Package 'PDE'

June 11, 2024

Type Package

Title Extract Tables and Sentences from PDFs with User Interface

Version 1.4.10

Author Erik Stricker [aut, cre]

Maintainer Erik Stricker <erik.stricker@gmx.com>

Description The PDE (Pdf Data Extractor) allows the extraction of information and tables optionally based on search words from PDF (Portable Document Format) files and enables the visualization of the results, both by providing a convenient user-interface.

License GPL-3 | file LICENSE

Encoding UTF-8

Imports tcltk

Depends tcltk2 (>= 1.2.11), R (>= 3.5)

SystemRequirements XPDF (4.02)(https://github.com/erikstricker/PDE/tree/master/inst/examples/bin)

RoxygenNote 7.3.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2024-06-11 18:10:06 UTC

Contents

`.PDE_extr_data_from_pdf` *Extracting data from a PDF (Protable Document Format) file*

Description

`PDE_extr_data_from_pdf` extracts sentences or tables from a single PDF file and writes output in the corresponding folder.

Usage

```
.PDE_extr_data_from_pdf(
  pdf,
  whattoextr,
  out = ".",
  filter.words = "",
  regex.fw = TRUE,
  ignore.case.fw = FALSE,
  filter.word.times = "0.2%",
  table.heading.words = "",
  ignore.case.th = FALSE,
  search.words,
  search.word.categories = NULL,
  save.tab.by.category = FALSE,
  regex.sw = TRUE,
  ignore.case.sw = FALSE,
  eval.abbrevs = TRUE,
  out.table.format = ".csv (WINDOWS-1252)",
  dev_x = 20,
  dev_y = 9999,
  context = 0,
  write.table.locations = FALSE,
  exp.nondetc.tabs = TRUE,
  write.tab.doc.file = TRUE,
  write.txt.doc.file = TRUE,
  delete = TRUE,
  cpy_mv = "nocpymv",
  verbose = TRUE
)
```

Arguments

pdf	String. Path to the PDF file to be analyzed.
whattoextr	String. Either *txt*, *tab*, or *tabandtxt* for PDFS2TXT (extract sentences from a PDF file) or PDFS2TABLE (table of a PDF file to a Microsoft Excel file) extraction. *tab* allows the extraction of tables with and without search words while *txt* and *tabandtxt* require search words.
out	String. Directory chosen to save analysis results in. Default: ".".
filter.words	List of strings. The list of filter words. If not NA or "" a hit will be counted every time a word from the list is detected in the article. Default: "".
regex.fw	Logical. If TRUE filter words will follow the regex rules (see https://github.com/erikstricker/PDE/blob/master/inst/examples/cheatsheets/regex.pdf). Default = TRUE.
ignore.case.fw	Logical. Are the filter words case-sensitive (does capitalization matter)? Default: FALSE.
filter.word.times	Numeric or string. Can either be expressed as absolute number or percentage of the total number of words (by adding the " filter.words for a paper to be further analyzed. Default: 0.2%.
table.heading.words	List of strings. Different than standard (TABLE, TAB or table plus number) headings to be detected. Regex rules apply (see also https://github.com/erikstricker/PDE/blob/master/inst/examples/cheatsheets/regex.pdf). Default = "".
ignore.case.th	Logical. Are the additional table headings (see table.heading.words) case-sensitive (does capitalization matter)? Default = FALSE.
search.words	List of strings. List of search words. To extract all tables from the PDF file leave search.words = "".
search.word.categories	List of strings. List of categories with the same length as the list of search words. Accordingly, each search word can be assigned to a category, of which the word counts will be summarized in the PDE_analyzer_word_stats.csv file. If search.word.categories is a different length than search.words the parameter will be ignored. Default: NULL.
save.tab.by.category	Logical. Can only be used with search.word.categories. If set to TRUE, tables that carry search words will be saved in sub-folders according to the search word category of the detected search word. Default: FALSE.
regex.sw	Logical. If TRUE search words will follow the regex rules (see https://github.com/erikstricker/PDE/blob/master/inst/examples/cheatsheets/regex.pdf). Default = TRUE.
ignore.case.sw	Logical. Are the search words case-sensitive (does capitalization matter)? Default: FALSE.
eval.abbrevs	Logical. Should abbreviations for the search words be automatically detected and then replaced with the search word + "$*"? Default: TRUE.

`out.table.format`
: String. Output file format. Either comma separated file `.csv` or tab separated file `.tsv`. The encoding indicated in parantheses should be selected according to the operational system exported tables are opened in, i.e., Windows: `"(WINDOWS-1252)"`; Mac: `(macintosh)`; Linux: `(UTF-8)`. Default: `".csv"` and encoding depending on the operational system.

`dev_x`
: Numeric. For a table the size of indention which would be considered the same column. Default: `20`.

`dev_y`
: Numeric. For a table the vertical distance which would be considered the same row. Can be either a number or set to dynamic detection [9999], in which case the font size is used to detect which words are in the same row. Default: `9999`.

`context`
: Numeric. Number of sentences extracted before and after the sentence with the detected search word. If `0` only the sentence with the search word is extracted. Default: `0`.

`write.table.locations`
: Logical. If `TRUE`, a separate file with the headings of all tables, their relative location in the generated html and txt files, as well as information if search words were found will be generated. Default: `FALSE`.

`exp.nondetc.tabs`
: Logical. If `TRUE`, if a table was detected in a PDF file but is an image or cannot be read, the page with the table with be exported as a png. Default: `TRUE`.

`write.tab.doc.file`
: Logical. If `TRUE`, if search words are used for table detection and no search words were found in the tables of a PDF file, a **no.table.w.search.words**. Default: `TRUE`.

`write.txt.doc.file`
: Logical. If `TRUE`, if no search words were found in the sentences of a PDF file, a file will be created with the PDF filename followed by **no.txt.w.search.words**. If the PDF file is empty, a file will be created with the PDF filename followed by **no.content.detected**. If the filter word threshold is not met, a file will be created with the PDF filename followed by **no.txt.w.filter.words**. Default: `TRUE`.

`delete`
: Logical. If `TRUE`, the intermediate **txt**, **keeplayouttxt** and **html** copies of the PDF file will be deleted. Default: `TRUE`.

`cpy_mv`
: String. Either "nocpymv", "cpy", or "mv". If filter words are used in the analyses, the processed PDF files will either be copied ("cpy") or moved ("mv") into the /pdf/ subfolder of the output folder. Default: `"nocpymv"`.

`verbose`
: Logical. Indicates whether messages will be printed in the console. Default: `TRUE`.

Value

If tables were extracted from the PDF file the function returns a list of following tables/items: 1) **htmltablelines**, 2) **txttablelines**, 3) **keeplayouttxttablelines**, 4) **id**, 5) **out_msg**. The **tablelines** are tables that provide the heading and position of the detected tables. The **id** provide the name of the PDF file. The **out_msg** includes all messages printed to the console or the suppressed messages if `verbose=FALSE`.

See Also

PDE_pdfs2table,PDE_pdfs2table_searchandfilter, PDE_pdfs2txt_searchandfilter

Examples

```
## Running a simple analysis with filter and search words to extract sentences and tables
if(PDE_check_Xpdf_install() == TRUE){
 outputtables <- .PDE_extr_data_from_pdf(pdf = "/examples/Methotrexate/29973177_!.pdf",
 whattoextr = "tabandtxt",
 out = paste0(system.file(package = "PDE"),"/examples/MTX_output+-0_test/"),
 filter.words = strsplit("cohort;case-control;group;study population;study participants", ";")[[1]],
 ignore.case.fw = TRUE,
 regex.fw = FALSE,
 search.words = strsplit("(M|m)ethotrexate;(T|t)rexal;(R|r)heumatrex;(O|o)trexup", ";")[[1]],
 ignore.case.sw = FALSE,
 regex.sw = TRUE)
}

## Running an advanced analysis with filter and search words to
## extract sentences and tables and obtain documentation files
if(PDE_check_Xpdf_install() == TRUE){
 outputtables <- .PDE_extr_data_from_pdf(pdf = paste0(system.file(package = "PDE"),
                      "/examples/Methotrexate/29973177_!.pdf"),
 whattoextr = "tabandtxt",
 out = paste0(system.file(package = "PDE"),"/examples/MTX_output+-1_test/"),
 context = 1,
 dev_x = 20,
 dev_y = 9999,
 filter.words = strsplit("cohort;case-control;group;study population;study participants", ";")[[1]],
 ignore.case.fw = TRUE,
 regex.fw = FALSE,
 filter.word.times = "0.2%",
 table.heading.words = "",
 ignore.case.th = FALSE,
 search.words = strsplit("(M|m)ethotrexate;(T|t)rexal;(R|r)heumatrex;(O|o)trexup", ";")[[1]],
 ignore.case.sw = FALSE,
 regex.sw = TRUE,
 eval.abbrevs = TRUE,
 out.table.format = ".csv (WINDOWS-1252)",
 write.table.locations = TRUE,
 write.tab.doc.file = TRUE,
 write.txt.doc.file = TRUE,
 exp.nondetc.tabs = TRUE,
 cpy_mv = "nocpymv",
 delete = TRUE)
}
```

Description

The package includes two main components: 1) The PDE analyzer performs the sentence and table extraction while 2) the PDE reader allows the user-friendly visualization and quick-processing of the obtained results.

PDE functions

PDE_analyzer, PDE_analyzer_i, PDE_extr_data_from_pdfs, PDE_pdfs2table, PDE_pdfs2table_searchandfilter,PD
PDE_reader_i, PDE_install_Xpdftools4.02, PDE_check_Xpdf_install

_PACKAGE

PDE-deprecated *Deprecated functions in package 'PDE'*

Description

These functions are provided for compatibility with older versions of 'PDE' only, and will be defunct at the next release.

Details

The following functions are deprecated and will be made defunct; use the replacement indicated below:

- PDE_path: `system.file(package = "PDE")`

PDE_analyzer *Extracting data from PDF (Portable Document Format) files*

Description

The `PDE_analyzer` allows the sentence and table extraction from multiple PDF files.

Usage

```
PDE_analyzer(PDE_parameters_file_path = NA, verbose = TRUE)
```

Arguments

`PDE_parameters_file_path`	String. This file includes all parameters to run `PDE_extr_data_from_pdfs` on multiple PDF files. If `PDE_parameters_file_path` does not exist or is `NA` a dialog box is opened prompting the user to select the parameter file.
`verbose`	Logical. Indicates whether messages will be printed in the console. Default: `TRUE`.

Value

If tables were extracted from the PDF file the function returns a list of following tables/items: 1) **htmltablelines**, 2) **txttablelines**, 3) **keeplayouttxttablelines**, 4) **id**, 5) **out_msg**. The **tablelines** are tables that provide the heading and position of the detected tables. The **id** provide the name of the PDF file. The **out_msg** includes all messages printed to the console or the suppressed messages if `verbose=FALSE`.

Details

The parameter file (also referred to as .tsv file) can either manually or with the help of the `PDE_analyzer_i` interface be filled.

Note

A detailed description of the parameters in the TSV file can be found in the markdown file (README_PDE.md) and in the description of `PDE_extr_data_from_pdfs`.

See Also

`PDE_extr_data_from_pdfs`

Examples

```
if(PDE_check_Xpdf_install() == TRUE){
  PDE_analyzer(paste0(system.file(package = "PDE"),
  "/examples/tsvs/PDE_parameters_v1.4_all_files+-0.tsv"))
}

## Not run:
 ## requires user file choice:
 PDE_analyzer()

## End(Not run)
```

`PDE_analyzer_i` *Extracting data from PDF (Portable Document Format) files using a user interface*

Description

The `PDE_analyzer_i` provides a user interface for the sentence and table extraction from multiple PDF files.

Usage

```
PDE_analyzer_i(verbose = TRUE)
```

Arguments

verbose	Logical. Indicates whether messages will be printed in the console. Default: TRUE.

Note

A detailed description of the elements in the user interface can be found in the markdown file (README_PDE.md).

Examples

```
PDE_analyzer_i()
```

PDE_check_Xpdf_install

Check if the Xpdftools are installed an in the system path

Description

PDE_check_Xpdf_install runs a version test for pdftotext, pdftohtml and pdftopng.

Usage

```
PDE_check_Xpdf_install(sysname = NULL, verbose = TRUE)
```

Arguments

sysname	String. In case the function returns "Unknown OS" the sysname can be set manually. Allowed options are "Windows", "Linux", "SunOS" for Solaris, and "Darwin" for Mac. Default: NULL.
verbose	Logical. Indicates whether messages will be printed in the console. Default: TRUE.

Value

The function returns a Boolean for the installation status and a message in case the commands are not detected.

Examples

```
PDE_check_Xpdf_install()
```

PDE_extr_data_from_pdfs

Extracting data from PDF (Portable Document Format) files

Description

PDE_extr_data_from_pdfs extracts sentences or tables from a single PDF file and writes output in the corresponding folder.

Usage

```
PDE_extr_data_from_pdfs(
  pdfs,
  whattoextr,
  out = ".",
  filter.words = "",
  regex.fw = TRUE,
  ignore.case.fw = FALSE,
  filter.word.times = "0.2%",
  table.heading.words = "",
  ignore.case.th = FALSE,
  search.words,
  search.word.categories = NULL,
  regex.sw = TRUE,
  save.tab.by.category = FALSE,
  ignore.case.sw = FALSE,
  eval.abbrevs = TRUE,
  out.table.format = ".csv (WINDOWS-1252)",
  dev_x = 20,
  dev_y = 9999,
  context = 0,
  write.table.locations = FALSE,
  exp.nondetc.tabs = TRUE,
  write.tab.doc.file = TRUE,
  write.txt.doc.file = TRUE,
  delete = TRUE,
  cpy_mv = "nocpymv",
  verbose = TRUE
)
```

Arguments

pdfs	String. A list of paths to the PDF files to be analyzed.
whattoextr	String. Either *txt*, *tab*, or *tabandtxt* for PDFS2TXT (extract sentences from a PDF file) or PDFS2TABLE (table of a PDF file to a Microsoft Excel file) extraction. *tab* allows the extraction of tables with and without search words while *txt* and *tabandtxt* require search words.

out String. Directory chosen to save analysis results in. Default: ".".

filter.words List of strings. The list of filter words. If not NA or "" a hit will be counted every time a word from the list is detected in the article. Default: "".

regex.fw Logical. If TRUE filter words will follow the regex rules (see https://github.com/erikstricker/PDE/blob/master/inst/examples/cheatsheets/regex.pdf). Default = TRUE.

ignore.case.fw Logical. Are the filter words case-sensitive (does capitalization matter)? Default: FALSE.

filter.word.times
Numeric or string. Can either be expressed as absolute number or percentage of the total number of words (by adding the " filter.words for a paper to be further analyzed. Default: 0.2%.

table.heading.words
List of strings. Different than standard (TABLE, TAB or table plus number) headings to be detected. Regex rules apply (see also https://github.com/erikstricker/PDE/blob/master/inst/examples/cheatsheets/regex.pdf). Default = "".

ignore.case.th Logical. Are the additional table headings (see table.heading.words) case-sensitive (does capitalization matter)? Default = FALSE.

search.words List of strings. List of search words. To extract all tables from the PDF files leave search.words = "".

search.word.categories
List of strings. List of categories with the same length as the list of search words. Accordingly, each search word can be assigned to a category, of which the word counts will be summarized in the PDE_analyzer_word_stats.csv file. If search.word.categories is a different length than search.words the parameter will be ignored. Default: NULL.

regex.sw Logical. If TRUE search words will follow the regex rules (see https://github.com/erikstricker/PDE/blob/master/inst/examples/cheatsheets/regex.pdf). Default = TRUE.

save.tab.by.category
Logical. Can only be used with search.word.categories. If set to TRUE, tables that carry search words will be saved in sub-folders according to the search word category of the detected search word. Default: FALSE.

ignore.case.sw Logical. Are the search words case-sensitive (does capitalization matter)? Default: FALSE.

eval.abbrevs Logical. Should abbreviations for the search words be automatically detected and then replaced with the search word + "$*"? Default: TRUE.

out.table.format
String. Output file format. Either comma separated file .csv or tab separated file .tsv. The encoding indicated in parantheses should be selected according to the operational system exported tables are opened in, i.e., Windows: "(WINDOWS-1252)"; Mac: (macintosh); Linux: (UTF-8). Default: ".csv" and encoding depending on the operational system.

dev_x Numeric. For a table the size of indention which would be considered the same column. Default: 20.

dev_y — Numeric. For a table the vertical distance which would be considered the same row. Can be either a number or set to dynamic detection [9999], in which case the font size is used to detect which words are in the same row. Default: 9999.

context — Numeric. Number of sentences extracted before and after the sentence with the detected search word. If 0 only the sentence with the search word is extracted. Default: 0.

write.table.locations — Logical. If TRUE, a separate file with the headings of all tables, their relative location in the generated html and txt files, as well as information if search words were found will be generated. Default: FALSE.

exp.nondetc.tabs — Logical. If TRUE, if a table was detected in a PDF file but is an image or cannot be read, the page with the table with be exported as a png. Default: TRUE.

write.tab.doc.file — Logical. If TRUE, if search words are used for table detection and no search words were found in the tables of a PDF file, a **no.table.w.search.words**. Default: TRUE.

write.txt.doc.file — Logical. If TRUE, if no search words were found in the sentences of a PDF file, a file will be created with the PDF filename followed by **no.txt.w.search.words**. If the PDF file is empty, a file will be created with the PDF filename followed by **no.content.detected**. If the filter word threshold is not met, a file will be created with the PDF filename followed by **no.txt.w.filter.words**. Default: TRUE.

delete — Logical. If TRUE, the intermediate **txt**, **keeplayouttxt** and **html** copies of the PDF files will be deleted. Default: TRUE.

cpy_mv — String. Either "nocpymv", "cpy", or "mv". If filter words are used in the analyses, the processed PDF files will either be copied ("cpy") or moved ("mv") into the /pdf/ subfolder of the output folder. Default: "nocpymv".

verbose — Logical. Indicates whether messages will be printed in the console. Default: TRUE.

Value

If tables were extracted from the PDF file the function returns a list of following tables/items: 1) **htmltablelines**, 2) **txttablelines**, 3) **keeplayouttxttablelines**, 4) **id**, 5) **out_msg**. The **tablelines** are tables that provide the heading and position of the detected tables. The **id** provide the name of the PDF file. The **out_msg** includes all messages printed to the console or the suppressed messages if verbose=FALSE.

See Also

PDE_pdfs2table,PDE_pdfs2table_searchandfilter,PDE_pdfs2txt_searchandfilter

Examples

```
## Running a simple analysis with filter and search words to extract sentences and tables
if(PDE_check_Xpdf_install() == TRUE){
 outputtables <- PDE_extr_data_from_pdfs(pdfs = c(paste0(system.file(package = "PDE"),
```

```
                                                  "/examples/Methotrexate/29973177_!.pdf"),
                                                    paste0(system.file(package = "PDE"),
                                                  "/examples/Methotrexate/31083238_!.pdf")),
 whattoextr = "tabandtxt",
 out = paste0(system.file(package = "PDE"),"/examples/MTX_output+-0_test/"),
 filter.words = strsplit("cohort;case-control;group;study population;study participants", ";")[[1]],
 ignore.case.fw = TRUE,
 regex.fw = FALSE,
 search.words = strsplit("(M|m)ethotrexate;(T|t)rexal;(R|r)heumatrex;(O|o)trexup", ";")[[1]],
 ignore.case.sw = FALSE,
 regex.sw = TRUE)
}

## Running an advanced analysis with filter and search words to
## extract sentences and tables and obtain documentation files
if(PDE_check_Xpdf_install() == TRUE){
 outputtables <- PDE_extr_data_from_pdfs(pdfs = c(paste0(system.file(package = "PDE"),
                                                  "/examples/Methotrexate/29973177_!.pdf"),
                                                     paste0(system.file(package = "PDE"),
                                                  "/examples/Methotrexate/31083238_!.pdf")),
 whattoextr = "tabandtxt",
 out = paste0(system.file(package = "PDE"),"/examples/MTX_output+-1_test/"),
 context = 1,
 dev_x = 20,
 dev_y = 9999,
 filter.words = strsplit("cohort;case-control;group;study population;study participants",";")[[1]],
 ignore.case.fw = TRUE,
 regex.fw = FALSE,
 filter.word.times = "0.2%",
 table.heading.words = "",
 ignore.case.th = FALSE,
 search.words = strsplit("(M|m)ethotrexate;(T|t)rexal;(R|r)heumatrex;(O|o)trexup", ";")[[1]],
 regex.sw = TRUE,
 ignore.case.sw = FALSE,
 eval.abbrevs = TRUE,
 out.table.format = ".csv (WINDOWS-1252)",
 write.table.locations = TRUE,
 write.tab.doc.file = TRUE,
 write.txt.doc.file = TRUE,
 exp.nondetc.tabs = TRUE,
 cpy_mv = "nocpymv",
 delete = TRUE)
}
```

PDE_install_Xpdftools4.02

Install the Xpdf command line tools 4.02

Description

PDE_install_Xpdftools4.02 downloads and installs the XPDF command line tools 4.02.

Usage

```
PDE_install_Xpdftools4.02(
  sysname = NULL,
  bin = NULL,
  verbose = TRUE,
  permission = 0
)
```

Arguments

sysname	String. In case the function returns "Unknown OS" the sysname can be set manually. Allowed options are "Windows", "Linux", "SunOS" for Solaris, and "Darwin" for Mac. Default: NULL.
bin	String. In case the function returns "Unknown OS" the bin of the operational system can be set manually. Allowed options are "64", and "32". Default: NULL.
verbose	Logical. Indicates whether messages will be printed in the console. Default: TRUE.
permission	Numerical. If set to 0 the user is ask for a permission to download Xpdftools. If set to 1, no user input is required. Default: 0.

Value

The function returns a Boolean for the installation status and a message in case the commands are not installed.

Examples

```
## Not run:

PDE_install_Xpdftools4.02()


## End(Not run)
```

PDE_path *Export the installation path the PDE (PDF Data Extractor) package*

Description

PDE_path is deprecated. Please run system.file(package = "PDE") instead.

Usage

```
PDE_path()
```

Value

The function returns a potential path for the PDE package. If the PDE tool was not correctly installed it returns "".

PDE_pdfs2table *Extracting all tables from a PDF (Portable Document Format) file*

Description

PDE_pdfs2table extracts all tables from a single PDF file and writes output in the corresponding folder.

Usage

```
PDE_pdfs2table(
  pdfs,
  out = ".",
  table.heading.words = "",
  ignore.case.th = FALSE,
  out.table.format = ".csv (WINDOWS-1252)",
  dev_x = 20,
  dev_y = 9999,
  write.table.locations = FALSE,
  exp.nondetc.tabs = TRUE,
  delete = TRUE,
  verbose = TRUE
)
```

Arguments

pdfs	String. A list of paths to the PDF files to be analyzed.
out	String. Directory chosen to save tables in. Default: ".".
table.heading.words	List of strings. Different than standard (TABLE, TAB or table plus number) headings to be detected. Regex rules apply (see also https://github.com/erikstricker/PDE/blob/master/inst/examples/cheatsheets/regex.pdf). Default = "".
ignore.case.th	Logical. Are the additional table headings (see table.heading.words) case-sensitive (does capitalization matter)? Default = FALSE.

`out.table.format`
: String. Output file format. Either comma separated file `.csv` or tab separated file `.tsv`. The encoding indicated in parantheses should be selected according to the operational system exported tables are opened in, i.e., Windows: `"(WINDOWS-1252)"`; Mac: `(macintosh)`; Linux: `(UTF-8)`. Default: `".csv"` and encoding depending on the operational system.

`dev_x`
: Numeric. For a table the size of indention which would be considered the same column. Default: `20`.

`dev_y`
: Numeric. For a table the vertical distance which would be considered the same row. Can be either a number or set to dynamic detection [9999], in which case the font size is used to detect which words are in the same row. Default: `9999`.

`write.table.locations`
: Logical. If `TRUE`, a separate file with the headings of all tables, their relative location in the generated html and txt files, as well as information if search words were found will be generated. Default: `FALSE`.

`exp.nondetc.tabs`
: Logical. If `TRUE`, if a table was detected in a PDF file but is an image or cannot be read, the page with the table with be exported as a png. Default: `FALSE`.

`delete`
: Logical. If `TRUE`, the intermediate **txt**, **keeplayouttxt** and **html** copies of the PDF file will be deleted. Default: `TRUE`.

`verbose`
: Logical. Indicates whether messages will be printed in the console. Default: `TRUE`.

See Also

PDE_extr_data_from_pdfs,PDE_pdfs2table_searchandfilter

Examples

```
## Running a simple table extraction
if(PDE_check_Xpdf_install() == TRUE){
outputtables <- PDE_pdfs2table(pdf = paste0(system.file(package = "PDE"),
                "/examples/Methotrexate/29973177_!.pdf"),
 out = paste0(system.file(package = "PDE"),"/examples/29973177_tables/"))
}

## Running a the same table extraction as above with all paramaters shown
if(PDE_check_Xpdf_install() == TRUE){
 outputtables <- PDE_pdfs2table(pdf = paste0(system.file(package = "PDE"),
                                "/examples/Methotrexate/29973177_!.pdf"),
 out = paste0(system.file(package = "PDE"),"/examples/29973177_tables/"),
 dev_x = 20,
 dev_y = 9999,
 table.heading.words = "",
 ignore.case.th = FALSE,
 out.table.format = ".csv (WINDOWS-1252)",
 write.table.locations = FALSE,
 exp.nondetc.tabs = FALSE,
 delete = TRUE)
```

```
}
```

`PDE_pdfs2table_searchandfilter` *Extracting tables from a PDF (Portable Document Format) file*

Description

`PDE_pdfs2table_searchandfilter` extracts tables from a single PDF file according to filter and search words and writes output in the corresponding folder.

Usage

```
PDE_pdfs2table_searchandfilter(
  pdfs,
  out = ".",
  filter.words = "",
  regex.fw = TRUE,
  ignore.case.fw = FALSE,
  filter.word.times = "0.2%",
  table.heading.words = "",
  ignore.case.th = FALSE,
  search.words,
  search.word.categories = NULL,
  save.tab.by.category = FALSE,
  regex.sw = TRUE,
  ignore.case.sw = FALSE,
  eval.abbrevs = TRUE,
  out.table.format = ".csv (WINDOWS-1252)",
  dev_x = 20,
  dev_y = 9999,
  write.table.locations = FALSE,
  exp.nondetc.tabs = TRUE,
  write.tab.doc.file = TRUE,
  delete = TRUE,
  cpy_mv = "nocpymv",
  verbose = TRUE
)
```

Arguments

`pdfs`	String. A list of paths to the PDF files to be analyzed.
`out`	String. Directory chosen to save analysis results in. Default: `"."`.
`filter.words`	List of strings. The list of filter words. If not `NA` or `""` a hit will be counted every time a word from the list is detected in the article. Default: `""`.

regex.fw Logical. If TRUE filter words will follow the regex rules (see https://github.com/erikstricker/PDE/blob/master/inst/examples/cheatsheets/regex.pdf). Default = TRUE.

ignore.case.fw Logical. Are the filter words case-sensitive (does capitalization matter)? Default: FALSE.

filter.word.times
Numeric or string. Can either be expressed as absolute number or percentage of the total number of words (by adding the " filter.words for a paper to be further analyzed. Default: 0.2%.

table.heading.words
List of strings. Different than standard (TABLE, TAB or table plus number) headings to be detected. Regex rules apply (see also https://github.com/erikstricker/PDE/blob/master/inst/examples/cheatsheets/regex.pdf). Default = "".

ignore.case.th Logical. Are the additional table headings (see table.heading.words) case-sensitive (does capitalization matter)? Default = FALSE.

search.words List of strings. List of search words. To extract all tables from the PDF file leave search.words = "".

search.word.categories
List of strings. List of categories with the same length as the list of search words. Accordingly, each search word can be assigned to a category, of which the word counts will be summarized in the PDE_analyzer_word_stats.csv file. If search.word.categories is a different length than search.words the parameter will be ignored. Default: NULL.

save.tab.by.category
Logical. Can only be used with search.word.categories. If set to TRUE, tables that carry search words will be saved in sub-folders according to the search word category of the detected search word. Default: FALSE.

regex.sw Logical. If TRUE search words will follow the regex rules (see https://github.com/erikstricker/PDE/blob/master/inst/examples/cheatsheets/regex.pdf). Default = TRUE.

ignore.case.sw Logical. Are the search words case-sensitive (does capitalization matter)? Default: FALSE.

eval.abbrevs Logical. Should abbreviations for the search words be automatically detected and then replaced with the search word + "$*"? Default: TRUE.

out.table.format
String. Output file format. Either comma separated file .csv or tab separated file .tsv. The encoding indicated in parantheses should be selected according to the operational system exported tables are opened in, i.e., Windows: "(WINDOWS-1252)"; Mac: (macintosh); Linux: (UTF-8). Default: ".csv" and encoding depending on the operational system.

dev_x Numeric. For a table the size of indention which would be considered the same column. Default: 20.

dev_y Numeric. For a table the vertical distance which would be considered the same row. Can be either a number or set to dynamic detection [9999], in which case the font size is used to detect which words are in the same row. Default: 9999.

write.table.locations
: Logical. If TRUE, a separate file with the headings of all tables, their relative location in the generated html and txt files, as well as information if search words were found will be generated. Default: FALSE.

exp.nondetc.tabs
: Logical. If TRUE, if a table was detected in a PDF file but is an image or cannot be read, the page with the table with be exported as a png. Default: TRUE.

write.tab.doc.file
: Logical. If TRUE, if search words are used for table detection and no search words were found in the tables of a PDF file, a **no.table.w.search.words**. Default: TRUE.

delete
: Logical. If TRUE, the intermediate **txt**, **keeplayouttxt** and **html** copies of the PDF file will be deleted. Default: TRUE.

cpy_mv
: String. Either "nocpymv", "cpy", or "mv". If filter words are used in the analyses, the processed PDF files will either be copied ("cpy") or moved ("mv") into the /pdf/ subfolder of the output folder. Default: "nocpymv".

verbose
: Logical. Indicates whether messages will be printed in the console. Default: TRUE.

Value

If tables were extracted from the PDF file the function returns a list of following tables/items: 1) **htmltablelines**, 2) **txttablelines**, 3) **keeplayouttxttablelines**, 4) **id**, 5) **out_msg**. The **tablelines** are tables that provide the heading and position of the detected tables. The **id** provide the name of the PDF file. The **out_msg** includes all messages printed to the console or the suppressed messages if verbose=FALSE.

See Also

PDE_extr_data_from_pdfs, PDE_pdfs2table

Examples

```
## Running a simple analysis with filter and search words to extract tables
if(PDE_check_Xpdf_install() == TRUE){
 outputtables <- PDE_pdfs2table_searchandfilter(pdf = paste0(system.file(package = "PDE"),
                                  "/examples/Methotrexate/29973177_!.pdf"),
 out = paste0(system.file(package = "PDE"),"/examples/29973177_tables/"),
 filter.words = strsplit("cohort;case-control;group;study population;study participants", ";")[[1]],
 regex.fw = FALSE,
 ignore.case.fw = TRUE,
 search.words = strsplit("(M|m)ethotrexate;(T|t)rexal;(R|r)heumatrex;(O|o)trexup", ";")[[1]],
 regex.sw = TRUE,
 ignore.case.sw = FALSE)
}

## Running an advanced analysis with filter and search words to
## extract tables and obtain documentation files
if(PDE_check_Xpdf_install() == TRUE){
```

```
outputtables <- PDE_pdfs2table_searchandfilter(pdf = paste0(system.file(package = "PDE"),
                                    "/examples/Methotrexate/29973177_!.pdf"),
out = paste0(system.file(package = "PDE"),"/examples/29973177_tables/"),
dev_x = 20,
dev_y = 9999,
filter.words = strsplit("cohort;case-control;group;study population;study participants", ";")[[1]],
regex.fw = FALSE,
ignore.case.fw = TRUE,
filter.word.times = "0.2%",
table.heading.words = "",
ignore.case.th = FALSE,
search.words = strsplit("(M|m)ethotrexate;(T|t)rexal;(R|r)heumatrex;(O|o)trexup", ";")[[1]],
regex.sw = TRUE,
ignore.case.sw = FALSE,
eval.abbrevs = TRUE,
out.table.format = ".csv (WINDOWS-1252)",
write.table.locations = TRUE,
write.tab.doc.file = TRUE,
exp.nondetc.tabs = TRUE,
cpy_mv = "nocpymv",
delete = TRUE)
}
```

PDE_pdfs2txt_searchandfilter

Extracting sentences from a PDF (Portable Document Format) file

---

Description

PDE_pdfs2txt_searchandfilter extracts sentences from a single PDF file according to search and filter words and writes output in the corresponding folder.

Usage

```
PDE_pdfs2txt_searchandfilter(
  pdfs,
  out = ".",
  filter.words = "",
  regex.fw = TRUE,
  ignore.case.fw = FALSE,
  filter.word.times = "0.2%",
  search.words,
  search.word.categories = NULL,
  regex.sw = TRUE,
  ignore.case.sw = FALSE,
  eval.abbrevs = TRUE,
  out.table.format = ".csv (WINDOWS-1252)",
  context = 0,
```

```
  write.txt.doc.file = TRUE,
  delete = TRUE,
  cpy_mv = "nocpymv",
  verbose = TRUE
)
```

Arguments

pdfs	String. A list of paths to the PDF files to be analyzed.
out	String. Directory chosen to save analysis results in. Default: ".".
filter.words	List of strings. The list of filter words. If not NA or "" a hit will be counted every time a word from the list is detected in the article. Default: "".
regex.fw	Logical. If TRUE filter words will follow the regex rules (see https://github.com/erikstricker/PDE/blob/master/inst/examples/cheatsheets/regex.pdf). Default = TRUE.
ignore.case.fw	Logical. Are the filter words case-sensitive (does capitalization matter)? Default: FALSE.
filter.word.times	Numeric or string. Can either be expressed as absolute number or percentage of the total number of words (by adding the " filter.words for a paper to be further analyzed. Default: 0.2%.
search.words	List of strings. List of search words.
search.word.categories	List of strings. List of categories with the same length as the list of search words. Accordingly, each search word can be assigned to a category, of which the word counts will be summarized in the PDE_analyzer_word_stats.csv file. If search.word.categories is a different length than search.words the parameter will be ignored. Default: NULL.
regex.sw	Logical. If TRUE search words will follow the regex rules (see https://github.com/erikstricker/PDE/blob/master/inst/examples/cheatsheets/regex.pdf). Default = TRUE.
ignore.case.sw	Logical. Are the search words case-sensitive (does capitalization matter)? Default: FALSE.
eval.abbrevs	Logical. Should abbreviations for the search words be automatically detected and then replaced with the search word + "$*"? Default: TRUE.
out.table.format	String. Output file format. Either comma separated file .csv or tab separated file .tsv. The encoding indicated in parantheses should be selected according to the operational system exported tables are opened in, i.e., Windows: "(WINDOWS-1252)"; Mac: (macintosh); Linux: (UTF-8). Default: ".csv" and encoding depending on the operational system.
context	Numeric. Number of sentences extracted before and after the sentence with the detected search word. If 0 only the sentence with the search word is extracted. Default: 0.

write.txt.doc.file
: Logical. If TRUE, if no search words were found in the sentences of a PDF file, a file will be created with the PDF filename followed by **no.txt.w.search.words**. If the PDF file is empty, a file will be created with the PDF filename followed by **no.content.detected**. If the filter word threshold is not met, a file will be created with the PDF filename followed by **no.txt.w.filter.words**. Default: TRUE.

delete
: Logical. If TRUE, the intermediate **txt**, **keeplayouttxt** and **html** copies of the PDF file will be deleted. Default: TRUE.

cpy_mv
: String. Either "nocpymv", "cpy", or "mv". If filter words are used in the analyses, the processed PDF files will either be copied ("cpy") or moved ("mv") into the /pdf/ subfolder of the output folder. Default: "nocpymv".

verbose
: Logical. Indicates whether messages will be printed in the console. Default: TRUE.

See Also

PDE_extr_data_from_pdfs

Examples

```
## Running a simple analysis with filter and search words to extract sentences
if(PDE_check_Xpdf_install() == TRUE){
 outputtables <- PDE_pdfs2txt_searchandfilter(pdf = paste0(system.file(package = "PDE"),
                                      "/examples/Methotrexate/29973177_!.pdf"),
 out = paste0(system.file(package = "PDE"),"/examples/MTX_txt+-0/"),
 filter.words = strsplit("cohort;case-control;group;study population;study participants", ";")[[1]],
 regex.fw = FALSE,
 ignore.case.fw = TRUE,
 search.words = strsplit("(M|m)ethotrexate;(T|t)rexal;(R|r)heumatrex;(O|o)trexup", ";")[[1]],
 regex.sw = TRUE,
 ignore.case.sw = FALSE)
}

## Running an advanced analysis with filter and search words to
## extract sentences and obtain documentation files
if(PDE_check_Xpdf_install() == TRUE){
 outputtables <- PDE_pdfs2txt_searchandfilter(pdf = paste0(system.file(package = "PDE"),
                                       "/examples/Methotrexate/29973177_!.pdf"),
 out = paste0(system.file(package = "PDE"),"/examples/MTX_txt+-1/"),
 context = 1,
 filter.words = strsplit("cohort;case-control;group;study population;study participants", ";")[[1]],
 regex.fw = FALSE,
 ignore.case.fw = TRUE,
 filter.word.times = "0.2%",
 search.words = strsplit("(M|m)ethotrexate;(T|t)rexal;(R|r)heumatrex;(O|o)trexup", ";")[[1]],
 regex.sw = TRUE,
 ignore.case.sw = FALSE,
 eval.abbrevs = TRUE,
 out.table.format = ".csv (WINDOWS-1252)",
 write.txt.doc.file = TRUE,
 cpy_mv = "nocpymv",
```

```
 delete = TRUE)
}
```

PDE_reader_i *Browsing the PDE (PDF Data Extractor) analyzer results.*

Description

The PDE_reader_i allows the user-friendly visualization and quick-processing of the obtained results.

Usage

```
PDE_reader_i(verbose = TRUE)
```

Arguments

verbose	Logical. Indicates whether messages will be printed in the console. Default: TRUE.

Note

A detailed description of the elements in the user interface can be found in the markdown file (README_PDE.md)

Examples

```
PDE_reader_i()
```

Index