Package 'ChocoLattes'

April 29, 2017

Type Package

Title Processing Data from Lattes CV Files

Date 2017-4-28

Version 0.1.0

Imports XML, tools, ggplot2, plotly, rmarkdown, knitr, WriteXLS, utils, grDevices, R.utils

Depends R (>= 3.3.1)

Author Felipe Campelo

Maintainer Felipe Campelo <fcampelo@ufmg.br>

Description Processes data from Lattes CV (<http://lattes.cnpq.br/>) XML files. Extract, condition, and plot lists of journal and conference papers, book chapters, books, and more.

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-04-29 14:44:28 UTC

R topics documented:

`capitalize_authors` *Capitalize author names*

Description

Performs capitalization of author names

Usage

```
capitalize_authors(x, make.lower = c("De", "Di", "Do", "Da", "Dos", "Das",
  "Dello", "Della", "Dalla", "Dal", "Del", "E", "Em", "Na", "No", "Nas", "Nos",
  "Van", "Von", "Y"), author.aliases = NULL)
```

Arguments

`x`	data frame containing a column "Authors" (or "Author"), which should describe author names (separated by commas, if multiple) for a given production type.
`make.lower`	character vector containing name conectors to keep in lowercase
`author.aliases`	list vector with author aliases. See `lattes_to_list()` for details.

Details

This function attempts to standardize capitalization of author names for the production listings

Value

data frame with (hopefully) correctly capitalized author names in column "Authors".

capitalize_titles *Capitalize titles*

Description

Performs capitalization of titles

Usage

```
capitalize_titles(x, which.fields = c("Title", "Bookname", "Journal",
  "Conference"))
```

Arguments

x	dataframe containing fields to be capitalized (using sentence case)
which.fields	character vector with names of dataframe columns to be capitalized

Details

This function attempts to standardize capitalization of titles for the productions listing, using sentence case.

Value

capitalized string

extract_qualis *Extract QUALIS information*

Description

Summarize production as classified by a given QUALIS extract

Usage

```
extract_qualis(lattes.list, years, qualis.file, isComputerScience = FALSE,
  output.file = c("csv", "xlsx"), plotQualis = TRUE,
  qualis.extract = NULL, plot.width = 16, plot.height = 8,
  plot.units = "in", plot.res = 300, plot.text.size = 18)
```

Arguments

`lattes.list`	a Lattes list object created using `lattes_to_list()`
`years`	integer vector with years to be extracted from lattes.list
`qualis.file`	CSV file containing the ranking of journals according to the QUALIS classification system for a certain area. See `Qualis File` for details.
`isComputerScience`	logical flag to indicate if the relevant Qualis file is related to the "Computer Science" area (in which case conferences also receive QUALIS ranks)
`output.file`	type of file to output (xlsx or csv).
`plotQualis`	logical flag, should a plot be generated?
`qualis.extract`	name of the qualis extract used (optional, for the plot only)
`plot.width, plot.height, plot.units, plot.res, plot.text.size`	graphical parameters

Value

This routine returns nothing. It generates one **.xlsx** or **csv** file for each year in `years`, as well as a summary plot per year (as a png file), if `plotQualis = TRUE`

Qualis File

The qualis file must be a CSV file with commas as separators and UTF-8 encoding. To ensure these properties, follow the steps:

- Generate the relevant file from https://sucupira.capes.gov.br/sucupira/public/consultas/coleta/veiculoPublicacaoQualis/listaConsultaGeralPeriodicos.jsf, selecting only the fields *Evento de Classificacao* and *Area de Avaliacao* (leave the others blank).
- Download the resulting **xls** file.
- Open the file using your reader of choice, select all and copy
- Open a new spreadsheet on Google Docs, https://docs.google.com/spreadsheets/u/0/
- Paste everything into the Google Docs Spreadsheet
- Download the resulting file as a comma-separated file (**File -> Download as -> Comma-separated values (.csv, current sheet)**)

`get_accepted_papers` *Extract accepted journal papers*

Description

Extracts accepted journal papers from Lattes list.

Usage

```
get_accepted_papers(x, ID = stats::runif(1))
```

Arguments

x	Lattes list (generated internally in lattes_to_list())
ID	a unique identifier for each CV being processed.

Details

This function extracts relevant information on accepted journal papers from a Lattes list.

Value

data frame containing parsed information on accepted journal papers

---

get_advised_dissertations

Extract advised MSc dissertations

Description

Extracts advised MSc dissertations from Lattes list.

Usage

```
get_advised_dissertations(x, ID = stats::runif(1))
```

Arguments

x	Lattes list (generated internally in lattes_to_list())
ID	a unique identifier for each CV being processed.

Details

This function extracts relevant information on advised MSc dissertations from a Lattes list.

Value

data frame containing parsed information on advised MSc dissertations

get_advised_theses *Extract advised PhD theses*

Description

Extracts advised PhD theses from Lattes list.

Usage

```
get_advised_theses(x, ID = stats::runif(1))
```

Arguments

x	Lattes list (generated internally in lattes_to_list())
ID	a unique identifier for each CV being processed.

Details

This function extracts relevant information on advised PhD theses from a Lattes list.

Value

data frame containing parsed information on advised PhD theses

get_authors *Extract author names*

Description

Extracts author names from items

Usage

```
get_authors(item)
```

Arguments

item	a single bibliographic item, selected from a lattes list object.

Details

This function extracts author names from bibliographic items.

Value

character string containing author names in "name surname" format, separated by commas.

get_books *Extract published books*

Description

Extracts published books from Lattes list

Usage

```
get_books(x, ID = stats::runif(1))
```

Arguments

x	Lattes list (generated internally in lattes_to_list())
ID	a unique identifier for each CV being processed.

Details

This function extracts relevant information on published books from a Lattes list

Value

data frame containing parsed information on published books

get_book_chapters *Extract published book chapters*

Description

Extracts published book chapters from Lattes list.

Usage

```
get_book_chapters(x, ID = stats::runif(1))
```

Arguments

x	Lattes list (generated internally in lattes_to_list())
ID	a unique identifier for each CV being processed.

Details

This function extracts relevant information on published book chapters from a Lattes list.

Value

data frame containing parsed information on published book chapters

get_conference_papers *Extract published conference papers*

Description

Extracts published conference papers from Lattes list.

Usage

```
get_conference_papers(x, ID = stats::runif(1))
```

Arguments

x	Lattes list (generated internally in lattes_to_list())
ID	a unique identifier for each CV being processed.

Details

This function extracts relevant information on published conference papers from a Lattes list.

Value

data frame containing parsed information on conference papers

get_journal_papers *Extract published journal papers*

Description

Extracts published journal papers from Lattes list.

Usage

```
get_journal_papers(x, ID = stats::runif(1))
```

Arguments

x	Lattes list (generated internally in lattes_to_list())
ID	a unique identifier for each CV being processed.

Details

This function extracts relevant information on published journal papers from a Lattes list.

Value

data frame containing parsed information on journal papers

`lattes_to_list` *Convert a set of Lattes CV XML files to a list object*

Description

Extract information from a set of Lattes XML files and convert it to a list vector

Usage

```
lattes_to_list(CV.dir = NULL, author.aliases = list())
```

Arguments

`CV.dir`	folder where CVs are contained. If NULL then the current working directory is used.
`author.aliases`	list vector with author aliases. See `Examples` for details.

Details

This function extracts relevant information from a set of Lattes CV XML files and outputs a list object containing specific information on the following aspects of a group's production:

- Accepted journal papers
- Published journal papers
- Published conference papers
- Published book chapters
- Published books
- Ph.D. student defenses
- M.Sc. student defenses

Journal and conference papers are checked for duplication using DOI and Title information. Duplicated entries are registered only once.

Value

list vector where each element is a dataframe with information on a specific aspect of the academic production

Examples

```
my.dir <- system.file("extdata", package="ChocoLattes")

# Define the aliases of authors "Felipe Campelo" and "Lucas S. Batista":
# (all aliases will be converted to the first name provided for each author)
my.aliases <- list(c("Felipe Campelo",
                     "Felipe Campelo Franca Pinto",
                     "Felipe Campelo F. Pinto",
```

```
                      "F.C.F. Pinto"),
                    c("Lucas S. Batista",
                      "Lucas Batista",
                      "Lucas de Souza Batista",
                      "Lucas Souza Batista"))

lattes.list <- lattes_to_list(CV.dir        = my.dir,
                              author.aliases = my.aliases)
```

`make_name_surname` *Put author names in "Name Surname" format*

Description

Put author names in "Name Surname" format

Usage

```
make_name_surname(x)
```

Arguments

x	string containing author name, either in "Surname, Name" or "Name Surname" format

Value

string containing author name in "Name Surname" format

`make_productions_page` *Generate HTML page with productions list and summary chart*

Description

This function generates an HTML page with productions list and summary chart. Use chart.type = "plotly" for an interactive plot, and "ggplot2" for a static one. Option "rCharts" is currently disabled, until the rCharts package (https://github.com/ramnathv/rCharts) becomes available on CRAN.

Usage

```
make_productions_page(lattes.list, chart.type = c("ggplot2", "plotly",
  "rCharts"), chart.width = 960, chart.height = 480,
  h1.title = "My Laboratory<br/>My Department<br/>My University",
  h2.title = "Academic Productions", language = c("EN", "PT"),
  which.fields = c("journal.accepted", "journal.published",
  "conference.international", "conference.national", "book.chapters", "books",
  "phd.theses", "msc.theses"))
```

Arguments

lattes.list	a Lattes list object created using lattes_to_list()
chart.type	package to use for generating the summary chart. "plotly" and "rCharts" output interactive charts, "ggplot2" outputs a static one.
chart.width	plot width (for "plotly" and "rCharts")
chart.height	plot height (for "plotly")
h1.title	H1 title for the page
h2.title	H2 subtitle for the page
language	Language to use in section headers
which.fields	Character vector indicating which fields to include in the productions page.

Examples

```
# Load example data and convert it to a lattes list
CV.dir <- system.file("extdata", package="ChocoLattes")

# Define the aliases of authors "Felipe Campelo" and "Lucas S. Batista":
# (all aliases will be converted to the first name provided for each author)
my.aliases <- list(c("Felipe Campelo",
                     "Felipe Campelo Franca Pinto",
                     "Felipe Campelo F. Pinto",
                     "F.C.F. Pinto"),
                   c("Lucas S. Batista",
                     "Lucas Batista",
                     "Lucas de Souza Batista",
                     "Lucas Souza Batista"))

lattes.list <- lattes_to_list(CV.dir         = CV.dir,
                              author.aliases = my.aliases)


# Build productions page:
make_productions_page(lattes.list,
                      chart.type     = "ggplot2",
                      h1.title       = "ORCS Lab",
                      h2.title       = "UFMG, Brazil",
                      language       = "EN")
```

---

plot_chart *Plot summary chart*

Description

Plot summary chart from a Lattes list object

Usage

```
plot_chart(lattes.list, chart.type = c("ggplot2", "plotly", "rCharts"),
  width = 960, height = 480, language = c("EN", "PT"),
  which.fields = c("journal.accepted", "journal.published",
  "conference.international", "conference.national", "book.chapters", "books",
  "phd.theses", "msc.theses"))
```

Arguments

lattes.list	a Lattes list object created using lattes_to_list()
chart.type	package to use for generating the summary chart. "plotly" and "rCharts" output interactive charts, "ggplot2" outputs a static one. Option "rCharts" is currently disabled, until the rCharts package (https://github.com/ramnathv/rCharts) becomes available on CRAN.
width	plot width (for "plotly" and "rCharts")
height	plot height (for "plotly")
language	Language to use in section headers
which.fields	Character vector indicating which fields to include in the productions page.

Details

This function plots production information from a Lattes list object generated by lattes_to_list():

- Accepted journal papers
- Published journal papers
- Published conference papers
- Published book chapters
- Published books
- Ph.D. student defenses
- M.Sc. student defenses

Value

plot object for inclusion in a productions page (see make_productions_page().

---

print_accepted *Print accepted papers*

Description

Prints accepted papers

Usage

```
print_accepted(x, language = c("EN", "PT"))
```

Arguments

x	data frame containing information on accepted papers
language	Language to use in section headers

---

print_books *Print books*

Description

Prints published books

Usage

```
print_books(x, language = c("EN", "PT"))
```

Arguments

x	data frame containing information on published books
language	Language to use in section headers

---

print_book_chapters *Print book chapters*

Description

Prints published book chapters

Usage

```
print_book_chapters(x, language = c("EN", "PT"))
```

Arguments

x	data frame containing information on published book chapters
language	Language to use in section headers

print_conferences *Print conference papers*

Description

Prints published conference papers

Usage

```
print_conferences(x, isIntl = TRUE, language = c("EN", "PT"))
```

Arguments

x	data frame containing information on published conference papers
isIntl	logical flag, TRUE for international conferences, FALSE for national/regional
language	Language to use in section headers

print_journal_papers *Print journal papers*

Description

Prints published journal papers

Usage

```
print_journal_papers(x, language = c("EN", "PT"))
```

Arguments

x	data frame containing information on published papers
language	Language to use in section headers

print_msc_theses *Print MSc theses*

Description

Prints MSc theses defended

Usage

```
print_msc_theses(x, language = c("EN", "PT"))
```

Arguments

x	data frame containing information on published MSc theses
language	Language to use in section headers

print_phd_theses *Print PhD theses*

Description

Prints PhD theses defended

Usage

```
print_phd_theses(x, language = c("EN", "PT"))
```

Arguments

x	data frame containing information on published phd theses
language	Language to use in section headers

remove_duplicates *Function to remove duplicated entries*

Description

Cleans up duplicated items on a production data frame based on DOI, ISBN, and Title. This function also removes any rows where the value of "year" is greater than the current year (since these are considered errors)

Usage

```
remove_duplicates(x)
```

Arguments

x	data frame containing information on a certain type of production (journal papers, conference papers, etc.). Must contain columns "Title" and "Year". Optional (very useful) columns include "DOI" and "ISBN".

Value

data frame with duplicates removed

sort_papers *Sort publications by year*

Description

Sorts publications by year

Usage

```
sort_papers(x, decreasing = TRUE)
```

Arguments

x	data frame containing information on a certain type of production (journal papers, conference papers, etc.). Must contain column "Year". Optional (useful) columns include "DOI" and "ISBN".
decreasing	flag, sort in decreasing or increasing order

Value

ordered data frame

Index