

Package ‘crqanlp’

March 21, 2018

Type Package

Title Cross-Recurrence Quantification Analysis for Dynamic Natural Language Processing

Version 0.3

Date 2018-03-19

Author Rick Dale and Moreno Coco

Maintainer Rick Dale <racdale@gmail.com>

Description Cross-recurrence quantification analysis for word series, from text, known as categorical recurrence analysis. Uses the 'crqa' R package by Coco and Dale (2014) <doi:10.3389/fpsyg.2014.00510>. Functions are wrappers to facilitate exploration of the sequential properties of text.

Depends R (>= 3.0.0), crqa, tm, corpus, gutenbergr, RCurl

License GPL (>= 2)

NeedsCompilation no

RoxygenNote 6.0.1

Repository CRAN

Date/Publication 2018-03-21 10:01:11 UTC

R topics documented:

crqanlp-package	2
assign_codes	3
clean_text	4
get_text_series	5
plot_rp	6
text_rqa	7
text_win_rqa	8

Index	10
--------------	-----------

crqanlp-package

Dynamic Natural Language Processing with Recurrence Quantification Analysis

Description

crqanlp is a package for conducting dynamic analysis of text under recurrence quantification, which builds upon the 'crqa' package.

Details

Package: crqanlp
Type: Package
Version: 0.1
Date: 2017-03-05
License: GPL >= 2

assign_codes:
clean_text:
get_text_series:
plot_rp:
text_rqa:
text_win_rqa:

Author(s)

Rick Dale (rdale@ucla.edu) and Moreno I. Coco (moreno.cocoi@gmail.com)

References

Coco, M. I., & Dale, R. (2014). Cross-recurrence quantification analysis of categorical and continuous time series: an R package. *Frontiers in psychology*, 5, 510.

Marwan, N., and Kurths, J. Nonlinear analysis of bivariate data with cross recurrence plots. *Physics Letters A* 302.5 (2002): 299-307.

Orsucci, F., Giuliani, A., Webber Jr, C., Zbilut, J., Fonagy, P. and Mazza, M. (2006). Combinatorics and synchronization in natural semiotics. *Physica A: Statistical Mechanics and its Applications*, 361(2), 665-676.

Examples

```
txt = "here is a raw raw raw string, literally"
```

```
res = text_rqa(txt,typ = "string")
plot_rp(res$RP)
```

assign_codes

Assign codes

Description

It converts a sequence of words into a sequence of numerical identifier.

Usage

```
assign_codes(words, wordCode)
```

Arguments

words	Vector of strings (tokens)
wordCode	vector of strings (types)

Details

Imagine we want to treat all words from the semantic category of animals in the same way. This function would allow the user to assign to different words, e.g. cat, dog or bird, the same identifier, e.g., 1. Note, words and wordCode need to have the same length.

Value

It returns the vector of words but now converted to a numerical identifier, which corresponds to the associated wordCode.

Author(s)

Rick Dale (rdale@ucla.edu)

Examples

```
library(corpus)
words = c("bad", "bed", "joy", "sad") # words that we wish to locate in affect_wordnet
wordCode = affect_wordnet[,1] # all words in affect_wordnet
assign_codes(words, wordCode)
```

`clean_text`*Clean text*

Description

Pre-processing of raw text. It removes stop-words, punctuations, and create sentence markers.

Usage

```
clean_text(rawText,removeStopwords=F)
```

Arguments

`rawText` A Vector of strings (tokens)
`removeStopwords` A boolean: TRUE (remove stop words) - FALSE (it retains them)

Details

A convenience function that removes unwanted information from a vector of text. The user has, at the moment, an argument to choose whether to remove stop words.

Value

It returns the vector of text all in lower case, and stripped from punctuations and stop-words.

Author(s)

Rick Dale (rdale@ucla.edu)

Examples

```
library(gutenbergr)
## let's get Alice's Adventures in Wonderland by Carroll
# gutenbergr_works(author == "Carroll, Lewis")
rawText = gutenbergr_download(11) ## take the text
rawText = as.vector(rawText$text) ## vectorize the text
rawText = paste(rawText, collapse = " ") ## collapse the text

cleanText = clean_text(rawText, removeStopwords = TRUE)
text      = cleanText$content
```

get_text_series	<i>Get a text series</i>
-----------------	--------------------------

Description

Acquire (and process) texts from different resources

Usage

```
get_text_series(rsrc, typ='file',removeStopwords=F)
```

Arguments

rsrc	The resource text file
typ	A flag indicating the type of resource file in input: typ = "file" (it's a file name); typ = "ulr" (it's a url, and the file gets downloaded) = ; typ = "string raw_chars" (it's a literal string); typ = "tibble" (it's a text formatted as tidytext by tibble)
removeStopwords	A boolean: TRUE (remove stop words) - FALSE (it retains them)

Details

A convenience function to obtain text from different types of source. It can be used to access text from the internet, from a file, from a tibble text, or just from a literal string. The user can also choose whether to remove stop words (or not).

Value

It returns the vector of text already converted into a numerical identifier.

Author(s)

Rick Dale (rdale@ucla.edu)

Examples

```
## from a literal string
rsrc = "here is a raw raw raw string, literally"
ts = get_text_series(rsrc,typ='string')
print(ts)

## from tibble
library(gutenbergr)
## let's get Alice's Adventures in Wonderland by Carroll
# gutenbergr::gutenberg_works(author == "Carroll, Lewis")
rsrc = gutenbergr::gutenberg_download(11) ## take the text
ts = get_text_series(rsrc, typ = "tibble", removeStopwords = TRUE)
```

```
print(ts[1:10])

## from URL
rsrc = "http://www.omegahat.net"
ts = get_text_series(rsrc, typ = "url")
print(ts[1:10])
```

plot_rp

Plot a recurrence matrix

Description

Visualize a Recurrence Plot

Usage

```
plot_rp(RP, xlab='i', ylab='j', cex=.1)
```

Arguments

RP	A recurrence plot as produce by 'crqa()'
xlab	The name for the x axis
ylab	The name for the y axis
cex	The size of the recurrent point in the plot

Details

A function to plot recurrence matrices.

Value

A plot of a recurrence matrix

Author(s)

Rick Dale (rdale@ucla.edu)

Examples

```
txt = "here is a raw raw raw string, literally"
res = text_rqa(txt, typ = "string")
plot_rp(res$RP)
```

text_rqa	<i>Recurrence quantification analysis on categorical series of text</i>
----------	---

Description

Compute recurrence quantification on text.

Usage

```
text_rqa(rsrc, typ = 'file', removeStopwords = F, embed = 1, tw = 1, limit = -1, shuffle = F)
```

Arguments

rsrc	Location of file or resource, or string literal
typ	A flag indicating the type of resource file in input: typ = "file" (it's a file name); typ = "ulr" (it's a url, and the file gets downloaded); typ = "string" or "raw_chars" (it's a literal string); typ = "tibble" (it's a text formatted as tidytext by tibble)
removeStopwords	A boolean: TRUE (remove stop words) - FALSE (it retains them)
embed	The number of embedding dimension for phase-reconstruction, i.e., the lag intervals.
tw	The Theiler window parameter
limit	A scalar indicating how much text should be considered for the analysis
shuffle	A boolean: if TRUE, it randomly shuffles the order of the text for surrogate analyses.

Details

A wrapper to the 'crqa()' function that runs recurrence quantification analysis on text. This function also calls 'get_text_series()' to simplify the text in case such simplification was not done before inputting the text.

Value

It returns a list with different measures extracted from the recurrence plot. Otherwise, the values for the output arguments will be either 0 or NA.

RR	The percentage of recurrent points falling within the specified radius (range between 0 and 100)
DET	Proportion of recurrent points forming diagonal line structures.
NRLINE	The total number of lines in the recurrent plot
maxL	The length of the longest diagonal line segment in the plot, excluding the main diagonal
L	The average length of line structures

ENTR	Shannon information entropy of diagonal line lengths longer than the minimum length
rENTR	Entropy measure normalized by the number of lines observed in the plot. Handy to compare across contexts and conditions
LAM	Proportion of recurrent points forming vertical line structures
TT	The average length of vertical line structures

Author(s)

Rick Dale (rdale@ucla.edu)

Examples

```
txt = "here is a raw raw raw string, literally"
res = text_rqa(txt, typ = "string")
plot_rp(res$RP)
```

text_win_rqa

Window Recurrence quantification analysis ib text

Description

It computes recurrence of text in overlapping windows over a number of delays smaller than the size of the window.

Usage

```
text_win_rqa(rsrc, typ = 'file', winsz = 10, wshft = 10,
removeStopwords = F, embed = 1, tw = 1, limit = -1, shuffle = F)
```

Arguments

rsrc	Location of file or resource, or string literal
typ	A flag indicating the type of resource file in input: typ = "file" (it's a file name); typ = "url" (it's a url, and the file gets downloaded); typ = "string" or "raw_chars" (it's a literal string); typ = "tibble" (it's a text formatted as tidytext by tibble)
winsz	The size of the window
wshft	Interval by which the window is moved.
removeStopwords	A boolean: TRUE (remove stop words) - FALSE (it retains them)
embed	The number of embedding dimension for phase-reconstruction, i.e., the lag intervals.

tw	The Theiler window parameter
limit	A scalar indicating how much text should be considered
shuffle	A boolean if TRUE, it randomly shuffles the order of the text for surrogate analyses.

Details

A wrapper to the ‘wincrqa()’ function that runs windowed recurrence quantification analysis on text. This function also calls ‘get_text_series()’ to simplify the text in case such simplification was not done before inputting the text.

Value

It returns a matrix where the rows are the different windows explored, and the columns are the recurrence measures observed in that particular window. Refer to text_win_rqa for the values returned.

Author(s)

Rick Dale (rdale@ucla.edu)

Examples

```
library(gutenbergr)
## let's get Alice's Adventures in Wonderland by Carroll
# gutenbergr_works(author == "Carroll, Lewis")
rsrc = gutenbergr_download(11) ## take the text
win_rqa_res = text_win_rqa(rsrc, typ = "tibble", wshft = 150, winsz = 200,
                           removeStopwords = FALSE, embed = 1, tw = 1, limit = -1, shuffle = FALSE)
win_rqa_res = as.data.frame(win_rqa_res$crqwin)

plot(win_rqa_res$window, win_rqa_res$RR)
abline(lm(win_rqa_res$RR ~ win_rqa_res$window), lwd = 3, col = "red")
```

Index

*Topic **misc**

- assign_codes, 3
- clean_text, 4
- get_text_series, 5
- plot_rp, 6

*Topic **package**

- crqanlp-package, 2

*Topic **ts**

- text_rqa, 7
- text_win_rqa, 8

assign_codes, 3

clean_text, 4

crqanlp-package, 2

get_text_series, 5

plot_rp, 6

text_rqa, 7

text_win_rqa, 8