Package ‘translateR’

February 20, 2015

Type Package
Title Bindings for the Google and Microsoft Translation APIs
Version 1.0
Author Christopher Lucas and Dustin Tingley
Maintainer Christopher Lucas <clucas@fas.harvard.edu>
Description translateR provides easy access to the Google and Microsoft APIs. The package is easy to use with the related R package `stm` for the estimation of multilingual topic models.
Imports RJSONIO,RCurl,textcat,parallel,httr
License GPL-3
NeedsCompilation no
Repository CRAN
Date/Publication 2014-07-09 08:21:30

R topics documented:

enron .................................................. 1
getGoogleLanguages ..................................... 3
getMicrosoftLanguages .................................... 3
translate ............................................. 4

Index 7

enron

Small subset of Enron email corpus

Description

This data set was constructed from a very small subset of the Enron email corpus (Klimt & Yang, 2004). A large set of email messages was made public during the legal investigation concerning the Enron corporation. The full corpus contained 619,446 emails from 158 users. This data set contains only ten emails and includes the body of the email, the email’s subject line, and the date.
Usage

data(enron)

Format

A data frame with 10 observations on the following 3 variables.

email  A character vector of the email’s body.
date   The email’s timestamp as a 'Date' type.
subject A character vector containing the email’s subject line.

Source


Examples

## Not run:
# Load example data. Three columns, the text  
# content ('email') and two metadata  
# fields (date and subject)

data(enron)

# Google, translate column in dataset
google.dataset.out <- translate(dataset = enron,  
    content.field = 'email',  
    google.api.key = my.api.key,  
    source.lang = 'en',  
    target.lang = 'de')

# Google, translate vector
google.vector.out <- translate(content.vec = enron$email,  
    google.api.key = my.api.key,  
    source.lang = 'en',  
    target.lang = 'de')

# Microsoft, translate column in dataset
google.dataset.out <- translate(dataset = enron,  
    content.field = 'email',  
    microsoft.client.id = my.client.id,  
    microsoft.client.secret =  
        my.client.secret,  
    source.lang = 'en',  
    target.lang = 'de')

# Microsoft, translate vector
google.vector.out <- translate(content.vec = enron$email,  
    microsoft.client.id = my.client.id,  
    microsoft.client.secret =  
        my.client.secret,
getGoogleLanguages

Print Google Language Codes

Description

This function prints the valid language Google language codes for use with the translate() function.

Usage

getGoogleLanguages()

Examples

# print valid language codes
googleLanguages()

googleMicrosoftLanguages

Print Microsoft Language Codes

Description

This function prints the valid language Microsoft language codes for use with the translate() function.

Usage

getMicrosoftLanguages()

Examples

# print valid language codes
googleMicrosoftLanguages()
translate

*Translate text with the Google or the Microsoft translation APIs.*

**Description**

This function provides easy access to the Google and Microsoft Translation APIs via R. It can translate any language supported by the APIs (to see a list of the available languages, see the `getGoogleLanguages()` and `getMicrosoftLanguages()` functions). Text can be provided as either a column in a dataframe or as a single vector of text, where the elements are the documents to be translated. Translated text is returned in the format it was provided. If text is provided as a single vector, translate() returns a single vector of translated text. If a dataframe is provided, the user must specify which column contains the text that is to be translated. Translated text is then bound to the dataframe in a new column named "translatedContent" and the entire dataframe is returned. The user must provide either a dataset and the content.field (column name) that contains the text to be translated, or a content.vec (a character vector) where the elements are the text to be translated.

**Usage**

```r
translate(dataset = NULL, 
          content.field = NULL, 
          content.vec = NULL, 
          google.api.key = NULL, 
          microsoft.client.id = NULL, 
          microsoft.client.secret = NULL, 
          source.lang = NULL, 
          target.lang = NULL)
```

**Arguments**

- **dataset** A dataframe with a column containing the text to be translated.
- **content.field** If a dataframe is passed to "dataset", the name of the column containing the text must be passed to "content.field".
- **content.vec** A character vector of text. This is an alternative to "dataset"/"content.field".
- **google.api.key** To use the Google API, an API key must be provided. For more information on getting your key, see https://developers.google.com/translate/v2/getting_started.
- **microsoft.client.id** To use the Microsoft API, a client id and a client secret value must be provided. For more information on getting these, see http://msdn.microsoft.com/en-us/library/hh454950.aspx. NOTE: you do not need to obtain an access token. translateR will retrieve a token internally.
- **microsoft.client.secret** The client secret value is a unique identifying string obtained when registering with Microsoft (see the link for more information). NOTE: you do not need to obtain an access token. translateR will retrieve a token internally.
source.lang The language code that corresponds with the language in which the source text is written. The translators use different language codes, so select accordingly. To see a list of language codes, enter getGoogleLanguages() or getMicrosoftLanguages() for Google or Microsoft, respectively.

target.lang The language code that corresponds with the language into which the source text is to be translated. The translators use different language codes, so select accordingly. To see a list of language codes, enter getGoogleLanguages() or getMicrosoftLanguages() for Google or Microsoft, respectively.

Value

translate returns a dataframe if it is passed a dataframe and a character vector if it is passed a character vector.

Examples

## Not run:
# Load example data. Three columns, the text
# content ('email') and two metadata
# fields (date and subject)
data(enron)

google.dataset.out <- translate(dataset = enron,
    content.field = 'email',
    google.api.key = my.api.key,
    source.lang = 'en',
    target.lang = 'de')

# Google, translate vector
google.vector.out <- translate(content.vec = enron$email,
    google.api.key = my.api.key,
    source.lang = 'en',
    target.lang = 'de')

# Microsoft, translate column in dataset
google.dataset.out <- translate(dataset = enron,
    content.field = 'email',
    microsoft.client.id = my.client.id,
    microsoft.client.secret =
        my.client.secret,
    source.lang = 'en',
    target.lang = 'de')

# Microsoft, translate vector
google.vector.out <- translate(content.vec = enron$email,
    microsoft.client.id = my.client.id,
    microsoft.client.secret =
        my.client.secret,
    source.lang = 'en',
    target.lang = 'de')
## End (Not run)
Index

*Topic datasets
  enron, 1

enron, 1

getGoogleLanguages, 3
getMicrosoftLanguages, 3

translate, 4