

Package ‘chatRater’

May 14, 2026

Type Package

Title A Tool for Rating Text/Image/Audio Stimuli via 'LLMs'

Version 1.3.0

Date 2026-05-13

Maintainer Shiyang Zheng <Shiyang.Zheng@nottingham.ac.uk>

Description Evaluates stimuli using Large Language Models.

Supports multiple LLM providers: 'OpenAI', 'Anthropic', 'Ollama', 'LM Studio', 'DeepSeek', 'Groq', 'Mistral', and 'OpenAI-compatible' endpoints.

Stimuli: plain text, local image/audio files, or image URLs.

Audio is transcribed via 'OpenAI Whisper' before rating.

Supports numeric, text, and raw return types.

License MIT + file LICENSE

Depends R (>= 4.1.0)

Encoding UTF-8

Imports base64enc, tools, httr2, curl, jsonlite

Suggests llmcode (>= 1.2.0), testthat (>= 3.0.0)

NeedsCompilation no

RoxygenNote 7.3.2

Config/testthat/edition 3

URL <https://github.com/ShiyangZheng/chatRater>

BugReports <https://github.com/ShiyangZheng/chatRater/issues>

Author Shiyang Zheng [aut, cre]

Repository CRAN

Date/Publication 2026-05-14 13:50:13 UTC

Contents

generate_ratings	2
generate_ratings_for_all	5

Index	8
--------------	----------

 generate_ratings

A Tool for Rating Text/Image/Audio Stimuli via 'LLMs'

Description

Evaluates stimuli using Large Language Models through the 'lmdecoder' package. Supports multiple LLM providers: 'OpenAI', 'Anthropic', 'Ollama', 'LM Studio', 'DeepSeek', 'Groq', 'Mistral', and 'OpenAI-compatible' endpoints. Designed for research rating tasks. Stimuli can be plain text, local image/audio files, or image URLs.

Usage

```
generate_ratings(
  model = NULL,
  stim,
  prompt = "You are an expert rater. Limit your answer to numbers only.",
  question = "Please rate this:",
  scale = "1-7",
  temp = 0,
  n_iterations = 1,
  provider = c("openai", "anthropic", "ollama", "lmstudio", "deepseek", "groq",
    "mistral", "openrouter", "openai_compatible"),
  api_key = NULL,
  base_url = NULL,
  debug = FALSE,
  return_type = c("numeric", "text", "raw"),
  columns = NULL
)
```

Arguments

model	LLM model name (default varies by provider)
stim	Input stimulus: a plain text string, a local file path (image or audio), or an image URL starting with http(s)://. Supported local files: images (png, jpg, gif, webp, bmp), audio (mp3, wav, ogg, flac, m4a, aac).
prompt	System instruction for the LLM
question	Specific rating question for the LLM
scale	Rating scale range (default: '1-7'). Ignored when return_type = 'text' or 'raw'.
temp	Temperature parameter (default: 0)
n_iterations	Number of rating iterations (default: 1)
provider	LLM provider: 'openai', 'anthropic', 'ollama', 'lmstudio', 'deepseek', 'groq', 'mistral', 'openrouter', 'openai_compatible'
api_key	API key (not needed for local providers like 'ollama')
base_url	Custom base URL for OpenAI-compatible APIs

debug	Debug mode flag (default: FALSE)
return_type	Output type: 'numeric' (default, extract numbers only), 'text' (return full text response), 'raw' (return unprocessed response)
columns	Character vector of column names to include in the returned data frame. Available columns: 'stim', 'rating', 'iteration', 'scale', 'type', 'provider', 'model'. Default is NULL (all columns returned).

Value

A data frame containing ratings and metadata. Columns depend on the columns argument. The 'rating' column type depends on return_type.

Examples

```
## Not run:
# -----
# 1. TEXT STIMULUS – numeric rating (local LLM, no API key needed)
# -----
# Rate the idiomaticity of an English expression on a 1-7 scale
result <- generate_ratings(
  stim      = "kick the bucket",
  prompt    = "You are a native English speaker.",
  question  = "Rate how idiomatic this expression is (1 = not at all, 7 = very idiomatic):",
  scale     = "1-7",
  provider  = "ollama"
)
result
# stim          rating iteration scale type provider  model
# kick the bucket 7         1         1-7  text ollama   llama3.2

# -----
# 2. TEXT STIMULUS – full text description (OpenAI)
# -----
# Ask the model to explain an expression instead of just rating it
result <- generate_ratings(
  stim      = "spill the beans",
  prompt    = "You are an expert linguist.",
  question  = "Explain the meaning of this expression and describe its usage:",
  provider  = "openai",
  api_key   = Sys.getenv("OPENAI_API_KEY"),
  return_type = "text"
)
cat(result$rating)
# "Spill the beans" means to reveal a secret or disclose information
# that was supposed to remain hidden...

# -----
# 3. IMAGE STIMULUS – local file, numeric rating
# -----
# Rate the visual complexity of a local image on a 1-5 scale
result <- generate_ratings(
  stim      = "/path/to/stimulus_image.png",
```

```

prompt = "You are an expert in visual perception research.",
question = "Rate the visual complexity of this image (1 = very simple, 5 = very complex):",
scale = "1-5",
provider = "openai",
api_key = Sys.getenv("OPENAI_API_KEY")
)
result$rating # e.g. "3"

# -----
# 4. IMAGE STIMULUS – local file, full description
# -----
# Ask the model to describe what is in an image
result <- generate_ratings(
  stim = "/path/to/scene.jpg",
  prompt = "You are a helpful assistant.",
  question = "Describe in detail what you see in this image:",
  provider = "openai",
  api_key = Sys.getenv("OPENAI_API_KEY"),
  return_type = "text"
)
cat(result$rating)
# "The image shows a busy market scene with several vendors..."

# -----
# 5. IMAGE STIMULUS – URL, numeric rating
# -----
# Rate an image hosted online (e.g. on OSF or a public server)
result <- generate_ratings(
  stim = "https://osf.io/download/example_stimulus.png",
  prompt = "You are an expert image rater.",
  question = "Rate the emotional valence of this image (1 = very negative, 7 = very positive):",
  scale = "1-7",
  provider = "openai",
  api_key = Sys.getenv("OPENAI_API_KEY")
)

# -----
# 6. AUDIO STIMULUS – local file, full transcription + description
# -----
# Whisper transcribes the audio first, then GPT-4o rates/describes it
result <- generate_ratings(
  stim = "/path/to/speech_sample.wav",
  prompt = "You are an expert in spoken language assessment.",
  question = "Describe the content and speaking style of this audio clip:",
  provider = "openai",
  api_key = Sys.getenv("OPENAI_API_KEY"),
  return_type = "text"
)
cat(result$rating)
# "The speaker describes a childhood memory in a calm, reflective tone..."

# -----
# 7. AUDIO STIMULUS – local file, numeric rating

```

```

# -----
# Rate the fluency of a speech recording
result <- generate_ratings(
  stim      = "/path/to/learner_speech.mp3",
  prompt    = "You are an expert language teacher assessing spoken fluency.",
  question  = "Rate the overall fluency of the speaker (1 = very disfluent, 7 = very fluent):",
  scale     = "1-7",
  provider  = "openai",
  api_key   = Sys.getenv("OPENAI_API_KEY")
)
result$rating # e.g. "5"

# -----
# 8. CUSTOM COLUMNS – keep only what you need
# -----
# Return only the stimulus and its rating (drop metadata columns)
result <- generate_ratings(
  stim      = "break a leg",
  prompt    = "You are a native English speaker.",
  question  = "Rate familiarity (1-7):",
  provider  = "ollama",
  columns   = c("stim", "rating")
)
result
# stim      rating
# break a leg 6

# -----
# 9. MULTIPLE ITERATIONS – reliability check
# -----
result <- generate_ratings(
  stim      = "once in a blue moon",
  prompt    = "You are a native English speaker.",
  question  = "Rate how familiar this expression is (1-7):",
  provider  = "openai",
  api_key   = Sys.getenv("OPENAI_API_KEY"),
  n_iterations = 3,
  columns   = c("stim", "rating", "iteration")
)
result
# stim      rating iteration
# once in a blue moon 7      1
# once in a blue moon 7      2
# once in a blue moon 6      3

## End(Not run)

```

generate_ratings_for_all

Batch Rating Generator

Description

Process multiple stimuli in sequence using 'LLMs'. Supports text, image, and audio stimuli (see [generate_ratings()]).

Usage

```
generate_ratings_for_all(model = NULL, stim_list, ...)
```

Arguments

model	LLM model name (default varies by provider)
stim_list	A character vector of stimuli to process. Each element can be a text string, a local file path (image/audio), or an image URL.
...	Additional arguments passed to [generate_ratings()]

Examples

```
## Not run:
# -----
# 1. BATCH TEXT – rate multiple idioms for familiarity
# -----
idioms <- c("kick the bucket", "spill the beans", "break a leg",
           "hit the nail on the head", "once in a blue moon")
results <- generate_ratings_for_all(
  stim_list = idioms,
  prompt    = "You are a native English speaker.",
  question  = "Rate how familiar this expression is (1 = unfamiliar, 7 = very familiar):",
  scale     = "1-7",
  provider  = "ollama",
  columns   = c("stim", "rating")
)
results

# -----
# 2. BATCH TEXT – get full descriptions
# -----
results <- generate_ratings_for_all(
  stim_list = c("kick the bucket", "spill the beans"),
  prompt    = "You are an expert linguist.",
  question  = "Explain the meaning of this idiom in one sentence:",
  provider  = "openai",
  api_key   = Sys.getenv("OPENAI_API_KEY"),
  return_type = "text",
  columns   = c("stim", "rating")
)

# -----
# 3. BATCH IMAGE – mix of local files and URLs
# -----
stimuli <- c(
  "/path/to/image1.png",          # local file
```

```
    "/path/to/image2.jpg",          # local file
    "https://osf.io/download/example_img.png" # URL
  )
results <- generate_ratings_for_all(
  stim_list = stimuli,
  prompt    = "You are an expert in visual perception.",
  question  = "Rate the visual complexity (1 = simple, 5 = complex):",
  scale     = "1-5",
  provider  = "openai",
  api_key   = Sys.getenv("OPENAI_API_KEY"),
  columns   = c("stim", "rating", "type")
)

# -----
# 4. BATCH AUDIO – rate fluency of multiple speech recordings
# -----
audio_files <- c(
  "/path/to/speaker1.wav",
  "/path/to/speaker2.mp3",
  "/path/to/speaker3.m4a"
)
results <- generate_ratings_for_all(
  stim_list = audio_files,
  prompt    = "You are an expert language teacher.",
  question  = "Rate the overall fluency of the speaker (1 = very disfluent, 7 = very fluent):",
  scale     = "1-7",
  provider  = "openai",
  api_key   = Sys.getenv("OPENAI_API_KEY"),
  columns   = c("stim", "rating")
)

## End(Not run)
```

Index

`generate_ratings`, [2](#)
`generate_ratings_for_all`, [5](#)