# Topical Crawler in R

library(RCurl)

library(XML)

library(stringr)

library(httr)

htmlToText <- function(input, ...) {

###---PACKAGES ---###

require(RCurl)

require(XML)

# convert HTML to plain text

convert\_html\_to\_text <- function(html) {

doc <- htmlParse(html, asText = TRUE)

text <- xpathSApply(doc, "//text()[not(ancestor::script)][not(ancestor::style)][not(ancestor::noscript)][not(ancestor::form)]", xmlValue)

return(text)

}

collapse\_text <- function(txt) {

return(paste(txt, collapse = " "))

}

html.list <- lapply(input, evaluate\_input)

text.list <- lapply(html.list, convert\_html\_to\_text)

text.vector <- sapply(text.list, collapse\_text)

return(text.vector)

}

frontier <- c("http://www.bbc.com")

sterms<-c("technology","school","web","sports","news")

num <- 50 #total number of items to crawl

result <- c()

repository<-NULL

tText<-NULL

j <- 0 #number of items in the repository

while (j < num){

if(length(frontier)<1){

break

}

#grab the first item in the frontier and place in the "exploredlink" variable

exploredlink<-frontier[1]

frontier<-frontier[-1]

if(str\_detect(exploredlink,"\\.jpg$"))

{

next

}

#fill in your code here

doc <- tryCatch(getURL(exploredlink),error=function(cond){return("")})

if(str\_length(doc)<10){

next

}

doc <- htmlParse(doc)

titleText <- xmlToDataFrame(nodes = getNodeSet(doc, "//title"))

titleText <- as.vector(titleText$text)

titleText <- unique(titleText)

bodyText<-tryCatch(htmlToText(content(GET(exploredlink),type="text/html",as="text")),error=function(cond){return("")})

domain<-str\_extract(exploredlink,pattern = ".\*\\.com")

repository<-append(repository,frontier[1])

tText<-append(tText,titleText)

if(is.na(domain)){

next

}

anchor <- getNodeSet(doc, "//a")

anchor <- sapply(anchor, function(x) xmlGetAttr(x, "href"))

if(length(anchor)>0){

temp <- c()

for(i in 1:length(anchor)){

if(is.null(anchor[[i]])){

next

}

if(!str\_detect(anchor[[i]][1],"^http")){

next

}

if(str\_detect(anchor[[i]][1],domain)){

next

}

temp <- append(temp,str\_trim(anchor[[i]][1]))

}

anchor <- temp

rm(temp)

frontier<-append(frontier,anchor)

frontier <- unique(frontier)

print(frontier)

}

}

