Package ‘tipitaka’

February 13, 2026

Type Package

Title Data and Tools for Analyzing the Pali Canon

Version 1.0.0

Description Provides access to the complete Pali Canon, or Tipitaka, the canonical scripture for Theravadin Buddhists worldwide. Based on the Chattha Sangayana Tipitaka version 4 (Vipassana Research Institute, 1990). Includes word frequency data and tools for Pali string sorting. For a lemmatized critical edition with sutta-level granularity, see the companion package 'tipitaka.critical'.

License CC0

Encoding UTF-8

LazyData true

LazyDataCompression xz

RoxygenNote 7.3.3

Depends R (>= 3.5)

Imports stringr

Suggests dplyr, magrittr, stringi

LinkingTo cpp11

NeedsCompilation yes

Author Dan Zigmond [aut, cre]

Maintainer Dan Zigmond <djz@shmonk.com>

Repository CRAN

Date/Publication 2026-02-13 09:50:02 UTC

Contents

`abhidhamma_pitaka` *All the books of the Abhidhamma Pitaka*

Description

A subset of tipitaka_names consisting of only the books of the Abhidhamma Pitaka. These are easier to read if you call `pali_string_fix() first`.

Usage

```
abhidhamma_pitaka
```

Format

A tibble with the variables:

book Abbreviated title

name Full title

\

Examples

```
# Clean up the Unicode characters to make things more readable:
abhidhamma_pitaka$name <-
  stringi::stri_unescape_unicode(abhidhamma_pitaka$name)


# Count all the words in the Abhidhamma Pitaka:
sum(tipitaka_long[tipitaka_long$book %in% abhidhamma_pitaka$book, "n"])
```

pali_alphabet *Pali alphabet in order*

Description

Pali alphabet in order

Usage

```
pali_alphabet
```

Format

The Pali alphabet in traditional order.

Examples

```
# Returns TRUE because a comes before b in Pali:
match("a", pali_alphabet) < match("b", pali_alphabet)
# Returns FALSE beceause c comes before b in Pali
match("b", pali_alphabet) < match("c", pali_alphabet)
```

pali_eq *Equal (==) comparison function for Pali words*

Description

Note that all Pali string comparisons are case-insensitive.

Usage

```
pali_eq(word1, word2)
```

Arguments

word1	A first Pali word as a string
word2	A second Pali word as a string

Value

TRUE if word1 and word2 are the same

pali_gt *Greater-than (>) comparison function for Pali words*

Description

Note that all Pali string comparisons are case-insensitive. #' Also non-Pali characters are placed at the end of the alphabet and are considered equivalent to each other.

Usage

```
pali_gt(word1, word2)
```

Arguments

word1	A first Pali word as a string
word2	A second Pali word as a string

Value

TRUE if word1 comes after word2 alphabetically

pali_lt *Less-than (<) comparison function for Pali words*

Description

Note that all Pali string comparisons are case-insensitive. Also non-Pali characters are placed at the end of the alphabet and are considered equivalent to each other. This has been implemented in C++ for speed.

Usage

```
pali_lt(word1, word2)
```

Arguments

word1	A first Pali word as a string
word2	A second Pali words as a string

Value

TRUE if word1 comes before word2 alphabetically

`pali_sort` *Sorting function for vectors of Pali words.*

Description

Note that all Pali string comparisons are case-insensitive. This algorithm is based on Quicksort, but creates lots of intermediate data structures instead of doing swaps in place. This has been implemented in C++ as the original R version was about 500x slower.

Usage

```
pali_sort(word_list)
```

Arguments

`word_list`	A vector of Pali words

Value

A new vector of Pali words in Pali alphabetical order

Examples

```
# A sorted list of 100 random words from the Tipitaka:
pali_sort(sample(tipitaka_long$word, 100))
```

`pali_stop_words` *Tentative set of "stop words" for Pali*

Description

A list of all declinables and particles from the PTS Pali-English Dictionary.

Usage

```
pali_stop_words
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 245 rows and 1 columns.

Source

https://dsal.uchicago.edu/dictionaries/pali/

Examples

```
# Show top content words in the Tipitaka (excluding stop words)
content_words <- tipitaka_long[!tipitaka_long$word %in% pali_stop_words$word, ]
head(content_words[order(-content_words$n), ], 10)
```

---

sutta_pitaka *All the books of the Sutta Pitaka*

Description

A subset of tipitaka_names consisting of only the books of the Sutta Pitaka. These are easier to read if you call `stringi::stri_unescape_unicode` first.

Usage

```
sutta_pitaka
```

Format

A tibble with the variables:

book Abbreviated title

name Full title

Examples

```
# Clean up the Unicode characters to make things more readable:
sutta_pitaka$name <-
  stringi::stri_unescape_unicode(sutta_pitaka$name)


# Count all the words in the Suttas:
sum(
  unique(
    tipitaka_long[tipitaka_long$book %in% sutta_pitaka$book, "total"]))
```

`tipitaka` *tipitaka: Data and Tools for Analyzing the Pali Canon*

Description

The tipitaka package provides access to the complete Pali Canon, or Tipitaka, from R. The Tipitaka is the canonical scripture for Theravadin Buddhists worldwide. This package includes the VRI (Vipassana Research Institute) Chattha Sangayana edition along with tools for working with Pali text.

Provides access to the complete Pali Canon, or Tipitaka, the canonical scripture for Theravadin Buddhists worldwide. Based on the Chattha Sangayana Tipitaka version 4 (Vipassana Research Institute, 1990). Includes word frequency data and tools for Pali string sorting. For a lemmatized critical edition with sutta-level granularity, see the companion package 'tipitaka.critical'.

Datasets

- tipitaka_raw: the complete text of the Tipitaka (VRI)
- tipitaka_names: the names of each book of the Tipitaka
- sutta_pitaka: the names of each volume of the Sutta Pitaka
- vinaya_pitaka: the names of each volume of the Vinaya Pitaka
- abhidhamma_pitaka: the names of each volume of the Abhidhamma Pitaka
- pali_alphabet: the complete Pali alphabet in traditional order
- pali_stop_words: a set of "stop words" for Pali

Derived Data

These are computed on demand from `tipitaka_raw` on first access:

- tipitaka_long: word frequencies per volume
- tipitaka_wide: word frequency matrix (volumes x words)

Tools

Functions for working with Pali text:

- pali_lt: less-than function for Pali strings
- pali_gt: greater-than function for Pali strings
- pali_eq: equals function for Pali strings
- pali_sort: sorting function for vectors of Pali strings

Related Packages

The companion package **tipitaka.critical** provides a lemmatized critical edition of the complete Tipitaka based on a five-witness collation with sutta-level granularity.

Author(s)

Maintainer: Dan Zigmond <djz@shmonk.com>

tipitaka_long *Tipitaka in "long" form*

Description

Every word of every volume of the Tipitaka, with one word per volume per line. Computed from `tipitaka_raw` on first access.

Usage

```
tipitaka_long
```

Format

A data frame with the variables:

word Pali word

n Number of times this word appears in this book

total Total number of words in this book

freq Frequency with which this word appears in this book

book Abbreviated book name

Source

Vipassana Research Institute, CST4, April 2020

tipitaka_names *Names of each book of the Tipitaka, both abbreviated and in full. These are easier to read if you call `pali_string_fix() first`.*

Description

Names of each book of the Tipitaka, both abbreviated and in full. These are easier to read if you call `pali_string_fix() first`.

Usage

```
tipitaka_names
```

Format

A tibble with the variables:

book Abbreviated title

name Full title

Examples

```
# Clean up the Unicode characters to make things more readable:
tipitaka_names$name <-
  stringi::stri_unescape_unicode(tipitaka_names$name)
```

`tipitaka_raw` *Tipitaka text in raw form*

Description

The unprocessed text of the Tipitaka, with one row per volume.

Usage

```
tipitaka_raw
```

Format

A tibble with the variables:

text Text of each Tipitaka volume

book Abbreviated book name of each volume

Source

Vipassana Research Institute, CST4, April 2020

`tipitaka_wide` *Tipitaka in "wide" form*

Description

Every word of every volume of the Tipitaka, with one word per column and one book per line. Each cell is the frequency at which that word appears in that book. Computed from `tipitaka_raw` on first access.

Usage

```
tipitaka_wide
```

Format

An object of class `data.frame` with 46 rows and 140433 columns.

Source

Vipassana Research Institute, CST4, April 2020

`vinaya_pitaka` *All the books of the Vinaya Pitaka*

Description

A subset of tipitaka_names consisting of only the books of the Vinaya Pitaka. These are easier to read if you call `stringi::stri_unescape_unicode` first.

Usage

```
vinaya_pitaka
```

Format

A tibble with the variables:

book Abbreviated title

name Full title

Examples

```
# Clean up the Unicode characters to make things more readable:
vinaya_pitaka$name <-
  stringi::stri_unescape_unicode(vinaya_pitaka$name)


# Count all the words in the Vinaya Pitaka:
sum(tipitaka_long[tipitaka_long$book %in% vinaya_pitaka$book, "n"])
```

Index