

Package ‘googlenlp’

May 8, 2026

Type Package

Title An Interface to Google's Cloud Natural Language API

Description Interact with Google's Cloud Natural Language API
<<https://cloud.google.com/natural-language/>> (v1) via R. The API has four main features, all of which are available through this R package: syntax analysis and part-of-speech tagging, entity analysis, sentiment analysis, and language identification.

Version 0.2.0

License MIT + file LICENSE

URL <https://github.com/BrianWeinstein/googlenlp>

LazyData TRUE

Imports dplyr, httr, jsonlite, purrr, readr, rlang

RoxygenNote 6.0.1

Suggests testthat

NeedsCompilation no

Author Brian Weinstien [aut, cre]

Maintainer Brian Weinstien <bweinstein02@gmail.com>

Repository CRAN

Date/Publication 2018-07-13 16:40:03 UTC

Contents

analyze_entities	2
analyze_sentiment	3
analyze_syntax	3
annotate_text	4
configure_googlenlp	5
flatten_entities	6
flatten_sentences	6
flatten_sentiment	7
flatten_tokens	8

gcnlp_key	9
gcnlp_post	9
get_config_file	10
set_api_key	11

Index	12
--------------	-----------

analyze_entities	<i>analyze_entities</i>
------------------	-------------------------

Description

Send a request, and retrieve the entities and language responses. This function retrieves the results from the `analyzeEntities` method.

Usage

```
analyze_entities(text_body, flatten = TRUE)
```

Arguments

<code>text_body</code>	The text string to send to the API.
<code>flatten</code>	If TRUE (default), then the results of each method are flattened and converted to a data frame.

Value

A list containing two elements: entities and language.

If `flatten` is TRUE, then the entities element is converted to a data frame.

Examples

```
## Not run:
sample_entities <- analyze_entities(text_body = "Google, headquartered in Mountain View, unveiled
the new Android phone at the Consumer Electronic Show.
Sundar Pichai said in his keynote that users love
their new Android phones.",
flatten = TRUE)

sample_entities$entities
sample_entities$language

## End(Not run)
```

analyze_sentiment *analyze_sentiment*

Description

Send a request, and retrieve the documentSentiment and language responses. This function retrieves the results from the [analyzeSentiment](#) method.

Usage

```
analyze_sentiment(text_body, flatten = TRUE)
```

Arguments

text_body	The text string to send to the API.
flatten	If TRUE (default), then the results of each method are flattened and converted to a data frame.

Value

A list containing two elements: documentSentiment and language.

If flatten is TRUE, then the documentSentiment element is converted to a data frame.

Examples

```
## Not run:
sample_sentiment <- analyze_sentiment(text_body = "Google, headquartered in Mountain View, unveiled
the new Android phone at the Consumer Electronic Show.
Sundar Pichai said in his keynote that users love
their new Android phones.",
flatten = TRUE)

sample_sentiment$documentSentiment
sample_sentiment$language

## End(Not run)
```

analyze_syntax *analyze_syntax*

Description

Send a request, and retrieve the sentences, tokens, and language responses. This function retrieves the results from the [analyzeSyntax](#) method.

Usage

```
analyze_syntax(text_body, flatten = TRUE)
```

Arguments

text_body	The text string to send to the API.
flatten	If TRUE (default), then the results of each method are flattened and converted to a data frame.

Value

A list containing three elements: sentences, tokens, and language.

If `flatten` is TRUE, then the sentences and tokens elements are each converted to data frames.

Examples

```
## Not run:
sample_syntax <- analyze_syntax(text_body = "Google, headquartered in Mountain View, unveiled
the new Android phone at the Consumer Electronic Show.
Sundar Pichai said in his keynote that users love
their new Android phones.",
flatten = TRUE)

sample_syntax$sentences
sample_syntax$tokens
sample_syntax$language

## End(Not run)
```

annotate_text	<i>annotate_text</i>
---------------	----------------------

Description

Send a request, and retrieve the sentences, tokens, entities, documentSentiment, and language responses. This function calls the `annotateText` method, which performs the `analyzeSyntax`, `analyzeEntities`, and `analyzeSentiment` methods all within one API call.

Usage

```
annotate_text(text_body, flatten = TRUE)
```

Arguments

text_body	The text string to send to the API.
flatten	If TRUE (default), then the results of each method are flattened and converted to a data frame.

Value

A list containing five elements: sentences, tokens, entities, documentSentiment, and language.

If flatten is TRUE, then the sentences, tokens, entities, and documentSentiment elements are each converted to data frames.

Examples

```
## Not run:
sample_annotate <- annotate_text(text_body = "Google, headquartered in Mountain View, unveiled
      the new Android phone at the Consumer Electronic Show.
      Sundar Pichai said in his keynote that users love
      their new Android phones.",
      flatten = TRUE)

sample_annotate$sentences
sample_annotate$tokens
sample_annotate$entities
sample_annotate$documentSentiment
sample_annotate$language

## End(Not run)
```

configure_googlenlp	<i>Configure your computer or a server to connect to the Google Cloud Natural Language API via R functions</i>
---------------------	--

Description

Creates variables in your .Renviron file for use by other googlenlp functions. This will edit your .Renviron file only if you call this function directly. If you prefer not to change your .Renviron file, use the set_api_key function instead.

Usage

```
configure_googlenlp()
```

Value

None

Examples

```
## Not run:
configure_googlenlp()

## End(Not run)
```

flatten_entities *Flatten entities*

Description

Convert the JSON/list entities response into a flattened data frame.

Usage

```
flatten_entities(entities_list)
```

Arguments

entities_list The entities component of the API response.

Value

A flattened data frame.

Examples

```
## Not run:
sample_post <- gcnlp_post(text_body = "Google, headquartered in Mountain View, unveiled
                                     the new Android phone at the Consumer Electronic Show.
                                     Sundar Pichai said in his keynote that users love
                                     their new Android phones.",
                           extract_syntax = TRUE,
                           extract_entities = TRUE,
                           extract_document_sentiment = TRUE)

flatten_entities(entities_list = sample_post$content$entities)

## End(Not run)
```

flatten_sentences *Flatten sentences*

Description

Convert the JSON/list sentences response into a flattened data frame.

Usage

```
flatten_sentences(sentences_list)
```

Arguments

sentences_list The sentences component of the API response.

Value

A flattened data frame.

Examples

```
## Not run:
sample_post <- gcnlp_post(text_body = "Google, headquartered in Mountain View, unveiled
                                     the new Android phone at the Consumer Electronic Show.
                                     Sundar Pichai said in his keynote that users love
                                     their new Android phones.",
                           extract_syntax = TRUE,
                           extract_entities = TRUE,
                           extract_document_sentiment = TRUE)

flatten_sentences(sentences_list = sample_post$content$sentences)

## End(Not run)
```

flatten_sentiment	<i>Flatten sentiment</i>
-------------------	--------------------------

Description

Convert the JSON/list sentiment response into a flattened data frame.

Usage

```
flatten_sentiment(sentiment_list)
```

Arguments

sentiment_list The sentiment component of the API response.

Value

A flattened data frame.

Examples

```
## Not run:
sample_post <- gcnlp_post(text_body = "Google, headquartered in Mountain View, unveiled
                                     the new Android phone at the Consumer Electronic Show.
                                     Sundar Pichai said in his keynote that users love
                                     their new Android phones.",
                           extract_syntax = TRUE,
                           extract_entities = TRUE,
                           extract_document_sentiment = TRUE)

flatten_sentiment(sentiment_list = sample_post$content$sentiment)

## End(Not run)
```

flatten_tokens	<i>Flatten tokens</i>
----------------	-----------------------

Description

Convert the JSON/list tokens response into a flattened data frame.

Usage

```
flatten_tokens(tokens_list)
```

Arguments

tokens_list The tokens component of the API response.

Value

A flattened data frame.

Examples

```
## Not run:
sample_post <- gcnlp_post(text_body = "Google, headquartered in Mountain View, unveiled
                                     the new Android phone at the Consumer Electronic Show.
                                     Sundar Pichai said in his keynote that users love
                                     their new Android phones.",
                           extract_syntax = TRUE,
                           extract_entities = TRUE,
                           extract_document_sentiment = TRUE)

flatten_tokens(tokens_list = sample_post$content$tokens)

## End(Not run)
```

gcnlp_key	<i>Retrieve API key</i>
-----------	-------------------------

Description

Retrieve API key

Usage

```
gcnlp_key()
```

Value

Your API key

Examples

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

gcnlp_post	<i>Send a POST request to the Google Cloud Natural Language API</i>
------------	---

Description

Send a POST request to the Google Cloud Natural Language API and retrieve the results.

Usage

```
gcnlp_post(text_body, extract_syntax = TRUE, extract_entities = TRUE,  
           extract_document_sentiment = TRUE)
```

Arguments

text_body	The text string to send to the API.
extract_syntax	Behavior for the analyzeSyntax method. Defaults to TRUE. See the API documentation for more information.
extract_entities	Behavior for the analyzeEntities method. Defaults to TRUE. See the API documentation for more information.
extract_document_sentiment	Behavior for the analyzeSentiment method. Defaults to TRUE. See the API documentation for more information.

Value

A list containing two elements: [1] content includes the parsed response, and contains the sentences, tokens, entities, documentSentiment, language results specified in the request. [2] raw_response contains the raw response from the API.

Examples

```
## Not run:
gcnlp_post(text_body = "Google, headquartered in Mountain View, unveiled
                    the new Android phone at the Consumer Electronic Show.
                    Sundar Pichai said in his keynote that users love
                    their new Android phones.",
           extract_syntax = TRUE,
           extract_entities = TRUE,
           extract_document_sentiment = TRUE)

## End(Not run)
```

get_config_file

Fetch session-specific gcnlp default values

Description

get_config_file() gets the value of config_file

Usage

```
get_config_file()
```

Value

The path to the user's config_file

Examples

```
## Not run:
get_config_file()

## End(Not run)
```

set_api_key	<i>Manually set access credentials</i>
-------------	--

Description

Manually define an API key. Only use this function if you haven't run `configure_googlenlp()`

Usage

```
set_api_key(api_key)
```

Arguments

api_key Your API key, from <https://console.cloud.google.com/apis/credentials>

Value

None

Examples

```
## Not run:  
set_api_key("YOUR_API_KEY")  
  
## End(Not run)
```

Index

[analyze_entities](#), 2
[analyze_sentiment](#), 3
[analyze_syntax](#), 3
[annotate_text](#), 4

[configure_googlenlp](#), 5

[flatten_entities](#), 6
[flatten_sentences](#), 6
[flatten_sentiment](#), 7
[flatten_tokens](#), 8

[gcnlp_key](#), 9
[gcnlp_post](#), 9
[get_config_file](#), 10

[set_api_key](#), 11