Package ‘tmcn’
August 8, 2019

License LGPL
Title A Text Mining Toolkit for Chinese
Type Package
LazyLoad yes
Author Jian Li
Maintainer Jian Li <rweibo@sina.com>
Description A Text mining toolkit for Chinese, which includes facilities for
Chinese string processing, Chinese NLP supporting, encoding detecting and
converting. Moreover, it provides some functions to support ‘tm’ package
in Chinese.
Version 0.2-13
Date 2019-08-04
Depends R (>= 3.0.0), utils
Suggests tm
RoxygenNote 6.1.1
NeedsCompilation yes
Repository CRAN
Date/Publication 2019-08-08 04:40:02 UTC

R topics documented:

catUTF8 ................................................. 2
createDTM ............................................. 3
createWordFreq ................................. 4
GBK .................................................. 4
getCharset ........................................... 5
isBIG5 ................................................. 6
isGB18030 .......................................... 6
isGB2312 ............................................. 7
isGBK ............................................... 8
isUTF8 .............................................. 8
Print the UTF-8 codes of a string.

\textbf{Usage}

\texttt{catUTF8(string, file = "")}

\textbf{Arguments}

- \texttt{string} A character vector.
- \texttt{file} A \texttt{connection}, or a character string naming the file to print to. If "" (the default), \texttt{cat} prints to the standard output connection, the console unless redirected by \texttt{sink}.

\textbf{Value}

No results.

\textbf{Author(s)}

Jian Li \texttt{<rweibo@sina.com>}

\textbf{Examples}

\texttt{catUTF8("hello")}
createDTM

Create a Chinese term-document matrix or a document-term matrix.

Description

Create a Chinese term-document matrix or a document-term matrix.

Usage

createDTM(string, language = c("zh", "en"), tokenize = NULL, removePunctuation = TRUE, removeNumbers = TRUE, removeStopwords = TRUE)
createTDM(string, language = c("zh", "en"), tokenize = NULL, removePunctuation = TRUE, removeNumbers = TRUE, removeStopwords = TRUE)

Arguments

string A character vector.
language The language type, 'zh' means Chinese.
tokenize A tokenizers function.
removePunctuation Whether to remove the punctuations.
removeNumbers Whether to remove the numbers.
removeStopwords Whether to remove the stop words.

Details

Package "tm" is required.

Value

An object of class TermDocumentMatrix or class DocumentTermMatrix.

Author(s)

Jian Li <<rweibo@sina.com>>
### createWordFreq

Create a word frequency data.frame.

**Description**

Create a word frequency data.frame.

**Usage**

```r
createWordFreq(obj, onlyCN = TRUE, nosymbol = TRUE, stopwords = NULL, useStopDic = FALSE)
```

**Arguments**

- `obj` A character vector or `DocumentTermMatrix` to calculate words frequency.
- `onlyCN` Whether to keep only Chinese words.
- `nosymbol` Whether to keep symbols.
- `stopwords` A character vector of stop words.
- `useStopDic` Whether to use the default stop words.

**Value**

A data.frame.

**Author(s)**

Jian Li <<rweibo@sina.com>>

**Examples**

```r
createWordFreq(c("a", "a", "b", "c"), onlyCN = FALSE, nosymbol = TRUE, useStopDic = FALSE)
```

---

### GBK

**GBK character set**

**Description**

GBK character set including some useful information.

**Usage**

data(GBK)
**Format**

A data frame with 8 columns.

- **GBK** Chinese characters in UTF-8.
- **py0** Unique Pinyin of each character.
- **py** Pinyin string of each character.
- **Radical** In Chinese, it means 'Bu Shou'.
- **Stroke_Num_Radical** In Chinese, it means the number of 'Bi Hua'.
- **Stroke_Order** In Chinese, it means 'Bi Shun'.
- **Structure** In Chinese, it means 'Zi Ti Jie Gou'.
- **Freq** Frequency of the character in Sogou news corpus from all sites between June and July 2012.

**Author(s)**

Jian Li <<rweibo@sina.com>>

---

**getCharset**

*Get the current encoding of the locale.*

**Description**

Get the current encoding of the locale.

**Usage**

getCharset()

**Value**

Character of encoding.

**Author(s)**

Jian Li <<rweibo@sina.com>>

**Examples**

getCharset()
isBIG5

Indicate whether the encoding of input string is BIG5.

Description

Indicate whether the encoding of input string is BIG5.

Usage

isBIG5(string, combine = FALSE)

Arguments

<table>
<thead>
<tr>
<th>string</th>
<th>A character vector.</th>
</tr>
</thead>
<tbody>
<tr>
<td>combine</td>
<td>Whether to combine all the strings.</td>
</tr>
</tbody>
</table>

Value

Logical value.

Author(s)

Jian Li <<rweibo@sina.com>>

Examples

isBIG5("hello")

isGB18030

Indicate whether the encoding of input string is GB18030.

Description

Indicate whether the encoding of input string is GB18030.

Usage

isGB18030(string, combine = FALSE)

Arguments

<table>
<thead>
<tr>
<th>string</th>
<th>A character vector.</th>
</tr>
</thead>
<tbody>
<tr>
<td>combine</td>
<td>Whether to combine all the strings.</td>
</tr>
</tbody>
</table>
isGB2312

Value
Logical value.

Author(s)
Jian Li <<rweibo@sina.com>>

Examples
isGB18030(“hello”)

isGB2312

Indicate whether the encoding of input string is GB2312.

Description
Indicate whether the encoding of input string is GB2312.

Usage
isGB2312(string, combine = FALSE)

Arguments
string A character vector.
combine Whether to combine all the strings.

Value
Logical value.

Author(s)
Jian Li <<rweibo@sina.com>>

Examples
isGB2312(“hello”)
isGBK

*Indicate whether the encoding of input string is GBK.*

**Description**

Indicate whether the encoding of input string is GBK.

**Usage**

`isGBK(string, combine = FALSE)`

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>A character vector.</td>
</tr>
<tr>
<td>combine</td>
<td>Whether to combine all the strings.</td>
</tr>
</tbody>
</table>

**Value**

Logical value.

**Author(s)**

Jian Li <<rweibo@sina.com>>

**Examples**

`isGBK("hello")`

---

isUTF8

*Indicate whether the encoding of input string is UTF-8.*

**Description**

Indicate whether the encoding of input string is UTF-8.

**Usage**

`isUTF8(string, combine = FALSE)`

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>A character vector.</td>
</tr>
<tr>
<td>combine</td>
<td>Whether to combine all the strings.</td>
</tr>
</tbody>
</table>
left

Value

Logical value.

Author(s)

Jian Li <rweibo@sina.com>

Examples

isUTF8("hello")

left

Extract the left or right substrings in a character vector.

Description

Extract the left or right substrings in a character vector.

Usage

left(string, n)
right(string, n)

Arguments

string A character vector.
n How many characters.

Value

A character vector.

Author(s)

Jian Li <rweibo@sina.com>

Examples

left("hello", 3)
NTUSD  National Taiwan University Semantic Dictionary

Description
National Taiwan University Semantic Dictionary.

Usage
data(NTUSD)

Format
A list with 4 components.
  positive_chs  Positive words in simplified Chinese
  negative_chs  Negative words in simplified Chinese
  positive_cht  Positive words in traditional Chinese
  negative_cht  Negative words in traditional Chinese

References
http://nlg.csie.ntu.edu.tw

revUTF8  Revert UTF-8 string to Chinese character.

Description
Revert UTF-8 string to Chinese character.

Usage
revUTF8(string, utype = "R")

Arguments
  string  A character vector.
  utype  UTF-8 string type, the default is R type, such as "<U+XXXX>".

Value
A character vector.

Author(s)
  Jian Li <<rweibo@sina.com>>
setchs

**Description**

Set locale to Simplified Chinese/Traditional Chinese/UK.

**Usage**

```r
setchs(rev = FALSE)
setcht(rev = FALSE)
setuk(rev = FALSE)
```

**Arguments**

- `rev` : Whether to set the locale back.

**Value**

No results.

**Author(s)**

Jian Li <rweibo@sina.com>

**Examples**

```r
setchs()
setchs(rev = TRUE)
```

---

SIMTRA

**Description**

Dictionary of simplified and traditional Chinese.

**Usage**

```r
data(SIMTRA)
```

**Format**

A data frame with 2 columns.

<table>
<thead>
<tr>
<th>Sim</th>
<th>Tra</th>
</tr>
</thead>
<tbody>
<tr>
<td>a simplified Chinese string.</td>
<td>a traditional Chinese string.</td>
</tr>
</tbody>
</table>
### SPORT

**Sport news.**

**Description**

Sport news.

**Usage**

`data(SPORT)`

**Format**

A data frame with 6 columns.

- **id** ID of the news.
- **time** Time of the news.
- **title** Title of the news.
- **class** Class of the news, 'B' means Basketball, 'F' means Football.
- **abstract** Abstract of the news.
- **content** Content of the news.

### STOPWORDS

**Dictionary of Chinese stop words**

**Description**

Dictionary of Chinese stop words.

**Usage**

`data(STOPWORDS)`

**Format**

A data frame with 1 column.

- **word** a string vector of the stop words.
stopwordsCN

Return Chinese stop words.

Description

Return Chinese stop words.

Usage

stopwordsCN(stopwords = NULL, useStopDic = TRUE)

Arguments

stopwords  A character vector of stop words.
useStopDic  Whether to use the default stop words.

Value

A vector of stop words.

Author(s)

Jian Li <rweibo@sina.com>

Examples

stopwordsCN("yes", useStopDic = FALSE)

strcap

Mixed case capitalizing.

Description

To capitalize every first letter of a word.

Usage

strcap(string, strict = FALSE)

Arguments

string  A character vector.
strict  Whether strict.
Value

A character vector with the first letter of each word capitalized.

Author(s)

Jian Li <<rweibo@sina.com>>

Examples

strcap("the quick red fox jumps over the lazy brown dog")

---

`strextract`  
*Extract matched substrings by regular expression.*

Description

Extract matched substrings by regular expression.

Usage

`strextract(string, pattern, invert = FALSE, ignore.case = FALSE, perl = FALSE, useBytes = FALSE)`

Arguments

- `string`: A character vector.
- `pattern`: A character string containing a regular expression to be matched in the given character vector.
- `invert`: A logical value: if TRUE, extract the non-matched substrings.
- `ignore.case`: If FALSE, the pattern matching is case sensitive and if TRUE, case is ignored during matching.
- `perl`: A logical value. Should perl-compatible regexps be used?
- `useBytes`: A logical value. If TRUE the matching is done byte-by-byte rather than character-by-character.

Value

A character vector with the matched or non-matched substrings.

Author(s)

Jian Li <<rweibo@sina.com>>
Examples

```r
txt1 <- c("(x1)a(aa2)a ", " bb(bb)")
strextract(txt1, "\([^)\]*\)\")

txt2 <- c(" Ben Franklin and Jefferson Davis", "\tMillard Fillmore")
strextract(txt2, "(?<first>[:upper:][[:lower:]]+)", perl = TRUE)
```

---

**strpad**

Pad a string to a specified length with a padding character.

Description

Pad a string to a specified length with a padding character.

Usage

```r
strpad(string, width = 0, side = c("left", "right", "both"),
pad = " ")
```

Arguments

- `string`: A character vector.
- `width`: The number of characters of the string after padding.
- `side`: Which side to pad.
- `pad`: The padding character.

Value

A character vector after padding.

Author(s)

Jian Li &lt;rweibo@sina.com&gt;

Examples

```r
strpad(1:5, width = 4, pad = "0")
```
strstrip  
Trim space of a string.

Description
Trim space of a string.

Usage
strstrip(string, side = c("both", "left", "right"))

Arguments
- string: A character vector.
- side: Which side of the string to be trimmed, 'both', 'left' or 'right'.

Value
Trimed vector.

Author(s)
Jian Li <rweibo@sina.com>

Examples
strstrip(c("\taaaa ", " bbbb "))

toPinyin  
Convert a chinese text to pinyin format.

Description
Convert a chinese text to pinyin format.

Usage
toPinyin(string, capitalize = FALSE)

Arguments
- string: A character vector.
- capitalize: Whether to capitalize the first letter of each word.
Value

A character vector in pinyin format.

Author(s)

Jian Li <<rweibo@sina.com>>

Examples

toPinyin("the quick red fox jumps over the lazy brown dog")

toTrad

Convert a Chinese text from simplified to traditional characters and vice versa.

Description

Convert a chinese text from simplified to traditional characters and vice versa.

Usage

toTrad(string, rev = FALSE)

Arguments

string A Chinese string vector.
rev Reverse. TRUE means traditional to simplified. Default is FALSE.

Value

Converted vectors.

Author(s)

Jian Li <<rweibo@sina.com>>

Examples

toTrad("hello")
Convert encoding of Chinese string to UTF-8.

**Description**

Convert encoding of Chinese string to UTF-8.

**Usage**

`toUTF8(cnstring)`

**Arguments**

- `cnstring`: A Chinese string vector.

**Value**

 Converted vectors.

**Author(s)**

Jian Li (<rweibo@sina.com>)

**Examples**

`toUTF8("hello")`
Index

*Topic NLP
createDTM, 3

*Topic datasets
GBK, 4
NTUSD, 10
SIMTRA, 11
SPORT, 12
STOPWORDS, 12

*Topic string
strcap, 13
strextract, 14
strpad, 15
strstrip, 16
toPinyin, 16
catUTF8, 2
connection, 2
createDTM, 3
createTDM(createDTM), 3
createWordFreq, 4

GBK, 4
getCharset, 5

isBIG5, 6
isGB18030, 6
isGB2312, 7
isGBK, 8
isUTF8, 8

left, 9

NTUSD, 10

revUTF8, 10
right(left), 9

setchs, 11
setcht(setchs), 11
setuk(setchs), 11
SIMTRA, 11

sink, 2
SPORT, 12
STOPWORDS, 12
stopwordsCN, 13
strcap, 13
strextract, 14
strpad, 15
strstrip, 16
toPinyin, 16
toTrad, 17
toUTF8, 18