Package: pdfsearch (via r-universe)

October 30, 2024

Type Package

Version 0.3.4

License MIT + file LICENSE

Title Search Tools for PDF Files

Description Includes functions for keyword search of pdf files. There is also a wrapper that includes searching of all files within a single directory.

Depends R (>= 3.3.0)

Imports pdftools, tibble, tokenizers, stringi

Suggests shiny, testthat, knitr, rmarkdown, covr

Maintainer Brandon LeBeau <lebebr01+pdfsearch@gmail.com>

RoxygenNote 7.1.2

URL https://github.com/lebebr01/pdfsearch

BugReports https://github.com/lebebr01/pdfsearch/issues

VignetteBuilder knitr

Encoding UTF-8

Repository https://lebebr01.r-universe.dev

RemoteUrl https://github.com/lebebr01/pdfsearch

RemoteRef HEAD

RemoteSha 5b4d3f28354c327bd2aa4c66f53e47c8c9b29f4c

Contents

convert_tokens *Ability to tokenize words.*

Description

Ability to tokenize words.

Usage

```
convert_tokens(
  x,
  path = FALSE,
  split_pdf = FALSE,
  remove_hyphen = TRUE,
  token_function = NULL
)
```

Arguments

x	The text of the pdf file. This can be specified directly or the pdftools package is used to read the pdf file from a file path. To use the pdftools, the path argument must be set to TRUE.
path	An optional path designation for the location of the pdf to be converted to text. The pdftools package is used for this conversion.
split_pdf	TRUE/FALSE indicating whether to split the pdf using white space. This would be most useful with multicolumn pdf files. The split_pdf function attempts to recreate the column layout of the text into a single column starting with the left column and proceeding to the right.
remove_hyphen	TRUE/FALSE indicating whether hyphenated words should be adjusted to combine onto a single line. Default is TRUE.
token_function	This is a function from the tokenizers package. Default is the tokenize_words function.

Value

A list of character vectors containing the tokens. More detail can be found looking at the documentation of the tokenizers package.

Examples

```
file <- system.file('pdf', '1610.00147.pdf', package = 'pdfsearch')
convert_tokens(file, path = TRUE)
```

format_text *Format PDF input text*

Description

Performs some formatting of pdf text upon import.

Usage

```
format_text(
  pdf_text,
  split_pdf = FALSE,
  remove_hyphen = TRUE,
  convert_sentence = TRUE,
  remove_equations = FALSE,
  split_pattern = "\\p{WHITE_SPACE}{3,}",
  ...
)
```

Arguments

pdf_text	A list of text from PDF import, most likely from 'pdftools::pdf_text()'. Each element of the list is a unique page of text from the PDF.
split_pdf	TRUE/FALSE indicating whether to split the pdf using white space. This would be most useful with multicolumn pdf files. The split_pdf function attempts to recreate the column layout of the text into a single column starting with the left column and proceeding to the right.
remove_hyphen	TRUE/FALSE indicating whether hyphenated words should be adjusted to combine onto a single line. Default is TRUE.
convert_sentence	TRUE/FALSE indicating if individual lines of PDF file should be collapsed into a single large paragraph to perform keyword searching. Default is TRUE
remove_equations	TRUE/FALSE indicating if equations should be removed. Default behavior is to search for the following regex: "\([0-9]1,\)$", essentially this matches a literal parenthesis, followed by at least one number followed by another parenthesis at the end of the text line. This will not detect other patterns or detect the entire equation if it is a multi-row equation.
split_pattern	Regular expression pattern used to split multicolumn PDF files using `stringi::stri_split_regex`. Default pattern is "\pWHITE_SPACE3," which can be interpreted as: split based on three or more consecutive white space characters.
...	Additional arguments, currently not used.

heading_search *Function to locate sections of pdf*

Description

The ability to extract the location of the text and separate by sections. The function will return the headings with their location in the pdf.

Usage

```
heading_search(
  x,
  headings,
  path = FALSE,
  pdf_toc = FALSE,
  full_line = FALSE,
  ignore_case = FALSE,
  split_pdf = FALSE,
  convert_sentence = FALSE
)
```

Arguments

x	Either the text of the pdf read in with the pdftools package or a path for the location of the pdf file.
headings	A character vector representing the headings to search for. Can be NULL if pdf_toc = TRUE.
path	An optional path designation for the location of the pdf to be converted to text. The pdftools package is used for this conversion.
pdf_toc	TRUE/FALSE whether the pdf_toc function should be used from the pdftools package. This is most useful if the pdf has the table of contents embedded within the pdf. Must specify path = TRUE if pdf_toc = TRUE.
full_line	TRUE/FALSE indicating whether the headings should reside on their own line. This can create problems with multiple column pdfs.
ignore_case	TRUE/FALSE/vector of TRUE/FALSE, indicating whether the case of the keyword matters. Default is FALSE meaning that case of the headings keywords are literal. If a vector, must be same length as the headings vector.
split_pdf	TRUE/FALSE indicating whether to split the pdf using white space. This would be most useful with multicolumn pdf files. The split_pdf function attempts to recreate the column layout of the text into a single column starting with the left column and proceeding to the right.
convert_sentence	TRUE/FALSE indicating if individual lines of PDF file should be collapsed into a single large paragraph to perform keyword searching. Default is FALSE

Examples

```
file <- system.file('pdf', '1501.00450.pdf', package = 'pdfsearch')

heading_search(file, headings = c('abstract', 'introduction'),
  path = TRUE)
```

`keyword_directory` *Wrapper for keyword search function*

Description

This will use the keyword_search function to loop over all pdf files in a directory. Includes the ability to include subdirectories as well.

Usage

```
keyword_directory(
  directory,
  keyword,
  surround_lines = FALSE,
  ignore_case = FALSE,
  token_results = TRUE,
  split_pdf = FALSE,
  remove_hyphen = TRUE,
  convert_sentence = TRUE,
  remove_equations = TRUE,
  split_pattern = "\\p{WHITE_SPACE}{3,}",
  full_names = TRUE,
  file_pattern = ".pdf",
  recursive = FALSE,
  max_search = NULL,
  ...
)
```

Arguments

`directory`	The directory to perform the search for pdf files to search.
`keyword`	The keyword(s) to be used to search in the text. Multiple keywords can be specified with a character vector.
`surround_lines`	numeric/FALSE indicating whether the output should extract the surrouding lines of text in addition to the matching line. Default is FALSE, if not false, include a numeric number that indicates the additional number of surrounding lines that will be extracted.
`ignore_case`	TRUE/FALSE/vector of TRUE/FALSE, indicating whether the case of the keyword matters. Default is FALSE meaning that case of the keyword is literal. If a vector, must be same length as the keyword vector.

token_results TRUE/FALSE indicating whether the results text returned should be split into tokens. See the tokenizers package and convert_tokens for more details. Defaults to TRUE.

split_pdf TRUE/FALSE indicating whether to split the pdf using white space. This would be most useful with multicolumn pdf files. The split_pdf function attempts to recreate the column layout of the text into a single column starting with the left column and proceeding to the right.

remove_hyphen TRUE/FALSE indicating whether hyphenated words should be adjusted to combine onto a single line. Default is TRUE.

convert_sentence TRUE/FALSE indicating if individual lines of PDF file should be collapsed into a single large paragraph to perform keyword searching. Default is TRUE.

remove_equations TRUE/FALSE indicating if equations should be removed. Default behavior is to search for the following regex: "\([0-9]1,\)$", essentially this matches a literal parenthesis, followed by at least one number followed by another parenthesis at the end of the text line. This will not detect other patterns or detect the entire equation if it is a multi-row equation.

split_pattern Regular expression pattern used to split multicolumn PDF files using stringi::stri_split_regex. Default pattern is "\pWHITE_SPACE3," which can be interpreted as: split based on three or more consecutive white space characters.

full_names TRUE/FALSE indicating if the full file path should be used. Default is TRUE, see list.files for more details.

file_pattern An optional regular expression to select specific file names. Only files that match the regular expression will be searched. Defaults to all pdfs, i.e. ".pdf". See list.files for more details.

recursive TRUE/FALSE indicating if subdirectories should be searched as well. Default is FALSE, see list.files for more details.

max_search An optional numeric vector indicating the maximum number of pdfs to search. Will only search the first n cases.

... token_function to pass to convert_tokens function.

Value

A tibble data frame that contains the keyword, location of match, the line of text match, and optionally the tokens associated with the line of text match. The output is combined (row binded) for all pdf input files.

Examples

```
# find directory
directory <- system.file('pdf', package = 'pdfsearch')

# do search over two files
keyword_directory(directory,
       keyword = c('repeated measures', 'measurement error'),
       surround_lines = 1, full_names = TRUE)
```

```
# can also split pdfs
keyword_directory(directory,
       keyword = c('repeated measures', 'measurement error'),
       split_pdf = TRUE, remove_hyphen = FALSE,
       surround_lines = 1, full_names = TRUE)
```

---

keyword_search *Search a pdf file for keywords*

Description

This uses the pdf_text from the pdftools package to perform keyword searches. Keyword locations indicating the line of the text as well as the page number that the keyword is found are returned.

Usage

```
keyword_search(
  x,
  keyword,
  path = FALSE,
  surround_lines = FALSE,
  ignore_case = FALSE,
  token_results = TRUE,
  heading_search = FALSE,
  heading_args = NULL,
  split_pdf = FALSE,
  remove_hyphen = TRUE,
  convert_sentence = TRUE,
  remove_equations = FALSE,
  split_pattern = "\\p{WHITE_SPACE}{3,}",
  ...
)
```

Arguments

x	Either the text of the pdf read in with the pdftools package or a path for the location of the pdf file.
keyword	The keyword(s) to be used to search in the text. Multiple keywords can be specified with a character vector.
path	An optional path designation for the location of the pdf to be converted to text. The pdftools package is used for this conversion.
surround_lines	numeric/FALSE indicating whether the output should extract the surrounding lines of text in addition to the matching line. Default is FALSE, if not false, include a numeric number that indicates the additional number of surrounding lines that will be extracted.

ignore_case	TRUE/FALSE/vector of TRUE/FALSE, indicating whether the case of the keyword matters. Default is FALSE meaning that case of the keyword is literal. If a vector, must be same length as the keyword vector.
token_results	TRUE/FALSE indicating whether the results text returned should be split into tokens. See the tokenizers package and convert_tokens for more details. Defaults to TRUE.
heading_search	TRUE/FALSE indicating whether to search for headings in the pdf.
heading_args	A list of arguments to pass on to the heading_search function. See heading_search for more details on arguments needed.
split_pdf	TRUE/FALSE indicating whether to split the pdf using white space. This would be most useful with multicolumn pdf files. The split_pdf function attempts to recreate the column layout of the text into a single column starting with the left column and proceeding to the right.
remove_hyphen	TRUE/FALSE indicating whether hyphenated words should be adjusted to combine onto a single line. Default is TRUE.
convert_sentence	TRUE/FALSE indicating if individual lines of PDF file should be collapsed into a single large paragraph to perform keyword searching. Default is TRUE
remove_equations	TRUE/FALSE indicating if equations should be removed. Default behavior is to search for the following regex: "\([0-9]1,\)$", essentially this matches a literal parenthesis, followed by at least one number followed by another parenthesis at the end of the text line. This will not detect other patterns or detect the entire equation if it is a multi-row equation.
split_pattern	Regular expression pattern used to split multicolumn PDF files using stringi::stri_split_regex. Default pattern is "\pWHITE_SPACE3," which can be interpreted as: split based on three or more consecutive white space characters.
...	token_function to pass to convert_tokens function.

Value

A tibble data frame that contains the keyword, location of match, the line of text match, and optionally the tokens associated with the line of text match.

Examples

```
file <- system.file('pdf', '1501.00450.pdf', package = 'pdfsearch')

keyword_search(file, keyword = c('repeated measures', 'mixed effects'),
  path = TRUE)

# Add surrounding text
keyword_search(file, keyword = c('variance', 'mixed effects'),
  path = TRUE, surround_lines = 1)

# split pdf
keyword_search(file, keyword = c('repeated measures', 'mixed effects'),
```

```
  path = TRUE, split_pdf = TRUE, remove_hyphen = FALSE)
```

`run_shiny` *Run Shiny Application Demo*

Description

Function runs Shiny Application Demo

Usage

```
run_shiny()
```

Details

This function does not take any arguments and will run the Shiny Application. If running from RStudio, will open the application in the viewer, otherwise will use the default internet browser.

Index