Package ‘rbm25’

January 16, 2025

Title A Light Wrapper Around the 'BM25' 'Rust' Crate for Okapi BM25 Text Search

Version 0.0.3

Description BM25 is a ranking function used by search engines to rank matching documents according to their relevance to a user's search query.
This package provides a light wrapper around the 'BM25' 'rust' crate for Okapi BM25 text search.
For more information, see Robertson et al. (1994) <https://trec.nist.gov/pubs/trec3/t3_proceedings.html>.

Encoding UTF-8

URL https://davzim.github.io/rbm25/, https://github.com/DavZim/rbm25/

BugReports https://github.com/DavZim/rbm25/issues

SystemRequirements Cargo (Rust's package manager), rustc >= 1.71.1

Imports R6

Suggests testthat (>= 3.0.0)

License MIT + file LICENSE

RoxygenNote 7.3.2

Config/rextendr/version 0.3.1.9001

Config/testthat/edition 3

Config/rbm25/MSRV 1.71.1

NeedsCompilation yes

Author David Zimmermann-Kollenda [aut, cre],
Michael Barlow [aut] (bm25 Rust library),
Authors of the dependency Rust crates [aut] (see AUTHORS file)

Maintainer David Zimmermann-Kollenda <david_j_zimmermann@hotmail.com>

Repository CRAN

Date/Publication 2025-01-16 10:40:02 UTC

Contents

BM25 *BM25 Object*

Description

Class to construct the BM25 search object

Methods

Public methods:

- `BM25$new()`
- `BM25$available_languages()`
- `BM25$get_data()`
- `BM25$get_lang()`
- `BM25$print()`
- `BM25$add_data()`
- `BM25$query()`
- `BM25$clone()`

Method `new()`**:** Creates a new instance of a BM25 class

Usage:

```
BM25$new(data = NULL, lang = "detect", k1 = 1.2, b = 0.75, metadata = NULL)
```

Arguments:

`data` text data, a vector of strings. Note any preprocessing steps (tolower, removing stopwords etc) need to have taken place before this!

`lang` language of the data, see self$available_languages(), can also be "detect" to automatically detect the language, default is "detect"

`k1` k1 parameter of BM25, default is 1.2

`b` b parameter of BM25, default is 0.75

`metadata` a data.frame with metadata for each document, default is NULL must be a data.frame with the same number of rows containing arbitrary metadata for each document, e.g. a file path or a URL

Returns: BM25 object

Examples:

```
corpus <- c(
 "The rabbit munched the orange carrot.",
 "The snake hugged the green lizard.",
 "The hedgehog impaled the orange orange.",
 "The squirrel buried the brown nut."
)
bm25 <- BM25$new(data = corpus, lang = "en",
                 metadata = data.frame(src = paste("file", 1:4)))
```

```
bm25
bm25$get_data()

bm25$query("orange", max_n = 2)
bm25$query("orange", max_n = 3)
bm25$query("orange") # return all, same as max_n = Inf or NULL
```

Method `available_languages()`**:** Returns the available languages

Usage:

```
BM25$available_languages()
```

Returns: a named character vector with language codes and their full names

Examples:

```
BM25$new()$available_languages()
```

Method `get_data()`**:** Returns the data

Usage:

```
BM25$get_data(add_metadata = TRUE)
```

Arguments:

`add_metadata` whether to add metadata to the data, default is TRUE

Returns: a data.frame with the data and metadata if available and selected

Examples:

```
BM25$new(data = letters, metadata = LETTERS)$get_data()
```

Method `get_lang()`**:** Returns the language used

Usage:

```
BM25$get_lang()
```

Returns: a character string with the language code

Examples:

```
BM25$new()$get_lang()
BM25$new(lang = "en")$get_lang()
BM25$new(lang = "detect")$get_lang()
```

Method `print()`**:** Prints a BM25 object

Usage:

```
BM25$print(n = 5, nchar = 20)
```

Arguments:

`n` number of data to print, default is 5

`nchar` number of characters to print for each text, default is 20

Returns: the object invisible

Examples:

```
BM25$new(data = letters, metadata = LETTERS)
```

Method `add_data()`**:** Adds data to the BM25 object

This can be useful to add more data later on, note this will rebuild the engine.

Usage:

```
BM25$add_data(data, metadata = NULL)
```

Arguments:

`data` a vector of strings

`metadata` a data.frame with metadata for each document, default is NULL

Returns: NULL

Examples:

```
bm25 <- BM25$new()
bm25$add_data(letters, metadata = LETTERS)
bm25
```

Method `query()`**:** Query the BM25 object for the N best matches

Usage:

```
BM25$query(query, max_n = NULL, return_text = TRUE, return_metadata = TRUE)
```

Arguments:

`query` the term to search for, note all preprocessing that was applied to the text corpus initially needs to be already performed on the term, e.g., tolower, removing stopwords etc

`max_n` the maximum number of results to return, default is all

`return_text` whether to return the text, default is TRUE

`return_metadata` whether to return metadata, default is TRUE

Returns: a data.frame with the results

Examples:

```
corpus <- c(
 "The rabbit munched the orange carrot.",
 "The snake hugged the green lizard.",
 "The hedgehog impaled the orange orange.",
 "The squirrel buried the brown nut."
)
bm25 <- BM25$new(data = corpus, lang = "en",
                 metadata = data.frame(src = paste("file", 1:4)))

bm25$query("orange", max_n = 2)
bm25$query("orange", max_n = 3)
bm25$query("orange", return_text = FALSE, return_metadata = FALSE)
bm25$query("orange", max_n = 3)
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
BM25$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
corpus <- c(
  "The rabbit munched the orange carrot.",
  "The snake hugged the green lizard.",
  "The hedgehog impaled the orange orange.",
  "The squirrel buried the brown nut."
)
bm25 <- BM25$new(data = corpus, lang = "en",
                 metadata = data.frame(src = paste("file", 1:4)))
bm25$query("orange", max_n = 2)
bm25$query("orange")

## ------------------------------------------------
## Method `BM25$new`
## ------------------------------------------------

corpus <- c(
 "The rabbit munched the orange carrot.",
 "The snake hugged the green lizard.",
 "The hedgehog impaled the orange orange.",
 "The squirrel buried the brown nut."
)
bm25 <- BM25$new(data = corpus, lang = "en",
                 metadata = data.frame(src = paste("file", 1:4)))
bm25
bm25$get_data()

bm25$query("orange", max_n = 2)
bm25$query("orange", max_n = 3)
bm25$query("orange") # return all, same as max_n = Inf or NULL

## ------------------------------------------------
## Method `BM25$available_languages`
## ------------------------------------------------

BM25$new()$available_languages()

## ------------------------------------------------
## Method `BM25$get_data`
## ------------------------------------------------

BM25$new(data = letters, metadata = LETTERS)$get_data()

## ------------------------------------------------
## Method `BM25$get_lang`
## ------------------------------------------------

BM25$new()$get_lang()
BM25$new(lang = "en")$get_lang()
BM25$new(lang = "detect")$get_lang()

## ------------------------------------------------
## Method `BM25$print`
## ------------------------------------------------

BM25$new(data = letters, metadata = LETTERS)

## ------------------------------------------------
## Method `BM25$add_data`
## ------------------------------------------------

bm25 <- BM25$new()
bm25$add_data(letters, metadata = LETTERS)
bm25

## ------------------------------------------------
## Method `BM25$query`
## ------------------------------------------------

corpus <- c(
 "The rabbit munched the orange carrot.",
 "The snake hugged the green lizard.",
 "The hedgehog impaled the orange orange.",
 "The squirrel buried the brown nut."
)
bm25 <- BM25$new(data = corpus, lang = "en",
                 metadata = data.frame(src = paste("file", 1:4)))

bm25$query("orange", max_n = 2)
bm25$query("orange", max_n = 3)
bm25$query("orange", return_text = FALSE, return_metadata = FALSE)
bm25$query("orange", max_n = 3)
```

bm25_score — *Score a text corpus based on the Okapi BM25 algorithm*

Description

A simple wrapper around the BM25 class.

Usage

```
bm25_score(data, query, lang = NULL, k1 = 1.2, b = 0.75)
```

Arguments

`data` text data, a vector of strings. Note any preprocessing steps (tolower, removing stopwords etc) need to have taken place before this!

`query` the term to search for, note all preprocessing that was applied to the text corpus initially needs to be already performed on the term, e.g., tolower, removing stopwords etc `lang` language of the data, see self$available_languages(), can also be "detect" to automatically detect the language, default is "detect"

`k1` k1 parameter of BM25, default is 1.2

`b` b parameter of BM25, default is 0.75

Value

a numeric vector of the BM25 scores, note higher values are showing a higher relevance of the text to the query

See Also

BM25

Examples

```
corpus <- c(
 "The rabbit munched the orange carrot.",
 "The snake hugged the green lizard.",
 "The hedgehog impaled the orange orange.",
 "The squirrel buried the brown nut."
)
scores <- bm25_score(data = corpus, query = "orange")
data.frame(text = corpus, scores_orange = scores)
```

Index