

Package: rdomains (via r-universe)

October 9, 2026

Title Get the Category of Content Hosted by a Domain

Version 0.5.0

Description Get the category of content hosted by a domain. Use Shallalist (service discontinued), 'VirusTotal' (which provides access to lots of services) <<https://www.virustotal.com/>>, 'DMOZ' <<https://archive.org/details/dmoz-rdf-20150327>>, University Domain list <<https://github.com/Hipo/university-domains-list>>, 'OpenAI' 'GPT' models, 'Anthropic' 'Claude' models, or validated machine learning classifiers based on 'Shallalist' data to learn about the kind of content hosted by a domain.

Depends R (>= 4.1.0)

Imports Matrix, urltools, glmnet, stats, methods, XML, httr, httr2, xml2, curl, virustotal, jsonlite, R.utils, dplyr (>= 1.1.0), purrr (>= 1.0.0), tibble (>= 3.2.0), stringr (>= 1.5.0), rlang (>= 1.1.0), cli (>= 3.6.0), checkmate (>= 2.3.0), glue (>= 1.6.0), readr (>= 2.1.0)

Suggests testthat, rmarkdown, knitr (>= 1.11), withr

VignetteBuilder knitr

License MIT + file LICENSE

Encoding UTF-8

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

NeedsCompilation no

Author Gaurav Sood [aut, cre]

Maintainer Gaurav Sood <gsood07@gmail.com>

Config/pak/sysreqs libicu-dev libxml2-dev libssl-dev libx11-dev

Repository <https://soodoku.r-universe.dev>

Date/Publication 2026-08-02 09:00:02 UTC

RemoteUrl <https://github.com/cran/rdomains>

RemoteRef HEAD

RemoteSha 4e5993b435a10560ed926176dfe552b4b4a763ea

Contents

rdomains-package	2
adult_ml1_cat	3
claude_cat	3
collect_content	4
dmoz_cat	6
fetch_error_codes	7
fetch_report	7
get_dmoz_data	8
get_shalla_data	9
get_stevenblack_data	9
glm_shalla	10
html_text_content	11
not_news	11
openai_cat	12
page_signals	13
shalla_cat	14
source_vintage	15
stevenblack_cat	16
uni_cat	17
virustotal_cat	17
Index	19

rdomains-package	<i>rdomains: Classify Domains by their Content</i>
------------------	--

Description

Want to know what kind of content is carried on a domain? Get the results quickly using rdomains. The package provides access to virustotal API, shalla, aws, OpenAI GPT models, Anthropic Claude models, and validated ML model based off shallalist data to predict content of a domain.

Details

To learn how to use rdomains, see this vignette: [../doc/rdomains.html](#).

Author(s)

Gaurav Sood

adult_ml1_cat	<i>Probability that Domain Hosts Adult Content Based on features of Domain Name and Suffix alone.</i>
---------------	---

Description

Uses a validated ML model that uses keywords in the domain name and suffix to predict probability that the domain hosts adult content. For more information see https://github.com/themains/keyword_porn

Usage

```
adult_ml1_cat(domains = NULL)
```

Arguments

domains required; string; vector of domain names

Value

data.frame with original list and content category of the domains

Examples

```
## Not run:  
adult_ml1_cat("http://www.google.com")  
  
## End(Not run)
```

claude_cat	<i>Get Category from Anthropic Claude</i>
------------	---

Description

Fetches category of content hosted by a domain using Anthropic's Claude API. The function uses Claude models to classify domains into specified categories.

Usage

```
claude_cat(  
  domains = NULL,  
  api_key = NULL,  
  categories = NULL,  
  model = "claude-3-haiku-20240307",  
  rate_limit = 0.5  
)
```

Arguments

domains	vector of domain names
api_key	Anthropic API key. If not provided, looks for ANTHROPIC_API_KEY or CLAUDE_API_KEY environment variable
categories	vector of categories to classify into. If NULL, uses default web categories
model	Claude model to use (default: "claude-3-haiku-20240307" for cost efficiency)
rate_limit	delay in seconds between API calls (default: 0.5)

Value

data.frame with original list and content category of the domain

Examples

```
## Not run:
claude_cat("google.com")
claude_cat(c("google.com", "facebook.com"))
claude_cat("google.com", categories = c("search", "social", "ecommerce", "news", "other"))

## End(Not run)
```

collect_content	<i>Fetch homepage HTML and text for domains</i>
-----------------	---

Description

Every requested domain comes back, in order, whether or not it was reachable. Failures carry a code from [fetch_error_codes()] rather than a bare 'NA'.

Usage

```
collect_content(
  domains = NULL,
  delay = 1,
  timeout = 10,
  max_bytes = 2 * 1024^2,
  obey_robots = TRUE,
  max_crawl_delay = 30,
  max_redirects = 5,
  user_agent = rdomains_user_agent()
)
```

Arguments

domains	Character vector of domains or URLs.
delay	Minimum seconds between requests to the same host. 'Crawl-delay' overrides this upward.
timeout	Per-request timeout, seconds.
max_bytes	Cap on the response body actually read.
obey_robots	Whether to fetch and honour robots.txt. Turning this off is discouraged and is your responsibility, not the package's.
max_crawl_delay	Skip a host that asks for a longer delay than this rather than sleeping on it.
max_redirects	Maximum redirect hops to follow. Every hop is re-validated, so a redirect cannot be used to reach an address the first check refused.
user_agent	Override the identifying user-agent.

Details

The crawler identifies itself as 'rdomains/<version>', obeys 'robots.txt' including 'Crawl-delay', spaces requests to the same host, caps the response body, and refuses to fetch hosts that resolve to private or link-local addresses.

Value

A tibble with one row per input: 'domain_name', 'status', 'stage', 'error_code', 'retryable', 'http_status', 'final_url', 'fetched_at', 'content_bytes', 'title', 'description', 'lang', 'text', 'n_tokens', 'page_state', 'block_vendor', 'robots_allowed', 'source_last_published'.

See Also

[fetch_report()] to summarise the run, [page_signals()] for what the page states are, [source_vintage()] for how a live fetch compares with the static lists.

Examples

```
## Not run:
res <- collect_content(c("example.com", "wikipedia.org"))
fetch_report(res)

# retry only what is worth retrying
again <- collect_content(res$domain_name[res$retryable])

## End(Not run)
```

`dmoz_cat`*Get Category from DMOZ*

Description

Fetches category (or categories) of content hosted by a domain according to DMOZ. The function checks if path to the DMOZ file is provided by the user. If not, it looks for `dmoz_domain_category.csv` in the working directory. It also returns results for prominent subdomains.

Usage

```
dmoz_cat(domains = NULL, use_file = NULL)
```

Arguments

<code>domains</code>	vector of domain names
<code>use_file</code>	path to the dmoz file, which can be downloaded using get_dmoz_data

Details

DMOZ closed in March 2017 and the snapshot shipped here is from 2015, so these labels describe the domain as it was a decade ago. The returned `source_last_published` column carries that vintage, and a warning is issued once per session. See [source_vintage](#).

Value

`data.frame` with the original list, the content category of the domain, and the vintage of the list that supplied it

See Also

[source_vintage](#) for the provenance of every category source

Examples

```
## Not run:
dmoz_cat(domains = "http://www.google.com")
dmoz_cat(domains = c("http://www.google.com", "http://plus.google.com"))

## End(Not run)
```

fetch_error_codes	<i>Reasons a domain can fail to get a category</i>
-------------------	--

Description

The closed set of ‘error_code’ values that [collect_content()] and [pie_cat()] can return, with the pipeline stage each belongs to and whether it is worth retrying.

Usage

```
fetch_error_codes()
```

Details

These strings are shared verbatim with the Python package ‘piedomains’, so results from the two can be aggregated together.

Value

A tibble with columns ‘code’, ‘stage’, ‘retryable’ and ‘description’.

See Also

[fetch_report()] to summarise an actual run, [source_vintage()] for the provenance of the static label sources.

Examples

```
fetch_error_codes()

# which failures are worth trying again?
subset(fetch_error_codes(), retryable)
```

fetch_report	<i>Summarise the outcome of a fetch or classification run</i>
--------------	---

Description

A pure function over the returned tibble, deliberately **not** a stored summary. A snapshot goes stale the moment you filter the rows; recomputing means ‘fetch_report(subset(x, retryable))’ is always correct.

Usage

```
fetch_report(x)
```

Arguments

x A tibble from [collect_content()] or [pie_cat()].

Value

A one-row tibble: 'n', 'ok', 'failed', 'retryable', and list-columns 'by_stage', 'by_reason' and 'missing' (the domains that produced nothing).

Examples

```
rows <- data.frame(
  domain_name = c("a.com", "b.com", "c.com"),
  status = c("ok", "failed", "failed"),
  stage = c("infer", "fetch", "process"),
  error_code = c(NA, "timeout", "thin_content"),
  retryable = c(FALSE, TRUE, FALSE)
)
fetch_report(rows)
```

get_dmoz_data

Get DMOZ Data

Description

Downloads archived DMOZ (Open Directory Project) data. DMOZ was discontinued in March 2017. This function downloads our preserved copy of the final DMOZ dataset. For more details, check: <https://github.com/themains/rdomains/tree/master/data-raw/dmoz/>

Usage

```
get_dmoz_data(outdir = ".", overwrite = FALSE)
```

Arguments

outdir Optional; folder to which you want to save the file; Default is same folder

overwrite Optional; default is FALSE. If TRUE, the file is overwritten.

References

<https://archive.org/details/dmoz-rdf-20150327>

Examples

```
## Not run:
get_dmoz_data()

## End(Not run)
```

get_shalla_data	<i>Get Shalla Data</i>
-----------------	------------------------

Description

Shallalist service was discontinued in January 2022. This function downloads the last archived copy (from 1/14/22) that we have preserved on GitHub. The original service at shallalist.de is no longer available. Downloads, unzips and saves the final version of shallalist data. By default, saves shalla data as shalla_domain_category.csv.

Usage

```
get_shalla_data(outdir = "./", overwrite = FALSE)
```

Arguments

outdir	Optional; folder to which you want to save the file; Default is same folder
overwrite	Optional; default is FALSE. If TRUE, the file is overwritten.

References

<https://web.archive.org/web/20210502020725/http://www.shallalist.de/>

Examples

```
## Not run:  
get_shalla_data()  
  
## End(Not run)
```

get_stevenblack_data	<i>Get Steven Black's Host List Data</i>
----------------------	--

Description

Downloads the latest version of Steven Black's unified hosts file. The hosts file contains domains known for serving ads, malware, and tracking.

Usage

```
get_stevenblack_data(outdir = "./", variant = "base", overwrite = FALSE)
```

Arguments

outdir	Optional; folder to which you want to save the file; Default is current directory
variant	Optional; which variant to download. Options: "base", "porn", "social", "gambling", "all"
overwrite	Optional; default is FALSE. If TRUE, the file is overwritten.

References

<https://github.com/StevenBlack/hosts>

Examples

```
## Not run:  
get_stevenblack_data()  
get_stevenblack_data(variant = "all")  
  
## End(Not run)
```

glm_shalla

ML Model

Description

ML Model

Usage

glm_shalla

Format

A list

Author(s)

Gaurav Sood

Source

ML model based on shallalist using keywords and domain suffixes,

html_text_content	<i>Extract text, title, description and language from HTML</i>
-------------------	--

Description

Extract text, title, description and language from HTML

Usage

```
html_text_content(html)
```

Arguments

html	Raw HTML as a single string.
------	------------------------------

Value

A list with ‘text’ (cleaned, lowercased), ‘title’, ‘description’ and ‘lang’. Missing elements are ‘NA_character_’; ‘text’ is “” when nothing could be extracted.

See Also

[page_signals()], which uses this to decide whether a page is classifiable.

Examples

```
html <- "<html lang='en'><head><title>Example</title>
<meta name='description' content='A demo page'>
<script>ignored()</script></head>
<body><p>Hello World</p><style>p{}</style></body></html>"
html_text_content(html)
```

not_news	<i>Classify News and Non-News Based on keywords in the URL</i>
----------	--

Description

Based on a slightly amended version of the regular expression used to classify news, and non-news in: “Exposure to ideologically diverse news and opinion on Facebook” by Bakshy, Messing, and Adamic. Science. 2015.

Usage

```
not_news(url_list = NULL)
```

Arguments

url_list vector of URLs

Details

Amendment: sport rather than sports

URL containing any of the following words is classified as soft news: "sport|entertainment|arts|fashion|style|lifestyle|leisure|ce

URL containing any of following words is classified as hard news: "politi|usnews|world|national|state|elect|vote|govern|campa

Note that it is based on patterns existing in a small set of domains. See paper for details.

Value

data.frame with 3 columns: url, not_news, news

References

<https://www.science.org/doi/10.1126/science.aaa1160>

Examples

```
## Not run:
not_news("http://www.bbc.com/sport")
not_news(c("http://www.bbc.com/sport", "http://www.washingtontimes.com/news/politics/"))

## End(Not run)
```

openai_cat

Get Category from OpenAI

Description

Fetches category of content hosted by a domain using OpenAI's chat completion API. The function uses GPT models to classify domains into specified categories.

Usage

```
openai_cat(
  domains = NULL,
  api_key = NULL,
  categories = NULL,
  model = "gpt-4o-mini",
  rate_limit = 0.5
)
```

Arguments

domains	vector of domain names
api_key	OpenAI API key. If not provided, looks for OPENAI_API_KEY environment variable
categories	vector of categories to classify into. If NULL, uses default web categories
model	OpenAI model to use (default: "gpt-4o-mini" for cost efficiency)
rate_limit	delay in seconds between API calls (default: 0.5)

Value

data.frame with original list and content category of the domain

Examples

```
## Not run:
openai_cat("google.com")
openai_cat(c("google.com", "facebook.com"))
openai_cat("google.com", categories = c("search", "social", "ecommerce", "news", "other"))

## End(Not run)
```

page_signals	<i>What kind of page is this?</i>
--------------	-----------------------------------

Description

Inspects already-fetched HTML for the three things that are not a classifiable site: an anti-bot interstitial, a domain-parking placeholder, and a server's "nothing here" page. No network access.

Usage

```
page_signals(html, text = NULL, domain = "", status = NULL)
```

Arguments

html	Raw response body.
text	Extracted page text. If 'NULL', derived from 'html'.
domain	The domain requested, used to spot a page whose only content is its own name.
status	HTTP status code, if known.

Details

'parked' and 'unavailable' are **answers**, not failures – they are facts about the domain, plainly stated in the page, that a caller can act on. They are also free: no classification service needs to be consulted.

Value

A one-row tibble: 'page_state' (one of "content", "blocked", "parked", "unavailable", "thin"), 'blocked', 'block_vendor', 'block_reason', 'parked', 'unavailable', 'thin', 'n_tokens'.

See Also

[fetch_error_codes()] for how these map onto run outcomes.

Examples

```
# A Cloudflare challenge, not a website
page_signals("<html><title>Just a moment...</title><body>cf_chl_opt</body></html>")

# A parking page
page_signals("<html><body>This domain is for sale. Inquire now.</body></html>")

# A real page that merely embeds reCAPTCHA is not blocked
page_signals(paste0("<html><title>Reddit</title><body>",
  paste(rep("real discussion content", 200), collapse = " "),
  "<script src='recaptcha/api.js'></script></body></html>"))
```

shalla_cat	<i>Get Category from Shallalist</i>
------------	-------------------------------------

Description

Fetches category of content hosted by a domain according to Shalla. The function checks if path to the shalla file is provided by the user. If not, it looks for shalla_domain_category.csv in the working directory.

Usage

```
shalla_cat(domains = NULL, use_file = NULL)
```

Arguments

domains	vector of domain names
use_file	path to the latest shallalist file downloaded using get_shalla_data

Details

Shallalist stopped publishing in 2022, so these labels describe the domain as it was then or earlier, not as it is now. The returned source_last_published column carries that vintage, and a warning is issued once per session. See [source_vintage](#).

Value

data.frame with the original list, the content category of the domain, and the vintage of the list that supplied it

See Also

[source_vintage](#) for the provenance of every category source

Examples

```
## Not run:
shalla_cat(domains = "http://www.google.com")

## End(Not run)
```

source_vintage	<i>Report what each category source is and when it was last published</i>
----------------	---

Description

Every lookup in this package answers from a static list, and those lists have very different vintages. Two of them are no longer published at all. Use this to see which answers can be trusted to describe a domain as it is today.

Usage

```
source_vintage(source = NULL)
```

Arguments

source	Optional; one of "shalla", "dmoz", "stevenblack", "uni". If NULL, every source is returned.
--------	---

Value

A tibble with one row per source: key, source, last_published, status, successor and note.

Examples

```
source_vintage()
source_vintage("dmoz")
```

stevenblack_cat*Get Category from Steven Black's Host List*

Description

Classifies domains based on Steven Black's unified host list which blocks ads, malware, and tracking domains. The function checks if a domain appears in the blocklist and categorizes it accordingly.

Usage

```
stevenblack_cat(domain = NULL, use_file = NULL)
```

Arguments

domain	domain names as character vector
use_file	path to a local Steven Black hosts file. If NULL, downloads from GitHub once per session and reuses it

Details

Steven Black's host list is a consolidated list from multiple sources including adaway.org, mvps.org, malwaredomainlist.com, and someonewhocares.org.

Unlike the Shallalist and DMOZ lookups, this list is actively maintained, so the returned `source_last_published` is the fetched file's own modification date rather than a constant. See [source_vintage](#).

Value

data.frame with the original domain name, the category, and the date of the list that supplied it

References

<https://github.com/StevenBlack/hosts>

See Also

[source_vintage](#) for the provenance of every category source

Examples

```
## Not run:  
stevenblack_cat("doubleclick.net")  
stevenblack_cat(c("google.com", "googleadservices.com", "malware-example.com"))  
  
## End(Not run)
```

`uni_cat`*Get Category from University Domain List*

Description

Fetches university domain json from: https://raw.githubusercontent.com/Hipo/university-domains-list/master/world_universities_and_domains.json

Usage

```
uni_cat(domains = NULL)
```

Arguments

`domains` vector of domain names

Value

data.frame with original list and all the other columns from the university json

Examples

```
## Not run:  
uni_cat(domains = "http://www.google.com")  
  
## End(Not run)
```

`virustotal_cat`*Get Category from VirusTotal*

Description

Returns category of content from multiple security vendors using the VirusTotal API v3. The function retrieves domain analysis results including categories from various security services. Not all services will have categories for all domains.

Usage

```
virustotal_cat(domains = NULL, apikey = NULL)
```

Arguments

`domains` domain names as character vector
`apikey` virustotal API key

Details

Get the API Access Key from <https://www.virustotal.com/>. Either pass the API Key to the function or set the environmental variable: VirustotalToken. Environment variables persist within a R session.

Value

data.frame with domain and VirusTotal analysis results

References

<https://docs.virustotal.com/reference/domains>

Examples

```
## Not run:  
virustotal_cat("http://www.google.com")  
virustotal_cat(c("google.com", "facebook.com"))  
  
## End(Not run)
```

Index

- * **keywords**

- glm_shalla, [10](#)

- * **model**

- glm_shalla, [10](#)

adult_ml1_cat, [3](#)

claudes_cat, [3](#)

collect_content, [4](#)

dmoz_cat, [6](#)

fetch_error_codes, [7](#)

fetch_report, [7](#)

get_dmoz_data, [6](#), [8](#)

get_shalla_data, [9](#), [14](#)

get_stevenblack_data, [9](#)

glm_shalla, [10](#)

html_text_content, [11](#)

not_news, [11](#)

openai_cat, [12](#)

page_signals, [13](#)

rdomains (rdomains-package), [2](#)

rdomains-package, [2](#)

shalla_cat, [14](#)

source_vintage, [6](#), [14](#), [15](#), [15](#), [16](#)

stevenblack_cat, [16](#)

uni_cat, [17](#)

virustotal_cat, [17](#)