Package: tangela (via r-universe)

January 1, 2025

Type Package

Title rJava Interface to Kuromoji

Version 0.2.0

Maintainer Akiru Kato <paithiov909@gmail.com>

Description An rJava wrapper for atilika/kuromoji (v0.7.7). This package will work fine, but it is too slow to be used in production.

License Apache License (>= 2)

URL https://github.com/paithiov909/tangela

BugReports https://github.com/paithiov909/tangela/issues

Depends R (>= 3.2.0)

Imports dplyr, purrr, readr, rJava, rlang (>= 0.1.2), stringi

Suggests knitr, rmarkdown, roxygen2

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

SystemRequirements Java

Config/pak/sysreqs make default-jdk libicu-dev libx11-dev

Repository https://paithiov909.r-universe.dev

RemoteUrl https://github.com/paithiov909/tangela

RemoteRef HEAD

RemoteSha 78cb23ab919763db447edae18fcca63f84e6e83a

Contents

`kuromoji` *Call kuromoji tokenizer*

Description

Call kuromoji tokenizer

Usage

```
kuromoji(chr)
```

Arguments

`chr`	Character vector to be tokenized.

Value

A tibble.

`ngram_tokenizer` *Ngrams tokenizer*

Description

Makes an ngram tokenizer function.

Usage

```
ngram_tokenizer(n = 1L)
```

Arguments

`n`	Integer.

Value

ngram tokenizer function

pack *Pack a data.frame of tokens*

Description

Packs a data.frame of tokens into a new data.frame of corpus, which is compatible with the Text Interchange Formats.

Usage

```
pack(tbl, pull = "token", n = 1L, sep = "-", .collapse = " ")
```

Arguments

`tbl`	A data.frame of tokens.
`pull`	<data-masked> Column to be packed into text or ngrams body. Default value is `token`.
`n`	Integer internally passed to ngrams tokenizer function created of `tangela::ngram_tokenizer()`
`sep`	Character scalar internally used as the concatenator of ngrams.
`.collapse`	This argument is passed to `stringi::stri_c()`.

Value

A tibble.

Text Interchange Formats (TIF)

The Text Interchange Formats (TIF) is a set of standards that allows R text analysis packages to target defined inputs and outputs for corpora, tokens, and document-term matrices.

Valid data.frame of tokens

The data.frame of tokens here is a data.frame object compatible with the TIF.

A TIF valid data.frame of tokens are expected to have one unique key column (named `doc_id`) of each text and several feature columns of each tokens. The feature columns must contain at least `token` itself.

See Also

https://github.com/ropenscilabs/tif

prettify *Prettify tokenized output*

Description

Turns a single character column into features while separating with delimiter.

Usage

```
prettify(
  tbl,
  col = "feature",
 into = c("POS1", "POS2", "POS3", "POS4", "X5StageUse1", "X5StageUse2", "Original",
    "Yomi1", "Yomi2"),
  col_select = seq_along(into),
  delim = ","
)
```

Arguments

tbl	A data.frame that has feature column to be prettified.
col	<data-masked> Column name where to be prettified.
into	Character vector that is used as column names of features.
col_select	Character or integer vector that will be kept in prettified features.
delim	Character scalar used to separate fields within a feature.

Value

A data.frame.

Examples

```
prettify(
  data.frame(x = c("x,y", "y,z", "z,x")),
  col = "x",
  into = c("a", "b"),
  col_select = "b"
)
```

`rebuild_tokenizer` *Initialize kuromoji tokenizer*

Description

Initialize kuromoji tokenizer

Usage

```
rebuild_tokenizer(user_dic = "")
```

Arguments

`user_dic`	file path to a user dictionary if any.

Value

The stored kuromoji tokenizer instance is returned invisibly.

Index