Package 'Rcrawler'

June 16, 2017

Type Package

Title Web Crawler and Scraper

Version 0.1

Date 2017-6-5

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 MIT + file LICENSE

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

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

LazyData TRUE

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

RoxygenNote 6.0.1

R topics documented:

ContentScraper *ContentScraper*

Description

From a given web page as text _character_ and a set of named XPath patterns, this function extracts selected parts of the HTML document then it returns a list of extracted contents.

Usage

```
ContentScraper(webpage, patterns, patnames, excludepat, astext = TRUE, encod)
```

Arguments

webpage	character, a web page as text.
patterns	character vector, one or more XPath patterns to extract from the web page.
patnames	character vector, given names for each xpath pattern to extract.
excludepat	character vector, one o more Xpath to exclude from the extracted content.
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

```
pageinfo<-LinkExtractor("http://glofile.com/index.php/2017/06/08/athletisme-m-a-rome/")
#Retreive the webpge header and data

Data<-ContentScraper(pageinfo[[1]][[10]],c("//head/title","//*/article"),c("title", "article"))
#Extract the title and the article from webpage content using Xpaths
```

getDistance *Calculate Distance between two SimHash fingerprint*

Description

A function that calculate the distance between two given fingerprint a _charachter_, distance is equal to 0 means the two strings are similar 100

Usage

```
getDistance(s1, s2)
```

Arguments

s1	character, the first fingerprint
s2	character, the second fingerprint

Value

return the distance as a nmeric value

Author(s)

salim khalil

Examples

```
text1<-"R is a free software environment for statistical computing and graphics.
        It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS"
text2<-"R is a language and environment for statistical computing and graphics.
       It is a GNU project which is similar to the S language and at Bell Laboratories"
text3<-" Astronomy is the scientific study of all objects beyond our world,
          and a way to understand the physical laws and origins of the universe."

dist1<-getDistance(getsimHash(text1,64),getsimHash(text2,64))
#dist1 is equal to 7, means the two strings are near-duplicate.

dist2<-getDistance(getsimHash(text1,64),getsimHash(text3,64))
#dist2 is equal to 21, means the two strings are not similar.
```

`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

`getsimHash` *Calculate SimHash fingerprint in R*

Description

A function that take a _charachter_ as input, and generate it's simhash.

Usage

```
getsimHash(string, hashbits)
```

Arguments

`string`	character, the content to hash.
`hashbits`	numeric, specify the hash bits 64 or 128

Details

This funcion call an external java class

Value

return the simhash as a nmeric value

Author(s)

salim khalil

Examples

```
text<-"R is a free software environment for statistical computing and graphics.
      It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS"
fingerprint<-getsimHash(text,64)
```

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 = 5,
  URLlenlimit = 255, urlExtfilter, statslinks = FALSE, encod, urlbotfiler,
  removeparams)
```

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 ...)
`statslinks`	boolean, specifies if input and output links shoud be counted, work only when the function is called from the main function `scrawler`
`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

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")
```

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("/finance/banks/page-2017.html",
         "./section/subscription.php",
         "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.
```

`Rcrawler` *Rcrawler*

Description

The crawler's main function, by providing only the website URL and the Xpath patterns to extract this function can crawl the whole website (traverse web pages and collect links) 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,
  duplicatedetect = FALSE, Obeyrobots = FALSE, Useragent, Timeout = 5,
  URLlenlimit = 255, urlExtfilter, urlregexfilter, ignoreUrlParams,
  statslinks = FALSE, Encod, ExtractPatterns, PatternsNames, ExcludePatterns,
  ExtractAsText = TRUE)
```

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.

`duplicatedetect` boolean, if true the crawler performs a near duplicate detection using SimHash algorithm to ignore documents that has been scraped.

`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.

`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 .

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

`Encod` character, set the website caharacter encoding, by default the crawler will automatically detect the website defined character encoding.

`ExtractPatterns` character's vector, vector of xpath patterns to use for data extraction process.

`PatternsNames` character vector, given names for each xpath pattern to extract.

`ExcludePatterns` character's vector, vector of xpath patterns to exclude from selected ExtractPatterns.

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

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 complexe charcter 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

Don't 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)

In addition, if data scraping is enabled :

- DATA: A vector in global environement contains scraped contents.

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

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}/",
 ExtractPatterns = 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 when it reaches the third level.

## 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