

# Package ‘polmineR’

October 8, 2018

**Type** Package

**Title** Toolkit for Corpus Analysis

**Version** 0.7.10

**Date** 2018-09-18

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

**Suggests** markdown, rmarkdown, htmltools, openxlsx, sendmailR, shiny,  
shinythemes, testthat, tidytext, magrittr

**VignetteBuilder** knitr

**LazyData** yes

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

**BugReports** <https://github.com/PolMine/polmineR/issues>

**Biarch** true

**License** GPL-3

**URL** <https://www.github.com/PolMine/polmineR>

**Collate** 'CQI.R' 'Labels.R' 'polmineR.R' 'S4classes.R' 'p\_attributes.R'  
'textstat.R' 'bundle.R' 'corpus.R' 'count.R' 'partition.R'  
'partition\_bundle.R' 'ngrams.R' 'features.R' 'context.R'  
'TermDocumentMatrix.R' 'as.VCorpus.R' 'as.markdown.R'  
'cooccurrences.R' 'as.sparseMatrix.R' 'as.speeches.R'  
'blapply.R' 'kwic.R' 'browse.R' 'chisquare.R' 'hits.R'  
'tempcorpus.R' 'cpos.R' 'cqpserver.R' 'decode.R' 'dispersion.R'  
'dotplot.R' 'encoding.R' 'enrich.R' 'highlight.R' 'html.R'  
'label.R' 'll.R' 'mail.R' 'means.R' 'noise.R' 'pmi.R'  
'regions.R' 'read.R' 'registry.R' 'reindex.R' 'renamed.R'  
's\_attributes.R' 'size.R' 'store.R' 't\_test.R' 'templates.R'  
'terms.R' 'token\_stream.R' 'tooltips.R' 'trim.R' 'type.R'  
'use.R' 'utils.R' 'view.R' 'weigh.R' 'zzz.R'

**RoxygenNote** 6.0.1

**NeedsCompilation** no

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

**Maintainer** Andreas Blaette <[andreas.blaette@uni-due.de](mailto:andreas.blaette@uni-due.de)>

**Repository** CRAN

**Date/Publication** 2018-09-18 11:30:03 UTC

## R topics documented:

as.markdown	3
as.sparseMatrix	4
as.speeches	5
as.TermDocumentMatrix	6
as.VCorpus	8
blapply	8
browse	9
bundle-class	10
chisquare	12
context	13
context-class	14
context_bundle-class	16
cooccurrences	16
cooccurrences-class	18
Corpus	19
corpus	20
count	21
count_class	23
cpos	24
CQI.super	25
cqp	26
decode	27
dispersion	28
dotplot	29
encoding	30
encodings	30
enrich	31
features	31
features-class	33
get_template	34
get_token_stream	35
get_type	36
highlight	37
hits	38
hits_class	39
html	40
kwic	41
kwic-class	43
label	45
Labels-class	46
ll	46

mail . . . . .	47
means . . . . .	48
ngrams . . . . .	48
ngrams_class . . . . .	49
noise . . . . .	49
partition . . . . .	50
partition_bundle . . . . .	52
partition_bundle-class . . . . .	54
partition_class . . . . .	56
polmineR . . . . .	58
p_attributes . . . . .	59
read . . . . .	60
regions . . . . .	61
registry . . . . .	62
registry_get_name . . . . .	63
registry_reset . . . . .	63
renamed . . . . .	64
size . . . . .	65
store . . . . .	66
subcorpus . . . . .	67
s_attributes . . . . .	67
tempcorpus . . . . .	69
tempcorpus_class . . . . .	69
terms . . . . .	69
textstat-class . . . . .	70
tooltips . . . . .	72
trim . . . . .	73
t_test . . . . .	74
use . . . . .	74
view . . . . .	75
weigh . . . . .	76

## Index 78

---

as.markdown	<i>Get markdown-formatted full text of a partition.</i>
-------------	---------------------------------------------------------

---

### Description

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

### Usage

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

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

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

### Arguments

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

### Examples

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

---

as.sparseMatrix	<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 corpus or partition into speeches.</i>
-------------	-------------------------------------------------

---

## Description

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

## Usage

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

## Arguments

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

## Value

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

## Examples

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

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

---

as.TermDocumentMatrix *Generate TermDocumentMatrix / DocumentTermMatrix.*


---

## Description

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

## Usage

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

as.DocumentTermMatrix(x, ...)

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

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

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

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

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

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

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

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

## Arguments

x	a character vector indicating a corpus, or an object of class bundle, or inheriting from class bundle (e.g. partition_bundle)
...	s-attribute definitions used for subsetting the corpus, compare partition-method

p_attribute	p-attribute counting is be based on
s_attribute	s-attribute that defines content of columns, or rows
verbose	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 s\_attribute provided. Counts are performed for the p\_attribute. Further parameters provided (passed in as ... are interpreted as s-attributes that define a subset of the corpus for splitting it according to s\_attribute. If str values for s\_attribute are not unique, the necessary aggregation is performed, slowing things somewhat down.

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

## Value

a TermDocumentMatrix

## Author(s)

Andreas Blaette

## Examples

```
use("polmineR")

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

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

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

---

as.VCorpus	<i>Coerce partition_bundle to VCorpus.</i>
------------	--------------------------------------------

---

### Description

Coerce partition\_bundle to VCorpus.

### Usage

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

### Arguments

x                      a partition\_bundle object

### Examples

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

---

blapply	<i>apply a function over a list or bundle</i>
---------	-----------------------------------------------

---

### Description

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

### Usage

```
blapply(x, ...)

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

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

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



**Arguments**

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

**Examples**

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

---

browse

*Display in browser*


---

**Description**

Display in browser

**Usage**

```
browse(object, ...)

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

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

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

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

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

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

**Arguments**

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

bundle-class

*Bundle Class***Description**

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

**Usage**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### Arguments

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

### Slots

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

objects An object of class "list"

p\_attribute Object of class "character"

encoding The encoding of the corpus.

### Author(s)

Andreas Blaette

### Examples

```
parties <- s_attributes("GERMAPARLMINI", "party")
parties <- parties[-which(parties == "NA")]
party_bundle <- partition_bundle("GERMAPARLMINI", s_attribute = "party")
length(party_bundle)
names(party_bundle)
party_bundle <- enrich(party_bundle, p_attribute = "word")
summary(party_bundle)
parties_big <- party_bundle[[c("CDU-CSU", "SPD")]]
summary(parties_big)
use("polmineR")
Ps <- partition_bundle(
  "REUTERS", s_attribute = "id",
  values = s_attributes("REUTERS", "id")
)
```

```
)  
Cs <- cooccurrences(Ps, query = "oil", cqp = FALSE, verbose = FALSE, progress = TRUE)  
dt <- as.data.table(Cs, col = "ll")  
m <- as.matrix(Cs, col = "ll")
```

---

chisquare

*perform chisquare-text*

---

## Description

Perform Chisquare-Test based on a table with counts

## Usage

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

## Arguments

.Object	object
...	further parameters

## Details

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

## Value

a table

## Author(s)

Andreas Blaette

---

context	Analyze context of a node word.
---------	---------------------------------

---

## Description

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

## Usage

```
context(.Object, ...)
```

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

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

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

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

## Arguments

.Object	a partition or a partition_bundle object
...	further parameters
query	A query, which may be a character vector or a CQP query.
cqp	defaults to is.cqp-function, or provide TRUE/FALSE
left	Number of tokens to the left of the query match.
right	Number of tokens to the right of the query match.
p_attribute	The p-attribute of the query.
boundary	If provided, a length-one character vector specifying a s-attribute. It will be checked that corpus positions do not extend beyond the region defined by the s-attribute.
stoplist	Exclude match for query if stopword(s) is/are present in context. See positivelist for further explanation.
positivelist	character vector or numeric/integer vector: include a query hit only if token in positivelist is present. If positivelist is a character vector, it may include regular expressions (see parameter regex)

regex	logical, defaults to FALSE - whether stoplist and/or positivelist are regular expressions
count	logical
mc	whether to use multicore; if NULL (default), the function will get the value from the options
verbose	report progress, defaults to TRUE
progress	logical, whether to show progress bar
complete	enhance completely

### Details

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

### Value

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

### Author(s)

Andreas Blaette

### Examples

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

---

context-class	<i>Context class.</i>
---------------	-----------------------

---

### Description

Class to organize information of context analysis.

### Usage

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

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

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

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

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

## S4 method for signature 'context'
as.regions(x, node = TRUE)

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

### Arguments

<code>x</code>	a context object
<code>.Object</code>	object
<code>size</code>	integer indicating sample size
<code>s_attribute</code>	s-attribute(s) to add to data.table in cpos-slot
<code>p_attribute</code>	p-attribute(s) to add to data.table in cpos-slot
<code>decode</code>	logical, whether to convert integer ids to expressive strings
<code>verbose</code>	logical, whether to be talkative
<code>...</code>	to maintain backwards compatibility if argument pAttribute is still used
<code>node</code>	A logical value, whether to include the node (i.e. query matches) in the region matrix generated when creating a partition from a context-object.
<code>object</code>	a context object
<code>positivelist</code>	tokens that are required to be present to keep a match
<code>regex</code>	logical, whether positivlist / stoplist is interpreted as regular expressions
<code>stoplist</code>	tokens that are used to exclude a match
<code>progress</code>	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 length-method will return the number of hits that were achieved.

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

### Slots

`query` The query used/node examined (character).  
`count` An integer value, the number of hits.  
`partition` The partition the context object is based on.  
`size_partition` A length-one integer, the size of the partition.

**left** An integer value, the number of tokens to the left.  
**right** An integer value, the number of tokens to the right.  
**size** An integer value, number of tokens in the right and left context of the node.  
**boundary** An s-attribute (character).  
**p\_attribute** The p-attribute of the query (character).  
**corpus** The CWB corpus used (character).  
**stat** A data.table, the statistics of the analysis.  
**encoding** Object of class character, encoding of the corpus.  
**cpos** A data.table, with the columns hit\_no, cpos, position, word\_id.  
**method** A character-vector, statistical test used.  
**call** Object of class character, call that generated the object.

---

context\_bundle-class    *S4 context\_bundle class*

---

### Description

class to organize information of multiple context analyses

### Slots

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

### Methods

**show** output of core information  
**summary** core statistical information  
**[** specific cooccurrences  
**[[** specific cooccurrences

---

cooccurrences                      *Get cooccurrence statistics.*

---

### Description

Get cooccurrence statistics.



**Usage**

```

cooccurrences(.Object, ...)

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

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

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

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

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

```

**Arguments**

.Object	a partition object, or a character vector with a CWB corpus
...	further parameters that will be passed into bigmatrix (applies only of big=TRUE)
query	query, may 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
p_attribute	the p-attribute of the tokens/the query
s_attribute	if provided, it will be checked that cpos do not extend beyond the region defined by the s-attribute
left	Number of tokens to the left of the query match.
right	Number of tokens to the right of the query match.
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, p_attribute = "word")
cooc <- cooccurrences(merkel, query = "Deutschland")
```

---

cooccurrences-class     *Cooccurrences class.*

---

**Description**

S4 class to organize information of context analysis

**Usage**

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

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

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

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

**Arguments**

<code>object</code>	object to work with
<code>x</code>	object to work with
<code>.Object</code>	object to work with

**Slots**

<code>call</code>	Object of class character the call that generated the object
<code>partition</code>	Object of class character the partition the analysis is based on
<code>size_partition</code>	Object of class integer the size of the partition
<code>left</code>	Object of class numeric number of tokens to the right
<code>right</code>	Object of class numeric number of tokens to the left
<code>p_attribute</code>	Object of class character p-attribute of the query
<code>corpus</code>	Object of class character the CWB corpus used
<code>stat</code>	Object of class data.table statistics of the analysis
<code>encoding</code>	Object of class character encoding of the corpus
<code>pos</code>	Object of class character part-of-speech tags filtered
<code>method</code>	Object of class character statistical test(s) used
<code>cutoff</code>	Object of class list cutoff levels that have been applied

---

Corpus

---

*Corpus class.*


---

**Description**

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

**Usage**

```
Corpus
```

**Format**

An object of class R6ClassGenerator of length 24.

**Fields**

<code>corpus</code>	character vector (length 1), a CWB corpus
<code>encoding</code>	encoding of the corpus (typically 'UTF-8' or 'latin1'), assigned automatically upon initialization of the corpus
<code>cpos</code>	a two-column matrix with regions of a corpus underlying the s-attributes of the data.table in field <code>s_attributes</code>
<code>s_attributes</code>	a data.table with the values of a set of s-attributes
<code>stat</code>	a data.table with counts

## Arguments

**corpus** a corpus

**registryDir** the directory where the registry file resides

**dataDir** the data directory of the corpus

**p\_attribute** p-attribute, to perform count

**s\_attributes** s-attributes

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

**as.html** logical

## Methods

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

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

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

`getInfo(as.html = FALSE)`

`showInfo()`

## Examples

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

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

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

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

---

corpus

---

*Get corpus/corpora available or used.*


---

## Description

Calling `corpus()` will return a `data.frame` listing the corpora described in the active registry directory, and some basic information on the corpora. If object is an object inheriting from the `textstat`, or the `bundle` class, the corpus used to generate the object is returned.

**Usage**

```
corpus(object)

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

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

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

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

**Arguments**

**object**                      An object inheriting from the textstat or bundle superclasses.

**Examples**

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

p <- partition("REUTERS", places = "kuwait")
corpus(p)

pb <- partition_bundle("REUTERS", s_attribute = "id")
corpus(pb)
```

---

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

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

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

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

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

### Arguments

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

### Details

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

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

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

### Value

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

## References

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

## See Also

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

count

## Examples

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

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

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

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

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

---

count\_class

*Count class.*

---

## Description

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

## Usage

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

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

## Arguments

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

## Details

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

**Slots**

stat Object of class data.table  
 corpus Object of class character the CWB corpus the partition is based on  
 encoding Object of class character encoding of the corpus  
 name Object of class character, a name for the object  
 size Object of class integer, the size of the partition or corpus the count is based upon

**Author(s)**

Andreas Blaette

**See Also**

The count-class inherits from the [textstat-class](#)

---

cpos

---

*Get corpus positions for a query or queries.*


---

**Description**

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

**Usage**

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

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

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

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

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

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



**Arguments**

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

**Details**

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

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

**Value**

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

---

CQI.super

---

*Interfaces for accessing the CWB*


---

**Description**

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

**Usage**

CQI.super

CQI.RcppCWB

CQI.cqpserver

CQI.cqpserver

**Format**

An object of class R6ClassGenerator of length 24.

---

cqp	<i>Tools for CQP queries.</i>
-----	-------------------------------

---

**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)
```

---

 decode

*Decode Structural Attribute or Entire Corpus.*


---

## Description

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

## Usage

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

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

## Arguments

<code>.Object</code>	the corpus to decode (character vector)
<code>...</code>	further parameters
<code>s_attribute</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", s_attribute = "date")
dt <- decode("GERMAPARLMINI", s_attribute = c("date", "speaker"))

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

dispersion

*Dispersion of a query or multiple queries***Description**

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

**Usage**

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

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

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

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

**Arguments**

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

```

dotplot

*dotplot***Description**

dotplot

**Usage**

```

dotplot(.Object, ...)

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

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

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

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

```

**Arguments**

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

---

encoding	<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.

---

enrich	<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>partition_bundle</code> or comp object
<code>...</code>	further parameters

## See Also

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

---

features	<i>Get features by comparison.</i>
----------	------------------------------------

---

## Description

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

## Usage

```
features(x, y, ...)

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

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

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

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

### Arguments

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

### Author(s)

Andreas Blaette

### References

Baker, Paul (2006): *Using Corpora in Discourse Analysis*. London: continuum, p. 121-149 (ch. 6).  
 Manning, Christopher D.; Schuetze, Hinrich (1999): *Foundations of Statistical Natural Language Processing*. MIT Press: Cambridge, Mass., pp. 151-189 (ch. 5).

### Examples

```
use("polmineR")

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

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

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



```

top100 <- subset(terms_kauder, rank_chisquare <= 100)
head(top100)

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

```

---

features-class

*Feature selection by comparison.*


---

## Description

The features-method returns a features-object. Several features-objects can be combined into a features\_bundle-object.

## Usage

```

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

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

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

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

```

## Arguments

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

## Details

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

## Slots

corpus	The CWB corpus the features are derived from, a character vector of length 1.
p_attribute	Object of class character.
encoding	Object of class character.
corpus	Object of class character.
stat	Object of class data.frame.
size_coi	Object of class integer.
size_ref	Object of class integer.

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

method Object of class character statisticalTest used

call Object of class character the call that generated the object

### Author(s)

Andreas Blaette

---

get_template	<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

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

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

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

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

set_template(.Object, ...)

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

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

### Arguments

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

---

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

---

## Description

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

## Usage

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

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

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

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

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

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

## Arguments

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

## Examples

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

---

get_type	<i>Get corpus/partition type.</i>
----------	-----------------------------------

---

## Description

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

## Usage

```
get_type(.Object)

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

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

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

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

## Arguments

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

## Details

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

## Examples

```
use("polmineR")

get_type("GERMAPARLMINI")

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

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

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

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

---

highlight	<i>Highlight tokens in text output.</i>
-----------	-----------------------------------------

---

## Description

Highlight tokens in fulltext 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, character, a kwic object.
...	Terms to be highlighted can be passed in as named character vectors of terms (or regular expressions); the name then needs to be a valid color name.
highlight	A character vector, or a list of character or integer vectors.
regex	Logical, whether character vectors are interpreted as regular expressions.
perl	Logical, whether to use perl-style regular expressions for highlighting when regex is TRUE.
verbose	Logical, whether to output messages.

## Details

If highlight is a character vector, the names of the vector are interpreted as colors. If highlight is a list, the names of the list are considered as colors. Values can be character values or integer values with token ids. Colors are inserted into the output html and need to be digestable for the browser used.

## 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

# use character vector for output, not list
K2 <- highlight(
  K,
  highlight = c(
    green = "pric.",
    red = "reduction",
    red = "decrease",
    orange = "dropped"),
  regex = TRUE
)
if (interactive()) K2

```

---

hits

*Get Hits.*


---

## Description

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

## Usage

```

hits(.Object, ...)

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

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

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

```

```

    verbose = TRUE, ...)

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

```

### Arguments

.Object	a character, partition or partition_bundle 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
s_attribute	s-attributes
p_attribute	p-attribute
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

### 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).

---

hits_class	<i>Hits class.</i>
------------	--------------------

---

### Description

Hits class.

### Usage

```

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

```

### Arguments

x	A hits object.
size	A non-negative integer giving the number of items to choose.

**Slots**

stat a "data.table"  
 corpus a "character" vector  
 query Object of class "character"  
 p\_attribute p-attribute that has been queried  
 encoding encoding of the corpus  
 name name of the object

---

html	<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)

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

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

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

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

```

**Arguments**

object	the object the fulltext output will be based on
...	further parameters that are passed into <code>as.markdown</code>
meta	metadata for output, if NULL (default), the s-attributes defining a partition will be used
cpos	logical, if TRUE (default), all tokens will be wrapped by elements with id attribute indicating corpus positions
verbose	logical, whether to be verbose
cutoff	maximum number of tokens to decode from token stream, passed into <code>as.markdown</code>



charoffset	logical, if TRUE, character offset positions are added to elements embracing tokens
beautify	logical, if TRUE, whitespace before interpunctuation will be removed
height	A character vector that will be inserted into the html as an optional height of a scroll box.
filename	the filename
type	the partition type
i	if object is a kwic-object, the index of the concordance for which the fulltext is to be generated
s_attribute	structural attributes that will be used to define the partition where the match occurred
x	object of class html to print

### Details

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

### Examples

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

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

---

kwc

*KWIC/concordance output.*


---

### Description

Prepare and show concordances / keyword-in-context (kwic).

## Usage

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

## S4 method for signature 'context'
kwic(.Object, s_attributes = 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"),
     s_attributes = getOption("polmineR.meta"), p_attribute = "word",
     boundary = NULL, cpos = TRUE, stoplist = NULL, positivelist = NULL,
     regex = FALSE, verbose = TRUE, ...)

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

## Arguments

<code>.Object</code>	A (length-one) character vector with the name of a CWB corpus, a partition or context object.
<code>...</code>	Further arguments, used to ensure backwards compatibility.
<code>s_attributes</code>	Structural attributes (s-attributes) to include into output table as meta-information.
<code>cpos</code>	Logical, if TRUE, the corpus positions ("cpos") if the hits will be included in the kwic-object that is returned.
<code>verbose</code>	Logical, whether to output progress messages
<code>query</code>	A query, CQP-syntax can be used.
<code>cqp</code>	Either a logical value (TRUE if query is a CQP query), or a function to check whether query is a CQP query or not (defaults to auxiliary function <code>is.query</code> ).
<code>left</code>	Number of tokens to the left of query match.
<code>right</code>	Number of tokens to the right of query match.
<code>p_attribute</code>	The p-attribute, defaults to 'word'.
<code>boundary</code>	If provided, a length-one character vector stating an s-attribute that will be used to check the boundaries of the text.
<code>stoplist</code>	Terms or ids to prevent a concordance from occurring in results.
<code>positivelist</code>	Terms or ids required for a concordance to occur in results
<code>regex</code>	Logical, whether stoplist/positivelist is interpreted as regular expression
<code>progress</code>	Logical, whether to show progress bars.

## Details

The method works with a whole CWB corpus defined by a character vector, and can be applied on a partition- or a context object.

If a `positivelist` is supplied, only concordances will be kept if at least one of the terms from the `positivelist` occurs in the context of the query match. Use argument `regex` if the `positivelist` should be interpreted as regular expressions. Tokens from the `positivelist` will be highlighted in the output table.

## 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

The return value is a `kwic-class` object; the documentation for the class explains the methods applicable to `kwic-class` objects. To read the whole text, see the `read`-method.

## Examples

```
use("polmineR")
kwic("GERMAPARLMINI", "Integration")
kwic(
  "GERMAPARLMINI",
  "Integration", left = 20, right = 20,
  s_attributes = c("date", "speaker", "party")
)
kwic(
  "GERMAPARLMINI",
  '"Integration" [] "(Menschen|Migrant.*|Personen)"', cqp = TRUE,
  left = 20, right = 20,
  s_attributes = c("date", "speaker", "party")
)

kwic(
  "GERMAPARLMINI",
  '"Sehr" "geehrte"', cqp = TRUE,
  boundary = "date"
)

P <- partition("GERMAPARLMINI", date = "2009-11-10")
kwic(P, query = "Integration")
kwic(P, query = '"Sehr" "geehrte"', cqp = TRUE, boundary = "date")
```

---

kwic-class

*kwic (S4 class)*

---

## Description

S4 class for organizing information for kwic/concordance output. A set of standard generics (`show`, `as.character`, `as.data.frame`, `length`, `sample`, `subset`) as well as indexing is implemented to process kwic class objects (see 'Usage'). See section 'Details' for the `enrich`, `view` and `knit_print` methods.

**Usage**

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

## S4 method for signature 'kwic'
knit_print(x, pagelength = getOption("polmineR.pagelength"),
  options = knitr::opts_chunk, ...)

## S4 method for signature 'kwic'
as.character(x, fmt = "<i>%s</i>")

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

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

## 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, s_attributes = NULL, table = FALSE, ...)

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

**Arguments**

<code>object</code>	A kwic class object.
<code>x</code>	A kwic class object.
<code>pagelength</code>	The number of kwic lines displayed per page in the datatables htmlwidget that is returned.
<code>options</code>	Chunk options.
<code>...</code>	Used for backwards compatibility.
<code>fmt</code>	A format string passed into <code>sprintf</code> to format the node of a KWIC display.
<code>i</code>	Single integer value, the kwic line for which the fulltext shall be inspected.
<code>size</code>	An integer, subset size for sampling.
<code>.Object</code>	A kwic class object.
<code>s_attributes</code>	Character vector of s-attributes with metainformation.
<code>table</code>	Logical, whether to turn <code>cpos.data.table</code> into <code>data.frame</code> for output.

## Details

The `knit_print` will be called by knitr when processing code chunks in Rmarkdown documents to include a `htmlwidget` into the resulting html document. It may be necessary to explicitly state `"render=knit_print"` in the chunk options.

The `subset-method` will apply `subset` to the table in the slot `table`, for filtering query results based on metadata (i.e. `s-attributes`) that need to be present.

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 `s_attributes` is a character vector with `s-attributes`, the respective `s-attributes` will be added as columns to the table with concordance lines.

## Slots

`metadata` A character vector with `s-attributes` of the metadata that are to be displayed.

`left` An integer value, words to the left of the query match.

`right` An integer value, words to the right of the query match.

`corpus` Length-one character vector, the CWB corpus.

`cpos` A `data.table` with the columns `"hit_no"`, `"cpos"`, `"position"`, `"word_id"`, `"word"` and `"direction"`.

`table` A `data.frame`, a table with columns `"left"`, `"node"`, `"right"`, and `metadata`, if the object has been enriched.

`encoding` A length-one character vector with the encoding of the corpus.

`labels` A character vector with labels.

`categories` A character vector.

## See Also

The constructor for generating `kwic` objects is the `kwic` method.

## Examples

```
use("polmineR")
K <- kwic("GERMAPARLMINI", "Integration")
length(K)
K[1]
K[1:5]
oil <- kwic("REUTERS", query = "oil")
as.character(oil)
```

---

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

*Labels class.*


---

**Description**

Labels class.

**Arguments**

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

**Fields**

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

choices character vector, a list of choices for labels

expandable whether choices may be expanded (logical)

---

11

*text statistics*


---

**Description**

text statistics

**Usage**

```

ll(.Object, ...)

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

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

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

pmi(.Object)

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

```

**Arguments**

```

.Object      an object
...          further parameters

```

---

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

---

**Description**

Send out a mail with the statistics of an object attached as an xlsx-file.

**Usage**

```

mail(.Object, ...)

## S4 method for signature 'textstat'
mail(.Object, to = getOption("polmineR.email"),
      rows = 1L:min(250L, nrow(.Object)))

## S4 method for signature 'data.frame'
mail(.Object, to = getOption("polmineR.email"),
      filename = tempfile(fileext = ".xlsx"), rows = 1L:min(250L,
nrow(.Object)))

## S4 method for signature 'kwic'
mail(.Object, to = getOption("polmineR.email"),
      rows = 1L:min(250L, nrow(.Object)))

```

**Arguments**

```

.Object      The object to deliver.
...          Further parameters.

```

to	The recipient of the mail message.
rows	The number of rows of the table (if NULL, the whole table will be sent).
filename	The filename of the (temporary) xlsx-file that is generated.

---

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

---

ngrams	<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, p_attribute = "word",
       char = NULL, progress = FALSE, ...)

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



Arguments

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

Examples

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

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

---

ngrams_class	<i>Ngrams class.</i>
--------------	----------------------

---

Description

Ngrams class.

---

noise	<i>detect noise</i>
-------	---------------------

---

Description

detect noise

**Usage**

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

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

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

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

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

**Arguments**

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

**Value**

a list

---

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

---

**Description**

Create a subcorpus and keep it in an object of the `partition` class. If defined, counts are performed for the `p`-attribute defined by the parameter `p_attribute`.

**Usage**

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

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

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

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

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

## S4 method for signature 'context'
partition(.Object, node = TRUE)
```

**Arguments**

<code>.Object</code>	A length-one character-vector, the CWB corpus to be used.
<code>...</code>	Arguments to define partition (see examples).
<code>def</code>	A named list of character vectors of s-attribute values, the names are the s-attributes (see details and examples)
<code>name</code>	A name for the new partition object, defaults to "".
<code>encoding</code>	The 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.
<code>p_attribute</code>	The p-attribute(s) for which a count is performed.
<code>regex</code>	A logical value (defaults to FALSE).
<code>xml</code>	Either 'flat' (default) or 'nested'.
<code>decode</code>	Logical, whether to turn token ids to strings (set FALSE to minimize object size / memory consumption) in data.table with counts.
<code>type</code>	A length-one character vector specifying the type of corpus / partition (e.g. "plpr")
<code>mc</code>	Whether to use multicore (for counting terms).
<code>verbose</code>	Logical, whether to be verbose.
<code>slots</code>	Object slots that will be reported columns of data.frame summarizing partition objects in environment.
<code>node</code>	A logical value, whether to include the node (i.e. query matches) in the region matrix generated when creating a partition from a context-object.

**Details**

If `.Object` is a length-one character vector, a subcorpus/partition for the corpus defined by `.Object` is generated.

If `.Object` is an environment (typically `.GlobalEnv`), the partition objects present in the environment are listed.

If `.Object` is a partition object, a subcorpus of the subcorpus is generated.

If `.Object` is a Corpus object, preparing the partition may work more efficiently than if `.Object` is a length-one character vector.

**Value**

An object of the S4 class `partition`.

**Author(s)**

Andreas Blaette

**See Also**

To learn about the methods available for objects of the class `partition`, see [partition\\_class](#),

**Examples**

```
use("polmineR")
spd <- partition("GERMAPARLMINI", party = "SPD", interjection = "speech")
kauder <- partition("GERMAPARLMINI", speaker = "Volker Kauder", p_attribute = "word")
merkel <- partition("GERMAPARLMINI", speaker = ".*Merkel", p_attribute = "word", regex = TRUE)
s_attributes(merkel, "date")
s_attributes(merkel, "speaker")
merkel <- partition(
  "GERMAPARLMINI", speaker = "Angela Dorothea Merkel",
  date = "2009-11-10", interjection = "speech", p_attribute = "word"
)
merkel <- subset(merkel, !word %in% punctuation)
merkel <- subset(merkel, !word %in% tm::stopwords("de"))

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

---

partition\_bundle

*Generate bundle of partitions.*

---

**Description**

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

**Usage**

```

partition_bundle(.Object, ...)

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

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

## S4 method for signature 'context'
partition_bundle(.Object, node = TRUE)

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

```

**Arguments**

<code>.Object</code>	A partition, a length-one character vector supplying a CWB corpus, or a <code>partition_bundle</code>
<code>...</code>	parameters to be passed into partition-method (see respective documentation)
<code>s_attribute</code>	The s-attribute to vary
<code>values</code>	Values the s-attribute provided shall assume.
<code>prefix</code>	A character vector that will be attached as a prefix to partition names.
<code>mc</code>	Logical, whether to use multicore parallelization.
<code>verbose</code>	Logical, whether to provide progress information.
<code>progress</code>	Logical, whether to show progress bar.
<code>type</code>	The type of partition to generate.
<code>xml</code>	logical
<code>node</code>	A logical value, whether to include the node (i.e. query matches) in the region matrix generated when creating a partition from a context-object.

**Details**

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

**Value**

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

**Author(s)**

Andreas Blaette

**See Also**[partition](#) and [bundle](#)**Examples**

```

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

summary(pb2)

```

---

partition\_bundle-class

*Bundle of partitions (partition\_bundle class).*


---

**Description**

Class and methods to manage bundles of partitions.

flatten may be useful if you have a list of partition\_bundle objects. This function will flatten the data structure and return a partition\_bundle object.

**Usage**

```

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

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

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

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

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

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

```

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

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

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

flatten(object)
```

### Arguments

object	a partition_bundle object
x	a partition_bundle object
name	the name for the new partition
verbose	logical
i	integer index
height	height
...	further parameters
.Object	a partition_bundle object
mc	logical or, if numeric, providing the number of cores
progress	logical
s_attribute	the s-attribute to use

### Details

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

Using brackets can be used to retrieve the count for a token from the partition objects in a partition\_bundle.

### Value

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

### Slots

objects	Object of class list the partitions making up the bundle
corpus	Object of class character the CWB corpus the partition is based on
s_attributes_fixed	Object of class list fixed s-attributes
encoding	Object of class character encoding of the corpus
explanation	Object of class character an explanation of the partition
xml	Object of class character whether the xml is flat or nested
call	Object of class character the call that generated the partition_bundle

**Author(s)**

Andreas Blaette

---

partition\_class

*Partition class and methods.*

---

**Description**

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

**Usage**

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

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

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

is.partition(x)

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

## S4 method for signature 'partition'
as.regions(x)
```

**Arguments**

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



## Details

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

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

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

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

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

## Slots

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

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

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

**s\_attributes** A named list with the s-attributes specifying the partition.

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

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

**annotations** Object of class list.

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

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

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

**strucs** Object of class integer, the strucs defining the partition.

**p\_attribute** Object of class character indicating the p\_attribute of the count in slot stat.

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

**s\_attribute\_strucs** Object of class character the base node

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

## Author(s)

Andreas Blaette

## See Also

The partition-class inherits from the [textstat-class](#), see respective documentation to learn more.

polmineR

*polmineR-package***Description**

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

**Usage**

```
polmineR()
```

**Details**

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

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

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

**Author(s)**

Andreas Blaette ([andreas.blaette@uni-due.de](mailto:andreas.blaette@uni-due.de))

**References**

Jockers, Matthew L. (2014): *Text Analysis with R for Students of Literature*. Cham et al: Springer.  
 Baker, Paul (2006): *Using Corpora in Discourse Analysis*. London: continuum.

**Examples**

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

# Core methods applied to corpus

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

# Core methods applied to partition

kuwait <- partition("REUTERS", places = "kuwait", regex = TRUE)
```

```

count(kuwait, query = "oil")
dispersion(kuwait, query = "oil", s_attribute = "id")
kwic(kuwait, query = "oil", meta = "id")
cooccurrences(kuwait, query = "oil")

# Go back to full text

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

# Generate term document matrix

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

```

---

p_attributes	<i>Get p-attributes.</i>
--------------	--------------------------

---

## Description

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

## Usage

```

p_attributes(.Object, ...)

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

```

## Arguments

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

## References

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

## Examples

```

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

```

read

*Display full text.***Description**

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

**Usage**

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

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

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

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

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

## S4 method for signature 'kwic'
read(.Object, i = NULL,
     type = registry_get_properties(corpus(.Object))["type"])

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

**Arguments**

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

<code>col</code>	column of <code>data.table</code> with terms to be highlighted
<code>partition_bundle</code>	a <code>partition_bundle</code> object
<code>def</code>	a named list used to define a partition (names are s-attributes, vectors are values of s-attributes)
<code>i</code>	if <code>.Object</code> is an object of the classes <code>kwic</code> or <code>hits</code> , the <code>ith</code> <code>kwic</code> line or hit to derive a partition to be inspected from
<code>type</code>	the partition type, see documentation for <code>partition-method</code>

### Details

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

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

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

### See Also

For concordances / a keyword-in-context display, see [kwic](#).

### Examples

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

---

regions	<i>Regions of a CWB corpus.</i>
---------	---------------------------------

---

### 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
...	Further arguments.
values	values to assign to a column that will be added

**Details**

The `as.regions`-method coerces objects to a regions-object.

**Slots**

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

`corpus` the CWB corpus (character vector length 1)

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

**Examples**

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

---

registry

*Get session registry directory.*


---

**Description**

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

**Usage**

```
registry()
```

**Examples**

```
registry()
```

---

registry_get_name	<i>Evaluate registry file.</i>
-------------------	--------------------------------

---

### Description

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

### Usage

```
registry_get_name(corpus, registry = Sys.getenv("CORPUS_REGISTRY"))
registry_get_id(corpus, registry = Sys.getenv("CORPUS_REGISTRY"))
registry_get_home(corpus, registry = Sys.getenv("CORPUS_REGISTRY"))
registry_get_info(corpus, registry = Sys.getenv("CORPUS_REGISTRY"))
registry_get_encoding(corpus, registry = Sys.getenv("CORPUS_REGISTRY"))
registry_get_p_attributes(corpus, registry = Sys.getenv("CORPUS_REGISTRY"))
registry_get_s_attributes(corpus, registry = Sys.getenv("CORPUS_REGISTRY"))
registry_get_properties(corpus, registry = Sys.getenv("CORPUS_REGISTRY"))
```

### Arguments

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

### Details

An appendix to the 'Corpus Encoding Tutorial' ([http://cwb.sourceforge.net/files/CWB\\_Encoding\\_Tutorial.pdf](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 = registry(), 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)
```

renamed

*Renamed Functions***Description**

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

**Usage**

```
sAttributes(...)
pAttributes(...)
getTokenStream(...)
getTerms(...)
getEncoding(...)
partitionBundle(...)
as.partitionBundle(...)
setTemplate(...)
getTemplate(...)
```



## Arguments

... argument that are passed to the renamed function

---

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, s_attribute = NULL, verbose = TRUE, ...)

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

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

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

## Arguments

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

## Details

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

## Value

an integer vector if s\_attribute is NULL, a data.table 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", s_attribute = "date")
size("GERMAPARLMINI", s_attribute = c("date", "party"))

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

---

store	<i>Store objects as Excel-file.</i>
-------	-------------------------------------

---

## Description

Store objects as Excel-file.

## Usage

```
store(.Object, ...)

## S4 method for signature 'textstat'
store(.Object, filename = tempfile(fileext = ".xlsx"),
      rows = 1L:nrow(.Object))

## S4 method for signature 'data.frame'
store(.Object, filename = tempfile(fileext = ".xlsx"),
      rows = 1L:nrow(.Object))

## S4 method for signature 'kwc'
store(.Object, filename = tempfile(fileext = ".xlsx"),
      rows = 1L:nrow(.Object))
```

## Arguments

.Object	An object that can be processed.
...	Further arguments.
filename	Name of the file to write.
rows	The rows of the table to export.

---

subcorpus	<i>Virtual class subcorpus</i>
-----------	--------------------------------

---

## Description

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

## Usage

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

## Arguments

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

## Details

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

## Examples

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

---

s_attributes	<i>Get s-attributes.</i>
--------------	--------------------------

---

## Description

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

**Usage**

```
s_attributes(.Object, ...)

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

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

**Arguments**

<code>.Object</code>	either a partition object or a character vector specifying a CWB corpus
<code>...</code>	to maintain backward compatibility, of argument <code>sAttribute</code> is used
<code>s_attribute</code>	name of a specific s-attribute
<code>unique</code>	logical, whether to return unique values only
<code>regex</code>	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 `s_attributes`-method: If the `s_attribute` parameter is `NULL` (default), the return value is a character vector with all s-attributes present in a corpus.

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

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

**Value**

a character vector

**Examples**

```
use("polmineR")

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

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

---

tempcorpus	<i>create a tempcorpus</i>
------------	----------------------------

---

**Description**

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

---

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

**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

---

terms	<i>Get terms in partition or corpus.</i>
-------	------------------------------------------

---

**Description**

Get terms in partition or corpus.

**Usage**

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

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

**Arguments**

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

**Examples**

```

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

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

```

---

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

---

**Description**

The textstat-class (technically an S4 class) serves as a superclass for the classes features, context, and partition. Usually, the class will not be used directly. It offers a set of standard generic methods (such as head, tail, dim, nrow, colnames) its childs inherit. The core feature of textstat and its childs is a data.table in the slot stat for keeping data on text statistics of a corpus, or a partition.

**Usage**

```

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

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

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

## 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'
p_attributes(.Object)

## S4 method for signature 'textstat'
knit_print(x,
  pagelength = getOption("polmineR.pagelength"),
  options = knitr::opts_chunk, ...)

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

### Arguments

<code>x</code>	A textstat object.
<code>value</code>	A character vector to assign as name to slot name of a textstat class object.
<code>digits</code>	no of digits
<code>by</code>	Column that will serve as the key for sorting.
<code>decreasing</code>	Logical, whether to return decreasing order.
<code>object</code>	a textstat object
<code>...</code>	Further arguments.
<code>e1</code>	A textstat object.
<code>e2</code>	Another textstat object.
<code>.Object</code>	A textstat object.
<code>pagelength</code>	The number of kwic lines displayed per page in the datatables htmlwidget that is returned.
<code>options</code>	Chunk options.

### Details

A head-method will return the first rows of the `data.table` in the `stat`-slot. Use argument `n` to specify the number of rows.

A tail-method will return the last rows of the `data.table` in the `stat`-slot. Use argument `n` to specify the number of rows.

The methods `dim`, `nrow` and `ncol` will return information on the dimensions, the number of rows, or the number of columns of the `data.table` in the `stat`-slot, respectively.

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`.

The `colnames`-method will return the column names of the `data-table` in the slot `stat`.

The methods `as.data.table`, and `as.data.frame` will extract the `data.table` in the slot `stat` as a `data.table`, or `data.frame`, respectively.

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

**Slots**

`p_attribute` Object of class character, p-attribute of the query.  
`corpus` A corpus specified by a length-one character vector.  
`stat` A `data.table` with statistical information.  
`name` The name of the object.  
`encoding` A length-one character vector, the encoding of the corpus.

**Examples**

```
use("polmineR")
P <- partition("GERMAPARLMINI", date = ".*", p_attribute = "word", regex = TRUE)
y <- cooccurrences(P, query = "Arbeit")

# Standard generic methods known from data.frames work for objects inheriting
# from the textstat class

head(y)
tail(y)
nrow(y)
ncol(y)
dim(y)
colnames(y)

# Use brackets for indexing

y[1:25]
y[,c("word", "l1")]
y[1:25, "word"]
y[1:25][["word"]]
y[which(y[["word"]] %in% c("Arbeit", "Sozial"))]
y[ y[["word"]] %in% c("Arbeit", "Sozial") ]
```

---

tooltips

---

*Add tooltips to text output.*


---

**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())

## S4 method for signature 'kwic'
tooltips(.Object, tooltips, regex = FALSE, ...)
```



## Arguments

<code>.Object</code>	A html or character object with html.
<code>tooltips</code>	A named list of character vectors, the names need to match colors in the list provided to <code>param highlight</code> . The value of the character vector is the tooltip to be displayed.
<code>...</code>	Further arguments are interpreted as assignments of tooltips to tokens.
<code>regex</code>	Logical, whether character vector values of argument <code>tooltips</code> are interpreted as regular expressions.

## Examples

```
use("polmineR")

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

# Using the tooltips-method in a pipe ...
if (require("magrittr")){
  P %>%
    html() %>%
    highlight(yellow = c("barrels", "oil", "gas")) %>%
    tooltips(list(yellow = "energy"))
}
```

---

trim	<i>trim an object</i>
------	-----------------------

---

## Description

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

## Usage

```
trim(object, ...)

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

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

punctuation
```

**Arguments**

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

**Format**

An object of class character of length 13.

**Author(s)**

Andreas Blaette

---

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

---

**Description**

S4 method for context object to perform t-test

**Usage**

```
t_test(.Object)

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

**Arguments**

.Object	a context or features object
---------	------------------------------

---

use	<i>Add corpora in R data packages to session registry.</i>
-----	------------------------------------------------------------

---

**Description**

Use CWB indexed corpora in R data packages by adding registry file to session registry.

**Usage**

```
use(pkg, lib.loc = .libPaths(), verbose = TRUE)
```

## Arguments

<code>pkg</code>	A package including at least one CWB indexed corpus.
<code>lib.loc</code>	A character vector with path names of R libraries.
<code>verbose</code>	Logical, whether to output status messages.

## Details

`pkg` is expected to be an installed data package that includes CWB indexed corpora. The use-function will add the registry files describing the corpus (or the corpora) to the session registry directory and adjust the path pointing to the data in the package.

The registry files within the package are assumed to be in the subdirectory `./extdata/cwb/registry` of the installed package. The data directories for corpora are assumed to be in a subdirectory named after the corpus (lower case) in the package subdirectory `./extdata/cwb/indexed_corpora/`. When adding a corpus to the registry, templates for formatting fulltext output are reloaded.

## See Also

To get the session registry registry, see [registry](#); to reset the registry, see [registry\\_reset](#).

## Examples

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

---

<code>view</code>	<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

*Apply Weight to Matrix***Description**

Apply Weight to Matrix

**Usage**

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

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

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

## S4 method for signature 'count'
weigh(.Object, with)

## S4 method for signature 'count_bundle'
weigh(.Object, with)
```

**Arguments**

<code>.Object</code>	A matrix, or a count-object.
<code>...</code>	further parameters
<code>method</code>	the kind of weight to apply
<code>with</code>	A <code>data.table</code> used to weigh p-attributes. A column 'weight' with term weights is required, and columns with the p-attributes of <code>.Object</code> for matching.

**Examples**

```
## Not run:
library(data.table)
if (require("zoo") && require("devtools") && require("magrittr")){

# Source in function 'get_sentiws' from a GitHub gist
gist_url <- file.path(
  "gist.githubusercontent.com",
  "PolMine",
  "70eeb095328070c18bd00ee087272adf",
  "raw",
  "c2eee2f48b11e6d893c19089b444f25b452d2adb",
  "sentiws.R"
)

devtools::source_url(sprintf("https://%s", gist_url))
SentiWS <- get_sentiws()

# Do the statistical word context analysis
use("GermaParl")
```

```

options("polmineR.left" = 10L)
options("polmineR.right" = 10L)
df <- context("GERMAPARL", query = "Islam", p_attribute = c("word", "pos")) %>%
  partition_bundle(node = FALSE) %>%
  set_names(s_attributes(., s_attribute = "date")) %>%
  weigh(with = SentiWS) %>%
  summary()

# Aggregate by year
df[["year"]] <- as.Date(df[["name"]]) %>% format("%Y-01-01")
df_year <- aggregate(df[,c("size", "positive_n", "negative_n")], list(df[["year"]]), sum)
colnames(df_year)[1] <- "year"

# Use shares instead of absolute counts
df_year$negative_share <- df_year$negative_n / df_year$size
df_year$positive_share <- df_year$positive_n / df_year$size

# Turn it into zoo object, and plot it
Z <- zoo(
  x = df_year[, c("positive_share", "negative_share")],
  order.by = as.Date(df_year[, "year"])
)
plot(
  Z, ylab = "polarity", xlab = "year",
  main = "Word context of 'Islam': Share of positive/negative vocabulary",
  cex = 0.8,
  cex.main = 0.8
)

# Note that we can use the kwic-method to check for the validity of our findings
words_positive <- SentiWS[weight > 0][["word"]]
words_negative <- SentiWS[weight < 0][["word"]]
kwic("GERMAPARL", query = "Islam", positivelist = c(words_positive, words_negative)) %>%
  highlight(lightgreen = words_positive, orange = words_negative) %>%
  tooltips(setNames(SentiWS[["word"]], SentiWS[["weight"]]))
}

## End(Not run)

```

# Index

- \*Topic **datasets**
  - Corpus, [19](#)
  - CQI.super, [25](#)
  - trim, [73](#)
- \*Topic **package**
  - polmineR, [58](#)
- \*Topic **textstatistics**
  - chisquare, [12](#)
- +, bundle, bundle-method (bundle-class), [10](#)
- +, bundle, textstat-method (bundle-class), [10](#)
- +, partition\_bundle, ANY-method (partition\_bundle-class), [54](#)
- +, partition\_bundle, partition-method (partition\_bundle-class), [54](#)
- +, partition\_bundle, partition\_bundle-method (partition\_bundle-class), [54](#)
- +, partition\_bundle-method (partition\_bundle-class), [54](#)
- +, textstat, textstat-method (textstat-class), [70](#)
- [, context, ANY, ANY, ANY-method (context-class), [14](#)
- [, context-method (context-class), [14](#)
- [, context\_bundle, ANY, ANY, ANY-method (context\_bundle-class), [16](#)
- [, context\_bundle-method (context\_bundle-class), [16](#)
- [, kwic, ANY, ANY, ANY-method (kwic-class), [43](#)
- [, kwic-method (kwic-class), [43](#)
- [, partition, ANY, ANY, ANY-method (partition\_class), [56](#)
- [, partition-method (partition\_class), [56](#)
- [, partition\_bundle, ANY, ANY, ANY-method (partition\_bundle-class), [54](#)
- [, partition\_bundle-method (partition\_bundle-class), [54](#)
- [, textstat, ANY, ANY, ANY-method (textstat-class), [70](#)
- [[, bundle-method (bundle-class), [10](#)
- [[, context-method (context-class), [14](#)
- [[, context\_bundle-method (context\_bundle-class), [16](#)
- [[, partition\_bundle-method (partition\_bundle-class), [54](#)
- [[, textstat-method (textstat-class), [70](#)
- aggregate, subcorpus-method (subcorpus), [67](#)
- as.bundle (textstat-class), [70](#)
- as.bundle, list-method (bundle-class), [10](#)
- as.bundle, textstat-method (bundle-class), [10](#)
- as.character, kwic-method (kwic-class), [43](#)
- as.corpusEnc (encodings), [30](#)
- as.cqp (cqp), [26](#)
- as.data.frame, cooccurrences\_bundle-method (cooccurrences-class), [18](#)
- as.data.frame, kwic-method (kwic-class), [43](#)
- as.data.frame, textstat-method (textstat-class), [70](#)
- as.data.table, bundle-method (bundle-class), [10](#)
- as.data.table, regions-method (regions), [61](#)
- as.data.table, textstat-method (textstat-class), [70](#)
- as.DataTables, context-method (context-class), [14](#)
- as.DataTables, textstat-method (textstat-class), [70](#)
- as.DocumentTermMatrix (as.TermDocumentMatrix), [6](#)
- as.DocumentTermMatrix, bundle-method (as.TermDocumentMatrix), [6](#)
- as.DocumentTermMatrix, character-method (as.TermDocumentMatrix), [6](#)
- as.DocumentTermMatrix, context-method (as.TermDocumentMatrix), [6](#)
- as.DocumentTermMatrix, partition\_bundle-method (as.TermDocumentMatrix), [6](#)
- as.list, bundle-method (bundle-class), [10](#)
- as.markdown, [3](#)

- as.markdown,partition-method  
(as.markdown), 3
- as.markdown,plpr\_partition-method  
(as.markdown), 3
- as.matrix,bundle-method (bundle-class),  
10
- as.matrix,context\_bundle-method  
(context), 13
- as.matrix,partition\_bundle-method  
(partition\_bundle-class), 54
- as.nativeEnc (encodings), 30
- as.partition\_bundle (partition\_class),  
56
- as.partition\_bundle,list-method  
(partition\_bundle-class), 54
- as.partition\_bundle,partition-method  
(partition\_class), 56
- as.partitionBundle (renamed), 64
- as.regions (regions), 61
- as.regions,context-method  
(context-class), 14
- as.regions,partition-method  
(partition\_class), 56
- as.sparseMatrix, 4
- as.sparseMatrix,bundle-method  
(as.sparseMatrix), 4
- as.sparseMatrix,simple\_triplet\_matrix-method  
(as.sparseMatrix), 4
- as.sparseMatrix,TermDocumentMatrix-method  
(as.sparseMatrix), 4
- as.speeches, 5
- as.TermDocumentMatrix, 6
- as.TermDocumentMatrix,bundle-method  
(as.TermDocumentMatrix), 6
- as.TermDocumentMatrix,character-method  
(as.TermDocumentMatrix), 6
- as.TermDocumentMatrix,context-method  
(as.TermDocumentMatrix), 6
- as.TermDocumentMatrix,partition\_bundle-method  
(as.TermDocumentMatrix), 6
- as.utf8 (encodings), 30
- as.VCorpus, 8
- as.VCorpus,partition\_bundle-method  
(as.VCorpus), 8
  
- barplot,partition\_bundle-method  
(partition\_bundle-class), 54
- blapply, 8
- blapply,bundle-method (blapply), 8
- blapply,list-method (blapply), 8
- blapply,vector-method (blapply), 8
- browse, 9
- browse,cooccurrences-method (browse), 9
- browse,html-method (browse), 9
- browse,kwic-method (browse), 9
- browse,partition-method (browse), 9
- browse,press\_partition-method (browse),  
9
- browse,textstat-method (browse), 9
- bundle, 54
- bundle (bundle-class), 10
- bundle-class, 10
  
- chisquare, 12
- chisquare,context-method (chisquare), 12
- chisquare,textstat-method (chisquare),  
12
- colnames,textstat-method  
(textstat-class), 70
- context, 13
- context,character-method (context), 13
- context,cooccurrences-method (context),  
13
- context,partition-method (context), 13
- context,partition\_bundle-method  
(context), 13
- context-class, 14
- context\_bundle-class, 16
- cooccurrences, 16
- cooccurrences,character-method  
(cooccurrences), 16
- cooccurrences,context-method  
(cooccurrences), 16
- cooccurrences,Corpus-method  
(cooccurrences), 16
- cooccurrences,partition-method  
(cooccurrences), 16
- cooccurrences,partition\_bundle-method  
(cooccurrences), 16
- cooccurrences-class, 18
- cooccurrences\_bundle  
(cooccurrences-class), 18
- cooccurrences\_bundle-class  
(cooccurrences-class), 18
- cooccurrences\_reshaped-class  
(cooccurrences-class), 18
- Corpus, 19
- corpus, 20
- corpus,bundle-method (corpus), 20
- corpus,kwic-method (corpus), 20
- corpus,missing-method (corpus), 20
- corpus,textstat-method (corpus), 20
- count, 21
- count,character-method (count), 21
- count,context-method (context-class), 14
- count,Corpus-method (count), 21

- count,partition-method (count), 21
- count,partition\_bundle-method (count), 21
- count,vector-method (count), 21
- count-class (count\_class), 23
- count-method (count), 21
- count\_bundle-class (count\_class), 23
- count\_class, 23
- cpos, 24
- cpos,character-method (cpos), 24
- cpos,hits-method (cpos), 24
- cpos,matrix-method (cpos), 24
- cpos,partition-method (cpos), 24
- cpos,tempcorpus-method (cpos), 24
- CQI (CQI.super), 25
- CQI.super, 25
- cqp, 26
- decode, 27
- decode,character-method (decode), 27
- dim,textstat-method (textstat-class), 70
- dispersion, 28, 65
- dispersion,character-method (dispersion), 28
- dispersion,hits-method (dispersion), 28
- dispersion,partition-method (dispersion), 28
- dotplot, 29
- dotplot,features-method (dotplot), 29
- dotplot,features\_ngrams-method (dotplot), 29
- dotplot,partition-method (dotplot), 29
- dotplot,textstat-method (dotplot), 29
- encoding, 30
- encoding,bundle-method (encoding), 30
- encoding,textstat-method (encoding), 30
- encoding<- (encoding), 30
- encodings, 30
- enrich, 31
- enrich,context-method (context-class), 14
- enrich,kwic-method (kwic-class), 43
- enrich,partition-method (partition\_class), 56
- enrich,partition\_bundle-method (partition\_bundle-class), 54
- enrich-method (enrich), 31
- export (partition\_class), 56
- export,partition-method (partition\_class), 56
- features, 31
- features,count-method (features), 31
- features,ngrams-method (features), 31
- features,partition-method (features), 31
- features,partition\_bundle-method (features), 31
- features-class, 33
- features\_bundle-class (features-class), 33
- features\_cooccurrences-class (features-class), 33
- features\_ngrams-class (features-class), 33
- flatten (partition\_bundle-class), 54
- get\_template, 34
- get\_template,character-method (get\_template), 34
- get\_template,missing-method (get\_template), 34
- get\_template,partition-method (get\_template), 34
- get\_token\_stream, 35
- get\_token\_stream,character-method (get\_token\_stream), 35
- get\_token\_stream,matrix-method (get\_token\_stream), 35
- get\_token\_stream,numeric-method (get\_token\_stream), 35
- get\_token\_stream,partition-method (get\_token\_stream), 35
- get\_token\_stream,regions-method (get\_token\_stream), 35
- get\_type, 36
- get\_type,character-method (get\_type), 36
- get\_type,Corpus-method (get\_type), 36
- get\_type,partition-method (get\_type), 36
- get\_type,partition\_bundle-method (get\_type), 36
- getEncoding (renamed), 64
- getTemplate (renamed), 64
- getTerms (renamed), 64
- getTokenStream (renamed), 64
- head,context-method (context-class), 14
- head,textstat-method (textstat-class), 70
- highlight, 37
- highlight,character-method (highlight), 37
- highlight,html-method (highlight), 37
- highlight,kwic-method (highlight), 37
- hist,count-method (count\_class), 23
- hits, 38, 65



- hits, character-method (hits), 38
- hits, context-method (hits), 38
- hits, partition-method (hits), 38
- hits, partition\_bundle-method (hits), 38
- hits\_class (hits\_class), 39
- hits\_class, 39
- html, 40
- html, character-method (html), 40
- html, kwic-method (html), 40
- html, partition-method (html), 40
- html, partition\_bundle-method (html), 40
  
- is.cqp (cqp), 26
- is.partition (partition\_class), 56
  
- knit\_print, kwic-method (kwic-class), 43
- knit\_print, textstat-method (textstat-class), 70
- kwic, 41, 45, 61
- kwic, character-method (kwic), 41
- kwic, context-method (kwic), 41
- kwic, partition-method (kwic), 41
- kwic-class, 43
  
- label, 45
- label, kwic-method (label), 45
- label<- (label), 45
- Labels (Labels-class), 46
- Labels-class, 46
- length, bundle-method (bundle-class), 10
- length, context-method (context-class), 14
- length, count-method (count\_class), 23
- length, kwic-method (kwic-class), 43
- ll, 46
- ll, context-method (ll), 46
- ll, cooccurrences-method (ll), 46
- ll, features-method (ll), 46
  
- mail, 47
- mail, data.frame-method (mail), 47
- mail, kwic-method (mail), 47
- mail, textstat-method (mail), 47
- mail-method (mail), 47
- means, 48
- means, DocumentTermMatrix-method (means), 48
- merge, partition\_bundle-method (partition\_bundle-class), 54
  
- name (textstat-class), 70
- name, textstat-method (textstat-class), 70
  
- name<- (textstat-class), 70
- name<- , bundle, character-method (bundle-class), 10
- name<- , textstat, character-method (textstat-class), 70
- names, bundle-method (bundle-class), 10
- names, partition\_bundle-method (partition\_bundle-class), 54
- names, textstat-method (textstat-class), 70
- names<- , bundle, vector-method (bundle-class), 10
- ncol, textstat-method (textstat-class), 70
- ngrams, 48
- ngrams, partition-method (ngrams), 48
- ngrams, partition\_bundle-method (ngrams), 48
- ngrams-class (ngrams\_class), 49
- ngrams\_class, 49
- noise, 49
- noise, character-method (noise), 49
- noise, DocumentTermMatrix-method (noise), 49
- noise, TermDocumentMatrix-method (noise), 49
- noise, textstat-method (noise), 49
- nrow, textstat-method (textstat-class), 70
  
- p\_attributes, 59
- p\_attributes, character-method (p\_attributes), 59
- p\_attributes, context-method (context-class), 14
- p\_attributes, partition-method (partition\_class), 56
- p\_attributes, textstat-method (textstat-class), 70
- partition, 50, 54
- partition, character-method (partition), 50
- partition, context-method (partition), 50
- partition, Corpus-method (partition), 50
- partition, environment-method (partition), 50
- partition, partition-method (partition), 50
- partition-class (partition\_class), 56
- partition\_bundle, 52
- partition\_bundle, character-method (partition\_bundle), 52

- partition\_bundle, context-method (partition\_bundle), 52
- partition\_bundle, environment-method (partition\_bundle-class), 54
- partition\_bundle, partition-method (partition\_bundle), 52
- partition\_bundle, partition\_bundle-method (partition\_bundle), 52
- partition\_bundle-class, 54
- partition\_class, 31, 52, 56
- partitionBundle (renamed), 64
- pAttributes (renamed), 64
- plpr\_partition-class (partition\_class), 56
- pmi (ll), 46
- pmi, context-method (ll), 46
- polmineR, 58
- polmineR-package (polmineR), 58
- press\_partition-class (partition\_class), 56
- print.html (html), 40
- punctuation (trim), 73
- read, 43, 60
- read, data.table-method (read), 60
- read, hits-method (read), 60
- read, kwic-method (read), 60
- read, partition-method (read), 60
- read, partition\_bundle-method (read), 60
- read, regions-method (read), 60
- regions, 61
- regions-class (regions), 61
- registry, 62, 75
- registry\_get\_encoding (registry\_get\_name), 63
- registry\_get\_home (registry\_get\_name), 63
- registry\_get\_id (registry\_get\_name), 63
- registry\_get\_info (registry\_get\_name), 63
- registry\_get\_name, 63
- registry\_get\_p\_attributes (registry\_get\_name), 63
- registry\_get\_properties (registry\_get\_name), 63
- registry\_get\_s\_attributes (registry\_get\_name), 63
- registry\_reset, 63, 75
- renamed, 64
- round, textstat-method (textstat-class), 70
- rownames, textstat-method (textstat-class), 70
- s\_attributes, 67
- s\_attributes, character-method (s\_attributes), 67
- s\_attributes, partition-method (s\_attributes), 67
- s\_attributes, partition\_bundle-method (partition\_bundle-class), 54
- sample, bundle-method (bundle-class), 10
- sample, context-method (context-class), 14
- sample, hits-method (hits\_class), 39
- sample, kwic-method (kwic-class), 43
- sAttributes (renamed), 64
- set\_template (get\_template), 34
- set\_template, character-method (get\_template), 34
- set\_template, missing-method (get\_template), 34
- setTemplate (renamed), 64
- show, context-method (context-class), 14
- show, context\_bundle-method (context\_bundle-class), 16
- show, cooccurrences-method (cooccurrences-class), 18
- show, features-method (features-class), 33
- show, html-method (html), 40
- show, kwic-method (kwic-class), 43
- show, partition-method (partition\_class), 56
- show, partition\_bundle-method (partition\_bundle-class), 54
- show, textstat-method (textstat-class), 70
- size, 65
- size, character-method (size), 65
- size, DocumentTermMatrix-method (size), 65
- size, partition-method (size), 65
- size, TermDocumentMatrix-method (size), 65
- sort, textstat-method (textstat-class), 70
- split (partition\_class), 56
- split, partition-method (partition\_class), 56
- store, 66
- store, data.frame-method (store), 66
- store, kwic-method (store), 66
- store, textstat-method (store), 66
- subcorpus, 67
- subcorpus-class (subcorpus), 67

- subset, bundle-method (bundle-class), 10
- subset, kwic-method (kwic-class), 43
- subset, textstat-method
  - (textstat-class), 70
- summary, context-method (context-class), 14
- summary, context\_bundle-method
  - (context\_bundle-class), 16
- summary, features-method
  - (features-class), 33
- summary, features\_bundle-method
  - (features-class), 33
- summary, partition-method
  - (partition\_class), 56
- summary, partition\_bundle-method
  - (partition\_bundle-class), 54
  
- t\_test, 74
- t\_test, context-method (t\_test), 74
- tail, textstat-method (textstat-class), 70
- tempcorpus, 69
- tempcorpus-class (tempcorpus\_class), 69
- tempcorpus\_class, 69
- terms, 69
- terms, character-method (terms), 69
- terms, partition-method (terms), 69
- textstat-class, 70
- tooltips, 72
- tooltips, character-method (tooltips), 72
- tooltips, html-method (tooltips), 72
- tooltips, kwic-method (tooltips), 72
- trim, 73
- trim, context-method (context-class), 14
- trim, DocumentTermMatrix-method (trim), 73
- trim, TermDocumentMatrix-method (trim), 73
- trim-method (trim), 73
  
- unique, bundle-method (bundle-class), 10
- use, 64, 74
  
- view, 75
- view, cooccurrences-method
  - (cooccurrences-class), 18
- view, cooccurrences\_resaped-method
  - (cooccurrences-class), 18
- view, features-method (features-class), 33
- view, kwic-method (kwic-class), 43
- view, textstat-method (textstat-class), 70
  
- weigh, 76
- weigh, count-method (weigh), 76
- weigh, count\_bundle-method (weigh), 76
- weigh, DocumentTermMatrix-method
  - (weigh), 76
- weigh, TermDocumentMatrix-method
  - (weigh), 76