

# Package: aws.comprehend (via r-universe)

October 23, 2024

**Type** Package

**Title** Client for 'AWS Comprehend'

**Version** 0.2.1.9000

**Date** 2020-03-10

**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** htrr, jsonlite, aws.signature (>= 0.3.4)

**Suggests** testthat (>= 2.1.0)

**Depends** R (>= 3.5.0)

**Encoding** UTF-8

**RoxygenNote** 7.0.2

**Repository** <https://cloudyr.r-universe.dev>

**RemoteUrl** <https://github.com/cloudyr/aws.comprehend>

**RemoteRef** HEAD

**RemoteSha** d48c95542c28f268d67929c8ea828437a7a9d72b

## Contents

aws.comprehend-package . . . . .	2
bind_and_index . . . . .	2
comprehendHTTP . . . . .	3
detect_entities . . . . .	4
detect_language . . . . .	5
detect_medical_entities . . . . .	5

detect_medical_phi . . . . .	6
detect_phrases . . . . .	7
detect_sentiment . . . . .	8
detect_syntax . . . . .	8
flatten . . . . .	9

## Index 10

aws.comprehend-package

*aws.comprehend*

### Description

AWS Comprehend Client Package

### Details

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.

### Author(s)

Thomas J. Leeper <[thosjleeper@gmail.com](mailto:thosjleeper@gmail.com)>

### See Also

[detect\\_language](#), [detect\\_sentiment](#), [detect\\_entities](#), [detect\\_phrases](#)

bind\_and\_index

*Bind and index a ResultList*

### Description

Turn a list of data.frames (of different lengths and potentially empty) into a single indexed data.frame. Useful to process a ResultList from 'comprehendHTTP'.

### Usage

```
bind_and_index(index, df_list)
```

### Arguments

index	Vector of indices
df_list	List of data.frames to bind and index. Should NOT be a data.frame.

**Details**

'index' and 'df\_list' should be the same length. An error is raised otherwise.

```
bind_and_index(1:2, list(data.frame(col = "a"), data.frame(col = "b")))
```

```
bind_and_index(1:3, list( data.frame(col = "a"), data.frame(), data.frame(c("b", "c"))))
```

---

 comprehendHTTP

*Execute AWS Comprehend API Request*


---

**Description**

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

**Usage**

```
comprehendHTTP(
  action,
  query = list(),
  headers = list(),
  body = NULL,
  verbose = getOption("verbose", FALSE),
  region = Sys.getenv("AWS_DEFAULT_REGION", "us-east-1"),
  key = NULL,
  secret = NULL,
  session_token = NULL,
  service = c("comprehend", "comprehendmedical"),
  ...
)
```

**Arguments**

action	A character string specifying the API action to take
query	An optional named list containing query string parameters and their character values.
headers	A list of headers to pass to the HTTP request.
body	A request body
verbose	A logical indicating whether to be verbose. Default is given by options("verbose").
region	A character string containing the AWS region. If missing, defaults to "us-east-1".
key	A character string containing an AWS Access Key ID. See <a href="#">locate_credentials</a> .
secret	A character string containing an AWS Secret Access Key. See <a href="#">locate_credentials</a> .
session_token	A character string containing an AWS Session Token. See <a href="#">locate_credentials</a> .
service	the Comprehend service to use. Currently either 'comprehend' for the base service or 'comprehendmedical' for the Comprehend Medical service.
...	Additional arguments passed to <a href="#">GET</a> .

**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	<i>Detect named entities in a source text</i>
-----------------	-----------------------------------------------

---

**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 <a href="#">comprehendHTTP</a> .

**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, like Google.",  
       "Jeff is their leader.")  
detect_entities(txt)  
  
## End(Not run)
```

---

detect_language	<i>Detect language in a source text</i>
-----------------	-----------------------------------------

---

**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 <a href="#">comprehendHTTP</a> .

**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_medical_entities	<i>Detect named entities in a source medical text</i>
-------------------------	-------------------------------------------------------

---

**Description**

Detect entities in a source medical text

**Usage**

```
detect_medical_entities(text, language = "en", version = c("2", "1"), ...)
```

**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 only “en” is supported.
version	A character string containing the version of the API that should be used. Currently only "1" or "2" are supported.
...	Additional arguments passed to <a href="#">comprehendHTTP</a> .

**Value**

A data frame

**Examples**

```
## Not run:
# simple example
medical_detect_entities("Mrs. Smith comes in today complaining of shortness of breath.")

txt <-c("Mrs. Smith comes in today.",
       "She is complaining of shortness of breath.")
medical_detect_entities(txt)

## End(Not run)
```

---

detect_medical_phi	<i>Detect Protected Health Information (PHI) in a source medical text</i>
--------------------	---------------------------------------------------------------------------

---

**Description**

Detect Protected Health Information (PHI) in a source medical text

**Usage**

```
detect_medical_phi(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 only “en” is supported.
...	Additional arguments passed to <a href="#">comprehendHTTP</a> .

**Value**

A data frame

**Examples**

```
## Not run:
# simple example
medical_detect_phi("Mrs. Smith comes in today complaining of shortness of breath.")

txt <-c("Mrs. Smith comes in today.",
       "She is complaining of shortnesss of breath.")
medical_detect_phi(txt)

## End(Not run)
```

---

detect_phrases	<i>Detect key phrases</i>
----------------	---------------------------

---

**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 <a href="#">comprehendHTTP</a> .

**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 <a href="#">comprehendHTTP</a> .

### 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)
```

---

detect\_syntax      *Detect syntax in a source text*

---

### Description

Detect syntax in a source text

### Usage

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



**Arguments**

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

language       A character string containing a two-letter language code.

...            Additional arguments passed to [comprehendHTTP](#).

**Value**

A data frame

**Examples**

```
## Not run:
# simple example
detect_syntax("The quick brown fox jumps over the lazy dog.")

txt <-c("The quick brown fox jumps over the lazy dog.",
       "I have never been happier!")
detect_syntax(txt)

## End(Not run)
```

---

flatten                      *Flatten embedded data.frames (1 level max)*

---

**Description**

Flatten embedded data.frames (1 level max)

**Usage**

```
flatten(df)
```

**Arguments**

df                      data.frame to flatten

# Index

## \* **package**

aws.comprehend-package, 2

aws.comprehend

(aws.comprehend-package), 2

aws.comprehend-package, 2

bind\_and\_index, 2

comprehendHTTP, 3, 4-9

detect\_entities, 2, 4

detect\_language, 2, 5

detect\_medical\_entities, 5

detect\_medical\_phi, 6

detect\_phrases, 2, 7

detect\_sentiment, 2, 8

detect\_syntax, 8

flatten, 9

GET, 3

locate\_credentials, 3