01_CollectHTML

Moritz Muller

February 14, 2019

This project collects all html from websites and searches the files for email addresses and appropriate tags. In principle, all websites can be used. In our case, we only select websites of interest groups to use the email addresses to send out a large-n survey

Load requires packages

```
library(Rcrawler)
library(stringr)
library(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
```

First trial run

with single website: download all html of website, clean the HTML and look for regex matches for email. I use MaxDepth = 1 since a majority of websites disclose emails addresses within one level from the starting page. However, if you look for private email addresses, they yend to be burried deeped in the website, so it is advisable to set MaxDepth to 2 or 3 in that case (which increases runtime).

The loop counts the number of mentions of the email addresses on the websites and returns the most likely general email address. It is entirely possible to adapt the search pattern to look for personal email addresses (e.g. FIRSTNAME.LASTNAME@STRING.STRING).

```
Rcrawler(Website = "https://www.electricmobilityeurope.eu/", no_cores = 4, no_conn = 4, MaxDepth = 1, D
```

```
##
## Preparing multihreading cluster .. In process : 1..
## Progress: 5.26 %  :  1  parssed from  19  | Collected pages: 1  | Level: 1
## In process : 2..3..4..5..
## Progress: 3.33 %  :  2  parssed from  60  | Collected pages: 5  | Level: 1
## In process : 6..7..8..9..
## Progress: 6.67 %  :  6  parssed from  90  | Collected pages: 9  | Level: 1
## In process : 10..11..12..13..
## Progress: 9.80 %  :  10  parssed from  102  | Collected pages: 12  | Level: 1
## In process : 14..15..16..17..
## Progress: 13.21 %  :  14  parssed from  106  | Collected pages: 13  | Level: 1
## In process : 18..19..20..21..
```

```
## Progress: 16.07 %  :  18  parssed from  112  | Collected pages: 15  | Level: 2
## + Check INDEX dataframe variable to see crawling details
## + Collected web pages are stored in Project folder
## + Project folder name : electricmobilityeurope.eu-191249
## + Project folder path : ./data/test/electricmobilityeurope.eu-191249
```

```
#Initiate files for loop
folders.website <- data.frame(list.files("./data/test"))
results <- list(list())
recommended.emails <- list()

#run loop
for(i in 1:nrow(folders.website)){
  files.html <- data.frame(list.files(paste0("./data/test/",folders.website[i,])))
  URL <- as.character(folders.website[i,])
  for (l in 1:nrow(files.html)){
    rawHTML <- paste(readLines(paste0("./data/test/", folders.website[i,],"/", files.html[l,])), collaps
    results[[URL]][l] <- as.data.frame(str_extract_all(rawHTML, "[a-zA-Z][a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+
  }
  temporary.stash <- as.data.frame(unlist(results[[URL]]))
  names(temporary.stash)[1] <- "emails"
  most_probable <- temporary.stash%>%
    count(emails)%>%
    top_n(3,n)
  most_probable <- most_probable[1:10,]
  recommended.emails[[i]] <- most_probable[order(most_probable$n, decreasing = T),]
  write.csv(temporary.stash, file = paste0("./data/test/", folders.website[i,],"_emails.csv"))
}

recommended.emails[[1]]
```

```
## # A tibble: 10 x 2
##    emails                                      n
##    <fct>                                   <int>
##  1 andreas.fertin@ffg.at                      30
##  2 callsecretariat@electricmobilityeurope.eu  30
##  3 christian.drakulic@bmvit.gv.at             30
##  4 marcia.giacomini@de.tuv.com                30
##  5 peter.wilbers@rws.nl                       30
##  6 <NA>                                       NA
##  7 <NA>                                       NA
##  8 <NA>                                       NA
##  9 <NA>                                       NA
## 10 <NA>                                       NA
```

```
#alternative regex: [[:alnum:].-_]+@[[:alnum:].-]+.[a-z]{2,4}
```

Batch run

Now do this on a big scale: First loop through all files and get email addresses. Then create smart database that returns most probable email address. Note: The CSV simply lists website URLs of the organizations. (Will not run this since it takes hours)

```
# Get all the data!
websites <- read.csv("./data/websites.csv", stringsAsFactors = F)
websites <- websites[59:100,]

for (i in websites$Website){
  tryCatch({
    Rcrawler(Website =i, no_cores = 4, no_conn = 4, MaxDepth = 1, DIR = "./data/scraped")
}, error = function(e){cat("ERROR: ", conditionMessage(e), "\n")})
}

#now extract addresses
folders.website <- data.frame(list.files("./data/scraped"), stringsAsFactors = F)
#folders.website <- as.data.frame(folders.website[2:3,])
results <- list(list())
recommended.emails <- list()

for(i in 1:nrow(folders.website)){
  tryCatch({
  print(paste("Next Website:", folders.website[i,]))
  files.html <- data.frame(list.files(paste0("./data/scraped/",folders.website[i,])))
  print(paste(nrow(files.html), "files in folder"))
  URL <- as.character(folders.website[i,])
  for (l in 1:nrow(files.html)){
    print(paste(folders.website[i,],": Searching in", files.html[l,]))
    rawHTML <- paste(readLines(paste0("./data/scraped/", folders.website[i,],"/", files.html[l,])), col
    results[[URL]][l] <- as.data.frame(str_extract_all(rawHTML, "[a-zA-Z][a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+
  }
  temporary.stash <- as.data.frame(unlist(results[[URL]]))
  names(temporary.stash)[1] <- "emails"
  most_probable <- temporary.stash%>%
    count(emails)%>%
    top_n(3,n)
  most_probable <- most_probable[1:5,]
  recommended.emails[[URL]] <- most_probable[order(most_probable$n, decreasing = T),]
  print("Generating CSV")
  write.csv(temporary.stash, file = paste0("./data/scraped/", folders.website[i,],"_emails.csv"))
  print("Done")
  }, error = function(e){cat("ERROR: ", conditionMessage(e), "\n")}
  , warning = function(e){cat("Warning: ", conditionMessage(e), "\n")})
}

save(recommended.emails, file = "./data/scraped/01_mostlikelyemails.RData")
save(results, file = "./data/scraped/02_allemails.RData")
```

Deprecated things

```
#Html cleaner (messed with white spaces, because it deleted them)
#write html cleaner function
cleanHTML <- function(x){
  x <- paste(x, collapse="\n")
  x <- gsub("<.*?>", " ", x)
  x <- gsub("\n", " ", x)
  x <- gsub("\t", " ", x)
  x <- gsub("@import", " ", x)
```

```
}
rawHTML <- cleanHTML(x = rawHTML)
view <- as.data.frame(rawHTML)
```