Package 'CorporaCoCo'

November 23, 2017

Encoding UTF-8

Type Package

Title Corpora Co-Occurrence Comparison

Version 1.1-0

Date 2017-11-22

Description A set of functions used to compare co-occurrence between two corpora.

URL

License GPL (>= 3)

Depends R (>= 3.2.0),

Imports methods, stats, data.table (>= 1.9.6), RColorBrewer, rlist

Suggests unittest, stringi, R.rsp

VignetteBuilder R.rsp

BugReports https://github.com/birmingham-ccr/CorporaCoCo/issues

LazyData yes

NeedsCompilation no

Author Anthony Hennessey [aut, cre],
Viola Wiegand [aut],
Michaela Mahlberg [aut],
Christopher R. Tench [aut],
Jamie Lentin [aut]

Maintainer Anthony Hennessey <anthony.hennessey@nottingham.ac.uk>

Repository CRAN

Date/Publication 2017-11-23 10:54:35 UTC

R topics documented:

`CorporaCoCo-package` *Comparing Co-occurrence between corpora.*

Description

Implements the method described in Hennessey and Wiegand et al. (2017).

Details

A good place to start is the 'Proof of Concept' vignette. There is also a 'FAQ' vignette. You can see a list of package vignettes with `vignette(package = "CorporaCoCo")` and you can see a particular vignette with something like `vignette("faq", package = "CorporaCoCo")`.

For a list of all documentation use `library(help="CorporaCoCo")`.

Author(s)

Maintainer: Anthony Hennessey <anthony.hennessey@nottingham.ac.uk>.

References

A. Hennessey and V. Wiegand and C. R. Tench and M. Mahlberg (2017) *Comparing co-occurrences between corpora.* In preparation.

`coco` *Co-occurrence comparison*

Description

Calculates statistically significant difference in co-occurrence counts.

Usage

```
coco(A, B, nodes, fdr = 0.01, collocates = NULL)
```

Arguments

`A`	A `data.frame` of co-occurrence counts. See details.
`B`	A `data.frame` of co-occurrence counts. See details.
`nodes`	A `character vector` of nodes or `character string` representing a single node.
`fdr`	The desired level at which to control the False Discovery Rate. Default value is `0.01`.

collocates — A character vector of collocates or character string representing a single collocate. The *collocates* essentially act as a filter on the *y* column of the returned data structure. collocates should be used to target the testing; reducing the number of tests will reduce the loss of power from the multiple test correction.

Details

This function implements the method described in Hennessey and Wiegand (2017).

A and *B* are data.frames of the form

```
Classes 'data.frame': ...
 $ x: chr
 $ y: chr
 $ H: int
 $ M: int
```

The data.frames encapsulate the co-occurrence counts for the (x, y) term pairs within a corpus. For a description of the columns see the details section of the surface function.

The *nodes* essentially act as a filter on the *A$x* and *B$x* columns. For a description of the use of nodes see Hennessey and Wiegand (2017).

fdr indicates the level at which the False Discovery Rate will be controlled. For a description of the form of FDR used see Benjamini and Hochberg (1995). For a description of the use of FDR in this context see Hennessey and Wiegand (2017). For description of the *p_adjusted* column in the returned structure see p.adjust.

The returned data structure is a data.table. A data.table is also a data.frame and will behave exactly as such if the data.table library is not loaded.

The returned data.table contains details of all the co-occurrences for which there is evidence of a difference in co-occurrence between the two supplied data sets. The effect size is calculated as the log base 2 of the odds ratio. The effects size and its confidence interval are captured in the *effect_size*, *CI_lower* and *CI_upper* columns. The *p_value* column contains the non-adjusted p-value from the Fisher's Exact Test. For more details see Hennessey and Wiegand (2017).

For an example of usage see the 'Proof of Concept' vignette.

Value

A data.table of the form

```
Classes 'data.table' and 'data.frame': 11 variables:
 $ x           : chr
 $ y           : chr
 $ H_A         : int
 $ M_A         : int
 $ H_B         : int
 $ M_B         : int
 $ effect_size : num
 $ CI_lower    : num
```

```
$ CI_upper    : num
$ p_value     : num
$ p_adjusted  : num
- attr(*, "sorted")= chr  "x" "y"
- attr(*, ".internal.selfref")=<externalptr>
- attr(*, "coco_metadata")=List of 4
 ..$ nodes : chr
 ..$ fdr       : num
 ..$ PACKAGE_VERSION:Classes 'package_version', 'numeric_version'
 .. ..$ : int
 ..$ date      : Date, format: "2016-11-01"
```

References

Y. Benjamini and Y. Hochberg (1995) *Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing.* Journal of the Royal Statistical Society. Series B (Methodological) **57 (1)289–300**.

A. Hennessey and V. Wiegand and C. R. Tench and M. Mahlberg (2017) *Comparing co-occurrences between corpora.* In preparation.

`coco-class` *coco class*

Description

Object of class coco.

Usage

```
  ## S3 method for class 'coco'
plot(x, as_matrix = FALSE, nodes = NULL, forest_plot_args = NULL, ...)
```

Arguments

`x`	An coco object.
`as_matrix`	If `as_matrix` is set to `TRUE` a matrix plot rather than a forest plot is produced.
`nodes`	If a vector of `nodes` is supplied this will be used to filter the set of results that are plotted. If `nodes` are supplied the plot will use the `nodes` order.
`forest_plot_args`	This is a list of arguments that is passed to the `plot.default` that produces the foundation of the forest plot. The list may contain a subset or all of the following documented arguments; any arguments that are not documented here will be ignored. A description of each argument can be found in the help for the `plot.default` function. Available arguments are • `xlim` Default: Calculated from the ranges of the confidence intervals. • `xlab` Default: 'Effect Size'

- `main` Default: NULL
- `sub` Default: NULL
- `asp` Default: NA
- `pch` Default: 15
- `cex.pch` Default: 1
- `lwd.xaxt` Default: 1
- `col.xaxt` Default: 'black'
- `col.whisker` Default: 'black'
- `col.zero` Default: 'darkgray'
- `length.wisker_end` Default: 0.05

For example usage see the 'plot' section in the 'FAQ' vignette.

`...` Other arguments will be ignored.

Details

An object of class `coco` is returned by `coco()` and `surface_coco()`. No constructor is exported.

Note

For example usage see the 'FAQ' vignette.

`surface` *Calculate Surface Co-occurrence Counts*

Description

Calculates co-occurrence counts for the supplied vector. For each co-occurrence the maximum possible number of co-occurrences is also calculated.

Usage

```
surface(x, span, nodes = NULL, collocates = NULL)
```

Arguments

`x`	A vector. This is the subject of the co-occurrence counting. See details.
`span`	A character string defining the co-occurrence span. See details.
`nodes`	A `character vector` of nodes or `character string` representing a single node. The *nodes* essentially act as a filter on the *x* column of the returned data structure. Use of `nodes` will significantly reduce memory usage.
`collocates`	A `character vector` of collocates or `character string` representing a single collocate. The *collocates* essentially act as a filter on the *y* column of the returned data structure.

Details

x is assumed to be an ordered vector of tokenized text. No processing will be applied to `x` prior to the co-occurrence count calculations.

'surface' co-occurrence is easiest to describe with an example. The following is a `span` of `'2LR'`, that is 2 to the left and 2 to the right.

```
("a", "man", "a", "plan", "a", "cat", "a", "canal", "panama")
      |___________|____|___________|
```

In this example the term "plan" would co-occur once each with the collocates "man" and "cat", and twice with the collocate "a".

Other examples of `span`:

```
span = '1L2R'

("a", "man", "a", "plan", "a", "cat", "a", "canal", "panama")
             |____|____|___________|

span = '2R'

("a", "man", "a", "plan", "a", "cat", "a", "canal", "panama")
                  |____|___________|
```

`NA`s can be used to implement co-occurrence barriers eg if two `NA` characters are inserted into *x* at each sentence boundary then with `span = 2` co-occurrences will not happen across sentences. See Evert (2008) for detailed description of co-occurrence barriers.

For a detailed description of 'surface' co-occurrence and the other types of co-occurrence see Evert (2008).

Value

Returns a `data.table` containing counts for all co-occurrences in *x*. Note that a `data.table` is also a `data.frame` so if the `data.table` library is not loaded the returned object will behave exactly as a `data.frame`; however, for large data sets there will be significant performance enhancement offered by exploiting `data.table` functionality.

The returned object is of the form:

```
Classes 'data.table' and 'data.frame': ...
 $ x: chr
 $ y: chr
 $ H: int
 $ M: int
 - attr(*, "sorted")= chr  "x" "y"
 - attr(*, ".internal.selfref")=<externalptr>
```

where `H` is the number of times `x` co-occurs with `y` (think *H*its), and `M` is the number of times `x` fails to co-occur with `y` when it could have (think *M*isses); hence `H + M` is the maximum number of times that `x` could have co-occurred with `y`.

References

S. Evert (2008) *Corpora and collocations.* Corpus Linguistics: An International Handbook **1212–1248**.

Examples

```
# =====================
# surface co-occurrence
# =====================

x <- c("a", "man", "a", "plan", "a", "canal", "panama")
surface(x, span = '2R')

##         x      y H M
##  1:     a      a 2 4
##  2:     a  canal 1 5
##  3:     a    man 1 5
##  4:     a panama 1 5
##  5:     a   plan 1 5
##  6: canal panama 1 0
##  7:   man      a 1 1
##  8:   man   plan 1 1
##  9:  plan      a 1 1
## 10:  plan  canal 1 1


# filter on nodes
surface(x, span = '2R', nodes = c("canal", "man", "plan"))

##         x      y H M
##  1: canal panama 1 0
##  2:   man      a 1 1
##  3:   man   plan 1 1
##  4:  plan      a 1 1
##  5:  plan  canal 1 1

# filter on nodes and collocates
surface(x, span = '2R', nodes = c("canal", "man", "plan"), collocates = c("panama", "a"))

##         x      y H M
##  1: canal panama 1 0
##  2:   man      a 1 1
##  3:  plan      a 1 1


# co-occurrence barrier
x <- c("a", "man", "a", "plan", NA, NA, "a", "canal", "panama")
surface(x, span = '2R')

#         x      y H M
#  1:     a      a 1 4
#  2:     a  canal 1 4
#  3:     a    man 1 4
```

```
#  4:     a panama 1 4
#  5:     a   plan 1 4
#  6: canal panama 1 0
#  7:   man      a 1 1
#  8:   man   plan 1 1
```

`surface_coco` *Surface co-occurrence comparison*

Description

Convenience function that combined the functionality of the `surface` and `coco` functions.

Usage

```
surface_coco(a, b, span, nodes, fdr = 0.01, collocates = NULL)
```

Arguments

`a`	A character vector.
`b`	A character vector.
`span`	A character string defining the co-occurrence span. See `surface` function for details.
`nodes`	A `character vector` of nodes or `character string` representing a single node.
`fdr`	The desired level at which to control the False Discovery Rate.
`collocates`	A `character vector` of collocates or `character string` representing a single collocate.

Details

See `surface` and `coco`.

For an example of usage see the 'Proof of Concept' vignette.

Value

A `data.table` of the form returned by the `coco` functions.

Index