Package ‘corporaexplorer’

March 18, 2021

Type Package

Title A 'Shiny' App for Exploration of Text Collections

Version 0.8.4

Description Facilitates dynamic exploration of text collections through an intuitive graphical user interface and the power of regular expressions. 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. The package also includes demo apps with which one can explore Jane Austen's novels and the State of the Union Addresses (data from the 'janeaustenr' and 'sotu' packages respectively).

Depends R (>= 3.0.0)

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

Suggests janeaustenr, shinytest, sotu, testthat

License GPL-3 | file LICENSE

Date/Publication 2021-03-18 02:40:02 UTC

LazyData true

RoxygenNote 7.1.1

Encoding UTF-8


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
## Demo: Jane Austen’s novels

**Description**

`run_janeausten_app()` is a convenience function to directly run the demo app without first creating a `corporaexplorer` object. Equals `explore(create_janeausten_app())`. Interrupt R to stop the application (usually by pressing Ctrl+C or Esc).

**Usage**

```r
run_janeausten_app(...) 
create_janeausten_app()
```

**Arguments**

... Arguments passed to `explore()`

**Details**

The demo app’s data are Jane Austen’s six novels, retrieved through the "janeaustenr" package (https://github.com/juliasilge/janeaustenr) – which must be installed for these functions to work – and converted to a `corporaexplorer` object as shown at https://kgjerde.github.io/corporaexplorer/articles/jane_austen.html.

**Value**

`run_janeausten_app()` launches a Shiny app. `create_janeausten_app()` returns a `corporaexplorer` object.

**Examples**

```r
## Create corporaexplorer object for demo app:
jane_austen <- create_janeausten_app()

if(interactive()){
```
## Run the corporaexplorer object:
explore(jane_austen)

## Or create and run the demo app in one step:
run_janeausten_app()
**explore**

*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).

`explore()` explores a 'corporaexplorerobject' created with the `prepare_data()` function. App settings optionally specified in the arguments to `explore()`.

`explore0()` is a convenience function to directly explore a data frame or character vector without first creating a corporaexplorerobject using `prepare_data()`, instead creating one on the fly as the app launches. Functionally equivalent to `explore(prepare_data(dataset,use_matrix = FALSE))`.

**Usage**

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

```
explore0(
  dataset,
  arguments_prepare_data = list(use_matrix = FALSE),
  arguments_explore = 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.
  - **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 occurrences 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.
  • tile_length. Either "scaled" or "uniform". With "scaled", the default, the length of the tiles in document wall view and day corpus view will vary according to length of document (see the tile_length_range argument in prepare_data()). If "uniform", all tiles will be of equal length.

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

dataset Data frame or character vector as specified in prepare_data()

arguments_prepare_data List. Arguments to be passed to prepare_data() in order to override this function’s default argument values.

arguments_explore List. Arguments to be passed to explore() in order to override this function’s default argument values.
Details

For `explore0()`: by default, no document term matrix will be generated, meaning that the data will be prepared for exploration faster than by using the default settings in `prepare_data()`, but also that searches in the app are likely to be slower.

Value

Launches a Shiny app.

Examples

```r
# 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:
explore(corpus)
explore(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)
}
if (interactive()) {

explore0(rep(sample(LETTERS), 10))
}
explore0(rep(sample(LETTERS), 10),
  arguments_explore = list(search_input = list(search_terms = "Z")))
}
```
Prepare data for corpus exploration

Description

Convert data frame or character vector to a ‘corporaexplorerobject’ for subsequent exploration.

Usage

prepare_data(dataset, ...)

## S3 method for class 'data.frame'
prepare_data(
  dataset,
  date_based_corpus = TRUE,
  grouping_variable = NULL,
  within_group_identifier = "Seq",
  columns_doc_info = c("Date", "Title", "URL"),
  corpus_name = NULL,
  use_matrix = TRUE,
  matrix_without_punctuation = TRUE,
  tile_length_range = c(1, 10),
  columns_for_ui_checkboxes = NULL,
  ...
)

## S3 method for class 'character'
prepare_data(
  dataset,
  corpus_name = NULL,
  use_matrix = TRUE,
  matrix_without_punctuation = TRUE,
  ...
)

Arguments

dataset  Object to convert to corporaexplorerobject:
  - A data frame with a column "Text" (class character), and optionally other
    columns. If date_based_corpus is TRUE (the default), dataset must con-
    tain a column "Date" (of class Date).
  - Or a non-empty character vector.

... Other arguments to be passed to prepare_data.
date_based_corpus Logical. Set to FALSE if the corpus is not to be organised according to document dates.
prepare_data

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 docu-
ments, e.g. if dataset is organised by chapters belonging to different books.

within_group_identifier
Character string indicating column name in dataset. "Seq", the default, means
the rows in each group are assigned a numeric sequence 1:n where n is the
number of rows in the group. Used in document tab title in non-date based
corpora. If date_based_corpus is TRUE, this argument is ignored.

columns_doc_info
Character vector. The columns from dataset to display in the "document informa-
tion" 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 corporaexplorer object will be more light-weight, but searching will
be slower.

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 so-
lutions: set this parameter to FALSE, create a corporaexplorer object without
a matrix by setting the use_matrix parameter to FALSE, or run explore
with the use_matrix parameter set to FALSE.

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

tile_length_range
Numeric vector of length two. Fine-tune the tile lengths in document wall and
day corpus view. Tile length is calculated by scales::rescale(nchar(dataset$Text),to
= tile_length_range,from = c(0,max(.))) Default is c(1,10).

columns_for_ui_checkboxes
Character. Character or factor column(s) in dataset. Include sets of check-
boxes in the app sidebar for convenient filtering of corpus. Typical useful for
columns with a small set of unique (and short) values. Checkboxes will be ar-
ranged by sort(), unless columns_for_ui_checkboxes is a vector of factors,
in which case the order will be according to factor level order (easy releveiling
with forcats::fct_relevel()). To use a different label in the sidebar than the
columnn name, simply pass a named character vector to columns_for_ui_checkboxes.
If `columns_for_ui_checkboxes` includes a column which is not present in `dataset`, it will be ignored.

**Details**

For `data.frame`: 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".

A character vector will be converted to a simple `corporaexplorer` object with no metadata.

**Value**

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

**Examples**

```r
## From data.frame
# 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:
    explore(corpus)

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

## From character vector
alphabet_corpus <- prepare_data(LETTERS)

if(interactive()){
    # Running exploration app:
    explore(alphabet_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:
  explore(corpus)

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

---

test_data  

* A tiny test dataset to test basic functionality

---

Description

Created by corporaexplorer::create_test_data().

Usage

test_data

Format

A corporaexplorerobject.
Index

* datasets
  test_data, 11

create_janeausten_app
  (demo_jane_austen), 2
create_sotu_app (demo_sotu), 3
create_sotu_decade_app (demo_sotu), 3

demo_jane_austen, 2
demo_sotu, 3

explore, 4, 8, 9
explore0 (explore), 4

prepare_data, 4, 7, 10

run_document_extractor, 9, 10
run_janeausten_app (demo_jane_austen), 2
run_sotu_app (demo_sotu), 3
run_sotu_decade_app (demo_sotu), 3
runApp, 5, 10

test_data, 11