Package 'peRspective'

May 20, 2019

Title Interface to the 'Perspective' API

Version 0.1.0

Description Interface to the 'Perspective' API, which can be found at the following URL: <https://github.com/conversationai/perspectiveapi#perspective-comment-analyzer-api>.
The 'Perspective' API uses machine learning models to score the perceived impact a comment might have on a conversation (i.e. TOXICITY, INFLAMMATORY, etc.).
'peRspective' provides access to the API and returns tidy data frames with results of the specified machine learning model(s).

URL https://github.com/favstats/peRspective

BugReports https://github.com/favstats/peRspective

Depends R (>= 3.5.0)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports crayon, dplyr, glue, httr, jsonlite, magrittr, purrr, rlang, rlist, stringr, tibble

Suggests testthat (>= 2.1.0), covr, badger

NeedsCompilation no

Author Fabio Votta [aut, cre]

Maintainer Fabio Votta <fabio.votta@gmail.com>

Repository CRAN

Date/Publication 2019-05-20 08:40:02 UTC

R topics documented:

`perspective-package` *peRspective: Interface to the Perspective API*

Description

Provides access to the Perspective API (`http://www.perspectiveapi.com/`). Perspective is an API that uses machine learning models to score the perceived impact a comment might have on a conversation. `peRspective` provides access to the API using the R programming language. For an excellent documentation of the Perspective API see here.

Get API Key

1. Create a Google Cloud project in your Google Cloud console
2. Go to Perspective API's overview page and click ***Enable***
3. Go to the API credentials page, just click ***Create credentials***, and choose "API Key".

Suggested Usage of API Key

peRspective functions will read the API key from environment variable `perspective_api_key`. You can specify it like this at the start of your script:

```
Sys.setenv(perspective_api_key = "**********")
```

To start R session with the initialized environment variable create an `.Renviron` file in your R home with a line like this:

```
perspective_api_key = "**********"
```

To check where your R home is, try `normalizePath("~")`.

Quota and character length Limits

You can check your quota limits by going to your google cloud project's Perspective API page, and check your projects quota usage at the cloud console quota usage page.

The maximum text size per request is 3000 bytes.

Alpha models

The following alpha models are **recommended** for use. They have been tested across multiple domains and trained on hundreds of thousands of comments tagged by thousands of human moderators. These are available in **English (en) and Spanish (es)**.

- **TOXICITY**: rude, disrespectful, or unreasonable comment that is likely to make people leave a discussion. This model is a Convolutional Neural Network (CNN) trained with word-vector inputs.
- **SEVERE_TOXICITY**: This model uses the same deep-CNN algorithm as the TOXICITY model, but is trained to recognize examples that were considered to be 'very toxic' by crowd-workers. This makes it much less sensitive to comments that include positive uses of curse-words for example. A labelled dataset and details of the methodology can be found in the same toxicity dataset that is available for the toxicity model.

Experimental models

The following experimental models give more fine-grained classifications than overall toxicity. They were trained on a relatively smaller amount of data compared to the primary toxicity models above and have not been tested as thoroughly.

- **IDENTITY_ATTACK**: negative or hateful comments targeting someone because of their identity.
- **INSULT**: insulting, inflammatory, or negative comment towards a person or a group of people.
- **PROFANITY**: swear words, curse words, or other obscene or profane language.
- **THREAT**: describes an intention to inflict pain, injury, or violence against an individual or group.
- **SEXUALLY_EXPLICIT**: contains references to sexual acts, body parts, or other lewd content.
- **FLIRTATION**: pickup lines, complimenting appearance, subtle sexual innuendos, etc.

For more details on how these were trained, see the Toxicity and sub-attribute annotation guidelines.

New York Times moderation models

The following experimental models were trained on New York Times data tagged by their moderation team.

- **ATTACK_ON_AUTHOR**: Attack on the author of an article or post.
- **ATTACK_ON_COMMENTER**: Attack on fellow commenter.
- **INCOHERENT**: Difficult to understand, nonsensical.
- **INFLAMMATORY**: Intending to provoke or inflame.
- **LIKELY_TO_REJECT**: Overall measure of the likelihood for the comment to be rejected according to the NYT's moderation.
- **OBSCENE**: Obscene or vulgar language such as cursing.
- **SPAM**: Irrelevant and unsolicited commercial content.
- **UNSUBSTANTIAL**: Trivial or short comments.

`form_request` *Create a GET request for Perspective API*

Description

For more details see `?peRspective` or Perspective API documentation

Usage

```
form_request(score_model, text, score_sentences, languages,
  doNotStore = F)
```

Arguments

`score_model`	Specify what model do you want to use (for example `TOXICITY` and/or `SEVERE_TOXICITY`). Specify a character vector if you want more than one score. See `peRspective::prsp_models`.
`text`	a character string.
`score_sentences`	A boolean value that indicates if the request should return spans that describe the scores for each part of the text (currently done at per sentence level). Defaults to `FALSE`.
`languages`	A vector of ISO 631-1 two-letter language codes specifying the language(s) that comment is in (for example, "en", "es", "fr", "de", etc). If unspecified, the API will autodetect the comment language. If language detection fails, the API returns an error.
`doNotStore`	Whether the API is permitted to store comment from this request. Stored comments will be used for future research and community model building purposes to improve the API over time. Perspective API also plans to provide dashboards and automated analysis of the comments submitted, which will apply only to those stored. Defaults to `FALSE` (request data may be stored). Important note: This should be set to true if data being submitted is private (i.e. not publicly accessible), or if the data submitted contains content written by someone under 13 years old.

Value

a `tibble`

msg *Send a fancy message*

Description

Print a beautiful message in the console

Usage

```
msg(type, type_style = crayon::make_style("red4"), msg)
```

Arguments

type	what message should be displayed in the beginning
type_style	crayon color or style
msg	what message should be printed

Examples

```
## Send a message to the world
msg("MESSAGE", crayon::make_style('blue4'), "This is a message to the world")
```

perspective_api_key *Check if API key is present*

Description

Check if API key is present

Usage

```
perspective_api_key(test = F)
```

Arguments

test	necessary when in a test environment. Defaults to 'FALSE'.

print_progress *Print progress in purrr::imap environment*

Description

Provide iterator number and total length of items to be iterated over

Usage

```
print_progress(x, total, print_prct = F)
```

Arguments

x	iterator number.
total	length of items to be iterated over.
print_prct	only print percentage progress (defaults to FALSE).

Value

a chr

Examples

```
## Print progress (1 out of 100)
print_progress(1, 100)

## Only print percentage
print_progress(1, 100, print_prct = TRUE)
```

prsp_exp_models *All valid experimental Perspective API models*

Description

All valid experimental Perspective API models

prsp_models *All valid (non-experimental) Perspective API models*

Description

All valid (non-experimental) Perspective API models

prsp_score *Analyze comments with Perspective API*

Description

Provide a character string with your text, your API key and what scores you want to obtain.

Usage

```
prsp_score(text, text_id = NULL, languages = NULL,
  score_sentences = F, score_model, sleep = 1, doNotStore = F,
  key = NULL)
```

Arguments

text	a character string.
text_id	a unique ID for the text that you supply (required).
languages	A vector of ISO 631-1 two-letter language codes specifying the language(s) that comment is in (for example, "en", "es", "fr", "de", etc). If unspecified, the API will autodetect the comment language. If language detection fails, the API returns an error.
score_sentences	A boolean value that indicates if the request should return spans that describe the scores for each part of the text (currently done at per sentence level). Defaults to FALSE.
score_model	Specify what model do you want to use (for example TOXICITY and/or SEVERE_TOXICITY). Specify a character vector if you want more than one score. See peRspective::prsp_models.
sleep	how long should prsp_score wait between each call
doNotStore	Whether the API is permitted to store comment from this request. Stored comments will be used for future research and community model building purposes to improve the API over time. Perspective API also plans to provide dashboards and automated analysis of the comments submitted, which will apply only to those stored. Defaults to FALSE (request data may be stored). Important note: This should be set to true if data being submitted is private (i.e. not publicly accessible), or if the data submitted contains content written by someone under 13 years old.
key	Your API key (see here to set up an API key).

Details

For more details see ?peRspective or Perspective API documentation

Value

a tibble

Examples

```
## Not run:
## GET TOXICITY SCORES for a comment
prsp_score("Hello, I am a test comment!",
           score_model = "TOXICITY")

## GET TOXICITY and SEVERE_TOXICITY Scores for a comment
prsp_score("Hello, I am a test comment!",
           score_model = c("TOXICITY", "SEVERE_TOXICITY"))

## GET TOXICITY and SEVERE_TOXICITY Scores for each sentence of a comment
prsp_score("Hello, I am a test comment!
           I am a second sentence and I will (hopefully) be scored seperately",
           score_model = c("TOXICITY", "SEVERE_TOXICITY"),
           score_sentences = T)

## End(Not run)
```

`prsp_stream` *Stream comment scores with Perspective API*

Description

This function wraps `prsp_score` and loops over your text input. Provide a character string with your text and which scores you want to obtain. Make sure to keep track of your ratelimit with on the cloud console quota usage page.

Usage

```
prsp_stream(.data, text = NULL, text_id = NULL, ..., safe_output = F,
  verbose = F)
```

Arguments

`.data`	a dataset with a text and text_id column.
`text`	a character vector with text you want to score.
`text_id`	a unique ID for the text that you supply (required)
`...`	arguments passed to `prsp_score`.
`safe_output`	wraps the function into a `purrr::safely` environment (defaults to `FALSE`). Loop will run without pause and catch + output errors in a tidy `tibble` along with the results.
`verbose`	narrates the streaming procedure (defaults to `FALSE`).

Details

For more details see `?peRspective` or Perspective API documentation

Value

a tibble

Examples

```
## Not run:
## Create a mock tibble
text_sample <- tibble(
ctext = c("You wrote this? Wow. This is dumb and childish, please go f**** yourself.",
        "I don't know what to say about this but it's not good. The commenter is just an idiot",
          "This goes even further!",
          "What the hell is going on?",
          "Please. I don't get it. Explain it again",
          "Annoying and irrelevant! I'd rather watch the paint drying on the wall!"),
textid = c("#efdcxct", "#ehfcsct",
           "#ekacxwt",  "#ewatxad",
           "#ekacswt",  "#ewftxwd")
)

## GET TOXICITY and SEVERE_TOXICITY Scores for a dataset with a text column
text_sample %>%
prsp_stream(text = ctext,
            text_id = textid,
            score_model = c("TOXICITY", "SEVERE_TOXICITY"))

## Safe Output argument means will not stop on error
prsp_stream(text = ctext,
           text_id = textid,
           score_model = c("TOXICITY", "SEVERE_TOXICITY"),
           safe_output = T)


## verbose = T means you get pretty narration of your scoring procedure
prsp_stream(text = ctext,
           text_id = textid,
           score_model = c("TOXICITY", "SEVERE_TOXICITY"),
           safe_output = T,
           verbose = T)

## End(Not run)
```

---

specify_decimal *Specify a decimal*

Description

Specify a decimal

Usage

```
specify_decimal(x, k)
```

Arguments

x	a number to be rounded
k	round to which position after the comma

Examples

```
## specify 2 decimals of a number
specify_decimal(1.0434, 2)
```

---

unnest_scores *Unnest scores coming out of Perspective API*

Description

For more details see ?peRspective or Perspective API documentation

Usage

```
unnest_scores(Output, score_model, score_sentences, text)
```

Arguments

Output	comes out of the GET call.
score_model	Specify what model do you want to use (for example TOXICITY and/or SEVERE_TOXICITY). Specify a character vector if you want more than one score. See peRspective::prsp_models.
score_sentences	A boolean value that indicates if the request should return spans that describe the scores for each part of the text (currently done at per sentence level). Defaults to FALSE.
text	a character string.

Value

a tibble

Index