Package 'tm.plugin.koRpus'

May 14, 2019

Type Package

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

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>).

Author m.eik michalke [aut, cre]

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

Depends R (>= 2.10.0),koRpus (>= 0.12-1),sylly

Imports methods,parallel,tm,NLP,Matrix

Suggests testthat,knitr,rmarkdown

VignetteBuilder knitr

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

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

License GPL (>= 3)

Encoding UTF-8

LazyLoad yes

Version 0.3-1

Date 2019-05-14

RoxygenNote 6.1.1

Collate '01_class_01_kRp.hierarchy.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_08_kRp.corpus-class_query.R'

'02_method_09_kRp.corpus-class_filterByClass.R'
'02_method_10_kRp.corpus-class_jumbleWords.R'
'02_method_11_kRp.corpus-class_clozeDelete.R'
'02_method_12_kRp.corpus-class_cTest.R'
'02_method_13_kRp.corpus-class_textTransform.R'
'02_method_14_kRp.corpus-class_docTermMatrix.R'
'02_method_20_kRp.corpus_get_set_is.R'
'02_method_21_kRp.corpus-class_show.R'
'deprecated.R'
'kRpSource.R'
'readCorpus.R'
'tm.plugin.koRpus-internal.R'
'tm.plugin.koRpus-package.R'

R topics documented:

`tm.plugin.koRpus-package`
A Compatibility Plugin Package for 'tm' and 'koRpus'

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>).

Details

The DESCRIPTION file:

Package:	tm.plugin.koRpus
Type:	Package
Version:	0.3-1
Date:	2019-05-14
Depends:	R (>= 2.10.0),koRpus (>= 0.12-1),sylly
Encoding:	UTF-8
License:	GPL (>= 3)
LazyLoad:	yes
URL:	https://reaktanz.de/?c=hacking&s=koRpus

Author(s)

m.eik michalke [aut, cre]

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

See Also

Useful links:

- https://reaktanz.de/?c=hacking&s=koRpus
- Report bugs at https://github.com/unDocUMeantIt/tm.plugin.koRpus/issues

---

clozeDelete,kRp.hierarchy-method

Apply clozeDelete() to all texts in kRp.hierarchy objects

Description

This method calls `clozeDelete` on all tagged text objects inside the given `obj` object (using `lapply`).

Usage

```
## S4 method for signature 'kRp.hierarchy'
clozeDelete(obj,
  mc.cores = getOption("mc.cores", 1L), ...)
```

Arguments

obj	An object of class kRp.hierarchy.
mc.cores	The number of cores to use for parallelization, see mclapply.
...	options to pass through to clozeDelete.

Value

An object of the same class as obj.

Examples

```
## Not run:
myCorpus <- readCorpus(
  dir=file.path(
    path.package("tm.plugin.koRpus"), "tests", "testthat", "samples", "C3S"
  ),
  hierarchy=list(
    Source=c(
      Wikipedia_alt="Wikipedia (alt)",
      Wikipedia_neu="Wikipedia (neu)"
    )
  )
)
# remove all punctuation
myCorpus <- clozeDelete(myCorpus)

## End(Not run)
```

---

corpusTagged *Getter/setter methods for kRp.hierarchy objects*

Description

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

Usage

```
corpusTagged(obj, level = NULL, id = NULL)

## S4 method for signature 'kRp.hierarchy'
corpusTagged(obj, level = NULL, id = NULL)

corpusTagged(obj) <- value

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

```
corpusReadability(obj, level = NULL, id = NULL)

## S4 method for signature 'kRp.hierarchy'
corpusReadability(obj, level = NULL,
  id = NULL)

corpusReadability(obj) <- value

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

corpusTm(obj, id = NULL)

## S4 method for signature 'kRp.hierarchy'
corpusTm(obj, id = NULL)

corpusTm(obj) <- value

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

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

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

corpusMeta(obj, meta = NULL) <- value

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

corpusHyphen(obj, level = NULL, id = NULL)

## S4 method for signature 'kRp.hierarchy'
corpusHyphen(obj, level = NULL, id = NULL)

corpusHyphen(obj) <- value

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

corpusTTR(obj, level = NULL, id = NULL)

## S4 method for signature 'kRp.hierarchy'
corpusTTR(obj, level = NULL, id = NULL)

corpusTTR(obj) <- value
```

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

corpusFreq(obj, level = NULL, id = NULL)

## S4 method for signature 'kRp.hierarchy'
corpusFreq(obj, level = NULL, id = NULL)

corpusFreq(obj) <- value

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

corpusLevel(obj)

## S4 method for signature 'kRp.hierarchy'
corpusLevel(obj)

corpusLevel(obj) <- value

## S4 replacement method for signature 'kRp.hierarchy'
corpusLevel(obj) <- value

corpusChildren(obj, level = NULL, id = NULL)

## S4 method for signature 'kRp.hierarchy'
corpusChildren(obj, level = NULL, id = NULL)

corpusChildren(obj) <- value

## S4 replacement method for signature 'kRp.hierarchy'
corpusChildren(obj) <- value

corpusCategory(obj, level = NULL, id = NULL)

## S4 method for signature 'kRp.hierarchy'
corpusCategory(obj, level = NULL, id = NULL)

corpusCategory(obj) <- value

## S4 replacement method for signature 'kRp.hierarchy'
corpusCategory(obj) <- value

corpusID(obj, level = NULL)

## S4 method for signature 'kRp.hierarchy'
corpusID(obj, level = NULL)
```

```
corpusID(obj) <- value

## S4 replacement method for signature 'kRp.hierarchy'
corpusID(obj) <- value

corpusPath(obj, level = NULL)

## S4 method for signature 'kRp.hierarchy'
corpusPath(obj, level = NULL)

corpusPath(obj) <- value

## S4 replacement method for signature 'kRp.hierarchy'
corpusPath(obj) <- value

corpusFiles(obj, level = 0, id = NULL, paths = FALSE)

## S4 method for signature 'kRp.hierarchy'
corpusFiles(obj, level = 0, id = NULL,
  paths = FALSE)

corpusFiles(obj) <- value

## S4 replacement method for signature 'kRp.hierarchy'
corpusFiles(obj) <- value

is.corpus(obj)

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

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

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

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

tif_as_tokens_df(tokens)

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

tif_as_corpus_df(corpus)
```

```
## S4 method for signature 'kRp.hierarchy'
tif_as_corpus_df(corpus)
```

Arguments

obj	An object of class kRp.hierarchy.
level	Either the integer value or name (character string) of the target level. If NULL the top level is used.
id	Character value specifying the category ID of child objects you want to get. If NULL then the child objects of the current top level are returned, otherwise all children with the ID specified. Only interpreted if level=NULL (you can't use both at the same time).
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.
paths	Logical, indicates for corpusFiles() whether full paths should be returned, or just the actual file name.
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	An object of class kRp.hierarchy.
corpus	An object of class kRp.hierarchy.

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.
- corpusLevel() returns the level value of the top level object.
- corpusCategory() returns the character vector of categories of the top level object.
- corpusID() returns the character vector of category IDs of the top level object.
- corpusPath() returns the root directory path of the top level object.
- corpusFiles() returns the character vector of file names of level 0 of the object.
- [/[[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.

Please note that the <- methods usually do not work on nested levels but only on single object nodes.

References

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

Examples

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

# export object to TIF compliant data frame
myCorpus_df <- tif_as_corpus_df(myCorpus)

## End(Not run)
```

---

correct.hyph,kRp.hierarchy-method

Methods to correct kRp.hierarchy objects

Description

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

Usage

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

Arguments

obj	An object of class kRp.hierarchy.
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.

cTest,kRp.hierarchy-method *Apply cTest() to all texts in kRp.hierarchy objects*

Description

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

Usage

```
## S4 method for signature 'kRp.hierarchy'
cTest(obj, mc.cores = getOption("mc.cores",
  1L), ...)
```

Arguments

obj	An object of class kRp.hierarchy.
mc.cores	The number of cores to use for parallelization, see mclapply.
...	options to pass through to cTest.

Value

An object of the same class as obj.

Examples

```
## Not run:
myCorpus <- readCorpus(
  dir=file.path(
    path.package("tm.plugin.koRpus"), "tests", "testthat", "samples", "C3S"
  ),
  hierarchy=list(
    Source=c(
      Wikipedia_alt="Wikipedia (alt)",
      Wikipedia_neu="Wikipedia (neu)"
    )
  )
)
# remove all punctuation
myCorpus <- cTest(myCorpus)

## End(Not run)
```

`docTermMatrix` *Generate a document-term matrix from a corpus object*

Description

Returns a sparse document-term matrix calculated from a given object of class `kRp.hierarchy`. You can also calculate the term frequency inverted document frequency value (tf-idf) for each term.

Usage

```
docTermMatrix(obj, terms = "token", case.sens = FALSE, tfidf = FALSE)

## S4 method for signature 'kRp.hierarchy'
docTermMatrix(obj, terms = "token",
  case.sens = FALSE, tfidf = FALSE)
```

Arguments

`obj`	An object of class `kRp.hierarchy`.
`terms`	A character string defining the `TT.res` column to be used for calculating the matrix.
`case.sens`	Logical, whether terms should be counted case sensitive.
`tfidf`	Logical, if `TRUE` calculates term frequency–inverse document frequency (tf-idf) values instead of absolute frequency.

Details

See the examples to learn how to limit the analysis to desired word classes.

Value

A sparse matrix of class `dgCMatrix`.

Examples

```
## Not run:
myCorpus <- readCorpus(
  dir=file.path(path.package("tm.plugin.koRpus"), "tests", "testthat", "samples"),
  hierarchy=list(
    Topic=c(
      C3S="C3S",
      GEMA="GEMA"
    ),
    Source=c(
      Wikipedia_alt="Wikipedia (alt)",
      Wikipedia_neu="Wikipedia (neu)"
    )
  )
```

```
)

# get the document-term frequencies in a sparse matrix
myDTMatrix <- docTermMatrix(myCorpus)

# combine with filterByClass() to, e.g.,  exclude all punctuation
myDTMatrix <- docTermMatrix(filterByClass(myCorpus))

# instead of absolute frequencies, get the tf-idf values
myDTMatrix <- docTermMatrix(
  filterByClass(myCorpus),
  tfidf=TRUE
)

## End(Not run)
```

`filterByClass,kRp.hierarchy-method`

Apply filterByClass() to all texts in kRp.hierarchy objects

Description

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

Usage

```
## S4 method for signature 'kRp.hierarchy'
filterByClass(txt,
  mc.cores = getOption("mc.cores", 1L), ...)
```

Arguments

`txt`	An object of class `kRp.hierarchy`.
`mc.cores`	The number of cores to use for parallelization, see `mclapply`.
`...`	options to pass through to `filterByClass`.

Value

An object of the same class as `txt`.

Examples

```
## Not run:
myCorpus <- readCorpus(
  dir=file.path(
    path.package("tm.plugin.koRpus"), "tests", "testthat", "samples", "C3S"
  ),
  hierarchy=list(
```

```
    Source=c(
      Wikipedia_alt="Wikipedia (alt)",
      Wikipedia_neu="Wikipedia (neu)"
    )
  )
)
# remove all punctuation
myCorpus <- filterByClass(myCorpus)

## End(Not run)
```

`freq.analysis,kRp.hierarchy-method`
Apply freq.analysis() to all texts in kRp.hierarchy 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.hierarchy'
freq.analysis(txt.file,
  mc.cores = getOption("mc.cores", 1L), ...)
```

Arguments

`txt.file`	An object of class `kRp.hierarchy`.
`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:
myCorpus <- readCorpus(
  dir=file.path(
    path.package("tm.plugin.koRpus"), "tests", "testthat", "samples", "C3S"
  ),
  hierarchy=list(
    Source=c(
      Wikipedia_alt="Wikipedia (alt)",
      Wikipedia_neu="Wikipedia (neu)"
    )
  )
```

```
)
# this will call read.corp.custom() recursively
myCorpus <- read.corp.custom(myCorpus)
myCorpus <- freq.analysis(myCorpus)

## End(Not run)
```

`hyphen,kRp.hierarchy-method`

Apply hyphen() to all texts in kRp.hierarchy objects

Description

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

Usage

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

Arguments

`words`	An object of class `kRp.hierarchy`.
`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:
myCorpus <- readCorpus(
  dir=file.path(
    path.package("tm.plugin.koRpus"), "tests", "testthat", "samples", "C3S",
      "Wikipedia_alt"
  )
)
myCorpus <- hyphen(myCorpus)

## End(Not run)
```

jumbleWords,kRp.hierarchy-method
Apply jumbleWords() to all texts in kRp.hierarchy objects

Description

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

Usage

```
## S4 method for signature 'kRp.hierarchy'
jumbleWords(words,
  mc.cores = getOption("mc.cores", 1L), ...)
```

Arguments

words	An object of class kRp.hierarchy.
mc.cores	The number of cores to use for parallelization, see mclapply.
...	options to pass through to jumbleWords.

Value

An object of the same class as words.

Examples

```
## Not run:
myCorpus <- readCorpus(
  dir=file.path(
    path.package("tm.plugin.koRpus"), "tests", "testthat", "samples", "C3S"
  ),
  hierarchy=list(
    Source=c(
      Wikipedia_alt="Wikipedia (alt)",
      Wikipedia_neu="Wikipedia (neu)"
    )
  )
)
# remove all punctuation
myCorpus <- jumbleWords(myCorpus)

## End(Not run)
```

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

Description

Objects of this class can contain full text corpora in a hierachical structure. 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 readCorpus function.

Slots

level A numeric value defining the hierachical level. Objects of this class can be nested, level=0 is the deepest (i.e., a collection of actual texts), and higher values indicate that the object represents only a category.

category A character string describing the category of this level (e.g., "source" or "topic").

id A character string naming this category level (i.e., a particular source or topic).

path A character string, full path to the directory of the current category level.

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

children If level > 0 a list of objects of class kRp.hierarchy, otherwise an empty list.

summary A summary data frame for the full corpus at the current level.

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. Only used at level=0.

tagged A list of objects of class kRp.taggedText (a class union for tagged text objects). Only used at level=0.

hyphen A list of objects of class kRp.hyphen. Only used at level=0.

TTR A list of objects of class kRp.TTR. Only used at level=0.

readability A list of objects of class kRp.readability. Only used at level=0.

freq A list with two elements, texts and corpus. At level=0, 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. At higher levels only corpus is used.

Contructor function

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

Note

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

Examples

```
## Not run:
# use readCorpus() to create objects of class kRp.hierarchy
myCorpus <- readCorpus(
  dir=file.path(path.package("tm.plugin.koRpus"), "tests", "testthat", "samples"),
  hierarchy=list(
    Topic=c(
      C3S="C3S",
      GEMA="GEMA"
    ),
    Source=c(
      Wikipedia_alt="Wikipedia (alt)",
      Wikipedia_neu="Wikipedia (neu)"
    )
  )
)

## End(Not run)
# manual creation
emptyCorpus <- kRp_hierarchy()
```

---

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.

lex.div,kRp.hierarchy-method

Apply lex.div() to all texts in kRp.hierarchy 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.hierarchy'
lex.div(txt, summary = TRUE,
  mc.cores = getOption("mc.cores", 1L), char = "", quiet = TRUE, ...)
```

Arguments

txt	An object of class kRp.hierarchy.
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:
myCorpus <- readCorpus(
  dir=file.path(path.package("tm.plugin.koRpus"), "tests", "testthat", "samples"),
  hierarchy=list(
    Topic=c(
      C3S="C3S",
      GEMA="GEMA"
    ),
    Source=c(
      Wikipedia_alt="Wikipedia (alt)",
      Wikipedia_neu="Wikipedia (neu)"
    )
  )
)
myCorpus <- lex.div(myCorpus)
```

```
## End(Not run)
```

---

query,kRp.hierarchy-method

Apply query() to all texts in kRp.hierarchy objects

Description

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

Usage

```
## S4 method for signature 'kRp.hierarchy'
query(obj, var, query, rel = "eq",
  as.df = TRUE, ignore.case = TRUE, perl = FALSE,
  regexp_var = "token", mc.cores = getOption("mc.cores", 1L))
```

Arguments

obj	An object of class kRp.hierarchy.
var	A character string naming a column in the tagged text. If set to "regexp", grepl is called on the column specified by regexp_var.
query	A character vector (for words), regular expression, or single number naming values to be matched in the variable. Can also be a vector of two numbers to query a range of frequency data, or a list of named lists for multiple queries (see "Query lists" section of query).
rel	A character string defining the relation of the queried value and desired results. Must either be "eq" (equal, the default), "gt" (greater than), "ge" (greater of equal), "lt" (less than) or "le" (less or equal). If var="word", is always interpreted as "eq"
as.df	Logical, if TRUE, returns a data frame, otherwise an object of the input class.
ignore.case	Logical, passed through to grepl if var="regexp".
perl	Logical, passed through to grepl if var="regexp".
regexp_var	A character string naming the column to query if var="regexp".
mc.cores	The number of cores to use for parallelization, see mclapply.

Value

Depending on the arguments, might include whole objects, lists, single values etc.

`read.corp.custom,kRp.hierarchy-method`
Apply read.corp.custom() to all texts in kRp.hierarchy 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.hierarchy'
read.corp.custom(corpus,
  mc.cores = getOption("mc.cores", 1L), ...)
```

Arguments

`corpus`	An object of class `kRp.hierarchy`.
`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:
myCorpus <- readCorpus(
  dir=file.path(
    path.package("tm.plugin.koRpus"), "tests", "testthat", "samples", "C3S"
  ),
  hierarchy=list(
    Source=c(
      Wikipedia_alt="Wikipedia (alt)",
      Wikipedia_neu="Wikipedia (neu)"
    )
  )
)
# this will call read.corp.custom() recursively
myCorpus <- read.corp.custom(myCorpus)

## End(Not run)
```

readability,kRp.hierarchy-method
Apply readability() to all texts in kRp.hierarchy 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.hierarchy'
readability(txt.file, summary = TRUE,
  mc.cores = getOption("mc.cores", 1L), quiet = TRUE, ...)
```

Arguments

txt.file	An object of class kRp.hierarchy.
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:
myCorpus <- readCorpus(
  dir=file.path(path.package("tm.plugin.koRpus"), "tests", "testthat", "samples"),
  hierarchy=list(
    Topic=c(
      C3S="C3S",
      GEMA="GEMA"
    ),
    Source=c(
      Wikipedia_alt="Wikipedia (alt)",
      Wikipedia_neu="Wikipedia (neu)"
    )
  )
)
myTexts <- readability(myCorpus)

## End(Not run)
```

readCorpus *Create kRp.hierarchy objects from text files or data frames*

Description

You can either read a corpus from text files (one file per text, also see the Hierarchy section below) or from TIF compliant data frames (see the Data frames section below).

Usage

```
readCorpus(dir, hierarchy = list(), lang = "kRp.env",
  tagger = "kRp.env", encoding = "", pattern = NULL,
  recursive = FALSE, ignore.case = FALSE, mode = "text",
  format = "file", mc.cores = getOption("mc.cores", 1L),
  category = "corpus", id = "", ...)
```

Arguments

dir	Either a file path to the root directory of the text corpus, or a TIF compliant data frame. If a directory path (character string), texts can be recursively ordered into subfolders named exactly as defined by `hierarchy`. If `hierarchy` is an emtpy list, all text files located in `dir` are parsed without a hierachical structure. If a data frame, also set `format="obj"` and provide hierarchy levels as additional columns, as described in the Data frames section.
hierarchy	A named list of named character vectors describing the directory hierarchy level by level. See section Hierarchy 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"`.

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. This value is passed through to simpleCorpus.
category	A character string describing the root level of the corpus.
id	A character string describing the main subject/purpose of the text corpus.
...	Additional options which are passed through to the defined tagger.

Value

An object of class kRp.hierarchy.

Hierarchy

To import a hierarchically structured text corpus you must categorize all texts in a directory structure that resembles the hierarchy. If for example you would like to import a corpus on two different topics and two differnt sources, your hierarchy has two nested levels (topic and source). The root directory dir would then need to have two subdirectories (one for each topic) which in turn must have two subdirectories (one for each source), and the actual text files are found in those.

To use this hierarchical structure in readCorpus, the hierarchy argument is used. It is a named list, where each list item represents one hierachical level (here again topic and source), and its value is a named character vector describing the actual topics and sources to be used. It is important to understand how these character vectors are treated: The names of elements must exactly match the corresponding subdirectroy name, whereas the value is a free text description. The names of the list items however describe the hierachical level and are not matched with directory names.

Data frames

In order to import a corpus from a data frame, the object must be in Text Interchange Format (TIF) as described by [1]. As a minimum, the data frame must have two character columns, doc_id and text.

You can provide additional information on hierarchical categories by using further columns, where the column name must match the category name (hierachical level). The order of those columns in the data frame is not important, as you must still fully define the hierarchical structure using the hierarchy argument. All columns you omit are ignored, but the values used in the hierarchy list and the respective columns must match, as rows with unmatched category levels will also be ignored.

Note that the special column names path and file will also be imported automatically.

References

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

Examples

```
## Not run:
# "flat" corpus, parse all texts in the given dir
myCorpus <- readCorpus(
  dir=file.path(
    path.package("tm.plugin.koRpus"), "tests", "testthat", "samples", "C3S",
      "Wikipedia_alt"
  )
)

# corpus with one category names "Source"
myCorpus <- readCorpus(
  dir=file.path(
    path.package("tm.plugin.koRpus"), "tests", "testthat", "samples", "C3S"
  ),
  hierarchy=list(
    Source=c(
      Wikipedia_alt="Wikipedia (alt)",
      Wikipedia_neu="Wikipedia (neu)"
    )
  )
)

# two hieraryhical levels, "Topic" and "Source"
myCorpus <- readCorpus(
  dir=file.path(path.package("tm.plugin.koRpus"), "tests", "testthat", "samples"),
  hierarchy=list(
    Topic=c(
      C3S="C3S",
      GEMA="GEMA"
    ),
    Source=c(
      Wikipedia_alt="Wikipedia (alt)",
      Wikipedia_neu="Wikipedia (neu)"
    )
  )
)

# if the same corpus is available as TIF compliant data frame
myCorpus <- readCorpus(
  dir=myCorpus_df,
  hierarchy=list(
    Topic=c(
      C3S="C3S",
      GEMA="GEMA"
    ),
    Source=c(
      Wikipedia_alt="Wikipedia (alt)",
      Wikipedia_neu="Wikipedia (neu)"
    )
  ),
  format="obj"
```

```
)

## End(Not run)
```

`show,kRp.hierarchy-method` *Show methods for kRp.hierarchy objects*

Description

Show methods for S4 objects of class `kRp.hierarchy`.

Usage

```
## S4 method for signature 'kRp.hierarchy'
show(object)
```

Arguments

`object`	An object of class `kRp.hierarchy`.

`simpleCorpus` *Deprecated functions and methods*

Description

These functions were used in earlier versions of the package but since replaced by `readCorpus`.

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), ...)

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

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

Arguments

dir	Use readCorpus instead.
lang	Use readCorpus instead.
tagger	Use readCorpus instead.
encoding	Use readCorpus instead.
pattern	Use readCorpus instead.
recursive	Use readCorpus instead.
ignore.case	Use readCorpus instead.
mode	Use readCorpus instead.
source	Use readCorpus instead.
topic	Use readCorpus instead.
format	Use readCorpus instead.
mc.cores	Use readCorpus instead.
...	Use readCorpus instead.
path	Use readCorpus instead.
sources	Use readCorpus instead.
paths	Use readCorpus instead.

---

summary,kRp.hierarchy-method

Apply summary() to all texts in kRp.hierarchy 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.hierarchy'
summary(object, missing = NA, ...)

corpusSummary(obj)

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

corpusSummary(obj) <- value

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

Arguments

object	An object of class kRp.hierarchy.
missing	Character string to use for missing values.
...	Used for internal processes.
obj	An object of class kRp.hierarchy.
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 objects.

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

corpusSummary is a simple method to get or set the summary slot in kRp.hierarchy objects directly.

Value

An object of the same class as object.

Examples

```
## Not run:
myCorpus <- readCorpus(
  dir=file.path(
    path.package("tm.plugin.koRpus"), "tests", "testthat", "samples", "C3S"
  ),
  hierarchy=list(
    Source=c(
      Wikipedia_alt="Wikipedia (alt)",
      Wikipedia_neu="Wikipedia (neu)"
    )
  )
)
myCorpus <- readability(myCorpus, summary=FALSE)
corpusSummary(myCorpus)
# add summaries
myCorpus <- summary(myCorpus)
corpusSummary(myCorpus)

## End(Not run)
```

`textTransform,kRp.hierarchy-method`
Apply textTransform() to all texts in kRp.hierarchy objects

Description

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

Usage

```
## S4 method for signature 'kRp.hierarchy'
textTransform(txt,
  mc.cores = getOption("mc.cores", 1L), ...)
```

Arguments

`txt`	An object of class `kRp.hierarchy`.
`mc.cores`	The number of cores to use for parallelization, see `mclapply`.
`...`	options to pass through to `textTransform`.

Value

An object of the same class as `txt`.

Examples

```
## Not run:
myCorpus <- readCorpus(
  dir=file.path(
    path.package("tm.plugin.koRpus"), "tests", "testthat", "samples", "C3S"
  ),
  hierarchy=list(
    Source=c(
      Wikipedia_alt="Wikipedia (alt)",
      Wikipedia_neu="Wikipedia (neu)"
    )
  )
)
# remove all punctuation
myCorpus <- textTransform(myCorpus, scheme="minor")

## End(Not run)
```

Index