Package: cat.stack (via r-universe)

June 4, 2026

Title General-Purpose LLM Text Classification Engine

Version 0.2.0

Description R interface to the Python cat-stack package.
General-purpose text, image, and PDF classification using LLMs
with no domain assumptions. The base engine for the CatLLM
ecosystem.

License GPL (>= 3)

URL https://christophersoria.com/cat-llm/cat.stack/,
https://github.com/chrissoria/cat-llm

BugReports https://github.com/chrissoria/cat-llm/issues

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

SystemRequirements Python (>= 3.9), pip, cat-stack (>= 1.6.0)

Imports reticulate (>= 1.28)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/pak/sysreqs libpng-dev python3

Repository https://chrissoria.r-universe.dev

Date/Publication 2026-06-04 16:16:50 UTC

RemoteUrl https://github.com/chrissoria/cat-llm

RemoteRef main

RemoteSha f2d83209be8d621fceb422d434fb5b3b98fe301b

RemoteSubdir r-package/cat.stack

Contents

check_ollama_model 2
classify 3
ensure_ollama_running 8
explore 10
extract 12
install_cat_stack 14
list_ollama_models 15
prompt_tune 15
pull_ollama_model 18
summarize 19

Index **22**

`check_ollama_model` *Check whether a specific Ollama model is installed locally*

Description

Returns `TRUE` if the named model is available in your local Ollama installation, `FALSE` otherwise. Partial name matching is supported (e.g. `"llama3.2"` matches `"llama3.2:latest"`).

Usage

```
check_ollama_model(model, host = "localhost", port = 11434L)
```

Arguments

`model`	Character. Model name to look for (e.g. `"qwen2.5:7b"`).
`host`	Character. Hostname Ollama is reachable on. Default `"localhost"`.
`port`	Integer. Port Ollama is reachable on. Default `11434L`.

Value

Logical scalar.

Examples

```
## Not run:
check_ollama_model("qwen2.5:7b")

## End(Not run)
```

`classify` *Classify text, images, or PDFs using LLMs*

Description

Wraps the Python `cat_stack.classify()` function. Supports both single-model and multi-model (ensemble) classification.

Usage

```
classify(
  input_data,
  categories,
  api_key = NULL,
  description = "",
  user_model = "gpt-4o",
  mode = "image",
  creativity = NULL,
  safety = FALSE,
  chain_of_verification = FALSE,
  chain_of_thought = FALSE,
  step_back_prompt = FALSE,
  context_prompt = FALSE,
  thinking_budget = 0L,
  example1 = NULL,
  example2 = NULL,
  example3 = NULL,
  example4 = NULL,
  example5 = NULL,
  example6 = NULL,
  filename = NULL,
  save_directory = NULL,
  model_source = "auto",
  max_categories = 12L,
  categories_per_chunk = 10L,
  divisions = 10L,
  research_question = NULL,
  models = NULL,
  consensus_threshold = "unanimous",
  survey_question = "",
  use_json_schema = TRUE,
  max_workers = NULL,
  fail_strategy = "partial",
  max_retries = 5L,
  batch_retries = 1L,
  json_retries = 2L,
  retry_delay = 1,
```

```
  row_delay = 0,
  pdf_dpi = 150L,
  auto_download = FALSE,
  add_other = "prompt",
  check_verbosity = TRUE,
  batch_mode = FALSE,
  batch_poll_interval = 30,
  batch_timeout = 86400,
  json_formatter = NULL,
  two_step_classify = NULL,
  embedding_tiebreaker = FALSE,
  min_centroid_size = 3L,
  auto_start_ollama = TRUE,
  system_prompt = "",
  prompt_tune = NULL,
  tune_iterations = 1L,
  tune_ui = "browser",
  tune_optimize = "balanced"
)
```

Arguments

input_data	A character vector, list of text strings, or data.frame column containing the items to classify. For image or PDF classification, a directory path or character vector of file paths.
categories	A character vector of category names, or "auto" to infer categories from the data (requires survey_question).
api_key	API key for the model provider (single-model mode). Not required when models is supplied.
description	Character. Context description for the classification task (e.g., the survey question or image subject). Default "".
user_model	Character. Model name to use in single-model mode. Default "gpt-4o".
mode	Character. PDF processing mode: "image" (default), "text", or "both".
creativity	Numeric or NULL. Temperature setting (0-2). NULL uses the provider default. Default NULL.
safety	Logical. If TRUE, saves progress after each item. Default FALSE.
chain_of_verification	Logical. Enable Chain of Verification. Empirically degrades accuracy – provided for research only. Default FALSE.
chain_of_thought	Logical. Enable chain-of-thought reasoning. Default FALSE.
step_back_prompt	Logical. Enable step-back prompting. Default FALSE.
context_prompt	Logical. Add expert context to prompts. Default FALSE.
thinking_budget	Integer. Extended thinking token budget (0 = off). Default 0L.

example1, example2, example3, example4, example5, example6
: Optional few-shot example strings. Empirically degrades accuracy – provided for research only.

filename
: Character or NULL. Output CSV filename. Default NULL.

save_directory
: Character or NULL. Directory to save results. Default NULL.

model_source
: Character. Provider hint for single-model mode: "auto", "openai", "anthropic", "google", "mistral", "perplexity", "huggingface", "xai", "ollama", or "claude-code". Default "auto" (detects from model name; falls back to "huggingface" for Qwen/Llama/DeepSeek-style names — use "ollama" explicitly to route those to a local Ollama server).

max_categories
: Integer. Maximum number of categories when categories = "auto". Default 12L.

categories_per_chunk
: Integer. Categories extracted per chunk when categories = "auto". Default 10L.

divisions
: Integer. Number of data chunks when categories = "auto". Default 10L.

research_question
: Character or NULL. Optional research context. Default NULL.

models
: A list of model specifications for multi-model ensemble mode. Each element is either a 3-element character vector c("model", "provider", "api_key") or a 4-element list list("model", "provider", "api_key", list(creativity = 0.5)). When models is supplied, api_key and user_model are ignored.

consensus_threshold
: Character or numeric. Agreement threshold for ensemble mode. Options:

  - "unanimous" (default, 100% — empirically the most accurate)
  - "majority" — STRICT majority. More than half of the models must vote positive. Ties (50/50 splits on even-model ensembles like 2-2 of 4) resolve to "0". This matches sklearn's VotingClassifier default and standard ensemble literature. For 2-model ensembles, "majority" effectively requires both models to agree on positive (there's no "more than half" of 2 without being all); use 3+ models for a non-degenerate majority vote, or pass 0.5 numerically to keep the old "tie favors positive" semantics.
  - "two-thirds" — ~67% agreement, >= semantics.
  - numeric between 0 and 1 — evaluated with >= semantics (the user picked a number; they get the literal interpretation).

  The output data.frame for multi-model runs includes category_N_agreement columns (fraction of models that match the consensus, 0.0-1.0). For even-model ensembles with "majority", pair with embedding_tiebreaker = TRUE to resolve true 50/50 ties via embedding-centroid similarity instead of the default "tie → 0"; that adds a category_N_resolved_by audit column (values: "vote" or "centroid").

survey_question
: Character. The survey question text (used when categories = "auto"). Default "".

use_json_schema
: Logical. Use JSON schema for structured output. Default TRUE.

`max_workers` Integer or `NULL`. Max parallel workers. `NULL` = auto. Default `NULL`.

`fail_strategy` Character. How to handle failures: `"partial"` (default) or `"strict"`.

`max_retries` Integer. Max retries per API call. Default `5L`.

`batch_retries` Integer. Max retries for batch-level failures. Default `1L`. Note: composes multiplicatively with `json_retries` — a row can hit the LLM up to `(1 + json_retries) * (1 + batch_retries)` times.

`json_retries` Integer. Per-row retries when the LLM returns JSON that fails schema validation. On each retry the prompt appends "Respond with ONLY valid JSON". On the final attempt the formatter fallback (if enabled via `json_formatter`) fires before the row is marked failed. Default `2L`.

`retry_delay` Numeric. Seconds between retries. Default `1.0`.

`row_delay` Numeric. Seconds between processing each row (useful for rate limiting). Default `0.0`.

`pdf_dpi` Integer. DPI for PDF page rendering. Default `150L`.

`auto_download` Logical. Auto-download Ollama models. Default `FALSE`.

`add_other` Logical or `"prompt"`. Controls auto-addition of an "Other" catch-all category. `"prompt"` (default) asks interactively – in non-interactive sessions this silently defaults to "no". `TRUE` silently adds "Other". `FALSE` never adds it.

`check_verbosity`
Logical. Check whether each category has a description and examples (1 API call). Default `TRUE`.

`batch_mode` Logical. If `TRUE`, use async batch APIs for ~50% cost savings and higher rate limits. Supported providers: OpenAI, Anthropic, Google, Mistral, xAI. HuggingFace / Perplexity / Ollama fall back to synchronous calls. Incompatible with PDF / image input and with `embedding_tiebreaker`. Default `FALSE`.

`batch_poll_interval`
Numeric. Seconds between batch-job status polls when `batch_mode = TRUE`. Default `30.0`.

`batch_timeout` Numeric. Maximum seconds to wait for a batch job to complete. Default `86400.0` (24 hours).

`json_formatter` `TRUE`, `FALSE`, or `NULL`. Three-state control for the local JSON-repair fallback model that fixes malformed LLM output before marking rows as failed. Runs only when `extract_json()` produces invalid output. The model (~1 GB) is downloaded from HuggingFace Hub on first use; requires `cat-stack[formatter]`.

- `TRUE` — eagerly load and use the formatter (implicit consent for the ~1.5 GB dependency install if needed).
- `FALSE` — disabled; malformed rows stay as failures.
- `NULL` (default) — auto-prompt on the first malformed row. If dependencies are installed, asks "Use the formatter for this run? (Y/n)"; if not, asks "Download deps (~1.5 GB) and use the formatter? (Y/n)". Non-TTY contexts (CI, batch scripts) decline silently and print a one-time suggestion.

Auto-enabled when `two_step_classify = TRUE` or any model uses the Ollama provider.

two_step_classify
: TRUE, FALSE, or NULL. Split classification into two LLM calls — (1) natural-language reasoning, (2) JSON formatting. More reliable for weaker models (local Ollama, lower-tier API models like gpt-4o-mini, claude-haiku-4-5, gemini-2.5-flash) that struggle to produce strict per-category JSON in a single shot. Default NULL (auto-enables for Ollama models, disabled otherwise). When enabled, json_formatter is auto-enabled too.

embedding_tiebreaker
: Logical. Resolve true ensemble ties (50/50 splits at the threshold) using embedding centroids built from unanimously-agreed rows; the closer centroid wins. Companion for consensus_threshold = "majority" on even-model ensembles — replaces the default "tie → 0" with an evidence-based decision. Adds a category_N_resolved_by audit column to the output (values: "vote" or "centroid"). Multi-model ensemble + text input only; not supported in batch_mode. Requires cat-stack[embeddings]. Default FALSE.

min_centroid_size
: Integer. Minimum number of unanimously-agreed rows needed to build a centroid for a category when embedding_tiebreaker = TRUE. Categories with fewer confident rows fall back to vote-based consensus. Default 3L.

auto_start_ollama
: Logical. If TRUE (default), automatically call ensure_ollama_running() when model_source = "ollama" or any ensemble entry uses the "ollama" provider. Set FALSE to skip the check (e.g. on CI runners where you don't want to launch Ollama).

system_prompt
: Character. Custom system-level instruction prepended to every classification call. Use this to apply a prompt returned by prompt_tune(): system_prompt = result$system_prompt. Default "".

prompt_tune
: Integer or NULL. If set, enables Automatic Prompt Optimization (APO). The value is the number of rows sampled per correction round. A browser window opens so you can correct misclassifications; the meta-LLM then rewrites the system prompt and re-classifies until accuracy converges or tune_iterations is reached. Categories are never modified — only the system prompt changes. Default NULL (disabled).

tune_iterations
: Integer. Number of APO optimization passes. Default 1L.

tune_ui
: Character. Correction UI: "browser" (default) opens an interactive browser window; "terminal" uses the console.

tune_optimize
: Character. Metric to optimize: "balanced" (default, maximizes average of accuracy, sensitivity, and precision), "sensitivity" (minimize false negatives), or "precision" (minimize false positives).

Value

A data.frame with one row per input item and classification columns. In single-model mode the columns are the category names. In ensemble mode additional consensus_* and agreement_* columns are included.

Examples

```
## Not run:
# Single-model classification
results <- classify(
  input_data  = c("I love this!", "Terrible service.", "It was okay."),
  categories  = c("Positive", "Negative", "Neutral"),
  description = "Customer feedback",
  api_key     = Sys.getenv("OPENAI_API_KEY")
)

# Multi-model ensemble
results <- classify(
  input_data  = df$responses,
  categories  = c("Positive", "Negative", "Neutral"),
  models      = list(
    c("gpt-4o",              "openai",    Sys.getenv("OPENAI_API_KEY")),
    c("claude-sonnet-4-5-20250929", "anthropic", Sys.getenv("ANTHROPIC_API_KEY"))
  ),
  consensus_threshold = "unanimous"
)

# Even-model ensemble with strict-majority + embedding tiebreaker
# (resolves true 50/50 ties via centroid similarity instead of
# the default "tie -> 0"; requires cat-stack[embeddings])
results <- classify(
  input_data           = df$responses,
  categories           = c("Positive", "Negative", "Neutral"),
  models               = list(
    c("gpt-4o-mini",      "openai",    Sys.getenv("OPENAI_API_KEY")),
    c("claude-haiku-4-5", "anthropic", Sys.getenv("ANTHROPIC_API_KEY"))
  ),
  consensus_threshold  = "majority",
  embedding_tiebreaker = TRUE
)

# Async batch mode (50% cheaper, slower) — OpenAI / Anthropic /
# Google / Mistral / xAI only; not yet supported with PDFs/images
# or embedding_tiebreaker.
results <- classify(
  input_data = df$responses,
  categories = c("Positive", "Negative", "Neutral"),
  api_key    = Sys.getenv("OPENAI_API_KEY"),
  batch_mode = TRUE
)

## End(Not run)
```

`ensure_ollama_running` *Ensure a local Ollama server is running*

Description

Checks whether an Ollama server is reachable at `host:port`. If not, attempts to start it using the platform-appropriate command and polls until the server responds (or `timeout` is reached). Call this once at the top of an R session before classifying with `model_source = "ollama"`.

Usage

```
ensure_ollama_running(
  auto_start = TRUE,
  timeout = 30,
  host = "localhost",
  port = 11434L,
  verbose = TRUE
)
```

Arguments

`auto_start`	Logical. If `TRUE` (default), attempt to launch Ollama when not running. If `FALSE`, just check and error if not running.
`timeout`	Numeric. Seconds to wait for Ollama to become ready after `auto_start`. Default `30`.
`host`	Character. Hostname Ollama is reachable on. Default `"localhost"`.
`port`	Integer. Port Ollama is reachable on. Default `11434L`.
`verbose`	Logical. Print status messages. Default `TRUE`.

Details

Platform start commands:

- **macOS** — `open -a Ollama` (launches the Ollama.app daemon). Falls back to `ollama serve` if the app is not installed.
- **Linux** — `ollama serve` (run in a detached process).
- **Windows** — `ollama serve`.

If Ollama is not installed, the function returns a clear error message linking to https://ollama.com.

Value

Invisibly returns `TRUE` when Ollama is running.

Examples

```
## Not run:
# Ensure Ollama is up before classifying with a local model
ensure_ollama_running()

results <- classify(
```

```
  input_data   = c("text 1", "text 2"),
  categories   = c("Positive", "Negative", "Neutral"),
  user_model   = "qwen2.5:7b",
  model_source = "ollama"
)

# Just check without auto-starting
ensure_ollama_running(auto_start = FALSE)

## End(Not run)
```

`explore` *Explore raw categories in text data*

Description

Wraps the Python `cat_stack.explore()` function. Returns every category string extracted from every chunk across every iteration – with duplicates intact. Useful for analysing category stability and saturation across repeated extraction runs.

Usage

```
explore(
  input_data,
  api_key,
  description = "",
  max_categories = 12L,
  categories_per_chunk = 10L,
  divisions = 12L,
  user_model = "gpt-4o",
  creativity = NULL,
  specificity = "broad",
  research_question = NULL,
  filename = NULL,
  model_source = "auto",
  iterations = 8L,
  random_state = NULL,
  focus = NULL,
  chunk_delay = 0,
  auto_start_ollama = TRUE
)
```

Arguments

`input_data`	A character vector, list, or `data.frame` column of text responses.
`api_key`	Character. API key for the model provider.
`description`	Character. The survey question or data description. Default `""`.

`max_categories` Integer. Maximum categories per chunk. Default `12L`.

`categories_per_chunk`
Integer. Categories to extract per chunk. Default `10L`.

`divisions` Integer. Number of data chunks. Default `12L`.

`user_model` Character. Model name. Default `"gpt-4o"`.

`creativity` Numeric or `NULL`. Temperature setting. `NULL` uses the provider default. Default `NULL`.

`specificity` Character. `"broad"` (default) or `"specific"`.

`research_question`
Character or `NULL`. Optional research context.

`filename` Character or `NULL`. Optional CSV filename to save the raw category list.

`model_source` Character. Provider hint. Default `"auto"`.

`iterations` Integer. Number of passes over the data. Default `8L`.

`random_state` Integer or `NULL`. Random seed for reproducibility.

`focus` Character or `NULL`. Optional focus instruction.

`chunk_delay` Numeric. Seconds between API calls. Default `0.0`.

`auto_start_ollama`
Logical. If `TRUE` (default), automatically call `ensure_ollama_running()` when `model_source = "ollama"`. Set `FALSE` to skip the check (e.g. on CI).

Details

Unlike `extract()`, which normalises and deduplicates categories, `explore()` returns the raw unprocessed output suitable for frequency and saturation analysis.

Value

A character vector of every category string extracted across all chunks and iterations. Length is approximately `iterations * divisions * categories_per_chunk`.

Examples

```
## Not run:
raw_cats <- explore(
  input_data  = df$responses,
  description = "Why did you move?",
  api_key     = Sys.getenv("OPENAI_API_KEY"),
  iterations  = 3L,
  divisions   = 5L
)
length(raw_cats)   # ~150
head(raw_cats, 10)

## End(Not run)
```

extract *Extract categories from text, images, or PDFs using LLMs*

Description

Wraps the Python `cat_stack.extract()` function. Discovers and returns a normalised, deduplicated set of categories found in the input data.

Usage

```
extract(
  input_data,
  api_key,
  input_type = "text",
  description = "",
  max_categories = 12L,
  categories_per_chunk = 10L,
  divisions = 12L,
  user_model = "gpt-4o",
  creativity = NULL,
  specificity = "broad",
  research_question = NULL,
  mode = "text",
  filename = NULL,
  model_source = "auto",
  iterations = 8L,
  random_state = NULL,
  focus = NULL,
  chunk_delay = 0,
  auto_start_ollama = TRUE
)
```

Arguments

`input_data`	A character vector, list, or `data.frame` column. For images/PDFs, a directory path or character vector of file paths.
`api_key`	Character. API key for the model provider.
`input_type`	Character. Type of input: `"text"` (default), `"image"`, or `"pdf"`.
`description`	Character. The survey question or data description. Default `""`.
`max_categories`	Integer. Maximum number of final categories to return. Default `12L`.
`categories_per_chunk`	Integer. Categories to extract per data chunk. Default `10L`.
`divisions`	Integer. Number of chunks to divide the data into. Default `12L`.
`user_model`	Character. Model name. Default `"gpt-4o"`.

`creativity` Numeric or `NULL`. Temperature setting. `NULL` uses the provider default. Default `NULL`.

`specificity` Character. Category granularity: `"broad"` (default) or `"specific"`.

`research_question` Character or `NULL`. Optional research context.

`mode` Character. Processing mode. For PDFs: `"text"` (default), `"image"`, or `"both"`. For images: `"image"` (default) or `"both"`.

`filename` Character or `NULL`. Optional CSV filename to save results.

`model_source` Character. Provider hint: `"auto"`, `"openai"`, `"anthropic"`, `"google"`, etc. Default `"auto"`.

`iterations` Integer. Number of passes over the data. Default `8L`.

`random_state` Integer or `NULL`. Random seed for reproducibility.

`focus` Character or `NULL`. Optional focus for extraction (e.g., `"decisions to move"`).

`chunk_delay` Numeric. Seconds between API calls (rate limiting). Default `0.0`.

`auto_start_ollama` Logical. If `TRUE` (default), automatically call `ensure_ollama_running()` when `model_source = "ollama"`. Set `FALSE` to skip the check (e.g. on CI).

Value

A named list with:

`counts_df` A `data.frame` of discovered categories with counts.

`top_categories` A character vector of the top category names.

`raw_top_text` The raw model output from the final merge step.

Examples

```
## Not run:
result <- extract(
  input_data  = df$responses,
  description = "Why did you move to this city?",
  api_key     = Sys.getenv("OPENAI_API_KEY")
)
print(result$top_categories)
print(result$counts_df)

## End(Not run)
```

`install_cat_stack` *Install the cat-stack Python package*

Description

Installs the `cat-stack` Python package into the Python environment used by reticulate. Optionally installs PDF extras.

Usage

```
install_cat_stack(
  method = "auto",
  conda = "auto",
  pdf = FALSE,
  upgrade = FALSE,
  ...
)
```

Arguments

`method`	Installation method passed to `reticulate::py_install()`. Default `"auto"`.
`conda`	Conda environment name. Default `"auto"`.
`pdf`	Logical. If `TRUE`, installs `cat-stack[pdf]` with PDF extras. Default `FALSE`.
`upgrade`	Logical. If `TRUE`, upgrades an existing installation. Default `FALSE`.
`...`	Additional arguments passed to `reticulate::py_install()`.

Details

The version floor is pinned to `cat-stack >= 1.6.0` — that release adds strict-majority consensus, embedding tiebreaker, async batch mode, and the JSON-formatter auto-consent flow that the R wrappers now expose. Older Python installs work, but R users will hit "unexpected keyword argument" errors from `reticulate` when the new parameters get forwarded.

Value

Invisibly `NULL`.

Examples

```
## Not run:
# Standard install
install_cat_stack()

# With PDF support (installs cat-stack[pdf])
install_cat_stack(pdf = TRUE)

# Upgrade an existing install
```

```
install_cat_stack(upgrade = TRUE)

## End(Not run)
```

`list_ollama_models` *List locally installed Ollama models*

Description

Returns the names of all models already downloaded to your local Ollama installation. Requires Ollama to be running (call `ensure_ollama_running()` first, or start it manually with `ollama serve`).

Usage

```
list_ollama_models(host = "localhost", port = 11434L)
```

Arguments

`host`	Character. Hostname Ollama is reachable on. Default `"localhost"`.
`port`	Integer. Port Ollama is reachable on. Default `11434L`.

Value

A character vector of model names (e.g. `c("qwen2.5:7b", "mistral:7b")`), or an empty character vector if Ollama is not running.

Examples

```
## Not run:
ensure_ollama_running()
list_ollama_models()

## End(Not run)
```

`prompt_tune` *Optimize a classification prompt with human-in-the-loop feedback*

Description

Wraps the Python `catstack.prompt_tune()` function. Runs a coordinate-descent loop: classifies a small sample, asks you to correct the model's output, then has a meta-LLM rewrite the classification instructions for each category that had errors. Returns the best system prompt found plus per-iteration metrics.

Usage

```
prompt_tune(
  input_data,
  categories,
  api_key = NULL,
  user_model = "gpt-4o",
  model_source = "auto",
  models = NULL,
  description = "",
  survey_question = "",
  sample_size = 10L,
  max_iterations = 3L,
  multi_label = TRUE,
  creativity = NULL,
  use_json_schema = TRUE,
  consensus_threshold = "unanimous",
  max_retries = 5L,
  input_mode = NULL,
  ui = "terminal",
  optimize = "balanced",
  add_other = "prompt",
  thinking_budget = 0L,
  auto_start_ollama = TRUE
)
```

Arguments

input_data	A character vector, list, or data.frame column of items to classify during tuning.
categories	A character vector of category names. The labels themselves are never modified by tuning — only the classification instructions change.
api_key	Character or NULL. API key for the LLM provider.
user_model	Character. Model name. Default "gpt-4o".
model_source	Character. Provider hint. Default "auto".
models	List of model specs for ensemble mode (each c(model, provider, api_key)). Overrides user_model/api_key/ model_source if given. Default NULL.
description	Character. Context description. Default "".
survey_question	Character. Survey question for context. Default "".
sample_size	Integer. Items to test per iteration. Default 10L.
max_iterations	Integer. Max instruction attempts per category. Default 3L.
multi_label	Logical. Multi-label classification. Default TRUE.
creativity	Numeric or NULL. Temperature. Default NULL.
use_json_schema	Logical. Default TRUE.

`consensus_threshold`
: Character or numeric. For ensemble mode. Default `"unanimous"`.

`max_retries`
: Integer. Default `5L`.

`input_mode`
: Character or `NULL`. Input mode override.

`ui`
: Character. Review interface for corrections. `"terminal"` (default in R) reads from stdin. `"browser"` opens a local web page with checkboxes (may not auto-launch from R sessions).

`optimize`
: Character. Which metric to maximize. `"balanced"` (default), `"precision"`, or `"sensitivity"`.

`add_other`
: Logical or `"prompt"`. Controls auto-addition of an "Other" catch-all category. Default `"prompt"`.

`thinking_budget`
: Integer. Default `0L`.

`auto_start_ollama`
: Logical. If `TRUE` (default), automatically call `ensure_ollama_running()` when `model_source = "ollama"` or any ensemble entry uses the `"ollama"` provider. Set `FALSE` to skip the check.

Details

This function is **interactive** — you'll be asked to review and correct the model's labels at least once. From an R session, the default `ui = "terminal"` reads your corrections from stdin (works in R, Rscript, and most IDE consoles). `ui = "browser"` opens a local web page with checkboxes; depending on your R setup this may or may not auto-launch the browser, so terminal is the safer default for R users.

Use the returned `system_prompt` with `classify()` via the `system_prompt =` argument to apply the tuned instructions.

Value

A named list with components:

- `system_prompt` — the optimized system prompt (best found)
- `iterations` — list of per-iteration records (label, system_prompt, metrics, per_category, total_flips)
- `per_category_summary` — per-category metrics from the best-scoring iteration

Examples

```
## Not run:
result <- prompt_tune(
  input_data    = df$open_response,
  categories    = c("Positive", "Negative", "Neutral"),
  api_key       = Sys.getenv("OPENAI_API_KEY"),
  user_model    = "gpt-4o-mini",
  sample_size   = 10L,
  max_iterations = 3L,
  ui            = "terminal"
```

```
)

# Inspect the optimized prompt
cat(result$system_prompt)

# Use it in classify() via the system_prompt argument
results <- classify(
  input_data    = df$open_response,
  categories    = c("Positive", "Negative", "Neutral"),
  api_key       = Sys.getenv("OPENAI_API_KEY"),
  user_model    = "gpt-4o-mini",
  system_prompt = result$system_prompt
)

## End(Not run)
```

`pull_ollama_model` *Pull (download) an Ollama model*

Description

Downloads the named model into your local Ollama installation. Prints the estimated model size and a resource check before downloading. Set `auto_confirm = TRUE` to skip the interactive confirmation prompt — useful in scripts and RMarkdown documents.

Usage

```
pull_ollama_model(
  model,
  host = "localhost",
  port = 11434L,
  auto_confirm = FALSE
)
```

Arguments

`model`	Character. Model name to download (e.g. `"llama3.2"`, `"qwen2.5:7b"`).
`host`	Character. Hostname Ollama is reachable on. Default `"localhost"`.
`port`	Integer. Port Ollama is reachable on. Default `11434L`.
`auto_confirm`	Logical. Skip the confirmation prompt. Default `FALSE`.

Value

Invisibly returns `TRUE` on success, `FALSE` on failure.

Examples

```
## Not run:
pull_ollama_model("llama3.2", auto_confirm = TRUE)

## End(Not run)
```

`summarize` *Summarize text, images, or PDFs using LLMs*

Description

Wraps the Python `cat_stack.summarize()` function. Generates summaries of input data using one or more LLM models. Supports single-model and multi-model (ensemble) summarization.

Usage

```
summarize(
  input_data,
  api_key = NULL,
  description = "",
  instructions = "",
  format = "paragraph",
  max_length = NULL,
  focus = NULL,
  user_model = "gpt-4o",
  model_source = "auto",
  mode = "image",
  input_mode = NULL,
  input_type = "auto",
  pdf_dpi = 150L,
  creativity = NULL,
  thinking_budget = 0L,
  chain_of_thought = TRUE,
  context_prompt = FALSE,
  step_back_prompt = FALSE,
  filename = NULL,
  save_directory = NULL,
  models = NULL,
  max_workers = NULL,
  parallel = NULL,
  auto_download = FALSE,
  safety = FALSE,
  max_retries = 5L,
  batch_retries = 1L,
  retry_delay = 1,
  row_delay = 0,
  fail_strategy = "partial",
```

```
  batch_mode = FALSE,
  batch_poll_interval = 30,
  batch_timeout = 86400,
  auto_start_ollama = TRUE
)
```

Arguments

input_data	A character vector, list, or data.frame column. For images/PDFs, a directory path or character vector of file paths.
api_key	Character or NULL. API key for the model provider (single-model mode). Not required when models is supplied. Default NULL.
description	Character. Context description for the summarization task. Default "".
instructions	Character. Specific instructions for the summary. Default "".
format	Character. Output format: "paragraph" (default) or other supported formats.
max_length	Integer or NULL. Maximum length of the summary. NULL uses the model default. Default NULL.
focus	Character or NULL. Optional focus for the summary. Default NULL.
user_model	Character. Model name. Default "gpt-4o".
model_source	Character. Provider hint: "auto", "openai", "anthropic", "google", etc. Default "auto".
mode	Character. Processing mode for images/PDFs: "image" (default), "text", or "both".
input_mode	Character or NULL. Explicit input mode override. Default NULL.
input_type	Character. Type of input: "auto" (default), "text", "image", or "pdf".
pdf_dpi	Integer. DPI for PDF page rendering. Default 150L.
creativity	Numeric or NULL. Temperature setting. NULL uses the provider default. Default NULL.
thinking_budget	Integer. Extended thinking token budget (0 = off). Default 0L.
chain_of_thought	Logical. Enable chain-of-thought reasoning. Default TRUE.
context_prompt	Logical. Add expert context to prompts. Default FALSE.
step_back_prompt	Logical. Enable step-back prompting. Default FALSE.
filename	Character or NULL. Output filename. Default NULL.
save_directory	Character or NULL. Directory to save results. Default NULL.
models	A list of model specifications for multi-model ensemble mode. Each element is either a 3-element character vector c("model", "provider", "api_key") or a 4-element list list("model", "provider", "api_key", list(creativity = 0.5)). Default NULL.
max_workers	Integer or NULL. Max parallel workers. NULL = auto. Default NULL.

parallel	Logical or NULL. Enable parallel processing. Default NULL.
auto_download	Logical. Auto-download Ollama models. Default FALSE.
safety	Logical. If TRUE, saves progress after each item. Default FALSE.
max_retries	Integer. Max retries per API call. Default 5L.
batch_retries	Integer. Max retries for batch-level failures. Default 1L.
retry_delay	Numeric. Seconds between retries. Default 1.0.
row_delay	Numeric. Seconds between processing each row. Default 0.0.
fail_strategy	Character. How to handle failures: "partial" (default) or "strict".
batch_mode	Logical. Use batch processing mode. Default FALSE.
batch_poll_interval	Numeric. Seconds between batch status polls. Default 30.0.
batch_timeout	Numeric. Maximum seconds to wait for batch completion. Default 86400.0.
auto_start_ollama	Logical. If TRUE (default), automatically call ensure_ollama_running() when model_source = "ollama" or any ensemble entry uses the "ollama" provider. Set FALSE to skip the check.

Value

A data.frame with summarization results.

Examples

```
## Not run:
# Single-model summarization
results <- summarize(
  input_data   = c("A long article about climate change...",
                   "A detailed report on economic trends..."),
  description  = "News articles",
  instructions = "Provide a 2-sentence summary",
  api_key      = Sys.getenv("OPENAI_API_KEY")
)

# PDF summarization
results <- summarize(
  input_data = "path/to/documents/",
  input_type = "pdf",
  api_key    = Sys.getenv("OPENAI_API_KEY")
)

## End(Not run)
```

Index

check_ollama_model, 2
classify, 3
classify(), *17*

ensure_ollama_running, 8
ensure_ollama_running(), *7*, *11*, *13*, *17*, *21*
explore, 10
extract, 12
extract(), *11*

install_cat_stack, 14

list_ollama_models, 15

prompt_tune, 15
prompt_tune(), *7*
pull_ollama_model, 18

reticulate::py_install(), *14*

summarize, 19