Package 'rainette'

May 9, 2020

Type Package

Title The Reinert Method for Textual Data Clustering

Version 0.1.1

Date 2020-05-09

Maintainer Julien Barnier <julien.barnier@cnrs.fr>

Description An R implementation of the Reinert text clustering method. For more details about the algorithm see the included vignettes or Reinert (1990) <doi:10.1177/075910639002600103>.

License GPL (>= 3)

VignetteBuilder knitr

URL https://juba.github.io/rainette/

BugReports https://github.com/juba/rainette/issues

Encoding UTF-8

Imports dplyr (>= 0.8.3), tidyr, purrr, ggplot2, stringr, quanteda (>= 1.5), RSpectra, dendextend, ggwordcloud, gridExtra, rlang, RColorBrewer, shiny, miniUI, formatR, highr, future, future.apply, progressr, Rcpp (>= 1.0.3)

Suggests testthat, knitr, rmarkdown, covr, vdiffr, tm, FNN, quanteda.textmodels

RoxygenNote 7.1.0

LinkingTo Rcpp

NeedsCompilation yes

Author Julien Barnier [aut, cre],
Florian Privé [ctb]

Repository CRAN

Date/Publication 2020-05-09 12:00:03 UTC

R topics documented:

`cluster_tab` *Split a dtm into two clusters with reinert algorithm*

Description

Split a dtm into two clusters with reinert algorithm

Usage

```
cluster_tab(dtm, cc_test = 0.3, tsj = 3)
```

Arguments

dtm	to be split, passed by `rainette`
cc_test	maximum contingency coefficient value for the feature to be kept in both groups.
tsj	minimum feature frequency in the dtm

Details

Internal function, not to be used directly

Value

An object of class `hclust` and `rainette`

`compute_uc` *Merges uces into uc according to minimum uc size*

Description

`rainette_uc_index` docvar

Usage

```
compute_uc(dtm, min_uc_size = 10)
```

Arguments

`dtm`	dtm of uces, with a `rainette_uce_id` docvar
`min_uc_size`	minimum number of forms by uc

Details

Internal function, not to be used directly

Value

the original dtm with a new `rainette_uc_id` docvar.

`cutree` *Cut a tree into groups*

Description

Cut a tree into groups

Usage

```
cutree(tree, ...)
```

Arguments

`tree`	the hclust tree object to be cut
`...`	arguments passed to other methods

Details

If `tree` is of class `rainette`, invokes `cutree_rainette()`. Otherwise, just run `stats::cutree()`.

Value

A vector with group membership.

cutree_rainette *Cut a rainette result tree into groups of documents*

Description

Cut a rainette result tree into groups of documents

Usage

```
cutree_rainette(hres, k = NULL, h = NULL, ...)
```

Arguments

hres	the rainette result object to be cut
k	the desired number of groups
h	unsupported
...	arguments passed to other methods

Value

A vector with group membership.

cutree_rainette2 *Cut a rainette2 result object into groups of documents*

Description

Cut a rainette2 result object into groups of documents

Usage

```
cutree_rainette2(res, k, criterion = c("chi2", "n"), ...)
```

Arguments

res	the rainette2 result object to be cut
k	the desired number of groups
criterion	criterion to use to choose the best partition. chi2 means the partition with the maximum sum of chi2, n the partition with the maximum size.
...	arguments passed to other methods

Value

A vector with group membership.

See Also

`rainette2_complete_groups()`

`import_corpus_iramuteq` *Import a corpus in Iramuteq format*

Description

Import a corpus in Iramuteq format

Usage

```
import_corpus_iramuteq(f, id_var = NULL, thematics = c("remove", "split"), ...)
```

Arguments

`f`	a file name or a connection
`id_var`	name of metadata variable to be used as documents id
`thematics`	if "remove", thematics lines are removed. If "split", texts as splitted at each thematic, and metadata duplicated accordingly
`...`	arguments passed to `base::file()` if `f` is a file name.

Details

A description of the Iramuteq corpus format can be found here : `http://www.iramuteq.org/documentation/html/2-2-2-les-regles-de-formatages`

Value

A quanteda corpus object. Note that metadata variables in docvars are all imported as characters.

`order_docs` *return documents indices ordered by CA first axis coordinates*

Description

return documents indices ordered by CA first axis coordinates

Usage

```
order_docs(m)
```

Arguments

`m`	dtm on which to compute the CA and order documents, converted to an integer matrix.

Details

Internal function, not to be used directly

Value

ordered list of document indices

`rainette` *Corpus clustering based on the Reinert method - Simple clustering*

Description

Corpus clustering based on the Reinert method - Simple clustering

Usage

```
rainette(
  dtm,
  k = 10,
  min_uc_size = 10,
  min_split_members = 5,
  cc_test = 0.3,
  tsj = 3,
  min_members
)
```

Arguments

`dtm`	quanteda dfm object of documents to cluster, usually the result of `split_segments()`
`k`	maximum number of clusters to compute
`min_uc_size`	minimum number of forms by document
`min_split_members`	don’t try to split groups with fewer members
`cc_test`	contingency coefficient value for feature selection
`tsj`	minimum frequency value for feature selection
`min_members`	deprecated, use `min_split_members` instead

Details

See the references for original articles on the method. Computations and results may differ quite a bit, see the package vignettes for more details.

The dtm object is automatically converted to boolean.

Value

The result is a list of both class hclust and rainette. Besides the elements of an hclust object, two more results are available :

- uce_groups give the group of each document for each k
- group give the group of each document for the maximum value of k available

References

- Reinert M, Une méthode de classification descendante hiérarchique : application à l'analyse lexicale par contexte, Cahiers de l'analyse des données, Volume 8, Numéro 2, 1983. http://www.numdam.org/item/?id=CAD_1983__8_2_187_0
- Reinert M., Alceste une méthodologie d'analyse des données textuelles et une application: Aurelia De Gerard De Nerval, Bulletin de Méthodologie Sociologique, Volume 26, Numéro 1, 1990. https://doi.org/10.1177/075910639002600103

See Also

split_segments(), rainette2(), cutree_rainette(), rainette_plot(), rainette_explor()

Examples

```
require(quanteda)
corpus <- data_corpus_inaugural
corpus <- head(corpus, n = 10)
corpus <- split_segments(corpus)
dtm <- dfm(corpus, remove = stopwords("en"), tolower = TRUE, remove_punct = TRUE)
dtm <- dfm_wordstem(dtm, language = "english")
dtm <- dfm_trim(dtm, min_termfreq = 3)
res <- rainette(dtm, k = 3)
```

rainette2 *Corpus clustering based on the Reinert method - Double clustering*

Description

Corpus clustering based on the Reinert method - Double clustering

Usage

```
rainette2(
  x,
  y = NULL,
  max_k = 5,
  uc_size1 = 10,
  uc_size2 = 15,
  min_members = 10,
  min_chi2 = 3.84,
  ...
)
```

Arguments

`x` either a quanteda dfm object or the result of `rainette()`

`y` if `x` is a `rainette()` result, this must be another `rainette()` result from same dfm but with different uc size.

`max_k` maximum number of clusters to compute

`uc_size1` if `x` is a dfm, minimum uc size for first clustering

`uc_size2` if `x` is a dfm, minimum uc size for second clustering

`min_members` minimum members of each cluster

`min_chi2` minimum chi2 for each cluster

`...` if `x` is a dfm object, parameters passed to `rainette()` for both simple clusterings

Details

You can pass a quanteda dfm as `x` object, the function then performs two simple clustering with varying minimum uc size, and then proceed to find optimal partitions based on the results of both clusterings.

If both clusterings have already been computed, you can pass them as `x` and `y` arguments and the function will only look for optimal partitions.

For more details on optimal partitions search algorithm, please see package vignettes.

Value

A tibble with optimal partitions found for each available value of `k` as rows, and the following columns :

- `clusters` list of the crossed original clusters used in the partition
- `k` the number of clusters
- `chi2` sum of the chi2 value of each cluster
- `n` sum of the size of each cluster
- `groups` group membership of each document for this partition (`NA` if not assigned)

References

- Reinert M, Une méthode de classification descendante hiérarchique : application à l'analyse lexicale par contexte, Cahiers de l'analyse des données, Volume 8, Numéro 2, 1983. http://www.numdam.org/item/?id=CAD_1983__8_2_187_0
- Reinert M., Alceste une méthodologie d'analyse des données textuelles et une application: Aurelia De Gerard De Nerval, Bulletin de Méthodologie Sociologique, Volume 26, Numéro 1, 1990. https://doi.org/10.1177/075910639002600103

See Also

rainette(), cutree_rainette2(), rainette2_plot(), rainette2_explor()

Examples

```
require(quanteda)
mini_corpus <- head(data_corpus_inaugural, n = 2)
mini_corpus <- split_segments(mini_corpus, 5)
dtm <- dfm(mini_corpus, remove = stopwords("en"), tolower = TRUE, remove_punct = TRUE)
dtm <- dfm_wordstem(dtm, language = "english")
dtm <- dfm_trim(dtm, min_termfreq = 3)

res1 <- rainette(dtm, k = 5, min_uc_size = 2, min_split_members = 2)
res2 <- rainette(dtm, k = 5, min_uc_size = 3, min_split_members = 2)

res <- rainette2(res1, res2, min_members = 2)
```

rainette2_complete_groups

Complete groups membership with knn classification

Description

Starting with groups membership computed from a rainette2 clustering, every document not assigned to a cluster is reassigned using a k-nearest neighbour classification.

Usage

```
rainette2_complete_groups(dfm, groups, k = 1, ...)
```

Arguments

dfm	dfm object used for rainette2 clustering.
groups	group membership computed by cutree on rainette2 result.
k	number of neighbours considered.
...	other arguments passed to FNN::knn.

Value

Completed group membership vector.

See Also

cutree_rainette2(), FNN::knn()

rainette2_explor *Shiny gadget for rainette2 clustering exploration*

Description

Shiny gadget for rainette2 clustering exploration

Usage

```
rainette2_explor(res, dtm)
```

Arguments

res	result object of a `rainette2` clustering
dtm	the dfm object used to compute the clustering

Value

No return value, called for side effects.

See Also

`rainette2_plot()`

rainette2_plot *Generate a clustering description plot from a rainette2 result*

Description

Generate a clustering description plot from a rainette2 result

Usage

```
rainette2_plot(
  res,
  dtm,
  k = NULL,
  criterion = c("chi2", "n"),
  complete_groups = FALSE,
  type = c("bar", "cloud"),
  n_terms = 15,
  free_scales = FALSE,
  measure = c("chi2", "lr"),
  show_negative = TRUE,
  text_size = 10
)
```

Arguments

`res`	result object of a `rainette2` clustering
`dtm`	the dfm object used to compute the clustering
`k`	number of groups. If NULL, use the biggest number possible
`criterion`	criterion to use to choose the best partition. `chi2` means the partition with the maximum sum of chi2, `n` the partition with the maximum size.
`complete_groups`	if TRUE, documents with NA cluster are reaffected by k-means clustering initialised with current groups centers.
`type`	type of term plots : barplot or wordcloud
`n_terms`	number of terms to display in keyness plots
`free_scales`	if TRUE, all the keyness plots will have the same scale
`measure`	statistics to compute
`show_negative`	if TRUE, show negative keyness features
`text_size`	font size for barplots, max word size for wordclouds

Value

A gtable object.

See Also

`quanteda::textstat_keyness()`, `rainette2_explor()`, `rainette2_complete_groups()`

`rainette_explor` *Shiny gadget for rainette clustering exploration*

Description

Shiny gadget for rainette clustering exploration

Usage

```
rainette_explor(res, dtm)
```

Arguments

`res`	result object of a `rainette` clustering
`dtm`	the dfm object used to compute the clustering

Value

No return value, called for side effects.

See Also

rainette_plot

Examples

```
## Not run:
library(quanteda)
corpus <- data_corpus_inaugural
corpus <- head(corpus, n = 10)
corpus <- split_segments(corpus)
dtm <- dfm(corpus, remove = stopwords("en"), tolower = TRUE, remove_punct = TRUE)
dtm <- dfm_trim(dtm, min_termfreq = 3)
res <- rainette(dtm, k = 3)
rainette_explor(dtm, res)

## End(Not run)
```

rainette_plot *Generate a clustering description plot from a rainette result*

Description

Generate a clustering description plot from a rainette result

Usage

```
rainette_plot(
  res,
  dtm,
  k = NULL,
  type = c("bar", "cloud"),
  n_terms = 15,
  free_scales = FALSE,
  measure = c("chi2", "lr"),
  show_negative = TRUE,
  text_size = NULL
)
```

Arguments

`res`	result object of a `rainette` clustering
`dtm`	the dfm object used to compute the clustering
`k`	number of groups. If NULL, use the biggest number possible
`type`	type of term plots : barplot or wordcloud
`n_terms`	number of terms to display in keyness plots

`free_scales` if TRUE, all the keyness plots will have the same scale

`measure` statistics to compute

`show_negative` if TRUE, show negative keyness features

`text_size` font size for barplots, max word size for wordclouds

Value

A gtable object.

See Also

`quanteda::textstat_keyness()`, `rainette_explor()`, `rainette_stats()`

Examples

```
library(quanteda)
corpus <- data_corpus_inaugural
corpus <- head(corpus, n = 10)
corpus <- split_segments(corpus)
dtm <- dfm(corpus, remove = stopwords(”en”), tolower = TRUE, remove_punct = TRUE)
dtm <- dfm_trim(dtm, min_termfreq = 3)
res <- rainette(dtm, k = 3)
rainette_plot(res, dtm)
```

`rainette_stats` *Generate cluster keyness statistics from a rainette result*

Description

Generate cluster keyness statistics from a rainette result

Usage

```
rainette_stats(
  groups,
  dtm,
  measure = c(”chi2”, ”lr”),
  n_terms = 15,
  show_negative = TRUE,
  max_p = 0.05
)
```

Arguments

groups	groups membership computed by cutree_rainette or cutree_rainette2
dtm	the dfm object used to compute the clustering
measure	statistics to compute
n_terms	number of terms to display in keyness plots
show_negative	if TRUE, show negative keyness features
max_p	maximum keyness statistic p-value

Value

A list with, for each group, a data.frame of keyness statistics for the most specific n_terms features.

See Also

quanteda::textstat_keyness(), rainette_explor(), rainette_plot()

Examples

```
library(quanteda)
corpus <- data_corpus_inaugural
corpus <- head(corpus, n = 10)
corpus <- split_segments(corpus)
dtm <- dfm(corpus, remove = stopwords("en"), tolower = TRUE, remove_punct = TRUE)
dtm <- dfm_trim(dtm, min_termfreq = 3)
res <- rainette(dtm, k = 3)
groups <- cutree_rainette(res, k = 3)
rainette_stats(groups, dtm)
```

select_features *Remove features from dtm of each group base don cc_test and tsj*

Description

Remove features from dtm of each group base don cc_test and tsj

Usage

```
select_features(m, indices1, indices2, cc_test = 0.3, tsj = 3)
```

Arguments

m	global dtm
indices1	indices of documents of group 1
indices2	indices of documents of group 2
cc_test	maximum contingency coefficient value for the feature to be kept in both groups.
tsj	minimum feature frequency in the dtm

Details

Internal function, not to be used directly

Value

a list of two character vectors : `cols1` is the name of features to keep in group 1, `cols2` the name of features to keep in group 2

`split_segments` *Split a character string or corpus into segments*

Description

Split a character string or corpus into segments, taking into account punctuation where possible

Usage

```
split_segments(
  obj,
  segment_size = 40,
  segment_size_window = NULL,
  force_single_core = FALSE
)

## S3 method for class 'character'
split_segments(
  obj,
  segment_size = 40,
  segment_size_window = NULL,
  force_single_core = FALSE
)

## S3 method for class 'Corpus'
split_segments(
  obj,
  segment_size = 40,
  segment_size_window = NULL,
  force_single_core = FALSE
)

## S3 method for class 'corpus'
split_segments(
  obj,
  segment_size = 40,
  segment_size_window = NULL,
  force_single_core = FALSE
)
```

Arguments

`obj`	character string, quanteda or tm corpus object
`segment_size`	segment size (in words)
`segment_size_window`	window around segment size to look for best splitting point
`force_single_core`	don't use multithreading even on large corpus

Details

By default, if the corpus is large (> 10 000 000 chars), multithreading is used for segments splitting.

Value

If obj is a tm or quanteda corpus object, the result is a quanteda corpus.

Examples

```
require(quanteda)
split_segments(data_corpus_inaugural)
```

switch_docs *Switch documents between two groups to maximize chi-square value*

Description

Switch documents between two groups to maximize chi-square value

Usage

```
switch_docs(m, indices, max_index, max_chisq)
```

Arguments

`m`	original dtm
`indices`	documents indices orderes by first CA axis coordinates
`max_index`	document index where the split is maximum
`max_chisq`	maximum chi-square value

Details

Internal function, not to be used directly

Value

a list of two vectors `indices1` and `indices2`, which contain the documents indices of each group after documents switching, and a `chisq` value, the new corresponding chi-square value after switching

Index