

Package: pdfsearch (via r-universe)

October 16, 2026

Type Package
Version 0.5.0
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 (>= 4.1.0)
Imports pdftools, tibble, tokenizers, stringi, readr
Suggests shiny, testthat, knitr, rmarkdown, covr
Maintainer Brandon LeBeau <lebebr01+pdfsearch@gmail.com>
RoxygenNote 7.3.3
URL <https://github.com/lebebr01/pdfsearch>
BugReports <https://github.com/lebebr01/pdfsearch/issues>
VignetteBuilder knitr
Encoding UTF-8
Config/pak/sysreqs libicu-dev libjpeg-dev libssl-dev libpoppler-cpp-dev poppler-data libx11-dev
Repository <https://lebebr01.r-universe.dev>
Date/Publication 2026-02-18 21:20:58 UTC
RemoteUrl <https://github.com/lebebr01/pdfsearch>
RemoteRef HEAD
RemoteSha f2fccac1e49fe13a9bd407270d01f7ae983575aa

Contents

compare_mask_effect	2
convert_tokens	3
extract_tables	4
format_text	6

heading_search	9
keyword_directory	12
keyword_search	16
run_shiny	20

Index	21
--------------	-----------

compare_mask_effect	<i>Compare keyword results with and without coordinate masking</i>
---------------------	--

Description

Runs [keyword_search](#) twice with coordinate splitting: once with mask_nonprose = FALSE and once with mask_nonprose = TRUE. Returns a compact summary for A/B checks.

Usage

```
compare_mask_effect(  
  x,  
  keyword,  
  path = FALSE,  
  column_count = c("auto", "1", "2"),  
  nonprose_digit_ratio = 0.35,  
  nonprose_symbol_ratio = 0.15,  
  nonprose_short_token_max = 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. Must be TRUE for coordinate splitting.
column_count	Expected number of columns for coordinate splitting. Options are "auto", "1", or "2".
nonprose_digit_ratio	Numeric threshold for classifying a line as non-prose based on digit character ratio.
nonprose_symbol_ratio	Numeric threshold for classifying a line as non-prose based on math-symbol character ratio.
nonprose_short_token_max	Maximum token count for short symbolic lines to classify as non-prose.
...	Additional arguments passed to keyword_search .

Value

A tibble data frame with one row per mode ("unmasked", "masked") and the number of matches.

Examples

```
file <- system.file('pdf', '1501.00450.pdf', package = 'pdfsearch')
compare_mask_effect(file, keyword = "error", path = TRUE, column_count = "2")
```

convert_tokens	<i>Ability to tokenize words.</i>
----------------	-----------------------------------

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)
```

extract_tables	<i>Function to extract tables</i>
----------------	-----------------------------------

Description

Function to extract tables

Usage

```
extract_tables(
  x,
  path = FALSE,
  split_pdf = FALSE,
  remove_equations = TRUE,
  delimiter = "\\s{2,}",
  delimiter_table = "\\s{2,}",
  split_pattern = "\\p{WHITE_SPACE}{3,}",
  split_method = c("regex", "coordinates"),
  column_count = c("auto", "1", "2"),
  remove_section_headers = FALSE,
  remove_page_headers = FALSE,
  remove_page_footers = FALSE,
  remove_repeated_furniture = FALSE,
  table_min_numeric_tokens = 3,
  table_min_digit_ratio = 0.18,
  table_min_block_lines = 2,
  table_block_max_gap = 3,
  table_include_headers = TRUE,
  table_header_lookback = 3,
  table_include_notes = FALSE,
  table_note_lookahead = 2,
  remove_captions = TRUE,
  caption_continuation_max = 2,
  replacement = "\\ /",
  col_names = FALSE,
  output = c("parsed", "blocks", "both"),
  merge_across_pages = TRUE
)
```

Arguments

<code>x</code>	Either the text of the pdf read in with the pdftools package or a path for the location of the pdf file.
<code>path</code>	An optional path designation for the location of the pdf to be converted to text. The pdftools package is used for this conversion.
<code>split_pdf</code>	TRUE/FALSE indicating whether to split the pdf using white space. This would be most useful with multicolumn pdf files. The <code>split_pdf</code> function attempts to recreate the column layout of the text into a single column starting with the left column and proceeding to the right.
<code>remove_equations</code>	TRUE/FALSE indicating if equations should be removed. Default behavior is to search for 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.
<code>delimiter</code>	A delimiter used to detect tables. The default is two consecutive blank white spaces.
<code>delimiter_table</code>	A delimiter used to separate table cells. The default value is two consecutive blank white spaces.
<code>split_pattern</code>	Regular expression pattern used to split multicolumn PDF files using <code>stringi::stri_split_regex</code> . Default pattern is to split based on three or more consecutive white space characters.
<code>split_method</code>	Method used for splitting multicolumn PDF text. Defaults to "regex". Use "coordinates" to split with <code>pdftools::pdf_data()</code> token coordinates.
<code>column_count</code>	Expected number of columns for coordinate splitting. Options are "auto", "1", or "2". Used when <code>split_method = "coordinates"</code> .
<code>remove_section_headers</code>	TRUE/FALSE indicating if section-header-like lines should be removed prior to table extraction.
<code>remove_page_headers</code>	TRUE/FALSE indicating if page-header furniture should be removed prior to table extraction.
<code>remove_page_footers</code>	TRUE/FALSE indicating if page-footer furniture should be removed prior to table extraction.
<code>remove_repeated_furniture</code>	TRUE/FALSE indicating if repeated text found in page edges should be removed prior to table extraction.
<code>table_min_numeric_tokens</code>	Minimum numeric tokens used to classify a line as table-like.
<code>table_min_digit_ratio</code>	Minimum digit-character ratio used to classify a line as table-like.
<code>table_min_block_lines</code>	Minimum number of adjacent table-like lines for a block to be treated as a table block.

table_block_max_gap	Maximum gap (in lines) allowed between table-like lines inside one table block.
table_include_headers	TRUE/FALSE indicating if table header lines adjacent to detected table blocks should be included in output blocks.
table_header_lookback	Number of lines above a detected table block to inspect for header rows.
table_include_notes	TRUE/FALSE indicating if note/source lines after table blocks should be included in output blocks.
table_note_lookahead	Number of lines after a detected table block to inspect for note/source rows.
remove_captions	TRUE/FALSE indicating if figure/table caption lines should be removed before table-block detection.
caption_continuation_max	Number of additional lines after a caption start line to remove when they appear to be caption continuations.
replacement	A delimiter used to separate table cells after the replacement of white space is done.
col_names	TRUE/FALSE value passed to 'readr::read_delim' to indicate if column names should be used. Default value is FALSE which means column names will be generic (i.e. X1, X2, etc). A value of TRUE would take the values from the first row of data extracted.
output	Output mode: "parsed" returns list of parsed data frames, "blocks" returns detected table blocks with metadata, and "both" returns a list with both representations.
merge_across_pages	TRUE/FALSE indicating if adjacent blocks on consecutive pages should be merged when they appear to be table continuations.

format_text

Format PDF input text

Description

Performs some formatting of pdf text upon import.

Usage

```
format_text(
  pdf_text,
  split_pdf = FALSE,
  blank_lines = TRUE,
  remove_hyphen = TRUE,
```

```

convert_sentence = TRUE,
remove_equations = FALSE,
split_pattern = "\\p{WHITE_SPACE}{3,}",
split_method = c("regex", "coordinates"),
pdf_data = NULL,
column_count = c("auto", "1", "2"),
mask_nonprose = FALSE,
nonprose_digit_ratio = 0.35,
nonprose_symbol_ratio = 0.15,
nonprose_short_token_max = 3,
remove_section_headers = FALSE,
remove_page_headers = FALSE,
remove_page_footers = FALSE,
page_margin_ratio = 0.08,
remove_repeated_furniture = FALSE,
repeated_edge_n = 3,
repeated_edge_min_pages = 4,
remove_captions = FALSE,
caption_continuation_max = 2,
table_mode = c("keep", "remove", "only"),
table_min_numeric_tokens = 3,
table_min_digit_ratio = 0.18,
table_min_block_lines = 2,
table_block_max_gap = 3,
table_include_headers = TRUE,
table_header_lookback = 3,
table_include_notes = FALSE,
table_note_lookahead = 2,
concatenate_pages = FALSE,
...
)

```

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.
blank_lines	TRUE/FALSE indicating whether blank text lines should be removed. Default is TRUE.
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

<code>remove_equations</code>	TRUE/FALSE indicating if equations should be removed. Default behavior is to search for 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.
<code>split_pattern</code>	Regular expression pattern used to split multicolumn PDF files using <code>stringi::stri_split_regex</code> . Default pattern is to split based on three or more consecutive white space characters.
<code>split_method</code>	Method used for splitting multicolumn PDF text. Defaults to "regex". Use "coordinates" to split with <code>pdftools::pdf_data()</code> token coordinates.
<code>pdf_data</code>	Optional token-level PDF data from <code>pdftools::pdf_data()</code> . Used when <code>split_method = "coordinates"</code> .
<code>column_count</code>	Expected number of columns for coordinate splitting. Options are "auto", "1", or "2". Used when <code>split_method = "coordinates"</code> .
<code>mask_nonprose</code>	TRUE/FALSE indicating if non-prose lines (likely equations, tables, figure/table captions) should be removed when using coordinate splitting.
<code>nonprose_digit_ratio</code>	Numeric threshold for classifying a line as non-prose based on digit character ratio.
<code>nonprose_symbol_ratio</code>	Numeric threshold for classifying a line as non-prose based on math-symbol character ratio.
<code>nonprose_short_token_max</code>	Maximum token count for short symbolic lines to classify as non-prose.
<code>remove_section_headers</code>	TRUE/FALSE indicating if section-header-like lines should be removed when using coordinate splitting.
<code>remove_page_headers</code>	TRUE/FALSE indicating if page-header furniture (e.g., arXiv identifiers, emails, URLs) should be removed when using coordinate splitting.
<code>remove_page_footers</code>	TRUE/FALSE indicating if page-footer furniture (e.g., page numbers, copyright markers) should be removed when using coordinate splitting.
<code>page_margin_ratio</code>	Numeric ratio used to define top and bottom page bands for header/footer removal.
<code>remove_repeated_furniture</code>	TRUE/FALSE indicating if repeated text found in the first/last lines across many pages should be removed.
<code>repeated_edge_n</code>	Number of lines from top and bottom of each page to consider for repeated edge-line detection.
<code>repeated_edge_min_pages</code>	Minimum number of pages an edge line must appear on before being removed.
<code>remove_captions</code>	TRUE/FALSE indicating if figure/table caption lines should be removed.

caption_continuation_max	Number of additional lines after a caption start line to remove when they appear to be caption continuations.
table_mode	How to handle detected table blocks. "keep" keeps all lines, "remove" excludes table blocks, and "only" keeps only table blocks.
table_min_numeric_tokens	Minimum numeric tokens used to classify a line as table-like.
table_min_digit_ratio	Minimum digit-character ratio used to classify a line as table-like.
table_min_block_lines	Minimum number of adjacent table-like lines for a block to be treated as a table block.
table_block_max_gap	Maximum gap (in lines) allowed between table-like lines inside one table block.
table_include_headers	TRUE/FALSE indicating if table header lines adjacent to detected table blocks should be included in table blocks.
table_header_lookback	Number of lines above a detected table block to inspect for header rows.
table_include_notes	TRUE/FALSE indicating if trailing note/source lines should be included with detected table blocks.
table_note_lookahead	Number of lines after a detected table block to inspect for note/source rows.
concatenate_pages	TRUE/FALSE indicating if page text should be concatenated before sentence conversion. This is only used when convert_sentence = TRUE.
...	Additional arguments, currently not used.

heading_search	<i>Function to locate sections of pdf</i>
----------------	---

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,
split_method = c("regex", "coordinates"),
column_count = c("auto", "1", "2"),
mask_nonprose = FALSE,
nonprose_digit_ratio = 0.35,
nonprose_symbol_ratio = 0.15,
nonprose_short_token_max = 3,
remove_section_headers = FALSE,
remove_page_headers = FALSE,
remove_page_footers = FALSE,
page_margin_ratio = 0.08,
remove_repeated_furniture = FALSE,
repeated_edge_n = 3,
repeated_edge_min_pages = 4,
remove_captions = FALSE,
caption_continuation_max = 2,
table_mode = c("keep", "remove", "only"),
table_min_numeric_tokens = 3,
table_min_digit_ratio = 0.18,
table_min_block_lines = 2,
table_block_max_gap = 3,
table_include_headers = TRUE,
table_header_lookback = 3,
table_include_notes = FALSE,
table_note_lookahead = 2,
concatenate_pages = 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.
split_method	Method used for splitting multicolumn PDF text. Defaults to "regex". Use "coordinates" to split with pdfutils::pdf_data() token coordinates.
column_count	Expected number of columns for coordinate splitting. Options are "auto", "1", or "2". Used when split_method = "coordinates".
mask_nonprose	TRUE/FALSE indicating if non-prose lines (likely equations, tables, figure/table captions) should be removed when using coordinate splitting.
nonprose_digit_ratio	Numeric threshold for classifying a line as non-prose based on digit character ratio.
nonprose_symbol_ratio	Numeric threshold for classifying a line as non-prose based on math-symbol character ratio.
nonprose_short_token_max	Maximum token count for short symbolic lines to classify as non-prose.
remove_section_headers	TRUE/FALSE indicating if section-header-like lines should be removed when using coordinate splitting.
remove_page_headers	TRUE/FALSE indicating if page-header furniture (e.g., arXiv identifiers, emails, URLs) should be removed when using coordinate splitting.
remove_page_footers	TRUE/FALSE indicating if page-footer furniture (e.g., page numbers, copyright markers) should be removed when using coordinate splitting.
page_margin_ratio	Numeric ratio used to define top and bottom page bands for header/footer removal.
remove_repeated_furniture	TRUE/FALSE indicating if repeated text found in the first/last lines across many pages should be removed.
repeated_edge_n	Number of lines from top and bottom of each page to consider for repeated edge-line detection.
repeated_edge_min_pages	Minimum number of pages an edge line must appear on before being removed.
remove_captions	TRUE/FALSE indicating if figure/table caption lines should be removed.
caption_continuation_max	Number of additional lines after a caption start line to remove when they appear to be caption continuations.
table_mode	How to handle detected table blocks. "keep" keeps all lines, "remove" excludes table blocks, and "only" keeps only table blocks.

`table_min_numeric_tokens`
Minimum numeric tokens used to classify a line as table-like.

`table_min_digit_ratio`
Minimum digit-character ratio used to classify a line as table-like.

`table_min_block_lines`
Minimum number of adjacent table-like lines for a block to be treated as a table block.

`table_block_max_gap`
Maximum gap (in lines) allowed between table-like lines inside one table block.

`table_include_headers`
TRUE/FALSE indicating if table header lines adjacent to detected table blocks should be included in table blocks.

`table_header_lookback`
Number of lines above a detected table block to inspect for header rows.

`table_include_notes`
TRUE/FALSE indicating if trailing note/source lines should be included with detected table blocks.

`table_note_lookahead`
Number of lines after a detected table block to inspect for note/source rows.

`concatenate_pages`
TRUE/FALSE indicating if page text should be concatenated after column rectification and cleaning, before sentence conversion. This is only used when `convert_sentence = 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 FALSE

Examples

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

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

keyword_directory	<i>Wrapper for keyword search function</i>
-------------------	--

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,}",
  split_method = c("regex", "coordinates"),
  column_count = c("auto", "1", "2"),
  mask_nonprose = FALSE,
  nonprose_digit_ratio = 0.35,
  nonprose_symbol_ratio = 0.15,
  nonprose_short_token_max = 3,
  remove_section_headers = FALSE,
  remove_page_headers = FALSE,
  remove_page_footers = FALSE,
  page_margin_ratio = 0.08,
  remove_repeated_furniture = FALSE,
  repeated_edge_n = 3,
  repeated_edge_min_pages = 4,
  remove_captions = FALSE,
  caption_continuation_max = 2,
  table_mode = c("keep", "remove", "only"),
  table_min_numeric_tokens = 3,
  table_min_digit_ratio = 0.18,
  table_min_block_lines = 2,
  table_block_max_gap = 3,
  table_include_headers = TRUE,
  table_header_lookback = 3,
  table_include_notes = FALSE,
  table_note_lookahead = 2,
  concatenate_pages = FALSE,
  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 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.
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 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 <code>stringi::stri_split_regex</code> . Default pattern is to split based on three or more consecutive white space characters.
split_method	Method used for splitting multicolumn PDF text. Defaults to "regex". Use "coordinates" to split with <code>pdfutils::pdf_data()</code> token coordinates.
column_count	Expected number of columns for coordinate splitting. Options are "auto", "1", or "2". Used when <code>split_method = "coordinates"</code> .
mask_nonprose	TRUE/FALSE indicating if non-prose lines (likely equations, tables, figure/table captions) should be removed when using coordinate splitting.
nonprose_digit_ratio	Numeric threshold for classifying a line as non-prose based on digit character ratio.
nonprose_symbol_ratio	Numeric threshold for classifying a line as non-prose based on math-symbol character ratio.
nonprose_short_token_max	Maximum token count for short symbolic lines to classify as non-prose.
remove_section_headers	TRUE/FALSE indicating if section-header-like lines should be removed when using coordinate splitting.

remove_page_headers	TRUE/FALSE indicating if page-header furniture (e.g., arXiv identifiers, emails, URLs) should be removed when using coordinate splitting.
remove_page_footers	TRUE/FALSE indicating if page-footer furniture (e.g., page numbers, copyright markers) should be removed when using coordinate splitting.
page_margin_ratio	Numeric ratio used to define top and bottom page bands for header/footer removal.
remove_repeated_furniture	TRUE/FALSE indicating if repeated text found in the first/last lines across many pages should be removed.
repeated_edge_n	Number of lines from top and bottom of each page to consider for repeated edge-line detection.
repeated_edge_min_pages	Minimum number of pages an edge line must appear on before being removed.
remove_captions	TRUE/FALSE indicating if figure/table caption lines should be removed.
caption_continuation_max	Number of additional lines after a caption start line to remove when they appear to be caption continuations.
table_mode	How to handle detected table blocks. "keep" keeps all lines, "remove" excludes table blocks, and "only" keeps only table blocks.
table_min_numeric_tokens	Minimum numeric tokens used to classify a line as table-like.
table_min_digit_ratio	Minimum digit-character ratio used to classify a line as table-like.
table_min_block_lines	Minimum number of adjacent table-like lines for a block to be treated as a table block.
table_block_max_gap	Maximum gap (in lines) allowed between table-like lines inside one table block.
table_include_headers	TRUE/FALSE indicating if table header lines adjacent to detected table blocks should be included in table blocks.
table_header_lookback	Number of lines above a detected table block to inspect for header rows.
table_include_notes	TRUE/FALSE indicating if trailing note/source lines should be included with detected table blocks.
table_note_lookahead	Number of lines after a detected table block to inspect for note/source rows.
concatenate_pages	TRUE/FALSE indicating if page text should be concatenated after column rectification and cleaning, before sentence conversion. This is only used when <code>convert_sentence = TRUE</code> .

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	<i>Search a pdf file for keywords</i>
----------------	---------------------------------------

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,
  blank_lines = TRUE,
  remove_hyphen = TRUE,
  convert_sentence = TRUE,
  remove_equations = FALSE,
  split_pattern = "\\p{WHITE_SPACE}{3,}",
  split_method = c("regex", "coordinates"),
  column_count = c("auto", "1", "2"),
  mask_nonprose = FALSE,
  nonprose_digit_ratio = 0.35,
  nonprose_symbol_ratio = 0.15,
  nonprose_short_token_max = 3,
  remove_section_headers = FALSE,
  remove_page_headers = FALSE,
  remove_page_footers = FALSE,
  page_margin_ratio = 0.08,
  remove_repeated_furniture = FALSE,
  repeated_edge_n = 3,
  repeated_edge_min_pages = 4,
  remove_captions = FALSE,
  caption_continuation_max = 2,
  table_mode = c("keep", "remove", "only"),
  table_min_numeric_tokens = 3,
  table_min_digit_ratio = 0.18,
  table_min_block_lines = 2,
  table_block_max_gap = 3,
  table_include_headers = TRUE,
  table_header_lookback = 3,
  table_include_notes = FALSE,
  table_note_lookahead = 2,
  concatenate_pages = 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.
---	--

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.
blank_lines	TRUE/FALSE indicating whether blank text lines should be removed. Default is TRUE.
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 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 <code>stringi::stri_split_regex</code> . Default pattern is to split based on three or more consecutive white space characters.
split_method	Method used for splitting multicolumn PDF text. Defaults to "regex". Use "coordinates" to split with <code>pdftools::pdf_data()</code> token coordinates.
column_count	Expected number of columns for coordinate splitting. Options are "auto", "1", or "2". Used when <code>split_method = "coordinates"</code> .
mask_nonprose	TRUE/FALSE indicating if non-prose lines (likely equations, tables, figure/table captions) should be removed when using coordinate splitting.
nonprose_digit_ratio	Numeric threshold for classifying a line as non-prose based on digit character ratio.

nonprose_symbol_ratio	Numeric threshold for classifying a line as non-prose based on math-symbol character ratio.
nonprose_short_token_max	Maximum token count for short symbolic lines to classify as non-prose.
remove_section_headers	TRUE/FALSE indicating if section-header-like lines should be removed when using coordinate splitting.
remove_page_headers	TRUE/FALSE indicating if page-header furniture (e.g., arXiv identifiers, emails, URLs) should be removed when using coordinate splitting.
remove_page_footers	TRUE/FALSE indicating if page-footer furniture (e.g., page numbers, copyright markers) should be removed when using coordinate splitting.
page_margin_ratio	Numeric ratio used to define top and bottom page bands for header/footer removal.
remove_repeated_furniture	TRUE/FALSE indicating if repeated text found in the first/last lines across many pages should be removed.
repeated_edge_n	Number of lines from top and bottom of each page to consider for repeated edge-line detection.
repeated_edge_min_pages	Minimum number of pages an edge line must appear on before being removed.
remove_captions	TRUE/FALSE indicating if figure/table caption lines should be removed.
caption_continuation_max	Number of additional lines after a caption start line to remove when they appear to be caption continuations.
table_mode	How to handle detected table blocks. "keep" keeps all lines, "remove" excludes table blocks, and "only" keeps only table blocks.
table_min_numeric_tokens	Minimum numeric tokens used to classify a line as table-like.
table_min_digit_ratio	Minimum digit-character ratio used to classify a line as table-like.
table_min_block_lines	Minimum number of adjacent table-like lines for a block to be treated as a table block.
table_block_max_gap	Maximum gap (in lines) allowed between table-like lines inside one table block.
table_include_headers	TRUE/FALSE indicating if table header lines adjacent to detected table blocks should be included in table blocks.
table_header_lookback	Number of lines above a detected table block to inspect for header rows.

`table_include_notes`
 TRUE/FALSE indicating if trailing note/source lines should be included with detected table blocks.

`table_note_lookahead`
 Number of lines after a detected table block to inspect for note/source rows.

`concatenate_pages`
 TRUE/FALSE indicating if page text should be concatenated after column rectification and cleaning, before sentence conversion. This is only used when `convert_sentence = TRUE`.

...
 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

`compare_mask_effect`, [2](#)
`convert_tokens`, [3](#), [14](#), [16](#), [18](#), [20](#)
`extract_tables`, [4](#)
`format_text`, [6](#)
`heading_search`, [9](#), [18](#)
`keyword_directory`, [12](#)
`keyword_search`, [2](#), [16](#)
`list.files`, [16](#)
`run_shiny`, [20](#)