Translator Tool using Python

Created by, Abhishek Sharma [Date : 16.10.2020]

In [37]:

```
from IPython.display import Image
Image ("https://specials-images.forbesimg.com/imageserve/1183427805/960x0.jpg")
```

Out[37]:

Importing the Library functions which we have to use to implement the Translator

In [2]:

```
from googletrans import Translator, constants
from pprint import pprint
```

In [3]:

```
translator = Translator()
```

Indonesia > English (By default)

In [4]:

```
# translate a Indonesia text to english text (by default)
translation = translator.translate("Selamat siang")
print(f"{translation.origin} ({translation.src}) --> {translation.text} ({translation.dest})")
```

```
Selamat siang (id) --> Good afternoon (en)
```

Indonesia > Arabic (By default)

In [5]:

```
# translate Indonesian text to arabic for instance
translation = translator.translate("Selamat siang", dest="ar")
print(f"{translation.origin} ({translation.src}) --> {translation.text} ({translation.dest})")

```

```
Selamat siang (id) --> طاب مسائك (ar)
```

specify source language

In [38]:

```
translation = translator.translate("Bagaimana kabarmu hari ini ?", src="id")
print(f"{translation.origin} ({translation.src}) --> {translation.text} ({translation.dest})")
```

```
Bagaimana kabarmu hari ini ? (id) --> How are you today ? (en)
```

In [7]:

```python
# print all translations and other data
pprint(translation.extra_data)
```

```
{'all-translations': None,
 'confidence': 1.0,
 'definitions': None,
 'examples': None,
 'language': [['id'], None, [1.0], ['id']],
 'original-language': 'id',
 'possible-mistakes': None,
 'possible-translations': [['Bagaimana kabarmu hari ini ?',
                            None,
                            [['How are you today ?', 1000, True, False],
                             ['How are you today?', 0, True, False]],
                            [[0, 28]],
                            'Bagaimana kabarmu hari ini ?',
                            0,
                            0]],
 'see-also': None,
 'synonyms': None,
 'translation': [['How are you today ?',
                  'Bagaimana kabarmu hari ini ?',
                  None,
                  None,
                  1]]}
```

Translate more than a phrase :

Indonesia > English

In [8]:

```python
# translate more than a phrase
## (Indonesian to English)

sentences = [
    "Halo Semua",
    "Apa kabar ?",
    "Apakah kamu bisa berbahsa Indonesia ?",
    "oke bagus!",
    "Senang berkenalan denganmu"
]
translations = translator.translate(sentences, dest="en")
for translation in translations:
    print(f"{translation.origin} ({translation.src}) --> {translation.text} ({translation.dest})")
```

```
Halo Semua (id) --> Hello, all (en)
Apa kabar ? (id) --> How are you ? (en)
Apakah kamu bisa berbahsa Indonesia ? (id) --> Can you speak Indonesian ? (en)
oke bagus! (id) --> OK good! (en)
Senang berkenalan denganmu (id) --> Nice to meet you (en)
```

Indonesian > Japanese

In [9]:

```python
# translate more than a phrase
## (Indonesian to Japanese)

sentences = [
    "Halo Semua",
    "Apa kabar ?",
    "Apakah kamu bisa berbahsa Indonesia ?",
    "oke bagus!",
    "Senang berkenalan denganmu"
]
translations = translator.translate(sentences, dest="ja")
for translation in translations:
    print(f"{translation.origin} ({translation.src}) --> {translation.text} ({translation.dest})")
```

```
Halo Semua (id) --> こんにちは皆さん (ja)
Apa kabar ? (id) --> お元気ですか ？ (ja)
Apakah kamu bisa berbahsa Indonesia ? (id) --> インドネシア語を話せますか？ (ja)
oke bagus! (id) --> ようし。いいぞ！ (ja)
Senang berkenalan denganmu (id) --> はじめまして (ja)
```

Indonesian > Azerbaijan

In [10]:

```python
# translate more than a phrase
#(Indonesian to Azerbaijan)
sentences = [
    "Halo Semua",
    "Apa kabar ?",
    "Apakah kamu bisa berbahsa Indonesia ?",
    "oke bagus!",
    "Senang berkenalan denganmu"
]
translations = translator.translate(sentences, dest="az")
for translation in translations:
    print(f"{translation.origin} ({translation.src}) --> {translation.text} ({translation.dest})")
```

```
Halo Semua (id) --> Salam, hamısı (az)
Apa kabar ? (id) --> Necəsən ? (az)
Apakah kamu bisa berbahsa Indonesia ? (id) --> İndoneziya dilində danışa bilərsən? (az)
oke bagus! (id) --> Yaxşı, oldu! (az)
Senang berkenalan denganmu (id) --> Tanış olmaqdan məmnunam (az)
```

Indonesian > Malay

In [11]:

```python
# translate more than a phrase
#(Indonesian to Malay)

sentences = [
    "Halo Semua",
    "Apa kabar ?",
    "Apakah kamu bisa berbahsa Indonesia ?",
    "oke bagus!",
    "Senang berkenalan denganmu"
]
translations = translator.translate(sentences, dest="ms")
for translation in translations:
    print(f"{translation.origin} ({translation.src}) --> {translation.text} ({translation.dest})")
```

```
Halo Semua (ms) --> Halo Semua (ms)
Apa kabar ? (ms) --> Apa kabar ? (ms)
Apakah kamu bisa berbahsa Indonesia ? (id) --> Bolehkah anda berbahasa Indonesia? (ms)
oke bagus! (id) --> Ok baik! (ms)
Senang berkenalan denganmu (ms) --> Senang berkenalan denganmu (ms)
```

Indoneisan > Turkish

In [12]:

```python
# translate more than a phrase
#(Indonesian to Turkish)

sentences = [
    "Halo Semua",
    "Apa kabar ?",
    "Apakah kamu bisa berbahsa Indonesia ?",
    "oke bagus!",
    "Senang berkenalan denganmu"
]
translations = translator.translate(sentences, dest="tr")
for translation in translations:
    print(f"{translation.origin} ({translation.src}) --> {translation.text} ({translation.dest})")
```

```
Halo Semua (id) --> Hepinize merhaba (tr)
Apa kabar ? (id) --> Nasılsın ? (tr)
Apakah kamu bisa berbahsa Indonesia ? (id) --> Endonezce konuşabiliyor musun? (tr)
oke bagus! (id) --> Tamam iyi! (tr)
Senang berkenalan denganmu (id) --> Tanıştığıma memnun oldum (tr)
```

Indonesian to Bengali

In [19]:

```
# translate more than a phrase
#(Indonesian to Bengali)

sentences = [
    "Halo Semua",
    "Apa kabar ?",
    "Apakah kamu bisa berbahsa Indonesia ?",
    "oke bagus!",
    "Senang berkenalan denganmu"
]
translations = translator.translate(sentences, dest="bn")
for translation in translations:
    print(f"{translation.origin} ({translation.src}) --> {translation.text} ({translation.dest})")
```

```
Halo Semua (id) --> ওহে সবাই (bn)
Apa kabar ? (id) --> আপনি কেমন আছেন ? (bn)
Apakah kamu bisa berbahsa Indonesia ? (id) --> আপনি Indonesian কথা বলতে পারেন ? (bn)
oke bagus! (id) --> ঠিক আছে ভালো! (bn)
Senang berkenalan denganmu (id) --> তোমার সাথে দেখা করে ভালো লাগলো (bn)
```

Indonesian > Hindi

In [13]:

```
# translate more than a phrase
#(Indonesian to Hindi)

sentences = [
    "Halo Semua",
    "Apa kabar ?",
    "Apakah kamu bisa berbahsa Indonesia ?",
    "oke bagus!",
    "Senang berkenalan denganmu"
]
translations = translator.translate(sentences, dest="hi")
for translation in translations:
    print(f"{translation.origin} ({translation.src}) --> {translation.text} ({translation.dest})")
```

```
Halo Semua (id) --> हैलो सभी को (hi)
Apa kabar ? (id) --> क्या हाल है ? (hi)
Apakah kamu bisa berbahsa Indonesia ? (id) --> क्या आप ईन्डोनेशियन बोल सकते हैं ? (hi)
oke bagus! (id) --> ठीक अच्छा! (hi)
Senang berkenalan denganmu (id) --> आपसे मिलकर अच्छा लगा (hi)
```

Detect a Language :

A. Checking of Hindi Language and also check the confidence level of the AI

In [39]:

```
# detect a language
detection = translator.detect("नमस्ते दुनिया")
print("Language code:", detection.lang)
print("Confidence:", detection.confidence)
```

```
Language code: hi
Confidence: 1.0
```

In [15]:

```
# detect a language
detection = translator.detect("kamu lagi apa ?")
print("Language code:", detection.lang)
print("Confidence:", detection.confidence)
```

```
Language code: id
Confidence: 0.685635
```

In [16]:

```
# detect a language
detection = translator.detect("jam berapa kamu pergi ke sekolah  ?")
print("Language code:", detection.lang)
print("Confidence:", detection.confidence)
```

```
Language code: idms
Confidence: 0.6471428
```

In [17]:

```
# detect a language
detection = translator.detect("sudah makan ?")
print("Language code:", detection.lang)
print("Confidence:", detection.confidence)
```

```
Language code: id
Confidence: 0.741919
```

B. Print all the languages and their respective codes :

In [18]:

```
# print all available languages
print("Total supported languages:", len(constants.LANGUAGES))
print("Languages:")
pprint(constants.LANGUAGES)
```

```
Total supported languages: 107
Languages:
{'af': 'afrikaans',
 'am': 'amharic',
 'ar': 'arabic',
 'az': 'azerbaijani',
 'be': 'belarusian',
 'bg': 'bulgarian',
 'bn': 'bengali',
 'bs': 'bosnian',
 'ca': 'catalan',
 'ceb': 'cebuano',
 'co': 'corsican',
 'cs': 'czech',
 'cy': 'welsh',
 'da': 'danish',
 'de': 'german',
 'el': 'greek',
 'en': 'english',
 'eo': 'esperanto',
 'es': 'spanish',
 'et': 'estonian',
 'eu': 'basque',
 'fa': 'persian',
 'fi': 'finnish',
 'fr': 'french',
 'fy': 'frisian',
 'ga': 'irish',
 'gd': 'scots gaelic',
 'gl': 'galician',
 'gu': 'gujarati',
 'ha': 'hausa',
 'haw': 'hawaiian',
 'he': 'hebrew',
 'hi': 'hindi',
 'hmn': 'hmong',
 'hr': 'croatian',
 'ht': 'haitian creole',
 'hu': 'hungarian',
 'hy': 'armenian',
 'id': 'indonesian',
 'ig': 'igbo',
 'is': 'icelandic',
 'it': 'italian',
 'iw': 'hebrew',
 'ja': 'japanese',
 'jw': 'javanese',
 'ka': 'georgian',
 'kk': 'kazakh',
 'km': 'khmer',
 'kn': 'kannada',
 'ko': 'korean',
 'ku': 'kurdish (kurmanji)',
 'ky': 'kyrgyz',
 'la': 'latin',
 'lb': 'luxembourgish',
 'lo': 'lao',
 'lt': 'lithuanian',
 'lv': 'latvian',
 'mg': 'malagasy',
 'mi': 'maori',
 'mk': 'macedonian',
 'ml': 'malayalam',
 'mn': 'mongolian',
 'mr': 'marathi',
 'ms': 'malay',
 'mt': 'maltese',
 'my': 'myanmar (burmese)',
 'ne': 'nepali',
 'nl': 'dutch',
 'no': 'norwegian',
 'ny': 'chichewa',
 'or': 'odia',
 'pa': 'punjabi',
 'pl': 'polish',
 'ps': 'pashto',
 'pt': 'portuguese',
 'ro': 'romanian',
 'ru': 'russian',
 'sd': 'sindhi',
 'si': 'sinhala',
 'sk': 'slovak',
 'sl': 'slovenian',
 'sm': 'samoan',
 'sn': 'shona',
 'so': 'somali',
 'sq': 'albanian',
 'sr': 'serbian',
 'st': 'sesotho',
 'su': 'sundanese',
 'sv': 'swedish',
 'sw': 'swahili',
 'ta': 'tamil',
 'te': 'telugu',
 'tg': 'tajik',
 'th': 'thai',
 'tl': 'filipino',
 'tr': 'turkish',
 'ug': 'uyghur',
 'uk': 'ukrainian',
 'ur': 'urdu',
 'uz': 'uzbek',
 'vi': 'vietnamese',
 'xh': 'xhosa',
 'yi': 'yiddish',
 'yo': 'yoruba',
 'zh-cn': 'chinese (simplified)',
 'zh-tw': 'chinese (traditional)',
 'zu': 'zulu'}
```

The Automated version of Google Translator 2.0

In [33]:

```python
# Creating the dummy version of Google Translate
print ("Welcome to the translator! Please follow the steps...\n\n")
while (1>0):
    s = input ("Enter the Text : ")
    change = input ("Enter in which language do you want to translate (code) : ")
    translation = translator.translate(s, dest=change)
    print(f"{translation.origin} ({translation.src}) --> {translation.text} ({translation.dest})")
    check = input("\nDo you wanna continue it [Y/N] ? ")
    if (check == 'N' or check == 'n'):
        print ("\n\n::Thank you for using the translator! Have a nice day ahead!!::")
        break

```

```
Welcome to the translator! Please follow the steps...


Enter the Text : Engineer
Enter in which language do you want to translate (code) : ja
Engineer (en) --> エンジニア (ja)

Do you wanna continue it [Y/N] ? N


::Thank you for using the translator! Have a nice day ahead!!::
```

In [42]:

```python
Image ("https://cdn.images.express.co.uk/img/dynamic/59/750x445/976492.jpg")
```

Out[42]:

Limitations :

Google Translate, like other automatic translation tools, has its limitations. The service limits the number of paragraphs and the range of technical terms that can be translated, and while it can help the reader understand the general content of a foreign language text, it does not always deliver accurate translations, and most times it tends to repeat verbatim the same word it's expected to translate. Grammatically, for example, Google Translate struggles to differentiate between imperfect and perfect aspects in Romance languages so habitual and continuous acts in the past often become single historical events. Although seemingly pedantic, this can often lead to incorrect results (to a native speaker of for example French and Spanish) which would have been avoided by a human translator. Knowledge of the subjunctive mood is virtually non-existent. Moreover, the formal second person (vous) is often chosen, whatever the context or accepted usage. Since its English reference material contains only "you" forms, it has difficulty translating a language with "you all" or formal "you" variations.

Due to differences between languages in investment, research, and the extent of digital resources, the accuracy of Google Translate varies greatly among languages. Some languages produce better results than others. Most languages from Africa, Asia, and the Pacific, tend to score poorly in relation to the scores of many well-financed European languages, with Afrikaans and Chinese being the high-scoring exceptions from their continents. No languages indigenous to Australia or the Americas are included within Google Translate. Higher scores for European can be partially attributed to the Europarl Corpus, a trove of documents from the European Parliament that have been professionally translated by the mandate of the European Union into as many as 21 languages. A 2010 analysis indicated that French to English translation is relatively accurate, and 2011 and 2012 analyses showed that Italian to English translation is relatively accurate as well. However, if the source text is shorter, rule-based machine translations often perform better; this effect is particularly evident in Chinese to English translations. While edits of translations may be submitted, in Chinese specifically one cannot edit sentences as a whole. Instead, one must edit sometimes arbitrary sets of characters, leading to incorrect edits A good example is Russian-to-English. Formerly one would use Google Translate to make a draft and then use a dictionary and common sense to correct the numerous mistakes. As of early 2018 Translate is sufficiently accurate to make the Russian Wikipedia accessible to those who can read English. The quality of Translate can be checked by adding it as an extension to Chrome or Firefox and applying it to the left language links of any Wikipedia article. It can be used as a dictionary by typing in words. One can translate from a book by using a scanner and an OCR like Google Drive, but this takes about five minutes per page.

In its Written Words Translation function, there is a word limit on the amount of text that can be translated at once. Therefore, long text should be transferred to a document form and translated through its Document Translate function.

Moreover, like all machine translation programs, Google Translate struggles with polysemy (the multiple meanings a word may have) and multiword expressions (terms that have meanings that cannot be understood or translated by analyzing the individual word units that compose them). A word in a foreign language might have two different meanings in the translated language. This might lead to mistranslations.

Additionally, grammatical errors remain a major limitation to the accuracy of Google Translate.

Thank You! Stay Safe! :)