Package: sudachir2 (via r-universe)

March 1, 2025

Title R Wrapper for 'sudachi.rs'

Version 0.6.10.4

Description Offers bindings to 'sudachi.rs' <https://github.com/WorksApplications/sudachi.rs>, a Rust implementation of 'Sudachi' Japanese morphological analyzer.

License Apache License (>= 2)

Depends R (>= 4.2)

Imports dplyr, purrr, readr, rlang, stringi

Suggests curl, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

SystemRequirements Cargo (Rust's package manager), rustc

Config/pak/sysreqs libicu-dev libx11-dev

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

RemoteUrl https://github.com/paithiov909/sudachir2

RemoteRef HEAD

RemoteSha 99415ce4f724be40ddf451fe3016cb5c8ca848fb

Contents

as_tokens . . . 2
create_tagger . . . 2
fetch_dict . . . 3
is_blank . . . 3
ngram_tokenizer . . . 4
pack . . . 5
prettify . . . 6
tokenize . . . 7

Index **8**

as_tokens *Create a list of tokens*

Description

Create a list of tokens

Usage

```
as_tokens(tbl, token_field = "token", pos_field = NULL, nm = NULL)
```

Arguments

tbl	A tibble of tokens out of `tokenize()`.
token_field	<data-masked> Column containing tokens.
pos_field	Column containing features that will be kept as the names of tokens. If you don't need them, give a `NULL` for this argument.
nm	Names of returned list. If left with `NULL`, "doc_id" field of `tbl` is used instead.

Value

A named list of tokens.

create_tagger *Create a tagger function*

Description

Create a tagger function

Usage

```
create_tagger(
  dictionary_path,
  config_file = system.file("resources/sudachi.json", package = "sudachir2"),
  resource_dir = system.file("resources", package = "sudachir2"),
  mode = c("C", "A", "B")
)
```

Arguments

dictionary_path	A path to a dictionary file such as `"system_core.dic"`.
config_file	A path to a config file.
resource_dir	A path to a resource directory.
mode	Split mode for 'sudachi.rs'. Either `"C"`, `"A"`, or `"B"`.

Details

This function just returns a wrapper function for tokenization, i.e. does not actually create a tagger instance. Even if arguments are invalid, this function does not raise any errors.

Value

A function inheriting class `purrr_function_partial`.

`fetch_dict` *Download and unarchive a dictionary for 'Sudachi'*

Description

Download and unarchive a dictionary for 'Sudachi'

Usage

```
fetch_dict(
  exdir,
  dict_version = "latest",
  dict_type = c("small", "core", "full")
)
```

Arguments

`exdir`	Directory where the dictionary will be unarchived.
`dict_version`	Version of the dictionary to be downloaded.
`dict_type`	Type of the dictionary to be downloaded. Either `"small"`, `"core"`, or `"full"`.

Value

`exdir` is invisibly returned.

`is_blank` *Check if scalars are blank*

Description

Check if scalars are blank

Usage

```
is_blank(x, trim = TRUE, ...)
```

Arguments

x	Object to check its emptiness.
trim	Logical.
...	Additional arguments for base::sapply().

Value

Logicals.

Examples

```
is_blank(list(c(a = "", b = NA_character_), NULL))
```

---

ngram_tokenizer *Ngrams tokenizer*

Description

Makes an ngram tokenizer function.

Usage

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

Arguments

n	Integer.

Value

ngram tokenizer function

Examples

```
bigram <- ngram_tokenizer(2)
bigram(letters, sep = "-")
```

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 `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 is 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", "cType", "cForm"),
  col_select = seq_along(into),
  delim = ","
)
```

Arguments

`tbl`	A data.frame that has feature column to be prettified.
`col`	<data-masked> Column containing features 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"
)
```

tokenize *Tokenize sentences using a tagger function*

Description

Tokenize sentences using a tagger function

Usage

```
tokenize(x, text_field = "text", docid_field = "doc_id", tagger)
```

Arguments

x	A data.frame like object or a character vector to be tokenized.
text_field	<data-masked> String or symbol; column containing texts to be tokenized.
docid_field	<data-masked> String or symbol; column containing document IDs.
tagger	A tagger function out of create_tagger()

Value

A tibble.

Index

as_tokens, 2

base::sapply(), *4*

create_tagger, 2
create_tagger(), *7*

fetch_dict, 3

is_blank, 3

ngram_tokenizer, 4
ngram_tokenizer(), *5*

pack, 5
prettify, 6

stringi::stri_c(), *5*

tokenize, 7