Package: MadanText (via r-universe)

October 17, 2024

Type Package

Title Persian Textmining Tool for Frequency Analysis, Statistical Analysis, and Word Clouds

Version 0.1.0

Description MadanText is an open-source software designed specifically for text mining in the Persian language. It allows users to examine word frequencies, download data for analysis, and generate word clouds. This tool is particularly useful for researchers and analysts working with Persian language data.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Depends R (>= 4.0.0)

Imports xlsx, lattice, stopwords,textmineR, tidytext, stringi, tidyr, udpipe, PersianStemmer, shiny (>= 1.8.0), shinythemes, tm, dplyr, hwordcloud, stringr, topicmodels

Repository https://kidoishi.r-universe.dev

RemoteUrl https://github.com/kidoishi/madantext

RemoteRef HEAD

RemoteSha 33ac066aa2446de0bcf297744466109de4a73435

Contents

`ASDATA.FRAME` *Convert to Data Frame*

Description

This function converts the given object to a data frame.

Usage

```
ASDATA.FRAME(x)
```

Arguments

`x`	An object to be converted into a data frame.

Value

A data frame.

Examples

```
data <- ASDATA.FRAME(matrix(1:4, ncol = 2))
```

`f3` *Persian Text Normalization and Stemming*

Description

This function normalizes Persian text by replacing specific characters and applies stemming.

Usage

```
f3(x)
```

Arguments

`x`	A character vector of Persian text.

Value

A character vector of normalized and stemmed text.

Examples

```
## Not run:
  text <- c(”Persian text here”)
  normalized_text <- f3(text)

## End(Not run)
```

`f5` *Filter Data Frame by Document ID*

Description

This function filters a data frame by the specified document ID. If the ID is 0, the entire data frame is returned.

Usage

```
f5(UPIP, I)
```

Arguments

`UPIP` A data frame with a column named ’doc_id’.

`I` An integer representing the document ID.

Value

A filtered data frame.

Examples

```
data <- data.frame(doc_id = 1:5, text = letters[1:5])
filtered_data <- f5(data, 2)
```

f6 *Extract Token Information from Data Frame*

Description

This function extracts token, lemma, and part-of-speech (POS) tag information from a given data frame and compiles them into a new data frame.

Usage

```
f6(UPIP)
```

Arguments

`UPIP`	A data frame containing columns 'token', 'lemma', and 'upos' for tokens, their lemmatized forms, and POS tags respectively.

Value

A data frame with columns 'TOKEN', 'LEMMA', and 'TYPE', each representing token, its lemma, and POS tag.

Examples

```
data <- data.frame(token = c("running", "jumps"),
                   lemma = c("run", "jump"),
                   upos = c("VERB", "VERB"))
token_info <- f6(data)
```

f7 *Extract and Count Specific Parts of Speech*

Description

This function extracts tokens of a specified part of speech (POS) from the given data frame and counts their frequency.

Usage

```
f7(UPIP, type)
```

Arguments

`UPIP`	A data frame with columns 'upos' (POS tags) and 'lemma' (lemmatized tokens).
`type`	A string representing the POS to filter (e.g., 'NOUN', 'VERB').

Value

A data frame with frequencies of each lemma for the specified POS.

Examples

```
data <- data.frame(upos = c('NOUN', 'VERB'), lemma = c('house', 'run'))
noun_freq <- f7(data, 'NOUN')
```

fun.all.sums *Apply Suffix Modifications to Persian Words*

Description

This function iteratively applies a series of suffix modifications to a vector of Persian words.

Usage

```
fun.all.sums(v, TYPE)
```

Arguments

`v`	A character vector of Persian words.
`TYPE`	A vector of suffix types for modification.

Value

A modified character vector.

Examples

```
## Not run:
  words <- c("Persian text here")
  modified_words <- fun.all.sums(words, TYPE)

## End(Not run)
```

fun.one.sums *General Persian Suffix Modification*

Description

This function modifies Persian words based on a specified suffix type.

Usage

```
fun.one.sums(v, type)
```

Arguments

v	A character vector of Persian words.
type	A character string representing the suffix type.

Value

A modified character vector.

Examples

```
## Not run:
  words <- c("Persian text here")
  modified_words <- fun.one.sums(words, "Persian text here")

## End(Not run)
```

fungan *Persian Suffix Modification for 'Persian text here' Suffix*

Description

This function modifies Persian words ending with 'Persian text here' suffix.

Usage

```
fungan(v)
```

Arguments

v	A character vector of Persian words.

Value

A modified character vector.

Examples

```
## Not run:
  words <- c("Persian text here")
  modified_words <- fungan(words)

## End(Not run)
```

`fungi` *Persian Suffix Modification*

Description

This function modifies Persian words ending with ’Persian text here’ suffix.

Usage

```
fungi(v)
```

Arguments

`v`	A character vector of Persian words.

Value

A modified character vector.

Examples

```
## Not run:
  words <- c("Persian text here")
  modified_words <- fungi(words)

## End(Not run)
```

`funmi` *Modify Persian Words Starting with ’Persian text here’*

Description

This function modifies Persian words starting with the prefix ’Persian text here’.

Usage

```
funmi(v)
```

Arguments

v	A character vector of Persian words.

Value

A modified character vector.

Examples

```
 ## Not run:
  words <- c("Persian text here")
  modified_words <- funmi(words)

## End(Not run)
```

LEMMA *Persian Lemmatization*

Description

This function performs lemmatization on a vector of Persian words.

Usage

```
LEMMA(Y, TYPE)
```

Arguments

Y	A character vector of Persian words.
TYPE	A vector of suffix types for modification.

Value

A vector of lemmatized Persian words.

Examples

```
## Not run:
  words <- c("Persian text here")
  lemmatized_words <- LEMMA(words, TYPE)

## End(Not run)
```

PMI *Calculate Pointwise Mutual Information (PMI)*

Description

This function calculates the PMI for collocations in a given text data.

Usage

```
PMI(x)
```

Arguments

`x`	A data frame with columns 'token' and 'doc_id'.

Value

A data frame of keywords with their PMI scores.

Examples

```
data <- data.frame(token = c("word1", "word2"), doc_id = c(1, 1))
pmi_scores <- PMI(data)
```

server *Server Logic for MadanText Shiny Application*

Description

This function contains the server-side logic for the MadanText application. It handles user inputs, processes data, and creates outputs to be displayed in the UI.

Usage

```
server(input, output)
```

Arguments

`input`	List of Shiny inputs.
`output`	List of Shiny outputs.

TYPE *Persian Suffixes*

Description

A vector of common Persian suffixes used for text processing.

Usage

```
TYPE
```

Format

An object of class `character` of length 39.

ui *User Interface for MadanText*

Description

This function creates a user interface for the MadanText Shiny application. It includes various input and output widgets for file uploads, text input, and visualization.

Usage

```
ui
```

Format

An object of class `shiny.tag.list` (inherits from `list`) of length 4.

Value

A Shiny UI object.

Index