Package ‘arabicStemR’

October 12, 2022

Type Package
Title Arabic Stemmer for Text Analysis
Version 1.3
Date 2022-07-14
Author Rich Nielsen
Maintainer Rich Nielsen <rnielsen@mit.edu>
Description Allows users to stem Arabic texts for text analysis.
License GPL (>= 2)
NeedsCompilation no
Repository CRAN
Date/Publication 2022-07-18 08:20:09 UTC

R topics documented:

arabicStemR-package ........................................ 2
cleanChars .................................................. 3
cleanLatinChars ........................................... 3
doStemming .................................................. 4
fixAlifs ..................................................... 5
removeArabicNumbers ..................................... 6
removeDiacritics .......................................... 6
removeEnglishNumbers ................................. 7
removeFarsiNumbers ..................................... 8
removeNewlineChars .................................... 8
removeNumbers .......................................... 9
removePrefixes .......................................... 10
removePunctuation .................................... 11
removeStopWords ....................................... 12
removeSuffixes .......................................... 13
reverse.transliterate .................................. 14
stem ....................................................... 15
stemArabic ............................................... 17
transliterate ............................................ 18
A package for stemming Arabic for text analysis.

Description

This package is a stemmer for texts in Arabic (Modern Standard). The stemmer is loosely based on the light 10 stemmer, but with a number of modifications.

Details

Use the stemArabic function.

Author(s)

Maintainer: Rich Nielsen <rnielsen@mit.edu>

See Also

stemArabic

Examples

```r
## generate some text in Arabic
x <- "\u628\u633\u645 \u0627\u0644\u0647 \\u0627\u0644\u0631\u062D\u0645\u0646 \\u0627\u0644\u0631\u062D\u064A\u0645"

## stem and transliterate
stemArabic(x)

## stem while not stemming certain words
stem(x, dontStemTheseWords = c("alr7mn"))

## stem and return the stemlist
out <- stemArabic(x, returnStemList = TRUE)
out$text
out$stemlist
```
**cleanChars**

*Clean all characters that are not Latin or Arabic*

**Description**

Cleans any characters in string that are not in either the Latin unicode range or in the Arabic alphabet.

**Usage**

`cleanChars(texts)`

**Arguments**

- **texts**  
  A string from which characters which are not Latin or Arabic should be removed.

**Value**

`cleanChars` returns a string with only Latin and Arabic characters.

**Author(s)**

Rich Nielsen

**Examples**

```r
## Create string with Arabic, latin, and Hebrew characters
x <- '\u0627\u0647\u0644\u0627 \u0648\u0633\u0647\u0644\u0627 Hello \u05d0'

## Remove characters from string that are not Arabic or latin
cleanChars(x)
```

---

**cleanLatinChars**

*Clean Latin characters*

**Description**

Cleans Latin characters from a string.

**Usage**

`cleanLatinChars(texts)`

**Examples**

```r
## Create string with Arabic, latin, and Hebrew characters
x <- '\u0627\u0647\u0644\u0627 \u0648\u0633\u0647\u0644\u0627 Hello \u05d0'

## Remove characters from string that are not Arabic or latin
cleanLatinChars(x)
```
Arguments
  texts          A string from which Latin characters should be removed.

Value
  cleanLatinChars returns a string with Latin characters removed.

Author(s)
  Rich Nielsen

Examples
  ## Create string with Arabic and latin characters
  x <- '\u0627\u0647\u0644\u0627 \u0648\u0633\u0647\u0644\u0627 Hello'

  ## Rewmove latin characters from string
  cleanLatinChars(x)

---

doStemming

Removes Arabic prefixes and suffixes

Description
  Removes prefixes and suffixes, and can return a list matching the words to stemmed words. Does
  not stem different forms of Allah.

Usage
  doStemming(texts, dontstem = c(\u0627\u0647\u0644\u0627,\u0644\u0647))

Arguments
  texts          The original texts.
  dontstem       By default, does not stem different forms of Allah

Value
  doStemming returns a named list with the following elements:
    text          The stemmed text
    stemmedWords   A list matching the words and the stemmed words.

Author(s)
  Rich Nielsen
Examples

## Create string with Arabic characters

```r
x <- '\u0627\u0644\u0644\u063a\u0629 \u0627\u0644\u0639\u0631\u0628\u064a\u0629 \\
\u062c\u0645\u0644\u0629 \u062c\u062f\u0627'
```

## Remove prefixes and suffixes

```r
y <- doStemming(x)
y$text
y$stemmedWords
```

---

`fixAlifs`  
*Standardize different hamzas on alif seats*

Description

Standardize different hamzas on alif seats in a string.

Usage

`fixAlifs(texts)`

Arguments

`texts`  
A string from which different alifs are standardized.

Value

`fixAlifs` returns a string with standardized alifs.

Author(s)

Rich Nielsen

Examples

```r
## Create string with Arabic characters
x <- '\u0622 \u0623 \u0675'
```

```r
## Standardize Alifs
fixAlifs(x)
```
removeArabicNumbers  Remove Arabic numbers

Description

Removes Arabic numerals from a string.

Usage

removeArabicNumbers(texts)

Arguments

texts  A string from which Arabic numerals should be removed.

Value

removeArabicNumbers returns a string with Arabic numerals removed.

Author(s)

Rich Nielsen

Examples

## Create string with Arabic characters and numbers
x <- '\u0627\u0647\u0644\u0627 \u0661\u0662\u0663'

## Remove Arabic numbers
removeArabicNumbers(x)

removeDiacritics  Remove Arabic diacritics

Description

Removes diacritics from Arabic unicode text.

Usage

removeDiacritics(texts)

Arguments

texts  A string from which Arabic diacritics should be removed.
Value
removeDiacritics returns a string with Arabic diacritics removed.

Author(s)
Rich Nielsen

Examples
## Create string with Arabic characters and diacritics
x<-
'\u0627\u0647\u0644\u0627\u064b \u0648\u0633\u0647\u0644\u0627\u064b'

## Remove diacritics
removeDiacritics(x)

removeEnglishNumbers  Remove English numbers

Description
Removes Arabic numerals from a string.

Usage
removeEnglishNumbers(texts)

Arguments
texts  A string from which English numerals should be removed.

Value
removeEnglishNumbers returns a string with English numerals removed.

Author(s)
Rich Nielsen

Examples
## Create string with Arabic characters and English number
x <- '\u0627\u0647\u0644\u0627 \u0648\u0633\u0647\u0644\u0627 123'

## Remove English Numbers
removeNumbers(x)
**removeFarsiNumbers**  
*Remove Farsi numbers*

**Description**  
Removes Farsi numerals from a string.

**Usage**  
\[\text{removeFarsiNumbers}(\text{texts})\]

**Arguments**  
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>texts</td>
<td>A string from which Farsi numerals should be removed.</td>
</tr>
</tbody>
</table>

**Value**  
\[\text{removeFarsiNumbers} \text{ returns a string with Arabic numerals removed}.\]

**Author(s)**  
Rich Nielsen

**Examples**

```r
## Create string with Arabic characters and numbers
x <- '\u0627\u0647\u0644\u0627 \u0648\u0633\u0647\u0644\u0627 \u06f1\u06f2\u06f3\u06f4\u06f5'

## Remove Farsi numbers
removeFarsiNumbers(x)
```

---

**removeNewlineChars**  
*Remove new line characters*

**Description**  
Removes new line characters from a string.

**Usage**  
\[\text{removeNewlineChars}(\text{texts})\]

**Arguments**  
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>texts</td>
<td>A string from which new line characters should be removed.</td>
</tr>
</tbody>
</table>
removeNumbers

Value

removeNewlineChars returns a string with new line characters removed.

Author(s)

Rich Nielsen

Examples

## Create string with Arabic characters

```r
x <- '\u0627\u0647\u0644\u0627 \u0648\u0633\u0647\u0644\u0627
 \u0627\u0647\u0644\u0627 \u0648\u0633\u0647\u0644\u0627'
```

## Remove newline characters (gets rid of \n\r\t\f\v)

```r
removeNewlineChars(x)
```

---

removeNumbers  
Remove English, Arabic, and Farsi numerals.

Description

Removes English, Arabic, and Farsi numerals from a string.

Usage

```r
removeNumbers(texts)
```

Arguments

- **texts**: A string from which English, Arabic, and Farsi numerals should be removed.

Value

removeNumbers returns a string with English, Arabic, and Farsi numerals removed.

Author(s)

Rich Nielsen
### Examples

```r
## Create string with Arabic characters and number
x <- '\u0627\u0647\u0644\u0627 \u0648\u0633\u0647\u0644\u0627 123 \u0661\u0662\u0663'

## Remove Numbers
removeNumbers(x)
```

---

**removePrefixes**  
*Remove Arabic prefixes*

#### Description

Removes some Arabic prefixes from a unicode string. The prefixes are: "waw", "alif-lam", "waw-alif-lam", "ba-alif-lam", "kaf-alif-lam", "fa-alif-lam", and "lam-lam." Prefixes are removed from a word (as defined by spaces) only if the remaining stem would not be too short.

#### Usage

```r
removePrefixes(texts, x1 = 4, x2 = 4, x3 = 5, x4 = 5, x5 = 5, x6 = 5, x7 = 4,  
dontstem = c('\u0627\u0644\u0647','\u0644\u0644\u0647'))
```

#### Arguments

- **texts**: An Arabic-language string in unicode  
- **x1**: The number of letters that must be in a word for the function to remove the prefix "waw".  
- **x2**: The number of letters that must be in a word for the function to remove the prefix "alif-lam".  
- **x3**: The number of letters that must be in a word for the function to remove the prefix "waw-alif-lam".  
- **x4**: The number of letters that must be in a word for the function to remove the prefix "ba-alif-lam".  
- **x5**: The number of letters that must be in a word for the function to remove the prefix "kaf-alif-lam".  
- **x6**: The number of letters that must be in a word for the function to remove the prefix "fa-alif-lam".  
- **x7**: The number of letters that must be in a word for the function to remove the prefix "lam-lam".  
- **dontstem**: Words that should not be stemmed (entered in unicode).
Value

Returns a string with Arabic prefixes removed.

Author(s)

Rich Nielsen

Examples

```r
## Create string with Arabic characters
x <- 'اللغة العية جميلة جدا'

# Remove Prefixes
removePrefixes(x)
```

Description

Removes punctuation from a string, including some specialized Arabic characters.

Usage

```r
removePunctuation(texts)
```

Arguments

- **texts**  
  A string from which punctuation should be removed.

Value

Returns a string with punctuation removed.

Author(s)

Rich Nielsen
## removeStopWords

### Examples

```r
## Create string with Arabic characters and punctuation
x <- '\u0627\u0647\u0644\u0627 \u0648\u0633\u0647\u0644\u0627!!!?'

## Remove punctuation
removePunctuation(x)
```

---

**removeStopWords**  
*Remove Arabic stopwords.*

### Description

Defines a list of Arabic-language stopwords and removes them from a string.

### Usage

```r
removeStopWords(texts, defaultStopwordList=TRUE, customStopwordList=NULL)
```

### Arguments

- `texts`  
  A string from which Arabic stopwords should be removed.

- `defaultStopwordList`  
  If TRUE, use the default stopword list of words to be removed. If FALSE, do not use the default stopword list. Default is TRUE.

- `customStopwordList`  
  Optional user-specified stopword list of words to be removed, supplied as a vector of strings in either Arabic UTF-8 or Latin characters following the stemmer's transliteration scheme (words without Arabic UTF-8 characters are processed with reverse.transliterate()). Default is NULL.

### Value

Returns a string with Arabic stopwords removed.

### Author(s)

Rich Nielsen
Examples

```r
## Create string with Arabic characters
x <- '\u0627\u0647\u0644\u0627 \u0648\u0633\u0647\u0644\u0627 \
\u064a\u0627 \u0635\u062f\u064a\u0642\u064a'

## Remove stop words
removeStopWords(x)$text

## Not run
## To see the full list of stop words
removeStopWords(x)$arabicStopwordList
```

Description

Removes some Arabic suffixes from a unicode string. The suffixes (in order of removal) are: "ha-alif", "alif-nun", "alif-ta", "waw-nun", "yah-nun", "yah-heh", "yah-ta marbutta", "heh", "ta marbutta", and "yah." Suffixes are removed from a word (as defined by spaces) only if the remaining stem would not be too short. Only one suffix is removed from each word.

Usage

```r
removeSuffixes(texts, x1 = 4, x2 = 4, x3 = 4, x4 = 4,
               x5 = 4, x6 = 4, x7 = 4, x8 = 3, x9 = 3, x10 = 3,
               dontstem = c('\u0627\u0644\u0644\u0647', '\u0644\u0644\u0647'))
```

Arguments

- **texts**: An Arabic-language string in unicode.
- **x1**: The number of letters that must be in a word for the function to remove the suffix "ha-alif".
- **x2**: The number of letters that must be in a word for the function to remove the suffix "alif-nun".
- **x3**: The number of letters that must be in a word for the function to remove the suffix "alif-ta".
- **x4**: The number of letters that must be in a word for the function to remove the suffix "waw-nun".
- **x5**: The number of letters that must be in a word for the function to remove the suffix "yah-nun".
- **x6**: The number of letters that must be in a word for the function to remove the suffix "yah-ta marbutta".
- **x7**: The number of letters that must be in a word for the function to remove the suffix "heh".
- **x8**: The number of letters that must be in a word for the function to remove the suffix "ta marbutta".
- **x9**: The number of letters that must be in a word for the function to remove the suffix "yah-heh".
- **x10**: The number of letters that must be in a word for the function to remove the suffix "yah-ta ma".
- **dontstem**: Characters that are not to be removed as suffixes.
The number of letters that must be in a word for the function to remove the suffix "yah-ta marbutta".

The number of letters that must be in a word for the function to remove the suffix "heh".

The number of letters that must be in a word for the function to remove the suffix "ta marbutta".

The number of letters that must be in a word for the function to remove the suffix "yah".

dontstem Words that should not be stemmed (entered in unicode).

Value

Returns a string with Arabic suffixes removed.

Author(s)

Rich Nielsen

Examples

## Create string with Arabic characters

```
x <- '\u0627\u0644\u0644\u063a\u0629 \u0627\u0644\u0639\u0628\u064a\u0629 \
\u062c\u0645\u064a\u0644\u0629 \u062c\u062f\u0627'
```

# Remove Suffixes

```
removeSuffixes(x)
```
**stem**

**Value**

Returns a string in Arabic characters.

**Author(s)**

Rich Nielsen

**Examples**

```r
## Create latin string following the arabicStemR package transliteration scheme.
x <- 'al3rby'

## Convert latin characters into Arabic unicode characters
reverse.transliterate(x)
```

---

**Description**

Allows users to stem Arabic texts for text analysis. Now deprecated. Please use stemArabic.

**Usage**

```r
stem(dat, cleanChars = TRUE, cleanLatinChars = TRUE,
     transliteration = TRUE, returnStemList = FALSE,
     defaultStopwordList=TRUE, customStopwordList=NULL,
     dontStemTheseWords = c("allh", "llh"))
```

**Arguments**

- `dat` The original data, as a vector of length one containing the text.
- `cleanChars` Removes all unicode characters except Latin characters and Arabic alphabet
- `cleanLatinChars` Removes Latin characters
- `transliteration` Transliterates the text
- `returnStemList` Performs stemming by removing prefixes and suffixes
- `defaultStopwordList` If TRUE, use the default stopword list of words to be removed. If FALSE, do not use the default stopword list. Default is TRUE.
customStopwordList

Optional user-specified stopword list of words to be removed, supplied as a vec-
tor of strings in either Arabic UTF-8 or Latin characters following the stemmer’s
transliteration scheme (words without Arabic UTF-8 characters are processed
with reverse.transliterate()). Default is NULL.

dontStemTheseWords

Optional vector of strings that should not be stemmed. These words can be sup-
plied as transliterated Arabic (according to the transliteration scheme of translit-
erate() and reverse.transliterate()) or in unicode Arabic. If a term matches an
element of this argument at any intermediate point in stemming, that term will
not be stemmed further. The default is c("allh","llh") because in most appli-
cations, stemming these common words for "God" creates some confusion by
resulting in the string "lh".

Details

stem prepares texts in Arabic for text analysis by stemming.

Value

stem returns a named list with the following elements:

text The stemmed text
stemlist A list of the stemmed words.

Author(s)

Rich Nielsen

Examples

## generate some text in Arabic
x <- "\u628\u633\u645 \u0627\u0644\u0647
\u0627\u0644\u0631\u062D\u0645\u0646
\u0627\u0644\u0631\u062D\u064A\u0645"

## stem and transliterate
## NOTE: the "stem()" function only accepts a vector of length 1.
## The function is deprecated in favor of stemArabic() which accepts vectors with multiple elements.
stem(x)

## stem while not stemming certain words
stem(x, dontStemTheseWords = c("alr7mn"))

## stem and return the stemlist
out <- stem(x,returnStemList=TRUE)
out$text
out$stemlist
stemArabic

Arabic Stemmer for Text Analysis

Description

Allows users to stem Arabic texts for text analysis.

Usage

stemArabic(dat, cleanChars = TRUE, cleanLatinChars = TRUE, transliteration = TRUE, returnStemList = FALSE, defaultStopwordList = TRUE, customStopwordList = NULL, dontStemTheseWords = c("allah", "llh"))

Arguments

dat The original data, as a vector of texts.
cleanChars Removes all unicode characters except Latin characters and Arabic alphabet
cleanLatinChars Removes Latin characters
transliteration Transliterates the text
returnStemList Performs stemming by removing prefixes and suffixes
defaultStopwordList If TRUE, use the default stopword list of words to be removed. If FALSE, do not use the default stopword list. Default is TRUE.
customStopwordList Optional user-specified stopword list of words to be removed, supplied as a vector of strings in either Arabic UTF-8 or Latin characters following the stemmer’s transliteration scheme (words without Arabic UTF-8 characters are processed with reverse.transliterate()). Default is NULL.
dontStemTheseWords Optional vector of strings that should not be stemmed. These words can be supplied as transliterated Arabic (according to the transliteration scheme of transliterate() and reverse.transliterate()) or in unicode Arabic. If a term matches an element of this argument at any intermediate point in stemming, that term will not be stemmed further. The default is c("allah","llh") because in most applications, stemming these common words for "God" creates some confusion by resulting in the string "lh".

Details

stemArabic prepares texts in Arabic for text analysis by stemming.
transliterate

Value

stemArabic returns a named list with the following elements:

- **text**: The stemmed text
- **stemlist**: A list of the stemmed words.

Author(s)

Rich Nielsen

Examples

```r
## generate some text in Arabic
x <- "\u628\u633\u645 \u0627\u0644\u0647
\u0627\u0644\u0631\u062D\u0646
\u0627\u0644\u0631\u062D\u0646\u064A\u0645"

## inspect
print(x)

## stem and transliterate
stemArabic(x)

## stem while not stemming certain words
stem(x, dontStemTheseWords = c("alr7mn"))

## stem and return the stemlist
out <- stemArabic(x, returnStemList=TRUE)
out$text
out$stemlist
```

transliterate  Transliterate Arabic unicode characters into latin characters

Description

Transliterates Arabic unicode characters into latin characters using a transliteration system developed by Rich Nielsen.

Usage

transliterate(texts)

Arguments

texts  A string in Arabic characters to be transliterated into latin characters.

Value

Returns a string in latin characters.
transliterate

**Author(s)**
Rich Nielsen

**Examples**

```r
## Create Arabic string
x <- '\u0627\u0647\u0644\u0627 \u0648\u0633\u0647\u0644\u0627'

## Performs transliteration of Arabic into latin characters.
transliterate(x)
```
Index

* package
  arabicStemR-package, 2
* utilities
  cleanChars, 3
  cleanLatinChars, 3

arabicStemR(arabicStemR-package), 2
arabicStemR-package, 2

cleanChars, 3
cleanLatinChars, 3

doStemming, 4
fixAlifs, 5
removeArabicNumbers, 6
removeDiacritics, 6
removeEnglishNumbers, 7
removeFarsiNumbers, 8
removeNewlineChars, 8
removeNumbers, 9
removePrefixes, 10
removePunctuation, 11
removeStopWords, 12
removeSuffixes, 13
reverse.transliterate, 14

stem, 15
stemArabic, 2, 17

transliterate, 18