

# Package ‘polmineR’

July 18, 2017

**Type** Package

**Title** Toolkit for Corpus Analysis

**Version** 0.7.4

**Date** 2017-07-17

**Author** Andreas Blaette

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

**Imports** methods, R6, data.table, slam, Matrix, tm, DT, utils, jsonlite

**Suggests** pbapply, parallel, polmineR.Rcpp, europarl.en, rcqp,  
doParallel, markdown, htmltools, xml2, xlsx, sendmailR, knitr,  
rJava, shiny, testthat

**Additional\_repositories** <http://polmine.sowi.uni-due.de/packages>

**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>

**License** GPL-3

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

**Collate** 'CQI.R' 'CQI.Rcpp.R' 'CQI.cqpserver.R' 'CQI.perl.R'  
'CQI.rcqp.R' 'Corpus\_class.R' 'Labels.R' 'Partition.R'  
'generics.R' 'textstat\_class.R' 'polmineR\_package.R'  
'partition\_class.R' 'Regions\_class.R' 'RegistryFile.R'  
'TermDocumentMatrix\_methods.R' 'Textstat.R'  
'TokenStream\_class.R' 'adjustEncoding\_method.R'  
'aggregate\_method.R' 'bundle\_class.R' 'features\_class.R'

```
'context_class.R' 'partitionBundle_class.R'
'as.DocumentTermMatrix_method.R' 'as.VCorpus_method.R'
'as.markdown_method.R' 'contextBundle_class.R'
'cooccurrences_class.R' 'as.sparseMatrix_method.R'
'as.speeches_method.R' 'blapply_method.R' 'kwic_method.R'
'kwic_class.R' 'browse_method.R' 'chisquare_method.R'
'context_method.R' 'cooccurrences_method.R' 'corpus_method.R'
'dispersion_class.R' 'count_method.R' 'tempcorpus.R'
'cpos_method.R' 'decode_method.R' 'hits_class.R'
'dispersion_method.R' 'divide_method.R' 'dotplot_method.R'
'encode_method.R' 'encoding_method.R' 'enrich_method.R'
'ngrams_method.R' 'features_method.R' 'freq_method.R'
'getEncoding_method.R' 'getTerms_method.R'
'getTokenStream_method.R' 'highlight_method.R' 'html_method.R'
'html_methods.R' 'install.corpus_function.R' 'label_method.R'
'Il_method.R' 'mail_method.R' 'matches_method.R'
'means_method.R' 'noise_method.R' 'pAttributes_method.R'
'partitionBundle_method.R' 'sAttributes2cpo_method.R'
'partition_method.R' 'plot_method.R' 'pmi_method.R'
'read_method.R' 'registry.R' 'reindex_method.R' 'rustyard.R'
'sAttributes_method.R' 'scatterplot_method.R' 'size_method.R'
'tTest.R' 'templates_method.R' 'terms_method.R' 'trim_method.R'
'use_function.R' 'utils.R' 'view_method.R' 'weigh_method.R'
'zzz.R'
```

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2017-07-17 22:42:23 UTC

## R topics documented:

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

cooccurrencesReshaped . . . . .	21
Corpus . . . . .	21
corpus . . . . .	22
count . . . . .	22
cpos . . . . .	24
CQI.super . . . . .	25
cqp . . . . .	26
cqpserver . . . . .	27
decode . . . . .	27
dispersion . . . . .	28
dispersion-class . . . . .	30
divide . . . . .	31
dotplot . . . . .	31
encode . . . . .	32
encoding . . . . .	33
encodings . . . . .	33
enrich . . . . .	34
features,partition-method . . . . .	34
features-class . . . . .	36
flatten . . . . .	37
getEncoding . . . . .	38
getObjects . . . . .	38
getSlot . . . . .	39
getTemplate . . . . .	39
getTerms . . . . .	40
getTokenStream . . . . .	40
highlight . . . . .	41
hits-class . . . . .	42
html . . . . .	44
install.corpus . . . . .	45
kwic . . . . .	46
kwic-class . . . . .	48
label . . . . .	49
Labels-class . . . . .	50
ll . . . . .	51
mail . . . . .	51
matches . . . . .	52
means . . . . .	53
name . . . . .	54
ngrams-class . . . . .	54
noise . . . . .	55
partition . . . . .	56
partitionBundle . . . . .	59
partitionBundle-class . . . . .	60
partition_class . . . . .	62
pAttribute . . . . .	64
pAttributes . . . . .	64
read . . . . .	65

Regions-class . . . . .	67
RegistryFile-class . . . . .	67
resetRegistry . . . . .	68
sAttributes,character-method . . . . .	69
scatterplot . . . . .	70
size . . . . .	71
split,partition-method . . . . .	72
tempcorpus-class . . . . .	73
TermDocumentMatrix . . . . .	73
terms-partition-method . . . . .	74
textstat-class . . . . .	74
TokenStream-class . . . . .	76
trim . . . . .	77
tTest . . . . .	78
use . . . . .	78
view . . . . .	79
weigh . . . . .	79
<b>Index</b>	<b>80</b>

---

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

---

## Description

Tools for mining CWB corpora.

## Usage

polmineR()

## Details

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

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

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

**Author(s)**

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

**References**

<http://polmine.sowi.uni-due.de>

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

**Examples**

```
# examples in the manual rely in a sample corpus that can be install as follows:
install.corpus("polmineR.sampleCorpus")
```

---

as.markdown

*Generate markdown from a partition.*


---

**Description**

The method is the worker behind the html-method.

**Usage**

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

## S4 method for signature 'numeric'
as.markdown(.Object, corpus, meta = NULL, cpos = FALSE,
  cutoff = NULL, ...)

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

**Arguments**

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

---

as.sparseMatrix	<i>Type conversion - get sparseMatrix.</i>
-----------------	--

---

### Description

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

### Usage

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

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

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

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

### Arguments

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

---

as.speeches	<i>Split partition into speeches</i>
-------------	--------------------------------------

---

### Description

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

### Usage

```
## S4 method for signature 'partition'
as.speeches(.Object, sAttributeDates, sAttributeNames,
  gap = 500, mc = FALSE, verbose = TRUE, progress = TRUE)
```

**Arguments**

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

**Value**

a partitionBundle object

**Examples**

```
## Not run:
use("polmineR.sampleCorpus")
bt <- partition("PLPRBTXT", text_year = "2009")
speeches <- as.speeches(bt, sAttributeDates = "text_date", sAttributeNames = "text_name")

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

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

## End(Not run)
```

---

as.TermDocumentMatrix *Generate TermDocumentMatrix / DocumentTermMatrix.*

---

**Description**

Method for type conversion, to generate the classes "TermDocumentMatrix" or "DocumentTermMatrix" contained in the "tm" package. The classes inherit from the "simple\_triplet\_matrix"-class defined in the "slam"-package. A "DocumentTermMatrix" is required as input by the "topicmodels" package, for instance.

**Usage**

```

as.TermDocumentMatrix(x, ...)

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

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

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

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

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

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

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

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

```

**Arguments**

x	some object
...	to make the check happy
pAttribute	the p-attribute
sAttribute	the s-attribute
verbose	bla
col	the column to use of assembling the matrix

**Details**

The type conversion-method can be applied on object of the class "bundle", or classes inheriting from the "bundle" class. If counts or some other measure is present in the "stat" slots of the objects in the bundle, then the values in the column indicated by "col" will be turned into the values of the sparse matrix that is generated. A special case is the generation of the sparse matrix

based on a "partitionBundle" that does not yet include counts. In this case, a "pAttribute" needs to be provided, then counting will be performed, too.

### Value

a TermDocumentMatrix

### Author(s)

Andreas Blaette

me

### Examples

```
## Not run:
use("polmineR.sampleCorpus")

# do-it-yourself
p <- partition("PLPRBTTXT", text_date=".*", regex=TRUE)
pB <- partitionBundle(p, sAttribute="text_date")
pB <- enrich(pB, pAttribute="word")
tdm <- as.TermDocumentMatrix(pB, col = "count")

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

# diretissima
tdm <- as.TermDocumentMatrix("PLPRBTTXT", pAttribute = "word", sAttribute = "text_date")

## End(Not run)
```

---

as.VCorpus,partitionBundle-method

*Coerce partitionBundle to VCorpus.*

---

### Description

Coerce partitionBundle to VCorpus.

### Usage

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

### Arguments

x                      a partitionBundle object

## Examples

```
## Not run:
use("polmineR.sampleCorpus")
P <- partition("PLPRBTTXT", text_date = "2009-11-10")
PB <- partitionBundle(P, sAttribute = "text_speaker")
VC <- as.VCorpus(PB)

## End(Not run)
```

---

blapply

*apply a function over a list or bundle*


---

## Description

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

## Usage

```
blapply(x, ...)

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

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

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

## Arguments

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

## Details

Parallel backend supported so far are the parallel package (mclapply), and doMC, doParallel and doSNOW in combination with foreach. The parallel backend to be used is taken from the option 'polmineR.backend' (getOption("polmineR.backend")), the number of cores from the option 'polmineR.cores' (getOption("polmineR.cores")).

## Examples

```
## Not run:
use("polmineR.sampleCorpus")
bt <- partition("PLPRBTXT", list(text_id=".*"), regex=TRUE)
speeches <- as.speeches(bt, sAttributeDates="text_date", sAttributeNames="text_name")
foo <- blapply(speeches, function(x, ...) slot(x, "cpos"))

## End(Not run)
```

---

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

---

## Description

bla

## Usage

```
browse(object, ...)

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

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

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

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

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

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

**Arguments**

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

---

bundle-class

*Bundle class*


---

**Description**

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

**Usage**

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

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

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

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

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

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

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

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

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

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

## S4 method for signature 'bundle'
```

```

as.data.table(x, col)

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

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

```

### Arguments

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

### Slots

```

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

```

### Author(s)

Andreas Blaette

### Examples

```

## Not run:
use("europarl.en")
Ps <- partitionBundle(
  "EUROPARL-EN", sAttribute = "text_year",
  values = sAttributes("EUROPARL-EN", "text_year")
)
Cs <- cooccurrences(Ps, query = "Europe", cq = FALSE, verbose = FALSE, progress = TRUE)
dt <- as.data.table(Cs, col = "11")
m <- as.matrix(Cs, col = "11")

## End(Not run)

```

---

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

---

### Description

Perform Chisquare-Test based on a table with counts

### Usage

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

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

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

### Arguments

.Object	object
...	further parameters

### Details

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

### Value

a table

### Author(s)

Andreas Blaette

---

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

---

### Description

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

**Usage**

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

## S4 method for signature 'character'
context(.Object, query,
  pAttribute = getOption("polmineR.pAttribute"), sAttribute = NULL, ...)

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

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

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

**Arguments**

<code>.Object</code>	a partition or a partitionBundle object
<code>query</code>	query, which may be a character vector or a cqpQuery object
<code>cqp</code>	defaults to <code>is.cqp</code> -function, or provide TRUE/FALSE
<code>left</code>	no of tokens and to the left of the node word
<code>right</code>	no of tokens to the right of the node word
<code>pAttribute</code>	p-attribute of the query
<code>sAttribute</code>	if provided, it will be checked that corpus positions do not extend beyond the region defined by the s-attribute
<code>stoplist</code>	exclude a query hit from analysis if stopword(s) is/are in context. See positivelist for further explanation.
<code>positivelist</code>	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)
<code>regex</code>	logical, defaults to FALSE - whether stoplist and/or positivelist are regular expressions
<code>count</code>	logical
<code>mc</code>	whether to use multicore; if NULL (default), the function will get the value from the options
<code>verbose</code>	report progress, defaults to TRUE
<code>progress</code>	logical, whether to show progress bar

... further parameters  
complete enhance completely

### Details

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

### Value

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

### Author(s)

Andreas Blaette

### Examples

```
## Not run:
use("polmineR.sampleCorpus")
p <- partition("PLPRBTTXT", list(text_type="speech"))
y <- context(p, query = "Integration", pAttribute = "word")
y <- context(p, query = "Integration", pAttribute = "word", positivelist = "Bildung")
y <- context(
  p, query = "Integration", pAttribute = "word",
  positivelist = c("[aA]rbeit.*", "Ausbildung"), regex = TRUE
)

## End(Not run)
```

---

context-class

*Context class.*

---

### Description

Class to organize information of context analysis.

### Usage

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

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

## S4 method for signature 'context'
enrich(.Object, sAttribute = NULL, pAttribute = NULL,
  id2str = FALSE, verbose = TRUE)
```

```
## S4 method for signature 'context'
trim(object, sAttribute = NULL, verbose = TRUE,
      progress = TRUE)
```

### Arguments

x	a context object
size	integer indicating sample size
.Object	object
sAttribute	s-attribute(s) to add to data.table in cpos-slot
pAttribute	p-attribute(s) to add to data.table in cpos-slot
id2str	logical, whether to convert integer ids to expressive strings
verbose	logical, whether to be talkative
object	a context object
progress	logical, whether to show progress bar

### Details

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

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

### Slots

query	Object of class "character", the query/node examined
count	Object of class "numeric" number of hits
partition	Object of class "partition", the partition the context object is based on
partitionSize	Object of class "numeric" the size of the partition
left	Object of class "numeric" number of tokens to the left
right	Object of class "numeric" number of tokens to the right
size	Object of class "numeric" number of tokens in the right and left context
sAttribute	Object of class "character" s-attribute
pAttribute	Object of class "character" p-attribute of the query
corpus	Object of class "character" the CWB corpus used
stat	Object of class "data.table" statistics of the analysis
encoding	Object of class "character" encoding of the corpus
cpos	Object of class "list" corpus positions of the hits
method	Object of class "character" statistical test used
call	Object of class "character" call that generated the object

---

contextBundle-class     *S4 contextBundle class*

---

### Description

class to organize information of multiple context analyses

### Slots

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

### Methods

**show**    output of core information

**summary**    core statistical information

[    specific cooccurrences

[[    specific cooccurrences

---

cooccurrences     *Get cooccurrence statistics.*

---

### Description

Get cooccurrence statistics.

### Usage

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

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

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

```
## S4 method for signature 'context'
```

```
cooccurrences(.Object, method = "ll", verbose = FALSE)
```

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

### Arguments

<code>.Object</code>	a partition object, or a character vector with a CWB corpus
<code>...</code>	further parameters that will be passed into bigmatrix (applies only if <code>big=TRUE</code> )
<code>query</code>	query, may be a character vector to match a token, or a CQP query
<code>cqp</code>	defaults to <code>is.cqp</code> -function, or provide TRUE/FALSE, relevant only if query is not NULL
<code>pAttribute</code>	the pAttribute of the tokens/the query
<code>sAttribute</code>	if provided, it will be checked that cpos do not extend beyond the region defined by the s-attribute
<code>left</code>	no of tokens and to the left of the node word
<code>right</code>	no of tokens to the right of the node word
<code>stoplist</code>	exclude a query hit from analysis if stopword(s) is/are in context (relevant only if query is not NULL)
<code>positivelist</code>	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)
<code>regex</code>	logical, whether stoplist/positivelist are dealt with as regular expressions
<code>keep</code>	list with tokens to keep
<code>cpos</code>	integer vector with corpus positions, defaults to NULL - then the corpus positions for the whole corpus will be used
<code>method</code>	statistical test to use (defaults to "ll")
<code>mc</code>	whether to use multicore
<code>verbose</code>	logical, whether to be verbose
<code>progress</code>	logical, whether to be verbose

### Value

a cooccurrences-class object

### Author(s)

Andreas Blaette

## Examples

```
## Not run:
use("polmineR.sampleCorpus")
merkel <- partition("PLPRBTTXT", text_type = "speech", text_name = ".*Merkel", regex = TRUE)
merkel <- enrich(merkel, pAttribute = "word")
cooc <- cooccurrences(merkel, query = "Deutschland")

## End(Not run)
```

---

cooccurrences-class     *Cooccurrences class.*

---

## Description

S4 class to organize information of context analysis

## Usage

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

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

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

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

## Arguments

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

## Slots

call Object of class "character" the call that generated the object  
 partition Object of class "character" the partition the analysis is based on  
 partitionSize Object of class "numeric" the size of the partition  
 left Object of class "numeric" number of tokens to the right  
 right Object of class "numeric" number of tokens to the left  
 pAttribute Object of class "character" p-attribute of the query  
 corpus Object of class "character" the CWB corpus used

stat Object of class "data.frame" statistics of the analysis  
encoding Object of class "character" encoding of the corpus  
pos Object of class "character" part-of-speech tags filtered  
method Object of class "character" statistical test(s) used  
cutoff Object of class "list" cutoff levels that have been applied  
svg Object of class "character" - valid XML with svg representation

---

cooccurrencesReshaped *Methods for manipulating cooccurrencesReshaped-class-objects*

---

**Description**

Methods for manipulating cooccurrencesReshaped-class-objects

**Arguments**

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

---

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

---

**Description**

Corpus class.

**Usage**

Corpus

**Format**

An object of class R6ClassGenerator of length 24.

**Fields**

corpus character vector, CWB corpus  
encoding encoding of the corpus  
count data.table with counts

**Methods**

count(pAttribute = getOption("polmineR.pAttribute"), id2str = TRUE) Perform counts.

---

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

---

### Description

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

### Usage

```
corpus(object)

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

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

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

### Arguments

object            the object

### Details

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

---

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

---

### Description

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

### Usage

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

## S4 method for signature 'partition'
count(.Object, query = NULL, cqp = is.cqp,
      id2str = TRUE, pAttribute = getOption("polmineR.pAttribute"),
      mc = getOption("polmineR.cores"), verbose = TRUE, progress = FALSE)
```

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

## S4 method for signature 'character'
count(.Object, query = NULL, cqp = is.cqp,
      pAttribute = getOption("polmineR.pAttribute"), sort = FALSE,
      id2str = TRUE, verbose = TRUE)

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

### Arguments

<code>.Object</code>	a "partition" or "partitionBundle" object, or a character vector (length 1) providing the name of a corpus
<code>...</code>	further parameters
<code>query</code>	a character vector (one or multiple terms to be looked up), 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>id2str</code>	logical, whether to add rownames (only if query is NULL)
<code>pAttribute</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
<code>freq</code>	logical, if FALSE, counts will be reported, if TRUE, frequencies
<code>total</code>	defaults to FALSE, if TRUE, the added value of counts (column: TOTAL) will be amended to the data.table that is returned
<code>sort</code>	logical, whether to sort stat
<code>corpus</code>	name of CWB corpus

### Details

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

If `.Object` is a character vector (length 1) and `query` is NULL, the count is performed for the whole partition. The method will check whether the `polmineR.Rcpp` package, or the `cwb-lexdecode` utilities are available, and use them respectively for performance reasons.

### Value

a "data.table"

**See Also**

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

count

**Examples**

```
## Not run:
use("polmineR.sampleCorpus")
debates <- partition("PLPRBTTXT", list(text_id=".*"), regex=TRUE)
count(debates, query = "Arbeit") # get frequencies for one token
count(debates, c("Arbeit", "Freizeit", "Zukunft")) # get frequencies for multiple tokens

count("PLPRBTTXT", query = c("Migration", "Integration"), pAttribute = "word")

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

## End(Not run)
```

---

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

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

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

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

### Arguments

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

### Details

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

If the cpos-method is called on a "matrix" object, the cpos matrix is unfolded.

### Value

Unless .Object is a "matrix", you get a matrix with two columns, the first column giving the start cpos of the hits obtained, the second column giving the end 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 three different interfaces to the Corpus Workbench (CWB): The package 'rcqp', via cqpserver, and by calling Perl scripts. 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.Rcpp
```

```
CQI.cqpserver
```

```
CQI.perl
```

```
CQI.rcqp
```

**Format**

An object of class R6ClassGenerator of length 24.

---

```
cqp
```

```
Tools for CQP queries.
```

---

**Description**

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

**Usage**

```
is.cqp(query)
```

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

**Arguments**

```
query
```

 character vector with at least one query

```
normalise.case
```

 logical

```
collapse
```

 logical, whether to collapse the queries into one
**Details**

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

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

**Value**

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

## References

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

## Examples

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

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

---

cqpserver	<i>start CQP server</i>
-----------	-------------------------

---

## Description

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

## Usage

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

## Arguments

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

---

decode	<i>Decode corpus.</i>
--------	-----------------------

---

## Description

An encoded CWB corpus is turned into a tibble.

**Usage**

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

## S4 method for signature 'Corpus'
decode(.Object, verbose = TRUE)

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

**Arguments**

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

**Examples**

```
## Not run:
PLPRBTTXT <- Corpus$new("PLPRBTTXT")

## End(Not run)
```

---

dispersion

*Dispersion of a query or multiple queries*


---

**Description**

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

**Usage**

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

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

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

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

**Arguments**

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

**Value**

depends on the input, as this is a wrapper function

**Author(s)**

Andreas Blaette

**See Also**

crosstab-class  
count

**Examples**

```
## Not run:
use("polmineR.sampleCorpus")
test <- partition("PLPRBTTXT", text_year = "2009", pAttribute = NULL)
integration <- dispersion(
  test, query = "Integration",
  pAttribute = "word", sAttribute = "text_date"
)
integration <- dispersion(
  test, "Integration",
  sAttribute = c("text_date", "text_party")
)
integration <- dispersion(
  test, '"Integration.*"',
  sAttribute = "text_year", cqp = TRUE
)

## End(Not run)
```

---

dispersion-class	<i>dispersion class</i>
------------------	-------------------------

---

## Description

class to organize results from dispersion analysis

Drop unwanted columns in a dispersion object, and merge columns by either explicitly stating the columns, or providing a regex. If merge\$old is length 1, it is assumed that a regex is provided

## Usage

```
## S4 method for signature 'dispersion'
t(x)

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

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

## S4 method for signature 'dispersion'
freq(.Object)

## S4 method for signature 'dispersion'
trim(object, drop = NULL, merge = list(old = c(), new
  = c()))
```

## Arguments

x	a dispersion object
object	a crosstab object to be adjusted
.Object	a dispersion object
drop	defaults to NULL, or a character vector giving columns to be dropped
merge	a list giving columns to be merged or exactly one string with a regex (see examples)

## Value

a modified crosstab object

## Slots

call call that generated this object  
 count Object of class "character" the call that generated the object  
 freq Object of class "character" the partition the analysis is based on

dim dimensions of the object  
 query the query / the queries that were used  
 sizes Object of class "numeric" the size of the partition

---

divide	<i>divide an object into equally sized parts</i>
--------	--

---

## Description

divide an object into equally sized parts

## Usage

```
divide(.Object, n)

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

## S4 method for signature 'vector'
divide(.Object, n)
```

## Arguments

.Object	object to divide
n	number of objects

---

dotplot	<i>dotplot</i>
---------	----------------

---

## Description

dotplot

## Usage

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

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

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

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

**Arguments**

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

---

encode	<i>Encode CWB Corpus.</i>
--------	---------------------------

---

**Description**

Encode CWB Corpus.

**Usage**

```
encode(.Object, ...)

## S4 method for signature 'data.frame'
encode(.Object, name, pAttributes = "word",
       sAttributes = NULL, registry = Sys.getenv("CORPUS_REGISTRY"),
       indexedCorpusDir = NULL, verbose = TRUE)

## S4 method for signature 'data.table'
encode(.Object, corpus, sAttribute)
```

**Arguments**

<code>.Object</code>	a <code>data.frame</code> to encode
<code>...</code>	further parameters
<code>name</code>	name of the (new) CWB corpus
<code>pAttributes</code>	columns of <code>.Object</code> with tokens (such as <code>word/pos/lemma</code> )
<code>sAttributes</code>	columns of <code>.Object</code> that will be encoded as structural attributes
<code>registry</code>	path to the corpus registry
<code>indexedCorpusDir</code>	directory where to create directory for indexed corpus files
<code>verbose</code>	logical, whether to be verbose
<code>corpus</code>	the name of the CWB corpus
<code>sAttribute</code>	a single s-attribute

---

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>partitionBundle</code> or <code>comp</code> object
<code>...</code>	further parameters

## See Also

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

---

features,partition-method	<i>Get features by comparison.</i>
---------------------------	------------------------------------

---

## Description

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

```
#' @rdname features-method
setMethod("features", "cooccurrences", function(x, y, included = FALSE, method = "ll", mc = TRUE, verbose = TRUE) {
  newObject <- new( 'compCooccurrences',
    encoding = x@encoding, included = included, corpus = x@corpus, sizeCoi = x@partitionSize,
    sizeRef = ifelse(included == FALSE, y@partitionSize, y@partitionSize - x@partitionSize), stat =
    data.table() )
  if (identical(x@pAttribute, y@pAttribute) == FALSE) warning("BEWARE: cooccurrences objects are not based on the same pAttribute!")
  else newObject@pAttribute <- x@pAttribute
})
```

```

if (verbose == TRUE) message("... preparing tabs for matching") keys <- unlist(lapply(c("a",
"b"), function(ab) paste(ab, x@pAttribute, sep="_")) setkeyv(x@stat, keys) setkeyv(y@stat, keys)
MATCH <- y@stat[x@stat]

# remove columns not needed colsToDrop <- c("l", "i.l", "exp_window", "i.exp_window", "rank_l",
"i.rank_l", "size_window", "i.size_window", "count_a", "i.count_a", "count_b", "i.count_b", "exp_partition",
"i.exp_partition" ) for (drop in colsToDrop) MATCH[, eval(drop) := NULL, with=TRUE] set-
names(MATCH, old=c("count_ab", "i.count_ab"), new=c("count_ref", "count_coi"))

if (included == TRUE) MATCH[, "count_ref" := MATCH[["count_ref"]] - MATCH[["count_coi"]]
]

newObject@stat <- MATCH for (how in method) if (verbose == TRUE) message("... statistical
test: ", how) newObject <- do.call(how, args = list(Object = newObject))

newObject )

```

## Usage

```

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

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

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

```

## Arguments

x	a partition or partitionBundle object
y	a partition object, it is assumed that the coi is a subcorpus of ref
included	TRUE if coi is part of ref, defaults to FALSE
method	the statistical test to apply (chisquare or log likelihood)
verbose	logical, defaults to TRUE
mc	logical, whether to use multicore
progress	logical
...	further parameters

## Value

The function returns a data frame with the following structure: - absolute frequencies in the first row - ...

## Author(s)

Andreas Blaette

## References

Manning / Schuetze ...

## Examples

```
## Not run:
use("polmineR.sampleCorpus")
kauder <- partition("PLPRBTTXT", text_name="Volker Kauder", pAttribute="word")
all <- partition("PLPRBTTXT", text_date=".*", regex=TRUE, pAttribute="word")
terms_kauder <- features(x = kauder, y = all, included = TRUE)
top100 <- subset(terms_kauder, rank_chisquare <= 100)

## End(Not run)
## Not run:
use("polmineR.sampleCorpus")
byName <- partitionBundle("PLPRBTTXT", sAttribute="text_name")
byName <- enrich(byName, pAttribute="word")
all <- partition("PLPRBTTXT", text_date=".*", regex=TRUE, pAttribute="word")
result <- features(byName, all, included=TRUE, progress=TRUE)
dtm <- as.DocumentTermMatrix(result, col="chisquare")

## End(Not run)
```

---

features-class

*Feature selection by comparison (S4 class).*

---

## Description

object resulting from features-method

## Usage

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

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

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

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

## Arguments

object	an object
.Object	an object

**Slots**

corpus Object of class "character"  
 pAttribute Object of class "character"  
 encoding Object of class "character"  
 corpus Object of class "character"  
 stat Object of class "data.frame"  
 sizeCoi Object of class "numeric"  
 sizeRef Object of class "numeric"  
 included Object of class "logical" whether corpus of interest is included in reference corpus  
 method Object of class "character" statisticalTest used  
 call Object of class "character" the call that generated the object  
 objects an object of class list

**Author(s)**

Andreas Blaette

---

flatten	<i>flatten a nested list</i>
---------	------------------------------

---

**Description**

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

**Usage**

```
flatten(object)
```

**Arguments**

object                    a list (with partitionBundle objects)

**Value**

a partitionBundle object

---

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

---

**Description**

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

**Usage**

```
getEncoding(.Object)

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

**Arguments**

.Object	the corpus name
---------	-----------------

---

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

---

**Description**

Get objects of a certain class.

**Usage**

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

**Arguments**

class	character, class to be looked for
envir	character string, namespace to be searched

**Value**

a list with the partitions found in the namespace

---

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

---

**Description**

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

**Usage**

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

**Arguments**

x	object to get slot from
name	name of the slot

---

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

---

**Description**

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

**Usage**

```
getTemplate(.Object, ...)

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

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

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

setTemplate(.Object, ...)

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

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

**Arguments**

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

---

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

---

**Description**

get terms available in a corpus or partition

**Usage**

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

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

**Arguments**

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

---

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

---

**Description**

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

**Usage**

```
getTokenStream(.Object, ...)

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

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

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

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

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

**Arguments**

.Object	an object of class matrix or partition
...	further arguments
corpus	the CWB corpus
pAttribute	the pAttribute to decode
encoding	encoding to use
collapse	character string length 1
beautify	logical, whether to adjust whitespace before and after interpunctuation
cpos	logical, whether to return cpos as names of the tokens
cutoff	maximum number of tokens to be reconstructed
left	left corpus position
right	right corpus position

---

highlight

*Highlight tokens.*


---

**Description**

Highlight tokens by term or cpos in a html document generated by read/html.

**Usage**

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

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

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

**Arguments**

.Object	either a "partition" or "character" object
...	further parameters
highlight	a "list" of character vectors, the names need to provide the colors
tooltips	a "list" of character vectors, all names need to be included in lists of the highlight-list
html	character vector with a website

---

hits-class

*Get Hits.*


---

**Description**

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

**Usage**

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

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

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

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

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

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

### Arguments

<code>.Object</code>	a character, partition or partitionBundle object
<code>...</code>	further parameters
<code>query</code>	a (optionally named, see details) character vector with one or more queries
<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
<code>sAttribute</code>	s-attributes
<code>pAttribute</code>	p-attribute (will be passed into cpos)
<code>size</code>	logical - return size of subcorpus
<code>freq</code>	logical - return relative frequencies
<code>mc</code>	logical, whether to use multicore
<code>verbose</code>	logical
<code>progress</code>	logical, whether to show progress bar
<code>x</code>	a hits object

### Details

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

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

### Slots

```
dt a "data.table"
corpus a "character" vector
query Object of class "character"
```

---

html	<i>restore fulltext as html</i>
------	---------------------------------

---

## Description

restore fulltext as html

## Usage

```
html(object, ...)

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

## S4 method for signature 'partition'
html(object, meta = getOption("polmineR.meta"),
      highlight = list(), cqp = FALSE, tooltips = NULL, cpos = FALSE,
      verbose = FALSE, cutoff = NULL, ...)

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

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

## Arguments

object	the object
...	further parameters
meta	metadata for output
highlight	list with regex to be highlighted
cqp	logical
tooltips	tooltips for highlighted text
cpos	logical
verbose	logical
cutoff	maximum number of tokens to decode from token stream
filename	the filename
type	the partition type
i	to be checked

---

install.corpus	<i>Install packaged corpus from repository.</i>
----------------	---

---

## Description

Convenience function for making the installation of indexed CWB corpora wrapped into R data packages as easy as possible. Packaged corpora can then be used by calling [use](#).

## Usage

```
install.corpus(pkgs, repo = "http://polmine.sowi.uni-due.de/packages",
  lib = .libPaths()[1], ...)
```

```
packaged.corpora()
```

## Arguments

pkgs	names of data packages with corpora
repo	URL of the repository
lib	directory for R packages, defaults to <code>.libPaths()[1]</code> ; the path may not include a whitespace sign
...	further parameters that will be passed into <code>install.packages</code>

## Details

The function combines two steps necessary to install a CWB corpus wrapped into a R data package. First, it calls `install.packages`, then it resets the path pointing to the directory with the indexed corpus files in the registry file.

The corpus will be installed to the standard library directory for installing R packages (`.libPaths()[1]`). Another location can be used by stating the param 'lib' explicitly (see documentation for [install.packages](#)).

The function can also be used to install a corpus from a password protected repository. Further parameters are handed over to `install.packages`, so you might add `method = "wget"` `extra = "--user donald --password d`

See examples how to check whether the directory has been set correctly.

An installed data package with a CWB corpus is assumed to include a directory `/extdata/cwb/registry` for registry files and a directory `/extdata/cwb/indexed_corpora` for the indexed corpus files.

## See Also

For managing registry files, see class [RegistryFile](#), see [use](#) for switching to a packaged corpus.

## Examples

```
install.corpus("plprbt.pvs2017")
# is equivalent to:
install.corpus("plprbt.pvs2017", repo = "http://polmine.sowi.uni-due.de/packages")
RegistryFile(package = "plprbt.pvs2017")$adjustHome()
# check the directory that has been set
RegistryFile$new(package = "plprbt.pvs2017")$getHome()
```

---

kwic	<i>KWIC output / concordances</i>
------	-----------------------------------

---

## Description

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

## Usage

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

## S4 method for signature 'context'

```
kwic(.Object, meta = getOption("polmineR.meta"),
     cpos = TRUE, verbose = FALSE)
```

## S4 method for signature 'partition'

```
kwic(.Object, query, cqp = is.cqp,
     left = getOption("polmineR.left"), right = getOption("polmineR.right"),
     meta = getOption("polmineR.meta"), pAttribute = "word",
     sAttribute = NULL, cpos = TRUE, stoplist = NULL, positivelist = NULL,
     regex = FALSE, verbose = TRUE)
```

## S4 method for signature 'character'

```
kwic(.Object, query, cqp = is.cqp,
     left = getOption("polmineR.left"), right = getOption("polmineR.right"),
     meta = getOption("polmineR.meta"), pAttribute = "word",
     sAttribute = NULL, cpos = TRUE, verbose = TRUE)
```

## Arguments

.Object	a partition or context object
...	further parameters to be passed
meta	metainformation to display
cpos	logical, if TRUE, the corpus positions ("cpos") if the hits will be handed over to the kwic-object that is returned

verbose	logical, whether to be talkative
query	a query, CQP-syntax can be used
cqp	either logical (TRUE if query is a CQP query), or a function to check whether query is a CQP query or not (defaults to is.query auxiliary function)
left	to the left
right	to the right
pAttribute	p-attribute, defaults to 'word'
sAttribute	if provided, the s-attribute will be used to check the boundaries of the text
stoplist	terms or ids to prevent a concordance from occurring in results
positivelist	terms or ids required for a concordance to occur in results
regex	logical, whether stoplist/positivelist is processed as regular expression

## Details

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

## See Also

To read the whole text, see the [read](#)-method.

## Examples

```
## Not run:
if (require(polmineR.sampleCorpus) && require(rcqp)){
  use("polmineR.sampleCorpus")
  bt <- partition("PLPRBTXT", def=list(text_date=".*"), regex=TRUE)
  kwic(bt, "Integration")
  kwic(
    bt, "Integration",
    left=20, right=20,
    meta=c("text_date", "text_name", "text_party")
  )
  kwic(
    bt, '"Integration" [] "(Menschen|Migrant.*|Personen)"',
    left=20, right=20,
    meta=c("text_date", "text_name", "text_party")
  )
}

## End(Not run)
```

---

kwic-class	<i>kwic (S4 class)</i>
------------	------------------------

---

## Description

S4 class for organizing information for concordance output

## Usage

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

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

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

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

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

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

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

## Arguments

object	an object of class kwic
x	a kwic-class object
size	integer, the subset size for sampling
.Object	a kwic object
meta	sAttributes (character vector) with metainformation
table	logical, whether to turn cpos data.table into data.frame for output
highlight	a list
regex	logical, whether to use regular expressions
perl	logical, whether to use perl-style regular expressions for highlighting when regex is TRUE
tooltips	a list
verbose	logical

## Details

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

## Slots

`metadata` Object of class "character" keeping the sAttributes of the metadata that are to be displayed

`left` words to the left

`right` words to the right

`corpus` the CWB corpus

`cpos` the corpus positions

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

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

`labels` Object of class "character"

`categories` Object of class "character"

## Methods

`[` indexing for seeing only some concordances

`show` get kwic output

## Examples

```
## Not run:
use("polmineR.sampleCorpus")
K <- kwic("PLPRBTXT", "Integration")
length(K)
K[1]
K[1:5]

## End(Not run)
```

---

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

---

**Description**

text statistics

**Usage**

```
ll(.Object, ...)  
  
## S4 method for signature 'context'  
ll(.Object)  
  
## S4 method for signature 'cooccurrences'  
ll(.Object)  
  
## S4 method for signature 'features'  
ll(.Object)  
  
pmi(.Object)  
  
## S4 method for signature 'context'  
pmi(.Object)
```

**Arguments**

.Object	an object
...	further parameters

---

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

---

**Description**

Mail a result (to yourself).

**Usage**

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

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

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

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

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

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

### Arguments

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

---

matches

*Matches for queries.*

---

### Description

If 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.

### Usage

```
matches(.Object, ...)

## S4 method for signature 'partition'
matches(.Object, query, cqp = is.cqp,
        pAttribute = getOption("polmineR.pAttribute"))
```

```
## S4 method for signature 'character'
matches(.Object, query, cqp = is.cqp,
        pAttribute = getOption("polmineR.pAttribute"))
```

### Arguments

.Object	a partition object
...	further parameters
query	a query, CQP syntax may be used
cqp	either logical to indicate whether the query uses CQP syntax, or a function to determine whether query is a CQP query (defaults to helper function <code>is.cqp</code> )
pAttribute	pAttribute

### Value

a data.table

### See Also

[hits-class](#)

### Examples

```
## Not run:
use("polmineR.sampleCorpus")
matches("PLPRBTTXT", '"Integration.*"')

P <- partition("PLPRBTTXT", text_date = "2009-11-11")
matches(P, '"Integration.*"')

## End(Not run)
```

---

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

---

### Description

calculate means

### Usage

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

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

**Arguments**

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

---

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

---

**Description**

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

**Usage**

```
name(x)

name(x) <- value

browse(x) <- value
```

**Arguments**

x	object
value	value

---

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

---

**Description**

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

**Usage**

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

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

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

**Arguments**

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

**Examples**

```
## Not run:
use("polmineR.sampleCorpus")
P <- partition("PLPRBTXT", text_date = "2009-10-27")
ngramObject <- ngrams(P, n = 2, pAttribute = "word", char = NULL)
# a more complex scenario: get most frequent ADJA/NN-combinations
ngramObject <- ngrams(P, n = 2, pAttribute = c("word", "pos"), char = NULL)
ngramObject2 <- subset(
  ngramObject,
  ngramObject[["1_pos"]] == "ADJA" & ngramObject[["2_pos"]] == "NN"
)
ngramObject2@stat[, "1_pos" := NULL, with = FALSE][, "2_pos" := NULL, with = FALSE]
ngramObject3 <- sort(ngramObject2, by = "count")
head(ngramObject3)

## End(Not run)
```

---

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, pAttribute, ...)
```

### Arguments

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

### Value

a list

---

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

---

### Description

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

**Usage**

```

partition(.Object, ...)

## S4 method for signature 'character'
partition(.Object, def = NULL, name = "",
  encoding = NULL, pAttribute = NULL, regex = FALSE, xml = "flat",
  id2str = TRUE, type = NULL, mc = FALSE, verbose = TRUE, ...)

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

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

## S4 method for signature 'partition'
partition(.Object, def = NULL, name = "",
  regex = FALSE, pAttribute = NULL, id2str = TRUE, type = NULL,
  verbose = TRUE, mc = FALSE, ...)

```

**Arguments**

<code>.Object</code>	character-vector - the CWB-corpus to be used
<code>...</code>	parameters passed into the partition-method
<code>def</code>	list consisting of a set of character vectors (see details and examples)
<code>name</code>	name of the new partition, defaults to "
<code>encoding</code>	encoding of the corpus (typically "LATIN1" or "(UTF-8)), if NULL, the encoding provided in the registry file of the corpus (charset="...") will be used b
<code>pAttribute</code>	the pAttribute(s) for which term frequencies shall be retrieved
<code>regex</code>	logical (defaults to FALSE)
<code>xml</code>	either 'flat' (default) or 'nested'
<code>id2str</code>	whether to turn token ids to strings (set FALSE to minimize object.size / memory consumption)
<code>type</code>	character vector (length 1) specifying the type of corpus / partition (e.g. "plpr")
<code>mc</code>	whether to use multicore (for counting terms)
<code>verbose</code>	logical, defaults to TRUE
<code>slots</code>	character vector

**Details**

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

The s-attributes defining the partition can be passed in as a list, e.g. `list(text_type="speech", text_year="2013")`, or - for convenience - directly.

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

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

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

### Value

An object of the S4 class `'partition'`

### Author(s)

Andreas Blaette

### See Also

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

### Examples

```
## Not run:
use("polmineR.sampleCorpus")
spd <- partition("PLPRBTTXT", text_party="SPD", text_type="speech")
kauder <- partition(
  "PLPRBTTXT", text_name="Volker Kauder", pAttribute="word"
)
merkel <- partition(
  "PLPRBTTXT", text_name=".*Merkel",
  pAttribute="word", regex=TRUE
)
sAttributes(merkel, "text_date")
sAttributes(merkel, "text_name")
merkel <- partition(
  "PLPRBTTXT", text_name="Angela Dorothea Merkel",
  text_date="2009-11-10", text_type="speech", pAttribute="word"
)
merkel <- subset(merkel, !word %in% punctuation)
merkel <- subset(merkel, !word %in% tm::stopwords("de"))

# a certain defined time segment
```

```

if (require("chron")){
  firstDay <- "2009-10-28"
  lastDay <- "2009-11-11"
  days <- strptime(
    chron::seq.dates(
      from = strptime(firstDay, format="%m/%d/%Y"),
      to = strptime(lastDay, format="%m/%d/%Y"),
      by="days"),
    format="%Y-%m-%d"
  )
  period <- partition("PLPRBTTXT", text_date=days)
}

## End(Not run)

```

---

partitionBundle	<i>Generate a bundle of partitions</i>
-----------------	--

---

## Description

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

## Usage

```

partitionBundle(.Object, ...)

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

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

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

```

## Arguments

.Object	character string, a partition, or a list
...	parameters to be passed into partition-method (see respective documentation)
sAttribute	the s-attribute to vary
values	values the s-attribute provided shall assume

prefix	a character vector that will be attached as a prefix to partition names
mc	logical, whether to use multicore parallelization
verbose	logical, whether to provide progress information
progress	logical, whether to show progress bar
xml	logical

**Value**

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

**Author(s)**

Andreas Blaette

**See Also**

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

**Examples**

```
## Not run:
use("polmineR.sampleCorpus")
bt2009 <- partition("PLPRBTTXT", text_year = "2009")
pBundle <- partitionBundle(bt2009, sAttribute = "text_date", progress = TRUE, pAttribute = "word")
dtm <- as.DocumentTermMatrix(pBundle, col = "count")
summary(pBundle)
btBundle <- partitionBundle("PLPRBTTXT", sAttribute = "text_date")

## End(Not run)
```

---

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

---

**Description**

Class and methods to manage bundles of partitions.

**Usage**

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

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

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

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

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

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

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

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

### Arguments

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

### Slots

objects Object of class "list" the partitions making up the bundle

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

sAttributesFixed Object of class "list" fixed sAttributes

encoding Object of class "character" encoding of the corpus

explanation Object of class "character" an explanation of the partition

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

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

### Author(s)

Andreas Blaette

---

partition_class	<i>Partition class and methods.</i>
-----------------	-------------------------------------

---

## Description

S4 partition class and methods for instances of class partition.

## Usage

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

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

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

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

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

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

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

## S4 method for signature 'partition'
enrich(.Object, size = FALSE, pAttribute = NULL,
  id2str = TRUE, verbose = TRUE, mc = FALSE, ...)

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

## S4 method for signature 'partition'
highlight(.Object, html, highlight = list(),
  cqp = is.cqp, tooltips = NULL)

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

## S4 method for signature 'partition'
```

```

view(.Object)

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

```

### Arguments

x	a partition object
value	value
...	further parameters
.Object	a partition object
meta	...
template	template to use
cpos	...
cutoff	maximum number of tokens to decode
verbose	logical
size	logical
pAttribute	a p-attribute (for enriching)
id2str	logical
mc	logical or, if numeric, providing the number of cores
html	...
highlight	...
cqp	...
tooltips	...

### Slots

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

Author(s)

Andreas Blaette

See Also

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

---

pAttribute	<i>get pAttribute</i>
------------	-----------------------

---

Description

get pAttribute

Usage

```
pAttribute(object)

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

Arguments

object	a textstat object
pAttribute	the pAttribute to get

---

pAttributes	<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 pAttributes-method.

Usage

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

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

**Arguments**

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

**References**

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

**Examples**

```
## Not run:
  use("polmineR.sampleCorpus")
  pAttributes("PLPRBTTXT")

## End(Not run)
```

---

read	<i>Display and read full text</i>
------	-----------------------------------

---

**Description**

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

**Usage**

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

## S4 method for signature 'partition'

```
read(.Object, meta = NULL, highlight = list(),
     cq = FALSE, tooltips = NULL, verbose = TRUE, cpos = FALSE,
     cutoff = getOption("polmineR.cutoff"), ...)
```

## S4 method for signature 'partitionBundle'

```
read(.Object, highlight = list(), cq = FALSE,
     cpos = FALSE, ...)
```

## S4 method for signature 'data.table'

```
read(.Object, col, partitionBundle, cq = FALSE,
     highlight = list(), cpos = FALSE, ...)
```

## S4 method for signature 'hits'

```
read(.Object, def, i = NULL, ...)
```

## S4 method for signature 'kwic'

```
read(.Object, i, type = NULL)
```

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

### Arguments

<code>.Object</code>	an object to be read ("partition" or "partitionBundle")
<code>...</code>	further parameters passed into read
<code>meta</code>	a character vector supplying s-attributes for the metainformation to be printed, if not stated explicitly, session settings will be used
<code>highlight</code>	a list
<code>cqp</code>	a list of character vectors with regular expressions to highlight relevant terms or expressions; the names of the list provide the colors
<code>tooltips</code>	a list
<code>verbose</code>	logical
<code>cpos</code>	logical
<code>cutoff</code>	maximum number of tokens to display
<code>col</code>	column
<code>partitionBundle</code>	a partitionBundle object
<code>def</code>	...
<code>i</code>	...
<code>type</code>	...

### See Also

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

### Examples

```
use("polmineR.sampleCorpus")
options("polmineR.meta" = "text_date")
merkel <- partition(
  "PLPRBTTXT",
  text_date="2009-11-10", text_name="Angela Dorothea Merkel",
  type="plpr"
)
read(merkel, meta=c("text_name", "text_date"))
read(
  merkel,
  highlight = list(yellow=c("Deutschland", "Bundesrepublik"), lightgreen="Regierung"),
  meta = c("text_name", "text_date")
)

all <- partition("PLPRBTTXT", list(text_id=".*"), regex=TRUE, type="plpr")
speeches <- as.speeches(
```

```

    all, sAttributeDates = "text_date", sAttributeNames = "text_name", gap = 500
  )
  read(speeches, meta = c("text_date", "text_name"))
  migVocab <- count(speeches, query=c("Migration", "Integration", "Zuwanderung"))
  read(migVocab, col="Integration", partitionBundle=speeches)

```

---

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

---

### Description

Regions of a CWB corpus.

### Slots

cpos a 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)

---

RegistryFile-class	<i>Read, parse and modify registry file.</i>
--------------------	--

---

### Description

Read, parse and modify registry file.

### Arguments

corpus	name of the CWB corpus
new	a new value to set
filename	a filename
package	name of a package
registry	directory of the registry (defaults to CORPUS_Registry environment variable)

### Fields

registryDir registry directory  
 encoding corpus encoding  
 txt registry as character vector  
 pAttributes p-attributes  
 properties corpus properties  
 id corpus id  
 home home directory  
 name corpus name  
 info path to info file

**Methods**

`getEncoding()` Get the encoding.  
`getHome()` Get the home directory of a corpus.  
`getId()` Get the id of a corpus.  
`getInfo()` Get path to the info file.  
`getName()` Get the name of a corpus.  
`getPAttributes()` Get the pAttributes.  
`getProperties()` Get corpus properties.  
`getSAttributes()` Get the sAttributes.  
`initialize(corpus = NULL, registry = Sys.getenv("CORPUS_REGISTRY"), package = NULL, filename = NULL)`  
Initialize a new RegistryFile object.  
`parse()` Parse the registry file.  
`read()` Read file from disc, as character vector in field 'txt'.

---

resetRegistry	<i>Reload using new CORPUS_REGISTRY.</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**

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

**Arguments**

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

**Details**

Resetting the CORPUS\_REGISTRY environment variable is achieved by unloading and reloading the C library 'rcqp' that is the backend for the rcqp package. It may not be the most artful way to do things, but it works.

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

**Value**

the registry directory used before resetting CORPUS\_REGISTRY

## Examples

```
sampleCorpusPkgDir <- system.file(package="polmineR.sampleCorpus")
sampleCorpusRegistryDir <- file.path(sampleCorpusPkgDir, "extdata", "cwb", "registry")
oldRegistryDir <- resetRegistry(sampleCorpusRegistryDir)
sampleCorpusPartition <- partition("PLPRBTTXT", text_speaker = "Volker Kauder")
resetRegistry(oldRegistryDir)
```

---

sAttributes,character-method

*Get s-attributes.*


---

## Description

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

## Usage

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

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

## Arguments

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

## Details

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

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

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

## Value

a character vector

**Examples**

```
## Not run:
use("polmineR.sampleCorpus")

sAttributes("PLPRBTTXT")
sAttributes("PLPRBTTXT", "text_date") # dates of plenary meetings

P <- partition("PLPRBTTXT", text_date = "2009-11-10")
sAttributes(P)
sAttributes(P, "text_name") # get names of speakers

## End(Not run)
```

---

scatterplot

---

*word scatterplot*


---

**Description**

plot a word scatterplot

**Usage**

```
## S4 method for signature 'data.frame'
scatterplot(object, xmax = c(), ymax = c(),
  fontSize = 0.7, rotation = 45, type = 0)
```

**Arguments**

object	a data frame with the ranks of cooccurrences
xmax	maximum on x axis for plot
ymax	maximum on y axis for plot
fontSize	the expansion factor for the words
rotation	rotation of the text
type	defaults to 0 for output of terms, can be set to symbol integer

**Value**

the plot

**Author(s)**

Andreas Blaette

---

size	<i>Get number of tokens.</i>
------	------------------------------

---

## Description

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

## Usage

```
size(x, ...)  
  
## S4 method for signature 'character'  
size(x, sAttribute = NULL, verbose = TRUE)  
  
## S4 method for signature 'partition'  
size(x, sAttribute = NULL)  
  
## S4 method for signature 'DocumentTermMatrix'  
size(x)
```

## Arguments

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

## Details

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

## Value

an integer vector if sAttribute is NULL, a `data.table` otherwise

## See Also

See [dispersion](#)-method for counts of hits. The [hits](#) method calls the size-method to get sizes of subcorpora.

**Examples**

```
## Not run:
use("polmineR.sampleCorpus")
size("PLPRBTTXT")
size("PLPRBTTXT", sAttribute = "text_date")
size("PLPRBTTXT", sAttribute = c("text_date", "text_party"))

P <- partition("PLPRBTTXT", text_date = "2009-11-11")
size(P, sAttribute = "text_name")
size(P, sAttribute = "text_party")
size(P, sAttribute = c("text_name", "text_party"))

## End(Not run)
```

---

split,partition-method

*split partition into partitionBundle*


---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

a partitionBundle

---

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

---

**Description**

S4 class to capture core information on a temporary CWB corpus

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

**Usage**

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

**Arguments**

.Object	a partition object
...	further parameters

**Slots**

cpos	matrix with start/end corpus positions
dir	directory where the tempcorpus is stored
registry	directory of the registry dir (subdirectory of dir)
indexed	directory of the dir with the indexed files

---

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

---

**Description**

Methods for TermDocumentMatrix / DocumentTermMatrix

**Usage**

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

**Arguments**

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

---

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

---

### Description

get terms available in a corpus

### Usage

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

### Arguments

x	a partition object
pAttribute	the pAttribute to be queried
regex	a regex to filter results

---

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

---

### Description

Superclass for features, context, and partition class.

### Usage

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

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

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

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

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

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

```

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

as.bundle(object, ...)

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

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

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

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

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

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

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

```

### Arguments

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

### Details

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

**Slots**

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

corpus Object of class "character"

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

name name of the object

encoding Object of class "character" encoding of the corpus

**Examples**

```
## Not run:
use("polmineR.sampleCorpus")
P <- partition("PLPRBTTXT", text_year = "2009", pAttribute = "word")
y <- cooccurrences(P, query = "Arbeit")
y[1:25]
y[,c("word", "11")]
y[1:25, "word"]
y[1:25][["word"]]
y[which(y[["word"]] %in% c("Arbeit", "Sozial"))]
y[ y[["word"]] %in% c("Arbeit", "Sozial") ]

## End(Not run)
```

---

TokenStream-class	<i>Class for token stream operations.</i>
-------------------	---

---

**Description**

Class for token stream operations.

**Slots**

tokenStream a data.table that will include a column "token"

corpus the CWB corpus (character vector length 1)

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

---

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 'partitionBundle' objects.

### Usage

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

### Arguments

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

### Format

An object of class character of length 13.

### Author(s)

Andreas Blaette

---

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

---

**Description**

S4 method for context object to perform t-test

**Usage**

```
tTest(.Object)

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

**Arguments**

.Object            a context or features object

---

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

---

**Description**

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

**Usage**

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

**Arguments**

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

**Details**

If pkg is the name of a data package with a CWB indexed corpus, the function will reset the CORPUS\_REGISTRY environment variable and restart rcqp to point to the registry directory in the package.

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

**Value**

the function returns invisibly the registry that was previously set

**See Also**

the worker to reset the registry is [resetRegistry](#)

---

view	<i>browse an object using View()</i>
------	--------------------------------------

---

**Description**

browse an object using View()

**Usage**

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

**Arguments**

.Object	an object
...	further parameters

---

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

---

**Description**

weigh a matrix

**Usage**

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

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

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

**Arguments**

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

# Index

## \*Topic **datasets**

- Corpus, [21](#)
- CQI.super, [25](#)
- trim, [77](#)

## \*Topic **package**

- polmineR-package, [4](#)

- +, bundle, bundle-method (bundle-class), [12](#)
- +, bundle, textstat-method (bundle-class), [12](#)
- +, partitionBundle, ANY-method (partitionBundle-class), [60](#)
- +, partitionBundle, partition-method (partitionBundle-class), [60](#)
- +, partitionBundle, partitionBundle-method (partitionBundle-class), [60](#)
- +, partitionBundle-method (partitionBundle-class), [60](#)
- +, textstat, textstat-method (textstat-class), [74](#)
- [, context, ANY, ANY, ANY-method (context-class), [16](#)
- [, context-method (context-class), [16](#)
- [, contextBundle, ANY, ANY, ANY-method (contextBundle-class), [18](#)
- [, contextBundle-method (contextBundle-class), [18](#)
- [, kwic, ANY, ANY, ANY-method (kwic-class), [48](#)
- [, kwic-method (kwic-class), [48](#)
- [, partition, ANY, ANY, ANY-method (partition\_class), [62](#)
- [, partition-method (partition\_class), [62](#)
- [, partitionBundle, ANY, ANY, ANY-method (partitionBundle-class), [60](#)
- [, partitionBundle-method (partitionBundle-class), [60](#)
- [, textstat, ANY, ANY, ANY-method (textstat-class), [74](#)

- [, bundle-method (bundle-class), [12](#)
- [, context-method (context-class), [16](#)
- [, contextBundle-method (contextBundle-class), [18](#)
- [, partitionBundle-method (partitionBundle-class), [60](#)
- [, textstat-method (textstat-class), [74](#)

- aggregate, partition-method (partition\_class), [62](#)
- as.bundle (textstat-class), [74](#)
- as.bundle, list-method (bundle-class), [12](#)
- as.bundle, textstat-method (bundle-class), [12](#)
- as.corpusEnc (encodings), [33](#)
- as.cqp (cqp), [26](#)
- as.data.frame, cooccurrencesBundle-method (cooccurrences-class), [20](#)
- as.data.frame, kwic-method (kwic-class), [48](#)
- as.data.frame, textstat-method (textstat-class), [74](#)
- as.data.table, bundle-method (bundle-class), [12](#)
- as.data.table, textstat-method (textstat-class), [74](#)
- as.DataTables, context-method (context-class), [16](#)
- as.DataTables, textstat-method (textstat-class), [74](#)
- as.DocumentTermMatrix (as.TermDocumentMatrix), [7](#)
- as.DocumentTermMatrix, bundle-method (as.TermDocumentMatrix), [7](#)
- as.DocumentTermMatrix, character-method (as.TermDocumentMatrix), [7](#)
- as.DocumentTermMatrix, context-method (as.TermDocumentMatrix), [7](#)
- as.DocumentTermMatrix, partitionBundle-method (as.TermDocumentMatrix), [7](#)

- as.markdown, [5](#)
- as.markdown,numeric-method
  - (as.markdown), [5](#)
- as.markdown,partition-method
  - (partition\_class), [62](#)
- as.markdown,plprPartition-method
  - (as.markdown), [5](#)
- as.matrix,bundle-method (bundle-class), [12](#)
- as.matrix,contextBundle-method
  - (context), [14](#)
- as.matrix,partitionBundle-method
  - (partitionBundle-class), [60](#)
- as.nativeEnc (encodings), [33](#)
- as.partitionBundle (partition\_class), [62](#)
- as.partitionBundle,list-method
  - (partitionBundle-class), [60](#)
- as.partitionBundle,partition-method
  - (partition\_class), [62](#)
- as.sparseMatrix, [6](#)
- as.sparseMatrix,bundle-method
  - (as.sparseMatrix), [6](#)
- as.sparseMatrix,simple\_triplet\_matrix-method
  - (as.sparseMatrix), [6](#)
- as.sparseMatrix,TermDocumentMatrix-method
  - (as.sparseMatrix), [6](#)
- as.speeches, [6](#)
- as.speeches,partition-method
  - (as.speeches), [6](#)
- as.TermContextBundle,contextBundle-method
  - (contextBundle-class), [18](#)
- as.TermDocumentMatrix, [7](#)
- as.TermDocumentMatrix,bundle-method
  - (as.TermDocumentMatrix), [7](#)
- as.TermDocumentMatrix,character-method
  - (as.TermDocumentMatrix), [7](#)
- as.TermDocumentMatrix,context-method
  - (as.TermDocumentMatrix), [7](#)
- as.TermDocumentMatrix,partitionBundle-method
  - (as.TermDocumentMatrix), [7](#)
- as.utf8 (encodings), [33](#)
- as.VCorpus,partitionBundle-method, [9](#)
- barplot,partitionBundle-method
  - (partitionBundle-class), [60](#)
- blapply, [10](#)
- blapply,bundle-method (blapply), [10](#)
- blapply,list-method (blapply), [10](#)
- blapply,vector-method (blapply), [10](#)
- browse, [11](#)
- browse,cooccurrences-method (browse), [11](#)
- browse,html-method (browse), [11](#)
- browse,kwic-method (browse), [11](#)
- browse,partition-method (browse), [11](#)
- browse,pressPartition-method (browse), [11](#)
- browse,textstat-method (browse), [11](#)
- browse<- (name), [54](#)
- bundle-class, [12](#)
- chisquare, [14](#)
- chisquare,context-method (chisquare), [14](#)
- chisquare,textstat-method (chisquare), [14](#)
- colnames,textstat-method
  - (textstat-class), [74](#)
- context, [14](#)
- context,character-method (context), [14](#)
- context,contextBundle-method (context), [14](#)
- context,cooccurrences-method (context), [14](#)
- context,partition-method (context), [14](#)
- context,partitionBundle-method
  - (context), [14](#)
- context-class, [16](#)
- context\_class (context-class), [16](#)
- contextBundle-class, [18](#)
- cooccurrences, [18](#)
- cooccurrences,character-method
  - (cooccurrences), [18](#)
- cooccurrences,context-method
  - (cooccurrences), [18](#)
- cooccurrences,Corpus-method (context), [14](#)
- cooccurrences,partition-method
  - (cooccurrences), [18](#)
- cooccurrences,partitionBundle-method
  - (cooccurrences), [18](#)
- cooccurrences-class, [20](#)
- cooccurrencesBundle
  - (cooccurrences-class), [20](#)
- cooccurrencesBundle-class
  - (cooccurrences-class), [20](#)
- cooccurrencesReshaped, [21](#)
- cooccurrencesReshaped-class
  - (cooccurrences-class), [20](#)
- Corpus, [21](#)

- corpus, [22](#)
- corpus, bundle-method (corpus), [22](#)
- corpus, missing-method (corpus), [22](#)
- corpus, partition-method (corpus), [22](#)
- count, [22](#)
- count, character-method (count), [22](#)
- count, context-method (context-class), [16](#)
- count, dispersion-method (dispersion-class), [30](#)
- count, partition-method (count), [22](#)
- count, partitionBundle-method (count), [22](#)
- count, vector-method (count), [22](#)
- count-method (count), [22](#)
- cpos, [24](#)
- cpos, character-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](#)
- cqpserver, [27](#)
- decode, [27](#)
- decode, character-method (decode), [27](#)
- decode, Corpus-method (decode), [27](#)
- dim, textstat-method (textstat-class), [74](#)
- dispersion, [28](#), [71](#)
- dispersion, character-method (dispersion), [28](#)
- dispersion, hits-method (dispersion), [28](#)
- dispersion, partition-method (dispersion), [28](#)
- dispersion-class, [30](#)
- divide, [31](#)
- divide, matrix-method (divide), [31](#)
- divide, vector-method (divide), [31](#)
- dotplot, [31](#)
- dotplot, features-method (dotplot), [31](#)
- dotplot, partition-method (dotplot), [31](#)
- dotplot, textstat-method (dotplot), [31](#)
- encode, [32](#)
- encode, data.frame-method (encode), [32](#)
- encode, data.table-method (encode), [32](#)
- encoding, [33](#)
- encoding, bundle-method (encoding), [33](#)
- encoding, textstat-method (encoding), [33](#)
- encoding<- (encoding), [33](#)
- encodings, [33](#)
- enrich, [34](#)
- enrich, context-method (context-class), [16](#)
- enrich, kwic-method (kwic-class), [48](#)
- enrich, partition-method (partition\_class), [62](#)
- enrich, partitionBundle-method (partitionBundle-class), [60](#)
- enrich-method (enrich), [34](#)
- export (partition\_class), [62](#)
- export, partition-method (partition\_class), [62](#)
- features (features, partition-method), [34](#)
- features, ngrams-method (features, partition-method), [34](#)
- features, partition-method, [34](#)
- features, partitionBundle-method (features, partition-method), [34](#)
- features-class, [36](#)
- featuresBundle-class (features-class), [36](#)
- featuresCooccurrences-class (features-class), [36](#)
- featuresNgrams-class (features-class), [36](#)
- flatten, [37](#)
- freq (dispersion-class), [30](#)
- freq, dispersion-method (dispersion-class), [30](#)
- freq, partition-method (partition\_class), [62](#)
- getEncoding, [38](#)
- getEncoding, character-method (getEncoding), [38](#)
- getObjects, [38](#)
- getSlot, [39](#)
- getTemplate, [39](#)
- getTemplate, character-method (getTemplate), [39](#)
- getTemplate, missing-method (getTemplate), [39](#)
- getTemplate, partition-method (getTemplate), [39](#)
- getTerms, [40](#)
- getTerms, character-method (getTerms), [40](#)
- getTokenStream, [40](#)

- getTokenStream, character-method  
(getTokenStream), 40
- getTokenStream, matrix-method  
(getTokenStream), 40
- getTokenStream, numeric-method  
(getTokenStream), 40
- getTokenStream, partition-method  
(getTokenStream), 40
- getTokenStream, Regions-method  
(getTokenStream), 40
  
- head, context-method (context-class), 16
- head, textstat-method (textstat-class),  
74
- highlight, 41
- highlight, character-method (highlight),  
41
- highlight, html-method (highlight), 41
- highlight, kwic-method (kwic-class), 48
- highlight, partition-method  
(partition\_class), 62
- hist, partition-method  
(partition\_class), 62
- hits, 71
- hits (hits-class), 42
- hits, character-method (hits-class), 42
- hits, context-method (hits-class), 42
- hits, partition-method (hits-class), 42
- hits, partitionBundle-method  
(hits-class), 42
- hits-class, 42
- html, 44
- html, character-method (html), 44
- html, kwic-method (html), 44
- html, partition-method (html), 44
- html, partitionBundle-method (html), 44
  
- install.corpus, 45
- install.packages, 45
- is.cqp (cqp), 26
  
- kwic, 46, 66
- kwic, character-method (kwic), 46
- kwic, context-method (kwic), 46
- kwic, partition-method (kwic), 46
- kwic-class, 48
  
- label, 49
- label, kwic-method (label), 49
  
- label<- (label), 49
- Labels (Labels-class), 50
- Labels-class, 50
- length, bundle-method (bundle-class), 12
- length, kwic-method (kwic-class), 48
- length, partition-method  
(partition\_class), 62
- ll, 51
- ll, context-method (ll), 51
- ll, cooccurrences-method (ll), 51
- ll, features-method (ll), 51
  
- mail, 51
- mail, cooccurrences-method (mail), 51
- mail, data.frame-method (mail), 51
- mail, dispersion-method (mail), 51
- mail, features-method (mail), 51
- mail, kwic-method (mail), 51
- mail, partition-method (mail), 51
- mail-method (mail), 51
- matches, 52
- matches, character-method (matches), 52
- matches, partition-method (matches), 52
- means, 53
- means, DocumentTermMatrix-method  
(means), 53
- merge, cooccurrencesReshaped-method  
(cooccurrencesReshaped), 21
- merge, partitionBundle-method  
(partitionBundle-class), 60
  
- name, 54
- name, partition-method  
(partition\_class), 62
- name<- (name), 54
- name<-, partition, character-method  
(partition\_class), 62
- names, bundle-method (bundle-class), 12
- names, partitionBundle-method  
(partitionBundle-class), 60
- names, textstat-method (textstat-class),  
74
- names<-, bundle, character-method  
(bundle-class), 12
- ngrams (ngrams-class), 54
- ngrams, partition-method (ngrams-class),  
54
- ngrams, partitionBundle-method  
(ngrams-class), 54

- ngrams-class, 54
- noise, 55
- noise, character-method (noise), 55
- noise, DocumentTermMatrix-method (noise), 55
- noise, TermDocumentMatrix-method (noise), 55
- noise, textstat-method (noise), 55
- nrow, textstat-method (textstat-class), 74
  
- packaged.corpora (install.corpus), 45
- partition, 56, 60
- partition, character-method (partition), 56
- partition, environment-method (partition), 56
- partition, list-method (partition), 56
- partition, partition-method (partition), 56
- partition-class (partition\_class), 62
- partition\_class, 34, 58, 62
- partitionBundle, 59
- partitionBundle, character-method (partitionBundle), 59
- partitionBundle, context-method (partitionBundle), 59
- partitionBundle, environment-method (partitionBundle-class), 60
- partitionBundle, partition-method (partitionBundle), 59
- partitionBundle-class, 60
- pAttribute, 64
- pAttribute, textstat-method (pAttribute), 64
- pAttributes, 64
- pAttributes, character-method (pAttributes), 64
- pAttributes, partition-method (partition\_class), 62
- plprPartition-class (partition\_class), 62
- pmi (ll), 51
- pmi, context-method (ll), 51
- polmineR (polmineR-package), 4
- polmineR-generics (name), 54
- polmineR-package, 4
- pressPartition-class (partition\_class), 62
  
- punctuation (trim), 77
  
- read, 47, 65
- read, data.table-method (read), 65
- read, hits-method (read), 65
- read, kwic-method (read), 65
- read, partition-method (read), 65
- read, partitionBundle-method (read), 65
- read, Regions-method (read), 65
- Regions-class, 67
- RegistryFile, 45
- RegistryFile (RegistryFile-class), 67
- RegistryFile-class, 67
- resetRegistry, 68, 79
- round, textstat-method (textstat-class), 74
- rownames, textstat-method (textstat-class), 74
  
- sample, bundle-method (bundle-class), 12
- sample, context-method (context-class), 16
- sample, hits-method (hits-class), 42
- sample, kwic-method (kwic-class), 48
- sAttributes
  - (sAttributes, character-method), 69
- sAttributes, character-method, 69
- sAttributes, partition-method (sAttributes, character-method), 69
- sAttributes, partitionBundle-method (partitionBundle-class), 60
- scatterplot, 70
- scatterplot, data.frame-method (scatterplot), 70
- scatterplot-method (scatterplot), 70
- setTemplate (getTemplate), 39
- setTemplate, character-method (getTemplate), 39
- setTemplate, missing-method (getTemplate), 39
- show, context-method (context-class), 16
- show, contextBundle-method (contextBundle-class), 18
- show, cooccurrences-method (cooccurrences-class), 20
- show, dispersion-method (dispersion-class), 30

- show, features-method (features-class),  
36
- show, html-method (html), 44
- show, kwic-method (kwic-class), 48
- show, partition-method  
(partition\_class), 62
- show, partitionBundle-method  
(partitionBundle-class), 60
- show, textstat-method (textstat-class),  
74
- size, 71
- size, character-method (size), 71
- size, DocumentTermMatrix-method (size),  
71
- size, partition-method (size), 71
- size, TermDocumentMatrix-method  
(TermDocumentMatrix), 73
- sort, textstat-method (textstat-class),  
74
- split (partition\_class), 62
- split, partition  
(split, partition-method), 72
- split, partition-method, 72
- startServer (cqpsrvr), 27
- subset, bundle-method (bundle-class), 12
- subset, textstat-method  
(textstat-class), 74
- summary, context-method (context-class),  
16
- summary, contextBundle-method  
(contextBundle-class), 18
- summary, cooccurrences-method  
(cooccurrences-class), 20
- summary, features-method  
(features-class), 36
- summary, featuresBundle-method  
(features-class), 36
- summary, partitionBundle-method  
(partitionBundle-class), 60
- t, dispersion-method (dispersion-class),  
30
- tail, textstat-method (textstat-class),  
74
- tempcorpus (tempcorpus-class), 73
- tempcorpus-class, 73
- TermDocumentMatrix, 73
- terms, partition-method  
(terms-partition-method), 74
- terms-partition-method, 74
- textstat-class, 74
- TokenStream-class, 76
- trim, 77
- trim, context-method (context-class), 16
- trim, dispersion-method  
(dispersion-class), 30
- trim, DocumentTermMatrix-method (trim),  
77
- trim, TermDocumentMatrix-method (trim),  
77
- trim-method (trim), 77
- tTest, 78
- tTest, context-method (tTest), 78
- unique, bundle-method (bundle-class), 12
- use, 45, 78
- view, 79
- view, cooccurrences-method  
(partition\_class), 62
- view, cooccurrencesReshaped-method  
(cooccurrences-class), 20
- view, features-method (features-class),  
36
- view, kwic-method (kwic-class), 48
- view, partition-method  
(partition\_class), 62
- view, textstat-method (textstat-class),  
74
- weigh, 79
- weigh, DocumentTermMatrix-method  
(weigh), 79
- weigh, TermDocumentMatrix-method  
(weigh), 79