

Package 'tm.plugin.koRpus'

July 12, 2016

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),methods,parallel,koRpus (>= 0.07-1),tm

Suggests testthat

Description This package 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: https://ml06.ispgateway.de/mailman/listinfo/korpus-dev_r.reaktanz.de

License GPL (>= 3)

Encoding UTF-8

LazyLoad yes

URL <http://reaktanz.de/?c=hacking&s=koRpus>

Version 0.01-3

Date 2016-07-12

RoxygenNote 5.0.1

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	2
corpusTagged	3
correct.hyph,kRp.corpus-method	6
freq.analysis,kRp.corpus-method	7
hyphen,kRp.corpus-method	8
kRp.corpus,-class	9
kRp.sourcesCorpus,-class	9
kRp.topicCorpus,-class	10
kRpSource	10
lex.div,kRp.corpus-method	11
read.corp.custom,kRp.corpus-method	12
readability,kRp.corpus-method	13
simpleCorpus	14
sourcesCorpus	16
summary,kRp.corpus-method	17
topicCorpus	18
Index	20

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: 0.01-3
 Date: 2016-07-12
 Depends: R (>= 2.10.0),methods,parallel,koRpus (>= 0.07-1),tm
 Encoding: UTF-8
 License: GPL (>= 3)
 LazyLoad: yes
 URL: <http://reaktanz.de/?c=hacking&s=koRpus>

This package 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: https://ml06.ispgateway.de/mailman/listinfo/korpus-dev_r.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)

```

Arguments

obj	An object of class <code>kRp.corpus-class</code> , <code>kRp.sourcesCorpus-class</code> or <code>kRp.topicCorpus-class</code> .
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 <code>invisible(NULL)</code> 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.

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.

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-class , kRp.sourcesCorpus-class or kRp.topicCorpus-class .
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 koRpus 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

<code>txt.file</code>	An object of class kRp.corpus-class , kRp.sourcesCorpus-class or kRp.topicCorpus-class .
<code>mc.cores</code>	The number of cores to use for parallelization, see mclapply .
<code>...</code>	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),  
  ...)  
  
## S4 method for signature 'kRp.sourcesCorpus'  
hyphen(words, mc.cores = getOption("mc.cores",  
  1L), ...)  
  
## S4 method for signature 'kRp.topicCorpus'  
hyphen(words, mc.cores = getOption("mc.cores",  
  1L), ...)
```

Arguments

<code>words</code>	An object of class kRp.corpus-class , kRp.sourcesCorpus-class or kRp.topicCorpus-class .
<code>mc.cores</code>	The number of cores to use for parallelization, see mclapply .
<code>...</code>	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 can 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-class` (a class union for tagged text objects).

`hyphen` A list of objects of class `kRp.hyphen-class`.

`TTR` A list of objects of class `kRp.TTR-class`.

`readability` A list of objects of class `kRp.readability-class`.

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

Note

There is also methods to transform get the lists of all particular slots from objects of this class.

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.

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-class](#), can contain word frequency information on the full corpus if this object was analysed with [read.corp.custom](#).

kRp.topicCorpus, -class

S4 Class kRp.topicCorpus

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](#).

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-class](#), can contain word frequency information on the full corpus if this object was analysed with [read.corp.custom](#).

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 <code>kRp.taggedText-class</code> (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`.

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), ...)

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

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

Arguments

txt	An object of class <code>kRp.corpus-class</code> , <code>kRp.sourcesCorpus-class</code> or <code>kRp.topicCorpus-class</code> .
summary	Logical, determines if the summary slot should automatically be updated by calling <code>summary</code> on the result.
mc.cores	The number of cores to use for parallelization, see <code>mclapply</code> .
...	options to pass through to <code>lex.div</code> .

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)
```

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-class , kRp.sourcesCorpus-class or kRp.topicCorpus-class .
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)

```

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), ...)

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

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

Arguments

txt.file	An object of class <code>kRp.corpus-class</code> , <code>kRp.sourcesCorpus-class</code> or <code>kRp.topicCorpus-class</code> .
summary	Logical, determines if the summary slot should automatically be updated by calling <code>summary</code> on the result.
mc.cores	The number of cores to use for parallelization, see <code>mclapply</code> .
...	options to pass through to <code>readability</code> .

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)
```

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

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

Details

The result, if succeeded, is a single object of class [kRp.corpus-class](#), 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-class](#).

sourcesCorpus	<i>Function to create kRp.sourcesCorpus objects from directory or object content</i>
---------------	--

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 character 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 <code>kRp.corpus-class</code> , <code>kRp.sourcesCorpus-class</code> or <code>kRp.topicCorpus-class</code> .
missing	Character string to use for missing values.
...	Used for internal processes.
obj	An object of class <code>kRp.corpus-class</code> , <code>kRp.sourcesCorpus-class</code> or <code>kRp.topicCorpus-class</code> .
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	<i>Function to create kRp.topicCorpus objects from directory or object content</i>
-------------	--

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

*Topic **classes**

- kRp.corpus, -class, 9
- kRp.sourcesCorpus, -class, 9
- kRp.topicCorpus, -class, 10

*Topic **package**

- tm.plugin.koRpus-package, 2

Corpus, 9

- corpusFreq (corpusTagged), 3
- corpusFreq, -methods (corpusTagged), 3
- corpusFreq, kRp.corpus-method (corpusTagged), 3
- corpusFreq, kRp.sourcesCorpus-method (corpusTagged), 3
- corpusFreq, kRp.topicCorpus-method (corpusTagged), 3
- corpusFreq<- (corpusTagged), 3
- corpusFreq<-, -methods (corpusTagged), 3
- corpusFreq<-, kRp.corpus-method (corpusTagged), 3
- corpusFreq<-, kRp.sourcesCorpus-method (corpusTagged), 3
- corpusFreq<-, kRp.topicCorpus-method (corpusTagged), 3
- corpusHyphen (corpusTagged), 3
- corpusHyphen, -methods (corpusTagged), 3
- corpusHyphen, kRp.corpus-method (corpusTagged), 3
- corpusHyphen<- (corpusTagged), 3
- corpusHyphen<-, -methods (corpusTagged), 3
- corpusHyphen<-, kRp.corpus-method (corpusTagged), 3
- corpusMeta (corpusTagged), 3
- corpusMeta, -methods (corpusTagged), 3
- corpusMeta, kRp.corpus-method (corpusTagged), 3
- corpusMeta<- (corpusTagged), 3
- corpusMeta<-, -methods (corpusTagged), 3

- corpusMeta<-, kRp.corpus-method (corpusTagged), 3
- corpusReadability (corpusTagged), 3
- corpusReadability, -methods (corpusTagged), 3
- corpusReadability, kRp.corpus-method (corpusTagged), 3
- corpusReadability<- (corpusTagged), 3
- corpusReadability<-, -methods (corpusTagged), 3
- corpusReadability<-, kRp.corpus-method (corpusTagged), 3
- corpusSources (corpusTagged), 3
- corpusSources, -methods (corpusTagged), 3
- corpusSources, kRp.sourcesCorpus-method (corpusTagged), 3
- corpusSources<- (corpusTagged), 3
- corpusSources<-, -methods (corpusTagged), 3
- corpusSources<-, kRp.sourcesCorpus-method (corpusTagged), 3
- corpusSummary (summary, kRp.corpus-method), 17
- corpusSummary, -methods (summary, kRp.corpus-method), 17
- corpusSummary, kRp.corpus-method (summary, kRp.corpus-method), 17
- corpusSummary, kRp.sourcesCorpus-method (summary, kRp.corpus-method), 17
- corpusSummary, kRp.topicCorpus-method (summary, kRp.corpus-method), 17
- corpusSummary<- (summary, kRp.corpus-method), 17
- corpusSummary<-, -methods (summary, kRp.corpus-method), 17
- corpusSummary<-, kRp.corpus-method (summary, kRp.corpus-method), 17
- corpusSummary<-, kRp.sourcesCorpus-method (summary, kRp.corpus-method), 17

- corpusSummary<- ,kRp.topicCorpus-method
(summary,kRp.corpus-method), 17
- corpusTagged, 3
- corpusTagged,-methods (corpusTagged), 3
- corpusTagged,kRp.corpus-method
(corpusTagged), 3
- corpusTagged<- (corpusTagged), 3
- corpusTagged<-,-methods (corpusTagged),
3
- corpusTagged<- ,kRp.corpus-method
(corpusTagged), 3
- corpusTm (corpusTagged), 3
- corpusTm,-methods (corpusTagged), 3
- corpusTm,kRp.corpus-method
(corpusTagged), 3
- corpusTm<- (corpusTagged), 3
- corpusTm<-,-methods (corpusTagged), 3
- corpusTm<- ,kRp.corpus-method
(corpusTagged), 3
- corpusTopics (corpusTagged), 3
- corpusTopics,-methods (corpusTagged), 3
- corpusTopics,kRp.topicCorpus-method
(corpusTagged), 3
- corpusTopics<- (corpusTagged), 3
- corpusTopics<-,-methods (corpusTagged),
3
- corpusTopics<- ,kRp.topicCorpus-method
(corpusTagged), 3
- corpusTTR (corpusTagged), 3
- corpusTTR,-methods (corpusTagged), 3
- corpusTTR,kRp.corpus-method
(corpusTagged), 3
- corpusTTR<- (corpusTagged), 3
- corpusTTR<-,-methods (corpusTagged), 3
- corpusTTR<- ,kRp.corpus-method
(corpusTagged), 3
- correct.hyph, 6
- correct.hyph
(correct.hyph,kRp.corpus-method),
6
- correct.hyph,kRp.corpus-method, 6
- DirSource, 14, 15
- freq.analysis, 7
- freq.analysis,kRp.corpus-method, 7
- freq.analysis,kRp.sourcesCorpus-method
(freq.analysis,kRp.corpus-method),
7
- freq.analysis,kRp.topicCorpus-method
(freq.analysis,kRp.corpus-method),
7
- get.kRp.env, 15
- hyphen, 8
- hyphen,kRp.corpus-method, 8
- hyphen,kRp.sourcesCorpus-method
(hyphen,kRp.corpus-method), 8
- hyphen,kRp.topicCorpus-method
(hyphen,kRp.corpus-method), 8
- is.corpus (corpusTagged), 3
- kRp.corpus, 10
- kRp.corpus,-class, 9
- kRp.corpus-class (kRp.corpus,-class), 9
- kRp.POS.tags, 15
- kRp.sourcesCorpus,-class, 9
- kRp.sourcesCorpus-class
(kRp.sourcesCorpus,-class), 9
- kRp.topicCorpus,-class, 10
- kRp.topicCorpus-class
(kRp.topicCorpus,-class), 10
- kRpSource, 10
- lex.div, 11
- lex.div,kRp.corpus-method, 11
- lex.div,kRp.sourcesCorpus-method
(lex.div,kRp.corpus-method), 11
- lex.div,kRp.topicCorpus-method
(lex.div,kRp.corpus-method), 11
- mclapply, 7, 8, 11, 12, 14–16, 19
- read.corp.custom, 10, 12
- read.corp.custom
(read.corp.custom,kRp.corpus-method),
12
- read.corp.custom,kRp.corpus-method, 12
- read.corp.custom,kRp.sourcesCorpus-method
(read.corp.custom,kRp.corpus-method),
12
- read.corp.custom,kRp.topicCorpus-method
(read.corp.custom,kRp.corpus-method),
12
- readability, 13, 14
- readability,kRp.corpus-method, 13

readability, kRp.sourcesCorpus-method
(readability, kRp.corpus-method),
[13](#)

readability, kRp.topicCorpus-method
(readability, kRp.corpus-method),
[13](#)

simpleCorpus, [3](#), [9](#), [14](#), [16](#)

Source, [10](#), [11](#)

sourcesCorpus, [16](#), [18](#), [19](#)

summary, [11](#), [14](#)

summary (summary, kRp.corpus-method), [17](#)

summary, kRp.corpus-method, [17](#)

summary, kRp.sourcesCorpus-method
(summary, kRp.corpus-method), [17](#)

summary, kRp.topicCorpus-method
(summary, kRp.corpus-method), [17](#)

tm.plugin.koRpus-package, [2](#)

tokenize, [14](#), [15](#)

topicCorpus, [18](#)

treetag, [14](#), [15](#)

VCorpus, [14](#)

VectorSource, [14](#), [15](#)