

# Package: mall (via r-universe)

October 25, 2024

**Title** Run Multiple Large Language Model Predictions Against a Table, or Vectors

**Version** 0.1.0

**Description** Run multiple 'Large Language Model' predictions against a table. The predictions run row-wise over a specified column. It works using a one-shot prompt, along with the current row's content. The prompt that is used will depend of the type of analysis needed.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Imports** cli, dplyr, fs, glue, jsonlite, ollamar, rlang

**Suggests** dbplyr, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**URL** <https://mlverse.github.io/mall/>

**Depends** R (>= 2.10)

**LazyData** true

**NeedsCompilation** no

**Author** Edgar Ruiz [aut, cre], Posit Software, PBC [cph, fnd]

**Maintainer** Edgar Ruiz <edgar@posit.co>

**Repository** CRAN

**Date/Publication** 2024-10-24 14:30:02 UTC

## Contents

llm_classify . . . . .	2
llm_custom . . . . .	3
llm_extract . . . . .	4
llm_sentiment . . . . .	6
llm_summarize . . . . .	8

llm_translate . . . . .	9
llm_use . . . . .	10
llm_verify . . . . .	11
reviews . . . . .	13

<b>Index</b>	<b>14</b>
--------------	-----------

---

llm_classify	<i>Categorize data as one of options given</i>
--------------	--

---

### Description

Use a Large Language Model (LLM) to classify the provided text as one of the options provided via the labels argument.

### Usage

```
llm_classify(
  .data,
  col,
  labels,
  pred_name = ".classify",
  additional_prompt = ""
)
```

```
llm_vec_classify(x, labels, additional_prompt = "", preview = FALSE)
```

### Arguments

.data	A data.frame or tbl object that contains the text to be analyzed
col	The name of the field to analyze, supports tidy-eval
labels	A character vector with at least 2 labels to classify the text as
pred_name	A character vector with the name of the new column where the prediction will be placed
additional_prompt	Inserts this text into the prompt sent to the LLM
x	A vector that contains the text to be analyzed
preview	It returns the R call that would have been used to run the prediction. It only returns the first record in x. Defaults to FALSE Applies to vector function only.

### Value

llm\_classify returns a data.frame or tbl object. llm\_vec\_classify returns a vector that is the same length as x.

**Examples**

```
library(mall)

data("reviews")

llm_use("ollama", "llama3.2", seed = 100, .silent = TRUE)

llm_classify(reviews, review, c("appliance", "computer"))

# Use 'pred_name' to customize the new column's name
llm_classify(
  reviews,
  review,
  c("appliance", "computer"),
  pred_name = "prod_type"
)

# Pass custom values for each classification
llm_classify(reviews, review, c("appliance" ~ 1, "computer" ~ 2))

# For character vectors, instead of a data frame, use this function
llm_vec_classify(
  c("this is important!", "just whenever"),
  c("urgent", "not urgent")
)

# To preview the first call that will be made to the downstream R function
llm_vec_classify(
  c("this is important!", "just whenever"),
  c("urgent", "not urgent"),
  preview = TRUE
)
```

---

llm\_custom

*Send a custom prompt to the LLM*

---

**Description**

Use a Large Language Model (LLM) to process the provided text using the instructions from prompt

**Usage**

```
llm_custom(.data, col, prompt = "", pred_name = ".pred", valid_resps = "")

llm_vec_custom(x, prompt = "", valid_resps = NULL)
```

**Arguments**

<code>.data</code>	A <code>data.frame</code> or <code>tbl</code> object that contains the text to be analyzed
<code>col</code>	The name of the field to analyze, supports <code>tidy-eval</code>
<code>prompt</code>	The prompt to append to each record sent to the LLM
<code>pred_name</code>	A character vector with the name of the new column where the prediction will be placed
<code>valid_resps</code>	If the response from the LLM is not open, but deterministic, provide the options in a vector. This function will set to <code>NA</code> any response not in the options
<code>x</code>	A vector that contains the text to be analyzed

**Value**

`llm_custom` returns a `data.frame` or `tbl` object. `llm_vec_custom` returns a vector that is the same length as `x`.

**Examples**

```
library(mall)

data("reviews")

llm_use("ollama", "llama3.2", seed = 100, .silent = TRUE)

my_prompt <- paste(
  "Answer a question.",
  "Return only the answer, no explanation",
  "Acceptable answers are 'yes', 'no'",
  "Answer this about the following text, is this a happy customer?:"
)

reviews |>
  llm_custom(review, my_prompt)
```

---

 llm\_extract

*Extract entities from text*


---

**Description**

Use a Large Language Model (LLM) to extract specific entity, or entities, from the provided text

**Usage**

```
llm_extract(
  .data,
  col,
  labels,
  expand_cols = FALSE,
  additional_prompt = "",
  pred_name = ".extract"
)

llm_vec_extract(x, labels = c(), additional_prompt = "", preview = FALSE)
```

**Arguments**

<code>.data</code>	A <code>data.frame</code> or <code>tbl</code> object that contains the text to be analyzed
<code>col</code>	The name of the field to analyze, supports <code>tidy-eval</code>
<code>labels</code>	A vector with the entities to extract from the text
<code>expand_cols</code>	If multiple <code>labels</code> are passed, this is a flag that tells the function to create a new column per item in <code>labels</code> . If <code>labels</code> is a named vector, this function will use those names as the new column names, if not, the function will use a sanitized version of the content as the name.
<code>additional_prompt</code>	Inserts this text into the prompt sent to the LLM
<code>pred_name</code>	A character vector with the name of the new column where the prediction will be placed
<code>x</code>	A vector that contains the text to be analyzed
<code>preview</code>	It returns the R call that would have been used to run the prediction. It only returns the first record in <code>x</code> . Defaults to <code>FALSE</code> Applies to vector function only.

**Value**

`llm_extract` returns a `data.frame` or `tbl` object. `llm_vec_extract` returns a vector that is the same length as `x`.

**Examples**

```
library(mall)

data("reviews")

llm_use("ollama", "llama3.2", seed = 100, .silent = TRUE)

# Use 'labels' to let the function know what to extract
llm_extract(reviews, review, labels = "product")

# Use 'pred_name' to customize the new column's name
llm_extract(reviews, review, "product", pred_name = "prod")
```

```
# Pass a vector to request multiple things, the results will be pipe delimited
# in a single column
llm_extract(reviews, review, c("product", "feelings"))

# To get multiple columns, use 'expand_cols'
llm_extract(reviews, review, c("product", "feelings"), expand_cols = TRUE)

# Pass a named vector to set the resulting column names
llm_extract(
  .data = reviews,
  col = review,
  labels = c(prod = "product", feels = "feelings"),
  expand_cols = TRUE
)

# For character vectors, instead of a data frame, use this function
llm_vec_extract("bob smith, 123 3rd street", c("name", "address"))

# To preview the first call that will be made to the downstream R function
llm_vec_extract(
  "bob smith, 123 3rd street",
  c("name", "address"),
  preview = TRUE
)
```

---

llm\_sentiment

*Sentiment analysis*

---

## Description

Use a Large Language Model (LLM) to perform sentiment analysis from the provided text

## Usage

```
llm_sentiment(
  .data,
  col,
  options = c("positive", "negative", "neutral"),
  pred_name = ".sentiment",
  additional_prompt = ""
)

llm_vec_sentiment(
  x,
  options = c("positive", "negative", "neutral"),
  additional_prompt = "",
  preview = FALSE
)
```

**Arguments**

<code>.data</code>	A <code>data.frame</code> or <code>tbl</code> object that contains the text to be analyzed
<code>col</code>	The name of the field to analyze, supports <code>tidy-eval</code>
<code>options</code>	A vector with the options that the LLM should use to assign a sentiment to the text. Defaults to: <code>'positive'</code> , <code>'negative'</code> , <code>'neutral'</code>
<code>pred_name</code>	A character vector with the name of the new column where the prediction will be placed
<code>additional_prompt</code>	Inserts this text into the prompt sent to the LLM
<code>x</code>	A vector that contains the text to be analyzed
<code>preview</code>	It returns the R call that would have been used to run the prediction. It only returns the first record in <code>x</code> . Defaults to <code>FALSE</code> Applies to vector function only.

**Value**

`llm_sentiment` returns a `data.frame` or `tbl` object. `llm_vec_sentiment` returns a vector that is the same length as `x`.

**Examples**

```
library(mall)

data("reviews")

llm_use("ollama", "llama3.2", seed = 100, .silent = TRUE)

llm_sentiment(reviews, review)

# Use 'pred_name' to customize the new column's name
llm_sentiment(reviews, review, pred_name = "review_sentiment")

# Pass custom sentiment options
llm_sentiment(reviews, review, c("positive", "negative"))

# Specify values to return per sentiment
llm_sentiment(reviews, review, c("positive" ~ 1, "negative" ~ 0))

# For character vectors, instead of a data frame, use this function
llm_vec_sentiment(c("I am happy", "I am sad"))

# To preview the first call that will be made to the downstream R function
llm_vec_sentiment(c("I am happy", "I am sad"), preview = TRUE)
```

---

llm_summarize	<i>Summarize text</i>
---------------	-----------------------

---

### Description

Use a Large Language Model (LLM) to summarize text

### Usage

```
llm_summarize(
  .data,
  col,
  max_words = 10,
  pred_name = ".summary",
  additional_prompt = ""
)

llm_vec_summarize(x, max_words = 10, additional_prompt = "", preview = FALSE)
```

### Arguments

.data	A data.frame or tbl object that contains the text to be analyzed
col	The name of the field to analyze, supports tidy-eval
max_words	The maximum number of words that the LLM should use in the summary. Defaults to 10.
pred_name	A character vector with the name of the new column where the prediction will be placed
additional_prompt	Inserts this text into the prompt sent to the LLM
x	A vector that contains the text to be analyzed
preview	It returns the R call that would have been used to run the prediction. It only returns the first record in x. Defaults to FALSE Applies to vector function only.

### Value

llm\_summarize returns a data.frame or tbl object. llm\_vec\_summarize returns a vector that is the same length as x.

### Examples

```
library(mall)

data("reviews")

llm_use("ollama", "llama3.2", seed = 100, .silent = TRUE)
```



```

# Use max_words to set the maximum number of words to use for the summary
llm_summarize(reviews, review, max_words = 5)

# Use 'pred_name' to customize the new column's name
llm_summarize(reviews, review, 5, pred_name = "review_summary")

# For character vectors, instead of a data frame, use this function
llm_vec_summarize(
  "This has been the best TV I've ever used. Great screen, and sound.",
  max_words = 5
)

# To preview the first call that will be made to the downstream R function
llm_vec_summarize(
  "This has been the best TV I've ever used. Great screen, and sound.",
  max_words = 5,
  preview = TRUE
)

```

---

llm_translate	<i>Translates text to a specific language</i>
---------------	---

---

## Description

Use a Large Language Model (LLM) to translate a text to a specific language

## Usage

```

llm_translate(
  .data,
  col,
  language,
  pred_name = ".translation",
  additional_prompt = ""
)

llm_vec_translate(x, language, additional_prompt = "", preview = FALSE)

```

## Arguments

.data	A data.frame or tbl object that contains the text to be analyzed
col	The name of the field to analyze, supports tidy-eval
language	Target language to translate the text to
pred_name	A character vector with the name of the new column where the prediction will be placed
additional_prompt	Inserts this text into the prompt sent to the LLM

x	A vector that contains the text to be analyzed
preview	It returns the R call that would have been used to run the prediction. It only returns the first record in x. Defaults to FALSE Applies to vector function only.

**Value**

llm\_translate returns a data.frame or tbl object. llm\_vec\_translate returns a vector that is the same length as x.

**Examples**

```
library(mall)

data("reviews")

llm_use("ollama", "llama3.2", seed = 100, .silent = TRUE)

# Pass the desired language to translate to
llm_translate(reviews, review, "spanish")
```

---

llm_use	<i>Specify the model to use</i>
---------	---------------------------------

---

**Description**

Allows us to specify the back-end provider, model to use during the current R session

**Usage**

```
llm_use(
  backend = NULL,
  model = NULL,
  ...,
  .silent = FALSE,
  .cache = NULL,
  .force = FALSE
)
```

**Arguments**

backend	The name of an supported back-end provider. Currently only 'ollama' is supported.
model	The name of model supported by the back-end provider
...	Additional arguments that this function will pass down to the integrating function. In the case of Ollama, it will pass those arguments to ollama:::chat().
.silent	Avoids console output

<code>.cache</code>	The path to save model results, so they can be re-used if the same operation is ran again. To turn off, set this argument to an empty character: <code>""</code> . It defaults to a temp folder. If this argument is left NULL when calling this function, no changes to the path will be made.
<code>.force</code>	Flag that tell the function to reset all of the settings in the R session

**Value**

A `mall_session` object

**Examples**

```
library(mall)

llm_use("ollama", "llama3.2")

# Additional arguments will be passed 'as-is' to the
# downstream R function in this example, to ollama::chat()
llm_use("ollama", "llama3.2", seed = 100, temperature = 0.1)

# During the R session, you can change any argument
# individually and it will retain all of previous
# arguments used
llm_use(temperature = 0.3)

# Use .cache to modify the target folder for caching
llm_use(.cache = "_my_cache")

# Leave .cache empty to turn off this functionality
llm_use(.cache = "")

# Use .silent to avoid the print out
llm_use(.silent = TRUE)
```

---

`llm_verify`*Verify if a statement about the text is true or not*

---

**Description**

Use a Large Language Model (LLM) to see if something is true or not based the provided text

**Usage**

```
llm_verify(
  .data,
  col,
  what,
  yes_no = factor(c(1, 0)),
```

```

  pred_name = ".verify",
  additional_prompt = ""
)

llm_vec_verify(
  x,
  what,
  yes_no = factor(c(1, 0)),
  additional_prompt = "",
  preview = FALSE
)

```

### Arguments

<code>.data</code>	A <code>data.frame</code> or <code>tbl</code> object that contains the text to be analyzed
<code>col</code>	The name of the field to analyze, supports <code>tidy-eval</code>
<code>what</code>	The statement or question that needs to be verified against the provided text
<code>yes_no</code>	A size 2 vector that specifies the expected output. It is positional. The first item is expected to be value to return if the statement about the provided text is true, and the second if it is not. Defaults to: <code>factor(c(1, 0))</code>
<code>pred_name</code>	A character vector with the name of the new column where the prediction will be placed
<code>additional_prompt</code>	Inserts this text into the prompt sent to the LLM
<code>x</code>	A vector that contains the text to be analyzed
<code>preview</code>	It returns the R call that would have been used to run the prediction. It only returns the first record in <code>x</code> . Defaults to <code>FALSE</code> Applies to vector function only.

### Value

`llm_verify` returns a `data.frame` or `tbl` object. `llm_vec_verify` returns a vector that is the same length as `x`.

### Examples

```

library(mall)

data("reviews")

llm_use("ollama", "llama3.2", seed = 100, .silent = TRUE)

# By default it will return 1 for 'true', and 0 for 'false',
# the new column will be a factor type
llm_verify(reviews, review, "is the customer happy")

# The yes_no argument can be modified to return a different response
# than 1 or 0. First position will be 'true' and second, 'false'
llm_verify(reviews, review, "is the customer happy", c("y", "n"))

```

```
# Number can also be used, this would be in the case that you wish to match
# the output values of existing predictions
llm_verify(reviews, review, "is the customer happy", c(2, 1))
```

---

reviews

*Mini reviews data set*

---

### **Description**

Mini reviews data set

### **Usage**

```
reviews
```

### **Format**

A data frame that contains 3 records. The records are of fictitious product reviews.

### **Examples**

```
library(mall)
data(reviews)
reviews
```

# Index

## \* datasets

reviews, [13](#)

llm\_classify, [2](#)

llm\_custom, [3](#)

llm\_extract, [4](#)

llm\_sentiment, [6](#)

llm\_summarize, [8](#)

llm\_translate, [9](#)

llm\_use, [10](#)

llm\_vec\_classify (llm\_classify), [2](#)

llm\_vec\_custom (llm\_custom), [3](#)

llm\_vec\_extract (llm\_extract), [4](#)

llm\_vec\_sentiment (llm\_sentiment), [6](#)

llm\_vec\_summarize (llm\_summarize), [8](#)

llm\_vec\_translate (llm\_translate), [9](#)

llm\_vec\_verify (llm\_verify), [11](#)

llm\_verify, [11](#)

reviews, [13](#)