Package: AIscreenR (via r-universe)

November 13, 2024

Title AI Screening Tools in R for Systematic Reviewing

Version 0.1.0.9000

Description Provides functions to conduct title and abstract screening
in systematic reviews using large language models, such as the
Generative Pre-trained Transformer (GPT) models from 'OpenAI'
<https://platform.openai.com/>. These functions can enhance the
quality of title and abstract screenings while reducing the
total screening time significantly. In addition, the package
includes tools for quality assessment of title and abstract
screenings, as described in Vembye, Christensen, Mølgaard, and
Schytt (2024) <DOI:10.31219/osf.io/yrhzm>.

Depends R (>= 4.1.0)

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Imports dplyr, tibble, httr2, stringr, furrr, tidyr, tictoc, askpass,
curl, purrr, lifecycle, jsonlite

Suggests future, knitr, rmarkdown, usethis, testthat (>= 3.0.0)

Config/testthat/edition 3

Config/testthat/parallel true

Config/testthat/start-first tabscreen_gpt, approximate_price_gpt,
rate_limits, api_key_functions

URL https://mikkelvembye.github.io/AIscreenR/,
https://github.com/MikkelVembye/AIscreenR

BugReports https://github.com/MikkelVembye/AIscreenR/issues

LazyData true

VignetteBuilder knitr

Config/pak/sysreqs libicu-dev libssl-dev

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

RemoteUrl https://github.com/mikkelvembye/aiscreenr

RemoteRef HEAD

RemoteSha 754b0c965c92a1ca0b778079bc229ca6e4f809b3

Contents

---

approximate_price_gpt   *Approximate price estimation for title and abstract screening using OpenAI's GPT API models*

Description

[Experimental]

This function supports the approximation of the price of title and abstract screenings when using OpenAI's GPT API models. The function only provide approximately accurate price estimates. When detailed descriptions are used, this will increase the completion tokens with an unknown amount.

Usage

```
approximate_price_gpt(
  data,
  prompt,
  studyid,
  title,
  abstract,
  model = "gpt-4o-mini",
  reps = 1,
  top_p = 1,
  token_word_ratio = 1.6
)
```

Arguments

data	Dataset containing the titles and abstracts.
prompt	Prompt(s) to be added before the title and abstract.
studyid	Unique Study ID. If missing, this is generated automatically.
title	Name of the variable containing the title information.
abstract	Name of variable containing the abstract information.
model	Character string with the name of the completion model. Can take multiple models, including gpt-4 models. Default = "gpt-4o-mini". Find available model at <https://platform.openai.com/docs/models/model-endpoint-compatibility>.
reps	Numerical value indicating the number of times the same question should be sent to the GPT server. This can be useful to test consistency between answers. Default is 1 but when using gpt-3.5-turbo or gpt-4o-mini models, we recommend setting this value to 10.
top_p	'An alternative to sampling with temperature, called nucleus sampling, where the model considers the results of the tokens with top_p probability mass. So 0.1 means only the tokens comprising the top 10% probability mass are considered. We generally recommend altering this or temperature but not both.' (OPEN-AI). Default is 1. Find documentation at <https://platform.openai.com/docs/api-reference/chat/create#chat/create-top_p>.
token_word_ratio	The multiplier used to approximate the number of tokens per word. Default is 1.6 which we empirically have found to be the average number of tokens per word.

Value

An object of class "gpt_price". The object is a list containing the following components:

price	numerical value indicating the total approximate price (in USD) of the screening across all gpt-models expected to be used for the screening.
price_data	dataset with prices across all gpt models expected to be used for screening.

Examples

```
prompt <- "This is a prompt"

app_price <- approximate_price_gpt(
  data = filges2015_dat[1:2,],
  prompt = prompt,
  studyid = studyid,
  title = title,
  abstract = abstract,
  model = c("gpt-4o-mini", "gpt-4"),
  reps = c(10, 1)
)

app_price
app_price$price_dollar
app_price$price_data
```

---

filges2015_dat   *RIS file data from Functional Family Therapy (FFT) systematic review*

Description

Bibliometric toy data from a systematic review regarding Functional Family Therapy (FFT) for Young People in Treatment for Non-opioid Drug Use (Filges et al., 2015). The data includes all 90 included and 180 excluded randomly sampled references from the literature search of the systematic review.

Usage

```
filges2015_dat
```

Format

A `tibble` with 270 rows/studies and 6 variables/columns

author	character	indicating the authors of the reference
eppi_id	character	indicating a unique eppi-ID for each study
studyid	numeric	indicating a unique study-ID for each study
title	character	with the title of the study
abstract	character	with the study abstract
human_code	numeric	indicating the human screening decision. 1 = included, 0 = excluded.

References

Filges, T., Andersen, D, & Jørgensen, A-M. K (2015). Functional Family Therapy (FFT) for Young People in Treatment for Non-opioid Drug Use: A Systematic Review *Campbell Systematic Reviews*, [doi:10.4073/csr.2015.14](doi:10.4073/csr.2015.14)

---

get_api_key   *Get API key from R environment variable.*

Description

Get API key from R environment variable.

Usage

```
get_api_key(env_var = "CHATGPT_KEY")
```

Arguments

env_var   Character string indicating the name of the temporary R environment variable
                 with the API key and the used AI model. Currently, the argument only takes
                 env_var = "CHATGPT_KEY". See set_api_key() to set/create this variable.

Details

get_api_key() can be used after executing set_api_key() or by adding the api key permanently
to your R environment by using usethis::edit_r_environ(). Then write CHATGPT_KEY=[insert your api key here]
and close the .Renviron window and restart R.

Value

The specified API key (NOTE: Avoid exposing this in the console).

Note

Find your personal API key at https://platform.openai.com/account/api-keys.

See Also

set_api_key.

Examples

```
## Not run:
get_api_key()

## End(Not run)
```

---

is_chatgpt   *Test if the object is a* 'chatgpt' *object*

Description

[Deprecated]

This function returns TRUE for chatgpt objects, and FALSE for all other objects.

Usage

```
is_chatgpt(x)
```

Arguments

x   An object

Value

TRUE if the object inherits from the chatgpt class.

---

is_chatgpt_tbl   *Test if the object is a* 'chatgpt_tbl' *object*

Description

[Deprecated]

This function returns TRUE for chatgpt_tbl objects, and FALSE for all other objects.

Usage

```
is_chatgpt_tbl(x)
```

Arguments

x   An object

Value

TRUE if the object inherits from the chatgpt_tbl class.

---

is_gpt   *Test if the object is a* 'gpt' *object*

Description

This function returns TRUE for gpt objects, and FALSE for all other objects.

Usage

```
is_gpt(x)
```

Arguments

x   An object

Value

TRUE if the object inherits from the gpt class.

---

is_gpt_agg_tbl   *Test if the object is a* 'gpt_agg_tbl' *object*

Description

This function returns TRUE for gpt_agg_tbl objects, and FALSE for all other objects.

Usage

```
is_gpt_agg_tbl(x)
```

Arguments

x   An object

Value

TRUE if the object inherits from the gpt_agg_tbl class.

---

is_gpt_tbl   *Test if the object is a* 'gpt_tbl' *object*

Description

This function returns TRUE for gpt_tbl objects, and FALSE for all other objects.

Usage

```
is_gpt_tbl(x)
```

Arguments

x   An object

Value

TRUE if the object inherits from the gpt_tbl class.

---

model_prizes   *Model prize data (last updated November 5, 2024)*

Description

Data set containing input and output prizes for all OpenAI's GPT API models.

Usage

```
model_prizes
```

Format

A data.frame containing 15 rows/models and 3 variables/columns

model	character	indicating the specific GPT model
price_in_per_token	character	indicating the input prize per token
price_out_per_token	character	indicating the output prize per token

References

OpenAI. *Pricing.* https://openai.com/api/pricing/

---

print.chatgpt   *Print methods for* 'chatgpt' *objects*

Description

Print methods for 'chatgpt' objects

Usage

```
## S3 method for class 'chatgpt'
print(x, ...)
```

Arguments

x   an object of class 'chatgpt'.

...   other print arguments.

Value

Information about how to find answer data sets and pricing information.

Examples

```
## Not run:
print(x)

## End(Not run)
```

---

print.gpt   *Print methods for* 'gpt' *objects*

Description

Print methods for 'gpt' objects

Usage

```
## S3 method for class 'gpt'
print(x, ...)
```

Arguments

x   an object of class 'gpt'.

...   other print arguments.

Value

Information about how to find answer data sets and pricing information.

Examples

```
## Not run:
print(x)

## End(Not run)
```

---

print.gpt_price   *Print methods for* 'gpt_price' *objects*

Description

Print methods for 'gpt_price' objects

Usage

```
## S3 method for class 'gpt_price'
print(x, ...)
```

Arguments

x	an object of class "gpt_price".
...	other print arguments.

Value

The total price of the screening across all gpt-models expected to be used for the screening.

Examples

```
## Not run:
print(x)

## End(Not run)
```

```
rate_limits_per_minute
```

Find updated rate limits for API models

Description

[Stable]

`rate_limits_per_minute` reports the rate limits for a given API model. The function returns the available requests per minute (RPM) as well as tokens per minute (TPM). Find general information at https://platform.openai.com/docs/guides/rate-limits/overview.

Usage

```
rate_limits_per_minute(
  model = "gpt-4o-mini",
  AI_tool = "gpt",
  api_key = get_api_key()
)
```

Arguments

`model`	Character string with the name of the completion model. Default is `"gpt-4o-mini"`. Can take multiple values. Find available model at https://platform.openai.com/docs/models/model-endpoint-compatibility.
`AI_tool`	Character string specifying the AI tool from which the API is issued. Default is `"gpt"`.
`api_key`	Numerical value with your personal API key. Default setting draws on the `get_api_key()` to retrieve the API key from the R environment, so that the key is not compromised. The API key can be added to the R environment via `set_api_key()` or by using `usethis::edit_r_environ()`. In the .Renviron file, write CHATGPT_KEY=INSERT_YOUR_KEY_HERE. After entering the API key, close and save the .Renviron file and restart RStudio (ctrl + shift + F10). Alternatively, one can use `httr2::secret_make_key()`, `httr2::secret_encrypt()`, and `httr2::secret_decrypt()` to scramble and decrypt the API key.

Value

A `tibble` including variables with information about the model used, the number of requests and tokens per minute.

Examples

```
## Not run:
set_api_key()

rate_limits_per_minute()

## End(Not run)
```

---

sample_references   *Random sample references*

Description

sample_referencessamples n rows from the dataset with titles and abstracts either with or without replacement. This function is supposed to support the construct of a test dataset, as suggested by Vembye et al. (2024).

Usage

```
sample_references(
  data,
  n,
  with_replacement = FALSE,
  prob_vec = rep(1/n, nrow(data))
)
```

Arguments

data	Dataset containing the titles and abstracts wanted to be screened.
n	A non-negative integer giving the number of rows to choose.
with_replacement	Logical indicating if sampling should be done with of without replacement. Default is FALSE.
prob_vec	'A vector of probability weights for obtaining the elements of the vector being sampled.' Default is a vector of 1/n.

Value

A dataset with n rows.

References

Vembye, M. H., Christensen, J., Mølgaard, A. B., & Schytt, F. L. W. (2024) *GPT API Models Can Function as Highly Reliable Second Screeners of Titles and Abstracts in Systematic Reviews: A Proof of Concept and Common Guidelines* https://osf.io/preprints/osf/yrhzm

Examples

```
excl_test_dat <- filges2015_dat[1:200,] |> sample_references(100)
```

| screen_analyzer | *Analyze performance between the human and AI screening.* |

Description

[Experimental]

When both the human and AI title and abstract screening has been done, this function allows you to calculate performance measures of the screening, including the overall accuracy, specificity and sensitivity of the screening, as well as inter-rater reliability kappa statistics.

Usage

```
screen_analyzer(x, human_decision = human_code, key_result = TRUE)
```

Arguments

x	An object of either class `'gpt'` or `'chatgpt'` or a dataset of either class `'gpt_tbl'`, `'chatgpt_tbl'`, or `'gpt_agg_tbl'`
human_decision	Indicate the variable in the data that contains the human_decision. This variable must be numeric, containing 1 (for included references) and 0 (for excluded references) only.
key_result	Logical indicating if only the raw agreement, recall, and specificity measures should be returned. Default is `TRUE`.

Value

A `tibble` with screening performance measures. The `tibble` includes the following variables:

promptid	integer	indicating the prompt ID.
model	character	indicating the specific gpt-model used.
reps	integer	indicating the number of times the same question was sent to GPT server.
top_p	numeric	indicating the applied top_p.
n_screened	integer	indicating the number of screened references.
n_missing	numeric	indicating the number of missing responses.
n_refs	integer	indicating the total number of references expected to be screened for the given condition
human_in_gpt_ex	numeric	indicating the number of references included by humans and excluded by gpt.
human_ex_gpt_in	numeric	indicating the number of references excluded by humans and included by gpt.
human_in_gpt_in	numeric	indicating the number of references included by humans and included by gpt.
human_ex_gpt_ex	numeric	indicating the number of references excluded by humans and excluded by gpt.
accuracy	numeric	indicating the overall percent disagreement between human and gpt (Gartlehner et al., 20
p_agreement	numeric	indicating the overall percent agreement between human and gpt.
precision	numeric	"measures the ability to include only articles that should be included" (Syriani et al., 202
recall	numeric	"measures the ability to include all articles that should be included" (Syriani et al., 2023)
npv	numeric	Negative predictive value (NPV) "measures the ability to exclude only articles that shoul
specificity	numeric	"measures the ability to exclude all articles that should be excluded" (Syriani et al., 2023
bacc	numeric	"capture the accuracy of deciding both inclusion and exclusion classes" (Syriani et al., 20

F2	numeric	F-measure that "consider the cost of getting false negatives twice as costly as getting fals
mcc	numeric	indicating percent agreement for excluded references (Gartlehner et al., 2019).
irr	numeric	indicating the inter-rater reliability as described in McHugh (2012).
se_irr	numeric	indicating standard error for the inter-rater reliability.
cl_irr	numeric	indicating lower confidence interval for the inter-rater reliability.
cu_irr	numeric	indicating upper confidence interval for the inter-rater reliability.
level_of_agreement	character	interpretation of the inter-rater reliability as suggested by McHugh (2012).

References

Gartlehner, G., Wagner, G., Lux, L., Affengruber, L., Dobrescu, A., Kaminski-Hartenthaler, A., & Viswanathan, M. (2019). Assessing the accuracy of machine-assisted abstract screening with DistillerAI: a user study. *Systematic Reviews*, 8(1), 277. doi:10.1186/s1364301912213

McHugh, M. L. (2012). Interrater reliability: The kappa statistic. *Biochemia Medica*, 22(3), 276-282. https://pubmed.ncbi.nlm.nih.gov/23092060/

Syriani, E., David, I., & Kumar, G. (2023). Assessing the Ability of ChatGPT to Screen Articles for Systematic Reviews. *ArXiv Preprint ArXiv:2307.06464*.

Examples

```
## Not run:

library(future)

set_api_key()

prompt <- "Is this study about a Functional Family Therapy (FFT) intervention?"

plan(multisession)

res <- tabscreen_gpt(
  data = filges2015_dat[1:2,],
  prompt = prompt,
  studyid = studyid,
  title = title,
  abstract = abstract
  )

plan(sequential)

res |> screen_analyzer()


## End(Not run)
```

screen_errors	*Generic function to re-screen failed title and abstract requests.*

Description

[Experimental]

This is a generic function to re-screen of failed title and abstract requests.

Usage

```
screen_errors(
  object,
  api_key = get_api_key(),
  max_tries = 4,
  max_seconds,
  is_transient,
  backoff,
  after,
  ...
)
```

Arguments

object	An object of either class `'gpt'` or `'chatgpt'`.
api_key	Numerical value with your personal API key. Default setting draws on the `get_api_key()` to retrieve the API key from the R environment, so that the key is not compromised. The API key can be added to the R environment via `set_api_key()` or by using `usethis::edit_r_environ()`. In the .Renviron file, write CHATGPT_KEY=INSERT_YOUR_KEY_HERE. After entering the API key, close and save the .Renviron file and restart RStudio (ctrl + shift + F10). Alternatively, one can use `httr2::secret_make_key()`, `httr2::secret_encrypt()`, and `httr2::secret_decrypt()` to scramble and decrypt the API key.
max_tries, max_seconds	
	'Cap the maximum number of attempts with max_tries or the total elapsed time from the first request with max_seconds. If neither option is supplied (the default), `httr2::req_perform()` will not retry' (Wickham, 2023). Default max_tries is 16. If missing, the value of max_seconds from the original screening conducted with `tabscreen_gpt()` will be used.
is_transient	'A predicate function that takes a single argument (the response) and returns TRUE or FALSE specifying whether or not the response represents a transient error' (Wickham, 2023). If missing, the is_transient function from the original screening conducted with `tabscreen_gpt()` will be used.
backoff	'A function that takes a single argument (the number of failed attempts so far) and returns the number of seconds to wait' (Wickham, 2023). If missing, the backoffvalue from the original screening conducted with `tabscreen_gpt()` will be used.

after
’A function that takes a single argument (the response) and returns either a number of seconds to wait or `NULL`, which indicates that a precise wait time is not available that the `backoff` strategy should be used instead’ (Wickham, 2023). If missing, the `after` value from the original screening conducted with `tabscreen_gpt()` will be used.

...
Further argument to pass to the request body. See `https://platform.openai.com/docs/api-reference/chat/create`. If used in the original screening in `tabscreen_gpt()`, the argument(s) must be specified here again.

Value

An object of class `'gpt'` or `'chatgpt'` similar to the object returned by `tabscreen_gpt()`. See documentation for `tabscreen_gpt()`.

See Also

`screen_errors.gpt()`, `screen_errors.chatgpt()`

Examples

```
## Not run:

set_api_key()
prompt <- "Is this study about a Functional Family Therapy (FFT) intervention?"

obj_with_error <-
  tabscreen_gpt(
    data = filges2015_dat[1:2,],
    prompt = prompt,
    studyid = studyid,
    title = title,
    abstract = abstract,
    model = "gpt-4o-mini"
    )

obj_rescreened <-
  obj_with_error |>
  screen_error()


## End(Not run)
```

---

screen_errors.chatgpt   *Re-screen failed requests.*

Description

[Deprecated]

This function supports re-screening of all failed title and abstract requests screened with `tabscreen_gpt.original()`. This function has been deprecated because OpenAI has deprecated the function_call and and functions argument that was used in `tabscreen_gpt.original()`.

Usage

```
## S3 method for class 'chatgpt'
screen_errors(
  object,
  ...,
  api_key = get_api_key(),
  max_tries = 4,
  max_seconds,
  is_transient,
  backoff,
  after
)
```

Arguments

object	An object of class `'chatgpt'`.
...	Further argument to pass to the request body. See `https://platform.openai.com/docs/api-reference/chat/create`. If used in the original screening (e.g., with `tabscreen_gpt.original()`), the argument(s) must be specified again here.
api_key	Numerical value with your personal API key.
max_tries, max_seconds	
	'Cap the maximum number of attempts with max_tries or the total elapsed time from the first request with max_seconds. If neither option is supplied (the default), `httr2::req_perform()` will not retry' (Wickham, 2023). Default max_tries is 4. If missing, the value of max_seconds from the original screening (e.g., conducted with `tabscreen_gpt.original()`) will be used.
is_transient	'A predicate function that takes a single argument (the response) and returns TRUE or FALSE specifying whether or not the response represents a transient error' (Wickham, 2023). If missing, the is_transient function from the original screening (e.g., conducted with `tabscreen_gpt.original()`) will be used.
backoff	'A function that takes a single argument (the number of failed attempts so far) and returns the number of seconds to wait' (Wickham, 2023). If missing, the backoff value from the original screening (e.g., conducted with `tabscreen_gpt.original()`) will be used.
after	'A function that takes a single argument (the response) and returns either a number of seconds to wait or NULL, which indicates that a precise wait time is not available that the backoff strategy should be used instead' (Wickham, 2023). If missing, the after value from the original screening (e.g., conducted with `tabscreen_gpt.original()`) will be used.

Value

Object of class `'chatgpt'` similar to the object returned by [`tabscreen_gpt.original()`](#). See
documentation value for [`tabscreen_gpt.original()`](#).

References

Wickham H (2023). *httr2: Perform HTTP Requests and Process the Responses*. https://httr2.r-lib.org, https://github.com/r-lib/httr2.

See Also

[`tabscreen_gpt.original()`](#)

Examples

```
## Not run:

set_api_key()

prompt <- "Is this study about a Functional Family Therapy (FFT) intervention?"

obj_with_error <-
  tabscreen_gpt(
    data = filges2015_dat[1:2,],
    prompt = prompt,
    studyid = studyid,
    title = title,
    abstract = abstract,
    model = c("gpt-3.5-turbo-0613", "gpt-3.5-turbo-16k-0613"),
    max_tries = 1,
    reps = 10
    )

obj_rescreened <-
  obj_with_error |>
  screen_error()

# Alternatively re-set max_tries if errors still appear
obj_rescreened <-
  obj_with_error |>
  screen_error(max_tries = 16)

## End(Not run)
```

---

screen_errors.gpt   *Re-screen failed requests.*

Description

[Experimental]

This function supports re-screening of all failed title and abstract requests screened with [tabscreen_gpt()](#)/[tabscreen_gpt.](#)

Usage

```
## S3 method for class 'gpt'
screen_errors(
  object,
  api_key = get_api_key(),
  max_tries = 16,
  max_seconds,
  is_transient,
  backoff,
  after,
  ...
)
```

Arguments

object   An object of class 'gpt'.

api_key   Numerical value with your personal API key. Default setting draws on the
                [get_api_key()](#) to retrieve the API key from the R environment, so that the
                key is not compromised. The API key can be added to the R environment via
                [set_api_key()](#) or by using [usethis::edit_r_environ()](#). In the .Renviron
                file, write CHATGPT_KEY=INSERT_YOUR_KEY_HERE. After entering the API key,
                close and save the .Renviron file and restart RStudio (ctrl + shift + F10). Alter-
                natively, one can use [httr2::secret_make_key()](#), [httr2::secret_encrypt()](#),
                and [httr2::secret_decrypt()](#) to scramble and decrypt the API key.

max_tries, max_seconds
                'Cap the maximum number of attempts with max_tries or the total elapsed
                time from the first request with max_seconds. If neither option is supplied
                (the default), [httr2::req_perform()](#) will not retry' (Wickham, 2023). De-
                fault max_tries is 16. If missing, the value of max_seconds from the original
                screening conducted with [tabscreen_gpt()](#) will be used.

is_transient   'A predicate function that takes a single argument (the response) and returns
                TRUE or FALSE specifying whether or not the response represents a transient er-
                ror' (Wickham, 2023). If missing, the is_transient function from the original
                screening conducted with [tabscreen_gpt()](#) will be used.

backoff   'A function that takes a single argument (the number of failed attempts so far)
                and returns the number of seconds to wait' (Wickham, 2023). If missing, the
                backoffvalue from the original screening conducted with [tabscreen_gpt()](#)
                will be used.

after   'A function that takes a single argument (the response) and returns either a
                number of seconds to wait or NULL, which indicates that a precise wait time
                is not available that the backoff strategy should be used instead' (Wickham,
```

2023). If missing, the `after` value from the original screening conducted with `tabscreen_gpt()` will be used.

...                  Further argument to pass to the request body. See `https://platform.openai.com/docs/api-reference/chat/create`. If used in the original screening in `tabscreen_gpt()`, the argument(s) must be specified again here.

## Value

An object of class `'gpt'` similar to the object returned by `tabscreen_gpt()`. See documentation for `tabscreen_gpt()`.

## References

Wickham H (2023). *httr2: Perform HTTP Requests and Process the Responses*. https://httr2.r-lib.org, https://github.com/r-lib/httr2.

## See Also

`tabscreen_gpt()`, `tabscreen_gpt.tools()`

## Examples

```
## Not run:
prompt <- "Is this study about a Functional Family Therapy (FFT) intervention?"

obj_with_error <-
  tabscreen_gpt(
    data = filges2015_dat[1:10,],
    prompt = prompt,
    studyid = studyid,
    title = title,
    abstract = abstract,
    model = "gpt-4o"
    )

obj_rescreened <-
  obj_with_error |>
  screen_error()


## End(Not run)
```

---

set_api_key   *Creating a temporary R environment API key variable*

Description

This function automatically sets/creates an interim R environment variable with the API key to call a given AI model (e.g. ChatGPT). Thereby users avoid exposing their API keys. If the API key is set in the console, it will/can be revealed via the .Rhistory. Find more information about this issue at https://httr2.r-lib.org/articles/wrapping-apis.html.

Usage

```
set_api_key(key, env_var = "CHATGPT_KEY")
```

Arguments

key   Character string with an (ideally encrypt) API key. See how to encrypt key here: https://httr2.r-lib.org/articles/wrapping-apis.html#basics. If not provided, it returns a password box in which the true API key can be secretly entered.

env_var   Character string indicating the name of the temporary R environment variable with the API key and the used AI model. Currently, the argument only takes env_var = "CHATGPT_KEY".

Details

When set_api_key() has successfully been executed, get_api_key() automatically retrieves the API key from the R environment and the users do not need to specify the API when running functions from the package that call the API. The API key can be permanently set by using usethis::edit_r_environ(). Then write CHATGPT_KEY=[insert your api key here] and close the .Renviron window and restart R.

Value

A temporary environment variable with the name from env_var. If key is missing, it returns a password box in which the true API key can be entered.

Note

Find your personal API key at https://platform.openai.com/account/api-keys.

See Also

get_api_key

Examples

```
## Not run:
set_api_key()

## End(Not run)
```

tabscreen_gpt.original

Title and abstract screening with GPT API models using function calls via the original function call arguments

Description

[Deprecated]

This function has been deprecated (but can still be used) because OpenAI has deprecated the function_call and and functions argument which is used in this function. Instead use the `tabscreen_gpt.tools()` that handles the function calling via the tools and tool_choice arguments.

This function supports the conduct of title and abstract screening with GPT API models in R. This function only works with GPT-4, more specifically gpt-4-0613. To draw on other models, use `tabscreen_gpt.tools()`. The function allows to run title and abstract screening across multiple prompts and with repeated questions to check for consistency across answers. This function draws on the newly developed function calling to better steer the output of the responses. This function was used in Vembye et al. (2024).

Usage

```
tabscreen_gpt.original(
  data,
  prompt,
  studyid,
  title,
  abstract,
  ...,
  model = "gpt-4",
  role = "user",
  functions = incl_function_simple,
  function_call_name = list(name = "inclusion_decision_simple"),
  top_p = 1,
  time_info = TRUE,
  token_info = TRUE,
  api_key = get_api_key(),
  max_tries = 16,
  max_seconds = NULL,
  is_transient = gpt_is_transient,
  backoff = NULL,
  after = NULL,
  rpm = 10000,
  reps = 1,
  seed_par = NULL,
  progress = TRUE,
  messages = TRUE,
  incl_cutoff_upper = 0.5,
```

```
    incl_cutoff_lower = incl_cutoff_upper - 0.1,
    force = FALSE
)
```

Arguments

`data`	Dataset containing the titles and abstracts.
`prompt`	Prompt(s) to be added before the title and abstract.
`studyid`	Unique Study ID. If missing, this is generated automatically.
`title`	Name of the variable containing the title information.
`abstract`	Name of variable containing the abstract information.
`...`	Further argument to pass to the request body. See https://platform.openai.com/docs/api-reference/chat/create.
`model`	Character string with the name of the completion model. Can take multiple models, including gpt-4 models. Default = "gpt-4" (i.e., gpt-4-0613). This model has been shown to outperform the gpt-3.5-turbo models in terms of its ability to detect relevant studies (Vembye et al., Under preparation). Find available model at https://platform.openai.com/docs/models/model-endpoint-compatibility.
`role`	Character string indicate the role of the user. Default is "user".
`functions`	Function to steer output. Default is incl_function_simple. To get detailed responses use the hidden function call incl_function from the package. Also see 'Examples below. Find further documentation for function calling at https://openai.com/blog/function-calling-and-other-api-updates.
`function_call_name`	Functions to call. Default is list(name = "inclusion_decision_simple"). To get detailed responses use list(name = "inclusion_decision"). Also see 'Examples below.
`top_p`	'An alternative to sampling with temperature, called nucleus sampling, where the model considers the results of the tokens with top_p probability mass. So 0.1 means only the tokens comprising the top 10% probability mass are considered. We generally recommend altering this or temperature but not both.' (OPEN-AI). Default is 1. Find documentation at https://platform.openai.com/docs/api-reference/chat/create#chat/create-top_p.
`time_info`	Logical indicating whether the run time of each request/question should be included in the data. Default = TRUE.
`token_info`	Logical indicating whether the number of prompt and completion tokens per request should be included in the output data. Default = TRUE. When TRUE, the output object will include price information of the conducted screening.
`api_key`	Numerical value with your personal API key. Find at https://platform.openai.com/account/api-keys. Use [httr2::secret_make_key()](), [httr2::secret_encrypt()](), and [httr2::secret_decrypt()]() to scramble and decrypt the api key and use [set_api_key()]() to securely automate the use of the api key by setting the api key as a locale environment variable.

max_tries, max_seconds

> 'Cap the maximum number of attempts with max_tries or the total elapsed time from the first request with max_seconds. If neither option is supplied (the default), httr2::req_perform() will not retry' (Wickham, 2023).

is_transient   'A predicate function that takes a single argument (the response) and returns TRUE or FALSE specifying whether or not the response represents a transient error' (Wickham, 2023).

backoff   'A function that takes a single argument (the number of failed attempts so far) and returns the number of seconds to wait' (Wickham, 2023).

after   'A function that takes a single argument (the response) and returns either a number of seconds to wait or NULL, which indicates that a precise wait time is not available that the backoff strategy should be used instead' (Wickham, 2023).

rpm   Numerical value indicating the number of requests per minute (rpm) available for the specified api key. Find more information at https://platform.openai.com/docs/guides/rate-limits/what-are-the-rate-limits-for-our-api. Alternatively, use rate_limits_per_minute().

reps   Numerical value indicating the number of times the same question should be sent to OpenAI's GPT API models. This can be useful to test consistency between answers. Default is 1 but when using 3.5 models, we recommend setting this value to 10.

seed_par   Numerical value for a seed to ensure that proper, parallel-safe random numbers are produced.

progress   Logical indicating whether a progress line should be shown when running the title and abstract screening in parallel. Default is TRUE.

messages   Logical indicating whether to print messages embedded in the function. Default is TRUE.

incl_cutoff_upper

> Numerical value indicating the probability threshold for which a studies should be included. Default is 0.5, which indicates that titles and abstracts that OpenAI's GPT API model has included more than 50 percent of the times should be included.

incl_cutoff_lower

> Numerical value indicating the probability threshold above which studies should be check by a human. Default is 0.4, which means that if you ask OpenAI's GPT API model the same questions 10 times and it includes the title and abstract 4 times, we suggest that the study should be check by a human.

force   Logical argument indicating whether to force the function to use more than 10 iterations for gpt-3.5 models and more than 1 iteration for gpt-4 models. This argument is developed to avoid the conduct of wrong and extreme sized screening. Default is FALSE.

Value

An object of class "chatgpt". The object is a list containing the following components:

answer_data_sum

> dataset with the summarized, probabilistic inclusion decision for each title and abstract across multiple repeated questions.

answer_data_all

        dataset with all individual answers.

price   numerical value indicating the total price (in USD) of the screening.

price_data   dataset with prices across all gpt models used for screening.

Note

The answer_data_sum data contains the following mandatory variables:

studyid	integer	indicating the study ID of the reference.
title	character	indicating the title of the reference.
abstract	character	indicating the abstract of the reference.
promptid	integer	indicating the prompt ID.
prompt	character	indicating the prompt.
model	character	indicating the specific gpt-model used.
question	character	indicating the final question sent to OpenAI's GPT API models.
top_p	numeric	indicating the applied top_p.
incl_p	numeric	indicating the probability of inclusion calculated across multiple repeated responses
final_decision_gpt	character	indicating the final decision reached by gpt - either 'Include', 'Exclude', or 'Check'
final_decision_gpt_num	integer	indicating the final numeric decision reached by gpt - either 1 or 0.
longest_answer	character	indicating the longest gpt response obtained across multiple repeated responses on t
reps	integer	indicating the number of times the same question has been sent to OpenAI's GPT A
n_mis_answers	integer	indicating the number of missing responses.

The answer_data_all data contains the following mandatory variables:

studyid	integer	indicating the study ID of the reference.
title	character	indicating the title of the reference.
abstract	character	indicating the abstract of the reference.
promptid	integer	indicating the prompt ID.
prompt	character	indicating the prompt.
model	character	indicating the specific gpt-model used.
iterations	numeric	indicating the number of times the same question has been sent to OpenAI's GPT API
question	character	indicating the final question sent to OpenAI's GPT API models.
top_p	numeric	indicating the applied top_p.
decision_gpt	character	indicating the raw gpt decision - either "1", "0", "1.1" for inclusion, exclusion, or un
detailed_description	character	indicating detailed description of the given decision made by OpenAI's GPT API mode
decision_binary	integer	indicating the binary gpt decision, that is 1 for inclusion and 0 for exclusion. 1.1 decisi
prompt_tokens	integer	indicating the number of prompt tokens sent to the server for the given request.
completion_tokens	integer	indicating the number of completion tokens sent to the server for the given request.
run_time	numeric	indicating the time it took to obtain a response from the server for the given request.
n	integer	indicating request ID.

If any requests failed to reach the server, the chatgpt object contains an error data set (error_data)
having the same variables as answer_data_all but with failed request references only.

The price_data data contains the following variables:

model	character	gpt model.
input_price_dollar	integer	price for all prompt/input tokens for the correspondent gpt-model.
output_price_dollar	integer	price for all completion/output tokens for the correspondent gpt-model.
price_total_dollar	integer	total price for all tokens for the correspondent gpt-model.

Find current token pricing at <https://openai.com/pricing>.

References

Vembye, M. H., Christensen, J., Mølgaard, A. B., & Schytt, F. L. W. (2024) *GPT API Models Can Function as Highly Reliable Second Screeners of Titles and Abstracts in Systematic Reviews: A Proof of Concept and Common Guidelines* <https://osf.io/preprints/osf/yrhzm>

Wickham H (2023). *httr2: Perform HTTP Requests and Process the Responses.* https://httr2.r-lib.org, https://github.com/r-lib/httr2.

Examples

```
## Not run:

set_api_key()

prompt <- "Is this study about a Functional Family Therapy (FFT) intervention?"

tabscreen_gpt.original(
  data = filges2015_dat[1:2,],
  prompt = prompt,
  studyid = studyid,
  title = title,
  abstract = abstract,
  max_tries = 2
  )

 # Get detailed descriptions of the gpt decisions by using the
 # embedded function calling functions from the package. See example below.
 tabscreen_gpt.original(
   data = filges2015_dat[1:2,],
   prompt = prompt,
   studyid = studyid,
   title = title,
   abstract = abstract,
   functions = incl_function,
   function_call_name = list(name = "inclusion_decision"),
   max_tries = 2
 )

## End(Not run)
```

tabscreen_gpt.tools *Title and abstract screening with GPT API models using function calls via the tools argument*

Description

[Stable]

This function supports the conduct of title and abstract screening with GPT API models in R. Specifically, it allows the user to draw on GPT-3.5, GPT-4, GPT-4o, GPT-4o-mini, and fine-tuned models. The function allows to run title and abstract screening across multiple prompts and with repeated questions to check for consistency across answers. All of which can be done in parallel. The function draws on the newly developed function calling which is called via the tools argument in the request body. This is the main different between tabscreen_gpt.tools() and tabscreen_gpt.original(). Function calls ensure more reliable and consistent responses to ones requests. See Vembye et al. (2024) for guidance on how adequately to conduct title and abstract screening with GPT models.

Usage

```
tabscreen_gpt.tools(data, prompt, studyid, title, abstract,
  model = "gpt-4o-mini", role = "user", tools = NULL, tool_choice = NULL, top_p = 1,
   time_info = TRUE, token_info = TRUE, api_key = get_api_key(), max_tries = 16,
   max_seconds = NULL, is_transient = gpt_is_transient, backoff = NULL,
   after = NULL, rpm = 10000, reps = 1, seed_par = NULL, progress = TRUE,
   decision_description = FALSE, messages = TRUE, incl_cutoff_upper = NULL,
   incl_cutoff_lower = NULL, force = FALSE, fine_tuned = FALSE, ...)

tabscreen_gpt(data, prompt, studyid, title, abstract,
  model = "gpt-4o-mini", role = "user", tools = NULL, tool_choice = NULL, top_p = 1,
   time_info = TRUE, token_info = TRUE, api_key = get_api_key(), max_tries = 16,
   max_seconds = NULL, is_transient = gpt_is_transient, backoff = NULL,
   after = NULL, rpm = 10000, reps = 1, seed_par = NULL, progress = TRUE,
   decision_description = FALSE, messages = TRUE, incl_cutoff_upper = NULL,
   incl_cutoff_lower = NULL, force = FALSE, fine_tuned = FALSE, ...)
```

Arguments

data	Dataset containing the titles and abstracts.
prompt	Prompt(s) to be added before the title and abstract.
studyid	Unique Study ID. If missing, this is generated automatically.
title	Name of the variable containing the title information.
abstract	Name of variable containing the abstract information.
model	Character string with the name of the completion model. Can take multiple models. Default is the latest "gpt-4o-mini". Find available model at https://platform.openai.com/docs/models/model-endpoint-compatibility.

role	Character string indicating the role of the user. Default is "user".
tools	This argument allows this user to apply customized functions. See https://platform.openai.com/docs/api-reference/chat/create#chat-create-tools. Default is NULL. If not specified the default function calls from AIscreenR are used.
tool_choice	If a customized function is provided this argument 'controls which (if any) tool is called by the model' (OpenAI). Default is NULL. If set to NULL when using a customized function, the default is "auto". See https://platform.openai.com/docs/api-reference/chat/create#chat-create-tool_choice.
top_p	'An alternative to sampling with temperature, called nucleus sampling, where the model considers the results of the tokens with top_p probability mass. So 0.1 means only the tokens comprising the top 10% probability mass are considered. We generally recommend altering this or temperature but not both.' (OpenAI). Default is 1. Find documentation at https://platform.openai.com/docs/api-reference/chat/create#chat/create-top_p.
time_info	Logical indicating whether the run time of each request/question should be included in the data. Default is TRUE.
token_info	Logical indicating whether token information should be included in the output data. Default is TRUE. When TRUE, the output object will include price information of the conducted screening.
api_key	Numerical value with your personal API key. Default setting draws on the get_api_key() to retrieve the API key from the R environment, so that the key is not compromised. The API key can be added to the R environment via set_api_key() or by using usethis::edit_r_environ(). In the .Renviron file, write CHATGPT_KEY=INSERT_YOUR_KEY_HERE. After entering the API key, close and save the .Renviron file and restart RStudio (ctrl + shift + F10). Alternatively, one can use httr2::secret_make_key(), httr2::secret_encrypt(), and httr2::secret_decrypt() to scramble and decrypt the API key.
max_tries, max_seconds	
	'Cap the maximum number of attempts with max_tries or the total elapsed time from the first request with max_seconds. If neither option is supplied (the default), httr2::req_perform() will not retry' (Wickham, 2023). The default of max_tries is 16.
is_transient	'A predicate function that takes a single argument (the response) and returns TRUE or FALSE specifying whether or not the response represents a transient error' (Wickham, 2023). This function runs automatically in the AIscreenR but can be customized by the user if necessary.
backoff	'A function that takes a single argument (the number of failed attempts so far) and returns the number of seconds to wait' (Wickham, 2023).
after	'A function that takes a single argument (the response) and returns either a number of seconds to wait or NULL, which indicates that a precise wait time is not available that the backoff strategy should be used instead' (Wickham, 2023).
rpm	Numerical value indicating the number of requests per minute (rpm) available for the specified model. Find more information at https://platform.openai.com/docs/guides/rate-limits/what-are-the-rate-limits-for-our-api. Alternatively, use rate_limits_per_minute().

reps	Numerical value indicating the number of times the same question should be send to the server. This can be useful to test consistency between answers, and/or can be used to make inclusion judgments based on how many times a study has been included across a the given number of screenings. Default is 1 but when using gpt-3.5-turbo models or gpt-4o-mini, we recommend setting this value to `10` to catch model uncertainty.
seed_par	Numerical value for a seed to ensure that proper, parallel-safe random numbers are produced.
progress	Logical indicating whether a progress line should be shown when running the title and abstract screening in parallel. Default is `TRUE`.
decision_description	
	Logical indicating whether a detailed description should follow the decision made by GPT. Default is `FALSE`. When conducting large-scale screening, we generally recommend not using this feature as it will substantially increase the cost of the screening. We generally recommend using it when encountering disagreements between GPT and human decisions.
messages	Logical indicating whether to print messages embedded in the function. Default is `TRUE`.
incl_cutoff_upper	
	Numerical value indicating the probability threshold for which a studies should be included. ONLY relevant when the same questions is requested multiple times (i.e., when any reps > 1). Default is 0.5, indicating that titles and abstracts should only be included if GPT has included the study more than 50 percent of the times.
incl_cutoff_lower	
	Numerical value indicating the probability threshold above which studies should be check by a human. ONLY relevant when the same questions is requested multiple times (i.e., when any reps > 1). Default is 0.4, meaning that if you ask GPT the same questions 10 times and it includes the title and abstract 4 times, we suggest that the study should be check by a human.
force	Logical argument indicating whether to force the function to use more than 10 iterations for gpt-3.5 models and more than 1 iteration for gpt-4 models other than gpt-4o-mini. This argument is developed to avoid the conduct of wrong and extreme sized screening. Default is `FALSE`.
fine_tuned	Logical indicating whether a fine-tuned model is used. Default is `FALSE`.
...	Further argument to pass to the request body. See https://platform.openai.com/docs/api-reference/chat/create.

Value

An object of class `'gpt'`. The object is a list containing the following datasets and components:

answer_data	dataset of class `'gpt_tbl'` with all individual answers.
price_dollar	numerical value indicating the total price (in USD) of the screening.
price_data	dataset with prices across all gpt models used for screening.
run_date	string indicating the date when the screening was ran. In some frameworks, time details are considered important to report (see e.g., Thomas et al., 2024).

... some additional attributed values/components, including an attributed list with the arguments used in the function. These are used in [screen_errors()](#) to re-screen transient errors.

If the same question is requested multiple times, the object will also contain the following dataset with results aggregated across the iterated requests/questions.

answer_data_aggregated

dataset of class `'gpt_agg_tbl'` with the summarized, probabilistic inclusion decision for each title and abstract across multiple repeated questions.

Note

The answer_data data contains the following *mandatory* variables:

studyid	integer	indicating the study ID of the reference.
title	character	indicating the title of the reference.
abstract	character	indicating the abstract of the reference.
promptid	integer	indicating the prompt ID.
prompt	character	indicating the prompt.
model	character	indicating the specific gpt-model used.
iterations	numeric	indicating the number of times the same question has been sent to OpenAI's GPT API
question	character	indicating the final question sent to OpenAI's GPT API models.
top_p	numeric	indicating the applied top_p.
decision_gpt	character	indicating the raw gpt decision - either "1", "0", "1.1" for inclusion, exclusion, or un
detailed_description	character	indicating detailed description of the given decision made by OpenAI's GPT API mode
decision_binary	integer	indicating the binary gpt decision, that is 1 for inclusion and 0 for exclusion. 1.1 decisi
prompt_tokens	integer	indicating the number of prompt tokens sent to the server for the given request.
completion_tokens	integer	indicating the number of completion tokens sent to the server for the given request.
submodel	character	indicating the exact (sub)model used for screening.
run_time	numeric	indicating the time it took to obtain a response from the server for the given request.
run_date	character	indicating the date the given response was received.
n	integer	indicating iteration ID. Is only different from 1, when reps > 1.

If any requests failed, the gpt object contains an error dataset (error_data) containing the same variables as answer_data but with failed request references only.

When the same question is requested multiple times, the answer_data_aggregated data contains the following *mandatory* variables:

studyid	integer	indicating the study ID of the reference.
title	character	indicating the title of the reference.
abstract	character	indicating the abstract of the reference.
promptid	integer	indicating the prompt ID.
prompt	character	indicating the prompt.
model	character	indicating the specific gpt-model used.
question	character	indicating the final question sent to OpenAI's GPT API models.
top_p	numeric	indicating the applied top_p.
incl_p	numeric	indicating the probability of inclusion calculated across multiple repeated responses

final_decision_gpt	character	indicating the final decision reached by gpt - either 'Include', 'Exclude', or 'Check'
final_decision_gpt_num	integer	indicating the final numeric decision reached by gpt - either 1 or 0.
longest_answer	character	indicating the longest gpt response obtained across multiple repeated responses on t
reps	integer	indicating the number of times the same question has been sent to OpenAI's GPT A
n_mis_answers	integer	indicating the number of missing responses.
submodel	character	indicating the exact (sub)model used for screening.

The `price_data` data contains the following variables:

prompt	character	if multiple prompts are used this variable indicates the given prompt-id.
model	character	the specific gpt model used.
iterations	integer	indicating the number of times the same question was requested.
input_price_dollar	integer	price for all prompt/input tokens for the correspondent gpt-model.
output_price_dollar	integer	price for all completion/output tokens for the correspondent gpt-model.
total_price_dollar	integer	total price for all tokens for the correspondent gpt-model.

Find current token pricing at `https://openai.com/pricing` or model_prizes.

References

Vembye, M. H., Christensen, J., Mølgaard, A. B., & Schytt, F. L. W. (2024) *GPT API Models Can Function as Highly Reliable Second Screeners of Titles and Abstracts in Systematic Reviews: A Proof of Concept and Common Guidelines* `https://osf.io/preprints/osf/yrhzm`

Thomas, J. et al. (2024). Responsible AI in Evidence SynthEsis (RAISE): guidance and recommendations. `https://osf.io/cn7x4`

Wickham H (2023). *httr2: Perform HTTP Requests and Process the Responses*. `https://httr2.r-lib.org`, `https://github.com/r-lib/httr2`.

Examples

```
## Not run:

library(future)

set_api_key()

prompt <- "Is this study about a Functional Family Therapy (FFT) intervention?"

plan(multisession)

tabscreen_gpt(
  data = filges2015_dat[1:2,],
  prompt = prompt,
  studyid = studyid,
  title = title,
  abstract = abstract
  )
```

```
plan(sequential)

 # Get detailed descriptions of the gpt decisions.

 plan(multisession)

 tabscreen_gpt(
   data = filges2015_dat[1:2,],
   prompt = prompt,
   studyid = studyid,
   title = title,
   abstract = abstract,
   decision_description = TRUE
 )

plan(sequential)


## End(Not run)
```

Index