Package 'polmineR'

July 9, 2018

Type Package

Title Toolkit for Corpus Analysis

Version 0.7.9

Date 2018-07-09

Imports methods, R6, data.table, slam, Matrix, tm, DT, xml2, stringi, utils, jsonlite, parallel, pbapply, RcppCWB (>= 0.2.2)

Suggests markdown, htmltools, xlsx, sendmailR, knitr, shiny, shinythemes, testthat, tidytext, magrittr

VignetteBuilder knitr

LazyData yes

Description Library for corpus analysis using the Corpus Workbench as an efficient back end for indexing and querying large corpora. The package offers functionality to flexibly create partitions and to carry out basic statistical operations (count, co-occurrences etc.). The original full text of documents can be reconstructed and inspected at any time. Beyond that, the package is intended to serve as an interface to packages implementing advanced statistical procedures. Respective data structures (document term matrices, term co-occurrence matrices etc.) can be created based on the indexed corpora.

BugReports https://github.com/PolMine/polmineR/issues

Biarch true

License GPL-3

URL https://www.github.com/PolMine/polmineR

Collate 'CQI.R' 'Labels.R' 'polmineR.R' 'S4classes.R' 'p_attributes.R' 'textstat.R' 'bundle.R' 'corpus.R' 'count.R' 'partition.R' 'partition_bundle.R' 'ngrams.R' 'features.R' 'context.R' 'TermDocumentMatrix.R' 'as.VCorpus.R' 'as.markdown.R' 'cooccurrences.R' 'as.sparseMatrix.R' 'as.speeches.R' 'blapply.R' 'kwic.R' 'browse.R' 'chisquare.R' 'hits.R' 'tempcorpus.R' 'cpos.R' 'cqpserver.R' 'decode.R' 'dispersion.R' 'dotplot.R' 'encoding.R' 'enrich.R' 'highlight.R' 'html.R' 'label.R' 'll.R' 'mail.R' 'means.R' 'noise.R' 'pmi.R'

'regions.R' 'read.R' 'registry.R' 'reindex.R' 'renamed.R'
's_attributes.R' 'size.R' 't_test.R' 'templates.R' 'terms.R'
'token_stream.R' 'tooltips.R' 'trim.R' 'type.R' 'use.R'
'utils.R' 'view.R' 'weigh.R' 'zzz.R'

RoxygenNote 6.0.1

NeedsCompilation no

Author Andreas Blaette [aut, cre] (<https://orcid.org/0000-0001-8970-8010>)

Maintainer Andreas Blaette <andreas.blaette@uni-due.de>

Repository CRAN

Date/Publication 2018-07-09 13:10:03 UTC

R topics documented:

polmineR-package *polmineR-package*

Description

A library for corpus analysis using the Corpus Workbench (CWB) as an efficient back end for indexing and querying large corpora.

Usage

```
polmineR()
```

Details

The package offers functionality to flexibly create partitions and to carry out basic statistical operations (count, co-occurrences etc.). The original full text of documents can be reconstructed and inspected at any time. Beyond that, the package is intended to serve as an interface to packages implementing advanced statistical procedures. Respective data structures (document term matrices, term co- occurrence matrices etc.) can be created based on the indexed corpora.

A session registry directory (see `registry()`) combines the registry files for corpora that may reside in anywhere on the system. Upon loading `polmineR`, the files in the registry directory defined by the environment variable CORPUS_REGISTRY are copied to the session registry directory. To see whether the environment variable CORPUS_REGISTRY is set, use the 'Sys.getenv()'-function. Corpora wrapped in R data packages can be activated using the function `use()`.

The package includes a draft shiny app that can be called using `polmineR()`.

Author(s)

Andreas Blaette (andreas.blaette@uni-due.de)

References

Jockers, Matthew L. (2014): *Text Analysis with R for Students of Literature*. Cham et al: Springer.

Baker, Paul (2006): *Using Corpora in Discourse Analysis*. London: continuum.

Examples

```
use("polmineR") # activate demo corpora included in the package

# Core methods applied to corpus

count("REUTERS", query = "oil")
count("REUTERS", query = c("oil", "barrel"))
count("REUTERS", query = '"Saudi" "Arab.*"', breakdown = TRUE, cqp = TRUE)
dispersion("REUTERS", query = "oil", s_attribute = "id")
kwic("REUTERS", query = "oil")
cooccurrences("REUTERS", query = "oil")
```

```
# Core methods applied to partition

kuwait <- partition("REUTERS", places = "kuwait", regex = TRUE)
count(kuwait, query = "oil")
dispersion(kuwait, query = "oil", s_attribute = "id")
kwic(kuwait, query = "oil", meta = "id")
cooccurrences(kuwait, query = "oil")


# Go back to full text

p <- partition("REUTERS", id = 127)
read(p)
h <- html(p)
h_highlighted <- highlight(h, highlight = list(yellow = "oil"))
h_highlighted


# Generate term document matrix

pb <- partition_bundle("REUTERS", s_attribute = "id")
cnt <- count(pb, p_attribute = "word")
tdm <- as.TermDocumentMatrix(cnt, col = "count")
```

---

as.markdown *Get markdown-formatted full text of a partition.*

Description

The method is the worker behind the read-method, which will be called usually to reconstruct the full text of a partition and read it. The as.markdown-method can be customized for different classes inheriting from the partition-class.

Usage

```
as.markdown(.Object, ...)

## S4 method for signature 'partition'
as.markdown(.Object, meta = getOption("polmineR.meta"),
  template = get_template(.Object), cpos = TRUE, cutoff = NULL,
  verbose = FALSE, ...)

## S4 method for signature 'plpr_partition'
as.markdown(.Object, meta = NULL,
  template = get_template(.Object), cpos = FALSE, interjections = TRUE,
  cutoff = NULL, ...)
```

Arguments

.Object	The object to be converted, a partition, or a class inheriting from partition, such as plpr_partition.
...	further arguments
meta	The metainformation (s-attributes) to be displayed.
template	A template for formating output.
cpos	A logical value, whether to add cpos as ids in span elements.
cutoff	The maximum number of tokens to reconstruct, to avoid that full text is excessively long.
verbose	A logical value, whether to output messages.
interjections	A logical value, whether to format interjections.

Examples

```
use("polmineR")
P <- partition("REUTERS", places = "argentina")
as.markdown(P)
as.markdown(P, meta = c("id", "places"))
if (interactive()) read(P, meta = c("id", "places"))
```

---

as.sparseMatrix *Type conversion - get sparseMatrix.*

Description

Turn objects into the sparseMatrix as defined in the Matrix package.

Usage

```
as.sparseMatrix(x, ...)

## S4 method for signature 'simple_triplet_matrix'
as.sparseMatrix(x, ...)

## S4 method for signature 'TermDocumentMatrix'
as.sparseMatrix(x, ...)

## S4 method for signature 'bundle'
as.sparseMatrix(x, col)
```

Arguments

x	object to convert
...	further parameters
col	column name to get values from (if x is a bundle)

as.speeches *Split corpus or partition into speeches.*

Description

Split entire corpus or a partition into speeches. The heuristic is to split the corpus/partition into partitions on day-to-day basis first, using the s-attribute provided by s_attribute_date. These subcorpora are then splitted into speeches by speaker name, using s-attribute s_attribute_name. If there is a gap larger than the number of tokens supplied by argument gap, contributions of a speaker are assumed to be two seperate speeches.

Usage

```
as.speeches(.Object, s_attribute_date = grep("date", s_attributes(.Object),
  value = TRUE), s_attribute_name = grep("name", s_attributes(.Object), value
  = TRUE), gap = 500, mc = FALSE, verbose = TRUE, progress = TRUE)
```

Arguments

.Object	A partition, or length-one character vector indicating a CWB corpus.
s_attribute_date	The s-attribute that provides the dates of sessions.
s_attribute_name	The s-attribute that provides the names of speakers.
gap	Number of tokens between strucs assumed to make the difference whether a speech has been interrupted (by an interjection or question), or whether to assume seperate speeches.
mc	Whether to use multicore, defaults to FALSE.
verbose	A logical value, defaults to TRUE.
progress	logical

Value

A partition_bundle, the names of the objects in the bundle are the speaker name, the date of the speech and an index for the number of the speech on a given day, concatenated by underscores.

Examples

```
use("polmineR")
speeches <- as.speeches(
  "GERMAPARLMINI",
  s_attribute_date = "date", s_attribute_name = "speaker"
)
speeches_count <- count(speeches, p_attribute = "word")
tdm <- as.TermDocumentMatrix(speeches_count, col = "count")
```

```
bt <- partition("GERMAPARLMINI", date = "2009-10-27")
speeches <- as.speeches(bt, s_attribute_name = "speaker")
summary(speeches)
```

`as.TermDocumentMatrix` *Generate TermDocumentMatrix / DocumentTermMatrix.*

Description

Methods to generate the classes `TermDocumentMatrix` or `DocumentTermMatrix` as defined in the `tm` package. These classes inherit from the `simple_triplet_matrix`-class defined in the `slam`-package. There are many text mining applications for document-term matrices. A `DocumentTermMatrix` is required as input by the `topicmodels` package, for instance.

Usage

```
as.TermDocumentMatrix(x, ...)

## S4 method for signature 'character'
as.TermDocumentMatrix(x, p_attribute, s_attribute,
  verbose = TRUE, ...)

## S4 method for signature 'character'
as.DocumentTermMatrix(x, p_attribute, s_attribute,
  verbose = TRUE, ...)

## S4 method for signature 'bundle'
as.TermDocumentMatrix(x, col, p_attribute = NULL,
  verbose = TRUE, ...)

## S4 method for signature 'bundle'
as.DocumentTermMatrix(x, col, p_attribute = NULL,
  verbose = TRUE, ...)

## S4 method for signature 'partition_bundle'
as.TermDocumentMatrix(x, p_attribute = NULL,
  col = NULL, verbose = TRUE, ...)

## S4 method for signature 'partition_bundle'
as.DocumentTermMatrix(x, p_attribute = NULL,
  col = NULL, verbose = TRUE, ...)

## S4 method for signature 'context'
as.DocumentTermMatrix(x, p_attribute, verbose = TRUE, ...)

## S4 method for signature 'context'
as.TermDocumentMatrix(x, p_attribute, verbose = TRUE, ...)
```

Arguments

x	a character vector indicating a corpus, or an object of class bundle, or inheriting from class bundle (e.g. partition_bundle)
...	s-attribute definitions used for subsetting the corpus, compare partition-method
p_attribute	p-attribute counting is be based on
s_attribute	s-attribute that defines content of columns, or rows
verbose	logial, whether to output progress messages
col	the column of data.table in slot stat (if x is a bundle) to use of assembling the matrix

Details

The method can be applied on objects of the class character, bundle, or classes inheriting from the bundle class.

If x refers to a corpus (i.e. is a length 1 character vector), a TermDocumentMatrix, or DocumentTermMatrix will be generated for subsets of the corpus based on the s_attribute provided. Counts are performed for the p_attribute. Further parameters provided (passed in as ... are interpreted as s-attributes that define a subset of the corpus for splitting it according to s_attribute. If struc values for s_attribute are not unique, the necessary aggregation is performed, slowing things somewhat down.

If x is a bundle or a class inheriting from it, the counts or whatever measure is present in the stat slots (in the column indicated by col) will be turned into the values of the sparse matrix that is generated. A special case is the generation of the sparse matrix based on a partition_bundle that does not yet include counts. In this case, a p_attribute needs to be provided. Then counting will be performed, too.

Value

a TermDocumentMatrix

Author(s)

Andreas Blaette

Examples

```
use("polmineR")

# do-it-yourself
p <- partition("GERMAPARLMINI", date = ".*", regex = TRUE)
pB <- partition_bundle(p, s_attribute = "date")
pB <- enrich(pB, p_attribute="word")
tdm <- as.TermDocumentMatrix(pB, col = "count")

 # leave the counting to the as.TermDocumentMatrix-method
pB2 <- partition_bundle(p, s_attribute = "date")
tdm <- as.TermDocumentMatrix(pB2, p_attribute = "word", verbose = TRUE)
```

```
# diretissima
tdm <- as.TermDocumentMatrix("GERMAPARLMINI", p_attribute = "word", s_attribute = "date")
```

`as.VCorpus,partition_bundle-method` *Coerce partition_bundle to VCorpus.*

Description

Coerce partition_bundle to VCorpus.

Usage

```
## S4 method for signature 'partition_bundle'
as.VCorpus(x)
```

Arguments

x	a `partition_bundle` object

Examples

```
use("polmineR")
P <- partition("GERMAPARLMINI", date = "2009-11-10")
PB <- partition_bundle(P, s_attribute = "speaker")
VC <- as.VCorpus(PB)
```

`blapply` *apply a function over a list or bundle*

Description

Very similar to lapply, but applicable to bundle-objects, in particular. The purpose of the method is to supply a uniform und convenient parallel backend for the polmineR package. In particular, progress bars are supported (the naming of the method is derived from bla bla).

Usage

```
blapply(x, ...)

## S4 method for signature 'list'
blapply(x, f, mc = TRUE, progress = TRUE,
  verbose = FALSE, ...)

## S4 method for signature 'vector'
blapply(x, f, mc = FALSE, progress = TRUE,
```

```
  verbose = FALSE, ...)

## S4 method for signature 'bundle'
blapply(x, f, mc = FALSE, progress = TRUE,
  verbose = FALSE, ...)
```

Arguments

x	a list or a bundle object
...	further parameters
f	a function that can be applied to each object contained in the bundle, note that it should swallow the parameters mc, verbose and progress (use ... to catch these params)
mc	logical, whether to use multicore - if TRUE, the number of cores will be taken from the polmineR-options
progress	logical, whether to display progress bar
verbose	logical, whether to print intermediate messages

Examples

```
use("polmineR")
bt <- partition("GERMAPARLMINI", date = ".*", regex=TRUE)
speeches <- as.speeches(bt, s_attribute_date = "date", s_attribute_name = "speaker")
foo <- blapply(speeches, function(x, ...) slot(x, "cpos"))
```

---

browse *Display in browser*

Description

Display in browser

Usage

```
browse(object, ...)

## S4 method for signature 'textstat'
browse(object)

## S4 method for signature 'cooccurrences'
browse(object)

## S4 method for signature 'partition'
browse(object, meta = NULL)

## S4 method for signature 'html'
```

```
browse(object)

## S4 method for signature 'kwic'
browse(object, colnames = NULL)

## S4 method for signature 'press_partition'
browse(object, meta = c("text_newspaper",
  "text_date"))
```

Arguments

object	what is to be displayed
...	further parameters
meta	metainformation to be displayed
colnames	colnames to be used for data.frame

---

bundle-class *Bundle Class*

Description

A bundle is used to combine several objects (partition, context, features, cooccurrences objects) into one S4 class object. Typically, a class inheriting from the bundle superclass will be used. When working with a context_bundle, a features_bundle, a cooccurrences_bundle, or a context_bundle, a similar set of standard methods is available to perform transformations.

Usage

```
## S4 method for signature 'bundle'
length(x)

## S4 method for signature 'bundle'
names(x)

## S4 replacement method for signature 'bundle,character'
names(x) <- value

## S4 method for signature 'bundle'
unique(x)

## S4 method for signature 'bundle,bundle'
e1 + e2

## S4 method for signature 'bundle,textstat'
e1 + e2
```

```
## S4 method for signature 'bundle'
x[[i]]

## S4 method for signature 'bundle'
sample(x, size)

## S4 method for signature 'list'
as.bundle(object, ...)

## S4 method for signature 'textstat'
as.bundle(object)

## S4 method for signature 'bundle'
as.data.table(x, col)

## S4 method for signature 'bundle'
as.matrix(x, col)

## S4 method for signature 'bundle'
subset(x, ...)

## S4 method for signature 'bundle'
as.list(x)
```

Arguments

x	a bundle object
value	character string with a name to be assigned
e1	object 1
e2	object 2
i	integer to index a bundle object
size	number of items to choose to generate a sample
object	a bundle object
...	further parameters
col	columns of the data.table to use to generate an object

Slots

`corpus` The CWB corpus the objects in the `bundle` are based on, a length 1 `character` vector.

`objects` An object of class `"list"`

`p_attribute` Object of class `"character"`

`encoding` The encoding of the corpus.

Author(s)

Andreas Blaette

Examples

```
parties <- s_attributes("GERMAPARLMINI", "party")
parties <- parties[-which(parties == "NA")]
party_bundle <- partition_bundle("GERMAPARLMINI", s_attribute = "party")
length(party_bundle)
names(party_bundle)
party_bundle <- enrich(party_bundle, p_attribute = "word")
summary(party_bundle)
parties_big <- party_bundle[[c("CDU_CSU", "SPD")]]
summary(parties_big)
use("polmineR")
Ps <- partition_bundle(
  "REUTERS", s_attribute = "id",
  values = s_attributes("REUTERS", "id")
)
Cs <- cooccurrences(Ps, query = "oil", cqp = FALSE, verbose = FALSE, progress = TRUE)
dt <- as.data.table(Cs, col = "ll")
m <- as.matrix(Cs, col = "ll")
```

---

chisquare *perform chisquare-text*

Description

Perform Chisquare-Test based on a table with counts

Usage

```
chisquare(.Object, ...)

## S4 method for signature 'textstat'
chisquare(.Object)

## S4 method for signature 'context'
chisquare(.Object)
```

Arguments

.Object	object
...	further parameters

Details

This function deliberately uses a self-made chi-square test for performance reason

Value

a table

Author(s)

Andreas Blaette

`context` *Analyze context of a node word.*

Description

Retrieve the word context of a token, optionally checking for boundaries of a XML region.

Usage

```
## S4 method for signature 'partition'
context(.Object, query, cqp = is.cqp,
  left = getOption("polmineR.left"), right = getOption("polmineR.right"),
  p_attribute = getOption("polmineR.p_attribute"), s_attribute = NULL,
  stoplist = NULL, positivelist = NULL, regex = FALSE, count = TRUE,
  mc = getOption("polmineR.mc"), verbose = TRUE, progress = TRUE, ...)

## S4 method for signature 'character'
context(.Object, query, cqp = is.cqp,
  p_attribute = getOption("polmineR.p_attribute"), s_attribute = NULL,
  left = getOption("polmineR.left"), right = getOption("polmineR.right"),
  stoplist = NULL, positivelist = NULL, regex = FALSE, count = TRUE,
  mc = getOption("polmineR.mc"), verbose = TRUE, progress = TRUE, ...)

## S4 method for signature 'partition_bundle'
context(.Object, query, p_attribute,
  verbose = TRUE, ...)

## S4 method for signature 'cooccurrences'
context(.Object, query, complete = FALSE)

## S4 method for signature 'Corpus'
cooccurrences(.Object, query,
  p_attribute = getOption("polmineR.p_attribute"), ...)
```

Arguments

`.Object`	a partition or a partition_bundle object
`query`	query, which may by a character vector or a cqpQuery object
`cqp`	defaults to is.cqp-function, or provide TRUE/FALSE
`left`	no of tokens and to the left of the node word
`right`	no of tokens to the right of the node word
`p_attribute`	p-attribute of the query

s_attribute	if provided, it will be checked that corpus positions do not extend beyond the region defined by the s-attribute
stoplist	exclude a query hit from analysis if stopword(s) is/are in context. See positivelist for further explanation.
positivelist	character vector or numeric/integer vector: include a query hit only if token in positivelist is present. If positivelist is a character vector, it may include regular expressions (see parameter regex)
regex	logical, defaults to FALSE - whether stoplist and/or positivelist are regular expressions
count	logical
mc	whether to use multicore; if NULL (default), the function will get the value from the options
verbose	report progress, defaults to TRUE
progress	logical, whether to show progress bar
...	further parameters
complete	enhance completely

Details

For formulating the query, CPQ syntax may be used (see examples). Statistical tests available are log-likelihood, t-test, pmi.

Value

depending on whether a partition or a partition_bundle serves as input, the return will be a context object, or a context_bundle object

Author(s)

Andreas Blaette

Examples

```
use("polmineR")
p <- partition("GERMAPARLMINI", interjection = "speech")
y <- context(p, query = "Integration", p_attribute = "word")
y <- context(p, query = "Integration", p_attribute = "word", positivelist = "Bildung")
y <- context(
  p, query = "Integration", p_attribute = "word",
  positivelist = c("[aA]rbeit.*", "Ausbildung"), regex = TRUE
)
```

context-class *Context class.*

Description

Class to organize information of context analysis.

Usage

```
## S4 method for signature 'context'
p_attributes(.Object)

## S4 method for signature 'context'
count(.Object)

## S4 method for signature 'context'
sample(x, size)

## S4 method for signature 'context'
enrich(.Object, s_attribute = NULL, p_attribute = NULL,
  decode = FALSE, verbose = TRUE, ...)

## S4 method for signature 'context'
trim(object, s_attribute = NULL, positivelist = NULL,
  p_attribute = p_attributes(object), regex = FALSE, stoplist = NULL,
  verbose = TRUE, progress = TRUE, ...)
```

Arguments

.Object	object
x	a context object
size	integer indicating sample size
s_attribute	s-attribute(s) to add to data.table in cpos-slot
p_attribute	p-attribute(s) to add to data.table in cpos-slot
decode	logical, whether to convert integer ids to expressive strings
verbose	logical, whether to be talkative
...	to maintain backwards compatibility if argument pAttribute is still used
object	a context object
positivelist	tokens that are required to be present to keep a match
regex	logical, whether positivlist / stoplist is interpreted as regular expressions
stoplist	tokens that are used to exclude a match
progress	logical, whether to show progress bar

Details

Objects of the class context include a data.table in the slot cpos. The data.table will at least include the columns "hit_no", "cpos" and "position".

The enrich-method can be used to add additional information to the data.table in the "cpos"-slot of a context-object.

Slots

query Object of class "character", the query/node examined

count Object of class "numeric" number of hits

partition Object of class "partition", the partition the context object is based on

size_partition Object of class "integer" the size of the partition

left Object of class "numeric" number of tokens to the left

right Object of class "numeric" number of tokens to the right

size Object of class "numeric" number of tokens in the right and left context

s_attribute Object of class "character" s-attribute

p_attribute Object of class "character" p-attribute of the query

corpus Object of class "character" the CWB corpus used

stat Object of class "data.table" statistics of the analysis

encoding Object of class "character" encoding of the corpus

cpos Object of class "list" corpus positions of the hits

method Object of class "character" statistical test used

call Object of class "character" call that generated the object

---

context_bundle-class *S4 context_bundle class*

Description

class to organize information of multiple context analyses

Slots

objects Object of class "list" a list of context objects

Methods

show output of core information

summary core statistical information

[specific cooccurrences

[[specific cooccurrences

cooccurrences *Get cooccurrence statistics.*

Description

Get cooccurrence statistics.

Usage

```
cooccurrences(.Object, ...)

## S4 method for signature 'character'
cooccurrences(.Object, query, cqp = is.cqp,
  p_attribute = getOption("polmineR.p_attribute"), s_attribute = NULL,
  left = getOption("polmineR.left"), right = getOption("polmineR.right"),
  stoplist = NULL, positivelist = NULL, regex = FALSE, keep = NULL,
  cpos = NULL, method = "ll", mc = getOption("polmineR.mc"),
  verbose = FALSE, progress = FALSE, ...)

## S4 method for signature 'partition'
cooccurrences(.Object, query, cqp = is.cqp,
  left = getOption("polmineR.left"), right = getOption("polmineR.right"),
  p_attribute = getOption("polmineR.p_attribute"), s_attribute = NULL,
  stoplist = NULL, positivelist = NULL, keep = NULL, method = "ll",
  mc = FALSE, progress = TRUE, verbose = FALSE, ...)

## S4 method for signature 'context'
cooccurrences(.Object, method = "ll", verbose = FALSE)

## S4 method for signature 'partition_bundle'
cooccurrences(.Object, query,
  mc = getOption("polmineR.mc"), ...)
```

Arguments

.Object	a partition object, or a character vector with a CWB corpus
...	further parameters that will be passed into bigmatrix (applies only of big=TRUE)
query	query, may by a character vector to match a token, or a CQP query
cqp	defaults to is.cqp-function, or provide TRUE/FALSE, relevant only if query is not NULL
p_attribute	the p-attribute of the tokens/the query
s_attribute	if provided, it will be checked that cpos do not extend beyond the region defined by the s-attribute
left	no of tokens and to the left of the node word
right	no of tokens to the right of the node word

stoplist	exclude a query hit from analysis if stopword(s) is/are in context (relevant only if query is nut NULL)
positivelist	character vector or numeric vector: include a query hit only if token in positivelist is present. If positivelist is a character vector, it is assumed to provide regex expressions (incredibly long if the list is long) (relevant only if query is nut NULL)
regex	logical, whether stoplist/positivelist are dealt with as regular expressions
keep	list with tokens to keep
cpos	integer vector with corpus positions, defaults to NULL - then the corpus positions for the whole corpus will be used
method	statistical test to use (defaults to "ll")
mc	whether to use multicore
verbose	logical, whether to be verbose
progress	logical, whether to be verbose

Value

a cooccurrences-class object

Author(s)

Andreas Blaette

References

Baker, Paul (2006): *Using Corpora in Discourse Analysis*. London: continuum, p. 95-120 (ch. 5).

Manning, Christopher D.; Schuetze, Hinrich (1999): *Foundations of Statistical Natural Language Processing*. MIT Press: Cambridge, Mass., pp. 151-189 (ch. 5).

Examples

```
use("polmineR")
merkel <- partition("GERMAPARLMINI", interjection = "speech", speaker = ".*Merkel", regex = TRUE)
merkel <- enrich(merkel, p_attribute = "word")
cooc <- cooccurrences(merkel, query = "Deutschland")
```

---

cooccurrences-class *Cooccurrences class.*

Description

S4 class to organize information of context analysis

Usage

```
## S4 method for signature 'cooccurrences'
summary(object)

## S4 method for signature 'cooccurrences'
show(object)

## S4 method for signature 'cooccurrences_bundle'
as.data.frame(x)

## S4 method for signature 'cooccurrences'
view(.Object)

## S4 method for signature 'cooccurrences_reshaped'
view(.Object)
```

Arguments

object	object to work with
x	object to work with
.Object	object to work with

Slots

`call` Object of class `character` the call that generated the object

`partition` Object of class `character` the partition the analysis is based on

`size_partition` Object of class `integer` the size of the partition

`left` Object of class `numeric` number of tokens to the right

`right` Object of class `numeric` number of tokens to the left

`p_attribute` Object of class `character` p-attribute of the query

`corpus` Object of class `character` the CWB corpus used

`stat` Object of class `data.table` statistics of the analysis

`encoding` Object of class `character` encoding of the corpus

`pos` Object of class `character` part-of-speech tags filtered

`method` Object of class `character` statistical test(s) used

`cutoff` Object of class `list` cutoff levels that have been applied

`svg` Object of class `character` - valid XML with svg representation

cooccurrences_reshaped

Methods for manipulating cooccurrences_reshaped-*class-objects*

Description

Methods for manipulating cooccurrences_reshaped-class-objects

Arguments

x	cooccurrences for a corpus of interest
y	cooccurrences for a reference corpus

Corpus

Corpus class.

Description

The R6 Corpus class offers a set of methods to retrieve and manage CWB indexed corpora.

Usage

```
Corpus
```

Format

An object of class R6ClassGenerator of length 24.

Fields

corpus character vector (length 1), a CWB corpus

encoding encoding of the corpus (typically 'UTF-8' or 'latin1'), assigned automatically upon initialization of the corpus

cpos a two-column matrix with regions of a corpus underlying the s-attributes of the data.table in field s_attributes

s_attributes a data.table with the values of a set of s-attributes

stat a data.table with counts

Arguments

corpus a corpus

registryDir the directory where the registry file resides

dataDir the data directory of the corpus

p_attribute p-attribute, to perform count

s_attributes s-attributes

decode logical, whether to turn token ids into strings upon counting

as.html logical

Methods

`initialize(corpus, p_attribute = NULL, s_attributes = NULL)` Initialize a new object of class `Corpus`.

`count(p_attribute = getOption("polmineR.p_attribute"), decode = TRUE)` Perform counts.

`as.partition()` turn `Corpus` into a partition

`getInfo(as.html = FALSE)`

`showInfo()`

Examples

```
use("polmineR")
REUTERS <- Corpus$new("REUTERS")
infofile <- REUTERS$getInfo()
if (interactive()) REUTERS$showInfo()

# use Corpus class to manage counts
REUTERS <- Corpus$new("REUTERS", p_attribute = "word")
REUTERS$stat

# use Corpus class for creating partitions
REUTERS <- Corpus$new("REUTERS", s_attributes = c("id", "places"))
usa <- partition(REUTERS, places = "usa")
sa <- partition(REUTERS, places = "saudi-arabia", regex = TRUE)

reut <- REUTERS$as.partition()
```

---

corpus *Get corpus.*

Description

Calling `corpus()` will return the corpora available. If the param 'packges' (logical) is TRUE, packages that include a corpus are returned.

Usage

```
corpus(object)

## S4 method for signature 'partition'
corpus(object)

## S4 method for signature 'bundle'
corpus(object)

## S4 method for signature 'missing'
corpus()
```

Arguments

object	the object

Details

If object is a partition or partition_bundle-object, the corpus the respective object is derived from is returned.

---

count *Get counts.*

Description

Count all tokens, or number of occurrences of a query (CQP syntax may be used), or matches for the query.

Usage

```
count(.Object, ...)

## S4 method for signature 'partition'
count(.Object, query = NULL, cqp = is.cqp,
  breakdown = FALSE, decode = TRUE,
  p_attribute = getOption("polmineR.p_attribute"),
  mc = getOption("polmineR.cores"), verbose = TRUE, progress = FALSE, ...)

## S4 method for signature 'partition_bundle'
count(.Object, query = NULL, cqp = FALSE,
  p_attribute = NULL, freq = FALSE, total = TRUE, mc = FALSE,
  progress = TRUE, verbose = FALSE, ...)

## S4 method for signature 'character'
count(.Object, query = NULL, cqp = is.cqp,
  p_attribute = getOption("polmineR.p_attribute"), breakdown = FALSE,
```

```
  sort = FALSE, decode = TRUE, verbose = TRUE, ...)

## S4 method for signature 'vector'
count(.Object, corpus, p_attribute, ...)

## S4 method for signature 'Corpus'
count(.Object, query = NULL, p_attribute)
```

Arguments

.Object	A partition or partition_bundle, or a length-one character vector providing the name of a corpus.
...	Further arguments.
query	A character vector (one or multiple terms), CQP syntax can be used.
cqp	Either logical (TRUE if query is a CQP query), or a function to check whether query is a CQP query or not (defaults to is.query auxiliary function).
breakdown	Logical, whether to report number of occurrences for different matches for a query.
decode	Logical, whether to turn token ids into decoded strings (only if query is NULL).
p_attribute	The p-attribute(s) to use.
mc	Logical, whether to use multicore (defaults to FALSE).
verbose	Logical, whether to be verbose.
progress	Logical, whether to show progress bar.
freq	Logical, if FALSE, counts will be reported, if TRUE, (relative) frequencies are added to table.
total	Defaults to FALSE, if TRUE, the total value of counts (column named 'TOTAL') will be amended to the data.table that is returned.
sort	Logical, whether to sort table with counts (in stat slot).
corpus	The name of a CWB corpus.

Details

If .Object is a partiton_bundle, the data.table returned will have the queries in the columns, and as many rows as there are in the partition_bundle.

If .Object is a length-one character vector and query is NULL, the count is performed for the whole partition.

If breakdown is TRUE and one query is supplied, the function returns a frequency breakdown of the results of the query. If several queries are supplied, frequencies for the individual queries are retrieved.

Value

A data.table if argument query is used, a count-object, if query is NULL and .Object is a character vector (referring to a corpus) or a partition, a count_bundle-object, if .Object is a partition_bundle.

References

Baker, Paul (2006): *Using Corpora in Discourse Analysis*. London: continuum, p. 47-69 (ch. 3).

See Also

For a metadata-based breakdown of counts (i.e. tabulation by s-attributes), see `dispersion`.

count

Examples

```
use("polmineR")
debates <- partition("GERMAPARLMINI", date = ".*", regex=TRUE)
count(debates, query = "Arbeit") # get frequencies for one token
count(debates, c("Arbeit", "Freizeit", "Zukunft")) # get frequencies for multiple tokens

count("GERMAPARLMINI", query = c("Migration", "Integration"), p_attribute = "word")

debates <- partition_bundle(
  "GERMAPARLMINI", s_attribute = "date", values = NULL,
  regex = TRUE, mc = FALSE, verbose = FALSE
)
y <- count(debates, query = "Arbeit", p_attribute = "word")
y <- count(debates, query = c("Arbeit", "Migration", "Zukunft"), p_attribute = "word")

count("GERMAPARLMINI", '"Integration.*"', breakdown = TRUE)

P <- partition("GERMAPARLMINI", date = "2009-11-11")
count(P, '"Integration.*"', breakdown = TRUE)
```

`count_class` *Count class.*

Description

S4 class to organize counts. The classes `polmineR` and `ngrams` inherit from the class.

Usage

```
## S4 method for signature 'count'
length(x)

## S4 method for signature 'count'
hist(x, ...)
```

Arguments

`x`	A `count` object, or a class inheriting from `count`.
`...`	Further parameters.

Details

The length-method is synonymous with the size-method and will return the size of the corpus or partition a count has been derived from.

Slots

stat Object of class data.table

corpus Object of class character the CWB corpus the partition is based on

encoding Object of class character encoding of the corpus

name Object of class character, a name for the object

size Object of class integer, the size of the partition or corpus the count is based upon

Author(s)

Andreas Blaette

See Also

The count-class inherits from the textstat-class

---

cpos *Get corpus positions for a query or queries.*

Description

Get matches for a query in a CQP corpus, optionally using the CQP syntax of the Corpus Workbench (CWB).

Usage

```
cpos(.Object, ...)

## S4 method for signature 'character'
cpos(.Object, query,
  p_attribute = getOption("polmineR.p_attribute"), cqp = is.cqp,
  encoding = NULL, verbose = TRUE, ...)

## S4 method for signature 'partition'
cpos(.Object, query, cqp = is.cqp, p_attribute = NULL,
  verbose = TRUE, ...)

## S4 method for signature 'tempcorpus'
cpos(.Object, query, shift = TRUE)

## S4 method for signature 'matrix'
cpos(.Object)
```

```
## S4 method for signature 'hits'
cpos(.Object)
```

Arguments

.Object	a "character" vector indicating a CWB corpus, a "partition" object, a "tempcorpus" object, or a "matrix" with corpus positions
...	further arguments
query	a character vector providing one or multiple queries (token or CQP query)
p_attribute	the p-attribute to search. Needs to be stated only if query is not a CQP query. Defaults to NULL.
cqp	either logical (TRUE if query is a CQP query), or a function to check whether query is a CQP query or not (defaults to is.query auxiliary function)
encoding	the encoding of the corpus (if NULL, the encoding provided in the registry file of the corpus will be used)
verbose	logical, whether to be talkative
shift	logical, if true, the cpos resulting from the query performed on the tempcorpus will be shifted so that they match the positions of the corpus from which the tempcorpus was generated

Details

If the cpos-method is applied on "character", "partition", or "tempcorpus" object, the result is a two-column matrix with the regions (start end end corpus positions of the matches) for a query. CQP syntax can be used. The encoding of the query is adjusted to conform to the encoding of the CWB corpus.

If the cpos-method is called on a matrix object, the cpos matrix is unfolded, the return value is an integer vector with the individual corpus positions. Equally, if .Object is a hits object, an integer vector is returned with the individual corpus positions.

Value

Unless .Object is a "matrix", you get a matrix with two columns, the first column giving the left/starting corpus positions (cpos) of the hits obtained, the second column giving the right/ending cpos of the respective hit. The number of rows is the number of hits. If there are no hits, a NULL object will is returned.

CQI.super *Interfaces for accessing the CWB*

Description

The package offers two different interfaces to the Corpus Workbench (CWB): The package 'RcppCWB', or via cqpserver. An object called 'CQI' will be instantiated in the environment of the polmineR package; the class will provide the functionality to access CWB corpora.

Usage

```
CQI.super

CQI.RcppCWB

CQI.cqpserver

CQI.cqpserver
```

Format

An object of class `R6ClassGenerator` of length 24.

`cqp` *Tools for CQP queries.*

Description

Test whether a character string is a CQP query, or turn a character vector into CQP queries.

Usage

```
is.cqp(query)

as.cqp(query, normalise.case = FALSE, collapse = FALSE)
```

Arguments

`query`	character vector with at least one query
`normalise.case`	logical
`collapse`	logical, whether to collapse the queries into one

Details

The `is.cqp` function guesses whether `query` is a CQP query and returns the respective logical value (TRUE/FALSE).

The `as.cqp` function takes a character vector as input and converts it to a CQP query by putting the individual strings in quotation marks.

Value

`is.cqp` returns a logical value, `as.cqp` a character vector

References

CQP Query Language Tutorial (http://cwb.sourceforge.net/files/CQP_Tutorial.pdf)

Examples

```
is.cqp("migration") # will return FALSE
is.cqp('"migration"') # will return TRUE
is.cqp('[pos = "ADJA"] "migration"') # will return TRUE

as.cqp("migration")
as.cqp(c("migration", "diversity"))
as.cqp(c("migration", "diversity"), collapse = TRUE)
as.cqp("migration", normalise.case = TRUE)
```

`cqpserver` *start CQP server*

Description

The function will start the CQP server by way of a system call to cqpserver.

Usage

```
startServer(registryDir = Sys.getenv("CORPUS_REGISTRY"),
  initFile = system.file("init", "cqpserver.init", package = "cqi"),
  debugMode = TRUE, exec = TRUE)
```

Arguments

`registryDir`	path to the registry directory
`initFile`	path to the init file required by cqpserver
`debugMode`	logical, whether to run debug mode
`exec`	logical, whether to start the server right away, or return a command that can be run in the shall

---

`decode` *Decode Structural Attribute or Entire Corpus.*

Description

If a `s_attribute` is a character vector providing one or several structural attributes, the return value is a `data.table` with the left and right corpus positions in the first and second columns ("cpos_left" and "cpos_right"). Values of further columns are the decoded s-attributes. The name of the s-attribute is the column name. An error is thrown if the lengths of structural attributes differ (i.e. if there is a nested data structure).

Usage

```
decode(.Object, ...)

## S4 method for signature 'character'
decode(.Object, s_attribute = NULL, verbose = TRUE,
  ...)
```

Arguments

.Object	the corpus to decode (character vector)
...	further parameters
s_attribute	the s-attribute to decode
verbose	logical

Details

If s_attribute is NULL, the token stream is decoded for all positional attributes that are present. Structural attributes are reported in additional columns. Decoding the entire corpus may be useful to make a transition to processing data following the 'tidy' approach, or to manipulate the corpus data and to re-encode the corpus.

The return value is a data.table.

Value

a data.table

Examples

```
use("polmineR")

# Scenario 1: Decode one or two s-attributes
dt <- decode("GERMAPARLMINI", s_attribute = "date")
dt <- decode("GERMAPARLMINI", s_attribute = c("date", "speaker"))

# Scenario 2: Decode corpus entirely
dt <- decode("GERMAPARLMINI")
```

---

dispersion *Dispersion of a query or multiple queries*

Description

The function returns the frequencies of a query or a multiple queries in sub-partitions defined by one or two dimensions. This is a wrapper function, so the output will depend on the number of queries and dimensions provided.

Usage

```
dispersion(.Object, ...)

## S4 method for signature 'partition'
dispersion(.Object, query, s_attribute, cqp = FALSE,
  p_attribute = getOption("polmineR.p_attribute"), freq = FALSE,
  mc = FALSE, progress = TRUE, verbose = FALSE, ...)

## S4 method for signature 'character'
dispersion(.Object, query, s_attribute, cqp = is.cqp,
  p_attribute = getOption("polmineR.p_attribute"), freq = FALSE,
  mc = FALSE, progress = TRUE, verbose = TRUE, ...)

## S4 method for signature 'hits'
dispersion(.Object, s_attribute, freq = FALSE,
  verbose = TRUE, ...)
```

Arguments

.Object	a partition object
...	further parameters
query	a character vector containing one or multiple queries
s_attribute	a character vector of length 1 or 2 providing the s-attributes
cqp	if logical, whether the query is a CQP query (TRUE/FALSE), if it is a function that is passed in, the function will be applied to the query to guess whether query is a CQP query
p_attribute	the p-attribute that will be looked up, typically 'word' or 'lemma'
freq	logical, whether to calculate normalized frequencies
mc	logical, whether to use multicore
progress	logical, whether to shop progress
verbose	logical, whether to be verbose

Value

depends on the input, as this is a wrapper function

Author(s)

Andreas Blaette

See Also

crosstab-class

count

Examples

```
use("polmineR")
test <- partition("GERMAPARLMINI", date = ".*", p_attribute = NULL, regex = TRUE)
integration <- dispersion(
  test, query = "Integration",
  p_attribute = "word", s_attribute = "date"
)
integration <- dispersion(test, "Integration", s_attribute = c("date", "party"))
integration <- dispersion(test, '"Integration.*"', s_attribute = "date", cqp = TRUE)
```

`dotplot` *dotplot*

Description

dotplot

Usage

```
dotplot(.Object, ...)

## S4 method for signature 'textstat'
dotplot(.Object, col, n = 20L, ...)

## S4 method for signature 'features'
dotplot(.Object, col = NULL, n = 20L, ...)

## S4 method for signature 'features_ngrams'
dotplot(.Object, col = NULL, n = 20L, ...)

## S4 method for signature 'partition'
dotplot(.Object, col = "count", n = 20L, ...)
```

Arguments

`.Object`	object
`...`	further arguments that will be passed into the `dotchart` function
`col`	column
`n`	number

encoding *Get and set encoding.*

Description

Method for `textstat` objects and classes inheriting from `textstat`.

Usage

```
encoding(object)

encoding(object) <- value

## S4 method for signature 'textstat'
encoding(object)

## S4 method for signature 'bundle'
encoding(object)
```

Arguments

`object`	the object with an 'encoding'-slot
`value`	value to be assigned

encodings *Conversion between corpus and native encoding.*

Description

Utility functions to convert encoding between the native encoding and the encoding of the corpus.

Usage

```
as.utf8(x, from)

as.nativeEnc(x, from)

as.corpusEnc(x, from = localeToCharset()[1], corpusEnc)
```

Arguments

`x`	the object (a character vector)
`from`	encoding of the input character vector
`corpusEnc`	encoding of the corpus (e.g. "latin1", "UTF-8")

Details

The encoding of a corpus and the encoding of the terminal (the native encoding) may differ and evoke strange output, or wrong results if no conversion is carried out between the potentially differing encodings. The functions `as.nativeEnc` and `as.corpusEnc` are auxiliary functions to assist this. The functions `as.nativeEnc` and `as.utf8` deliberately remove the explicit statement of the encoding, to avoid warnings that may occur with character vector columns in a `data.table` object.

---

enrich *Enrich an object.*

Description

Methods to enrich objects with additional (statistical) information. The methods are documented with the classes to which they adhere. See the references in the `seealso`-section.

Usage

```
enrich(.Object, ...)
```

Arguments

`.Object`	a `partition`, `partition_bundle` or comp object
`...`	further parameters

See Also

The enrich method is defined for the following classes: "partition", (see `partition_class`), "partition_bundle" (see `partition_bundle-class`), "kwic" (see `kwic-class`), and "context" (see `context-class`). See the linked documentation to learn how the enrich method can be applied to respective objects.

---

features,partition-method *Get features by comparison.*

Description

The features of two objects, usually a `partition` defining a corpus of interest (coi), and a `partition` defining a reference corpus (ref) are compared. The most important purpose is term extraction.

Usage

```
## S4 method for signature 'partition'
features(x, y, included = FALSE, method = "chisquare",
  verbose = FALSE)

## S4 method for signature 'count'
features(x, y, by = NULL, included = FALSE,
  method = "chisquare", verbose = TRUE)

## S4 method for signature 'partition_bundle'
features(x, y, included = FALSE,
  method = "chisquare", verbose = TRUE, mc = getOption("polmineR.mc"),
  progress = FALSE)

## S4 method for signature 'ngrams'
features(x, y, included = FALSE, method = "chisquare",
  verbose = TRUE, ...)
```

Arguments

x	A partition or partition_bundle object.
y	A partition object, it is assumed that the coi is a subcorpus of ref
included	TRUE if coi is part of ref, defaults to FALSE
method	the statistical test to apply (chisquare or log likelihood)
verbose	A logical value, defaults to TRUE
by	the columns used for merging, if NULL (default), the p-attribute of x will be used
mc	logical, whether to use multicore
progress	logical
...	further parameters

Author(s)

Andreas Blaette

References

Baker, Paul (2006): *Using Corpora in Discourse Analysis*. London: continuum, p. 121-149 (ch. 6).

Manning, Christopher D.; Schuetze, Hinrich (1999): *Foundations of Statistical Natural Language Processing*. MIT Press: Cambridge, Mass., pp. 151-189 (ch. 5).

Examples

```
use("polmineR")

kauder <- partition(
```

```
  "GERMAPARLMINI",
  speaker = "Volker Kauder", interjection = "speech",
  p_attribute = "word"
  )
all <- partition("GERMAPARLMINI", interjection = "speech", p_attribute = "word")

terms_kauder <- features(x = kauder, y = all, included = TRUE)
top100 <- subset(terms_kauder, rank_chisquare <= 100)
head(top100)

# a different way is to compare count objects
kauder_count <- as(kauder, "count")
all_count <- as(all, "count")
terms_kauder <- features(kauder_count, all_count, included = TRUE)
top100 <- subset(terms_kauder, rank_chisquare <= 100)
head(top100)

speakers <- partition_bundle("GERMAPARLMINI", s_attribute = "speaker")
speakers <- enrich(speakers, p_attribute = "word")
speaker_terms <- features(speakers[[1:5]], all, included = TRUE, progress = TRUE)
dtm <- as.DocumentTermMatrix(speaker_terms, col = "chisquare")
```

`features-class` *Feature selection by comparison.*

Description

The `features`-method returns a `features`-object. Several `features`-objects can be combined into a `features_bundle`-object.

Usage

```
## S4 method for signature 'features'
summary(object)

## S4 method for signature 'features'
show(object)

## S4 method for signature 'features_bundle'
summary(object)

## S4 method for signature 'features'
view(.Object)
```

Arguments

`object`	A `features` or `features_bundle` object.
`.Object`	a `features` object.

Details

A set of features objects can be combined into a features_bundle. Typically, a features_bundle will result from applying the features-method on a partition_bundle. See the documentation for bundle to learn about the methods for bundle objects that are available for a features_bundle.

Slots

corpus The CWB corpus the features are derived from, a character vector of length 1.

p_attribute Object of class character.

encoding Object of class character.

corpus Object of class character.

stat Object of class data.frame.

size_coi Object of class integer.

size_ref Object of class integer.

included Object of class logical whether corpus of interest is included in reference corpus

method Object of class character statisticalTest used

call Object of class character the call that generated the object

Author(s)

Andreas Blaette

---

flatten *flatten a nested list*

Description

If you have a list of partition_bundle objects, this function will flatten the data structure and return a partition_bundle object.

Usage

```
flatten(object)
```

Arguments

object	a list (with partition_bundle objects)

Value

a partition_bundle object

getSlot *Get slot from object.*

Description

Auxiliary function to unify access to slots of S4 or R6 object.

Usage

```
getSlot(x, name)
```

Arguments

x	object to get slot from
name	name of the slot

get_template *Get and set templates.*

Description

Templates are used to format the markdown/html output of partitions. Upon loading the polmineR package, templates for corpora are loaded into the option 'polmineR.templates'.

Usage

```
get_template(.Object, ...)

## S4 method for signature 'character'
get_template(.Object)

## S4 method for signature 'partition'
get_template(.Object)

## S4 method for signature 'missing'
get_template(.Object)

set_template(.Object, ...)

## S4 method for signature 'character'
set_template(.Object)

## S4 method for signature 'missing'
set_template(.Object, verbose = FALSE)
```

Arguments

.Object	object
...	further parameters
verbose	logical, whether to be verbose

---

get_token_stream *Get Token Stream Based on Corpus Positions.*

Description

Turn regions of a corpus defined by corpus positions into the original text.

Usage

```
get_token_stream(.Object, ...)

## S4 method for signature 'numeric'
get_token_stream(.Object, corpus, p_attribute,
  encoding = NULL, collapse = NULL, beautify = TRUE, cpos = FALSE,
  cutoff = NULL, ...)

## S4 method for signature 'matrix'
get_token_stream(.Object, ...)

## S4 method for signature 'character'
get_token_stream(.Object, left = NULL, right = NULL,
  ...)

## S4 method for signature 'partition'
get_token_stream(.Object, p_attribute, collapse = NULL,
  cpos = FALSE, ...)

## S4 method for signature 'regions'
get_token_stream(.Object, p_attribute = "word", ...)
```

Arguments

.Object	an object of class matrix or partition
...	further arguments
corpus	the CWB corpus
p_attribute	the p-attribute to decode
encoding	encoding to use
collapse	character string length 1
beautify	logical, whether to adjust whitespace before and after interpunctation

cpos	logical, whether to return cpos as names of the tokens
cutoff	maximum number of tokens to be reconstructed
left	left corpus position
right	right corpus position

Examples

```
get_token_stream(0:9, corpus = "GERMAPARLMINI", p_attribute = "word")
get_token_stream(0:9, corpus = "GERMAPARLMINI", p_attribute = "word", collapse = " ")
fulltext <- get_token_stream("GERMAPARLMINI", p_attribute = "word")
```

---

get_type *Get corpus/partition type.*

Description

To generate fulltext output, different templates can be used with a behavior that depends on the type of a corpus. get_type will return the type of corpus if it is a specialized one, or NULL.

Usage

```
get_type(.Object)

## S4 method for signature 'character'
get_type(.Object)

## S4 method for signature 'Corpus'
get_type(.Object)

## S4 method for signature 'partition'
get_type(.Object)

## S4 method for signature 'partition_bundle'
get_type(.Object)
```

Arguments

.Object	A partition, partition_bundle, Corpus object, or a length-one character vector indicating a CWB corpus.

Details

When generating a partition, the corpus type will be prefixed to the class that is generated (separated by underscore). If the corpus type is not NULL, a class inheriting from the partition-class is instantiated. Note that at this time, only plpr_partition and press_partition is implemented.

Examples

```
use("polmineR")

get_type("GERMAPARLMINI")

p <- partition("GERMAPARLMINI", date = "2009-10-28")
get_type(p)
is(p)

pb <- partition_bundle("GERMAPARLMINI", s_attribute = "date")
get_type(pb)

gp <- Corpus$new("GERMAPARLMINI")
get_type(gp)

get_type("REUTERS") # returns NULL - no specialized corpus
```

`highlight` *Highlight tokens.*

Description

Highlight tokens based on exact match, a regular expression or corpus position in kwic output or html document.

Usage

```
highlight(.Object, ...)

## S4 method for signature 'character'
highlight(.Object, highlight = list())

## S4 method for signature 'html'
highlight(.Object, highlight = list())

## S4 method for signature 'kwic'
highlight(.Object, highlight = list(), regex = FALSE,
  perl = TRUE, verbose = TRUE)
```

Arguments

`.Object`	a `html` or `character` object with html, or a `kwic` object
`...`	further parameters (unused)
`highlight`	a `"list"` of character or integer vectors, the names need to provide the colors, the values of the vector the term to be matched or a corpus position
`regex`	logical, whether character vectors give regular expressions

perl	logical, whether to use perl-style regular expressions for highlighting when regex is TRUE
verbose	logical, whether to output verbose messages

Examples

```
use("polmineR")
P <- partition("REUTERS", places = "argentina")
H <- html(P)
Y <- highlight(H, list(lightgreen = "higher"))
if (interactive()) htmltools::html_print(Y)

# highlight matches for a CQP query
H2 <- highlight(
  H,
  highlight = list(yellow = cpos(hits(P, query = '"prod.*"', cqp = TRUE)))
)

# the method can be used in pipe
if (require("magrittr")){
  P %>% html() %>% highlight(list(lightgreen = "1986")) -> H
  P %>% html() %>% highlight(list(lightgreen = c("1986", "higher"))) -> H
  P %>% html() %>% highlight(list(lightgreen = 4020:4023)) -> H
}

# use highlight for kwic output
K <- kwic("REUTERS", query = "barrel")
K2 <- highlight(K, highlight = list(yellow = c("oil", "price")))
if (interactive()) K2
```

---

hits-class *Get Hits.*

Description

Get hits for a (set of) queries, optionally with s-attribute values.

Usage

```
hits(.Object, ...)

## S4 method for signature 'character'
hits(.Object, query, cqp = FALSE, s_attribute = NULL,
  p_attribute = "word", size = FALSE, freq = FALSE, mc = FALSE,
  verbose = TRUE, progress = TRUE, ...)

## S4 method for signature 'partition'
hits(.Object, query, cqp = FALSE, s_attribute = NULL,
  p_attribute = "word", size = FALSE, freq = FALSE, mc = FALSE,
```

```
  progress = FALSE, verbose = TRUE, ...)

## S4 method for signature 'partition_bundle'
hits(.Object, query, cqp = FALSE,
  p_attribute = getOption("polmineR.p_attribute"), size = TRUE,
  freq = FALSE, mc = getOption("polmineR.mc"), progress = FALSE,
  verbose = TRUE, ...)

## S4 method for signature 'hits'
sample(x, size)

## S4 method for signature 'context'
hits(.Object, s_attribute = NULL, verbose = TRUE, ...)
```

Arguments

.Object	a character, partition or partition_bundle object
...	further parameters
query	a (optionally named, see datails) character vector with one or more queries
cqp	either logical (TRUE if query is a CQP query), or a function to check whether query is a CQP query or not
s_attribute	s-attributes
p_attribute	p-attribute
size	logical - return size of subcorpus
freq	logcial - return relative frequencies
mc	logical, whether to use multicore
verbose	logical
progress	logical, whether to show progress bar
x	a hits object

Details

If the query character vector is named, the names of the query occurr in the data.table that is returned rather than the queries.

If freq is TRUE, the data.table returned in the DT-slot will deliberately include the subsets of the partition/corpus with no hits (query is NA, count is 0).

Slots

stat a "data.table"

corpus a "character" vector

query Object of class "character"

p_attribute p-attribute that has been queried

encoding encoding of the corpus

name name of the object

html *Generate html from object.*

Description

Prepare a html document to inspect the full text.

Usage

```
html(object, ...)

## S4 method for signature 'character'
html(object)

.addCharacterOffset(x)

## S4 method for signature 'partition'
html(object, meta = NULL, cpos = TRUE,
  verbose = FALSE, cutoff = NULL, charoffset = FALSE, beautify = TRUE,
  ...)

## S4 method for signature 'partition_bundle'
html(object, filename = c(), type = "debate")

## S4 method for signature 'kwic'
html(object, i, s_attribute = NULL, type = NULL,
  verbose = FALSE, ...)

## S3 method for class 'html'
print(x, ...)
```

Arguments

object	the object the fulltext output will be based on
...	further parameters that are passed into as.markdown
x	object of class html to print
meta	metadata for output, if NULL (default), the s-attributes defining a partition will be used
cpos	logical, if TRUE (default), all tokens will be wrapped by elements with id attribute indicating corpus positions
verbose	logical, whether to be verbose
cutoff	maximum number of tokens to decode from token stream, passed into as.markdown
charoffset	logical, if TRUE, character offset positions are added to elements embracing tokens
beautify	logical, if TRUE, whitespace before interpunctuation will be removed

filename	the filename
type	the partition type
i	if object is a kwic-object, the index of the concordance for which the fulltext is to be generated
s_attribute	structural attributes that will be used to define the partition where the match occurred

Details

If param charoffset is TRUE, character offset positions will be added to tags that embrace tokens. This may be useful, if exported html document is annotated with a tools that stores annotations with character offset positions.

Examples

```
use("polmineR")
P <- partition("REUTERS", places = "argentina")
H <- html(P)
if (interactive()) H # show full text in viewer pane

# html-method can be used in a pipe
if (require("magrittr")){
  H <- partition("REUTERS", places = "argentina") %>% html()
  # use html-method to get from concordance to full text
  K <- kwic("REUTERS", query = "barrels")
  H <- html(K, i = 1, s_attribute = "id")
  H <- html(K, i = 2, s_attribute = "id")
  for (i in 1:length(K)) {
    H <- html(K, i = i, s_attribute = "id")
    if (interactive()){
      show(H)
      userinput <- readline("press 'q' to quit or any other key to continue")
      if (userinput == "q") break
    }
  }
}
```

---

kwic *KWIC/concordance output.*

Description

Prepare and show concordances / keyword-in-context (kwic). The same result can be achieved by applying the kwic method on either a partition or a context object.

Usage

```
kwic(.Object, ...)

## S4 method for signature 'context'
kwic(.Object, meta = getOption("polmineR.meta"),
  cpos = TRUE, verbose = FALSE)

## S4 method for signature 'partition'
kwic(.Object, query, cqp = is.cqp,
  left = getOption("polmineR.left"), right = getOption("polmineR.right"),
  meta = getOption("polmineR.meta"), p_attribute = "word",
  s_attribute = NULL, cpos = TRUE, stoplist = NULL, positivelist = NULL,
  regex = FALSE, verbose = TRUE, ...)

## S4 method for signature 'character'
kwic(.Object, query, cqp = is.cqp,
  left = as.integer(getOption("polmineR.left")),
  right = as.integer(getOption("polmineR.right")),
  meta = getOption("polmineR.meta"), p_attribute = "word",
  s_attribute = NULL, cpos = TRUE, stoplist = NULL, positivelist = NULL,
  regex = FALSE, verbose = TRUE, ...)
```

Arguments

.Object	a partition or context object
...	further parameters to be passed
meta	metainformation to display
cpos	logical, if TRUE, the corpus positions ("cpos") if the hits will be handed over to the kwic-object that is returned
verbose	logical, whether to be talkative
query	a query, CQP-syntax can be used
cqp	either logical (TRUE if query is a CQP query), or a function to check whether query is a CQP query or not (defaults to is.query auxiliary function)
left	to the left
right	to the right
p_attribute	p-attribute, defaults to 'word'
s_attribute	if provided, the s-attribute will be used to check the boundaries of the text
stoplist	terms or ids to prevent a concordance from occurring in results
positivelist	terms or ids required for a concordance to occurr in results
regex	logical, whether stoplist/positivelist is processed as regular expression

Details

If a positivelist ist supplied, the tokens will be highlighted.

References

Baker, Paul (2006): *Using Corpora in Discourse Analysis*. London: continuum, pp. 71-93 (ch. 4).

Jockers, Matthew L. (2014): *Text Analysis with R for Students of Literature*. Cham et al: Springer, pp. 73-87 (chs. 8 & 9).

See Also

To read the whole text, see the `read`-method.

Examples

```
use("polmineR")
bt <- partition("GERMAPARLMINI", def = list(date = ".*"), regex=TRUE)
kwic(bt, "Integration")
kwic(bt, "Integration", left = 20, right = 20, meta = c("date", "speaker", "party"))
kwic(
  bt, '"Integration" [] "(Menschen|Migrant.*|Personen)"',
  left = 20, right = 20,
  meta = c("date", "speaker", "party")
)
```

---

kwic-class *kwic (S4 class)*

Description

S4 class for organizing information for concordance output

Usage

```
## S4 method for signature 'kwic'
show(object)

## S4 method for signature 'kwic'
as.data.frame(x)

## S4 method for signature 'kwic'
length(x)

## S4 method for signature 'kwic'
sample(x, size)

## S4 method for signature 'kwic'
enrich(.Object, meta = NULL, table = FALSE)

## S4 method for signature 'kwic'
view(.Object)
```

Arguments

object	an object of class kwic
x	a kwic-class object
size	integer, the subset size for sampling
.Object	a kwic object
meta	s-attributes (character vector) with metainformation
table	logical, whether to turn cpos data.table into data.frame for output

Details

The enrich method is used to generate the actual output for the kwic method. If param table is TRUE, corpus positions will be turned into a data.frame with the concordance lines. If param meta is a character vector with s-attributes, the respective s-attributes will be added as columns to the table with concordance lines.

Slots

metadata Object of class "character" keeping the s-attributes of the metadata that are to be displayed

left words to the left

right words to the right

corpus the CWB corpus

cpos the corpus positions

table Object of class data.frame a table with the relevant information for kwic output

encoding Object of class character encoding of the corpus

labels Object of class character

categories Object of class character

Methods

[indexing for seeing only some concordances

show get kwic output

Examples

```
use("polmineR")
K <- kwic("GERMAPARLMINI", "Integration")
length(K)
K[1]
K[1:5]
```

label *Assign and get labels.*

Description

Assign and get labels.

Usage

```
label(x, ...)

label(x) <- value

## S4 method for signature 'kwic'
label(x, n = NULL)
```

Arguments

x	object
...	further parameters
value	length (character vector, length 1)
n	label index

Labels-class *Labels class.*

Description

Labels class.

Arguments

n	length of character vector in field labels
choices	choices to be assigned to field choices
expandable	whether choices are expandable

Fields

`labels` character vector with labels; if logical or numeric labels are intended, assign them as character vector anyway

`choices` character vector, a list of choices for labels

`expandable` whether choices may be expanded (logical)

ll *text statistics*

Description

text statistics

Usage

```
ll(.Object, ...)

## S4 method for signature 'context'
ll(.Object)

## S4 method for signature 'cooccurrences'
ll(.Object)

## S4 method for signature 'features'
ll(.Object)

pmi(.Object)

## S4 method for signature 'context'
pmi(.Object)
```

Arguments

.Object	an object
...	further parameters

mail *Mail result.*

Description

Mail a result (to yourself).

Usage

```
mail(object, ...)

## S4 method for signature 'partition'
mail(object, to = NULL,
  filename = "drillerExport.html", what = "html")
```

```
## S4 method for signature 'cooccurrences'
mail(object, to = NULL, nrow = NULL,
  fileFormat = c("csv", "xlsx"))

## S4 method for signature 'features'
mail(object, to = NULL, nrow = NULL,
  fileFormat = c("csv", "xlsx"))

## S4 method for signature 'kwic'
mail(object, to = NULL, nrow = NULL,
  fileFormat = c("csv", "xlsx"))

## S4 method for signature 'data.frame'
mail(object, to = NULL, nrow = NULL,
  fileFormat = c("csv", "xlsx"))
```

Arguments

object	object to deliver
...	further parameters
to	the receiver of the mail message
filename	filename
what	what to send (defaults to "html")
nrow	the number of rows of the table (if NULL, the whole table will be sent)
fileFormat	csv or xlsx, or both

---

means *calculate means*

Description

calculate means

Usage

```
means(.Object, ...)

## S4 method for signature 'DocumentTermMatrix'
means(.Object, dim = 1)
```

Arguments

.Object	object to work on
...	further parameters @exportMethod means
dim	numeric, 1 or 2 whether to work on rows or columns

ngrams-class *Get N-Grams*

Description

Count n-grams, either of words, or of characters.

Usage

```
ngrams(.Object, ...)

## S4 method for signature 'partition'
ngrams(.Object, n = 2, p_attribute = "word",
  char = NULL, progress = FALSE, ...)

## S4 method for signature 'partition_bundle'
ngrams(.Object, n = 2, char = NULL,
  p_attribute = "word", mc = FALSE, progress = FALSE, ...)
```

Arguments

.Object	object of class partition
...	further parameters
n	number of tokens/characters
p_attribute	the p-attribute to use (can be > 1)
char	if NULL, tokens will be counted, else characters, keeping only those provided by a character vector
progress	logical
mc	logical, whether to use multicore, passed into call to blapply (see respective documentation)

Examples

```
use("polmineR")
P <- partition("GERMAPARLMINI", date = "2009-10-27")
ngramObject <- ngrams(P, n = 2, p_attribute = "word", char = NULL)

# a more complex scenario: get most frequent ADJA/NN-combinations
ngramObject <- ngrams(P, n = 2, p_attribute = c("word", "pos"), char = NULL)
ngramObject2 <- subset(
 ngramObject,
 ngramObject[["1_pos"]] == "ADJA"  & ngramObject[["2_pos"]] == "NN"
 )
ngramObject2@stat[, "1_pos" := NULL, with = FALSE][, "2_pos" := NULL, with = FALSE]
ngramObject3 <- sort(ngramObject2, by = "count")
head(ngramObject3)
```

noise *detect noise*

Description

detect noise

Usage

```
noise(.Object, ...)

## S4 method for signature 'DocumentTermMatrix'
noise(.Object, minTotal = 2,
  minTfIdfMean = 0.005, sparse = 0.995, stopwordsLanguage = "german",
  minNchar = 2, specialChars = getOption("polmineR.specialChars"),
  numbers = "^[0-9\\.,]+$", verbose = TRUE)

## S4 method for signature 'TermDocumentMatrix'
noise(.Object, ...)

## S4 method for signature 'character'
noise(.Object, stopwordsLanguage = "german",
  minNchar = 2, specialChars = getOption("polmineR.specialChars"),
  numbers = "^[0-9\\.,]+$", verbose = TRUE)

## S4 method for signature 'textstat'
noise(.Object, p_attribute, ...)
```

Arguments

.Object	an .Object of class "DocumentTermMatrix"
...	further parameters
minTotal	minimum colsum (for DocumentTermMatrix) to qualify a term as non-noise
minTfIdfMean	minimum mean value for tf-idf to qualify a term as non-noise
sparse	will be passed into "removeSparseTerms" from "tm"-package
stopwordsLanguage	e.g. "german", to get stopwords defined in the tm package
minNchar	min char length ti qualify a term as non-noise
specialChars	special characters to drop
numbers	regex, to drop numbers
verbose	logical
p_attribute	relevant if applied to a textstat object

Value

a list

partition *Initialize a partition.*

Description

Create a subcorpus stored in an object of the partition class. Counts are performed for the p-attribute defined by the parameter p_attribute.

Usage

```
partition(.Object, ...)

## S4 method for signature 'character'
partition(.Object, def = NULL, name = "",
  encoding = NULL, p_attribute = NULL, regex = FALSE, xml = "flat",
  decode = TRUE, type = get_type(.Object), mc = FALSE, verbose = TRUE,
  ...)

## S4 method for signature 'list'
partition(.Object, ...)

## S4 method for signature 'environment'
partition(.Object, slots = c("name", "corpus", "size",
  "p_attribute"))

## S4 method for signature 'partition'
partition(.Object, def = NULL, name = "",
  regex = FALSE, p_attribute = NULL, decode = TRUE, xml = NULL,
  verbose = TRUE, mc = FALSE, ...)

## S4 method for signature 'Corpus'
partition(.Object, def = NULL, name = "",
  encoding = NULL, regex = FALSE, xml = "flat",
  type = get_type(.Object), verbose = TRUE, ...)
```

Arguments

.Object	character-vector - the CWB-corpus to be used
...	parameters passed into the partition-method
def	list consisting of a set of character vectors (see details and examples)
name	name of the new partition, defaults to "
encoding	encoding of the corpus (typically "LATIN1 or "(UTF-8)), if NULL, the encoding provided in the registry file of the corpus (charset="...") will be used b
p_attribute	the p_attribute(s) for which term frequencies shall be retrieved
regex	logical (defaults to FALSE)

xml	either 'flat' (default) or 'nested'
decode	whether to turn token ids to strings (set FALSE to minimize object.size / memory consumption)
type	character vector (length 1) specifying the type of corpus / partition (e.g. "plpr")
mc	whether to use multicore (for counting terms)
verbose	logical, defaults to TRUE
slots	character vector

Details

The function sets up a partition (subcorpus) based on a list of s-attributes with respective values.

The s-attributes defining the partition can be passed in as a list, e.g. list(interjection="speech", year="2013"), or - for convencience - directly.

The values defining the partition may contain regular expressions. To use regular expression syntax, set the parameter regex to "TRUE". Regular expressions are passed into grep, i.e. the regex syntax used in R needs to be used (double backlashes etc.). If regular expressions are used, the length of the character vector needs to be 1. If regex is "FALSE", the length of the character vectors can be > 1, matching s-attributes are identifies with the operator in.

The XML imported into the CWB may be "flat" or "nested". This needs to be indicated with the parameter xml (default is "flat"). If you generate a partition based on a flat XML structure, some performance gain may be achieved when ordering the s-attributes with decreasingly restrictive conditions. If you have a nested XML, it is mandatory that the order of the s-attributes provided reflects the hierarchy of the XML: The top-level elements need to be positioned at the beginning of the list with the s-attributes, the the most restrictive elements at the end.

If p_attribute is not NULL, a count of tokens in the corpus will be performed and kept in the stat-slot of the partition-object. The length of the p_attribute character vector may be 1 or more. If two or more p-attributes are provided, The occurrence of combinations will be counted. A typical scenario is to combine the p-attributes "word" or "lemma" and "pos".

Value

An object of the S4 class 'partition'

Author(s)

Andreas Blaette

See Also

To learn about the methods available for objects of the class partition, see partition_class,

Examples

```
use("polmineR")
spd <- partition("GERMAPARLMINI", party = "SPD", interjection = "speech")
kauder <- partition("GERMAPARLMINI", speaker = "Volker Kauder", p_attribute = "word")
merkel <- partition("GERMAPARLMINI", speaker = ".*Merkel", p_attribute = "word", regex = TRUE)
```

```
s_attributes(merkel, "date")
s_attributes(merkel, "speaker")
merkel <- partition(
  "GERMAPARLMINI", speaker = "Angela Dorothea Merkel",
  date = "2009-11-10", interjection = "speech", p_attribute = "word"
  )
merkel <- subset(merkel, !word %in% punctuation)
merkel <- subset(merkel, !word %in% tm::stopwords("de"))

# a certain defined time segment
days <- seq(
  from = as.Date("2009-10-28"),
  to = as.Date("2009-11-11"),
  by = "1 day"
)
period <- partition("GERMAPARLMINI", date = days)
```

---

partition_bundle *Generate bundle of partitions.*

Description

Use partition_bundle to create a partition_bundle object, which combines a set of partition objects.

Usage

```
partition_bundle(.Object, ...)

## S4 method for signature 'partition'
partition_bundle(.Object, s_attribute, values = NULL,
  prefix = "", mc = getOption("polmineR.mc"), verbose = TRUE,
  progress = FALSE, type = get_type(.Object), ...)

## S4 method for signature 'character'
partition_bundle(.Object, s_attribute, values = NULL,
  prefix = "", mc = getOption("polmineR.mc"), verbose = TRUE,
  progress = FALSE, xml = "flat", type = get_type(.Object), ...)

## S4 method for signature 'context'
partition_bundle(.Object, mc = getOption("polmineR.mc"),
  verbose = FALSE, progress = TRUE)

## S4 method for signature 'partition_bundle'
partition_bundle(.Object, s_attribute,
  prefix = character(), progress = TRUE, mc = getOption("polmineR.mc"))
```

Arguments

.Object	A partition, a length-one character vector supplying a CWB corpus, or a partition_bundle
...	parameters to be passed into partition-method (see respective documentation)
s_attribute	The s-attribute to vary
values	Values the s-attribute provided shall assume.
prefix	A character vector that will be attached as a prefix to partition names.
mc	Logical, whether to use multicore parallelization.
verbose	Logical, whether to provide progress information.
progress	Logical, whether to show progress bar.
type	The type of partition to generate.
xml	logical

Details

Applying the partition_bundle-method to a partition_bundle-object will iterate through the partition objects in the object-slot in the partition_bundle, and apply partition_bundle on each partition, splitting it up by the s-attribute provided by the argument s_attribute. The return value is a partition_bundle, the names of which will be the names of the incoming partition_bundle concatenated with the s-attribute values used for splitting. The argument prefix can be used to achieve a more descriptive name.

Value

S4 class partition_bundle, with list of partition objects in slot 'objects'

Author(s)

Andreas Blaette

See Also

partition and bundle

Examples

```
use("polmineR")
bt2009 <- partition("GERMAPARLMINI", date = "2009-.*", regex = TRUE)
pb <- partition_bundle(bt2009, s_attribute = "date", progress = TRUE, p_attribute = "word")
dtm <- as.DocumentTermMatrix(pb, col = "count")
summary(pb)
pb <- partition_bundle("GERMAPARLMINI", s_attribute = "date")
# split up objects in partition_bundle by using partition_bundle-method
use("polmineR")
pb <- partition_bundle("GERMAPARLMINI", s_attribute = "date")
pb2 <- partition_bundle(pb, s_attribute = "speaker", progress = FALSE)

summary(pb2)
```

partition_bundle-class *Bundle of partitions (partition_bundle class).*

Description

Class and methods to manage bundles of partitions.

Usage

```
## S4 method for signature 'partition_bundle'
show(object)

## S4 method for signature 'partition_bundle'
summary(object)

## S4 method for signature 'partition_bundle'
merge(x, name = "", verbose = TRUE)

## S4 method for signature 'partition_bundle,ANY,ANY,ANY'
x[i]

## S4 method for signature 'partition_bundle'
barplot(height, ...)

## S4 method for signature 'list'
as.partition_bundle(.Object, ...)

## S4 method for signature 'environment'
partition_bundle(.Object)

## S4 method for signature 'partition_bundle'
enrich(.Object, mc = FALSE, progress = TRUE,
  verbose = FALSE, ...)

## S4 method for signature 'partition_bundle'
s_attributes(.Object, s_attribute, ...)
```

Arguments

object	a partition_bundle object
x	a partition_bundle object
name	the name for the new partition
verbose	logical
i	integer index

height	height
...	further parameters
.Object	a partition_bundle object
mc	logical or, if numeric, providing the number of cores
progress	logical
s_attribute	the s-attribute to use

Details

The merge-method aggregates several partitions into one partition. The prerequisite for this function to work properly is that there are no overlaps of the different partitions that are to be summarized. Encodings and the root node need to be identical, too.

Value

An object of the class 'partition. See partition for the details on the class.

Slots

objects Object of class list the partitions making up the bundle

corpus Object of class character the CWB corpus the partition is based on

s_attributes_fixed Object of class list fixed s-attributes

encoding Object of class character encoding of the corpus

explanation Object of class character an explanation of the partition

xml Object of class character whether the xml is flat or nested

call Object of class character the call that generated the partition_bundle

Author(s)

Andreas Blaette

---

partition_class *Partition class and methods.*

Description

The partition class is used to manage subcorpora. It is an S4 class, and a set of methods is defined for the class. The class inherits from the classes count and textstat.

Usage

```
## S4 method for signature 'partition'
p_attributes(.Object, p_attribute = NULL, ...)

is.partition(x)

## S4 method for signature 'partition'
enrich(.Object, p_attribute = NULL, decode = TRUE,
  verbose = TRUE, mc = FALSE, ...)
```

Arguments

.Object	a partition object
p_attribute	a p-attribute (for enriching) / performing count.
...	further parameters passed into count when calling enrich, and ...
x	a partition object
decode	logical value, whether to decode token ids into strings when performing count
verbose	logical value, whether to output messages
mc	logical or, if numeric, providing the number of cores

Details

As partition objects inherit from count and textstat class, methods available are view to inspect the table in the stat slot, name and name<- to retrieve/set the name of an object, and more.

The p_attributes-method returns the p-attributes defined for the corpus the partition is derived from, if argument p_attribute is NULL (the default). If p_attribute is defined, the unique values for the p-attribute are returned.

The is.partition function returns a logical value whether x is a partition, or not.

The enrich-method will add a count of tokens defined by argument p_attribute to slot stat of the partition object.

Slots

name A name to identify the object (character vector with length 1); useful when multiple partition objects are combined to a partition_bundle.

corpus The CWB indexed corpus the partition is derived from (character vector with length 1).

encoding Encoding of the corpus (character vector with length 1).

s_attributes A named list with the s-attributes specifying the partition.

explanation Object of class character, an explanation of the partition.

cpos A matrix with left and right corpus positions defining regions (two columns).

annotations Object of class list.

size Total size of the partition (integer vector, length 1).

stat An (optional) data.table with counts. If present, speeds up computation of cooccurrences, as count is already present.

`metadata` Object of class `data.frame`, metadata information.

`strucs` Object of class `intger`, the strucs defining the partition.

`p_attribute` Object of class `character` indicating the p_attribute of the count in slot `stat`.

`xml` Object of class `character`, whether the xml is flat or nested.

`s_attribute_strucs` Object of class `character` the base node

`call` Object of class `character` the call that generated the partition

Author(s)

Andreas Blaette

See Also

The `partition`-class inherits from the `textstat-class`, see respective documentation to learn more.

`polmineR-generics` *generic methods defined in the polmineR-package*

Description

The package defines a set of generic functions. This doc file only provides an overview. Please consult the documentation of the classes to learn which methods can be applied to a class of a certain type.

Usage

```
name(x)

name(x) <- value

browse(x) <- value
```

Arguments

`x`	object
`value`	value

p_attributes *Get p-attributes.*

Description

In a CWB corpus, every token has positional attributes. While s-attributes cover a range of tokens, every single token in the token stream of a corpus will have a set of positional attributes (such as part-of-speech, or lemma). The available p-attributes are returned by the p_attributes-method.

Usage

```
p_attributes(.Object, ...)

## S4 method for signature 'character'
p_attributes(.Object, p_attribute = NULL, ...)
```

Arguments

.Object	a character vector (length 1) or partition object
...	further arguments
p_attribute	p-attribute to decode

References

Stefan Evert & The OCWB Development Team, CQP Query Language Tutorial, http://cwb.sourceforge.net/files/CQP_Tutoria

Examples

```
use("polmineR")
p_attributes("GERMAPARLMINI")
```

read *Display full text.*

Description

Generate text (i.e. html) and display it in the viewer pane of RStudio for reading it. If called on a partition_bundle-object, skip through the partitions contained in the bundle.

Usage

```
read(.Object, ...)

## S4 method for signature 'partition'
read(.Object, meta = NULL, highlight = list(),
  tooltips = list(), verbose = TRUE, cpos = TRUE,
  cutoff = getOption("polmineR.cutoff"), template = get_template(.Object),
  ...)

## S4 method for signature 'partition_bundle'
read(.Object, highlight = list(), cpos = TRUE,
  ...)

## S4 method for signature 'data.table'
read(.Object, col, partition_bundle,
  highlight = list(), cpos = FALSE, ...)

## S4 method for signature 'hits'
read(.Object, def, i = NULL, ...)

## S4 method for signature 'kwic'
read(.Object, i, type = NULL)

## S4 method for signature 'regions'
read(.Object, meta = NULL)
```

Arguments

.Object	an object to be read ("partition" or "partition_bundle")
...	further parameters passed into read
meta	a character vector supplying s-attributes for the metainformation to be printed; if not stated explicitly, session settings will be used
highlight	a named list of character vectors (see details)
tooltips	a named list (names are colors, vectors are tooltips)
verbose	logical
cpos	logical, if TRUE, corpus positions will be assigned (invisibly) to a cpos tag of a html element surrounding the tokens
cutoff	maximum number of tokens to display
template	template to format output
col	column of data.table with terms to be highlighted
partition_bundle	a partition_bundle object
def	a named list used to define a partition (names are s-attributes, vectors are values of s-attributes)

i	if .Object is an object of the classes kwic or hits, the ith kwic line or hit to derive a partition to be inspected from
type	the partition type, see documentation for partition-method

Details

To prepare the html output, the method read will call html and as.markdown subsequently, the latter method being the actual worker. Consult these methods to understand how preparing the output works.

The param highlight can be used to highlight terms. It is expected to be a named list of character vectors, the names providing the colors, and the vectors the terms to be highlighted. To add tooltips, use the param tooltips.

The method read is a high-level function that calls the methods mentioned before. Results obtained through read can also be obtained through combining these methods in a pipe using the package magrittr. That may offer more flexibility, e.g. to highlight matches for CQP queries. See examples and the documentation for the different methods to learn more.

See Also

For concordances / a keword-in-context display, see kwic.

Examples

```
use("polmineR")
merkel <- partition("GERMAPARLMINI", date = "2009-11-10", speaker = "Merkel", regex = TRUE)
read(merkel, meta = c("speaker", "date"))
read(
  merkel,
  highlight = list(yellow = c("Deutschland", "Bundesrepublik"), lightgreen = "Regierung"),
  meta = c("speaker", "date")
)
```

---

regions *Regions of a CWB corpus.*

Description

A coerce-method is available to coerce a partition object to a regions object.

Usage

```
as.regions(x)

## S4 method for signature 'regions'
as.data.table(x, values = NULL)
```

Arguments

x	object of class regions
values	values to assign to a column that will be added

Slots

cpos a two-column data.table that will include a "cpos_left" and "cpos_right" column

corpus the CWB corpus (character vector length 1)

encoding the encoding of the CWB corpus (character vector length 1)

Examples

```
use("polmineR")
P <- partition("GERMAPARLMINI", date = "2009-11-12", speaker = "Jens Spahn")
R <- as.regions(P)
```

---

registry_get_name *Evaluate registry file.*

Description

Functions to extract information from a registry file describing a corpus. Several operations could be accomplished with the 'cwb-regedit' tool, the functions defined here ensure that manipulating the registry is possible without a full installation of the CWB.

Usage

```
registry_get_name(corpus, registry = Sys.getenv("CORPUS_REGISTRY"))

registry_get_id(corpus, registry = Sys.getenv("CORPUS_REGISTRY"))

registry_get_home(corpus, registry = Sys.getenv("CORPUS_REGISTRY"))

registry_get_info(corpus, registry = Sys.getenv("CORPUS_REGISTRY"))

registry_get_encoding(corpus, registry = Sys.getenv("CORPUS_REGISTRY"))

registry_get_p_attributes(corpus, registry = Sys.getenv("CORPUS_REGISTRY"))

registry_get_s_attributes(corpus, registry = Sys.getenv("CORPUS_REGISTRY"))

registry_get_properties(corpus, registry = Sys.getenv("CORPUS_REGISTRY"))
```

Arguments

corpus	name of the CWB corpus
registry	directory of the registry (defaults to CORPUS_Registry environment variable)

Details

An appendix to the 'Corpus Encoding Tutorial' (http://cwb.sourceforge.net/files/CWB_Encoding_Tutorial.pdf) includes an explanation of the registry file format.

`registry_reset` *Reset Registry Directory.*

Description

A utility function to reset the environment variable CORPUS_REGISTRY. That may be necessary if you want use a CWB corpus that is not stored in the usual place. In particular, resetting the environment variable is required if you want to use a corpus delivered in a R package,

The `polmineR` package uses a subdirectory of the per-session temporary directory as a (temporary) registry. The `registry` function will return the path to this directory.

Usage

```
registry_reset(registryDir = registry(), verbose = TRUE)

registry()
```

Arguments

`registryDir`	path to the registry directory to be used
`verbose`	logical, whether to be verbose

Details

Resetting the CORPUS_REGISTRY environment variable is also necessary for the interface to CWB corpora.

To get the path to a package that contains a CWB corpus, use `system.file` (see examples).

Value

the registry directory used before resetting CORPUS_REGISTRY

See Also

To conveniently reset registry, see `use`.

Examples

```
x <- system.file(package = "polmineR", "extdata", "cwb", "registry")
registry_reset(registryDir = x)
registry()
```

renamed *Renamed Functions*

Description

These functions have been renamed in order to have a consistent coding style that follows the snake_case convention. The "old" function still work to maintain backwards compatiblity.

Usage

```
sAttributes(...)

pAttributes(...)

getTokenStream(...)

getTerms(...)

getEncoding(...)

partitionBundle(...)

as.partitionBundle(...)

setTemplate(...)

getTemplate(...)
```

Arguments

...	argument that are passed to the renamed function

size *Get Number of Tokens.*

Description

The method will get the number of tokens in a corpus or partition, or the dispersion across one or more s-attributes.

Usage

```
size(x, ...)

## S4 method for signature 'character'
size(x, s_attribute = NULL, verbose = TRUE, ...)

## S4 method for signature 'partition'
size(x, s_attribute = NULL, ...)

## S4 method for signature 'DocumentTermMatrix'
size(x)
```

Arguments

x	object to get size(s) for
...	further arguments
s_attribute	character vector with s-attributes (one or more)
verbose	logical, whether to print messages

Details

One or more s-attributes can be provided to get the dispersion of tokens across one or more dimensions. Two or more s-attributes can lead to reasonable results only if the corpus XML is flat.

Value

an integer vector if s_attribute is NULL, a `data.table` otherweise

See Also

See `dispersion`-method for counts of hits. The `hits` method calls the `size`-method to get sizes of subcorpora.

Examples

```
use("polmineR")
size("GERMAPARLMINI")
size("GERMAPARLMINI", s_attribute = "date")
size("GERMAPARLMINI", s_attribute = c("date", "party"))

P <- partition("GERMAPARLMINI", date = "2009-11-11")
size(P, s_attribute = "speaker")
size(P, s_attribute = "party")
size(P, s_attribute = c("speaker", "party"))
```

split,partition-method

split partition into partition_bundle

Description

Split a partition object into a partition_bundle if gap between strucs exceeds a minimum number of tokens specified by 'gap'. Relevant to split up a plenary protocol into speeches. Note: To speed things up, the returned partitions will not include frequency lists. The lists can be prepared by applying enrich on the partition_bundle object that is returned.

Usage

```
## S4 method for signature 'partition'
split(x, gap, drop = FALSE, ...)
```

Arguments

x	a partition object
gap	an integer specifying the minimum gap for performing the split
drop	not yet implemented
...	further arguments

Value

a partition_bundle

subcorpus

Virtual class subcorpus

Description

The classes regions and partition can be used to define subcorpora. Unlike the regions class, the partition class may include statistical evaluations. The virtual class subcorpora is a mechanism to define methods for these classes without making regions the superclass of partition.

Usage

```
## S4 method for signature 'subcorpus'
aggregate(x)
```

Arguments

x	An object of a class belonging to the virtual class subcorpus, i.e. a partition or regions object.

Details

The method aggregate will deflate the matrix in the slot cpos, i.e. it checks for each new row in the matrix whether it increments the end of the previous region (by 1), and ensure that the cpos matrix defines disjoined regions.

Examples

```
P <- new(
  "partition",
  cpos = matrix(data = c(1:10, 20:29), ncol = 2, byrow = TRUE),
  stat = data.table::data.table()
)
P2 <- aggregate(P)
P2@cpos
```

---

s_attributes,character-method

Get s-attributes.

Description

Structural annotations (s-attributes) of a corpus provide metainformation for regions of tokens. Gain access to the s-attributes available for a corpus or partition, or the values of s-attributes in a corpus/partition with the s_attributes-method.

Usage

```
## S4 method for signature 'character'
s_attributes(.Object, s_attribute = NULL,
  unique = TRUE, regex = NULL, ...)

## S4 method for signature 'partition'
s_attributes(.Object, s_attribute = NULL,
  unique = TRUE, ...)
```

Arguments

.Object	either a partition object or a character vector specifying a CWB corpus
s_attribute	name of a specific s-attribute
unique	logical, whether to return unique values only
regex	filter return value by applying a regex
...	to maintain backward compatibility, of argument sAttribute is used

Details

Importing XML into the Corpus Workbench (CWB) turns elements and element attributes into so-called s-attributes. There are two uses of the s_attributes-method: If the `s_attribute` parameter is NULL (default), the return value is a character vector with all s-attributes present in a corpus.

If s_attribute is the name of a specific s-attribute (a length 1 character vector), the values of the s-attributes available in the corpus/partition are returned.

If a character vector of s-attributes is provided, the method will return a `data.table`.

Value

a character vector

Examples

```
use("polmineR")

s_attributes("GERMAPARLMINI")
s_attributes("GERMAPARLMINI", "date") # dates of plenary meetings

P <- partition("GERMAPARLMINI", date = "2009-11-10")
s_attributes(P)
s_attributes(P, "speaker") # get names of speakers
```

---

tempcorpus-class *S4 class to capture core information on a temporary CWB corpus*

Description

S4 class to capture core information on a temporary CWB corpus

Based on the corpus positions defining a partition, a temporary CWB corpus is generated that is stored in a temporary directory.

Usage

```
tempcorpus(.Object, ...)
```

Arguments

`.Object`	a partition object
`...`	further parameters

Slots

`cpos` matrix with start/end corpus positions

`dir` directory where the tempcorpus is stored

`registry` directory of the registry dir (subdirectory of dir)

`indexed` directory of the dir with the indexed files

TermDocumentMatrix *Methods for TermDocumentMatrix / DocumentTermMatrix*

Description

Methods for TermDocumentMatrix / DocumentTermMatrix

Usage

```
## S4 method for signature 'TermDocumentMatrix'
size(x)
```

Arguments

x	object

terms *Get terms in* partition *or corpus.*

Description

Get terms in partition or corpus.

Usage

```
## S4 method for signature 'partition'
terms(x, p_attribute, regex = NULL, ...)

## S4 method for signature 'character'
terms(x, p_attribute, regex = NULL, robust = FALSE,
  ...)
```

Arguments

x	an atomic character vector with a corpus id or partition object
p_attribute	the p-attribute to be analyzed
regex	regular expression(s) to filter results
...	for backward compatibility
robust	logical, whether to check for potential failures

Examples

```
use("polmineR")
session <- partition("GERMAPARLMINI", date = "2009-10-27")
words <- terms(session, "word")
terms(session, p_attribute = "word", regex = "^Arbeit.*")
terms(session, p_attribute = "word", regex = c("Arbeit.*", ".*arbeit"))

terms("GERMAPARLMINI", p_attribute = "word")
terms("GERMAPARLMINI", p_attribute = "word", regex = "^Arbeit.*")
```

---

textstat-class *S4 textstat class*

Description

Superclass for features, context, and partition class.

Usage

```
## S4 method for signature 'textstat'
name(x)

## S4 replacement method for signature 'textstat,character'
name(x) <- value

## S4 method for signature 'textstat'
head(x, ...)

## S4 method for signature 'textstat'
tail(x, ...)

## S4 method for signature 'textstat'
dim(x)

## S4 method for signature 'textstat'
nrow(x)

## S4 method for signature 'textstat'
round(x, digits = 2)

## S4 method for signature 'textstat'
colnames(x)

## S4 method for signature 'textstat'
sort(x, by, decreasing = TRUE)

as.bundle(object, ...)
```

```
## S4 method for signature 'textstat,textstat'
e1 + e2

## S4 method for signature 'textstat'
subset(x, ...)

## S4 method for signature 'textstat'
as.data.table(x)

## S4 method for signature 'textstat'
as.data.frame(x)

## S4 method for signature 'textstat'
p_attributes(.Object)

## S4 method for signature 'textstat'
x[[i]]

## S4 method for signature 'textstat,ANY,ANY,ANY'
x[i, j]

## S4 method for signature 'textstat'
view(.Object)
```

Arguments

x	textstat object
value	A character vector to assign as name to slot name of a textstat class object.
...	further parameters
digits	no of digits
by	by
decreasing	logical
object	a textstat object
e1	object 1
e2	object 2
.Object	an object
i	vector to index data.table in stat-slot
j	vector to index data.table in stat-slot

Details

Objects derived from the textstat class can be indexed with simple square brackets ("[") to get rows specified by an numeric/integer vector, and with double square brackets ("[[") to get specific columns from the data.table in the slot stat.

textstat objects can have a name, which can be retrieved, and set using the name-method and name<-, respectively.

Slots

p_attribute Object of class "character" p-attribute of the query

corpus Object of class "character"

stat Object of class "data.table" statistics of the analysis

name name of the object

encoding Object of class "character" encoding of the corpus

Examples

```
use("polmineR")
P <- partition("GERMAPARLMINI", date = ".*", p_attribute = "word", regex = TRUE)
y <- cooccurrences(P, query = "Arbeit")
y[1:25]
y[,c("word", "ll")]
y[1:25, "word"]
y[1:25][["word"]]
y[which(y[["word"]] %in% c("Arbeit", "Sozial"))]
y[ y[["word"]] %in% c("Arbeit", "Sozial") ]
```

---

tooltips *Add tooltips to html document.*

Description

Highlight tokens based on exact match, a regular expression or corpus position in kwic output or html document.

Usage

```
tooltips(.Object, tooltips)

## S4 method for signature 'character'
tooltips(.Object, tooltips = list())

## S4 method for signature 'html'
tooltips(.Object, tooltips = list())
```

Arguments

.Object	a html or character object with html
tooltips	a named "list" of character vectors (length 1), the names need to match colors in the list provided to param highlight, the value of the character vector is the tooltip to be displayed

Examples

```
use("polmineR")
P <- partition("REUTERS", places = "argentina")
H <- html(P)
Y <- highlight(H, list(lightgreen = "higher"))
T <- tooltips(Y, list(lightgreen = "Further information"))
T

if (require("magrittr")){
  P %>%
    html() %>%
    highlight(list(yellow = c("barrels", "oil", "gas"))) %>%
    tooltips(list(yellow = "energy"))
}
```

---

trim *trim an object*

Description

Method to trim and adjust objects by applying thresholds, minimum frequencies etc. It can be applied to context, features, context, partition and partition_bundle objects.

Usage

```
trim(object, ...)

## S4 method for signature 'TermDocumentMatrix'
trim(object, termsToKeep = NULL,
  termsToDrop = NULL, docsToKeep = NULL, docsToDrop = NULL,
  verbose = TRUE)

## S4 method for signature 'DocumentTermMatrix'
trim(object, ...)

punctuation
```

Arguments

object	the object to be trimmed
...	further arguments
termsToKeep	...
termsToDrop	...
docsToKeep	...
docsToDrop	...
verbose	logical

Format

An object of class `character` of length 13.

Author(s)

Andreas Blaette

---

t_test *perform t-test*

Description

S4 method for context object to perform t-test

Usage

```
t_test(.Object)

## S4 method for signature 'context'
t_test(.Object)
```

Arguments

`.Object`	a context or features object

---

use *Use a packaged corpus.*

Description

Use a CWB corpus shipped in a package, or reset registry directory.

Usage

```
use(pkg = NULL, lib.loc = .libPaths(),
  dir = getOption("polmineR.defaultRegistry"), verbose = TRUE)
```

Arguments

`pkg`	package with a CWB indexed corpus to use (defaults to NULL)
`lib.loc`	a character vector with path names of `R` libraries
`dir`	a registry directory, defaults to `getOption("polmineR.defaultRegistry")`
`verbose`	logical, whether to output status messages

Details

If pkg is the name of a data package with a CWB indexed corpus, the function will reset the CORPUS_REGISTRY environment variable and re-direct the interfacing libraries (RcppCWB or rcqp) to the registry directory in the package. The registry directory is assumed to be the ./extdata/cwb/registry subdirectory of the installed package.

If pkg is NULL (default), calling use will reset the registry directory to the directory defined by dir (defaults to the option polmineR.defaultRegistry, to return to the registry that was used when loading polmineR).

When resetting the registry directory, templates for formatting fulltext output are reloaded.

Value

the function returns invisibly the registry that was previously set

See Also

the worker to reset the registry is registry_reset

Examples

```
# to get the registry directory of the sample data in the polmineR package
system.file(package = "polmineR", "extdata", "cwb", "registry")

use("polmineR")
```

---

view *browse an object using View()*

Description

browse an object using View()

Usage

```
view(.Object, ...)
```

Arguments

.Object	an object
...	further parameters

weigh *Apply Weight to Matrix*

Description

Apply Weight to Matrix

Usage

```
weigh(.Object, ...)

## S4 method for signature 'TermDocumentMatrix'
weigh(.Object, method = "tfidf")

## S4 method for signature 'DocumentTermMatrix'
weigh(.Object, method = "tfidf")
```

Arguments

.Object	the matrix to be weighed
...	further parameters
method	the kind of weight to apply

Index