Package 'tm.plugin.koRpus'

March 7, 2018

Type Package

Title A Compatibility Plugin Package for 'tm' and 'koRpus'

Author m.eik michalke [aut, cre]

Maintainer m.eik michalke <meik.michalke@hhu.de>

Depends R (>= 2.10.0),koRpus (>= 0.11-3),sylly

Imports methods,parallel,tm,NLP

Suggests testthat,knitr,rmarkdown

VignetteBuilder knitr

Description Provides classes and methods to enhance the ability to use the 'koRpus' package together with the 'tm' package. It is in its early stages. To ask for help, report bugs, suggest feature improvements, or discuss the global development of the package, please subscribe to the koRpus-dev mailing list (<http://korpusml.reaktanz.de>).

License GPL (>= 3)

Encoding UTF-8

LazyLoad yes

URL https://reaktanz.de/?c=hacking&s=koRpus

BugReports https://github.com/unDocUMeantIt/tm.plugin.koRpus/issues

Version 0.01-4

Date 2018-03-07

RoxygenNote 6.0.1

Collate '01_class_01_kRp.corpus.R'
'01_class_02_kRp.sourcesCorpus.R'
'01_class_03_kRp.topicCorpus.R'
'02_method_01_kRp.corpus-class_readability.R'
'02_method_02_kRp.corpus-class_hyphen.R'
'02_method_03_kRp.corpus-class_lex.div.R'
'02_method_04_kRp.corpus-class_read.corp.custom.R'
'02_method_05_kRp.corpus-class_freq.analysis.R'

'02_method_06_kRp.corpus-class_summary.R'
'02_method_07_kRp.corpus-class_correct.R'
'02_method_20_kRp.corpus_get_set_is.R'
'kRpSource.R'
'simpleCorpus.R'
'sourcesCorpus.R'
'tm.plugin.koRpus-internal.R'
'tm.plugin.koRpus-package.R'
'topicCorpus.R'

R topics documented:

`tm.plugin.koRpus-package`
The tm.plugin.koRpus Package

Description

A Compatibility Plugin Package for 'tm' and 'koRpus'.

Details

Package:	tm.plugin.koRpus
Type:	Package
Version:	0.01-4
Date:	2018-03-07
Depends:	R (>= 2.10.0),koRpus (>= 0.11-3),sylly
Encoding:	UTF-8

License:	GPL (>= 3)
LazyLoad:	yes
URL:	https://reaktanz.de/?c=hacking&s=koRpus

Provides classes and methods to enhance the ability to use the 'koRpus' package together with the 'tm' package. It is in its early stages. To ask for help, report bugs, suggest feature improvements, or discuss the global development of the package, please subscribe to the koRpus-dev mailing list (<http://korpusml.reaktanz.de>).

Author(s)

m.eik michalke

`corpusTagged` *Getter/setter methods for kRp.corpus objects*

Description

These methods should be used to get or set values of text objects generated by functions like `simpleCorpus`.

Usage

```
corpusTagged(obj)

## S4 method for signature 'kRp.corpus'
corpusTagged(obj)

corpusTagged(obj) <- value

## S4 replacement method for signature 'kRp.corpus'
corpusTagged(obj) <- value

corpusReadability(obj)

## S4 method for signature 'kRp.corpus'
corpusReadability(obj)

corpusReadability(obj) <- value

## S4 replacement method for signature 'kRp.corpus'
corpusReadability(obj) <- value

corpusTm(obj)

## S4 method for signature 'kRp.corpus'
```

```
corpusTm(obj)

corpusTm(obj) <- value

## S4 replacement method for signature 'kRp.corpus'
corpusTm(obj) <- value

corpusMeta(obj, meta = NULL, fail = TRUE)

## S4 method for signature 'kRp.corpus'
corpusMeta(obj, meta = NULL, fail = TRUE)

corpusMeta(obj, meta = NULL) <- value

## S4 replacement method for signature 'kRp.corpus'
corpusMeta(obj, meta = NULL) <- value

corpusHyphen(obj)

## S4 method for signature 'kRp.corpus'
corpusHyphen(obj)

corpusHyphen(obj) <- value

## S4 replacement method for signature 'kRp.corpus'
corpusHyphen(obj) <- value

corpusTTR(obj)

## S4 method for signature 'kRp.corpus'
corpusTTR(obj)

corpusTTR(obj) <- value

## S4 replacement method for signature 'kRp.corpus'
corpusTTR(obj) <- value

corpusSources(obj, source = NULL)

## S4 method for signature 'kRp.sourcesCorpus'
corpusSources(obj, source = NULL)

corpusSources(obj, source = NULL) <- value

## S4 replacement method for signature 'kRp.sourcesCorpus'
corpusSources(obj, source = NULL) <- value

corpusTopics(obj, topic = NULL)
```

```
## S4 method for signature 'kRp.topicCorpus'
corpusTopics(obj, topic = NULL)

corpusTopics(obj, topic = NULL) <- value

## S4 replacement method for signature 'kRp.topicCorpus'
corpusTopics(obj, topic = NULL) <- value

corpusFreq(obj)

## S4 method for signature 'kRp.corpus'
corpusFreq(obj)

## S4 method for signature 'kRp.sourcesCorpus'
corpusFreq(obj)

## S4 method for signature 'kRp.topicCorpus'
corpusFreq(obj)

corpusFreq(obj, topic) <- value

## S4 replacement method for signature 'kRp.corpus'
corpusFreq(obj) <- value

## S4 replacement method for signature 'kRp.sourcesCorpus'
corpusFreq(obj) <- value

## S4 replacement method for signature 'kRp.topicCorpus'
corpusFreq(obj) <- value

is.corpus(obj)

## S4 method for signature 'kRp.corpus'
x[i, j]

## S4 method for signature 'kRp.sourcesCorpus'
x[i, j]

## S4 method for signature 'kRp.topicCorpus'
x[i, j]

## S4 replacement method for signature 'kRp.corpus'
x[i, j] <- value

## S4 replacement method for signature 'kRp.sourcesCorpus'
x[i, j] <- value
```

```
## S4 replacement method for signature 'kRp.topicCorpus'
x[i, j] <- value

## S4 method for signature 'kRp.corpus'
x[[i]]

## S4 method for signature 'kRp.sourcesCorpus'
x[[i]]

## S4 method for signature 'kRp.topicCorpus'
x[[i]]

## S4 replacement method for signature 'kRp.corpus'
x[[i]] <- value

## S4 replacement method for signature 'kRp.sourcesCorpus'
x[[i]] <- value

## S4 replacement method for signature 'kRp.topicCorpus'
x[[i]] <- value

tif_as_tokens_df(tokens)

## S4 method for signature 'kRp.corpus'
tif_as_tokens_df(tokens)

## S4 method for signature 'kRp.sourcesCorpus'
tif_as_tokens_df(tokens)

## S4 method for signature 'kRp.topicCorpus'
tif_as_tokens_df(tokens)
```

Arguments

obj	An object of class kRp.corpus, kRp.sourcesCorpus or kRp.topicCorpus.
value	The new value to replace the current with.
meta	If not NULL, the meta list entry of the given name.
fail	Logical, whether the method should fail with an error if meta was not found. If set to FALSE, returns invisible(NULL) instead.
source	Character value, name of a source in this object. If given, you get/set only the value of this source.
topic	Character value, name of a topic in this object. If given, you get/set only the value of this topic.
x	See obj.
i	Defines the row selector ([) or the name to match ([[) in the summary slot.
j	Defines the column selector in the summary slot.
tokens	See obj.

Details

- corpusTagged() returns the list of kRp.tagged objects.
- corpusReadability() returns the list of kRp.readability objects.
- corpusTm() returns the VCorpus object.
- corpusMeta() returns the list with meta information.
- corpusHyphen() returns the list of kRp.hyphen objects.
- corpusTTR() returns the list of kRp.TTR objects.
- [/[[Can be used as a shortcut to index the results of corpusSummary().
- tif_as_tokens_df returns the TT.res slots of all texts in a single TIF[1] compliant data.frame, i.e., doc_id is not a factor but a character vector.

References

[1] Text Interchange Formats (https://github.com/ropensci/tif)

Examples

```
## Not run:
corpusTagged(myCorpus)
corpusMeta(myCorpus, "note") <- "an interesting read!"

## End(Not run)
```

---

correct.hyph,kRp.corpus-method

Methods to correct kRp.copus objects

Description

These methods enable you to correct errors that occurred during automatic processing, e.g., wrong hyphenation.

Usage

```
## S4 method for signature 'kRp.corpus'
correct.hyph(obj, word = NULL, hyphen = NULL,
  cache = TRUE)
```

Arguments

obj	An object of class kRp.corpus, kRp.sourcesCorpus or kRp.topicCorpus.
word	A character string, the (possibly incorrectly hyphenated) word entry to be replaced with hyphen.
hyphen	A character string, the new manually hyphenated version of word. Mustn't contain anything other than characters of word plus the hyphenation mark "-".
cache	Logical, if TRUE, the given hyphenation will be added to the sessions' hyphenation cache. Existing entries for the same word will be replaced.

Details

For details on what these methods do on a per text object basis, please refer to the documentation of `correct.hyph` in the `sylly` package.

Value

An object of the same class as `obj`.

`freq.analysis,kRp.corpus-method`

Apply freq.analysis() to all texts in kRp.corpus objects

Description

This method calls `freq.analysis` on all tagged text objects inside the given `txt.file` object (using `lapply`).

Usage

```
## S4 method for signature 'kRp.corpus'
freq.analysis(txt.file,
  mc.cores = getOption("mc.cores", 1L), ...)

## S4 method for signature 'kRp.sourcesCorpus'
freq.analysis(txt.file,
  mc.cores = getOption("mc.cores", 1L), ...)

## S4 method for signature 'kRp.topicCorpus'
freq.analysis(txt.file,
  mc.cores = getOption("mc.cores", 1L), ...)
```

Arguments

`txt.file`	An object of class `kRp.corpus`, `kRp.sourcesCorpus` or `kRp.topicCorpus`.
`mc.cores`	The number of cores to use for parallelization, see `mclapply`.
`...`	options to pass through to `freq.analysis`.

Value

An object of the same class as `txt.file`.

Examples

```
## Not run:
myTexts <- simpleCorpus(dir=file.path("/home","me","textCorpus"))
myTexts <- freq.analysis(myTexts)

## End(Not run)
```

hyphen,kRp.corpus-method *Apply hyphen() to all texts in kRp.corpus objects*

Description

This method calls hyphen on all tagged text objects inside the given words object (using lapply).

Usage

```
## S4 method for signature 'kRp.corpus'
hyphen(words, mc.cores = getOption("mc.cores", 1L),
  quiet = TRUE, ...)

## S4 method for signature 'kRp.sourcesCorpus'
hyphen(words, mc.cores = getOption("mc.cores",
  1L), quiet = TRUE, ...)

## S4 method for signature 'kRp.topicCorpus'
hyphen(words, mc.cores = getOption("mc.cores",
  1L), quiet = TRUE, ...)
```

Arguments

words	An object of class kRp.corpus, kRp.sourcesCorpus or kRp.topicCorpus.
mc.cores	The number of cores to use for parallelization, see mclapply.
quiet	Logical, if FALSE shows a status bar for the hyphenation process of each text.
...	options to pass through to hyphen.

Value

An object of the same class as words.

Examples

```
## Not run:
myTexts <- simpleCorpus(dir=file.path("/home","me","textCorpus"))
myTexts <- hyphen(myTexts)

## End(Not run)
```

kRp.corpus,-class *S4 Class kRp.corpus*

Description

Objects of this class can contain multiple texts simultaneously. It supports both the tm package's Corpus class and koRpus' own object classes and stores them in separated slots.

Details

Objects should be created using the simpleCorpus function.

Slots

summary A summary data frame for the full corpus.

meta A named list. Can be used to store meta information. Currently, no particular format is defined.

raw A list of objects of class Corpus.

tagged A list of objects of class kRp.taggedText (a class union for tagged text objects).

hyphen A list of objects of class kRp.hyphen.

TTR A list of objects of class kRp.TTR.

readability A list of objects of class kRp.readability.

freq A list with two elements, texts and corpus. Both hold objects of class kRp.corp.freq, where texts is a list of these objects (one for each text), and corpus is a single object for the full corpus.

Contructor function

Should you need to manually generate objects of this class (which should rarely be the case), the contructor function kRp_corpus(...) can be used instead of new("kRp.corpus", ...). Whenever possible, stick to simpleCorpus.

Note

There is also getter and setter methods for objects of this class.

kRpSource *A source function for tm*

Description

An rather untested attempt to sketch a Source function for tm. Supposed to be used to translate tagged koRpus objects into tm objects.

Usage

```
kRpSource(obj, encoding = "UTF-8")
```

Arguments

obj	An object of class kRp.taggedText (a class union for tagged text objects).
encoding	Character string, defining the character encoding of the object.

Details

Also provided are the methods getElem and pGetElem for S3 class kRpSource.

Value

An object of class Source, also inheriting class kRpSource.

---

kRp.sourcesCorpus,-class *S4 Class kRp.sourcesCorpus*

Description

Objects of this class can contain multiple texts simultaneously. Adding to that, these texts are ordered by their source in a number of slots. Each slot is a list of entries, one for each source.

Details

Objects should be created using the sourcesCorpus function.

Slots

summary A summary data.frame for all sources combined.

paths A list of character strings with paths to all files

sources A list of objects of class kRp.corpus

files A list of character strings with only the file names of all texts

freq An object of class kRp.corp.freq, can contain word frequency information on the full corpus if this object was analysed with read.corp.custom.

Contructor function

Should you need to manually generate objects of this class (which should rarely be the case), the contructor function `kRp_sourcesCorpus(...)` can be used instead of `new("kRp.sourcesCorpus", ...)`. Whenever possible, stick to `sourcesCorpus`.

Note

There is also `getter and setter methods` for objects of this class.

---

kRp.topicCorpus,-class

S4 Class kRp.topicCorpus and contructor

Description

Objects of this class can contain multiple texts simultaneously. Adding to that, these texts can be ordered at two levels: topic and source. This is useful for comparisons if you have a defined number of topics and textst from different sources on these topics. Note however, that there must be at least one text on each topic in all of the given sources.

Details

For each combination of topic and source, there is exactly one object of class `kRp.corpus`.

Objects should be created using the `topicCorpus` function.

Slots

`summary` A summary data.frame for all topics combined.

`topics` A named list of nested objects. Each element is named after a topic and contains an object of class kRp.sourcesCorpus.

`freq` An object of class `kRp.corp.freq`, can contain word frequency information on the full corpus if this object was analysed with `read.corp.custom`.

Contructor function

Should you need to manually generate objects of this class (which should rarely be the case), the contructor function `kRp_topicCorpus(...)` can be used instead of `new("kRp.topicCorpus", ...)`. Whenever possible, stick to `topicCorpus`.

Note

There is also `getter and setter methods` for objects of this class.

lex.div,kRp.corpus-method

Apply lex.div() to all texts in kRp.corpus objects

Description

This method calls lex.div on all tagged text objects inside the given txt object (using lapply).

Usage

```
## S4 method for signature 'kRp.corpus'
lex.div(txt, summary = TRUE,
  mc.cores = getOption("mc.cores", 1L), char = "", quiet = TRUE, ...)

## S4 method for signature 'kRp.sourcesCorpus'
lex.div(txt, summary = TRUE,
  mc.cores = getOption("mc.cores", 1L), char = "", quiet = TRUE, ...)

## S4 method for signature 'kRp.topicCorpus'
lex.div(txt, summary = TRUE,
  mc.cores = getOption("mc.cores", 1L), char = "", quiet = TRUE, ...)
```

Arguments

txt	An object of class kRp.corpus, kRp.sourcesCorpus or kRp.topicCorpus.
summary	Logical, determines if the summary slot should automatically be updated by calling summary on the result.
mc.cores	The number of cores to use for parallelization, see mclapply.
char	Character vector to specify measures of which characteristics should be computed, see lex.div for details.
quiet	Logical, if FALSE shows a status bar for some measures of each text, see lex.div for details.
...	options to pass through to lex.div.

Value

An object of the same class as txt.

Examples

```
## Not run:
myTexts <- simpleCorpus(dir=file.path("/home","me","textCorpus"))
myTexts <- lex.div(myTexts)

## End(Not run)
```

readability,kRp.corpus-method

Apply readability() to all texts in kRp.corpus objects

Description

This method calls readability on all tagged text objects inside the given txt.file object (using lapply).

Usage

```
## S4 method for signature 'kRp.corpus'
readability(txt.file, summary = TRUE,
  mc.cores = getOption("mc.cores", 1L), quiet = TRUE, ...)

## S4 method for signature 'kRp.sourcesCorpus'
readability(txt.file, summary = TRUE,
  mc.cores = getOption("mc.cores", 1L), quiet = TRUE, ...)

## S4 method for signature 'kRp.topicCorpus'
readability(txt.file, summary = TRUE,
  mc.cores = getOption("mc.cores", 1L), quiet = TRUE, ...)
```

Arguments

txt.file	An object of class kRp.corpus, kRp.sourcesCorpus or kRp.topicCorpus.
summary	Logical, determines if the summary slot should automatically be updated by calling summary on the result.
mc.cores	The number of cores to use for parallelization, see mclapply.
quiet	Logical, if FALSE shows a status bar for some calculations of each text, see readability for details.
...	options to pass through to readability.

Value

An object of the same class as txt.file.

Examples

```
## Not run:
myTexts <- simpleCorpus(dir=file.path("/home","me","textCorpus"))
myTexts <- readability(myTexts)

## End(Not run)
```

`read.corp.custom,kRp.corpus-method`
Apply read.corp.custom() to all texts in kRp.corpus objects

Description

This method calls `read.corp.custom` on all tagged text objects inside the given `corpus` object (using `lapply`).

Usage

```
## S4 method for signature 'kRp.corpus'
read.corp.custom(corpus,
  mc.cores = getOption("mc.cores", 1L), ...)

## S4 method for signature 'kRp.sourcesCorpus'
read.corp.custom(corpus,
  mc.cores = getOption("mc.cores", 1L), ...)

## S4 method for signature 'kRp.topicCorpus'
read.corp.custom(corpus,
  mc.cores = getOption("mc.cores", 1L), ...)
```

Arguments

`corpus`	An object of class `kRp.corpus`, `kRp.sourcesCorpus` or `kRp.topicCorpus`.
`mc.cores`	The number of cores to use for parallelization, see `mclapply`.
`...`	options to pass through to `read.corp.custom`.

Value

An object of the same class as `corpus`.

Examples

```
## Not run:
myBasePath <- file.path("/home","me","textCorpus")
# analyse a single corpus
myTexts <- simpleCorpus(dir=myBasePath)
myTexts <- read.corp.custom(myTexts)

# you can also analyse multiple corpora from different sources simultaneously
# the following would assume that below 'myBasePath', there are subfolders
# named "Wikipedia" and Journal of Applied Geschwurbel"
mySources <- c(
   wp="Wikipedia",
   jg="Journal of Applied Geschwurbel"
 )
```

```
mySourcesTexts <- sourcesCorpus(
  path=myBasePath,
  sources=mySources
)
# this will also call read.corp.custom() recursively on
# the single text level
mySourcesTexts <- read.corp.custom(mySourcesTexts)

# and you might have guessed, you can also add a topic level
# with the following two vectors you would describe your
# data structure as two subfolders called "proc" and "gesw"
# below myBasePath, and two further subfolders named "Wikipedia"
# and Journal of Applied Geschwurbel" below each of the topic folders,
# containing the actual texts
myTopicPaths <- c(
  procrastination=file.path(myBasePath, "proc"),
  geschwurbel=file.path(myBasePath, "gesw")
)
mySources <- c(
   wp="Wikipedia",
   jg="Journal of Applied Geschwurbel"
 )
myTopicTexts <- topicCorpus(
  paths=myTopicPaths,
  sources=mySources
)
# this will also call read.corp.custom() recursively on
# the source and single text level
myTopicTexts <- read.corp.custom(myTopicTexts)


## End(Not run)
```

`simpleCorpus` *Function to create kRp.corpus objects from directory or object content*

Description

This function is a combined wrapper that calls `DirSource` (or `VectorSource`, if `format="obj"`), `VCorpus` and `tokenize` or `treetag`.

Usage

```
simpleCorpus(dir = ".", lang = "kRp.env", tagger = "kRp.env",
  encoding = "", pattern = NULL, recursive = FALSE, ignore.case = FALSE,
  mode = "text", source = "", topic = "", format = "file",
  mc.cores = getOption("mc.cores", 1L), ...)
```

Arguments

dir	Character vector with path names to search for text files, or the actual texts to be analyzed if format="obj". See DirSource and VectorSource for details.
lang	A character string naming the language of the analyzed corpus. See kRp.POS.tags for all supported languages. If set to "kRp.env" this is got from get.kRp.env. This information will also be passed to the readerControl list of the VCorpus call.
tagger	A character string pointing to the tokenizer/tagger command you want to use for basic text analysis. Defaults to tagger="kRp.env" to get the settings by get.kRp.env. Set to "tokenize" to use tokenize.
encoding	Character string describing the current encoding. See DirSource for details, omitted if format="obj".
pattern	A regular expression for file matching. See DirSource for details, omitted if format="obj".
recursive	Logical, indicates whether directories should be read recursively. See DirSource for details, omitted if format="obj".
ignore.case	Logical, indicates whether pattern is matched case sensitive. See DirSource for details, omitted if format="obj".
mode	Character string defining the reading mode. See DirSource for details, omitted if format="obj".
source	Character string, naming the source of the corpus.
topic	Character string, a topic this corpus deals with.
format	Either "file" or "obj", depending on whether you want to scan files or analyze the text in a given object, like a character vector. If the latter and treetag is used as the tagger, texts will be written to temporary files for the process (see dir).
mc.cores	The number of cores to use for parallelization, see mclapply.
...	Additional options which are passed through to the defined tagger.

Details

The result, if succeeded, is a single object of class kRp.corpus, which includes all read texts in a tm style VCorpus format, as well as in koRpus style kRp.taggedText class format.

Value

An object of class kRp.corpus.

sourcesCorpus *Function to create kRp.sourcesCorpus objects from directory or object content*

Description

Internally, this is a wrapper for simpleCorpus you can use to analyze texts from more than one source.

Usage

```
sourcesCorpus(path, sources, topic = "", format = "file",
  mc.cores = getOption("mc.cores", 1L), ...)
```

Arguments

path	Usually a charcter vector with the path to the texts to analyse. Below this path, texts must be ordered into subfolders named exactly like defined by sources. However, if format="obj", you use this argument to provide the texts themselves as a list of named character vectors, where the name must match the names of sources and each element of the vectors is a single text.
sources	A named character vector defining all sources to regard. The names of each entry will be used internally for the sources.
topic	A character string naming the topic covered.
format	Either "file" or "obj", depending on whether you want to scan files or analyze the text in a given object, like a character vector. See simpleCorpus for more information.
mc.cores	The number of cores to use for parallelization, see mclapply. This value is passed through to simpleCorpus.
...	Additional options, passed through to simpleCorpus.

Examples

```
## Not run:
myTopic <- sourcesCorpus("~/data/foo", sources=c(bar="The Bar Magazine",
      baz="BAZ Monthly"))

# providing text directly
myTopic <- sourcesCorpus(
  list(
    source1=c("the first text.", "the second text."),
    source2=c("the third text.", "the last text.")
  ),
  sources=c(
    source1="The Important",
    source2="Must Read Magazine"
  ),
```

```
  topic="short stories",
  format="obj"
)

## End(Not run)
```

---

summary,kRp.corpus-method

Apply summary() to all texts in kRp.corpus objects

Description

This method performs a summary call on all text objects inside the given object object (using lapply). Contrary to what other summary methods do, these methods always return the full object with an updated summary slot.

Usage

```
## S4 method for signature 'kRp.corpus'
summary(object, missing = NA, ...)

## S4 method for signature 'kRp.sourcesCorpus'
summary(object, ...)

## S4 method for signature 'kRp.topicCorpus'
summary(object)

corpusSummary(obj)

## S4 method for signature 'kRp.corpus'
corpusSummary(obj)

## S4 method for signature 'kRp.sourcesCorpus'
corpusSummary(obj)

## S4 method for signature 'kRp.topicCorpus'
corpusSummary(obj)

corpusSummary(obj) <- value

## S4 replacement method for signature 'kRp.corpus'
corpusSummary(obj) <- value

## S4 replacement method for signature 'kRp.sourcesCorpus'
corpusSummary(obj) <- value

## S4 replacement method for signature 'kRp.topicCorpus'
corpusSummary(obj) <- value
```

Arguments

object	An object of class kRp.corpus, kRp.sourcesCorpus or kRp.topicCorpus.
missing	Character string to use for missing values.
...	Used for internal processes.
obj	An object of class kRp.corpus, kRp.sourcesCorpus or kRp.topicCorpus.
value	The new value to replace the current with.

Details

The methods for nested object classes also recursively invoke the summary methods for lower corpus object classes; i.e., if you call summary on an object of class kRp.topicCorpus, the nested objects of class kRp.sourcesCorpus will also be summarised, and that in turn causes summaries on all nested corpora of class kRp.corpus.

The summary slot contains a data.frame with aggregated information of all texts that the respective object level contains.

Value

An object of the same class as object.

Examples

```
## Not run:
myTexts <- simpleCorpus(dir=file.path("/home","me","textCorpus"))
summary(myTexts)

## End(Not run)
```

---

topicCorpus *Function to create kRp.topicCorpus objects from directory or object content*

Description

Internally, this is a wrapper for sourcesCorpus you can use to analyze texts both from more than one source, and on different topics.

Usage

```
topicCorpus(paths, sources, format = "file",
  mc.cores = getOption("mc.cores", 1L), ...)
```

Arguments

paths	A named list, usually with the paths to the texts to analyze. Each named list element will define one topic. Below this path, texts must be ordered into sub-folders named exactly like defined by sources. However, if format="obj", you use this argument to provide the texts themselves as a list of named lists of named character vectors, where the names must match the names of sources and each element of the vectors is a single text. See the examples below.
sources	A named character vector defining all sources to regard. The names of each entry will be used internally for the sources.
format	Either "file" or "obj", depending on whether you want to scan files or analyze the text in a given object, like a character vector. See sourcesCorpus for more information.
mc.cores	The number of cores to use for parallelization, see mclapply. This value is passed through to simpleCorpus.
...	Additional options, passed through to sourcesCorpus.

Examples

```
## Not run:
myOrderedCorpus <- topicCorpus(
  paths=list(
    lottery="~/data/foo/lottery",
    waste="~/data/foo/waste"),
  sources=c(
    bar="The Bar Magazine",
    baz="BAZ Monthly")
)

# providing texts directly
myOrderedCorpus <- topicCorpus(
  paths=list(
    lottery=list(
      source1=c("the first text.", "the second text."),
      source2=c("the third text.", "the fourth text.")
    ),
    waste=list(
      source1=c("the fifth text.", "the sixth text."),
      source2=c("the seventh text.", "the last text.")
    )
  ),
  sources=c(
    source1="The Important",
    source2="Must Read Magazine"
  ),
  format="obj"
)

## End(Not run)
```

Index