Package 'corporaexplorer'

November 15, 2019

Type Package

Title A 'Shiny' App for Exploration of Text Collections

Version 0.7.0

Description Facilitates dynamic exploration of text collections through an intuitive graphical user interface. The package contains 1) a helper function to convert a data frame to a 'corporaexplorerobject', 2) a 'Shiny' app for fast and flexible exploration of a 'corporaexplorerobject', and 3) a 'Shiny' app for simple retrieval/extraction of documents from a 'corporaexplorerobject' in a reading-friendly format.

Depends R (>= 3.0.0)

Imports data.table, dplyr, forcats, ggplot2, lubridate, magrittr, padr, plyr, RColorBrewer, re2r, rlang, rmarkdown, scales, shiny, shinycssloaders, shinydashboard, shinyjs, shinyWidgets, stringi, stringr, tibble, tidyr, zoo

Suggests shinytest, testthat

License GPL-3 | file LICENSE

Date/Publication 2019-11-15 10:50:02 UTC

LazyData true

RoxygenNote 7.0.0

Encoding UTF-8

URL https://kgjerde.github.io/corporaexplorer, https://github.com/kgjerde/corporaexplorer

BugReports https://github.com/kgjerde/corporaexplorer/issues

NeedsCompilation no

Author Kristian Lundby Gjerde [aut, cre]

Maintainer Kristian Lundby Gjerde <kristian.gjerde@gmail.com>

Repository CRAN

R topics documented:

`prepare_data` *Prepare data for corpus exploration*

Description

Prepare data for corpus exploration

Usage

```
prepare_data(dataset, ...)

## S3 method for class 'data.frame'
prepare_data(
  dataset,
  date_based_corpus = TRUE,
  grouping_variable = NULL,
  columns_doc_info = c("Date", "Title", "URL"),
  corpus_name = NULL,
  use_matrix = TRUE,
  normalise = TRUE,
  matrix_without_punctuation = TRUE,
  ...
)
```

Arguments

`dataset`	Object to be converted to a corporaexplorerobject. Converts a data frame with a column "Text" (class character), and optionally other columns. If `date_based_corpus` is `TRUE` (the default), `dataset` must contain a column "Date" (of class Date).
`...`	Ignored.
`date_based_corpus`	Logical. Set to `FALSE` if the corpus is not to be organised according to document dates.
`grouping_variable`	Character string. If `date_based_corpus` is `TRUE`, this argument is ignored. If `date_based_corpus` is `FALSE`, this argument can be used to group the documents, e.g. if `dataset` is organised by chapters belonging to different books.

`columns_doc_info`
: Character vector. The columns from `dataset` to display in the "document information" tab in the corpus exploration app. By default "Date", "Title" and "URL" will be displayed, if included. If `columns_doc_info` includes a column which is not present in dataset, it will be ignored.

`corpus_name`
: Character string with name of corpus.

`use_matrix`
: Logical. Should the function create a document term matrix for fast searching? If `TRUE`, data preparation will run longer and demand more memory. If `FALSE`, the returning corporaexplorerobject will be more light-weight, but searching will be slower.

`normalise`
: Should non-breaking spaces (U+00A0) and soft hyphens (U+00ad) be normalised?

`matrix_without_punctuation`
: Should punctuation and digits be stripped from the text before constructing the document term matrix? If `TRUE`, the default:

  - The corporaexplorer object will be lighter and most searches in the corpus exploration app will be faster.
  - Searches including punctuation and digits will be carried out in the full text documents.
  - The only "risk" with this strategy is that the corpus exploration app in some cases can produce false positives. E.g. searching for the term "donkey" will also find the term "don%key". This should not be a problem for the vast opportunity of use cases, but if one so desires, there are three different solutions: set this parameter to `FALSE`, create a corporaexplorerobject without a matrix by setting the `use_matrix` parameter to `FALSE`, or run `run_corpus_explorer` with the `use_matrix` parameter set to `FALSE`.

  If `FALSE`, the corporaexplorer object will be larger, and most simple searches will be slower.

Details

Each row in `dataset` is treated as a base differentiating unit in the corpus, typically chapters in books, or a single document in document collections.

The following column names are reserved and cannot be used in `dataset`: "ID", "Text_original_case", "Tile_length", "Year", "Seq", "Weekday_n", "Day_without_docs", "Invisible_fake_date", "Tile_length".

Value

A `corporaexplorer` object to be passed as argument to `run_corpus_explorer` and `run_document_extractor`.

Examples

```
# Constructing test data frame:
dates <- as.Date(paste(2011:2020, 1:10, 21:30, sep = "-"))
texts <- paste0(
  "This is a document about ", month.name[1:10], ". ",
  "This is not a document about ", rev(month.name[1:10]), "."
)
titles <- paste("Text", 1:10)
```

```
test_df <- tibble::tibble(Date = dates, Text = texts, Title = titles)

# Converting to corporaexplorer object:
corpus <- prepare_data(test_df, corpus_name = "Test corpus")

if(interactive()){
# Running exploration app:
run_corpus_explorer(corpus)

# Running app to extract documents:
run_document_extractor(corpus)
}
```

`prepare_data.character` *Quickly explore character vector*

Description

Quick convertion of character vector to simple corporaexplorerobject with no metadata.

Usage

```
## S3 method for class 'character'
prepare_data(dataset, ...)
```

Arguments

`dataset`	A non-empty character vector.
`...`	Other arguments to be passed to `prepare_data`.

Value

A corporaexplorerobject.

Examples

```
alphabet_corpus <- prepare_data(LETTERS)

if(interactive()){
# Running exploration app:
run_corpus_explorer(alphabet_corpus)
}
```

`run_corpus_explorer` *Launch Shiny app for exploration of text collection*

Description

Launch Shiny app for exploration of text collection. Interrupt R to stop the application (usually by pressing Ctrl+C or Esc).

Usage

```
run_corpus_explorer(
  corpus_object,
  search_options = list(),
  ui_options = list(),
  search_input = list(),
  plot_options = list(),
  ...
)
```

Arguments

`corpus_object` A corporaexplorerobject created by `prepare_data`.

`search_options` List. Specify how search operations in the app are carried out. Available options:

- `use_matrix` Logical. If the corporaexplorerobject contains a document term matrix, should it be used for searches? (See `prepare_data`.) Defaults to `TRUE`.
- `regex_engine` Character. Specify regular expression engine to be used (defaults to `"default"`). Available options:
  - "default": use the `re2r` package (`https://github.com/qinwf/re2r`) for simple searches and the `stringr` package (`https://github.com/tidyverse/stringr` for complex regexes (i.e. when special regex characters are used).
  - "stringr": use `stringr` for all searches.
  - "re2r": use `re2r` for all searches.
- `optional_info` Logical. If `TRUE`, information about search method (regex engine and whether the search was conducted in the document term matrix or in the full text documents).
- `allow_unreasonable_patterns` Logical. If `FALSE`, the default, the app will not allow patterns that will result in an enormous amount of hits or will lead to a very slow search. (Examples of such patterns will include '.' and '\b'.)

`ui_options` List. Specify custom app settings (see example below). Currently available:

- `font_size`. Character string specifying font size in document view, e.g. `"10px"`

`search_input` List. Gives the opportunity to pre-populate the following sidebar fields (see example below):

- `search_terms`: The 'Term(s) to chart and highlight' field. Character vector with maximum length 5.
- `highlight_terms`: The 'Additional terms for text highlighting' field. Character vector.
- `filter_terms`: The 'Filter corpus?' field. Character vector.
- `case_sensitivity`: Should the 'Case sensitive search' box be checked? Logical.

`plot_options` List. Specify custom plot settings (see example below). Currently available:

- `max_docs_in_wall_view`. Integer specifying the maximum number of documents to be rendered in the 'document wall' view. Default value is 12000.
- `plot_size_factor`. Numeric. Tweaks the corpus map plot's height. Value > 1 increases height, value < 1 decreases height. Ignored if value <= 0.
- `documents_per_row_factor`. Numeric. Tweaks the number of documents included in each row in 'document wall' view. Value > 1 increases number of documents, value < 1 decreases number of documents. Ignored if value <= 0.
- `document_tiles`. Integer specifying the number of tiles used in the tile chart representing occurences of terms in document. Ignored if value < 1 or if value > 50.
- `colours`. Character vector of length 1 to 6. Specify the order of the colours used to represent search (and highlight) terms in plots and documents. The default order and available colours are defined by the character vector `c("red","blue","green","purple","orange","gray")`. Passing e.g. `plot_options = list(colours = c("gray","green"))` will change that order to `c("gray","green","red","blue","purple","orange")`. Arguments with duplicated colours or with colours not present in the default character vector will be ignored.

`...` Other arguments passed to `runApp` in the Shiny package.

Examples

```
# Constructing test data frame:
dates <- as.Date(paste(2011:2020, 1:10, 21:30, sep = "-"))
texts <- paste0(
  "This is a document about ", month.name[1:10], ". ",
  "This is not a document about ", rev(month.name[1:10]), "."
)
titles <- paste("Text", 1:10)
test_df <- tibble::tibble(Date = dates, Text = texts, Title = titles)

# Converting to corporaexplorerobject:
corpus <- prepare_data(test_df, corpus_name = "Test corpus")

if(interactive()){
```

```
# Running exploration app:
run_corpus_explorer(corpus)
run_corpus_explorer(corpus,
                    search_options = list(optional_info = TRUE),
                    ui_options = list(font_size = "10px"),
                    search_input = list(search_terms = c("Tottenham", "Spurs")),
                    plot_options = list(MAX_DOCS_IN_WALL_VIEW = 12001,
                                        colours = c("gray", "green")))

# Running app to extract documents:
run_document_extractor(corpus)
}
```

`run_document_extractor` *Launch Shiny app for retrieval of documents from text collection*

Description

Shiny app for simple retrieval/extraction of documents from a "corporaexplorerobject" in a reading-friendly format. Interrupt R to stop the application (usually by pressing Ctrl+C or Esc).

Usage

```
run_document_extractor(corpus_object, max_html_docs = 400, ...)
```

Arguments

`corpus_object`	A `corporaexplorer` object created by `prepare_data`.
`max_html_docs`	The maximum number of documents allowed in one HTML report.
`...`	Other arguments passed to `runApp` in the Shiny package.

Examples

```
# Constructing test data frame:
dates <- as.Date(paste(2011:2020, 1:10, 21:30, sep = "-"))
texts <- paste0(
  "This is a document about ", month.name[1:10], ". ",
  "This is not a document about ", rev(month.name[1:10]), "."
)
titles <- paste("Text", 1:10)
test_df <- tibble::tibble(Date = dates, Text = texts, Title = titles)

# Converting to corporaexplorer object:
corpus <- prepare_data(test_df, corpus_name = "Test corpus")
if(interactive()){
# Running exploration app:
run_corpus_explorer(corpus)
```

```
# Running app to extract documents:
run_document_extractor(corpus)
}
```

`test_data` *A tiny test dataset to test basic functionality*

Description

Created by `corporaexplorer:::create_test_data()`.

Usage

```
test_data
```

Format

A corporaexplorerobject.

Index