Package 'baserater'

November 11, 2025

Title Base-Rate Item Evaluation and Typicality Scoring Using Large Language Models

Version 0.1.1

Description Download typicality rating datasets, generate new stereotype-based typicality ratings using large language models via the Inference Providers API (<https://huggingface.co/docs/inference-providers>), and evaluate them against human-annotated validation data. Also includes functions to extract stereotype strength and base-rate items from typicality matrices. For more details see Beucler et al. (2025) <doi:10.31234/osf.io/eqrfu_v1>.

License MIT + file LICENSE

Depends R (>= 4.1.0)

Encoding UTF-8

RoxygenNote 7.3.3

Imports cli, dplyr, glue, readr, tibble, tidyr, httr2

Suggests devtools, knitr, rmarkdown, tidyverse

VignetteBuilder knitr

URL https://jeremie-beucler.github.io/baserater/

NeedsCompilation no

Author Jeremie Beucler [aut, cre]

Maintainer Jeremie Beucler <jeremie.beucler@gmail.com>

Repository CRAN

Date/Publication 2025-11-11 21:50:02 UTC

Contents

`download_data` *Load base-rate database, model typicality matrices, or human validation ratings*

Description

This function gives access to key datasets included in the `baserater` package.

Usage

```
download_data(
  which = c("database", "validation_ratings", "typicality_matrix_gpt4",
    "typicality_matrix_llama3.3", "material"),
  dest = NULL
)
```

Arguments

`which`	One of "database", "validation_ratings", "typicality_matrix_gpt4", "typicality_matrix_llama3.3", or "material".
`dest`	Optional path to copy the file to (returns the data either way).

Details

- The `"database"` object includes all base-rate items along with stereotype strength estimates from 'GPT-4' and 'LLaMA 3.3'.
- The `"validation_ratings"` object contains average typicality judgments from 50 human participants on 100 group–adjective pairs, as well as ratings from 'GPT-4' and 'LLaMA 3.3'.
- The `"typicality_matrix_gpt4"` and `"typicality_matrix_llama3.3"` objects are raw typicality matrices generated by each model.
- The `"material"` object contains the lists of individual groups and adjectives used to build the base-rate database.

Value

A tibble with the requested data.

Examples

```
database <- download_data("database")

ratings <- download_data("validation_ratings")

gpt4_matrix <- download_data("typicality_matrix_gpt4")

llama3_matrix <- download_data("typicality_matrix_llama3.3")

material <- download_data("material")
```

`evaluate_external_ratings` *Evaluate how new typicality ratings predict human ratings and compares performance to LLM baselines*

Description

This function compares external typicality ratings (e.g., generated by a new LLM) against the validation dataset included in 'baserater'. The validation set contains average typicality ratings collected from 50 Prolific participants on a subset of 100 group–adjective pairs, as described in the accompanying paper.

The input ratings are merged with this reference set, and then:

1. Computes a correlation (`cor.test`) between the external ratings and the human average;
2. Compares it to one or more built-in model baselines (default: 'GPT-4' and 'LLaMA 3.3');
3. Prints a clear summary of all correlation coefficients and flags whether the external model outperforms each baseline;
4. Returns a tidy result invisibly.

Usage

```
evaluate_external_ratings(
  df,
  method = "pearson",
  baselines = c("mean_gpt4_rating", "mean_llama3_rating"),
  verbose = TRUE
)
```

Arguments

`df`	A data frame with columns `adjective`, `group`, and `rating`. Must contain typicality scores for all 100 validation items used in the original study.
`method`	The correlation method to use in `stats::cor.test()`. Must be one of: `"pearson"` (default), `"spearman"`, or `"kendall"`.
`baselines`	Character vector of column names in the validation set to compare against (default: `c("mean_gpt4_rating", "mean_llama3_rating")`).
`verbose`	Logical. If `TRUE` (default), prints a summary of the correlations and baseline comparisons. Set to `FALSE` to suppress console output.

Value

A tibble (invisibly) with one row per model (`external` and each baseline), and columns `model`, `r`, and `p` for the correlation coefficient and p-value.

Examples

```
## Not run:
new_scores <- tibble::tibble(
  group = ratings$group,
  adjective = ratings$adjective,
  rating = runif(100)  # Replace with model predictions
)
evaluate_external_ratings(new_scores)

## End(Not run)
```

`extract_base_rate_items` *Create base-rate items from groups x descriptions typicality matrix*

Description

This function processes a typicality matrix to identify base-rate items by comparing typicality scores of descriptions between all unique pairs of groups.

Usage

```
extract_base_rate_items(typicality_matrix)
```

Arguments

`typicality_matrix`
: A numeric matrix or data frame where rows are groups and columns are descriptions. If a data frame, the first column is assumed to contain the group names.

Details

For each pair of groups and each description (e.g., adjective), it identifies which group received the higher typicality score. The output includes the names of both groups, their scores, and the log-ratio between the higher and lower score.

It can be quite slow for large matrices, as the number of items becomes very large.

By construction, the returned `Group1` always has a higher or equal typicality score than `Group2` for a given description. This ensures that the resulting `StereotypeStrength` (defined as `log(Score1 / Score2)`) is always **positive or zero**, and represents the strength of the stereotypical association in favor of `Group1`.

Value

A data frame with the following columns:

Group1 The group with the higher typicality score for the description.

Group2 The group with the lower typicality score.

Description The description (e.g., adjective) being compared.

Score1 The typicality score for Group1.

Score2 The typicality score for Group2.

StereotypeStrength The log-ratio: `log(Score1 / Score2)`. Always >= 0.

Examples

```
mat <- matrix(runif(9, 1, 100), nrow = 3,
              dimnames = list(c("GroupA", "GroupB", "GroupC"),
                              c("smart", "brave", "greedy")))
extract_base_rate_items(mat)
```

`generate_typicality` *Generate typicality ratings via an 'Inference Provider' (experimental)*

Description

This function uses a compatible 'Inference Provider' API (e.g., 'Together AI' or 'Fireworks') to generate typicality ratings by querying a large language model (LLM). It generates one or multiple ratings for each group-description pair and returns the mean score. It can be quite slow to run depending on the API.

Important: Before running this function, please ensure that:

- You have a valid API token from your inference provider (via `api_token` or an environment variable);
- You have provided the correct and complete URL for the provider's chat completions endpoint;
- The specified model is available and accessible via the endpoint;
- The model supports the standard `messages` array format (with system/user roles) and generates numeric outputs in response to the prompts.

Calls to the API are rate-limited, may incur usage costs, and require an internet connection. This feature is **experimental** and is not guaranteed to work with all models or providers.

Usage

```
generate_typicality(
  groups,
  descriptions,
  api_url,
  api_token,
  model = "meta-llama/Llama-3.3-70B-Instruct-Turbo",
  n = 25,
  min_valid = ceiling(0.8 * n),
  temperature = 1,
  top_p = 1,
```

```
  max_tokens = 3,
  retries = 4,
  matrix = TRUE,
  return_raw_scores = TRUE,
  return_full_responses = FALSE,
  verbose = interactive(),
  system_prompt = default_system_prompt(),
  user_prompt_template = default_user_prompt_template()
)
```

Arguments

groups, descriptions	Character vectors. *When* matrix = FALSE they **must** be the same length.
api_url	Fully-qualified HTTPS URL for the provider's chat completions endpoint (e.g., "https://api.together.xyz/v1/chat/completions").
api_token	API token for the inference provider.
model	Model identifier string to be passed in the API request body. Check your provider's documentation for the available models and correct names.
n	Samples requested per retry block (>= 1).
min_valid	Minimum numeric scores required per pair (>= 1).
temperature, top_p, max_tokens	Generation controls.
retries	Maximum number of *additional* retry blocks.
matrix	TRUE = cross-product, FALSE = paired.
return_raw_scores	If TRUE, also returns the vector(s) of raw valid numeric scores.
return_full_responses	If TRUE, also returns all raw text model outputs (or error strings from failed attempts) for each query.
verbose	If TRUE, prints progress: pair labels, retry counts, running tallies, and raw model responses/errors as they occur.
system_prompt	Prompt string for the system message. See the 'Prompting Details' section and function signature for default content and customization.
user_prompt_template	Prompt template for the user message, with {group} and {description} placeholders. No additional formatting is added by the function. See the 'Prompting Details' section and function signature for default content and customization.

Value

If a pair cannot reach min_valid, its mean is NA; raw invalid strings remain available when return_full_responses = TRUE. Cross-product mode (matrix = TRUE) -> a list containing:

- scores: A matrix of mean typicality scores.

- raw (if return_raw_scores = TRUE): A matrix of lists, where each list contains the raw numeric scores for that pair.
- full_responses (if return_full_responses = TRUE): A matrix of lists, where each list contains all raw text model outputs (or error strings) for that pair.

Paired mode (matrix = FALSE) -> a tibble with columns for group, description, mean_score, and additionally:

- raw (if return_raw_scores = TRUE): A list-column where each element is a vector of raw numeric scores.
- full_responses (if return_full_responses = TRUE): A list-column where each element is a character vector of all raw text model outputs (or error strings).

Get Typicality Ratings from Large Language Models

generate_typicality() sends structured prompts to any text-generation model served via an compatible API endpoint and collects *numeric* ratings (0-100) of how well a *description* (e.g., an adjective) fits a *group* (e.g., an occupation). Responses that cannot be parsed into numbers are discarded.

Modes:

- **Cross-product** (matrix = TRUE, *default*) Rate every combination of the *unique* groups and descriptions. Returns a list containing matrices.
- **Paired** (matrix = FALSE) Rate the pairs row-by-row (length(groups) == length(descriptions)). Returns a tibble.

Each pair is queried repeatedly until at least min_valid clean scores are obtained or the retry budget is exhausted. One *retry block* consists of n new samples; invalid or out-of-range answers are silently dropped.

Prompting Details

The function constructs a messages array for the API request. The system_prompt becomes the content of the system role message, and the rendered user_prompt_template (where {group} and {description} are substituted with the actual values) becomes the content of the user role message.

The default system_prompt is:

```
You are expert at accurately reproducing the stereotypical associations
humans make, in order to annotate data for experiments.
Your focus is to capture common societal perceptions and stereotypes,
rather than factual attributes of the groups,
even when they are negative or unfounded.
```

The default user_prompt_template is:

```
Rate how well the description "{description}" reflects the prototypical
member of the group "{group}" on a scale from 0 ("Not at all") to 100
("Extremely").
```

```
To clarify, consider the following examples:
1. "Rate how well the description "FUNNY" reflects the prototypical member
   of the group "CLOWN" on a scale from 0 (Not at all) to 100 (Extremely)."
   A high rating is expected because "FUNNY" closely aligns with typical
   characteristics of a "CLOWN".
2. "Rate how well the description "FEARFUL" reflects the prototypical member
   of the group "FIREFIGHTER" on a scale from 0 (Not at all) to 100
   (Extremely)." A low rating is expected because "FEARFUL" diverges from
   typical characteristics of a "FIREFIGHTER".
3. "Rate how well the description "PATIENT" reflects the prototypical member
   of the group "ENGINEER" on a scale from 0 (Not at all) to 100
   (Extremely)." A mid-scale rating is expected because "PATIENT" neither
   strongly aligns with nor diverges from typical characteristics of an
   "ENGINEER".

Your response should be a single score between 0 and 100, with no additional
text, letters, or symbols.
```

Rate-limit friendliness: transient HTTP 429/5xx errors are retried (exponential back-off).

Examples

```
## Not run:

Sys.setenv(PROVIDER_API_URL = "https://api.together.xyz/v1/chat/completions")
Sys.setenv(PROVIDER_API_TOKEN = "your_secret_token_here")

toy_groups <- c("engineer", "clown", "firefighter") # Minimal example
toy_descriptions <- c("patient", "funny", "fearful")

toy_result <- generate_typicality(
  groups = toy_groups,
  descriptions = toy_descriptions,
  api_url = Sys.getenv("PROVIDER_API_URL"),
  api_token = Sys.getenv("PROVIDER_API_TOKEN"),
  model = "meta-llama/Llama-3.3-70B-Instruct-Turbo",
  n = 10,
  min_valid = 8,
  matrix = FALSE,
  return_raw_scores = TRUE,
  return_full_responses = FALSE,
  verbose = TRUE
)

print(toy_result)

## End(Not run)

## Not run:

ratings <- download_data("validation_ratings") # Full-scale example
```

```
new_scores <- generate_typicality(
  groups                = ratings$group,
  descriptions          = ratings$adjective,
  api_url               = Sys.getenv("PROVIDER_API_URL"),
  api_token             = Sys.getenv("PROVIDER_API_TOKEN"),
  model                 = "meta-llama/Llama-3.3-70B-Instruct-Turbo",
  n                     = 25,
  min_valid             = 20,
  max_tokens            = 5,
  retries               = 1,
  matrix                = FALSE,
  return_raw_scores     = TRUE,
  return_full_responses = TRUE,
  verbose               = TRUE
)

head(new_scores)

## End(Not run)
```

Index