Package: cat.pol (via r-universe)

June 4, 2026

Title Political Document Classification with LLMs

Version 0.1.2

Description R interface to the Python catpol package. Classifies, extracts, explores, and summarizes political and policy documents using LLMs. A thin domain wrapper around cat.stack that adds a registered-source fetcher (city ordinances, federal laws, executive orders, presidential speeches, social media) and policy-document prompt framing.

License GPL (>= 3)

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

Contents

`classify` *Classify political and policy documents using LLMs*

Description

Wraps the Python `catpol.classify()` function. Can classify either raw text (via `input_data`) or pull directly from a registered political data source (via `source`). All catstack classification arguments are supported.

Usage

```
classify(
  categories,
  input_data = NULL,
  source = NULL,
  doc_type = NULL,
  since = NULL,
  until = NULL,
  n = NULL,
  document_context = "",
  description = "",
  api_key = NULL,
  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",
  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, or data.frame column, or NULL to fetch from a registered source. Default NULL.
source	Character or NULL. Registered source name (e.g. "city_san_diego", "federal_laws", "federal_executive_orders", "social_trump_truth"). Use list_sources() for all options.
doc_type	Character or NULL. Filter source by document type (e.g. "ordinance", "resolution").
since	Character or NULL. Earliest source row date (YYYY-MM-DD).
until	Character or NULL. Latest source row date (YYYY-MM-DD).
n	Integer or NULL. Max number of source rows to classify.
document_context	Character. Context about the policy document being analyzed. Default "".
description	Character. Additional context description. Default "".
api_key	Character or NULL. API key for the LLM provider.
user_model	Character. Model name. Default "gpt-4o".
mode	Character. Processing mode. Default "image".
creativity	Numeric or NULL. Temperature. Default NULL.
safety	Logical. Save progress after each item. 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.

`filename` Character or `NULL`. Output CSV filename.

`save_directory` Character or `NULL`. Output directory.

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

`max_categories` Integer. Default `12L`.

`categories_per_chunk`
: Integer. Default `10L`.

`divisions` Integer. Default `10L`.

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

`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:
# Pull recent San Diego ordinances from a registered source
results <- classify(
  source     = "city_san_diego",
  doc_type   = "ordinance",
  since      = "2024-01-01",
  n          = 50L,
  categories = c("Housing", "Public Safety", "Finance",
                 "Infrastructure", "Health"),
  api_key    = Sys.getenv("OPENAI_API_KEY"),
  user_model = "gpt-4o-mini"
)

# Or classify your own text directly
results <- classify(
  input_data = df$bill_text,
  categories = c("Housing", "Public Safety", "Finance"),
  api_key    = Sys.getenv("OPENAI_API_KEY")
)

## End(Not run)
```

`explore` *Explore raw categories in political and policy documents*

Description

Wraps the Python `catpol.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 = NULL,
  doc_type = NULL,
  since = NULL,
  until = NULL,
  n = NULL,
  document_context = "",
  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
)
```

Arguments

input_data	A character vector, list, or NULL to fetch from a registered source. Default NULL.
api_key	Character or NULL. API key for the LLM provider.
source	Character or NULL. Registered source name.
doc_type	Character or NULL. Filter source by document type.
since	Character or NULL. Earliest source row date (YYYY-MM-DD).
until	Character or NULL. Latest source row date (YYYY-MM-DD).
n	Integer or NULL. Max number of source rows.
document_context	Character. Context about the document. Default "".
description	Character. Additional context. Default "".
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(
  source     = "federal_executive_orders",
  since      = "2025-01-01",
  n          = 30L,
  api_key    = Sys.getenv("OPENAI_API_KEY"),
  user_model = "gpt-4o-mini",
  iterations = 4L
)
sort(table(raw_cats), decreasing = TRUE)

## End(Not run)
```

`extract` *Discover categories from political and policy documents using LLMs*

Description

Wraps the Python `catpol.extract()` function. Returns a normalised, deduplicated set of categories from policy text or a registered source.

Usage

```
extract(
  input_data = NULL,
  api_key = NULL,
  source = NULL,
  doc_type = NULL,
  since = NULL,
  until = NULL,
  n = NULL,
  document_context = "",
  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
)
```

Arguments

input_data	A character vector, list, or NULL to fetch from a registered source. Default NULL.
api_key	Character or NULL. API key for the LLM provider.
source	Character or NULL. Registered source name. Use list_sources() for options.
doc_type	Character or NULL. Filter source by document type.
since	Character or NULL. Earliest source row date (YYYY-MM-DD).
until	Character or NULL. Latest source row date (YYYY-MM-DD).
n	Integer or NULL. Max number of source rows.
document_context	Character. Context about the document. Default "".
description	Character. Additional context. Default "".
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       = df$bill_text,
  document_context = "California state legislation",
  api_key          = Sys.getenv("OPENAI_API_KEY"),
  user_model       = "gpt-4o-mini"
)
print(result$top_categories)

## End(Not run)
```

`list_sources` *List registered political data sources*

Description

Returns the names of all data sources registered with the Python `catpol` package (city ordinances, federal laws, executive orders, presidential speeches, social media archives, etc.).

Usage

```
list_sources()
```

Value

A character vector of source names.

Examples

```
## Not run:
list_sources()
#> [1] "city_san_diego"           "city_san_francisco"
#> [3] "federal_laws"             "federal_executive_orders"
#> [5] "social_trump_truth"       ...

## End(Not run)
```

summarize *Summarize political and policy documents using LLMs*

Description

Wraps the Python `catpol.summarize()` function. Generates summaries from policy text or from a registered political data source. Adds a `tone` parameter for policy-specific framing.

Usage

```
summarize(
  input_data = NULL,
  source = NULL,
  doc_type = NULL,
  since = NULL,
  until = NULL,
  n = NULL,
  format = "paragraph",
  tone = "eli5",
  api_key = NULL,
  description = "",
  instructions = "",
  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	A character vector, list, or PDF/URL paths; NULL to fetch from a registered source.
source	Character or NULL. Registered source name.
doc_type	Character or NULL. Filter source by document type.
since	Character or NULL. Earliest source row date (YYYY-MM-DD).
until	Character or NULL. Latest source row date (YYYY-MM-DD).
n	Integer or NULL. Max number of source rows.
format	Character. Output format. Default "paragraph".
tone	Character. Policy-specific tone, e.g. "eli5", "neutral", "academic". Default "eli5".
api_key	Character or NULL. API key for the LLM provider.
description	Character. Default "".
instructions	Character. Specific instructions for the summary. Default "".
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:
results <- summarize(
  source     = "federal_executive_orders",
  since      = "2025-01-01",
  n          = 20L,
  format     = "paragraph",
  tone       = "eli5",
  api_key    = Sys.getenv("OPENAI_API_KEY"),
  user_model = "gpt-4o-mini"
)

## End(Not run)
```

Index