

Package ‘ollamar’

May 2, 2024

Title 'Ollama' Language Models

Version 1.1.1

Description

An interface to easily run local language models with 'Ollama' <<https://ollama.com>> server and API endpoints (see <<https://github.com/ollama/ollama/blob/main/docs/api.md>> for details). It lets you run open-source large language models locally on your machine.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.1

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Imports glue, httr2, jsonlite, tibble

BugReports <https://github.com/hauselin/ollama-r/issues>

URL <https://hauselin.github.io/ollama-r/>,
<https://github.com/hauselin/ollama-r>

NeedsCompilation no

Author Hause Lin [aut, cre, cph] (<<https://orcid.org/0000-0003-4590-7039>>)

Maintainer Hause Lin <hauselin@gmail.com>

Repository CRAN

Date/Publication 2024-05-02 15:12:40 UTC

R topics documented:

chat	2
create_request	3
delete	3
embeddings	4
generate	5
list_models	6

package_config	6
pull	7
resp_process	7
test_connection	8

Index	9
--------------	----------

chat	<i>Chat with Ollama models</i>
------	--------------------------------

Description

Chat with Ollama models

Usage

```
chat(
  model,
  messages,
  stream = FALSE,
  output = c("resp", "jsonlist", "raw", "df"),
  endpoint = "/api/chat"
)
```

Arguments

model	A character string of the model name such as "llama3".
messages	A list with list of messages for the model (see examples below).
stream	Enable response streaming. Default is FALSE.
output	The output format. Default is "resp". Other options are "jsonlist", "raw", "df".
endpoint	The endpoint to chat with the model. Default is "/api/chat".

Value

A httr2 response object, json list, raw or data frame.

Examples

```
# one message
messages <- list(
  list(role = "user", content = "How are you doing?")
)
chat("llama3", messages)
chat("llama3", messages, stream = TRUE)
chat("llama3", messages, stream = TRUE, output = "df")

# multiple messages
```

```
messages <- list(  
  list(role = "user", content = "Hello!"),  
  list(role = "assistant", content = "Hi! How are you?"),  
  list(role = "user", content = "Who is the prime minister of the uk?"),  
  list(role = "assistant", content = "Rishi Sunak"),  
  list(role = "user", content = "List all the previous messages.")  
)  
chat("llama3", messages, stream = TRUE)
```

create_request	<i>Create a htr2 request object.</i>
----------------	--------------------------------------

Description

Creates a htr2 request object with base URL, headers and endpoint. Used by other functions in the package and not intended to be used directly.

Usage

```
create_request(endpoint)
```

Arguments

endpoint	The endpoint to create the request
----------	------------------------------------

Value

A htr2 request object.

Examples

```
create_request("/api/tags")  
create_request("/api/chat")  
create_request("/api/embeddings")
```

delete	<i>Delete a model</i>
--------	-----------------------

Description

Delete a model from your local machine that you downloaded using the pull() function. To see which models are available, use the list_models() function.

Usage

```
delete(model, endpoint = "/api/delete")
```

Arguments

model	A character string of the model name such as "llama3".
endpoint	The endpoint to delete the model. Default is "/api/delete".

Value

A htrr2 response object.

Examples

```
## Not run:  
delete("llama3")  
  
## End(Not run)
```

embeddings	<i>Get vector embedding for a prompt</i>
------------	--

Description

Get vector embedding for a prompt

Usage

```
embeddings(model, prompt, normalize = TRUE, endpoint = "/api/embeddings")
```

Arguments

model	A character string of the model name such as "llama3".
prompt	A character string of the prompt that you want to get the vector embedding for.
normalize	Normalize the vector to length 1. Default is TRUE.
endpoint	The endpoint to get the vector embedding. Default is "/api/embeddings".

Value

A numeric vector of the embedding.

Examples

```
embeddings("nomic-embed-text:latest", "The quick brown fox jumps over the lazy dog.")
```

generate	<i>Generate a completion.</i>
----------	-------------------------------

Description

Generate a response for a given prompt with a provided model.

Usage

```
generate(  
  model,  
  prompt,  
  stream = FALSE,  
  output = c("resp", "jsonlist", "raw", "df"),  
  endpoint = "/api/generate"  
)
```

Arguments

model	A character string of the model name such as "llama3".
prompt	A character string of the prompt like "The sky is..."
stream	Enable response streaming. Default is FALSE.
output	A character vector of the output format. Default is "resp". Options are "resp", "jsonlist", "raw", "df".
endpoint	The endpoint to generate the completion. Default is "/api/generate".

Value

A response in the format specified in the output parameter.

Examples

```
generate("llama3", "The sky is...", stream = FALSE, output = "df")  
generate("llama3", "The sky is...", stream = TRUE, output = "df")  
generate("llama3", "The sky is...", stream = FALSE, output = "jsonlist")
```

list_models	<i>Get available local models</i>
-------------	-----------------------------------

Description

Get available local models

Usage

```
list_models(  
  output = c("df", "resp", "jsonlist", "raw"),  
  endpoint = "/api/tags"  
)
```

Arguments

output The output format. Default is "df". Other options are "resp", "jsonlist", "raw".
endpoint The endpoint to get the models. Default is "/api/tags".

Value

A httr2 response object, json list, raw or data frame. Default is "df".

Examples

```
list_models() # returns dataframe/tibble by default  
list_models("df")  
list_models("resp")  
list_models("jsonlist")  
list_models("raw")
```

package_config	<i>Package configuration</i>
----------------	------------------------------

Description

Package configuration

Usage

```
package_config
```

Format

An object of class `list` of length 3.

pull	<i>Pull/download a model</i>
------	------------------------------

Description

See <https://ollama.com/library> for a list of available models. Use the `list_models()` function to get the list of models already downloaded/installed on your machine.

Usage

```
pull(model, stream = TRUE, endpoint = "/api/pull")
```

Arguments

model	A character string of the model name such as "llama3".
stream	Enable response streaming. Default is TRUE.
endpoint	The endpoint to pull the model. Default is "/api/pull".

Value

A htr2 response object.

Examples

```
pull("llama3")
```

resp_process	<i>Process htr2 response object.</i>
--------------	--------------------------------------

Description

Process htr2 response object.

Usage

```
resp_process(resp, output = c("df", "jsonlist", "raw", "resp"))
```

Arguments

resp	A htr2 response object.
output	The output format. Default is "df". Other options are "jsonlist", "raw", "resp" (htr2 response object).

Value

A data frame, json list, raw or httr2 response object.

Examples

```
resp <- list_models("resp")
resp_process(resp, "df") # parse response to dataframe/tibble
resp_process(resp, "jsonlist") # parse response to list
resp_process(resp, "raw") # parse response to raw string
resp_process(resp, "resp") # return input response object
```

test_connection	<i>Test connection to Ollama server</i>
-----------------	---

Description

Test connection to Ollama server

Usage

```
test_connection(url = "http://localhost:11434")
```

Arguments

url The URL of the Ollama server. Default is http://localhost:11434

Value

A httr2 response object.

Examples

```
test_connection()
test_connection("http://localhost:11434")
test_connection("http://127.0.0.1:11434")
```

Index

* datasets

package_config, 6

chat, 2

create_request, 3

delete, 3

embeddings, 4

generate, 5

list_models, 6

package_config, 6

pull, 7

resp_process, 7

test_connection, 8