

# Package: RcppJagger (via r-universe)

September 11, 2024

**Title** An R Wrapper for Jagger

**Version** 0.0.2

**Description** A wrapper for Jagger, a morphological analyzer proposed in Yoshinaga (2023) <[arXiv:2305.19045](https://arxiv.org/abs/2305.19045)>. Jagger uses patterns derived from morphological dictionaries and training data sets and applies them from the beginning of the input. This simultaneous and deterministic process enables it to effectively perform tokenization, POS tagging, and lemmatization.

**License** GPL-2

**Depends** R (>= 4.0)

**Encoding** UTF-8

**SystemRequirements** C++17

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3

**URL** <https://shusei-e.github.io/RcppJagger/>,  
<https://www.tkl.iis.u-tokyo.ac.jp/~ynaga/jagger/index.en.html>

**BugReports** <https://github.com/Shusei-E/RcppJagger/issues>

**LinkingTo** Rcpp

**Imports** cli (>= 3.6.1), purrr (>= 1.0.0), Rcpp (>= 1.0.7), rlang (>= 1.1.0)

**Suggests** dplyr (>= 1.1.0), testthat (>= 3.1.5), tibble

**Config/testthat/edition** 3

**LazyData** TRUE

**Repository** <https://shusei-e.r-universe.dev>

**RemoteUrl** <https://github.com/shusei-e/rcppjagger>

**RemoteRef** HEAD

**RemoteSha** b5fd958cfb730267bc00cdf49806212f77d6c2f8

## Contents

lemmatize . . . . .	2
lemmatize_tbl . . . . .	3
pos . . . . .	3
pos_simple . . . . .	4
sentence_example . . . . .	5
tokenize . . . . .	5
tokenize_tbl . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

---

lemmatize	<i>An R wrapper for Jagger's lemmatizer</i>
-----------	---------------------------------------------

---

### Description

An R wrapper for Jagger's lemmatizer

### Usage

```
lemmatize(input, model_path = NULL, keep = NULL, concat = TRUE)
```

### Arguments

input	an input.
model_path	a path to the model.
keep	a vector of POS(s) to keep. Default is NULL.
concat	logical. If TRUE, the function returns a concatenated string. Default is TRUE.

### Value

a vector (if concat = TRUE) or a list (if concat = FALSE).

### Examples

```
data(sentence_example)
res_lemmatize <- lemmatize(sentence_example$text)
```

---

lemmatize_tbl	<i>An R wrapper for Jagger's lemmatizer (a tibble input)</i>
---------------	--------------------------------------------------------------

---

**Description**

An R wrapper for Jagger's lemmatizer (a tibble input)

**Usage**

```
lemmatize_tbl(tbl, column, model_path = NULL, keep = NULL)
```

**Arguments**

tbl	a tibble object.
column	a column name of the tibble to tokenize.
model_path	a path to the model.
keep	a vector of POS(s) to keep. Default is NULL.

**Value**

a tibble.

**Examples**

```
data(sentence_example)
res_lemmatize <- lemmatize_tbl(tibble::as_tibble(sentence_example), "text")
```

---

pos	<i>An R wrapper for Jagger's POS tagger</i>
-----	---------------------------------------------

---

**Description**

An R wrapper for Jagger's POS tagger

**Usage**

```
pos(input, model_path = NULL, keep = NULL, format = c("list", "data.frame"))
```

**Arguments**

input	an input.
model_path	a path to the model.
keep	a vector of POS(s) to keep. Default is NULL.
format	a format of the output. Default is list.

**Value**

a list object.

**Examples**

```
data(sentence_example)
res_pos <- pos(sentence_example$text)
```

---

pos\_simple

*An R wrapper for Jagger's POS tagger (only returning POS)*

---

**Description**

An R wrapper for Jagger's POS tagger (only returning POS)

**Usage**

```
pos_simple(
  input,
  model_path = NULL,
  keep = NULL,
  format = c("list", "data.frame")
)
```

**Arguments**

input	an input.
model_path	a path to the model.
keep	a vector of POS(s) to keep. Default is NULL.
format	a format of the output. Default is list.

**Value**

a list object.

**Examples**

```
data(sentence_example)
res_pos <- pos_simple(sentence_example$text)
```

---

sentence_example	<i>An example sentence</i>
------------------	----------------------------

---

**Description**

An example sentence

**Usage**

```
sentence_example
```

**Format**

A data.frame with a single row and a single column:

**text** a sentence in Japanese

**Source**

Aozora Bunko: <https://www.aozora.gr.jp/>

---

tokenize	<i>An R wrapper for Jagger's tokenizer</i>
----------	--------------------------------------------

---

**Description**

An R wrapper for Jagger's tokenizer

**Usage**

```
tokenize(input, model_path = NULL, keep = NULL, concat = TRUE)
```

**Arguments**

input	an input.
model_path	a path to the model.
keep	a vector of POS(s) to keep. Default is NULL.
concat	logical. If TRUE, the function returns a concatenated string. Default is TRUE.

**Value**

a vector (if concat = TRUE) or a list (if concat = FALSE).

**Examples**

```
data(sentence_example)
res_tokenize <- tokenize(sentence_example$text)
```

---

tokenize_tbl	<i>An R wrapper for Jagger's tokenizer (a tibble input)</i>
--------------	-------------------------------------------------------------

---

**Description**

An R wrapper for Jagger's tokenizer (a tibble input)

**Usage**

```
tokenize_tbl(tbl, column, model_path = NULL, keep = NULL)
```

**Arguments**

tbl	a tibble.
column	a column name of the tibble to tokenize.
model_path	a path to the model.
keep	a vector of POS(s) to keep. Default is NULL.

**Value**

a tibble.

**Examples**

```
data(sentence_example)  
res_tokenize <- tokenize_tbl(tibble::as_tibble(sentence_example), "text")
```

# Index

## \* datasets

sentence\_example, 5

lemmatize, 2

lemmatize\_tbl, 3

pos, 3

pos\_simple, 4

sentence\_example, 5

tokenize, 5

tokenize\_tbl, 6