Package ‘aws.comprehend’

April 12, 2018

Type Package

Title 'AWS Comprehend' Client Package

Version 0.1.2

Date 2018-04-09

Description Client for 'AWS Comprehend' <https://aws.amazon.com/comprehend>, a cloud natural language processing service that can perform a number of quantitative text analyses, including language detection, sentiment analysis, and feature extraction.

License GPL (>= 2)

URL https://github.com/cloudyr/aws.comprehend

BugReports https://github.com/cloudyr/aws.comprehend/issues

Imports httr, jsonlite, aws.signature (>= 0.3.4)

Suggests testthat

Encoding UTF-8

RoxygenNote 6.0.1

NeedsCompilation no

Author Thomas J. Leeper [aut, cre]

Maintainer Thomas J. Leeper <thosjleeper@gmail.com>

Repository CRAN

Date/Publication 2018-04-12 12:54:27 UTC

R topics documented:

aws.comprehend-package *aws.comprehend*

Description

AWS Comprehend Client Package

Details

Client for AWS Comprehend (https://aws.amazon.com/comprehend0, a cloud natural language processing service that can perform a number of quantitative text analyses, including language detection, sentiment analysis, and feature extraction.

Author(s)

Thomas J. Leeper <thosjleeper@gmail.com>

See Also

detect_language, detect_sentiment, detect_entities, detect_phrases

comprehendHTTP *Execute AWS Comprehend API Request*

Description

This is the workhorse function to execute calls to the Comprehend API.

Usage

```
comprehendHTTP(action, query = list(), body = NULL, region = NULL,
  key = NULL, secret = NULL, session_token = NULL, ...)
```

Arguments

action	A character string specifying the API action to take
query	An optional named list containing query string parameters and their character values.
body	A request body
region	A character string containing an AWS region. If missing, the default “us-east-1” is used.
key	A character string containing an AWS Access Key ID. The default is pulled from environment variable “AWS_ACCESS_KEY_ID”.

`secret` A character string containing an AWS Secret Access Key. The default is pulled from environment variable “AWS_SECRET_ACCESS_KEY”.

`session_token` Optionally, a character string containing an AWS temporary Session Token. If missing, defaults to value stored in environment variable “AWS_SESSION_TOKEN”.

`...` Additional arguments passed to `GET`.

Details

This function constructs and signs an Polly API request and returns the results thereof, or relevant debugging information in the case of error.

Value

If successful, a named list. Otherwise, a data structure of class “aws-error” containing any error message(s) from AWS and information about the request attempt.

Author(s)

Thomas J. Leeper

`detect_entities` *Detect named entities in a source text*

Description

Detect entities in a source text

Usage

```
detect_entities(text, language = "en", ...)
```

Arguments

`text` A character string containing a text to entities analyze, or a character vector to perform analysis separately for each element.

`language` A character string containing a two-letter language code. Currently “en” and “es” are supported.

`...` Additional arguments passed to `comprehendHTTP`.

Value

A data frame

Examples

```
## Not run:
  # simple example
  detect_entities("Amazon provides web services. Jeff is their leader.")

  txt <-c("Amazon provides web services.",
          "Jeff is their leader.")
  detect_entities(txt)

## End(Not run)
```

detect_language *Detect language in a source text*

Description

Detect language(s) in a source text

Usage

```
detect_language(text, ...)
```

Arguments

`text` A character string containing a textual source, or a character vector to detect languages separately for each element.

`...` Additional arguments passed to `comprehendHTTP`.

Value

A data frame of language probabilities.

Examples

```
## Not run:
  # simple example
  detect_language("This is a test sentence in English")

  # two languages in a single text
  txt <- "A: ¡Hola! ¿Como está, usted?\nB: Ça va bien. Merci. Et toi?"
  detect_language(txt)

  # "batch" mode
  detect_language(c("A: ¡Hola! ¿Como está, usted?",
                    "B: Ça va bien. Merci. Et toi?"))

## End(Not run)
```

detect_phrases *Detect key phrases*

Description

Detect key phrases in a source text

Usage

```
detect_phrases(text, language = "en", ...)
```

Arguments

`text`	A character string containing a text to analyze, or a character vector to perform analysis separately for each element.
`language`	A character string containing a two-letter language code. Currently “en” and “es” are supported.
`...`	Additional arguments passed to `comprehendHTTP`.

Value

A data frame

Examples

```
## Not run:
  # simple example
  detect_phrases("Amazon provides web services. Jeff is their leader.")

  txt <-c("Amazon provides web services.",
          "Jeff is their leader.")
  detect_phrases(txt)

## End(Not run)
```

detect_sentiment *Detect sentiment in a source text*

Description

Detect sentiment in a source text

Usage

```
detect_sentiment(text, language = "en", ...)
```

Arguments

`text`	A character string containing a text to sentiment analyze, or a character vector to perform analysis separately for each element.
`language`	A character string containing a two-letter language code. Currently “en” and “es” are supported.
`...`	Additional arguments passed to `comprehendHTTP`.

Value

A data frame

Examples

```
## Not run:
  # simple example
  detect_sentiment("I have never been happier. This is the best day ever.")

  txt <-c("I have never been happier. This is the best day ever.",
          "I have always been happier. This is the worst day ever.")
  detect_sentiment(txt)

## End(Not run)
```

Index