Package ‘arabicStemR’

February 7, 2017

Type Package
Title Arabic Stemmer for Text Analysis
Version 1.2
Date 2017-02-07
Author Rich Nielsen
Maintainer Rich Nielsen <nielsen@mit.edu>
Description Allows users to stem Arabic texts for text analysis.
License GPL (>= 2)
NeedsCompilation no
Repository CRAN
Date/Publication 2017-02-07 10:08:37

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

Description

This package is a stemmer for texts in Arabic, as part of the txtorg utility for text analysis workflow. The stemmer is loosely based on the light 10 stemmer, but with a number of modifications.

Details

Use the `stem` function.

Author(s)

Maintainer: Rich Nielsen <rnielsen@mit.edu>

See Also

`stem`

Examples

```r
# Load data
data(aljazeera)

## stem and transliterate the results
stem(aljazeera)

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

aljazeera  Arabic text

Description

Arabic text from the front page of Aljazeera

Usage

data("aljazeera")
**cleanChars**

**Format**

Arabic text data

**Source**

http://aljazeera.net/portal

**Examples**

```r
## Load data
data(aljazeera)
```

---

**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**

```r
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\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)`

**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\u0627 \u0633\u0647\u0644\u0627 Hello'

## Remove 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\u0644\u0644\u0647','\u0644\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
x <- '\u0627\u0644\u0644\u062a\u0629 \u0627\u0644\u0639\u0628\u064a\u0629 \ج\u0645\u064a\u0644\u0629 \u062c\u062f\u0627'

## Remove prefixes and suffixes
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

## Create string with Arabic characters

```r
x <- '\u0622 \u0623 \u0675'
```

## Standardize Alifs

```r
fixAlifs(x)
```

---

removeArabicNumbers  Remove Arabic numbers

Description

Removes Arabic numerals from a string.

Usage

```r
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

```r
x <- '\u0627\u0647\u0644\u0627 \u0648\u0633\u0647\u0644\u0627 \u0661\u0662\u0663'
```

## Remove Arabic numbers

```r
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

```sh
x <- '\u0627\u0647\u0627\u064b \u0648\u0633\u0647\u0644\u0627\u064b'
```

## Remove diacritics

```sh
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

```r
CC create string with Arabic characters and English number
x <- '\u0627\u0647\u0644\u0627 \u0648\u0633\u0647\u0644\u0627 123'

CC remove English numbers
removeNumbers(x)
```

---

### removeFarsiNumbers

**Remove Farsi numbers**

Description

Removes Farsi numerals from a string.

Usage

`removeFarsiNumbers(texts)`

Arguments

texts A string from which Farsi numerals should be removed.

Value

removeFarsiNumbers returns a string with Arabic numerals removed.

Author(s)

Rich Nielsen

Examples

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

CC remove Farsi numbers
removeFarsiNumbers(x)
```
**removeNewlineChars**

*Remove new line characters*

**Description**

Removes new line characters from a string.

**Usage**

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

**Arguments**

- **texts**
  A string from which new line characters should be removed.

**Value**

\[\text{removeNewlineChars} \text{ returns a string with new line characters removed.}\]

**Author(s)**

Rich Nielsen

**Examples**

```r
## Create string with Arabic characters
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)

\text{removeNewlineChars}(x)
```

---

**removeNumbers**

*Remove English, Arabic, and Farsi numerals.*

**Description**

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

**Usage**

\[\text{removeNumbers}(\text{texts})\]
removePrefixes

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", "kaft-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 remain in a word for the function to remove the prefix "waw".

x2
The number of letters that must remain in a word for the function to remove the prefix "alif-lam".

x3
The number of letters that must remain in a word for the function to remove the prefix "waw-alif-lam".
The number of letters that must remain in a word for the function to remove the prefix "ba-alif-lam".

The number of letters that must remain in a word for the function to remove the prefix "kaf-alif-lam".

The number of letters that must remain in a word for the function to remove the prefix "fa-alif-lam".

The number of letters that must remain 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

## Create string with Arabic characters

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

# Remove Prefixes

removePrefixes(x)
```

removePunctuation

Remove punctuation.

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.
removeStopWords

Author(s)
Rich Nielsen

Examples

## Create string with Arabic characters and punctuation

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

## Remove punctuation

```r
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 \ي\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 remain in a word for the function to remove the suffix "ha-alif".
- **x2**: The number of letters that must remain in a word for the function to remove the suffix "alif-nun".
- **x3**: The number of letters that must remain in a word for the function to remove the suffix "alif-ta".
- **x4**: The number of letters that must remain in a word for the function to remove the suffix "waw-nun".
- **x5**: The number of letters that must remain in a word for the function to remove the suffix "yah-nun".
- **x6**: The number of letters that must remain in a word for the function to remove the suffix "yah-heh".
reverse.transliterate

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

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

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

The number of letters that must remain 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

```r
## Create string with Arabic characters
x <- '\u0627\u0644\u0644\u063a\u0629 \u0627\u0644\u0639\u0628\u064a\u0629 \\
\u062c\u0645\u0644\u0629 \u062c\u062f\u0627'

# Remove Suffixes
removeSuffixes(x)
```

---

reverse.transliterate  Transliterate latin characters into Arabic unicode characters

Description

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

Usage

```r
reverse.transliterate(texts)
```

Arguments

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

Value

Returns a string in Arabic characters.

Author(s)

Rich Nielsen

Examples

```r
## Create latin string
x <- 'hello'

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

Description

Allows users to stem Arabic texts for text analysis.

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.
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("allh","llh") because in most applications, 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

# Load data
data(aljazeera)

## stem and transliterate the results
stem(aljazeera)

## stem while not stemming certain words
stem(aljazeera, dontStemTheseWords = c("aljzyr\"\r\"\"))

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

## This allows you to see which words are being combined
## Interpret this as follows:
i <- 1
## This is the i'th stem in quotes (with the original word as the label)
out$stemlist[[i]]
## These are all the words that resolve to the same stem.
names(out$stemlist[[out$stemlist==out$stemlist[[i]]])
## And this will provide a count.
mytab <- table(names(out$stemlist[[out$stemlist==out$stemlist[[i]]])
for(i in 1:length(mytab))(print(mytab[i]))
## Note that if you just look at “mytab”, it will appear incorrect because
## R displays the Arabic labels from right to left but the numbers from left
## to right (thanks R!).

## This can be done for all of the stems
result <- sapply(out$stemlist, function(x)(table(names(out$stemlist[[out$stemlist==x]]))))
for(i in 1:length(result)){
    cat(paste("stemmed:" ,out$stemlist[[i]],"\n"))
    cat("unstemmed:")
    print(result[[i]])
    cat("\n")
}
## display the results correctly for the i'th stem
i <- 1
for(j in 1:length(result[[i]])){print(result[[i]][j])}

---

transliterate

**Transliterate Arabic unicode characters into latin characters**

**Description**

Transliterate 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.

**Author(s)**

Rich Nielsen
Examples

## Create Arabic string

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

## Performs transliteration of Arabic into latin characters

transliterate(x)
Index

- Topic `textscititledekw1`
  - cleanChars, 3
  - cleanLatinChars, 4
  - doStemming, 4
  - fixAlifs, 5
  - removeArabicNumbers, 6
  - removeDiacritics, 7
  - removeEnglishNumbers, 7
  - removeFarsiNumbers, 8
  - removeNewlineChars, 9
  - removeNumbers, 9
  - removePrefixes, 10
  - removePunctuation, 11
  - removeStopWords, 12
  - removeSuffixes, 13
  - reverse.transliterate, 14
  - transliterate, 17

- Topic `textscititledekw2`
  - cleanChars, 3
  - cleanLatinChars, 4
  - doStemming, 4
  - fixAlifs, 5
  - removeArabicNumbers, 6
  - removeDiacritics, 7
  - removeEnglishNumbers, 7
  - removeFarsiNumbers, 8
  - removeNewlineChars, 9
  - removeNumbers, 9
  - removePrefixes, 10
  - removePunctuation, 11
  - removeStopWords, 12
  - removeSuffixes, 13
  - reverse.transliterate, 14
  - transliterate, 17

- Topic `datasets`
  - aljazeera, 2

- Topic `package`
  - arabicStemR, 2
  - arabicStemR-package, 2
  - cleanChars, 3
  - cleanLatinChars, 4
  - doStemming, 4
  - fixAlifs, 5
  - removeArabicNumbers, 6
  - removeDiacritics, 7
  - removeEnglishNumbers, 7
  - removeFarsiNumbers, 8
  - removeNewlineChars, 9
  - removeNumbers, 9
  - removePrefixes, 10
  - removePunctuation, 11
  - removeStopWords, 12
  - removeSuffixes, 13
  - reverse.transliterate, 14
  - transliterate, 17
  - stem, 2, 15
  - transliterate, 17