Package 'LLMR'

January 20, 2025

Title Interface for Large Language Model APIs in R

Version 0.1.2

Description

A unified interface to interact with various Large Language Model (LLM) APIs such as 'OpenAI' (see <https://platform.openai.com/docs/quickstart> for details), 'Anthropic' (see <https://docs.anthropic.com/en/api/getting-started> for details), 'Groq' (see <https://console.groq.com/docs/api-reference> for details), 'Together AI' (see <https://docs.together.ai/docs/quickstart> for details), 'DeepSeek' (see <https://api-docs.deepseek.com> for details), and 'Voyage AI' (see <https://docs.voyageai.com/docs/introduction> for details). Allows users to configure API parameters, send messages, and retrieve responses seamlessly within R.

License MIT + file LICENSE

Encoding UTF-8

Imports httr2, purrr, rlang

Suggests testthat (>= 3.0.0), roxygen2 (>= 7.1.2), httptest2

RoxygenNote 7.2.3

Config/testthat/edition 3

NeedsCompilation no

Author Ali Sanaei [aut, cre]

Maintainer Ali Sanaei <sanaei@uchicago.edu>

Repository CRAN

Date/Publication 2025-01-17 09:50:02 UTC

Contents

call_llm 2
llm_config 3
parse_embeddings 4

Index **6**

call_llm *Call LLM API*

Description

Sends a message to the specified LLM API and retrieves the response.

Usage

```
call_llm(config, messages, verbose = FALSE, json = FALSE)
```

Arguments

config	An ‘llm_config‘ object created by ‘llm_config()‘.
messages	A list of message objects (or a character vector for embeddings) to send to the API.
verbose	Logical. If ‘TRUE‘, prints the full API response.
json	Logical. If ‘TRUE‘, returns the raw JSON response as an attribute.

Value

The generated text response or embedding results with additional attributes.

Examples

```
## Not run:
  # OpenAI Embedding Example (overwriting api_url):
  openai_embed_config <- llm_config(
    provider = "openai",
    model = "text-embedding-3-small",
    api_key = Sys.getenv("OPENAI_KEY"),
    temperature = 0.3,
    api_url = "https://api.openai.com/v1/embeddings"
  )

  text_input <- c("Political science is a useful subject",
                  "We love sociology",
                  "German elections are different",
                  "A student was always curious.")

  embed_response <- call_llm(openai_embed_config, text_input)

  # Voyage AI Example:
  voyage_config <- llm_config(
    provider = "voyage",
    model = "voyage-large-2",
    api_key = Sys.getenv("VOYAGE_API_KEY")
  )
```

```
  embedding_response <- call_llm(voyage_config, text_input)
  embeddings <- parse_embeddings(embedding_response)
  embeddings |> cor() |> print()

## End(Not run)
```

`llm_config` *Create LLM Configuration*

Description

Creates a configuration object for interacting with a specified LLM API provider.

Usage

```
llm_config(provider, model, api_key, ...)
```

Arguments

`provider`	A string specifying the API provider. Supported providers include: "openai" for OpenAI, "anthropic" for Anthropic, "groq" for Groq, "together" for Together AI, "deepseek" for DeepSeek, "voyage" for Voyage AI.
`model`	The model name to use. This depends on the provider.
`api_key`	Your API key for the provider.
`...`	Additional model-specific parameters (e.g., ‘temperature‘, ‘max_tokens‘, etc.).

Value

An object of class ‘llm_config‘ containing API and model parameters.

Examples

```
## Not run:
  # OpenAI Example (chat)
  openai_config <- llm_config(
    provider = "openai",
    model = "gpt-4o-mini",
    api_key = Sys.getenv("OPENAI_KEY"),
    temperature = 0.7,
    max_tokens = 500
  )

  # OpenAI Embedding Example (overwriting api_url):
  openai_embed_config <- llm_config(
    provider = "openai",
    model = "text-embedding-3-small",
    api_key = Sys.getenv("OPENAI_KEY"),
```

```
  temperature = 0.3,
  api_url = "https://api.openai.com/v1/embeddings"
)

text_input <- c("Political science is a useful subject",
                "We love sociology",
                "German elections are different",
                "A student was always curious.")

embed_response <- call_llm(openai_embed_config, text_input)
# parse_embeddings() can then be used to convert the embedding results.

# Voyage AI Example:
voyage_config <- llm_config(
  provider = "voyage",
  model = "voyage-large-2",
  api_key = Sys.getenv("VOYAGE_API_KEY")
)

embedding_response <- call_llm(voyage_config, text_input)
embeddings <- parse_embeddings(embedding_response)
# Additional processing:
embeddings |> cor() |> print()

## End(Not run)
```

`parse_embeddings` *Parse Embedding Response into a Numeric Matrix*

Description

Converts the embedding response data to a numeric matrix.

Usage

```
parse_embeddings(embedding_response)
```

Arguments

`embedding_response`	The response returned from an embedding API call.

Value

A numeric matrix of embeddings with column names as sequence numbers.

Examples

```
## Not run:
  text_input <- c("Political science is a useful subject",
                  "We love sociology",
                  "German elections are different",
                  "A student was always curious.")

  # Configure the embedding API provider (example with Voyage API)
  voyage_config <- llm_config(
    provider = "voyage",
    model = "voyage-large-2",
    api_key = Sys.getenv("VOYAGE_API_KEY")
  )

  embedding_response <- call_llm(voyage_config, text_input)
  embeddings <- parse_embeddings(embedding_response)
  # Additional processing:
  embeddings |> cor() |> print()

## End(Not run)
```

Index

call_llm, 2

llm_config, 3

parse_embeddings, 4