Package 'lang'

June 11, 2026

Title Translates R Help Documentation using Large Language Models

Version 0.1.2

Description Translates R help documentation on the fly by using a Large Language model of your choice. If you are using 'RStudio' or 'Positron' the translated help will appear in the help pane.

License MIT + file LICENSE

URL https://mlverse.github.io/lang/, https://github.com/mlverse/lang

BugReports https://github.com/mlverse/lang/issues

Depends R (>= 4.1)

Imports callr, cli, fs, glue, lifecycle, mall, rlang (>= 1.1.0), rstudioapi, tools, withr

Suggests ellmer, testthat (>= 3.0.0)

Config/testthat/edition 3

Config/usethis/last-upkeep 2025-11-10

Encoding UTF-8

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Edgar Ruiz [aut, cre], Posit Software, PBC [cph, fnd] (ROR: <https://ror.org/03wc8by49>)

Maintainer Edgar Ruiz <edgar@posit.co>

Repository CRAN

Date/Publication 2026-06-11 07:00:07 UTC

Contents

help 2
lang_help 2
lang_use 3

Index **6**

help *Drop-in replacements for help and ? functions*

Description

The ? and help functions are drop-in replacements for those in the utils package. They automatically translate help to the language specified by the LANG or LANGUAGE environment variables, or by lang_use().

Usage

```
# help(topic, package = NULL, ...)

# ?e2
# e1?e2
```

Arguments

topic	A name or character string specifying the help topic.
package	A name or character string specifying the package in which to search for the help topic. If NULL, search all packages.
...	Additional arguments to pass to utils::help().
e1	First argument to pass along to utils::¿‘.
e2	Second argument to pass along to utils::¿‘.

lang_help *Translates help documentation to another language*

Description

Translates a given topic into a target language. It uses the lang argument to determine which language to translate to. If not passed, this function will look for a target language in the LANG and LANGUAGE environment variables, or if something has been passed to the .lang argument in lang_use(), to determine the target language. If the target language is English, no translation will be processed, so the help returned will be the original package's documentation.

To improve translation consistency, lang generates a short summary of the full help page and passes it as context with each individual translation request. This helps the LLM maintain consistent terminology across sections. You can control the length of this summary with the context_size argument, or set it to 0 to disable context-aware translation entirely. To avoid the LLM getting confused when the context is much longer than the field being translated, context is automatically omitted for fields of 10 words or fewer.

Usage

```
lang_help(
  topic,
  package = NULL,
  lang = NULL,
  context_size = NULL,
  type = getOption("help_type")
)
```

Arguments

topic	A character string specifying the help topic to translate.
package	The R package to look for the topic, if not provided the function will attempt to find the topic based on the loaded packages.
lang	A character vector language to translate the topic to
context_size	Maximum number of words for the context summary included with each translation request. Set to `0` to disable context-aware translation. When `NULL`, the value set via `lang_use()` is used (default `100`).
type	Produce "html" or "text" output for the help. It defaults to `getOption("help_type")`

Value

Original or translated version of the help documentation in the output type specified

Examples

```
## Not run:
# Requires an interactive session with Ollama running locally
library(lang)

lang_use("ollama", "llama3.2", seed = 100)

lang_help("lang_help", lang = "spanish", type = "text")

## End(Not run)
```

---

lang_use *Specifies the LLM provider and model to use during the R session*

Description

Specifies the back-end provider and model to use during the current R session. The target language is not processed by the function, as in converting "english" to "en" for example. The value is passed directly to the LLM, and it lets the LLM interpret the target language.

Usage

```
lang_use(
  backend = NULL,
  model = NULL,
  .cache = NULL,
  .lang = NULL,
  .context_size = NULL,
  .silent = FALSE,
  ...
)
```

Arguments

backend	"ollama" or an `ellmer` `Chat` object. If using "ollama", `lang` provides built-in support via the `ollamar` package. Defaults to "ollama".
model	The name of model supported by the back-end provider
.cache	Character path where translations are cached. Set to `""` to disable caching. When `NULL`, the current session value is kept unchanged. Defaults to a temporary folder.
.lang	Target language to translate to. This will override values found in the LANG and LANGUAGE environment variables.
.context_size	Maximum number of words for the context summary included with each translation request. Set to `0` to disable context-aware translation. Defaults to `100`.
.silent	Boolean flag that controls whether there is output to the console. Defaults to FALSE.
...	Additional arguments that this function will pass down to the integrating function. In the case of Ollama, it will pass those arguments to `ollamar::chat()`.

Value

Invisibly returns `NULL`. Prints the current configuration to the console.

Examples

```
## Not run:
# Requires an interactive session with Ollama or another LLM provider
library(lang)

# Using an `ellmer` chat object
lang_use(ellmer::chat_openai(model = "gpt-4o"))

# Using Ollama directly
lang_use("ollama", "llama3.2", seed = 100)

# Turn off cache by setting `.cache` to ""
lang_use("ollama", "llama3.2", seed = 100, .cache = "")

# Use `.lang` to set the target language to translate to,
```

```
# it will be set for the current R session
lang_use("ollama", "llama3.2", .lang = "spanish")

# Use `.silent` to avoid console output
lang_use("ollama", "llama3.2", .lang = "spanish", .silent = TRUE)

# To see current settings, simply call the function
lang_use()

## End(Not run)
```

Index

? *(*help*)*, 2

help, 2

lang_help, 2
lang_use, 3

shim_lang_help *(*help*)*, 2
shim_lang_question *(*help*)*, 2

utils::help(), *2*