Package: cat.web (via r-universe)

June 4, 2026

Title Web Content Classification with LLMs

Version 0.1.2

Description R interface to the Python catweb package. Classifies, extracts, explores, and summarizes web content (URLs or text) using LLMs. A thin domain wrapper around cat.stack that adds automatic URL fetching and web-context prompt injection (source domain, content type, metadata).

License GPL (>= 3)

URL https://christophersoria.com/cat-llm/cat.web/, 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

Imports reticulate (>= 1.28), cat.stack (>= 0.1.0)

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

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.web

Contents

`classify` *Classify web content using LLMs*

Description

Wraps the Python `catweb.classify()` function. Accepts URLs (auto-fetched to text) or raw text strings. Injects web context (source domain, content type, metadata) into the classification prompt.

Usage

```
classify(
  categories,
  input_data = NULL,
  api_key = NULL,
  source_domain = NULL,
  content_type = NULL,
  web_metadata = NULL,
  description = "",
  filename = NULL,
  save_directory = NULL,
  timeout = 30L,
  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,
  model_source = "auto",
  max_categories = 12L,
  categories_per_chunk = 10L,
```

```
  divisions = 10L,
  research_question = NULL,
  models = NULL,
  consensus_threshold = "unanimous",
  use_json_schema = TRUE,
  max_workers = NULL,
  fail_strategy = "partial",
  max_retries = 5L,
  batch_retries = 2L,
  retry_delay = 1,
  row_delay = 0,
  pdf_dpi = 150L,
  auto_download = FALSE,
  add_other = "prompt",
  check_verbosity = TRUE,
  prompt_tune = NULL,
  tune_iterations = 1L,
  tune_ui = "browser",
  tune_optimize = "balanced"
)
```

Arguments

categories	A character vector of category names.
input_data	A character vector / list / data.frame column of URLs or text strings. Default NULL.
api_key	Character or NULL. API key for the LLM provider.
source_domain	Character or NULL. Source domain injected into the prompt as context (e.g. "nytimes.com").
content_type	Character or NULL. Content type (e.g. "news article", "blog post").
web_metadata	Named list or NULL. Additional metadata injected into the prompt.
description	Character. Context description. Default "".
filename	Character or NULL. Output CSV filename.
save_directory	Character or NULL. Output directory.
timeout	Integer. URL fetch timeout (seconds). Default 30L.
user_model	Character. Model name. Default "gpt-4o".
mode	Character. Processing mode. Default "image".
creativity	Numeric or NULL. Temperature. Default NULL.
safety	Logical. Default FALSE.
chain_of_verification	Logical. Default FALSE.
chain_of_thought	Logical. Default FALSE.
step_back_prompt	Logical. Default FALSE.

`context_prompt` Logical. Default `FALSE`.

`thinking_budget`
Integer. Default `0L`.

`example1`, `example2`, `example3`, `example4`, `example5`, `example6`
Optional few-shot examples.

`model_source` Character. Default `"auto"`.

`max_categories` Integer. Default `12L`.

`categories_per_chunk`
Integer. Default `10L`.

`divisions` Integer. Default `10L`.

`research_question`
Character or `NULL`.

`models` List of model specs for ensemble mode. Default `NULL`.

`consensus_threshold`
Character or numeric. Default `"unanimous"`.

`use_json_schema`
Logical. Default `TRUE`.

`max_workers` Integer or `NULL`. Default `NULL`.

`fail_strategy` Character. Default `"partial"`.

`max_retries` Integer. Default `5L`.

`batch_retries` Integer. Default `2L`.

`retry_delay` Numeric. Default `1.0`.

`row_delay` Numeric. Default `0.0`.

`pdf_dpi` Integer. Default `150L`.

`auto_download` Logical. Default `FALSE`.

`add_other` Logical or `"prompt"`. Default `"prompt"`.

`check_verbosity`
Logical. Default `TRUE`.

`prompt_tune` Integer or `NULL`. Rows sampled per APO correction round. Default `NULL`.

`tune_iterations`
Integer. APO optimization passes. Default `1L`.

`tune_ui` Character. Correction UI: `"browser"` or `"terminal"`. Default `"browser"`.

`tune_optimize` Character. Metric to optimize: `"balanced"`, `"sensitivity"`, or `"precision"`. Default `"balanced"`.

Value

A `data.frame` with classification results.

Examples

```
## Not run:
# Classify a list of URLs (auto-fetched to text)
results <- classify(
  categories    = c("News", "Opinion", "Tutorial"),
  input_data    = c("https://example.com/article-1",
                    "https://example.com/article-2"),
  source_domain = "example.com",
  content_type  = "blog post",
  api_key       = Sys.getenv("OPENAI_API_KEY"),
  user_model    = "gpt-4o-mini"
)

# Or classify raw text (no fetching)
results <- classify(
  categories = c("News", "Opinion", "Tutorial"),
  input_data = df$article_text,
  api_key    = Sys.getenv("OPENAI_API_KEY")
)

## End(Not run)
```

---

explore *Explore raw categories in web content*

Description

Wraps the Python `catweb.explore()` function. Returns every category string extracted from every chunk across every iteration – with duplicates intact.

Usage

```
explore(
  input_data = NULL,
  api_key = NULL,
  source_domain = NULL,
  content_type = NULL,
  web_metadata = NULL,
  description = "",
  timeout = 30L,
  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
)
```

Arguments

input_data	A character vector / list of URLs or text. Default NULL.
api_key	Character or NULL. API key for the LLM provider.
source_domain	Character or NULL. Source domain context.
content_type	Character or NULL. Content type context.
web_metadata	Named list or NULL. Additional metadata.
description	Character. Default "".
timeout	Integer. URL fetch timeout (seconds). Default 30L.
max_categories	Integer. Default 12L.
categories_per_chunk	Integer. Default 10L.
divisions	Integer. Default 12L.
user_model	Character. Default "gpt-4o".
creativity	Numeric or NULL. Default NULL.
specificity	Character. Default "broad".
research_question	Character or NULL.
filename	Character or NULL.
model_source	Character. Default "auto".
iterations	Integer. Default 8L.
random_state	Integer or NULL.
focus	Character or NULL.
chunk_delay	Numeric. Default 0.0.

Value

A character vector of every category string extracted.

Examples

```
## Not run:
raw_cats <- explore(
  input_data    = urls,
  source_domain = "example.com",
  api_key       = Sys.getenv("OPENAI_API_KEY"),
  user_model    = "gpt-4o-mini",
```

```
  iterations    = 4L
)
table(raw_cats)

## End(Not run)
```

`extract` *Discover categories from web content using LLMs*

Description

Wraps the Python `catweb.extract()` function. Accepts URLs (auto-fetched) or raw text. Returns a normalised, deduplicated set of categories.

Usage

```
extract(
  input_data = NULL,
  api_key = NULL,
  source_domain = NULL,
  content_type = NULL,
  web_metadata = NULL,
  description = "",
  timeout = 30L,
  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
)
```

Arguments

`input_data`	A character vector / list of URLs or text. Default `NULL`.
`api_key`	Character or `NULL`. API key for the LLM provider.
`source_domain`	Character or `NULL`. Source domain context.
`content_type`	Character or `NULL`. Content type context.

web_metadata	Named list or NULL. Additional metadata.
description	Character. Default "".
timeout	Integer. URL fetch timeout (seconds). Default 30L.
max_categories	Integer. Default 12L.
categories_per_chunk	Integer. Default 10L.
divisions	Integer. Default 12L.
user_model	Character. Default "gpt-4o".
creativity	Numeric or NULL. Default NULL.
specificity	Character. Default "broad".
research_question	Character or NULL.
mode	Character. Default "text".
filename	Character or NULL.
model_source	Character. Default "auto".
iterations	Integer. Default 8L.
random_state	Integer or NULL.
focus	Character or NULL.
chunk_delay	Numeric. Default 0.0.

Value

A named list with counts_df, top_categories, and raw_top_text.

Examples

```
## Not run:
result <- extract(
  input_data    = c("https://example.com/page1",
                    "https://example.com/page2"),
  source_domain = "example.com",
  api_key       = Sys.getenv("OPENAI_API_KEY"),
  user_model    = "gpt-4o-mini"
)
print(result$top_categories)

## End(Not run)
```

summarize *Summarize web content using LLMs*

Description

Wraps the Python `catweb.summarize()` function. Accepts URLs (auto-fetched) or raw text. Web context (source domain, content type, metadata) is injected into the summarization prompt.

Usage

```
summarize(
  input_data = NULL,
  source_domain = NULL,
  content_type = NULL,
  web_metadata = NULL,
  timeout = 30L,
  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 = 2L,
  retry_delay = 1,
  row_delay = 0,
  fail_strategy = "partial",
  batch_mode = FALSE,
  batch_poll_interval = 30,
```

```
  batch_timeout = 86400
)
```

Arguments

input_data	Data to summarize: URLs, text, or data.frame column.
source_domain	Character or NULL. Source domain context.
content_type	Character or NULL. Content type context.
web_metadata	Named list or NULL. Additional metadata.
timeout	Integer. URL fetch timeout (seconds). Default 30L.
api_key	Character or NULL. API key for the LLM provider.
description	Character. Default "".
instructions	Character. Specific instructions for the summary. Default "".
format	Character. Default "paragraph".
max_length	Integer or NULL. Default NULL.
focus	Character or NULL. Default NULL.
user_model	Character. Default "gpt-4o".
model_source	Character. Default "auto".
mode	Character. Default "image".
input_mode	Character or NULL. Default NULL.
input_type	Character. Default "auto".
pdf_dpi	Integer. Default 150L.
creativity	Numeric or NULL. Default NULL.
thinking_budget	Integer. Default 0L.
chain_of_thought	Logical. Default TRUE.
context_prompt	Logical. Default FALSE.
step_back_prompt	Logical. Default FALSE.
filename	Character or NULL.
save_directory	Character or NULL.
models	List of model specs for ensemble mode. Default NULL.
max_workers	Integer or NULL. Default NULL.
parallel	Logical or NULL. Default NULL.
auto_download	Logical. Default FALSE.
safety	Logical. Default FALSE.
max_retries	Integer. Default 5L.
batch_retries	Integer. Default 2L.

retry_delay	Numeric. Default 1.0.
row_delay	Numeric. Default 0.0.
fail_strategy	Character. Default "partial".
batch_mode	Logical. Default FALSE.
batch_poll_interval	Numeric. Default 30.0.
batch_timeout	Numeric. Default 86400.0.

Value

A data.frame with summarization results.

Examples

```
## Not run:
summaries <- summarize(
  input_data    = c("https://example.com/article-1",
                    "https://example.com/article-2"),
  source_domain = "example.com",
  content_type  = "news article",
  format        = "bullets",
  api_key       = Sys.getenv("OPENAI_API_KEY"),
  user_model    = "gpt-4o-mini"
)

## End(Not run)
```

Index