[T|t]ype',ath)[1]<spaceath[1])
        {
          athh=ath[2:(spaceath[1]-1)]
          con=textConnection(athh)
          header_ath=read.dcf(con, all=TRUE)
          close(con)
          body_ath=ath[spaceath[1]:(length(ath)-1)]
          return(list(header=header_ath, body=body_ath))
        }

        if(grep('[C|c]ontent-[T|t]ype',ath)[1]>spaceath[1])
        {
          athh=ath[grep('[C|c]ontent-[T|t]ype',ath)[1]]
          con=textConnection(athh)
          header_ath=read.dcf(con, all=TRUE)
          close(con)
          body_ath=ath[spaceath[1]:(grep('[C|c]ontent-[T|t]ype',ath)[1]-1)]
          return(list(header=header_ath, body=body_ath))
        }
    }

    if(length(names(table(grepl('[C|c]ontent-[T|t]ype',ath))))==1)
    {
      body_ath=ath[2:(length(ath)-1)]
      header_ath='null'
      return(list(header=header_ath, body=body_ath))
    }
}

attachmentspecial<-function(x,i)
{  # function to get attachments out

    ath=tt2[x[i]:length(tt2)]
    spaceath=which(ath=="")
```

```
    if(length(names(table(grepl('[C|c]ontent-[T|t]ype',ath))))==2)
    {

        if(grep('[C|c]ontent-[T|t]ype',ath)[1]<spaceath[1])
        {
          athh=ath[2:(spaceath[1]-1)]
          con=textConnection(athh)
          header_ath=read.dcf(con, all=TRUE)
          close(con)
          body_ath=ath[spaceath[1]:(length(ath)-1)]
          return(list(header=header_ath, body=body_ath))
        }

        if(grep('[C|c]ontent-[T|t]ype',ath)[1]>spaceath[1])
        {
          athh=ath[grep('[C|c]ontent-[T|t]ype',ath)[1]]
          con=textConnection(athh)
          header_ath=read.dcf(con, all=TRUE)
          close(con)
          body_ath=ath[spaceath[1]:(grep('[C|c]ontent-[T|t]ype',ath)[1]-1)]
          return(list(header=header_ath, body=body_ath))
        }
    }

    if(length(names(table(grepl('[C|c]ontent-[T|t]ype',ath))))==1)
    {
      body_ath=ath[2:(length(ath)-1)]
      header_ath='null'
      return(list(header=header_ath, body=body_ath))
    }
}

#-------- begin ----------

for(p in 1:5)
{
  setwd('E:/U course/STA 141/SpamAssassinTraining')
  dir=list.files()
  setwd(dir[p])
  emails=list.files()
  emailfile=vector("list", length(emails))
  names(emailfile)=emails

  for(j in 1:length(emails))
  { # ttw is the whole email message
    ttw=readLines(emails[j])
    ttw=gsub("\t"," ",ttw)

      if(grepl("mv ",ttw[1])=="TRUE")
      { # assuming the first Line containing "mv "
```

```
  body=ttw[1:length(ttw)]
  header="null"
  achment="null"
  emailfile[[j]]=list('header'=header, 'body'=body, 'attachment'=achment)
}

if(grepl("mv ",ttw[1])=="FALSE")
{ # assuming the first Line not containing "mv "
  space=which(ttw=="")

  # ---------------------Get email header----------------------------

  tt1=ttw[1:(space[1]-1)]
  con=textConnection(tt1)
  header=read.dcf(con, all=TRUE)
  close(con)

  # verify if there is an attachment
  bndr_indx=which((grepl('boundary',tt1) | grepl('BOUNDARY', tt1) | grepl('Boundary', tt1))==TRUE)

  # split the email as tt1 and tt2 two parts
  tt2=ttw[space[1]:length(ttw)]

 # -----------Get email boundary part and the boundary location---------

    if(sum(bndr_indx)>0)
    { # verify if there is a boundary

      # Step1: get part of attachment signal or boundary
      bndr=tt1[bndr_indx]
      sign_atch=strsplit(bndr,';')
      sign_atch=gsub("BOUNDARY","boundary",sign_atch[[1]])
      sign_atch=gsub("Boundary","boundary",sign_atch)
      sign_atch=gsub('\"',"",sign_atch)
      sign_atch=gsub('-00',"",sign_atch)
      idx=grep('boundary', sign_atch)
      sign_atch=sign_atch[idx]
      sign_atch=strsplit(sign_atch,'boundary=')
      len=length(strsplit(sign_atch[[1]][2],"")[[1]]) # compute the length of string
      # compute the length of string after split by "+"
      lenspecial=length(strsplit(strsplit(sign_atch[[1]][2],'[+]')[[1]][2],"")[[1]])
      sign_atch=sign_atch[[1]][2]
      copy=sign_atch

      if(grepl("[+]", copy))
      { # find if there is "+" in the boundary part
        sign_atch=substr(sign_atch, len-lenspecial+1, len)
      }

      if(grepl("[+]", copy)==FALSE)
```

```
    {  # find if there is "+" in the boundary part
      sign_atch=substr(copy, len-5, len)
    }


    # Step2: find possiable attachment location
    #        where ath_idx1 is possible location
    #    ath_iw is possible location without the last value in ath_idx1 in most case
    #              athi is the right boundary location in attachment part
    if(length(names(table(grepl(sign_atch,tt2))))==1)
    { # assuming there is no boundary matched becaues of the upper of boundary in attachment
      sign_atch=toupper(sign_atch)
      ath_idx1=grep(sign_atch, tt2)
    }

    if(length(names(table(grepl(sign_atch,tt2))))==2)
    { # assuming there is boundary found in attachment part
      ath_idx1=grep(sign_atch, tt2)
    }

    if(length(ath_idx1)>1)
    { # assuming there is the end boundary in attachment
      athi_w=ath_idx1[-length(ath_idx1)]
    }
    if(length(ath_idx1)==1)
    { # assuming there is no the end boundary in attachment
      athi_w=ath_idx1
    }

    # step3: delete the wrong attachment location
    wrong=sapply(1:length(athi_w), function(i) length(strsplit(tt2[athi_w],"")[[i]] ))
    right_loc=which(wrong==as.numeric(names(sort(table(wrong))[length(names(sort(table(wrong))))]))
)
    athi=athi_w[right_loc]
  }


    # -------------Get the email body, there are three situations-------------

  # Situation 1: if there is no attachment, then get the body directly.
  if(sum(bndr_indx)==0) body=tt2[-1]

  # situation2: if there is an attachment but not followed by header,then get the body.
  if(sum(bndr_indx)>0)
  { # assuming there is an attachment

      if(length(ath_idx1)!=1)
      { # assuming there is an end boundary

        if(length(names(table(grepl('[^""]',tt2[1:(athi[1]-1)])))==2)
```

```
      { # verify if there is anything between the first boundary and the first space
        body=c(tt2[2:(athi[1]-1)],tt2[(ath_idx1[length(ath_idx1)]+1):length(tt2)])
      }
    }

    if(length(ath_idx1)==1)
    { # assuming there is no end boundary

      if(length(names(table(grepl('[^""]',tt2[1:(athi[1]-1)]))))==2)
      { # verify if there is anything between the first boundary and the first space
        body=tt2[2:(athi[1]-1)]
      }
    }
}

# situation3: if there is an attachment followed by header, get the body
if(sum(bndr_indx)>0)
{ # assuming there is an attachment

    if(length(ath_idx1)!=1)
    { # assuming there is an end boundary

      if(length(names(table(grepl('[^""]',tt2[1:(athi[1]-1)]))))==1)
      { # verify if there is nothing between the first boundary and the first space

        body=tt2[(ath_idx1[length(ath_idx1)]+1):length(tt2)]
      }
    }

    if(length(ath_idx1)==1)
    { # assuming there is no end boundary

      if(length(names(table(grepl('[^""]',tt2[1:(athi[1]-1)]))))==1)
      {# verify if there is nothing between the first boundary and the first space
        body="null"
      }
    }
 }

# Get email attachment

 if(sum(bndr_indx)>0)
 { # assuming there is an attainment

    if(length(ath_idx1)!=1)
    { # assuming there is an end boundary
      x=c(athi,ath_idx1[length(ath_idx1)])
      achment=vector("list",(length(x)-1))

      for(i in 1:(length(x)-1))
```

```
                {
                  achment[[i]]=attachment(x,i)
                }
              }

              if(length(ath_idx1)==1)
              { # assuming there is no end boundary
                x=athi
                achment=vector("list",1)
                achment[[1]]=attachmentspecial(x,1)
              }
            }

        # assuming there is no attachment
        if(sum(bndr_indx)==0) achment="null"

        emailfile[[j]]=list('header'=header, 'body'=body, 'attachment'=achment)
     }

   }
  trainMessages[[p]]=emailfile
  setwd('..')
}
```

```
## Warning in readLines(emails[j]): incomplete final line found on
## '00228.0eaef7857bbbf3ebf5edbbdae2b30493'
```

```
## Warning in readLines(emails[j]): incomplete final line found on
## '0231.7c6cc716ce3f3bfad7130dd3c8d7b072'
```

```
## Warning in readLines(emails[j]): incomplete final line found on
## '0250.7c6cc716ce3f3bfad7130dd3c8d7b072'
```

```
## Warning in readLines(emails[j]): incomplete final line found on
## '00136.faa39d8e816c70f23b4bb8758d8a74f0'
```

```
## Warning in readLines(emails[j]): incomplete final line found on
## '0143.260a940290dcb61f9327b224a368d4af'
```