Package ‘metaextractoR’

September 3, 2025

Type Package

Title What the Package Does (Title Case)

Version 0.1.0

Description This is an R package that utilises Large Language Models to assist with data extraction during meta-analysis. This package incorporates three modular Shiny apps to implement a human-in-the-loop framework. (1) a manual extraction interface for abstracts, (2) refining prompt engineering and model selection, and (3) validation of LLM-generated outputs. These apps enable researchers to iteratively collaborate with LLMs, with each abstract undergoing double data extraction—once manually by a human researcher and once independently by an LLM-assisted process—to emulate the double extraction process recommended by international standards. Notably, the package runs fully on local machines, with no need for API setup or external data transfer, maximising data privacy and accessibility. Robust logging features further enhance transparency and reproducibility by recording all prompt iterations and outputs.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Depends R (>= 2.10)

Imports ellmer,
shiny,
bslib,
DT,
shinyjs,
tidyr,
shinyFiles,
cli,
vctrs,
dplyr,
purrr,

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Contents

`abstracts` *Sample Dataset: abstracts*

Description

This dataset includes sample abstracts downloaded from one of my Covidence project. Pre-processing was done by running the clean_name function from janitor package.

Usage

```
abstracts
```

Format

A data frame with 20 rows and 16 variables:

title Title of the manuscript

authors Authors

abstract The column that contains all abstracts

published_year Year of the publication

published_month Month of the publication

journal Journal's name

volume Volume number

issue Issue number

pages Page number

accession_number Accession number

doi doi number

ref Reference

covidence_number Covidence number

study Study

notes Notes

tags Tags

Source

Created for demonstration purposes

`add_predefined_vars` *Pre-processing functions before shinyapps*

Description

Pre-processing functions and saving intermediate data

Usage

```
add_predefined_vars(data, list_vars)
```

Arguments

`data`	a csv file contains abstract information. This could be the csv file downloaded from covidence
`list_vars`	a vector of data elements you want to extract. i.e. c("no_participants,"no_female",".."")

Value

new dataset with additional empty columns

`glance_manual_app` *Launch the first shinyapp for to look into what variables are available in the abstract.*

Description

This function calls a shinyapp that display the abstract and allow for randomly reading in data to glace what are the available information in abstract.

Usage

```
glance_manual_app()
```

Value

Shiny app.

manual_validation_app *manual_validation_app*

Description

A shinyapp that manually check the LLM outputs

Usage

```
manual_validation_app()
```

Examples

```
## Not run:
manual_validation_app()

## End(Not run)
```

process_with_ollama *process_with_ollama*

Description

A function that batch process data extraction from text.

Usage

```
process_with_ollama(input, model = "llama3.1:8b", type_abstract, i)
```

Arguments

`input`	A data frame contains abstracts and variables we want to extract.
`model`	Large Language Model name you want to use i.e. "llama3.1:8b"
`type_abstract`	is created using the type_object function from the ellmer package. Objects represent a collection of named values and are created with type_object(). Objects can contain any number of scalars, arrays, and other objects. They are similar to named lists in R.
`i`	number of abstracts you want to process at once.

Examples

```
#example code
type_abstract <- ellmer::type_object(
"Some key information from abstract.",
no_patients_llm = ellmer::type_integer("Find the total number of patients included in this study.",required = F
no_AKI_llm = ellmer::type_integer("Number of Acute Kidney Injury (AKI) patients included in this study.", requi
per_AKI_llm = ellmer::type_number("Percentage of Acute Kidney Injury ",required = FALSE),
ICU_llm = ellmer::type_boolean("Included only Intensive Care Unit (ICU) patients.",required = FALSE),
start_date = ellmer::type_string("The starting date of the study written in YYYY-MM-DD format",required = FALSE
end_date = ellmer::type_string("The end date of the study written in YYYY-MM-DD format",required = FALSE),
age_mean_llm = ellmer::type_number("Find the average age of the study cohort",required = FALSE),
age_median_llm = ellmer::type_number("Find the median age of the study cohort",required = FALSE),
)


## Not run:
process_with_ollama()

## End(Not run)
```

`prompt_engineering_app` *prompt_engineering_app*

Description

A shinyapp that designed for model testing and prompt engineering.

Usage

```
prompt_engineering_app()
```

Examples

```
## Not run:
prompt_engineering_app()

## End(Not run)
```

`save_testing_data` *save_testing_data*

Description

This function will save the testing abstract data with empty columns to the processed data folder. This csv file will be used in the shinyapp 2 The data will be stored in metaextroctor_process_data

Usage

```
save_testing_data(testing_abs)
```

Arguments

`testing_abs`	training abstracts including the variables you want to extract.

Value

a csv file saved in the metaextroctor_process_data file named training_stage_0_data.csv

save_training_data *save_training_data*

Description

This function will save the training abstract data with empty columns to the processed data folder. This csv file will be used in the shinyapp 1 The data will be stored in metaextroctor_process_data

Usage

```
save_training_data(training_abs)
```

Arguments

`training_abs`	training abstracts including the variables you want to extract.

Value

a csv file saved in the metaextroctor_process_data file named training_stage_0_data.csv

separate_training *separate_training*

Description

This function will separate the abstracts into training and testing sets.

Usage

```
separate_training(data, percentage = 0.1)
```

Arguments

`data`	The csv file contains abstracts with
`percentage`	percentage of separation training sets.

Index