

# Package ‘polyglotr’

May 3, 2024

**Title** Translate Text

**Version** 1.5.0

**Description** The goal of the this package is to provide easy methods to translate pieces of text. Functions send requests to translation services online.

**License** MIT + file LICENSE

**URL** <https://github.com/Tomeriko96/polyglotr/>,  
<https://tomeriko96.github.io/polyglotr/>

**BugReports** <https://github.com/Tomeriko96/polyglotr/issues>

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**Imports** dplyr, httr, jsonlite, magrittr, purrr, RCurl, rlang, rvest,  
stringr, tibble, urltools

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0), text2vec

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**Depends** R (>= 2.10)

**LazyData** true

**NeedsCompilation** no

**Author** Tomer Iwan [aut, cre, cph]

**Maintainer** Tomer Iwan <t.iwan@vu.nl>

**Repository** CRAN

**Date/Publication** 2024-05-03 12:50:02 UTC

## R topics documented:

batch_translate . . . . .	2
create_translation_table . . . . .	3
create_transliteration_table . . . . .	3
google_get_supported_languages . . . . .	4

google_is_valid_language_code . . . . .	4
google_supported_languages . . . . .	5
google_translate . . . . .	5
google_transliterate . . . . .	6
language_detect . . . . .	7
linguee_external_sources . . . . .	7
linguee_translation_examples . . . . .	8
linguee_word_translation . . . . .	9
microsoft_supported_languages . . . . .	10
mymemory_translate . . . . .	11
pons_dictionaries . . . . .	11
pons_translate . . . . .	12
qcri_api_key . . . . .	13
qcri_get_domains . . . . .	13
qcri_get_language_pairs . . . . .	14
qcri_translate_text . . . . .	14
translate_file . . . . .	15
translate_to_morse . . . . .	16
translate_to_morse_audio . . . . .	17
wikimedia_detect_language . . . . .	17
wikipedia_get_language_names . . . . .	18
wmcloud_translate . . . . .	18

## Index 20

---

batch_translate	<i>Batch Translation Function</i>
-----------------	-----------------------------------

---

### Description

This function translates a file into each target language using the polyglotr package's `translate_file` function, and saves the translated files.

### Usage

```
batch_translate(input_file, source_language, target_languages)
```

### Arguments

<code>input_file</code>	A character string indicating the path to the input file.
<code>source_language</code>	A character string indicating the source language.
<code>target_languages</code>	A character vector indicating the target languages.

### Value

Nothing is returned.

**Examples**

```
## Not run:  
batch_translate("README.md", "nl", c("fr", "es", "de"))  
  
## End(Not run)
```

---

```
create_translation_table  
      Create a Translation Table
```

---

**Description**

This function generates a translation table by translating a list of words into multiple languages.

**Usage**

```
create_translation_table(words, languages)
```

**Arguments**

words	A character vector containing the words to be translated.
languages	A character vector specifying the target languages for translation.

**Value**

A data frame representing the translation table with original words and translations in each language.

**Examples**

```
words <- c("Hello", "Translate", "Table", "Script")  
languages <- c("es", "fr", "de", "nl")  
translations <- create_translation_table(words, languages)  
print(translations)
```

---

```
create_transliteration_table  
      Create a Transliteration Table
```

---

**Description**

This function generates a transliteration table by transliterating a list of words into multiple languages.

**Usage**

```
create_transliteration_table(words, languages)
```

**Arguments**

words            A character vector containing the words to be transliterated.  
languages        A character vector specifying the target languages for transliteration.

**Value**

A data frame representing the transliteration table with original words and transliterations in each language.

**Examples**

```
words <- c("Hello world", "Goodbye", "Thank you", "Please")  
languages <- c("ar", "he", "el", "ru", "fa")  
transliterations <- create_transliteration_table(words, languages)  
print(transliterations)
```

---

google\_get\_supported\_languages  
*Get Supported Languages*

---

**Description**

This function fetches the supported languages from the Google Cloud Translate documentation page.

**Usage**

```
google_get_supported_languages()
```

**Value**

A data frame containing the supported languages and their corresponding ISO 639-1 codes.

---

google\_is\_valid\_language\_code  
*Check if a language code is valid*

---

**Description**

This function checks if a given language code is in the google\_supported\_languages dataset.

**Usage**

```
google_is_valid_language_code(language_code)
```

**Arguments**

language\_code The language code to check.

**Value**

A logical value indicating if the language code is valid.

**Examples**

```
google_is_valid_language_code("en") # TRUE
google_is_valid_language_code("fr") # TRUE
google_is_valid_language_code("xx") # FALSE
```

---

google\_supported\_languages

*Google Supported Languages*

---

**Description**

This dataset contains the language names and iso codes of languages supported by Google Translate API.

**Usage**

```
google_supported_languages
```

**Format**

A data frame with two variables: language\_name and iso\_code

**Source**

Google Translate API

---

google\_translate

*Translate text using google translate*

---

**Description**

Translate text using google translate

**Usage**

```
google_translate(text, target_language = "en", source_language = "auto")
```

**Arguments**

text	This is the text that you want to translate.
target_language	This is the language that you want to translate the text into. The default value for this argument is "en" for English.
source_language	This is the language of the text that you want to translate. The default value for this argument is "auto", which means that the function will try to automatically detect the language of the text.

**Value**

Translated text.

**Examples**

```
google_translate("I love languages", target_language = "es")
text_to_translate <- c("the", "quick", "brown")
google_translate(text_to_translate, "fr", "en")
```

---

google\_transliterate *Transliterate a single word or a sentence to the required language.*

---

**Description**

Transliterate a single word or a sentence to the required language.

**Usage**

```
google_transliterate(text, language_tag = "el", num = 5)
```

**Arguments**

text	The word or sentence to transliterate from Latin/Roman (English) script.
language_tag	The target language's ISO639 code. The default value for this argument is "el" for Greek.
num	The maximum number of suggestions to fetch. The default value for this argument is 5.

**Value**

Character vector of transliterated sentences or larger pieces of text.

**Examples**

```
## Not run:
google_transliterate("Hello world", "fr", 10)
google_transliterate("hello", "el", 10)

## End(Not run)
```

---

language_detect	<i>Detect Language using Google Translate API</i>
-----------------	---

---

**Description**

This function detects the language of a given text using the Google Translate API.

**Usage**

```
language_detect(text)
```

**Arguments**

text            The text for which the language needs to be detected.

**Value**

A character string representing the detected language.

---

linguee_external_sources	<i>Retrieve external sources using Linguee Translation API</i>
--------------------------	--

---

**Description**

Retrieve external sources using Linguee Translation API

**Usage**

```
linguee_external_sources(query, src, dst, limit = 5)
```

**Arguments**

query            The word or phrase for which you want to retrieve external sources.

src              The source language of the word or phrase. Accepts language codes such as "en", "es", "fr", etc.

dst              The target language for the external source retrieval. Accepts language codes such as "en", "es", "fr", etc.

limit            The maximum number of external sources to retrieve. Defaults to 5.

**Value**

A dataframe of external sources with columns: src, dst, src\_url, dst\_url.

**See Also**

linguee\_word\_translation, linguee\_translation\_examples

**Examples**

```
linguee_external_sources(query = "hello", src = "en", dst = "es")
```

---

```
linguee_translation_examples
```

*Provide translation examples using Linguee Translation API*

---

**Description**

Provide translation examples using Linguee Translation API

**Usage**

```
linguee_translation_examples(  
    query,  
    src,  
    dst,  
    guess_direction = FALSE,  
    follow_corrections = "always"  
)
```

**Arguments**

query	The word or phrase for which you want translation examples.
src	The source language of the word or phrase. Accepts language codes such as "en", "es", "fr", etc.
dst	The target language for the translation examples. Accepts language codes such as "en", "es", "fr", etc.
guess_direction	A boolean flag that determines whether the API should guess the translation direction. The default value is FALSE.
follow_corrections	Specifies how to treat responses with a "did you mean" link. Possible values are "always", "never", or "on_empty_translations". The default value is "always".



**Value**

A dataframe of translation examples with columns: source, target, pos.

**See Also**

linguee\_word\_translation

**Examples**

```
linguee_translation_examples(query = "hello", src = "en", dst = "es")
```

---

linguee\_word\_translation

*Translate word using Linguee Translation API*

---

**Description**

Translate word using Linguee Translation API

**Usage**

```
linguee_word_translation(  
  word,  
  target_language,  
  source_language,  
  guess_direction = FALSE,  
  follow_corrections = "always"  
)
```

**Arguments**

word	This is the word that you want to translate.
target_language	This is the language that you want to translate the word into.
source_language	This is the language of the word that you want to translate.
guess_direction	Specifies whether the API should guess the translation direction when the source language is set to "auto". The default value is FALSE.
follow_corrections	Specifies whether the API should include translations that have been marked as corrections. The default value is "always" to include corrections.

**Value**

Translated word options.

**Examples**

```
linguee_word_translation("hello", target_language = "es", source_language = "en")
```

---

microsoft\_supported\_languages

*Get the set of languages currently supported by the Microsoft Translator API*

---

**Description**

Get the set of languages currently supported by the Microsoft Translator API

**Usage**

```
microsoft_supported_languages(scope = NULL)
```

**Arguments**

scope (optional) A comma-separated list of names defining the group of languages to return. Allowed group names are: translation, transliteration, and dictionary. If no scope is given, then all groups are returned.

**Value**

A list of supported languages for the specified groups.

**Examples**

```
## Not run:  
microsoft_supported_languages(scope = "translation,transliteration,dictionary")  
  
## End(Not run)
```

---

mymemory\_translate     *Translate text using mymemory translate*

---

**Description**

Translate text using mymemory translate

**Usage**

```
mymemory_translate(text, target_language = "en", source_language = "auto")
```

**Arguments**

text	Text to translate.
target_language	Language to translate text to.
source_language	Language to translate text from

**Value**

Translated text.

**Examples**

```
mymemory_translate("Hello World", target_language = "es", source_language = "en")
```

---

pons\_dictionaries     *Get the list of available dictionaries from PONS API*

---

**Description**

Get the list of available dictionaries from PONS API

**Usage**

```
pons_dictionaries(language = "en")
```

**Arguments**

language	The language of the output (ISO 639-1 - two-letter codes). Supported languages are de, el, en, es, fr, it, pl, pt, ru, sl, tr, zh.
----------	--

**Value**

A list of available dictionaries in the specified language.

**Examples**

```
## Not run:
pons_dictionaries(language = "es")

## End(Not run)
```

---

pons_translate	<i>Translate text using PONS</i>
----------------	----------------------------------

---

**Description**

Translate text using PONS

**Usage**

```
pons_translate(text, target_language = "pt", source_language = "en")
```

**Arguments**

text	This is the text that you want to translate. Can be a single string or a vector of strings.
target_language	This is the language that you want to translate the text into. The default value for this argument is "pt" for Portuguese.
source_language	This is the language of the text that you want to translate. The default value for this argument is "en" for English.

**Value**

Translated text. If the input is a vector, it returns a character vector of translated strings.

**Examples**

```
## Not run:
pons_translate("I love languages!", target_language = "pt", source_language = "en")
text_to_translate <- c("The", "Greatest", "Language")
pons_translate(text_to_translate, "pt", "en")

## End(Not run)
```

---

qcri_api_key	<i>Get the QCRI API key from the environment variable</i>
--------------	---

---

**Description**

Get the QCRI API key from the environment variable

**Usage**

```
qcri_api_key()
```

**Value**

The QCRI API key stored in the QCRI\_API\_KEY environment variable.

---

qcri_get_domains	<i>QCRI Get Domains</i>
------------------	-------------------------

---

**Description**

This function retrieves the supported domains from the QCRI Multiterm API.

**Usage**

```
qcri_get_domains(api_key = qcri_api_key())
```

**Arguments**

api_key	The API key associated with the user account being used. If not provided, the function will attempt to retrieve it from the QCRI_API_KEY environment variable.
---------	--

**Value**

A list with keys:

- success: Boolean indicating whether the request succeeded.
- domains: Array of supported domains, such as news, tedtalks etc. Only present if success is true.
- error: Error message in case success is false.

**Examples**

```
## Not run:  
qcri_get_domains(api_key = "YourApiKey")  
qcri_get_domains()  
  
## End(Not run)
```

---

`qcri_get_language_pairs`*QCRI Get Language Pairs*

---

**Description**

This function retrieves the supported language pairs from the QCRI Multiterm API.

**Usage**

```
qcri_get_language_pairs(api_key = qcri_api_key())
```

**Arguments**

<code>api_key</code>	The API key associated with the user account being used. If not provided, the function will attempt to retrieve it from the <code>QCRI_API_KEY</code> environment variable. You can register for an API key at <a href="https://mt.qcri.org/api/register">https://mt.qcri.org/api/register</a>
----------------------	--

**Value**

Language pairs.

**Examples**

```
## Not run:  
qcri_get_language_pairs(api_key = "YourApiKey")  
qcri_get_language_pairs()  
  
## End(Not run)
```

---

`qcri_translate_text`*QCRI Translate Text*

---

**Description**

This function translates a text from the source language to the target language using the QCRI Multiterm API.

**Usage**

```
qcri_translate_text(text, langpair, domain, api_key = qcri_api_key())
```

**Arguments**

text	The text to be translated. This must be URL encoded.
langpair	The source-target language pair, where source is language of the provided text and target is the language into which the text has to be translated.
domain	The domain over which the translation is tuned.
api_key	The API key associated with the user account being used. If not provided, the function will attempt to retrieve it from the QCRI_API_KEY environment variable.

**Value**

Translated text.

**Examples**

```
## Not run:
qcri_translate_text(text = "Hello, world!",
  langpair = "en-ar",
  domain = "general",
  api_key = "YourApiKey")
qcri_translate_text(text = "Hello, world!",
  langpair = "en-ar",
  domain = "general")

## End(Not run)
```

---

translate\_file

*Translate File*

---

**Description**

Translates the content of a file using Google Translate API.

**Usage**

```
translate_file(
  file_path,
  target_language = "en",
  source_language = "auto",
  overwrite = FALSE
)
```

**Arguments**

file_path	The path to the file to be translated.
target_language	The target language to translate the file content to. Default is "en".
source_language	The source language of the file content. Default is "auto".
overwrite	Logical indicating whether to overwrite the original file with the translated content. Default is FALSE.

**Examples**

```
## Not run:  
translate_file("path/to/file.txt", target_language = "fr", source_language = "en", overwrite = TRUE)  
  
## End(Not run)
```

---

translate\_to\_morse      *Translate Text to Morse Code using the FunTranslations API*

---

**Description**

This function takes a string of text as input and translates it to Morse code using the FunTranslations API.

**Usage**

```
translate_to_morse(text, api_key = NULL)
```

**Arguments**

text	A character string containing the text to be translated to Morse code.
api_key	(optional) Your FunTranslations API key, if you have a paid subscription.

**Value**

A list containing the translated Morse code text and other metadata.



---

`translate_to_morse_audio`*Translate English Text to Morse Code with Audio*

---

**Description**

This function takes an English text string as input and translates it to Morse code with an audio output using the FunTranslations API.

**Usage**

```
translate_to_morse_audio(text, api_key = NULL)
```

**Arguments**

<code>text</code>	A character string containing the English text to be translated.
<code>api_key</code>	(optional) Your FunTranslations API key, if you have a paid subscription.

**Value**

A list containing the translated Morse code text, the Morse code audio as a base64-encoded string, and other metadata.

---

`wikimedia_detect_language`*Detect the language of a text*

---

**Description**

This function sends a POST request to the Wikimedia Language ID API with the specified text, parses the JSON response, and returns the detected language.

**Usage**

```
wikimedia_detect_language(text)
```

**Arguments**

<code>text</code>	The text whose language is to be detected.
-------------------	--

**Value**

The detected language.

**Examples**

```
# Detect the language of a text
wikimedia_detect_language("Hallo, wereld")
```

---

```
wikipedia_get_language_names
    Get language names
```

---

**Description**

This function sends a GET request to the Wikipedia API and returns the language names as a dataframe.

**Usage**

```
wikipedia_get_language_names()
```

**Value**

A dataframe of language names.

**Examples**

```
# Get language names
wikipedia_get_language_names()
```

---

```
wmcloud_translate    Translate content using WMCloud
```

---

**Description**

This function sends a POST request to the WMCloud translation API with the specified parameters, parses the JSON response, and returns the translated content.

**Usage**

```
wmcloud_translate(
  content,
  target_language = "en",
  source_language = "en",
  format = "text",
  model = "nllb200-600M"
)
```

**Arguments**

content	The content to translate. Can be plain text, a URL (for a webpage), a JSON string, or a Markdown string.
target_language	The target language for the translation (default is "en").
source_language	The source language of the content (default is "en").
format	The format of the content ("json", "markdown", "text", "webpage").
model	The model to use for the translation (only "nllb200-600M" is currently known to work).

**Value**

The translated content.

**Examples**

```
## Not run:
# Translate plain text
wmcloud_translate("rijst",
  target_language = "es",
  source_language = "nl", format = "text")

# Translate a webpage
wmcloud_translate("https://en.m.wikivoyage.org/wiki/Goes",
  target_language = "es",
  source_language = "en", format = "webpage")

# Translate JSON content
wmcloud_translate('{
  "id": 1,
  "title": "Chicken Biryani",
  "description": "Chicken Biryani is a savory chicken and rice dish",
  "ingredients": [ "Vegetable oil", "Garlic", "Ginger" ,"Rice"]
}
', target_language = "es", source_language = "en", format = "json")

# Translate Markdown content
wmcloud_translate('# Heading

This is a [link to Wikipedia](https://wikipedia.org)
', target_language = "es", source_language = "en", format = "markdown")

## End(Not run)
```

# Index

## \* **data**

- google\_supported\_languages, [5](#)
- batch\_translate, [2](#)
- create\_translation\_table, [3](#)
- create\_transliteration\_table, [3](#)
- google\_get\_supported\_languages, [4](#)
- google\_is\_valid\_language\_code, [4](#)
- google\_supported\_languages, [5](#)
- google\_translate, [5](#)
- google\_transliterate, [6](#)
- language\_detect, [7](#)
- linguee\_external\_sources, [7](#)
- linguee\_translation\_examples, [8](#)
- linguee\_word\_translation, [9](#)
- microsoft\_supported\_languages, [10](#)
- mymemory\_translate, [11](#)
- pons\_dictionaries, [11](#)
- pons\_translate, [12](#)
- qcri\_api\_key, [13](#)
- qcri\_get\_domains, [13](#)
- qcri\_get\_language\_pairs, [14](#)
- qcri\_translate\_text, [14](#)
- translate\_file, [15](#)
- translate\_to\_morse, [16](#)
- translate\_to\_morse\_audio, [17](#)
- wikimedia\_detect\_language, [17](#)
- wikipedia\_get\_language\_names, [18](#)
- wmcloud\_translate, [18](#)