Package: moranajp (via r-universe)

September 1, 2024

Title Morphological Analysis for Japanese

Version 0.9.7

Description Supports morphological analysis for Japanese by using 'MeCab' <https://taku910.github.io/mecab/>, 'Sudachi' <https://github.com/WorksApplications/Sudachi>, 'Chamame' <https://chamame.ninjal.ac.jp/>, or 'Ginza' <https://github.com/megagonlabs/ginza>. Can input a data.frame and obtain all results of 'MeCab' and the row number of the original data.frame as a text id.

License MIT + file LICENSE

Depends R (>= 3.5.0)

URL https://github.com/matutosi/moranajp, https://matutosi.github.io/moranajp/

BugReports https://github.com/matutosi/moranajp/issues

Imports dplyr, ggplot2, ggraph, grid, igraph, purrr, rlang, rvest, stats, stringr, stringi, tibble, tidyr, utils

Suggests devtools, knitr, rmarkdown, spelling, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Language en-US

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

RemoteUrl https://github.com/matutosi/moranajp

RemoteRef HEAD

RemoteSha 45007cd60630c4e6fc8bf6fdf1cd22f7c24c8f56

Contents

`add_group` *Add group id column into result of morphological analysis*

Description

Add group id column into result of morphological analysis

Usage

```
add_group(
  tbl,
  col,
  brk = "EOS",
  grp = "group",
  cond = NULL,
  end_with_brk = TRUE
)
```

Arguments

tbl	A dataframe
col	A string to specify the column including breaks
brk	A string to specify breaks
grp	A string to specify group
cond	A string to specify condition
end_with_brk	A logical

Value

A dataframe

Examples

```
brk <- "EOS"
tbl <- tibble::tibble(col=c(rep("a", 2), brk, rep("b", 3), brk, rep("c", 4), brk))
add_group(tbl, col = "col")
add_group(tbl, col = "col", end_with_brk = FALSE)
```

---

add_id *Add id in each group*

Description

Add id in each group

Usage

```
add_id(tbl, grp = "group", id = "id")
```

Arguments

tbl	A dataframe
grp, id	A string to specify the column of group and id

Value

A dataframe

Examples

```
brk <- "EOS"
tbl <- tibble::tibble(col=c(rep("a", 2), brk, rep("b", 3), brk, rep("c", 4), brk))
add_group(tbl, col = "col") |>
  add_id(id = "id_in_group")
```

add_sentence_no *Wrapper function for add_group() to add sentence id*

Description

Wrapper function for add_group() to add sentence id

Usage

```
add_sentence_no(df, s_id = "sentence")
```

Arguments

df	A dataframe
s_id	A string for sentence colame

Value

A dataframe

Examples

```
review_mecab |>
  unescape_utf() |>
  add_sentence_no() |>
  print(n=200)
```

add_text_id *Add id column into result of morphological analysis*

Description

Internal function for moranajp_all(). Add text_id column when there is brk ("BPMJP"). "BPMJP": Break Point Of MoranaJP

Usage

```
add_text_id(tbl, method, brk = "BPMJP")
```

Arguments

tbl	A tibble or data.frame.
method	A text. Method to use: "mecab", "ginza", "sudachi_a", "sudachi_b", "sudachi_c", or "chamame". "a", "b" and "c" specify the mode of splitting. "a" split shortest, "b" middle and "c" longest. See https://github.com/WorksApplications/Sudachi for detail. "chamame" use https://chamame.ninjal.ac.jp/ and rvest.
brk	A string of break point

Value

A data.frame with column "text_id".

---

clean_up *Clean up result of morphological analyzed data frame*

Description

Clean up result of morphological analyzed data frame

Usage

```
clean_up(df, add_depend = FALSE, ...)

pos_filter(df)

add_depend_ginza(df)

delete_stop_words(df, use_common_data = TRUE, add_stop_words = NULL, ...)

replace_words(
  df,
  synonym_df = tibble::tibble(),
  synonym_from = "",
  synonym_to = "",
  ...
)

term_lemma(df)

term_pos_0(df)

term_pos_1(df)
```

Arguments

df	A dataframe including result of morphological analysis.
add_depend	A logical. Available for ginza
...	Extra arguments to internal functions.
use_common_data	A logical. TRUE: use data(stop_words).
add_stop_words	A string vector adding into stop words. When use_common_data is TRUE and add_stop_words are given, both of them will be used as stop_words.
synonym_df	A data.frame including synonym word pairs. The first column: replace from, the second: replace to.

synonym_from, synonym_to
: A string vector. Length of synonym_from and synonym_to should be the same. When synonym_df and synonym pairs (synonym_from and synonym_to) are given, both of them will be used as synonym.

Value

A data.frame.

Examples

```
data(neko_mecab)
data(neko_ginza)
data(review_sudachi_c)
data(synonym)
synonym <-
  synonym |> unescape_utf()

neko_mecab <-
  neko_mecab |>
  unescape_utf() |>
  print()

neko_mecab |>
  clean_up(use_common_data = TRUE, synonym_df = synonym)

review_ginza |>
  unescape_utf() |>
  add_sentence_no() |>
  clean_up(add_depend = TRUE, use_common_data = TRUE, synonym_df = synonym)

review_sudachi_c |>
  unescape_utf() |>
  add_sentence_no() |>
  clean_up(use_common_data = TRUE, synonym_df = synonym)
```

---

combine_words *Combine words after morphological analysis*

Description

Combine words after morphological analysis

Usage

```
combine_words(df, combi, sep = "-")

combi_words(x, combi, sep = "-")
```

Arguments

df	A dataframe including result of morphological analysis.
combi	A string (combi_words()) or string vector (combine_words()) to combine words.
sep	A string of separator of words
x	A pair of string joining with "-"

Value

A data.frame with combined words.

Examples

```
x <- letters[1:10]
combi <- c("b-c")
combi_words(x, combi)
expected <- c("a", "bc", NA, "d", "e", "f", "g", "h", "i", "j")
testthat::expect_equal(combi_words(x, combi), expected)

df <- unescape_utf(review_chamame) |> head(20)
combi <- unescape_utf(
           c("\\u751f\\u7269-\\u591a\\u69d8", "\\u8fb2\\u5730-\\u306f"       ,
             "\\u8fb2\\u7523-\\u7269"       , "\\u751f\\u7523-\\u3059\\u308b"))
combine_words(df, combi)
```

---

draw_bigram_network *Draw bigram network using morphological analysis data.*

Description

Draw bigram network using morphological analysis data.

Usage

```
draw_bigram_network(df, draw = TRUE, ...)

bigram(df, group = "sentence", depend = FALSE, term_depend = NULL, ...)

trigram(df, group = "sentence")

bigram_depend(df, group = "sentence")

bigram_network(bigram, rand_seed = 12, threshold = 100, ...)

word_freq(df, big_net, ...)

bigram_network_plot(
```

```
  big_net,
  freq,
  ...,
  arrow_size = 5,
  circle_size = 5,
  text_size = 5,
  font_family = "",
  arrow_col = "darkgreen",
  circle_col = "skyblue",
  x_limits = NULL,
  y_limits = NULL,
  no_scale = FALSE
)
```

Arguments

df	A dataframe including result of morphological analysis.
draw	A logical.
...	Extra arguments to internal functions.
group	A string to specify sentence.
depend	A logical.
term_depend	A string of dependent terms column to use bigram.
bigram	A result of bigram().
rand_seed	A numeric.
threshold	A numeric used as threshold for frequency of bigram.
big_net	A result of bigram_network().
freq	A numeric of word frequency in bigram_network. Can be got using word_freq().
arrow_size, circle_size, text_size	A numeric.
font_family	A string.
arrow_col, circle_col	A string to specify arrow and circle color in bigram network.
x_limits, y_limits	A Pair of numeric to specify range.
no_scale	A logical. FALSE: Not draw x and y axis.

Value

A list including df (input), bigram, freq (frequency) and gg (ggplot2 object of bigram network plot).

Examples

```
sentences <- 50
len <- 30
n <- sentences * len
```

```
x <- letters
prob <- (length(x):1) ^ 3
df <-
  tibble::tibble(
    lemma = sample(x = x, size = n, replace = TRUE, prob = prob),
    sentence = rep(seq(sentences), each = len))
draw_bigram_network(df)
```

---

escape_japanese *Generate code like "stringi::stri_unescape_unicode(...)"*

Description

Generate code like "stringi::stri_unescape_unicode(...)"

Usage

```
escape_japanese(x)
```

Arguments

x	A string or vector of Japanese

Value

```
A string or vector
```

Examples

```
stringi::stri_unescape_unicode("\\u8868\\u5c64\\u5f62") |>
  print() |>
  escape_japanese()
```

---

iconv_x *iconv x*

Description

iconv x

Usage

```
iconv_x(x, iconv = "", reverse = FALSE)
```

Arguments

x	A string vector or a tibble.
iconv	A text. Convert encoding of MeCab output. Default (""): don't convert. "CP932_UTF-8": iconv(output, from = "Shift-JIS" to = "UTF-8") "EUC_UTF-8" : iconv(output, from = "eucjp", to = "UTF-8") iconv is also used to convert input text before running MeCab. "CP932_UTF-8": iconv(input, from = "UTF-8", to = "Shift-JIS")
reverse	A logical.

Value

```
A string vector.
```

---

make_groups *Make groups by splitting string length*

Description

Using 'MeCab' for morphological analysis. Keep other colnames in dataframe.

Usage

```
make_groups(
  tbl,
  text_col = "text",
  length = 8000,
  tmp_group = "tmp_group",
  str_length = "str_length"
)

make_groups_sub(tbl, text_col, n_group, tmp_group, str_length)

max_sum_str_length(tbl, tmp_group, str_length)
```

Arguments

tbl	A tibble or data.frame.
text_col	A text. Colnames for morphological analysis.
length	A numeric.
tmp_group, str_length	A string to use temporary.
n_group	A numeric.

Value

A tibble. Output of morphological analysis and added column "text_id".

A string

A string

A string

A character vector

A character vector

A character vector

A character vector

A character vector

A data.frame

Examples

```
  # sample data of Japanese sentences
  data(neko)
  neko <-
      neko |>
      unescape_utf()
  # chamame
  neko |>
    moranajp_all(method = "chamame") |>
        print(n=100)

## Not run:
  # Need to install 'mecab', 'ginza', or 'sudachi' in local PC

  # mecab
  bin_dir <- "d:/pf/mecab/bin"
  iconv <- "CP932_UTF-8"
  neko |>
    moranajp_all(text_col = "text", bin_dir = bin_dir, iconv = iconv) |>
        print(n=100)

  # ginza
  neko |>
    moranajp_all(text_col = "text", method = "ginza") |>
      print(n=100)

  # sudachi
  bin_dir <- "d:/pf/sudachi"
  iconv <- "CP932_UTF-8"
  neko |>
    moranajp_all(text_col = "text", bin_dir = bin_dir,
                 method = "sudachi_a", iconv = iconv) |>
        print(n=100)

## End(Not run)
```

moranajp_all *Morphological analysis for a specific column in dataframe*

Description

Using 'MeCab' for morphological analysis. Keep other colnames in dataframe.

Usage

```
moranajp_all(
  tbl,
  bin_dir = "",
  method = "mecab",
  text_col = "text",
  option = "",
  iconv = "",
  col_lang = "jp"
)

moranajp(tbl, bin_dir, method, text_col, option = "", iconv = "", col_lang)

remove_linebreaks(tbl, text_col)

separate_cols_ginza(tbl, col_lang)

make_input(tbl, text_col, iconv, brk = "BPMJP ")

make_cmd(method, bin_dir, option = "")

make_cmd_mecab(option = "")

out_cols_mecab(col_lang = "jp")

out_cols_ginza(col_lang = "jp")

out_cols_sudachi(col_lang = "jp")

out_cols_jp()

out_cols_en()

out_cols()

mecab_all(tbl, text_col = "text", bin_dir = "")

mecab(tbl, bin_dir)
```

Arguments

tbl	A tibble or data.frame.
bin_dir	A text. Directory of mecab.
method	A text. Method to use: "mecab", "ginza", "sudachi_a", "sudachi_b", "sudachi_c", or "chamame". "a", "b" and "c" specify the mode of splitting. "a" split shortest, "b" middle and "c" longest. See https://github.com/WorksApplications/Sudachi for detail. "chamame" use https://chamame.ninjal.ac.jp/ and rvest.
text_col	A text. Colnames for morphological analysis.
option	A text. Options for mecab. "-b" option is already set by moranajp. To see option, use "mecab -h" in command (win) or terminal (Mac).
iconv	A text. Convert encoding of MeCab output. Default (""): don't convert. "CP932_UTF-8": iconv(output, from = "Shift-JIS" to = "UTF-8") "EUC_UTF-8" : iconv(output, from = "eucjp", to = "UTF-8") iconv is also used to convert input text before running MeCab. "CP932_UTF-8": iconv(input, from = "UTF-8", to = "Shift-JIS")
col_lang	A text. "jp" or "en"
brk	A string of break point

Value

A tibble. Output of morphological analysis and added column "text_id".

A string

A string

A string

A character vector

A character vector

A character vector

A character vector

A character vector

A data.frame

Examples

```
  # sample data of Japanese sentences
  data(neko)
  neko <-
      neko |>
      unescape_utf()
  # chamame
  neko |>
    moranajp_all(method = "chamame") |>
        print(n=100)

## Not run:
  # Need to install 'mecab', 'ginza', or 'sudachi' in local PC
```

```
# mecab
bin_dir <- "d:/pf/mecab/bin"
iconv <- "CP932_UTF-8"
neko |>
  moranajp_all(text_col = "text", bin_dir = bin_dir, iconv = iconv) |>
      print(n=100)

# ginza
neko |>
  moranajp_all(text_col = "text", method = "ginza") |>
    print(n=100)

# sudachi
bin_dir <- "d:/pf/sudachi"
iconv <- "CP932_UTF-8"
neko |>
  moranajp_all(text_col = "text", bin_dir = bin_dir,
               method = "sudachi_a", iconv = iconv) |>
      print(n=100)

## End(Not run)
```

---

neko *The first part of 'I Am a Cat' by Soseki Natsume*

Description

The first part of 'I Am a Cat' by Soseki Natsume

Usage

```
neko
```

Format

A data frame with 9 rows and 1 variable:

text Body text. Escaped by stringi::stri_escape_unicode().

Examples

```
data(neko)
neko |>
  unescape_utf()
```

neko_chamame *Analyzed data of neko by chamame*

Description

chamame: https://chamame.ninjal.ac.jp/index.html

Usage

```
neko_chamame
```

Format

A data frame with 2959 rows and 7 variable: (column names are escaped by stringi::stri_escape_unicode(), stringi::stri_unescape_unicode() or unescape_utf() will show Japanese)

text_id id

\u8868\u5c64\u5f62 result of chamame

\u54c1\u8a5e result of chamame

\u54c1\u8a5e\u7d30\u5206\u985e1 result of chamame

\u54c1\u8a5e\u7d30\u5206\u985e2 result of chamame

\u54c1\u8a5e\u7d30\u5206\u985e3 result of chamame

\u539f\u5f62 result of chamame

Examples

```
data(neko_chamame)
neko_chamame |>
  unescape_utf()
```

neko_ginza *Analyzed data of neko by GiNZA*

Description

GiNZA: https://megagonlabs.github.io/ginza/

Usage

```
neko_ginza
```

Format

A data frame with 2945 rows and 13 variable:

text_id id

id result of GiNZA

\u8868\u5c64\u5f62 result of GiNZA

\u539f\u5f62 result of GiNZA

UD\u54c1\u8a5e\u30bf\u30b0 result of GiNZA

\u54c1\u8a5e result of GiNZA

\u54c1\u8a5e\u7d30\u5206\u985e1 result of GiNZA

\u54c1\u8a5e\u7d30\u5206\u985e2 result of GiNZA

\u5c5e\u6027 result of GiNZA

\u4fc2\u53d7\u5143 result of GiNZA

\u4fc2\u53d7\u30bf\u30b0 result of GiNZA

\u4fc2\u53d7\u30da\u30a2 result of GiNZA

\u305d\u306e\u4ed6 result of GiNZA

Examples

```
data(neko_ginza)
neko_ginza |>
  unescape_utf()
```

---

neko_mecab *Analyzed data of neko by MeCab*

Description

MeCab: https://taku910.github.io/mecab/

Usage

```
neko_mecab
```

Format

A data frame with 2884 rows and 11 variable: (column names are escaped by stringi::stri_escape_unicode(), stringi::stri_unescape_unicode() or unescape_utf() will show Japanese)

text_id id

\u8868\u5c64\u5f62 result of MeCab

\u54c1\u8a5e result of MeCab

\u54c1\u8a5e\u7d30\u5206\u985e1 result of MeCab

\u54c1\u8a5e\u7d30\u5206\u985e2 result of MeCab

\u54c1\u8a5e\u7d30\u5206\u985e3 result of MeCab

\u6d3b\u7528\u578b result of MeCab

\u6d3b\u7528\u5f62 result of MeCab

\u539f\u5f62 result of MeCab

\u8aad\u307f result of MeCab

\u767a\u97f3 result of MeCab

Examples

```
data(neko_mecab)
neko_mecab |>
  unescape_utf()
```

`neko_sudachi_a` *Analyzed data of neko by Sudachi*

Description

Sudachi: https://github.com/WorksApplications/Sudachi

Usage

```
neko_sudachi_a

neko_sudachi_b

neko_sudachi_c
```

Format

A data frame with 3130 rows and 9 variable:

text_id id

\u8868\u5c64\u5f62 result of Sudachi

\u54c1\u8a5e result of Sudachi

\u54c1\u8a5e\u7d30\u5206\u985e1 result of Sudachi

\u54c1\u8a5e\u7d30\u5206\u985e2 result of Sudachi

\u54c1\u8a5e\u7d30\u5206\u985e3 result of Sudachi

\u54c1\u8a5e\u7d30\u5206\u985e4 result of Sudachi

\u54c1\u8a5e\u7d30\u5206\u985e5 result of Sudachi

\u539f\u5f62 result of Sudachi

A data frame with 3088 rows and 9 variable:

A data frame with 3080 rows and 9 variable:

Examples

```
data(neko_sudachi_a)
neko_sudachi_a |>
  unescape_utf()
```

`out_cols_chamame` *Morphological analysis for Japanese text by web chamame*

Description

Using https://chamame.ninjal.ac.jp/ and rvest.

Usage

```
out_cols_chamame(col_lang = "jp")

web_chamame(text, col_lang = "jp")

html_radio_set(form, ...)

is_radio(fields)
```

Arguments

`col_lang`	A text. "jp" or "en"
`text`	A text.
`form`	vest_form object
`...`	dynamic-dots Name-value pairs giving radio button to modify.
`fields`	$fields in vest_form object

Value

A character vector

A dataframe

vest_form object

A boolean or vector

Examples

```
text <-
  paste0("\\u3059",
         paste0(rep("\\u3082",8),collapse=""),
         "\\u306e\\u3046\\u3061") |>
  unescape_utf()
web_chamame(text)
```

remove_brk *Remove break point and other unused rows from the result of morphological analysis*

Description

Internal function for moranajp_all().

Usage

```
remove_brk(tbl, method, brk = "BPMJP")
```

Arguments

tbl	A tibble or data.frame.
method	A text. Method to use: "mecab", "ginza", "sudachi_a", "sudachi_b", "sudachi_c", or "chamame". "a", "b" and "c" specify the mode of splitting. "a" split shortest, "b" middle and "c" longest. See https://github.com/WorksApplications/Sudachi for detail. "chamame" use https://chamame.ninjal.ac.jp/ and rvest.
brk	A string of break point

Value

A data.frame.

review *Full text of review article*

Description

Full text of review article

Usage

```
review
```

Format

A data frame with 457 rows and 4 variables:

text Body text. Escaped by stringi::stri_escape_unicode(). Body text. Escaped by stringi::stri_escape_unicode(). Citation is as below. Matsumura et al. 2014. Conditions and conservation for biodiversity of the semi-natural grassland vegetation on rice paddy levees. Vegetation Science, 31, 193-218. doi = 10.15031/vegsci.31.193 https://www.jstage.jst.go.jp/article/vegsci/31/2/31_193/_article/-char/en

chap chapter

sect section

para paragraph

Examples

```
data(review)
review |>
  unescape_utf()
```

`review_chamame` *Analyzed data of review by chamame*

Description

chamame: https://chamame.ninjal.ac.jp/index.html

Usage

```
review_chamame
```

Format

A data frame with 21125 rows and 10 variable (column names are escaped by stringi::stri_escape_unicode(), stringi::stri_unescape_unicode() or unescape_utf() will show Japanese)

text_id id

chap chapter

sect section

para paragraph

\u8868\u5c64\u5f62 result of chamame

\u54c1\u8a5e result of chamame

\u54c1\u8a5e\u7d30\u5206\u985e1 result of chamame

\u54c1\u8a5e\u7d30\u5206\u985e2 result of chamame

\u54c1\u8a5e\u7d30\u5206\u985e3 result of chamame

\u539f\u5f62 result of chamame

Examples

```
data(review_chamame)
review_chamame |>
  unescape_utf()
```

`review_ginza` *Analyzed data of review by GiNZA*

Description

GiNZA: https://megagonlabs.github.io/ginza/

Usage

```
review_ginza
```

Format

A data frame with 19514 rows and 16 variable:

text_id id

chap chapter

sect section

para paragraph

id result of GiNZA

\u8868\u5c64\u5f62 result of GiNZA

\u539f\u5f62 result of GiNZA

UD\u54c1\u8a5e\u30bf\u30b0 result of GiNZA

\u54c1\u8a5e result of GiNZA

\u54c1\u8a5e\u7d30\u5206\u985e1 result of GiNZA

\u54c1\u8a5e\u7d30\u5206\u985e2 result of GiNZA

\u5c5e\u6027 result of GiNZA

\u4fc2\u53d7\u5143 result of GiNZA

\u4fc2\u53d7\u30bf\u30b0 result of GiNZA

\u4fc2\u53d7\u30da\u30a2 result of GiNZA

\u305d\u306e\u4ed6 result of GiNZA

Examples

```
data(review_ginza)
review_ginza |>
  unescape_utf()
```

`review_mecab` *Analyzed data of review by MeCab*

Description

MeCab: https://taku910.github.io/mecab/

Usage

```
review_mecab
```

Format

A data frame with 199985 rows and 14 variable: (column names are escaped by stringi::stri_escape_unicode(), stringi::stri_unescape_unicode() or unescape_utf() will show Japanese)

text_id id

chap chapter

sect section

para paragraph

\u8868\u5c64\u5f62 result of MeCab

\u54c1\u8a5e result of MeCab

\u54c1\u8a5e\u7d30\u5206\u985e1 result of MeCab

\u54c1\u8a5e\u7d30\u5206\u985e2 result of MeCab

\u54c1\u8a5e\u7d30\u5206\u985e3 result of MeCab

\u6d3b\u7528\u578b result of MeCab

\u6d3b\u7528\u5f62 result of MeCab

\u539f\u5f62 result of MeCab

\u8aad\u307f result of MeCab

\u767a\u97f3 result of MeCab

Examples

```
data(review_mecab)
review_mecab |>
  unescape_utf()
```

`review_sudachi_a` *Analyzed data of review by Sudachi*

Description

Sudachi: https://github.com/WorksApplications/Sudachi

Usage

```
review_sudachi_a

review_sudachi_b

review_sudachi_c
```

Format

A data frame with 20100 rows and 12 variable:

text_id id

chap chapter

sect section

para paragraph

\u8868\u5c64\u5f62 result of Sudachi

\u54c1\u8a5e result of Sudachi

\u54c1\u8a5e\u7d30\u5206\u985e1 result of Sudachi

\u54c1\u8a5e\u7d30\u5206\u985e2 result of Sudachi

\u54c1\u8a5e\u7d30\u5206\u985e3 result of Sudachi

\u54c1\u8a5e\u7d30\u5206\u985e4 result of Sudachi

\u54c1\u8a5e\u7d30\u5206\u985e5 result of Sudachi

\u539f\u5f62 result of Sudachi

A data frame with 19565 rows and 12 variable:

A data frame with 19526 rows and 12 variable:

Examples

```
data(review_sudachi_a)
review_sudachi_a |>
  unescape_utf()
```

stop_words *Stop words for morphological analysis*

Description

Stop words for morphological analysis

Usage

```
stop_words
```

Format

A data frame with 310 rows and 1 variable:

stop_word Stop words can be used with delete_stop_words(). Escaped by stringi::stri_escape_unicode(). Downloaded from http://svn.sourceforge.jp/svnroot/slothlib/CSharp/Version1/SlothLib/NLP/Filter/StopWord/word/Jap

Examples

```
data(stop_words)
stop_words |>
  unescape_utf()
```

synonym *An example of synonym word pairs*

Description

An example of synonym word pairs

Usage

```
synonym
```

Format

A data frame with 25 rows and 2 variables:

from Words to be replaced from. Escaped by stringi::stri_escape_unicode().

to Words to be replaced to.

Examples

```
data(synonym)
synonym |>
  unescape_utf()
```

text_id_with_break *Add ids.*

Description

Add ids.

Usage

```
text_id_with_break(x, brk, end_with_brk = TRUE)

add_text_id_df(df, col, brk, end_with_brk = TRUE)
```

Arguments

x	A string vector.
brk	A string to specify the break between ids.
end_with_brk	A logical. TRUE: brk means the end of groups. FALSE: brk means the beginning of groups.
df	A dataframe.
col	A string to specify the column.

Value

id_with_break() returns id vector, add_id_df() returns dataframe.

Examples

```
tmp <- c("a", "brk", "b", "brk", "c")
brk <- "brk"
text_id_with_break(tmp, brk)
add_text_id_df(tibble::tibble(tmp), col = "tmp", "brk")
```

unescape_utf *Wrapper functions for escape and unescape unicode*

Description

Wrapper functions for escape and unescape unicode

Usage

```
unescape_utf(x)

escape_utf(x)
```

Arguments

x	A dataframe or character vector

Value

A dataframe or character vector

Examples

```
data(review_mecab)
review_mecab |>
  print() |>
  unescape_utf() |>
  print() |>
  escape_utf()
```

Index