Package ‘orderanalyzer’

December 12, 2024

Type Package

Title Extracting Order Position Tables from PDF-Based Order Documents

Version 1.0.0

Date 2024-12-11

Maintainer Michael Scholz <michael.scholz@th-deg.de>

Description Functions for extracting text and tables from PDF-based order documents. It provides an n-gram-based approach for identifying the language of an order document. It furthermore uses R-package 'pdftools' to extract the text from an order document. In the case that the PDF document is only including an image (because it is scanned document), R package 'tesseract' is used for OCR. Furthermore, the package provides functionality for identifying and extracting order position tables in order documents based on a clustering approach.

License GPL-3

SystemRequirements Tesseract >= 5.0.0, libtesseract-dev (deb), tesseract-devel (rpm), libleptonica-dev (deb), leptonica-devel (rpm), tesseract-ocr-eng (deb), libpoppler-cpp-dev (deb), poppler-cpp-devel (rpm), poppler-data (rpm/deb), libxml2-dev (deb), libxml2-devel (rpm)

Depends R(>= 4.3.0), tidyselect

Imports data.table, dplyr, matrixcalc, quanteda, rlist, stringr, tibble, tidyr, utils, purrr, digest, lubridate

Suggests pdftools, tesseract, xml2

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Michael Scholz [cre, aut], Joerg Bauer [aut]

Repository CRAN

Date/Publication 2024-12-12 15:20:02 UTC

Contents

`orderanalyzer-package` *Extracting order position tables from PDF-based order documents*

Description

This packages provides functions for extracting text and order-position-tables from PDF-based order documents.

Details

Package:	orderanalyzer
Type:	Package
Version:	1.0.0
Date:	2024-12-11
License:	GPL-3
Depends:	R (>= 4.3.0)

Author(s)

Michael Scholz <`michael.scholz@th-deg.de`>

Joerg Bauer <`joerg.bauer@th-deg.de`>

`extractTables` *Extract tables from a given words-dataframe*

Description

This function extracts order-position-tables from PDF-based order documents. It tries to identify table rows based on a clustering approach and thereafter identifies the column structure. A table row can consist of multiple text rows and the text rows can span different columns. This function furthermore tries to identify the meaning of the columns (position, articleID, description, quantity, quanity unit, unit price, total price, currency, date).

Usage

```
extractTables(text, minCols = 3, maxDistance = 20, entityNames = NA)
```

Arguments

`text` List including several representations of text extracted from a PDF file. This list is generated by the function extractText.

`minCols` Number of columns a table must minimal consist of

`maxDistance` Number of text lines that can maximally exist between the start of two table rows

`entityNames` A list of four name vectors (currencyUnits, quantityUnits, headerNames, noTableNames). Each vector contains strings that correspond to currency units, quantity units, header names or names of entities not being a table.

Value

List of lists describing the tables. Each sublist includes a data frame (data) which is the identified table, the position of text lines that constitute the table and the position of the significant lines.

Examples

```
file <- system.file("extdata", "OrderDocument_en.pdf", package = "orderanalyzer")
text <- extractText(file)

# Extracting order tables without any further information
tables <- extractTables(text)
tables[[1]]$data

# Extracting order tables with further information
tables <- extractTables(text,
  entityNames = list(currencyUnits = enc2utf8(c("eur", "euro", "\u20AC")),
                     quantityUnits = enc2utf8(c("pcs", "pcs.")),
                     headerNames = enc2utf8(c("pos", "item", "quantity")),
                     noTableNames = enc2utf8(c("order total", "supplier number")))
)
tables[[1]]$data

# Extracting order tables from a German document
file <- system.file("extdata", "OrderDocument_de.pdf", package = "orderanalyzer")
text <- extractText(file)
tables <- extractTables(text)
tables[[1]]$data
```

`extractText` *Extracts the text from a PDF file*

Description

This function extracts text from PDF documents and returns the text as a string, as a list of lines and as a list of words. It uses 'pdftools' to extract the content from textual PDF files and 'tesseract' to extract the content from image-based PDF-files.

Usage

```
extractText(file)
```

Arguments

`file`	Path to the PDF file

Value

List including the extracted text, a data table including the lines, a data table including the words, the type and language of the document.

Examples

```
file <- system.file("extdata", "OrderDocument_en.pdf", package = "orderanalyzer")
text <- extractText(file)
text$words
```

`identifyLanguage` *Identifies the language of a given text based on frequent trigrams*

Description

This function identifies the language of a given string based on the most frequent trigrams in different languages. Supported languages are Czech, Dutch, English, French, German, Spanish, Latvian and Lithuanian.

Usage

```
identifyLanguage(text)
```

Arguments

`text`	String for which the language should be identified

Value

Name of the detected language.

Examples

```
text <- "The tea in the cup still is hot."
language <- identifyLanguage(text)
language
```

Index