Package 'Rcrawler'

November 1, 2017

Type Package

Title Web Crawler and Scraper

Version 0.1.7-0

Date 2017-11-1

Description Performs parallel web crawling and web scraping. It is designed to crawl, parse and store web pages to produce data that can be directly used for analysis application. For details see Khalil and Fakir (2017) <DOI:10.1016/j.softx.2017.04.004>.

License GPL (>= 2)

URL https://github.com/salimk/Rcrawler/

BugReports https://github.com/salimk/Rcrawler/issues

LazyData TRUE

Imports httr, xml2, data.table, foreach, doParallel, parallel, selectr

RoxygenNote 6.0.1

NeedsCompilation no

Author Salim Khalil [aut, cre]

Maintainer Salim Khalil <khalilsalim1@gmail.com>

Repository CRAN

Date/Publication 2017-11-01 04:13:08 UTC

R topics documented:

`ContentScraper` *ContentScraper*

Description

ContentScraper

Usage

```
ContentScraper(Url, HTmlText, XpathPatterns, CssPatterns, PatternsName,
  ExcludeXpathPat, ExcludeCSSPat, ManyPerPattern = FALSE, astext = TRUE,
  encod)
```

Arguments

`Url`	character, one url or a vector of urls of web pages to scrape.
`HTmlText`	character, web page as HTML text to be scraped.use either Url or HtmlText not both.
`XpathPatterns`	character vector, one or more XPath patterns to extract from the web page.
`CssPatterns`	character vector, one or more CSS selector patterns to extract from the web page.
`PatternsName`	character vector, given names for each xpath pattern to extract, just as an indication .
`ExcludeXpathPat`	character vector, one or more Xpath pattern to exclude from extracted content (like excluding quotes from forum replies or excluding middle ads from Blog post) .
`ExcludeCSSPat`	character vector, one or more Css pattern to exclude from extracted content.
`ManyPerPattern`	boolean, If False only the first matched element by the pattern is extracted (like in Blogs one page has one article/post and one title). Otherwise if set to True all nodes matching the pattern are extracted (Like in galleries, listing or comments, one page has many elements with the same pattern)
`astext`	boolean, default is TRUE, HTML and PHP tags is stripped from the extracted piece.
`encod`	character, set the weppage character encoding.

Value

return a named list of extracted content

Author(s)

salim khalil

Examples

```
## Not run:

DATA<-ContentScraper(Url ="http://glofile.com/index.php/2017/06/08/sondage-quel-budget/",
CssPatterns = c(".entry-title",".published",".entry-content"), astext = TRUE)
#Extract title, publishing date and article from the web page using css selectors

txthml<-"<html><title>blah</title><div><p>I m the content</p></div></html>"
DATA<-ContentScraper(HTmlText = txthml ,XpathPatterns = "//*/p")
#The web page source can be provided also as HTML text (characters)

DATA<-ContentScraper(Url ="http://glofile.com/index.php/2017/06/08/athletisme-m-a-rome/",
XpathPatterns=c("//head/title","//*/article"),PatternsName=c("title", "article"))
#Extract the title and the article from the web page using Xpath patterns,
#Patterns Name are provided as an indication.
urllist<-c("http://glofile.com/index.php/2017/06/08/sondage-quel-budget/",
"http://glofile.com/index.php/2017/06/08/cyril-hanouna-tire-a-boulets-rouges-sur-le-csa/",
"http://glofile.com/index.php/2017/06/08/placements-quelles-solutions-pour-doper/")

DATA<-ContentScraper(Url =urllist, CssPatterns = c(".entry-title",".entry-content"),
PatternsName = c("title","content"))
#Extract titles and contents of all 3 given Urls using CSS selectors, As result DATA variable
#will handle 6 elements.

DATA<-ContentScraper(Url =urllist, CssPatterns = c(".entry-title",".comment-content p"),
PatternsName = c("title","comments"), astext = TRUE, ManyPerPattern = TRUE)
#Extract titles and comments from a list of blog posts, ManyPerPattern argument enables extracting
#multiple similar elements from each page like comments,reviews, quotes and listing.

DATA<-ContentScraper(Url = "https://bitcointalk.org/index.php?topic=2334331.0",
CssPatterns = c(".post"),
ExcludeCSSPat = c(".quote",".quoteheader"),
PatternsName = c("posts"), ManyPerPattern = TRUE)
# From this Forum post Url we extract the post title and all replies using these CSS selectors
# c(".post"), However, we know that each reply contain the the previous Reply as quote so we exclude
# all quotes and quotes header from extracted posts using ExcludeCSSPat c(".quote",".quoteheader a")

## End(Not run)
```

Getencoding *Getencoding*

Description

This function retreives the encoding charset of web page based on HTML tags and HTTP header

Usage

```
Getencoding(url)
```

Arguments

url	character, the web page url.

Value

return the encoding charset as character

Author(s)

salim khalil

LinkExtractor *LinkExtractor*

Description

A function that take a _charachter_ url as input, fetches its html document, and extract all links following a set of rules.

Usage

```
LinkExtractor(url, id, lev, IndexErrPages, Useragent, Timeout = 6,
  URLlenlimit = 255, urlExtfilter, encod, urlbotfiler, removeparams,
  ExternalLInks = FALSE)
```

Arguments

url	character, url to fetch and extract links.
id	numeric, an id to identify a specific web page in a website collection, it's auto-generated by default
lev	numeric, the depth level of the web page, auto-generated by the Rcrawler function.
IndexErrPages	character vector, vector of html error code-statut to process, by default it's c(200),eg to include 404 and 403 pages c(404,403)
Useragent	, default to "Rcrawler"
Timeout	,default to 5s
URLlenlimit	interger, the url character length limit to index, default to 255 characters (to avoid spider traps)
urlExtfilter	character vector, the list of file extensions to exclude from indexing, by dfault a large list is defined (html pages only are permitted) in order to prevent large files downloading; To define your own use c(ext1,ext2,ext3 ...)
encod	character, specify the encoding of th web page
urlbotfiler	character vector , directories/files restricted by robot.txt
removeparams	character vector, list of url parameters to be removed/ignored
ExternalLInks	boolean, default FALSE, if set to TRUE external links also are returned.

Value

return a list of two elements, the first is a list containing the web page details (url, encoding-type, content-type, content ... etc), the second is a character-vector containing the list of retreived urls.

Author(s)

salim khalil

Examples

```
pageinfo<-LinkExtractor(url="http://www.glofile.com", ExternalLInks = TRUE)
#Pageinfo handle page header detail, as well as content, and internal links.
#pageinfo[[1]][[10]] : page content
#pageinfo[[2]] : Internal hyperlinks
#pageinfo[[3]] : External hyperlinks
```

LinkNormalization *Link Normalization*

Description

A function that take a URL _charachter_ as input, and transforms it into a canonical form.

Usage

```
LinkNormalization(links, current)
```

Arguments

links	character, the URL to Normalize.
current	character, The URL of the current page source of the link.

Details

This funcion call an external java class

Value

return the simhash as a nmeric value

Author(s)

salim khalil

Examples

```
# Normalize a set of links

links<-c("http://www.twitter.com/share?url=http://glofile.com/page.html",
         "/finance/banks/page-2017.html",
         "./section/subscription.php",
         "//section/",
         "www.glofile.com/home/",
         "glofile.com/sport/foot/page.html",
         "sub.glofile.com/index.php",
         "http://glofile.com/page.html#1"
                  )

links<-LinkNormalization(links,"http://glofile.com" )
```

`Linkparameters` *Get the list of parameters and values from an URL*

Description

A function that take a URL _charachter_ as input, and extract the parameters and values from this URL .

Usage

```
Linkparameters(URL)
```

Arguments

`URL`	character, the URL to extract

Details

This function extract the link parameters and values (Up to 10 parameters)

Value

return the URL paremeters=values

Author(s)

salim khalil

Examples

```
Linkparameters("http://www.glogile.com/index.php?name=jake&age=23&template=2&filter=true")
# Extract all URL parameters with values as vector
```

`Linkparamsfilter` *Link parameters filter*

Description

This function remove a given set of parameters from a specific URL

Usage

```
Linkparamsfilter(URL, params)
```

Arguments

`URL`	character, the URL from which params and values have to be removed
`params`	character vector, List of url parameters to be removed

Details

This function exclude given parameters from the urls,

Value

return a URL wihtout given parameters

Author(s)

salim khalil

Examples

```
url<-"http://www.glogile.com/index.php?name=jake&age=23&tmp=2&ord=1"
url<-Linkparamsfilter(url,c("ord","tmp"))

#exclude filter and template parameters from URL.
```

ListProjects *ListProjects*

Description

List all crawling project in your R local directory, or in a custom directory

Usage

```
ListProjects(DIR)
```

Arguments

DIR	character By default it's your local R workspace, if you set a custom folder for your crawling project then user DIR param to access this folder.

Value

ListProjects, a character vector.

Author(s)

salim khalil

Examples

```
## Not run:
ListProjects()

## End(Not run)
```

LoadHTMLFiles *LoadHTMLFiles @rdname LoadHTMLFiles*

Description

LoadHTMLFiles @rdname LoadHTMLFiles

Usage

```
LoadHTMLFiles(ProjectName, type = "vector", max)
```

Arguments

ProjectName	character, the name of the folder holding collected HTML files, use ListProjects fnuction to see all projects.
type	character, the type of returned variable, either vector or list.
max	Integer, maximum number of files to load.

Value

LoadHTMLFiles, a character vector or a list;

Author(s)

salim khalil

Examples

```
## Not run:
ListProjects()
#show all crawling project folders stored in your local R wokspace folder
DataHTML<-LoadHTMLFiles("glofile.com-301010")
#Load all HTML files in DataHTML vector
DataHTML2<-LoadHTMLFiles("glofile.com-301010",max = 10, type = "list")
#Load only 10 first HTMl files in DataHTML2 list

## End(Not run)
```

Rcrawler *Rcrawler*

Description

The crawler's main function, by providing only the website URL and the Xpath or CSS selector patterns this function can crawl the whole website (traverse all web pages) download webpages, and scrape/extract its contents in an automated manner to produce a structured dataset. The process of a crawling operation is performed by several concurrent processes or nodes in parallel, so it's recommended to use 64bit version of R.

Usage

```
Rcrawler(Website, no_cores, no_conn, MaxDepth, DIR, RequestsDelay = 0,
  Obeyrobots = FALSE, Useragent, Encod, Timeout = 5, URLlenlimit = 255,
  urlExtfilter, urlregexfilter, ignoreUrlParams, KeywordsFilter,
  KeywordsAccuracy, FUNPageFilter, ExtractXpathPat, ExtractCSSPat,
  PatternsNames, ExcludeXpathPat, ExcludeCSSPat, ExtractAsText = TRUE,
  ManyPerPattern = FALSE, NetworkData = FALSE, NetwExtLinks = FALSE,
  statslinks = FALSE)
```

Arguments

Website	character, the root URL of the website to crawl and scrape.
no_cores	integer, specify the number of clusters (logical cpu) for parallel crawling, by default it's the numbers of available cores.
no_conn	integer, it's the number of concurrent connections per one core, by default it takes the same value of no_cores.

MaxDepth	integer, repsents the max deph level for the crawler, this is not the file depth in a directory structure, but 1+ number of links between this document and root document, default to 10.
DIR	character, correspond to the path of the local repository where all crawled data will be stored ex, "C:/collection" , by default R working directory.
RequestsDelay	integer, The time interval between each round of parallel http requests, in seconds used to avoid overload the website server. default to 0.
Obeyrobots	boolean, if TRUE, the crawler will parse the website\'s robots.txt file and obey its rules allowed and disallowed directories.
Useragent	character, the User-Agent HTTP header that is supplied with any HTTP requests made by this function.it is important to simulate different browser's user-agent to continue crawling without getting banned.
Encod	character, set the website caharacter encoding, by default the crawler will automatically detect the website defined character encoding.
Timeout	integer, the maximum request time, the number of seconds to wait for a response until giving up, in order to prevent wasting time waiting for responses from slow servers or huge pages, default to 5 sec.
URLlenlimit	integer, the maximum URL length limit to crawl, to avoid spider traps; default to 255.
urlExtfilter	character's vector, by default the crawler avoid irrelevant files for data scraping such us xml,js,css,pdf,zip ...etc, it's not recommanded to change the default value until you can provide all the list of filetypes to be escaped.
urlregexfilter	character's vector, filter crawled Urls by regular expression pattern, this is useful when you try to scrape content or index only specific web pages (product pages, post pages).
ignoreUrlParams	character's vector, the list of Url paremeter to be ignored during crawling .
KeywordsFilter	character vector, For users who desires to scrape or collect only web pages that contains some keywords one or more. Rcrawler calculate an accuracy score based of the number of founded keywords. This parameter must be a vector with at least one keyword like c("mykeyword").
KeywordsAccuracy	integer value range bewteen 0 and 100, used only with KeywordsFilter parameter to determine the accuracy of web pages to collect. The web page Accuracy value is calculated using the number of matched keywords and their occurence.
FUNPageFilter	function, filter out pages to collect by your custom function (prediction, calssification model). This function should take two arguments URL and Content then test the content arg if is eligible or not following your rules then finally it must returns TRUE or FALSE.
ExtractXpathPat	character's vector, vector of xpath patterns to match for data extraction process.
ExtractCSSPat	character's vector, vector of CSS selector pattern to match for data extraction process.
PatternsNames	character vector, given names for each xpath pattern to extract.

`ExcludeXpathPat` character's vector, one or more Xpath pattern to exclude from extracted content ExtractCSSPat or ExtractXpathPat (like excluding quotes from forum replies or excluding middle ads from Blog post) .

`ExcludeCSSPat` character's vector, similar to ExcludeXpathPat but using Css selectors.

`ExtractAsText` boolean, default is TRUE, HTML and PHP tags is stripped from the extracted piece.

`ManyPerPattern` boolean, ManyPerPattern boolean, If False only the first matched element by the pattern is extracted (like in Blogs one page has one article/post and one title). Otherwise if set to True all nodes matching the pattern are extracted (Like in galleries, listing or comments, one page has many elements with the same pattern)

`NetworkData` boolean, If set to TRUE, then the crawler map all the internal hyperlink connections within the given website and return DATA for Network construction using igraph or other tools.(two global variables is returned see details)

`NetwExtLinks` boolean, If TRUE external hyperlinks (outlinks) also will be counted on Network edges and nodes.

`statslinks` boolean, if TRUE, the crawler counts the number of input and output links of each crawled web page.

Details

To start Rcrawler task you need the provide the root URL of the website you want to scrape, it can be a domain, a subdomain or a website section (eg. http://www.domain.com, http://sub.domain.com or http://www.domain.com/section/). The crawler then will go through all its internal links. The process of a crawling is performed by several concurrent processes or nodes in parallel, So, It is recommended to use R 64-bit version.

For more tutorials check https://github.com/salimk/Rcrawler/

For scraping complexe character content such as arabic execute Sys.setlocale("LC_CTYPE","Arabic_Saudi Arabia.1256") then set the encoding of the web page in Rcrawler function.

If you want to learn more about web scraper/crawler architecture, functional properties and implementation using R language, Follow this link and download the published paper for free .

Link: http://www.sciencedirect.com/science/article/pii/S2352711017300110

Dont forget to cite Rcrawler paper:

Khalil, S., & Fakir, M. (2017). RCrawler: An R package for parallel web crawling and scraping. SoftwareX, 6, 98-106.

Value

The crawling and scraping process may take a long time to finish, therefore, to avoid data loss in the case that a function crashes or stopped in the middle of action, some important data are exported at every iteration to R global environement:

- INDEX: A data frame in global environement representing the generic URL index,including the list of fetched URLs and page details (contenttype,HTTP state, number of out-links and in-links, encoding type, and level).

- A repository in workspace that contains all downloaded pages (.html files)

If data scraping is enabled by setting ExtractXpathPat or ExtractCSSPat parameter:

- DATA: A list of lists in global environement holding scraped contents.

- A csv file 'extracted_contents.csv' holding all extracted data.

If NetworkData is set to TRUE two additional global variables returned by the function are:

- NetwIndex : Vector maps alls hyperlinks (nodes) with a unique integer ID

- NetwEdges : data.frame representing edges of the network, with these column : From, To, Weight (the Depth level where the link connection has been discovered) and Type (1 for internal hyperlinks 2 for external hyperlinks).

Author(s)

salim khalil

Examples

```
## Not run:
 Rcrawler(Website ="http://glofile.com/", no_cores = 4, no_conn = 4)

 #Crawl, index, and store web pages using 4 cores and 4 parallel requests

 Rcrawler(Website = "http://glofile.com/", urlregexfilter =  "/[0-9]{4}/[0-9]{2}/",
 ExtractXpathPat = c("//*/article","//*/h1"), PatternsNames = c("content","title"))

  # Crawl the website using the default configuration and scrape content matching two XPath
  patterns only from post pages matching a specific regular expression "/[0-9]{4}/[0-9]{2}/".
  # Note that the user can use the excludepattern  parameter to exclude a node from being extracted,
  # e.g., in the case that a desired node includes (is a parent of) an undesired "child" node.

  Rcrawler(Website = "http://www.example.com/", no_cores=8, no_conn=8, Obeyrobots = TRUE,
  Useragent="Mozilla 3.11")
  # Crawl and index the website using 8 cores and 8 parallel requests with respect to
  # robot.txt rules.
  Rcrawler(Website = "http://www.example.com/", no_cores = 4, no_conn = 4,
  urlregexfilter =  "/[0-9]{4}/[0-9]{2}/", DIR = "./myrepo", MaxDepth=3)

  # Crawl the website using  4 cores and 4 parallel requests. However, this will only
 # index URLs matching the regular expression pattern (/[0-9]{4}/[0-9]{2}/), and stores pages
  # in a custom directory "myrepo".
  # The crawler stops After reaching the third level of website depth.

 Rcrawler(Website = "http://www.example.com/", KeywordsFilter = c("keyword1", "keyword2"))
  # Crawl the website and collect only webpages containing keyword1 or keyword2 or both.

 Rcrawler(Website = "http://www.example.com/", KeywordsFilter = c("keyword1", "keyword2"),
   KeywordsAccuracy = 50)
 # Crawl the website and collect only webpages that has an accuracy percentage higher than 50%
  # of matching keyword1 and keyword2.
```

```
Rcrawler(Website = "http://glofile.com/" , no_cores = 4, no_conn = 4, NetworkData = TRUE)
 # Crawl the entire website, and create network edges DATA of internal links.
 # Using Igraph for exmaple you can plot the network by the following commands
  library(igraph)
  network<-graph.data.frame(NetwEdges, directed=T)
  plot(network)

Rcrawler(Website = "http://glofile.com/" , no_cores = 4, no_conn = 4, NetworkData = TRUE)
# Crawl the entire website, and create network edges DATA of internal and external links .
 #FUNPageFilter parameter usage
# So first you create a function in your computer, it must take two arguments url, content
 # then returns true or false.
 # Inside the function you should test content variable using your rules
# Finally, you just call it inside Rcrawler function, Then the crawler will evaluate each
  page using your function.
  Rcrawler(Website = "http://glofile.com", no_cores=2, FUNPageFilter= Mytestfunction )


## End(Not run)
```

RobotParser *RobotParser fetch and parse robots.txt*

Description

This function fetch and parse robots.txt file of the website which is specified in the first argument and return the list of corresponding rules .

Usage

```
RobotParser(website, useragent)
```

Arguments

website	character, url of the website which rules have to be extracted .
useragent	character, the useragent of the crawler

Value

return a list of three elements, the first is a character vector of Disallowed directories, the third is a Boolean value which is TRUE if the user agent of the crawler is blocked.

Examples

```
RobotParser("http://www.glofile.com","AgentX")
#Return robot.txt rules and check whether AgentX is blocked or not.
```

Index