Package 'rosettaR'

January 14, 2026

Type Package

Title Turn Plain Language Statements Into RDF, CSV, or Anything

Version 0.1.0

Description Core functions to work with Rosetta Statements, which use templates to convert plain language strings into RDF, CSV or other representations. This package is designed to make the creation of knowledge graphs easier.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports dplyr, jinjar, shiny, utils, reticulate, jsonlite, rlang, digest

Depends R (>= 3.5)

Suggests knitr, rmarkdown, testthat (>= 3.0.0), shinytest2

VignetteBuilder knitr

URL https://timalamenciak.github.io/rosettaR/

BugReports https://github.com/timalamenciak/rosettaR/issues

Config/testthat/edition 3

NeedsCompilation no

Author Tim Alamenciak [aut, cre], Tarek Al Mustafa [aut]

Maintainer Tim Alamenciak <tim.alamenciak@gmail.com>

Contents

add_template *Add a template string to the library*

Description

Add a template string to the library

Usage

```
add_template(library, template)
```

Arguments

library	Input must be a library dataframe created by initLibrary()
template	This should be one string template.

Value

A new library data frame containing the template that was added.

Examples

```
templates <- init_library()
templates <- add_template(templates, apple_template)
```

apple_csv *Apple Output Template Example This is an example of an output template that reformats the statement as a CSV.*

Description

Apple Output Template Example This is an example of an output template that reformats the statement as a CSV.

Usage

```
apple_csv
```

Format

apple_csv:
A string that contains Jinja syntax placeholders that match the apple_template.

References

https://arxiv.org/pdf/2407.20007

`apple_statement` *Apple Rosetta Statement Example This is an example of a statement that describes the weight of Apple X.*

Description

Apple Rosetta Statement Example This is an example of a statement that describes the weight of Apple X.

Usage

```
apple_statement
```

Format

`apple_statement`:

A string that specifies an object, a quality, a measurement and a unit.

References

https://arxiv.org/pdf/2407.20007

`apple_template` *Apple Rosetta Statement Example This is an example of a template to describe the weight of a particular apple.*

Description

Apple Rosetta Statement Example This is an example of a template to describe the weight of a particular apple.

Usage

```
apple_template
```

Format

`apple_template`:

A string with slots for an object, a quality, a measurement and a unit.

References

https://arxiv.org/pdf/2407.20007

df_to_statements *Change a data frame into a series of statements based on templates.*

Description

Change a data frame into a series of statements based on templates.

Usage

```
df_to_statements(df, templates)
```

Arguments

df	A data frame with headers that match the slots in a template. The dataframe must have one column called "TemplateID" that contains the ID of the template used to interpret the row.
templates	A Rosetta Statement template library created by the initLibrary function.

Value

A data frame statement library containing the statements and associated templates.

Examples

```
templates <- init_library()
templates <- add_template(templates, apple_template)
df <- data.frame(object = "Apple X",
                              quality = "weight", value="100.2",
                             unit="grams", TemplateID = "1")
statements <- df_to_statements(df,templates)
```

init_library *Load a set of Rosetta templates into a dataframe.*

Description

Load a set of Rosetta templates into a dataframe.

Usage

```
init_library(file = NA)
```

Arguments

file	A CSV with the headers 'TemplateID' and 'templateText'. Headers are case-sensitive and required. Other functions like `df_to_statements()` and `add_template()` will expect a data frame in the format created by this function.

Value

A dataframe containing either the loaded templates or the proper headers for use with other functions.

Examples

```
#Initialize an empty library
templates <- init_library()

#Load a CSV library
apple_templates <- init_library(system.file("extdata", "apple_templates.csv",
 package="rosettaR"))
```

`rosetta_format` *Parse a Rosetta Statement*

Description

Converts a plain language statement into a structured dataframe using a template. Supports variables {{ var }} and optional blocks [...].

Usage

```
rosetta_format(s, in_template, out_template = "df")
```

Arguments

`s`	The input statement string.
`in_template`	The Rosetta template string.
`out_template`	The output format ('df' for dataframe, 'rdf' for generic Turtle, or a Jinja string).

Value

Either a data frame or the output template with values filled in from statement and input template.

Examples

```
# example code
rosetta_format("Apple X weighs 235 grams", "{{ fruit }} weighs {{ value }} {{ unit}}")
```

rosetta_match *Match Multiple Statements Against a Template Library*

Description

Iterates through a vector of statements and attempts to match them against a library of templates. Returns a long-format dataframe.

Usage

```
rosetta_match(statements, templates)
```

Arguments

statements	A character vector of plain language statements, OR a dataframe containing a column of statements.
templates	A dataframe created by `init_library()`, containing 'TemplateID' and 'templateText' columns.

Value

A data.frame in long format with columns: `statement_id`, `statement_text`, `template_id`, variable, value.

Examples

```
## Not run:
   # Vector Input
   statements <- c("Kitchener is in Canada", "Paris is in France")

   # Dataframe Input (e.g. from read.csv)
   # df <- read.csv("my_statements.csv")

   templates <- init_library(system.file("extdata/apple_templates.csv",
    package="rosettaR"))
   results <- rosetta_match(statements, templates)

## End(Not run)
```

rosetta_triplify *Batch Convert Statements to RSO RDF*

Description

Converts a list of statements into a single, cohesive Turtle (TTL) string using the Rosetta Statement Ontology (RSO). Handles prefix management automatically.

Usage

```
rosetta_triplify(statements, templates)
```

Arguments

statements	A character vector of plain language statements.
templates	A dataframe created by `init_library()`.

Value

A single character string containing the full RDF document.

Examples

```
## Not run:
  # (Use the same setup as above)
  ttl <- rosetta_triplify(statements, templates)
  cat(ttl)

## End(Not run)
```

---

rosetta_validate *Validate a data frame against a LinkML schema*

Description

Validate a data frame against a LinkML schema

Usage

```
rosetta_validate(data, schema, target_class = NULL)
```

Arguments

data	A data frame (e.g. from 'rosetta_format) or list.
schema	Path to the LinkML schema YAML file.
target_class	The class in the schema to validate against.

Value

A list containing ok (boolean) and `issues` (list of errors).

Examples

```
## Not run:
  # Create a YAML schema for this example.
  schema_yaml <- "id: https://example.org/apple-schema
  name: AppleSchema
   imports:
   - linkml:types
   default_range: string
 classes:
   AppleObservation:
    attributes:
      object:
        required: true
```

```
      value:
        range: float  # Value must be a number!
      unit:
        range: string"

# Save to a temporary file for this example
schema_file <- tempfile(fileext = ".yaml")
writeLines(schema_yaml, schema_file)# (Use the same setup as above)

good_data <- data.frame(object = "Apple A", value = 150.5, unit = "g")
res_good <- rosetta_validate(good_data, schema_file,
  target_class = "AppleObservation")
print(paste("Good Data is Valid:", res_good$ok))

## End(Not run)
```

`run_rosetta_ui` *Launch the RosettaR UI*

Description

Opens a Shiny application to interactively test Rosetta templates and validation.

Usage

```
run_rosetta_ui()
```

Value

A character string containing the processed text (if `out_template` is text), or a dataframe (if `out_template` is "df").

Examples

```
## Not run:
run_rosetta_ui()

## End(Not run)
```