

Package: ldccr (via r-universe)

September 21, 2026

Title Utilities for Various Japanese Corpora

Version 2026.08.22

Maintainer Akiru Kato <paithiov909@gmail.com>

Description The goal of ldccr package is to make easy to use Japanese language resources. This package provides parsers for several Japanese corpora that are free or open licensed and a downloader of zipped text files published on Aozora Bunko.

License MIT + file LICENSE

URL <https://github.com/paithiov909/ldccr>,
<https://paithiov909.github.io/ldccr/>

BugReports <https://github.com/paithiov909/ldccr/issues>

Depends R (>= 3.5)

Imports dplyr, lifecycle, memoise, purrr (>= 1.0.0), Rcpp,
RcppSimdJson, readr, rlang, stringi, vctrs

Suggests testthat (>= 3.0.0)

LinkingTo Rcpp

Config/roxygen2/version 8.1.0

Config/testthat/edition 3

Encoding UTF-8

LazyData true

LazyDataCompression xz

Roxygen list(markdown = TRUE)

Config/pak/sysreqs libicu-dev libx11-dev

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

Date/Publication 2026-08-22 08:43:39 UTC

RemoteUrl <https://github.com/paithiov909/ldccr>

RemoteRef HEAD

RemoteSha caf70e0a94aa8a8dfb7693f8ad3522906f8a3b5e

Contents

AozoraBunkoSnapshot	2
jrte_rte_files	3
ldnws_categories	3
NekoText	4
parse_jrte_reasoning	4
read_aozora	5
read_ja_text8	5
read_jrte	6
read_ldnws	7
sqids	8
unidic_availables	9
utils	9
Index	10

AozoraBunkoSnapshot	<i>Meta data of text files published on Aozora Bunko</i>
---------------------	--

Description

Meta data of text files published on Aozora Bunko

Usage

AozoraBunkoSnapshot

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 19502 rows and 55 columns.

Source

http://www.aozora.gr.jp/index_pages/list_person_all_extended_utf8.zip

See Also

The structure of the data is described [here](#).

jrte_rte_files	<i>Data for Textual Entailment</i>
----------------	------------------------------------

Description

Data for Textual Entailment

Usage

```
jrte_rte_files(  
  keep = c("rte.nlp2020_base", "rte.nlp2020_append", "rte.lrec2020_surf",  
    "rte.lrec2020_sem_short", "rte.lrec2020_sem_long", "rte.lrec2020_me")  
)
```

Arguments

keep Character vector. File names to parse.

Value

tsv file names.

See Also

Other jrte-reader: [parse_jrte_judges\(\)](#), [parse_jrte_reasoning\(\)](#), [read_jrte\(\)](#)

ldnws_categories	<i>List of categories of the Livedoor News Corpus</i>
------------------	---

Description

List of categories of the Livedoor News Corpus

Usage

```
ldnws_categories(  
  keep = c("dokujo-tsushin", "it-life-hack", "kaden-channel", "livedoor-homme",  
    "movie-enter", "peachy", "smax", "sports-watch", "topic-news")  
)
```

Arguments

keep Category names to parse.

Value

A character vector.

See Also

Other ldnws-reader: [read_ldnws\(\)](#)

NekoText	<i>Whole text of ‘Wagahai Wa Neko Dearu’ written by Natsume Souseki from Aozora Bunko</i>
----------	---

Description

Whole text of ‘Wagahai Wa Neko Dearu’ written by Natsume Souseki from Aozora Bunko

Usage

NekoText

Format

An object of class character of length 2258.

Source

https://www.aozora.gr.jp/cards/000148/files/789_ruby_5639.zip

parse_jrte_reasoning	<i>Parse reasoning column of ‘rte.*.tsv’</i>
----------------------	--

Description

Parse reasoning column of ‘rte.*.tsv’

Usage

parse_jrte_reasoning(tbl)

Arguments

tbl A tibble returned from [read_jrte\(\)](#) of which name is rte.*.tsv.

Value

A tibble.

See Also

Other jrte-reader: [jrte_rte_files\(\)](#), [parse_jrte_judges\(\)](#), [read_jrte\(\)](#)

read_aozora	<i>Download text file from Aozora Bunko</i>
-------------	---

Description

[Superseded] Downloads a file from specified URL, unzips and converts it as UTF-8.

If you want to read a large part of texts published at Aozora Bunko, you can download them at once via [globis-university/aozorabunko-clean](https://github.com/globis-university/aozorabunko-clean).

Usage

```
read_aozora(  
  url = "https://www.aozora.gr.jp/cards/000081/files/472_ruby_654.zip",  
  txtname = NULL,  
  directory = file.path(getwd(), "cache")  
)
```

Arguments

url	URL of text download link.
txtname	New file name as which text is saved. If left with NULL, keeps name of the source file.
directory	Path where new file is saved.

Value

The path to the file downloaded.

read_ja_text8	<i>Read the ja.text8 corpus</i>
---------------	---------------------------------

Description

Downloads and reads the ja.text8 corpus as a tibble.

Usage

```
read_ja_text8(  
  url =  
    "https://s3-ap-northeast-1.amazonaws.com/dev.tech-sketch.jp/chakki/public/ja.text8.zip",  
  size = NULL  
)
```

Arguments

url	String.
size	Integer. If supplied, samples rows by this argument.

Details

By default, this function reads the [ja.text8](#) corpus as a tibble by splitting it into sentences. The ja.text8 as whole corpus consists of over 582,000 sentences, 16,900,026 tokens, and 290,811 vocabularies.

Value

A tibble.

read_jrte	<i>Read the JRTE Corpus</i>
-----------	-----------------------------

Description

Download and read the Japanese Realistic Textual Entailment Corpus. The result of this function is memoised with [memoise::memoise\(\)](#) internally.

Usage

```
read_jrte(
  url = "https://github.com/megagonlabs/jrte-corpus/archive/refs/heads/master.zip",
  exdir = tempdir(),
  keep = jrte_rte_files(),
  keep_rhr = FALSE,
  keep_pn = FALSE
)
```

Arguments

url	String. If left with NULL, the function will skip downloading the file.
exdir	String. Path to temporarily unzip text files.
keep	List. File names to parse and keep in returned value.
keep_rhr	Logical. If supplied TRUE, keeps rhr.tsv.
keep_pn	Logical. If supplied TRUE, keeps pn.tsv.

Value

A list of tibbles.

See Also

Other jrte-reader: [jrte_rte_files\(\)](#), [parse_jrte_judges\(\)](#), [parse_jrte_reasoning\(\)](#)

read_ldnws	<i>Read the Livedoor News Corpus</i>
------------	--------------------------------------

Description

Downloads and reads the Livedoor News Corpus. The result of this function is memoised with `memoise::memoise()` internally.

Usage

```
read_ldnws(  
  url = "https://www.rondhuit.com/download/ldcc-20140209.tar.gz",  
  exdir = tempdir(),  
  keep = ldnws_categories(),  
  collapse = "\n\n",  
  include_title = TRUE  
)
```

Arguments

<code>url</code>	String. If left with NULL, the function will skip downloading the file.
<code>exdir</code>	String. Directory to temporarily untar text files.
<code>keep</code>	Categories to parse and keep in the tibble.
<code>collapse</code>	String with which <code>base::paste()</code> collapses lines.
<code>include_title</code>	Logical. Whether to include title in text body field. Defaults to TRUE.

Details

This function downloads the Livedoor News Corpus and parses it to a tibble. For details about the Livedoor News Corpus, please see [this page](#).

Value

A tibble.

See Also

Other ldnws-reader: [ldnws_categories\(\)](#)

sqids	<i>Generate random-looking IDs from integer ranks</i>
-------	---

Description

`sqids()` is an alternative to `dplyr::row_number()` that generates random-looking IDs from integer ranks using **Sqids** (formerly **Hashids**).

IDs that generated with `sqids()` can be easily decoded back into the original ranks using `unsqids()`.

Usage

```
sqids(
  x,
  .salt = sample.int(1000, 3),
  .ties = c("sequential", "min", "max", "dense")
)

unsqids(x)
```

Arguments

<code>x</code>	For <code>sqids()</code> , a vector to rank. You can leave this argument missing to refer to the "current" row number in 'dplyr' verbs. For <code>unsqids()</code> , a character vector of IDs.
<code>.salt</code>	Integers to use with each value of <code>x</code> to generate IDs.
<code>.ties</code>	Method to rank duplicate values. One of "sequential", "min", "max", or "dense". See <code>ties</code> argument of <code>vctrs::vec_rank()</code> for more details.

Value

For `sqids()`, a character vector of IDs.

For `unsqids()`, integers.

See Also

[sqids/sqids-cpp](#)

Examples

```
ids <- sqids(c(5, 1, 3, 2, 2, NA))
ids
unsqids(ids)

df <- data.frame(
  grp = c(1, 1, 1, 2, 2, 2, 3, 3, 3)
)
# You can use `sqids()` without referencing `x` in dplyr verbs.
```

```
dplyr::mutate(df, sqids = sqids(), row_id = unsqids(sqids))
# Use `.ties` to control how to rank duplicate values.
dplyr::mutate(df, sqids = sqids(grp, .ties = "min"), grp_id = unsqids(sqids))
# When you need to generate the same IDs for each group, fix the `.salt`:
dplyr::mutate(df, sqids = sqids(.salt = 1234L), .by = grp)
```

unidic_availables	<i>Download and unzip 'UniDic'</i>
-------------------	------------------------------------

Description

Downloads 'UniDic' of specified version into dirname. This function is a partial port of [polm/unidic-py](#). Note that to unzip dictionary will take up 770MB on disk after downloading.

Usage

```
unidic_availables()

download_unidic(version = "latest", dirname = "unidic")
```

Arguments

version	String; version of 'UniDic'.
dirname	String; directory to unzip the dictionary.

Value

Full path to dirname is returned invisibly.

utils	<i>Utility functions</i>
-------	--------------------------

Description

[Experimental] These functions are experimental and may withdraw in the future.

Usage

```
clean_url(text, replacement = "")

clean_emoji(text, replacement = "")
```

Arguments

text	A character vector.
replacement	String.

Index

- * **datasets**
 - AozoraBunkoSnapshot, 2
 - NekoText, 4
- * **jrte-reader**
 - jrte_rte_files, 3
 - parse_jrte_reasoning, 4
 - read_jrte, 6
- * **ldnws-reader**
 - ldnws_categories, 3
 - read_ldnws, 7
- AozoraBunkoSnapshot, 2
- base::paste(), 7
- clean_emoji (utils), 9
- clean_url (utils), 9
- download_unidic (unidic_availables), 9
- dplyr::row_number(), 8
- jrte_rte_files, 3
- jrte_rte_files(), 4, 6
- ldnws_categories, 3
- ldnws_categories(), 7
- memoise::memoise(), 6, 7
- NekoText, 4
- parse_jrte_judges(), 3, 4, 6
- parse_jrte_reasoning, 4
- parse_jrte_reasoning(), 3, 6
- read_aozora, 5
- read_ja_text8, 5
- read_jrte, 6
- read_jrte(), 3, 4
- read_ldnws, 7
- read_ldnws(), 4
- sqids, 8
- unidic_availables, 9
- unsqids (sqids), 8
- utils, 9
- vctrs::vec_rank(), 8