Package 'KDViz'

December 13, 2016

Type Package

Title Knowledge Domain Visualization

Version 0.1.3

Date 2016-11-28

Author Andres Palacios <anfpalacioscl@unal.edu.co>

Maintainer Andres Palacios <anfpalacioscl@unal.edu.co>

Description
Knowledge domain visualization using mpa co-words method as the word clustering method.

License GPL (>= 2)

Encoding UTF-8

LazyData TRUE

RoxygenNote 5.0.1

Imports ade4, mpa, rvest, xml2, tm, stringr, FactoClass

NeedsCompilation no

R topics documented:

`ArticleSearch` *Article search*

Description

Search articles by specifying a list of key terms and a journal database

Usage

```
ArticleSearch(keywords, size, webSite = "ScienceDirect", addInfo = FALSE,
  infoList)
```

Arguments

`keywords`	a vector containing the key terms to search
`size`	the number of articles from which the information is extracted
`webSite`	the journal databases where the information of the articles will be searched
`addInfo`	a logical value indicating whether the info of abstract, keyword, journalTitle, journalVol and authorName should be retrieved by each article
`infoList`	a data frame of titles and URLs to skip the first step of getting the main information of each article

Value

a data frame containing the information requested in the function call

---

`CorpusFromBibData` *Corpus from an article database*

Description

Obtaining of a corpus from an article database

Usage

```
CorpusFromBibData(bibData, bibUnits, controlList, stopwords, wordsToRemove,
  replaceWords)
```

Arguments

bibData	a vector containing the key terms to search
bibUnits	a vector containing bibliometric units of analysis (e.g., Title, Abstract, Keywords, Journal, Authors, Year)
controlList	a vector of transformations that will be applied to the corpus
stopwords	a vector of stopwords to be removed from the corpus
wordsToRemove	a vector of custom words to be removed
replaceWords	a data frame of custom words and its corresponding replacement word

Value

a corpus object

DTMFromCorpus *Document-term matrix from a corpus*

Description

Obtaining of a binary document-term matrix from a corpus removing null rows

Usage

```
DTMFromCorpus(corpus, rowNames)
```

Arguments

corpus	a corpus object
rowNames	the row names of the matrix where the corpus comes from

Value

a document-term matrix

First *First element*

Description

Returns the first element of an array

Usage

```
First(x)
```

Arguments

x	a vector

Value

the first element of the incoming object

GetHREF *Get HREF attribute*

Description

Get the href attribute from a html object

Usage

```
GetHREF(nodeSet)
```

Arguments

nodeSet	a html node or node set

Value

the href attribute from the node (or the nodes)

GetRISElement *Get text from a RIS element*

Description

Get the text line from a RIS element (used internally for the GetRISList function)

Usage

```
GetRISElement(x, pattern, replacement = "", collapse = ";")
```

Arguments

x	a vector
pattern	the pattern to match the line contents
replacement	the text for replace the matched pattern (empty by default)
collapse	the symbol for collapse the resulting array of contents in the text line

Value

the text line of a RIS element

GetRISList *Get text from a RIS bibliometric unit*

Description

Get the text lines from a RIS bibliometric unit

Usage

```
GetRISList(data, pattern, replacement = "", collapse = ";")
```

Arguments

data	a list containing the RIS info of each article
pattern	the pattern to match the line contents
replacement	the text for replace the matched pattern (empty by default)
collapse	the symbol for collapse the resulting array of contents in the text line

Value

the text lines of a RIS bibliometric unit

KDSummary *Summary of word groups*

Description

Summary of word groups used to visualize knowledge domains

Usage

```
KDSummary(matriz.mpa, mpa)
```

Arguments

matriz.mpa	vector from the different words that appears in the corpus (returned value by matriz.mpa function)
mpa	a list of values resulting from the mpa function

Value

a list of objects to summarize the term clustering mpa method

KDViz *Knowledge domain visualization*

Description

Knowledge domain visualization, to perform a SCA of each partitioned document-term matrix

Usage

```
KDViz(dtmGroup, graph = FALSE, ex = 1, ey = 2, ucal = 0,
  cex.row = 0.6, cex.col = 0.7)
```

Arguments

dtmGroup	a document-term matrix
graph	a logical value, if TRUE a graph is displayed
ex	number identifying the factor to be used as horizontal axis (1 by default)
ey	number identifying the factor to be used as vertical axis (1 by default)
ucal	quality representation threshold (percentage) in the plane (0 by default)
cex.row	scale for row points and row labels (0.6 by default)
cex.col	scale for column points and column labels (0.7 by default)

Value

the SCA from a document-term matrix group

Examples

```
## Not run:
rm(list = ls())
library("KDViz")

risFile <- system.file("ScienceDirectRIS.ris", package = "KDViz") # Original data

myData <- ReadRIS(risFile, "bibDataRIS", saveRda = TRUE, saveCSV = FALSE) # RIS file to data object

bibData <- system.file("bibData.Rda", package = "KDViz")
load(bibData)


# Create a corpus from the bib data
corpus <- CorpusFromBibData(bibData = bibData, bibUnits = c("Keywords"),
   controlList = "", stopwords = "", wordsToRemove = "")

dtm <- DTMFromCorpus(corpus, row.names(bibData)) # Create a doc-term matrix from the corpus
dim(dtm)

# A first review of the raw corpus

bibUnits <- c("Keywords") # Selection of bibliometric units to analyze
controlList <- c("stripWhitespace", "removeNumbers") # List of tm process to perform
```

```
# Decide which stopwords are going to be used (a file or FALSE if they are not required)
stopwords <- FALSE
#stopwords <- system.file("stopwords_en.txt", package = "KDViz") # Optional
wordsToRemove <- c("nanotechnology") # List of custom words to remove

# Custom dictionary to replace some selected words
replaceWords <- system.file("keywordReplace.txt", package = "KDViz")

# Corpus from bibdata with and a control list to perform the entire tm process
corpus <- CorpusFromBibData(bibData = bibData, bibUnits = bibUnits,
   controlList = controlList, stopwords = stopwords,
   wordsToRemove = wordsToRemove, replaceWords = replaceWords)

termFreqTable <- TermFrequency(corpus) # See the frequency of terms in the corpus
head(termFreqTable, 98)

# Search for words containing the term in 'word' parameter
TermFreqByWord(termFreqTable = termFreqTable, word = "reduction")

# An optional function (contained yet in the previous process) to
#  replace other words after getting a corpus
#corpus <- ReplaceByList(corpus = corpus, wordsFile = replaceWords)

termFreqTable <- TermFrequency(corpus) # See the frequency of terms in the current corpus
head(termFreqTable, 100)

dtm <- DTMFromCorpus(corpus, row.names(bibData)) # Create a doc-term matrix from the corpus
dim(dtm)
rownames(dtm)

termFreq <- TermFrequency(dtm) # See the frequency of terms in the doc-term matrix
head(termFreq, 100)

mpaWords <- matriz.mpa.corpus(corpus, fmin = 5, cmin = 1) # mpa matrices from a corpus object
mpaWords$Palabras

# mpa method from the calculated objects in 'mpaWords'
classes <- mpa::mpa(mpaWords$Matriza, 10, mpaWords$Palabras)
classes

kdSummary <- KDSummary(matriz.mpa = mpaWords, mpa = classes) # a quick summary of the mpa process

mpa::plotmpa(3, mpaWords$Matriza, classes) # Plot the network of selected class

# Extract a partition of the original 'dtm' matrix depending on the class that you want
WordGroupDTM(dtm, wordClasses = kdSummary$wordClasses, class = 7, graph = TRUE)

group1 <- WordGroupDTM(dtm, wordClasses = kdSummary$wordClasses, class = 1, graph = TRUE)
group2 <- WordGroupDTM(dtm, wordClasses = kdSummary$wordClasses, class = 2, graph = TRUE)
group3 <- WordGroupDTM(dtm, wordClasses = kdSummary$wordClasses, class = 3, graph = TRUE)
group4 <- WordGroupDTM(dtm, wordClasses = kdSummary$wordClasses, class = 4, graph = TRUE)
group5 <- WordGroupDTM(dtm, wordClasses = kdSummary$wordClasses, class = 5, graph = TRUE)
group6 <- WordGroupDTM(dtm, wordClasses = kdSummary$wordClasses, class = 6, graph = TRUE)
group7 <- WordGroupDTM(dtm, wordClasses = kdSummary$wordClasses, class = 7, graph = TRUE)
group8 <- WordGroupDTM(dtm, wordClasses = kdSummary$wordClasses, class = 8, graph = TRUE)
group9 <- WordGroupDTM(dtm, wordClasses = kdSummary$wordClasses, class = 9, graph = TRUE)
group10 <- WordGroupDTM(dtm, wordClasses = kdSummary$wordClasses, class = 10, graph = TRUE)
```

```
plot(group1$coaGroup, ucal = 0, cex.col = 0.8, cex.row = 0.5)

LoadArticle(bibData, "A625") # Load the info of an article (it will open the URL by default)

## End(Not run)
```

`Last` *Last element*

Description

Returns the last element of an array

Usage

```
Last(x)
```

Arguments

`x`	a vector

Value

the last element of the incoming object

`leer.mpa.corpus` *Reads a corpus and passes it to mpa format*

Description

Returns the content of a corpus object to use mpa package methods

Usage

```
leer.mpa.corpus(corpus)
```

Arguments

`corpus`	a corpus object

Value

a vector containing the term list per document

`LoadArticle` *Load the info of an article*

Description

Load the info of an article and if wanted, shows the webpage of it

Usage

```
LoadArticle(articleData, articleName, browser = TRUE)
```

Arguments

`articleData`	a data frame containing the info (Title, Abstract, Keywords, URL, ...) of an article
`articleName`	the name of an article (rowname from the articleData)
`browser`	a logical value. If TRUE, the article URL is opened in a browser window

Value

the info from the article and the website where it is available

`matriz.mpa.corpus` *Calculation of co-occurrences matrix and matrix associations from a corpus*

Description

Similar to the mpa package, it calculates the co-occurrences matrix and the matrix associations from the resulting object of the leer.mpa.corpus function

Usage

```
matriz.mpa.corpus(corpus, fmin = 3, cmin = 3)
```

Arguments

`corpus`	a corpus object
`fmin`	minimal appearance frequency of key words inside the corpus
`cmin`	minimal co-occurrence between words

Value

a list that contains the associations and the co-ocurrence matrices, the vector of words and a lexical table (obtained from the original matriz.mpa function0)

ReadRIS *Read a RIS file*

Description

Read the entire info from a RIS file

Usage

```
ReadRIS(risFile, fileName, saveRda = FALSE, saveCSV = FALSE)
```

Arguments

risFile	a file of RIS extension
fileName	a character giving the name of the resulting file to export
saveRda	a logical value that indicates whether the file should be saved or not in a Rda file
saveCSV	a logical value that indicates whether the file should be saved or not in a csv file

Value

a data frame of bibliometric units by each article

ReplaceByList *Replace a list of words by a pair list*

Description

A process similar to lemmatization with a custom dictionary file in the form of a data frame of custom words and its corresponding replacement word

Usage

```
ReplaceByList(corpus, wordsFile)
```

Arguments

corpus	a corpus object
wordsFile	a file with custom words to be replaced (first column with the replacement words, second column with the original words; separated by tabulation)

Value

a corpus with replaced words

Author(s)

Camila Góngora <mcgongoraa@unal.edu.co>, Andrés Palacios <anfpalacioscl@unal.edu.co>

ScienceDirectArticles *Article search from ScienceDirect*

Description

Search articles from the ScienceDirect database by specifying a list of key terms

Usage

```
ScienceDirectArticles(keywords, size, addInfo = FALSE, infoList)
```

Arguments

keywords	a vector containing the key terms to search
size	the number of articles from which the information is extracted
addInfo	a logical value indicating whether the info of abstract, keyword, journalTitle, journalVol and authorName should be retrieved by each article
infoList	a data frame of titles and URLs to skip the first step of getting the main information of each article

Value

a data frame containing the information requested in the function call

SparseRate *Sparse rate to remove a proportion of terms*

Description

Returns the threshold to decide the sparse rate to use depending on the minimum allowed frequency of the terms in the document term matrix

Usage

```
SparseRate(termFreq, dtm)
```

Arguments

termFreq	a minimum allowed frecuency to define the sparse of the terms
dtm	document-term matrix

Value

sparse percentage of non empty elements

TermFreqByWord *Term frequency by a specific words*

Description

Returns the frequency of the terms containing a specific word

Usage

```
TermFreqByWord(termFreqTable, word)
```

Arguments

termFreqTable	a TermFrequency table
word	a custom word to be matched

Value

a list of terms and their respective frecuencies

TermFrequency *Term frequency*

Description

A list of terms and their absolute frequencies in a corpus or a document-term matrix

Usage

```
TermFrequency(x)
```

Arguments

x	a corpus or a document-term matrix object

Value

a list of terms and their respective frecuencies

TextFromHtml *Text from html*

Description

Extracts the text attribute from an html node depending on the type and the desired quantity of selectors

Usage

```
TextFromHtml(html, selector, names, sep = " ")
```

Arguments

html	an html node
selector	the type of html selector ("class" or "id")
names	the possible names of the selector you are looking for (the first not null is selected)
sep	a separator, the symbol to replace the html text spaces between words (" " by default)

Value

the plain text extracted from the html element

WordGroupDTM *Document-term matrix and SCA by word group*

Description

A portion of an entire document-term matrix depending on the class found using the kdSummary function

Usage

```
WordGroupDTM(class, dtm, wordClasses, graph = FALSE)
```

Arguments

class	the number of the class to be partitioned
dtm	a document-term matrix
wordClasses	the value resulting from performing a KDSummary
graph	a logical value. If TRUE, the knowledge domain map of the corresponding class is plotted

Value

the doc-term matrix and the SCA object from the document and word group

`WordsInCorpus` *See words inside a corpus*

Description

A function to return all words found in a corpus

Usage

```
WordsInCorpus(corpus)
```

Arguments

`corpus`	a corpus object

Value

a vector of words

Index