

# Package ‘polmineR’

May 15, 2018

**Type** Package

**Title** Toolkit for Corpus Analysis

**Version** 0.7.8

**Date** 2018-05-15

**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' 'CQI.cqpserver.R' 'Corpus\_class.R' 'Labels.R'  
'TermDocumentMatrix\_methods.R' 'TokenStream\_class.R'  
'adjustEncoding\_method.R' 'generics.R' 'textstat\_class.R'  
'polmineR\_package.R' 'count\_class.R' 'partition\_class.R'  
'aggregate\_method.R' 'bundle\_class.R' 'features\_class.R'  
'context\_class.R' 'partitionBundle\_class.R'  
'as.DocumentTermMatrix\_method.R' 'as.VCorpus\_method.R'  
'as.markdown\_method.R' 'contextBundle\_class.R'  
'cooccurrences\_class.R' 'as.sparseMatrix\_method.R'

'as.speeches\_method.R' 'blapply\_method.R' 'kwic\_method.R'  
 'kwic\_class.R' 'browse\_method.R' 'chisquare\_method.R'  
 'context\_method.R' 'cooccurrences\_method.R' 'corpus\_method.R'  
 'count\_method.R' 'hits\_class.R' 'tempcorpus.R' 'cpos\_method.R'  
 'decode\_method.R' 'dispersion\_method.R' 'dotplot\_method.R'  
 'encoding\_method.R' 'enrich\_method.R' 'ngrams\_method.R'  
 'features\_method.R' 'getEncoding\_method.R' 'getTerms\_method.R'  
 'regions\_class.R' 'getTokenStream\_method.R'  
 'highlight\_method.R' 'html\_method.R' 'html\_methods.R'  
 'label\_method.R' 'll\_method.R' 'mail\_method.R' 'means\_method.R'  
 'noise\_method.R' 'pAttributes\_method.R'  
 'partitionBundle\_method.R' 'sAttributes2cpos\_method.R'  
 'partition\_method.R' 'plot\_method.R' 'pmi\_method.R'  
 'read\_method.R' 'registry.R' 'reindex\_method.R'  
 'sAttributes\_method.R' 'size\_method.R' 'tTest.R'  
 'templates\_method.R' 'terms\_method.R' 'tooltips\_method.R'  
 'trim\_method.R' 'use\_function.R' 'utils.R' 'view\_method.R'  
 'weigh\_method.R' 'zzz.R'

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Author** Andreas Blaette [aut, cre]

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

**Repository** CRAN

**Date/Publication** 2018-05-15 20:24:41 UTC

## R topics documented:

polmineR-package . . . . .	4
as.markdown . . . . .	5
as.sparseMatrix . . . . .	6
as.speeches . . . . .	6
as.TermDocumentMatrix . . . . .	7
as.VCorpus.partitionBundle-method . . . . .	9
blapply . . . . .	10
browse . . . . .	11
bundle-class . . . . .	12
chisquare . . . . .	14
context . . . . .	14
context-class . . . . .	16
contextBundle-class . . . . .	18
cooccurrences . . . . .	18
cooccurrences-class . . . . .	20
cooccurrencesReshaped . . . . .	21
Corpus . . . . .	21
corpus . . . . .	23
count . . . . .	23

count_class . . . . .	25
cpos . . . . .	26
CQI.super . . . . .	27
cqp . . . . .	28
cqpserver . . . . .	29
decode . . . . .	29
dispersion . . . . .	30
dotplot . . . . .	32
encoding . . . . .	33
encodings . . . . .	33
enrich . . . . .	34
features,partition-method . . . . .	34
features-class . . . . .	36
flatten . . . . .	37
getEncoding . . . . .	38
getObjects . . . . .	38
getSlot . . . . .	39
getTemplate . . . . .	39
getTerms . . . . .	40
getTokenStream . . . . .	40
highlight . . . . .	41
hits-class . . . . .	43
html . . . . .	44
kwic . . . . .	46
kwic-class . . . . .	48
label . . . . .	49
Labels-class . . . . .	50
ll . . . . .	50
mail . . . . .	51
means . . . . .	52
name . . . . .	52
ngrams-class . . . . .	53
noise . . . . .	54
partition . . . . .	55
partitionBundle . . . . .	57
partitionBundle-class . . . . .	58
partition_class . . . . .	60
pAttribute . . . . .	62
pAttributes . . . . .	63
read . . . . .	63
regions . . . . .	65
registry_get_name . . . . .	66
registry_reset . . . . .	67
sAttributes,character-method . . . . .	68
size . . . . .	69
split,partition-method . . . . .	70
tempcorpus-class . . . . .	71
TermDocumentMatrix . . . . .	71

terms . . . . .	72
textstat-class . . . . .	72
TokenStream-class . . . . .	75
tooltips . . . . .	75
trim . . . . .	76
tTest . . . . .	77
use . . . . .	77
view . . . . .	78
weigh . . . . .	79

<b>Index</b>	<b>80</b>
--------------	-----------

---

polmineR-package	<i>polmineR-package</i>
------------------	-------------------------

---

## Description

Tools for mining CWB corpora.

## Usage

polmineR()

## Details

The package provides functions for basic text statistics for corpora that are managed by the Corpus Workbench (CWB). A core feature is to generate subcorpora/partitions based on metadata. The package is also meant to serve as an interface between the CWB and R-packages implementing more sophisticated statistical procedures (e.g. lsa, lda, topicmodels) or providing further functionality for text mining (e.g. tm).

Any analysis using this package will usually start with setting up a subcorpus/partition (with `partition`). A set of partitions can be generated with `partitionBundle`. Once a partition or a set of partitions has been set up, core functions are `cooccurrences` and `features`. Based on a partition bundle, a term-document matrix (class 'TermDocumentMatrix' from the tm package) can be generated (with `as.TermDocumentMatrix`). This opens the door to the wealth of statistical methods implemented in R.

When the package is loaded and attached, the package will look for a file name 'polmineR.conf' in a directory defined by the environment variable 'POLMINER\_DIR'. It will take general settings for polmineR from that file. Second, templates are restored.

## 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.

---

as.markdown	<i>Generate markdown from a partition.</i>
-------------	--

---

## Description

The method is the worker behind the html-method.

## Usage

```
as.markdown(.Object, ...)  
  
## S4 method for signature 'numeric'  
as.markdown(.Object, corpus, meta = NULL, cpos = FALSE,  
  cutoff = NULL, ...)  
  
## S4 method for signature 'plprPartition'  
as.markdown(.Object, meta = NULL,  
  template = getTemplate(.Object), cpos = FALSE, interjections = TRUE,  
  cutoff = NULL, ...)
```

## Arguments

.Object	object to be converted
...	further arguments
corpus	name of CWB corpus
meta	metainformation i.e. s-attributes) to be displayed
cpos	logical, whether to add cpos as ids in span elements
cutoff	maximum number of tokens to reconstruct
template	a template for formatting output
interjections	logical, 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	<i>Type conversion - get sparseMatrix.</i>
-----------------	--

---

### 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	<i>Split partition into speeches</i>
-------------	--------------------------------------

---

### Description

A method designed for corpora from the PolMine corpora of plenary protocols. A partition is split into speeches.

### Usage

```
## S4 method for signature 'partition'
as.speeches(.Object,
  sAttributeDates = getTemplate(.Object)[["metadata"]][["date"]],
  sAttributeNames = getTemplate(.Object)[["metadata"]][["name"]], gap = 500,
  mc = FALSE, verbose = TRUE, progress = TRUE)
```

**Arguments**

.Object	a partition .Object
sAttributeDates	the s-attribute that provides the dates of sessions
sAttributeNames	the s-attribute that provides the names of speakers
gap	number of tokens between strucs to identify speeches
mc	whether to use multicore, defaults to FALSE
verbose	logical, defaults to TRUE
progress	logical

**Value**

a partitionBundle object

**Examples**

```

use("polmineR")
bt <- partition("GERMAPARLMINI", date = ".*", regex = TRUE)
speeches <- as.speeches(bt, sAttributeDates = "date", sAttributeNames = "speaker")

# step-by-step, not the fastest way
speeches <- enrich(speeches, pAttribute = "word")
tdm <- as.TermDocumentMatrix(speeches, col = "count")

# fast option (counts performed when assembling the sparse matrix)
# tdm <- as.TermDocumentMatrix(speeches, pAttribute = "word")
# termsToDropList <- noise(tdm)
# whatToDrop <- c("stopwords", "specialChars", "numbers", "minNchar")
# termsToDrop <- unlist(lapply(whatToDrop, function(x) termsToDropList[[x]]))
# tdm <- trim(tdm, termsToDrop = termsToDrop)

```

---

as.TermDocumentMatrix *Generate TermDocumentMatrix / DocumentTermMatrix.*

---

**Description**

Method 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, pAttribute, sAttribute,
  verbose = TRUE, ...)

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

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

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

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

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

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

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

```

**Arguments**

x	a character vector indicating a corpus, or an object of class bundle, or inheriting from class bundle (e.g. partitionBundle)
...	s-attribute definitions used for subsetting the corpus, compare partition-method
pAttribute	p-attribute counting is be based on
sAttribute	s-attribute that defines content of columns, or rows
verbose	logical, 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 `sAttribute` provided. Counts are performed for the `pAttribute`. Further parameters provided (passed in `as ...`) are interpreted as `sAttributes` that define a subset of the corpus for splitting it according to `sAttribute`. If struc values for `sAttribute` 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 `partitionBundle` that does not yet include counts. In this case, a `pAttribute` 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 <- partitionBundle(p, sAttribute = "date")
pB <- enrich(pB, pAttribute="word")
tdm <- as.TermDocumentMatrix(pB, col = "count")

# leave the counting to the as.TermDocumentMatrix-method
pB2 <- partitionBundle(p, sAttribute = "date")
tdm <- as.TermDocumentMatrix(pB2, pAttribute = "word", verbose = TRUE)

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

---

as.VCorpus,partitionBundle-method

*Coerce partitionBundle to VCorpus.*

---

### Description

Coerce `partitionBundle` to `VCorpus`.

### Usage

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

**Arguments**

x                    a partitionBundle object

**Examples**

```
use("polmineR")
P <- partition("GERMAPARLMINI", date = "2009-11-10")
PB <- partitionBundle(P, sAttribute = "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 and 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 = ".x", regex=TRUE)
speeches <- as.speeches(bt, sAttributeDates = "date", sAttributeNames = "speaker")
foo <- blapply(speeches, function(x, ...) slot(x, "cpos"))

```

---

browse	<i>display in browser</i>
--------	---------------------------

---

**Description**

bla

**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 'pressPartition'
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 class to bundle several objects (partition, context, comp, cooccurrences objects) in one S4 object.

### 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, ...)
```

**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**

objects	Object of class "list"
pAttribute	Object of class "character"
encoding	encoding of objects

**Author(s)**

Andreas Blaette

**Examples**

```

parties <- sAttributes("GERMAPARLMINI", "party")
parties <- parties[-which(parties == "NA")]
party_bundle <- partitionBundle("GERMAPARLMINI", sAttribute = "party")
length(party_bundle)
names(party_bundle)
party_bundle <- enrich(party_bundle, pAttribute = "word")
summary(party_bundle)
parties_big <- party_bundle[[c("CDU_CSU", "SPD")]]
summary(parties_big)
use("polmineR")
Ps <- partitionBundle(
  "REUTERS", sAttribute = "id",
  values = sAttributes("REUTERS", "id")
)
Cs <- cooccurrences(Ps, query = "oil", cq = FALSE, verbose = FALSE, progress = TRUE)
dt <- as.data.table(Cs, col = "l1")
m <- as.matrix(Cs, col = "l1")

```

---

chisquare	<i>perform chisquare-text</i>
-----------	-------------------------------

---

**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	<i>Analyze context of a node word.</i>
---------	--

---

**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"),
  pAttribute = getOption("polmineR.pAttribute"), sAttribute = 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,
  pAttribute = getOption("polmineR.pAttribute"), sAttribute = 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 'partitionBundle'
context(.Object, query, verbose = TRUE, ...)

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

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

```

**Arguments**

<code>.Object</code>	a partition or a partitionBundle object
<code>query</code>	query, which may be a character vector or a cqpQuery object
<code>cqp</code>	defaults to <code>is.cqp</code> -function, or provide TRUE/FALSE
<code>left</code>	no of tokens and to the left of the node word
<code>right</code>	no of tokens to the right of the node word
<code>pAttribute</code>	p-attribute of the query
<code>sAttribute</code>	if provided, it will be checked that corpus positions do not extend beyond the region defined by the s-attribute
<code>stoplist</code>	exclude a query hit from analysis if stopword(s) is/are in context. See <code>positivelist</code> for further explanation.
<code>positivelist</code>	character vector or numeric/integer vector: include a query hit only if token in <code>positivelist</code> is present. If <code>positivelist</code> is a character vector, it may include regular expressions (see parameter <code>regex</code> )
<code>regex</code>	logical, defaults to FALSE - whether <code>stoplist</code> and/or <code>positivelist</code> are regular expressions
<code>count</code>	logical
<code>mc</code>	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 partitionBundle serves as input, the return will be a context object, or a contextBundle object

**Author(s)**

Andreas Blaette

**Examples**

```
use("polmineR")
p <- partition("GERMAPARLMINI", interjection = "speech")
y <- context(p, query = "Integration", pAttribute = "word")
y <- context(p, query = "Integration", pAttribute = "word", positivelist = "Bildung")
y <- context(
  p, query = "Integration", pAttribute = "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'
sample(x, size)

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

## S4 method for signature 'context'
enrich(.Object, sAttribute = NULL, pAttribute = NULL,
  decode = FALSE, verbose = TRUE)
```



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

## S4 method for signature 'context'
trim(object, sAttribute = NULL, positivelist = NULL,
      pAttribute = pAttributes(object), regex = FALSE, stoplist = NULL,
      verbose = TRUE, progress = TRUE)
```

### Arguments

x	a context object
size	integer indicating sample size
.Object	object
sAttribute	s-attribute(s) to add to data.table in cpos-slot
pAttribute	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
object	a context object
positivelist	tokens that are required to be present to keep a match
regex	logical, whether positivelist / 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
partitionSize	Object of class "numeric" 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
sAttribute	Object of class "character" s-attribute
pAttribute	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

---

contextBundle-class    *S4 contextBundle 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,
  pAttribute = getOption("polmineR.pAttribute"), sAttribute = 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"),
  pAttribute = getOption("polmineR.pAttribute"), sAttribute = 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 'partitionBundle'
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 if big=TRUE)
query	query, may be 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
pAttribute	the pAttribute of the tokens/the query
sAttribute	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 not 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 not 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, pAttribute = "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 'cooccurrencesBundle'
as.data.frame(x)
```

```
## S4 method for signature 'cooccurrencesReshaped'
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  
 partitionSize Object of class "numeric" 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  
 pAttribute Object of class "character" p-attribute of the query  
 corpus Object of class "character" the CWB corpus used  
 stat Object of class "data.frame" 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

---

cooccurrencesReshaped *Methods for manipulating cooccurrencesReshaped-class-objects*

---

**Description**

Methods for manipulating cooccurrencesReshaped-class-objects

**Arguments**

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

---

Corpus	<i>Corpus class.</i>
--------	----------------------

---

**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 `sAttributes`  
`sAttributes` a data.table with the values of a set of sAttributes  
`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  
**pAttribute** p-attribute, to perform count  
**sAttributes** s-attributes  
**decode** logical, whether to turn token ids into strings upon counting  
**as.html** logical

**Methods**

`initialize(corpus, pAttribute = NULL, sAttributes = NULL)` Initialize a new object of class `Corpus`.  
`count(pAttribute = getOption("polmineR.pAttribute"), 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$new("REUTERS", pAttribute = "word")
REUTERS$stat

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

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

---

corpus	<i>Get corpus.</i>
--------	--------------------

---

**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 partitionBundle-object, the corpus the respective object is derived from is returned.

---

count	<i>Get counts.</i>
-------	--------------------

---

**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,
      pAttribute = getOption("polmineR.pAttribute"),
```

```

mc = getOption("polmineR.cores"), verbose = TRUE, progress = FALSE)

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

## S4 method for signature 'character'
count(.Object, query = NULL, cqp = is.cqp,
      pAttribute = getOption("polmineR.pAttribute"), breakdown = FALSE,
      sort = FALSE, decode = TRUE, verbose = TRUE)

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

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

```

### Arguments

.Object	a "partition" or "partitionBundle" object, or a character vector (length 1) providing the name of a corpus
...	further parameters
query	a character vector (one or multiple terms to be looked up), 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 count occurrences for different matches for a query
decode	logical, whether to add rownames (only if query is NULL)
pAttribute	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
freq	logical, if FALSE, counts will be reported, if TRUE, frequencies
total	defaults to FALSE, if TRUE, the added value of counts (column: TOTAL) will be amended to the data.table that is returned
sort	logical, whether to sort stat
corpus	name of CWB corpus

### Details

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



If .Object is a character vector (length 1) 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"

### 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"), pAttribute = "word")

debates <- partitionBundle(
  "GERMAPARLMINI", sAttribute = "date", values = NULL,
  regex = TRUE, mc = FALSE, verbose = FALSE
)
y <- count(debates, query = "Arbeit", pAttribute = "word")
y <- count(debates, query = c("Arbeit", "Migration", "Zukunft"), pAttribute = "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.

**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	<i>Get corpus positions for a query or queries.</i>
------	---

---

**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,
      pAttribute = getOption("polmineR.pAttribute"), cqp = is.cqp,
      encoding = NULL, verbose = TRUE, ...)
```

## S4 method for signature 'partition'

```
cpos(.Object, query, cqp = is.cqp, pAttribute = 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)
pAttribute	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 be returned.

---

CQI.super

*Interfaces for accessing the CWB*


---

**Description**

The package offers two different interfaces to the Corpus Workbench (CWB): The package 'Rcp-pCWB', 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](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	<i>start CQP server</i>
-----------	-------------------------

---

**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 shell

---

decode	<i>Decode Structural Attribute or Entire Corpus.</i>
--------	--

---

**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, sAttribute = NULL, verbose = TRUE)
```

**Arguments**

<code>.Object</code>	the corpus to decode (character vector)
<code>...</code>	further parameters
<code>sAttribute</code>	the s-attribute to decode
<code>verbose</code>	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", sAttribute = "date")
dt <- decode("GERMAPARLMINI", sAttribute = 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, sAttribute, cqp = FALSE,
  pAttribute = getOption("polmineR.pAttribute"), freq = FALSE, mc = FALSE,
  progress = TRUE, verbose = FALSE)

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

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

```

**Arguments**

.Object	a partition object
...	further parameters
query	a character vector containing one or multiple queries
sAttribute	a character vector of length 1 or 2 providing the sAttributes
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
pAttribute	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 show 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 = ".*", pAttribute = NULL, regex = TRUE)
integration <- dispersion(
  test, query = "Integration",
  pAttribute = "word", sAttribute = "date"
)
integration <- dispersion(test, "Integration", sAttribute = c("date", "party"))
integration <- dispersion(test, "'Integration.*'", sAttribute = "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 'featuresNgrams'
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	<i>Get and set encoding.</i>
----------	------------------------------

---

**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	<i>Conversion between corpus and native encoding.</i>
-----------	---

---

**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.

---

<code>enrich</code>	<i>Enrich an object.</i>
---------------------	--------------------------

---

**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**

<code>.Object</code>	a partition, <code>partitionBundle</code> or comp object
<code>...</code>	further parameters

**See Also**

The `enrich` method is defined for the following classes: "partition", (see [partition\\_class](#)), "partitionBundle" (see [partitionBundle-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	<i>Get features by comparison.</i>
---------------------------	------------------------------------

---

**Description**

The features of two objects, usually a partition defining a corpus of interest, and a partition defining a reference corpus 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 'partitionBundle'
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 partitionBundle 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	logical, defaults to TRUE
by	the columns used for merging, if NULL (default), the pAttribute of x will be used
mc	logical, whether to use multicore
progress	logical
...	further parameters

**Author(s)**

Andreas Blaette

**References**

Manning / Schuetze ...

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",
  pAttribute="word"
)
all <- partition("GERMAPARLMINI", interjection = "speech", pAttribute = "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 <- partitionBundle("GERMAPARLMINI", sAttribute = "speaker")
speakers <- enrich(speakers, pAttribute = "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 (S4 class).*


---

**Description**

object resulting from features-method

**Usage**

```

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

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

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

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

```

**Arguments**

object            an object  
 .Object          an object

**Slots**

corpus Object of class "character"  
 pAttribute Object of class "character"  
 encoding Object of class "character"  
 corpus Object of class "character"  
 stat Object of class "data.frame"  
 sizeCoi Object of class "numeric"  
 sizeRef Object of class "numeric"  
 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  
 objects an object of class list

**Author(s)**

Andreas Blaette

---

flatten                      *flatten a nested list*

---

**Description**

If you have a list of partitionBundles, this function will flatten the data structure and return a partition Bundle object.

**Usage**

```
flatten(object)
```

**Arguments**

object            a list (with partitionBundle objects)

**Value**

a partitionBundle object

---

getEncoding	<i>Get the encoding of a corpus.</i>
-------------	--------------------------------------

---

**Description**

Parse the registry file and get the encoding of a corpus.

**Usage**

```
getEncoding(.Object)
```

```
## S4 method for signature 'character'  
getEncoding(.Object)
```

**Arguments**

.Object            the corpus name

---

getObjects	<i>Get objects of a certain class.</i>
------------	--

---

**Description**

Get objects of a certain class.

**Usage**

```
getObjects(class, envir = .GlobalEnv)
```

**Arguments**

class            character, class to be looked for

envir            character string, namespace to be searched

**Value**

a list with the partitions found in the namespace

---

getSlot	<i>Get slot from object.</i>
---------	------------------------------

---

**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

---

getTemplate	<i>Get and set templates.</i>
-------------	-------------------------------

---

**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**

```
getTemplate(.Object, ...)  
  
## S4 method for signature 'character'  
getTemplate(.Object)  
  
## S4 method for signature 'partition'  
getTemplate(.Object)  
  
## S4 method for signature 'missing'  
getTemplate(.Object)  
  
setTemplate(.Object, ...)  
  
## S4 method for signature 'character'  
setTemplate(.Object)  
  
## S4 method for signature 'missing'  
setTemplate(.Object, verbose = FALSE)
```

**Arguments**

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

---

getTerms	<i>get terms available in a corpus or partition</i>
----------	---

---

**Description**

get terms available in a corpus or partition

**Usage**

```
getTerms(.Object, ...)

## S4 method for signature 'character'
getTerms(.Object, pAttribute, robust = FALSE,
         verbose = TRUE, mc = FALSE)
```

**Arguments**

.Object	the object
...	further parameters
pAttribute	the pAttribute
robust	logical, whether to be robust
verbose	logical, whether to be verbose
mc	logical, whether to use multicore

---

getTokenStream	<i>Get Token Stream Based on Corpus Positions.</i>
----------------	--

---

**Description**

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



**Usage**

```

getTokenStream(.Object, ...)

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

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

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

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

## S4 method for signature 'regions'
getTokenStream(.Object, pAttribute = "word", ...)

```

**Arguments**

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

---

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	<i>Get Hits.</i>
------------	------------------

---

### 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, sAttribute = NULL,
     pAttribute = "word", size = FALSE, freq = FALSE, mc = FALSE,
     verbose = TRUE, progress = TRUE)

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

## S4 method for signature 'partitionBundle'
hits(.Object, query, cqp = FALSE,
     pAttribute = getOption("polmineR.pAttribute"), 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, sAttribute = NULL, verbose = TRUE)
```

### Arguments

.Object	a character, partition or partitionBundle object
...	further parameters
query	a (optionally named, see details) 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
sAttribute	s-attributes
pAttribute	p-attribute (will be passed into cpos)
size	logical - return size of subcorpus
freq	logical - 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 occur 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"  
 pAttribute p-attribute that has been queried  
 encoding encoding of the corpus  
 name name of the object

---

html	<i>Generate html from object.</i>
------	-----------------------------------

---

### 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 'partitionBundle'
html(object, filename = c(), type = "debate")

## S4 method for signature 'kwic'
```

```
html(object, i, sAttribute = 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 <code>as.markdown</code>
x	object of class <code>html</code> to print
meta	metadata for output, if <code>NULL</code> (default), the s-attributes defining a partition will be used
cpos	logical, if <code>TRUE</code> (default), all tokens will be wrapped by elements with <code>id</code> attribute indicating corpus positions
verbose	logical, whether to be verbose
cutoff	maximum number of tokens to decode from token stream, passed into <code>as.markdown</code>
charoffset	logical, if <code>TRUE</code> , character offset positions are added to elements embracing tokens
beautify	logical, if <code>TRUE</code> , whitespace before interpunctuation will be removed
filename	the filename
type	the partition type
i	if object is a <code>kwic</code> -object, the index of the concordance for which the fulltext is to be generated
sAttribute	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, sAttribute = "id")
}
```

```

H <- html(K, i = 2, sAttribute = "id")
for (i in 1:length(K)) {
  H <- html(K, i = i, sAttribute = "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"), pAttribute = "word",
     sAttribute = NULL, cpos = TRUE, stoplist = NULL, positivelist = NULL,
     regex = FALSE, verbose = TRUE)
```

```
## S4 method for signature 'character'
kwic(.Object, query, cqp = is.cqp,
     left = getOption("polmineR.left"), right = getOption("polmineR.right"),
     meta = getOption("polmineR.meta"), pAttribute = "word",
     sAttribute = 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
pAttribute	p-attribute, defaults to 'word'
sAttribute	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 occur in results
regex	logical, whether stoplist/positivelist is processed as regular expression

### Details

If a positivelist is 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	<i>kwic (S4 class)</i>
------------	------------------------

---

### 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	sAttributes (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 sAttributes 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	<i>Assign and get labels.</i>
-------	-------------------------------

---

**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	<i>Labels class.</i>
--------------	----------------------

---

**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	<i>text statistics</i>
----	------------------------

---

**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	<i>Mail result.</i>
------	---------------------

---

### 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	<i>calculate means</i>
-------	------------------------

---

**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

---

name	<i>generic methods defined in the polmineR-package</i>
------	--

---

**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

---

ngrams-class	<i>Get N-Grams</i>
--------------	--------------------

---

**Description**

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

**Usage**

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

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

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

**Arguments**

.Object	object of class partition
...	further parameters
n	number of tokens/characters
pAttribute	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, pAttribute = "word", char = NULL)
# a more complex scenario: get most frequent ADJA/NN-combinations
ngramObject <- ngrams(P, n = 2, pAttribute = 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, pAttribute, ...)

```

**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 to qualify a term as non-noise
specialChars	special characters to drop
numbers	regex, to drop numbers
verbose	logical
pAttribute	relevant if applied to a textstat object

**Value**

a list

---

partition	<i>Initialize a partition.</i>
-----------	--------------------------------

---

**Description**

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

**Usage**

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

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

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

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

## S4 method for signature 'partition'
partition(.Object, def = NULL, name = "",
  regex = FALSE, pAttribute = 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 = NULL,
  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

<code>pAttribute</code>	the <code>pAttribute(s)</code> for which term frequencies shall be retrieved
<code>regex</code>	logical (defaults to <code>FALSE</code> )
<code>xml</code>	either 'flat' (default) or 'nested'
<code>decode</code>	whether to turn token ids to strings (set <code>FALSE</code> to minimize object.size / memory consumption)
<code>type</code>	character vector (length 1) specifying the type of corpus / partition (e.g. "plpr")
<code>mc</code>	whether to use multicore (for counting terms)
<code>verbose</code>	logical, defaults to <code>TRUE</code>
<code>slots</code>	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 convenience - 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 backslashes 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 identified 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 `sAttributes` with decreasingly restrictive conditions. If you have a nested XML, it is mandatory that the order of the `sAttributes` 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 `pAttribute` 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 `pAttribute` 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", pAttribute = "word")
merkel <- partition("GERMAPARLMINI", speaker = ".*Merkel", pAttribute = "word", regex = TRUE)
sAttributes(merkel, "date")
sAttributes(merkel, "speaker")
merkel <- partition(
  "GERMAPARLMINI", speaker = "Angela Dorothea Merkel",
  date = "2009-11-10", interjection = "speech", pAttribute = "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)

```

---

partitionBundle

*Generate a bundle of partitions*


---

**Description**

A partitionBundle object is a S4 class object. partitionBundle,character-method will create a bundle of partitions, but not yet enriched.

**Usage**

```

partitionBundle(.Object, ...)

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

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

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

```

**Arguments**

.Object	character string, a partition, or a list
...	parameters to be passed into partition-method (see respective documentation)
sAttribute	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
xml	logical

**Value**

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

**Author(s)**

Andreas Blaette

**See Also**

[partition](#) and [bundle-class](#)

**Examples**

```
use("polmineR")
bt2009 <- partition("GERMAPARLMINI", date = "2009-.*", regex = TRUE)
pBundle <- partitionBundle(bt2009, sAttribute = "date", progress = TRUE, pAttribute = "word")
dtm <- as.DocumentTermMatrix(pBundle, col = "count")
summary(pBundle)
btBundle <- partitionBundle("GERMAPARLMINI", sAttribute = "date")
```

---

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

---

**Description**

Class and methods to manage bundles of partitions.

**Usage**

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

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

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

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

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

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

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

## S4 method for signature 'partitionBundle'
sAttributes(.Object, sAttribute)
```

**Arguments**

object	a partitionBundle object
x	a partitionBundle object
i	integer index
height	height
...	further parameters
.Object	a partitionBundle object
mc	logical or, if numeric, providing the number of cores
progress	logical
verbose	logical
sAttribute	the s-attribute to use

**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  
 sAttributesFixed Object of class "list" fixed sAttributes  
 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 partitionBundle

**Author(s)**

Andreas Blaette

---

partition\_class      *Partition class and methods.*

---

**Description**

S4 partition class and methods for instances of class partition.

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 disjointed regions.

**Usage**

```

## S4 method for signature 'partition'
name(x)

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

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

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

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

## S4 method for signature 'partition'
aggregate(x)

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

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

```

```

## S4 method for signature 'partition'
pAttributes(.Object, pAttribute = NULL)

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

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

```

### Arguments

x	a partition object
value	value
...	further parameters
.Object	a partition object
meta	...
template	template to use
cpos	...
cutoff	maximum number of tokens to decode
verbose	logical
size	logical
pAttribute	a p-attribute (for enriching)
decode	logical
mc	logical or, if numeric, providing the number of cores

### Slots

name Object of class "character" a name that may be useful  
corpus Object of class "character" the CWB corpus the partition is based on  
encoding Object of class "character" encoding of the corpus  
sAttributes Object of class "list" s-attributes specifying the partition  
explanation Object of class "character" an explanation of the partition  
cpos Object of class "matrix" corpus positions  
annotations Object of class "list"  
pos Object of class "list" with tables "abs", "rel" and "max"  
size Object of class "numeric" total size of the partition  
metadata Object of class "data.frame" metadata information  
strucs Object of class "numeric" the strucs defining the partition  
pAttribute Object of class "character" indicating the pAttribute of the count in slot tf  
xml Object of class "character" whether the xml is flat or nested  
sAttributeStrucs 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.

**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@cpo
```

---

pAttribute

*get pAttribute*

---

**Description**

get pAttribute

**Usage**

```
pAttribute(object)

## S4 method for signature 'textstat'
pAttribute(object)
```

**Arguments**

object	a textstat object
pAttribute	the pAttribute to get

---

pAttributes                      *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 pAttributes-method.

### Usage

```
pAttributes(.Object, ...)
```

```
## S4 method for signature 'character'
```

```
pAttributes(.Object, pAttribute = NULL)
```

### Arguments

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

### References

Stefan Evert & The OCWB Development Team, CQP Query Language Tutorial, [http://cwb.sourceforge.net/files/CQP\\_Tutorial](http://cwb.sourceforge.net/files/CQP_Tutorial)

### Examples

```
use("polmineR")
```

```
pAttributes("GERMAPARLMINI")
```

---

read                                      *Display full text.*

---

### Description

Generate text (i.e. html) and read it in the viewer pane of RStudio. If called on a "partitionBundle"-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 = getTemplate(.Object),
     ...)

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

## S4 method for signature 'data.table'
read(.Object, col, partitionBundle, 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**

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

### Examples

```
use("polmineR")
template <- jsonlite::fromJSON(system.file(package = "polmineR", "templates", "plpr.template.json"))
options(polmineR.templates = list("GERMAPARLMINI" = template))
merkel <- partition(
  "GERMAPARLMINI",
  date = "2009-11-10", speaker = "Angela Dorothea Merkel",
  type = "plpr"
)
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)
encode(R, sAttribute = "foo", values = "Jens")
```

---

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-regeedit' 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](http://cwb.sourceforge.net/files/CWB_Encoding_Tutorial.pdf)) includes an explanation of the registry file format.

---

registry_reset	<i>Reset Registry Directory.</i>
----------------	----------------------------------

---

### 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,

### Usage

```
registry_reset(registryDir = getOption("polmineR.defaultRegistry"),  
              verbose = TRUE)
```

### 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)
```

---

sAttributes,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 sAttributes-method.

**Usage**

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

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

**Arguments**

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

**Details**

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

If sAttribute 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")

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

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

```

---

size	<i>Get Number of Tokens.</i>
------	------------------------------

---

**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, sAttribute = NULL, verbose = TRUE)

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

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

```

**Arguments**

x	object to get size(s) for
...	further arguments
sAttribute	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 sAttribute is NULL, a data.table otherwise

**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", sAttribute = "date")
size("GERMAPARLMINI", sAttribute = c("date", "party"))

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

---

split,partition-method

*split partition into partitionBundle*

---

**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 partitionBundle 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 partitionBundle

---

tempcorpus-class	<i>S4 class to capture core information on a temporary CWB corpus</i>
------------------	---

---

**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	<i>Methods for TermDocumentMatrix / DocumentTermMatrix</i>
--------------------	--

---

**Description**

Methods for TermDocumentMatrix / DocumentTermMatrix

**Usage**

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

**Arguments**

x	object
---	--------

---

terms	<i>Get Terms Occurring in Partition or Corpus.</i>
-------	--

---

### Description

Get Terms Occurring in Partition or Corpus.

### Usage

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

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

### Arguments

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

### Examples

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

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

---

textstat-class	<i>S4 textstat class</i>
----------------	--------------------------

---

### Description

Superclass for features, context, and partition class.



**Usage**

```
## 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'  
x[[i]]  
  
## S4 method for signature 'textstat,ANY,ANY,ANY'  
x[i, j]  
  
## S4 method for signature 'textstat'  
view(.Object)
```

**Arguments**

x                    textstat object

...	further parameters
digits	no of digits
by	by
decreasing	logical
object	an object
e1	object 1
e2	object 2
i	vector to index data.table in stat-slot
j	vector to index data.table in stat-slot
.Object	an object

### 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`.

### Slots

`pAttribute` 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 = ".*", pAttribute = "word", regex = TRUE)
y <- cooccurrences(P, query = "Arbeit")
y[1:25]
y[,c("word", "11")]
y[1:25, "word"]
y[1:25][["word"]]
y[which(y[["word"]] %in% c("Arbeit", "Sozial"))]
y[ y[["word"]] %in% c("Arbeit", "Sozial") ]
```

---

TokenStream-class      *Class for token stream operations.*

---

### Description

Class for token stream operations.

### Slots

tokenStream a data.table that will include a column "token"  
 corpus the CWB corpus (character vector length 1)  
 encoding the encoding of the CWB corpus (character vector length 1)

---

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 'partitionBundle' 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

---

tTest	<i>perform t-test</i>
-------	-----------------------

---

**Description**

S4 method for context object to perform t-test

**Usage**

```
tTest(.Object)

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

**Arguments**

.Object            a context or features object

---

use	<i>Use a packaged corpus.</i>
-----	-------------------------------

---

**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 `./ext-data/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	<i>browse an object using View()</i>
------	--------------------------------------

---

**Description**

browse an object using `View()`

**Usage**

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

**Arguments**

<code>.Object</code>	an object
<code>...</code>	further parameters

---

weigh	<i>weigh a matrix</i>
-------	-----------------------

---

**Description**

weigh a 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

- \*Topic **datasets**
  - Corpus, 21
  - CQI.super, 27
  - trim, 76
- \*Topic **package**
  - polmineR-package, 4
- + , bundle, bundle-method (bundle-class), 12
- + , bundle, textstat-method (bundle-class), 12
- + , partitionBundle, ANY-method (partitionBundle-class), 58
- + , partitionBundle, partition-method (partitionBundle-class), 58
- + , partitionBundle, partitionBundle-method (partitionBundle-class), 58
- + , partitionBundle-method (partitionBundle-class), 58
- + , textstat, textstat-method (textstat-class), 72
- .addCharacterOffset (html), 44
- [ , context, ANY, ANY, ANY-method (context-class), 16
- [ , context-method (context-class), 16
- [ , contextBundle, ANY, ANY, ANY-method (contextBundle-class), 18
- [ , contextBundle-method (contextBundle-class), 18
- [ , kwic, ANY, ANY, ANY-method (kwic-class), 48
- [ , kwic-method (kwic-class), 48
- [ , partition, ANY, ANY, ANY-method (partition\_class), 60
- [ , partition-method (partition\_class), 60
- [ , partitionBundle, ANY, ANY, ANY-method (partitionBundle-class), 58
- [ , partitionBundle-method (partitionBundle-class), 58
- [ , textstat, ANY, ANY, ANY-method (textstat-class), 72
- [ , bundle-method (bundle-class), 12
- [ , context-method (context-class), 16
- [ , contextBundle-method (contextBundle-class), 18
- [ , partitionBundle-method (partitionBundle-class), 58
- [ , textstat-method (textstat-class), 72
- aggregate, partition-method (partition\_class), 60
- as.bundle (textstat-class), 72
- as.bundle, list-method (bundle-class), 12
- as.bundle, textstat-method (bundle-class), 12
- as.corpusEnc (encodings), 33
- as.cqp (cqp), 28
- as.data.frame, cooccurrencesBundle-method (cooccurrences-class), 20
- as.data.frame, kwic-method (kwic-class), 48
- as.data.frame, textstat-method (textstat-class), 72
- as.data.table, bundle-method (bundle-class), 12
- as.data.table, regions-method (regions), 65
- as.data.table, textstat-method (textstat-class), 72
- as.DataTables, context-method (context-class), 16
- as.DataTables, textstat-method (textstat-class), 72
- as.DocumentTermMatrix (as.TermDocumentMatrix), 7
- as.DocumentTermMatrix, bundle-method (as.TermDocumentMatrix), 7
- as.DocumentTermMatrix, character-method (as.TermDocumentMatrix), 7



- as.DocumentTermMatrix, context-method  
(as.TermDocumentMatrix), 7
- as.DocumentTermMatrix, partitionBundle-method  
(as.TermDocumentMatrix), 7
- as.markdown, 5
- as.markdown, numeric-method  
(as.markdown), 5
- as.markdown, partition-method  
(partition\_class), 60
- as.markdown, plprPartition-method  
(as.markdown), 5
- as.matrix, bundle-method (bundle-class),  
12
- as.matrix, contextBundle-method  
(context), 14
- as.matrix, partitionBundle-method  
(partitionBundle-class), 58
- as.nativeEnc (encodings), 33
- as.partitionBundle (partition\_class), 60
- as.partitionBundle, list-method  
(partitionBundle-class), 58
- as.partitionBundle, partition-method  
(partition\_class), 60
- as.regions (regions), 65
- as.sparseMatrix, 6
- as.sparseMatrix, bundle-method  
(as.sparseMatrix), 6
- as.sparseMatrix, simple\_triplet\_matrix-method  
(as.sparseMatrix), 6
- as.sparseMatrix, TermDocumentMatrix-method  
(as.sparseMatrix), 6
- as.speeches, 6
- as.speeches, partition-method  
(as.speeches), 6
- as.TermContextBundle, contextBundle-method  
(contextBundle-class), 18
- as.TermDocumentMatrix, 7
- as.TermDocumentMatrix, bundle-method  
(as.TermDocumentMatrix), 7
- as.TermDocumentMatrix, character-method  
(as.TermDocumentMatrix), 7
- as.TermDocumentMatrix, context-method  
(as.TermDocumentMatrix), 7
- as.TermDocumentMatrix, partitionBundle-method  
(as.TermDocumentMatrix), 7
- as.utf8 (encodings), 33
- as.VCorpus, partitionBundle-method, 9
- barplot, partitionBundle-method  
(partitionBundle-class), 58
- blapply, 10
- blapply, bundle-method (blapply), 10
- blapply, list-method (blapply), 10
- blapply, vector-method (blapply), 10
- browse, 11
- browse, cooccurrences-method (browse), 11
- browse, html-method (browse), 11
- browse, kwic-method (browse), 11
- browse, partition-method (browse), 11
- browse, pressPartition-method (browse),  
11
- browse, textstat-method (browse), 11
- browse<- (name), 52
- bundle-class, 12
- chisquare, 14
- chisquare, context-method (chisquare), 14
- chisquare, textstat-method (chisquare),  
14
- colnames, textstat-method  
(textstat-class), 72
- context, 14
- context, character-method (context), 14
- context, contextBundle-method (context),  
14
- context, cooccurrences-method (context),  
14
- context, partition-method (context), 14
- context, partitionBundle-method  
(context), 14
- context-class, 16
- context\_class (context-class), 16
- contextBundle-class, 18
- cooccurrences, 18
- cooccurrences, character-method  
(cooccurrences), 18
- cooccurrences, context-method  
(cooccurrences), 18
- cooccurrences, Corpus-method (context),  
14
- cooccurrences, partition-method  
(cooccurrences), 18
- cooccurrences, partitionBundle-method  
(cooccurrences), 18
- cooccurrences-class, 20
- cooccurrencesBundle  
(cooccurrences-class), 20

- cooccurrencesBundle-class
  - (cooccurrences-class), 20
- cooccurrencesReshaped, 21
- cooccurrencesReshaped-class
  - (cooccurrences-class), 20
- Corpus, 21
- corpus, 23
- corpus, bundle-method (corpus), 23
- corpus, missing-method (corpus), 23
- corpus, partition-method (corpus), 23
- count, 23
- count, character-method (count), 23
- count, context-method (context-class), 16
- count, Corpus-method (count), 23
- count, partition-method (count), 23
- count, partitionBundle-method (count), 23
- count, vector-method (count), 23
- count-class (count\_class), 25
- count-method (count), 23
- count\_class, 25
- cpos, 26
- cpos, character-method (cpos), 26
- cpos, hits-method (cpos), 26
- cpos, matrix-method (cpos), 26
- cpos, partition-method (cpos), 26
- cpos, tempcorpus-method (cpos), 26
- CQI (CQI.super), 27
- CQI.super, 27
- cqp, 28
- cqpserver, 29
  
- decode, 29
- decode, character-method (decode), 29
- dim, textstat-method (textstat-class), 72
- dispersion, 30, 70
- dispersion, character-method
  - (dispersion), 30
- dispersion, hits-method (dispersion), 30
- dispersion, partition-method
  - (dispersion), 30
- dotplot, 32
- dotplot, features-method (dotplot), 32
- dotplot, featuresNgrams-method
  - (dotplot), 32
- dotplot, partition-method (dotplot), 32
- dotplot, textstat-method (dotplot), 32
  
- encoding, 33
- encoding, bundle-method (encoding), 33
- encoding, textstat-method (encoding), 33
- encoding<- (encoding), 33
- encodings, 33
- enrich, 34
- enrich, context-method (context-class),
  - 16
- enrich, kwic-method (kwic-class), 48
- enrich, partition-method
  - (partition\_class), 60
- enrich, partitionBundle-method
  - (partitionBundle-class), 58
- enrich-method (enrich), 34
- export (partition\_class), 60
- export, partition-method
  - (partition\_class), 60
  
- features (features, partition-method), 34
- features, count-method
  - (features, partition-method), 34
- features, ngrams-method
  - (features, partition-method), 34
- features, partition-method, 34
- features, partitionBundle-method
  - (features, partition-method), 34
- features-class, 36
- featuresBundle-class (features-class),
  - 36
- featuresCooccurrences-class
  - (features-class), 36
- featuresNgrams-class (features-class),
  - 36
- flatten, 37
  
- getEncoding, 38
- getEncoding, character-method
  - (getEncoding), 38
- getObjects, 38
- getSlot, 39
- getTemplate, 39
- getTemplate, character-method
  - (getTemplate), 39
- getTemplate, missing-method
  - (getTemplate), 39
- getTemplate, partition-method
  - (getTemplate), 39
- getTerms, 40
- getTerms, character-method (getTerms), 40
- getTokenStream, 40

- getTokenStream, character-method (getTokenStream), 40
- getTokenStream, matrix-method (getTokenStream), 40
- getTokenStream, numeric-method (getTokenStream), 40
- getTokenStream, partition-method (getTokenStream), 40
- getTokenStream, regions-method (getTokenStream), 40
  
- head, context-method (context-class), 16
- head, textstat-method (textstat-class), 72
- highlight, 41
- highlight, character-method (highlight), 41
- highlight, html-method (highlight), 41
- highlight, kwic-method (highlight), 41
- hist, partition-method (partition\_class), 60
- hits, 70
- hits (hits-class), 43
- hits, character-method (hits-class), 43
- hits, context-method (hits-class), 43
- hits, partition-method (hits-class), 43
- hits, partitionBundle-method (hits-class), 43
- hits-class, 43
- html, 44
- html, character-method (html), 44
- html, kwic-method (html), 44
- html, partition-method (html), 44
- html, partitionBundle-method (html), 44
  
- is.cqp (cqp), 28
  
- kwic, 46, 65
- kwic, character-method (kwic), 46
- kwic, context-method (kwic), 46
- kwic, partition-method (kwic), 46
- kwic-class, 48
  
- label, 49
- label, kwic-method (label), 49
- label<- (label), 49
- Labels (Labels-class), 50
- Labels-class, 50
- length, bundle-method (bundle-class), 12
- length, kwic-method (kwic-class), 48
- length, partition-method (partition\_class), 60
- ll, 50
- ll, context-method (ll), 50
- ll, cooccurrences-method (ll), 50
- ll, features-method (ll), 50
  
- mail, 51
- mail, cooccurrences-method (mail), 51
- mail, data.frame-method (mail), 51
- mail, features-method (mail), 51
- mail, kwic-method (mail), 51
- mail, partition-method (mail), 51
- mail-method (mail), 51
- means, 52
- means, DocumentTermMatrix-method (means), 52
- merge, cooccurrencesReshaped-method (cooccurrencesReshaped), 21
- merge, partitionBundle-method (partitionBundle-class), 58
  
- name, 52
- name, partition-method (partition\_class), 60
- name<- (name), 52
- name<- , partition, character-method (partition\_class), 60
- names, bundle-method (bundle-class), 12
- names, partitionBundle-method (partitionBundle-class), 58
- names, textstat-method (textstat-class), 72
- names<- , bundle, character-method (bundle-class), 12
- ngrams (ngrams-class), 53
- ngrams, partition-method (ngrams-class), 53
- ngrams, partitionBundle-method (ngrams-class), 53
- ngrams-class, 53
- noise, 54
- noise, character-method (noise), 54
- noise, DocumentTermMatrix-method (noise), 54
- noise, TermDocumentMatrix-method (noise), 54
- noise, textstat-method (noise), 54

- nrow, textstat-method (textstat-class),  
72
- partition, 55, 58
- partition, character-method (partition),  
55
- partition, Corpus-method (partition), 55
- partition, environment-method  
(partition), 55
- partition, list-method (partition), 55
- partition, partition-method (partition),  
55
- partition-class (partition\_class), 60
- partition\_class, 34, 56, 60
- partitionBundle, 57
- partitionBundle, character-method  
(partitionBundle), 57
- partitionBundle, context-method  
(partitionBundle), 57
- partitionBundle, environment-method  
(partitionBundle-class), 58
- partitionBundle, partition-method  
(partitionBundle), 57
- partitionBundle-class, 58
- pAttribute, 62
- pAttribute, textstat-method  
(pAttribute), 62
- pAttributes, 63
- pAttributes, character-method  
(pAttributes), 63
- pAttributes, context-method  
(context-class), 16
- pAttributes, partition-method  
(partition\_class), 60
- plprPartition-class (partition\_class),  
60
- pmi (ll), 50
- pmi, context-method (ll), 50
- polmineR (polmineR-package), 4
- polmineR-generics (name), 52
- polmineR-package, 4
- pressPartition-class (partition\_class),  
60
- print.html (html), 44
- punctuation (trim), 76
- read, 47, 63
- read, data.table-method (read), 63
- read, hits-method (read), 63
- read, kwic-method (read), 63
- read, partition-method (read), 63
- read, partitionBundle-method (read), 63
- read, regions-method (read), 63
- regions, 65
- regions-class (regions), 65
- registry\_get\_encoding  
(registry\_get\_name), 66
- registry\_get\_home (registry\_get\_name),  
66
- registry\_get\_id (registry\_get\_name), 66
- registry\_get\_info (registry\_get\_name),  
66
- registry\_get\_name, 66
- registry\_get\_p\_attributes  
(registry\_get\_name), 66
- registry\_get\_properties  
(registry\_get\_name), 66
- registry\_get\_s\_attributes  
(registry\_get\_name), 66
- registry\_reset, 67
- round, textstat-method (textstat-class),  
72
- rownames, textstat-method  
(textstat-class), 72
- sample, bundle-method (bundle-class), 12
- sample, context-method (context-class),  
16
- sample, hits-method (hits-class), 43
- sample, kwic-method (kwic-class), 48
- sAttributes  
(sAttributes, character-method),  
68
- sAttributes, character-method, 68
- sAttributes, partition-method  
(sAttributes, character-method),  
68
- sAttributes, partitionBundle-method  
(partitionBundle-class), 58
- setTemplate (getTemplate), 39
- setTemplate, character-method  
(getTemplate), 39
- setTemplate, missing-method  
(getTemplate), 39
- show, context-method (context-class), 16
- show, contextBundle-method  
(contextBundle-class), 18

- show, cooccurrences-method  
(cooccurrences-class), 20
- show, features-method (features-class), 36
- show, html-method (html), 44
- show, kwic-method (kwic-class), 48
- show, partition-method  
(partition\_class), 60
- show, partitionBundle-method  
(partitionBundle-class), 58
- show, textstat-method (textstat-class), 72
- size, 69
- size, character-method (size), 69
- size, DocumentTermMatrix-method (size), 69
- size, partition-method (size), 69
- size, TermDocumentMatrix-method  
(TermDocumentMatrix), 71
- sort, textstat-method (textstat-class), 72
- split (partition\_class), 60
- split, partition  
(split, partition-method), 70
- split, partition-method, 70
- startServer (cqpsrver), 29
- subset, bundle-method (bundle-class), 12
- subset, textstat-method  
(textstat-class), 72
- summary, context-method (context-class), 16
- summary, contextBundle-method  
(contextBundle-class), 18
- summary, cooccurrences-method  
(cooccurrences-class), 20
- summary, features-method  
(features-class), 36
- summary, featuresBundle-method  
(features-class), 36
- summary, partitionBundle-method  
(partitionBundle-class), 58
  
- tail, textstat-method (textstat-class), 72
- tempcorpus (tempcorpus-class), 71
- tempcorpus-class, 71
- TermDocumentMatrix, 71
- terms, 72
- terms, character-method (terms), 72
- terms, partition-method (terms), 72
- textstat-class, 72
- TokenStream-class, 75
- tooltips, 75
- tooltips, character-method (tooltips), 75
- tooltips, html-method (tooltips), 75
- trim, 76
- trim, context-method (context-class), 16
- trim, DocumentTermMatrix-method (trim), 76
- trim, TermDocumentMatrix-method (trim), 76
- trim-method (trim), 76
- tTest, 77
- tTest, context-method (tTest), 77
  
- unique, bundle-method (bundle-class), 12
- use, 67, 77
  
- view, 78
- view, cooccurrences-method  
(partition\_class), 60
- view, cooccurrencesReshaped-method  
(cooccurrences-class), 20
- view, features-method (features-class), 36
- view, kwic-method (kwic-class), 48
- view, partition-method  
(partition\_class), 60
- view, textstat-method (textstat-class), 72
  
- weigh, 79
- weigh, DocumentTermMatrix-method  
(weigh), 79
- weigh, TermDocumentMatrix-method  
(weigh), 79