Text Analytics Demo: https://aidemos.microsoft.com/text-analytics (https://aidemos.microsoft.com/text-analytics)

Pricing: https://azure.microsoft.com/en-gb/pricing/details/cognitive-services/text-analytics/#pricing (https://azure.microsoft.com/en-gb/pricing/details/cognitive-services/text-analytics/#pricing)

In [1]:

```
import os
os.path.abspath(os.getcwd())
```

Out[1]:

```
'C:\\Users\\Ingopn01\\OneDrive - Ingram Micro\\Pictures\\Python\\2021\\Azur
e'
```

In [9]:

```python
import os

# Read the reviews in the /data/reviews folder
reviews_folder = os.path.join('data', 'text', 'reviews')

# Create a collection of reviews with id (file name) and text (contents) properties
reviews = []
for file_name in os.listdir(reviews_folder):
    review_text = open(os.path.join(reviews_folder, file_name),encoding="utf8").read()
    review = {"id": file_name, "text": review_text}
    reviews.append(review)

for review_num in range(len(reviews)):
    # print the review text
    print('{}\n{}\n'.format(reviews[review_num]['id'], reviews[review_num]['text']))
```

```
Review1.txt
Good Hotel and staff
The Royal Hotel, London, UK
3/2/2018
Clean rooms, good service, great location near Buckingham Palace and Westmin
ster Abbey, and so on. We thoroughly enjoyed our stay. The courtyard is very
peaceful and we went to a restaurant which is part of the same group and is
Indian ( West coast so plenty of fish) with a Michelin Star. We had the tast
er menu which was fabulous. The rooms were very well appointed with a kitche
n, lounge, bedroom and enormous bathroom. Thoroughly recommended.

Review2.txt
Tired hotel with poor service
The Royal Hotel, London, United Kingdom
5/6/2018
This is a old hotel (has been around since 1950's) and the room furnishings
are average - becoming a bit old now and require changing. The internet did
n't work and had to come to one of their office rooms to check in for my fli
ght home. The website says it's close to the British Museum, but it's too fa
r to walk.

Review3.txt
Tired hotel with poor service
The Royal Hotel, London, United Kingdom
5/6/2018
This is a old hotel (has been around since 1950's) and the room furnishings
are average - becoming a bit old now and require changing. The internet did
n't work and had to come to one of their office rooms to check in for my fli
ght home. The website says it's close to the British Museum, but it's too fa
r to walk.

Review4.txt
Very noisy and rooms are tiny
The Lombard Hotel, San Francisco, USA
9/5/2018
Hotel is located on Lombard street which is a very busy SIX lane street dire
ctly off the Golden Gate Bridge. Traffic from early morning until late at ni
ght especially on weekends. Noise would not be so bad if rooms were better i
nsulated but they are not. Had to put cotton balls in my ears to be able to
sleep--was too tired to enjoy the city the next day. Rooms are TINY. I picke
d the room because it had two queen size beds--but the room barely had space
to fit them. With family of four in the room it was tight. With all that sai
d, rooms are clean and they've made an effort to update them. The hotel is i
```

```
n Marina district with lots of good places to eat, within walking distance t
o Presidio. May be good hotel for young stay-up-late adults on a budget


Review6.txt
Saturn is the sixth planet from the Sun and the second-largest in the Solar
System, after Jupiter. It is a gas giant with an average radius about nine t
imes that of Earth.
```

#Review 5 :Ganeshan loved cricket. After attending the English subject class when he rushed to the playground, due to the distance of playground he was always late. His old friends persuaded him to miss his class for next two days so that he could practice for an important match. His coach was also agree to help him by speaking to his class teacher, for this. But the coach ditched him and Ganeshan was taken to task for his absence of two days without prior school permission.

In [1]:

```
cog_key = 'Your Key'
cog_endpoint = 'https://nlpsession.cognitiveservices.azure.com/'

print('Ready to use cognitive services at {} using key {}'.format(cog_endpoint, cog_key))
```

```
Ready to use cognitive services at https://nlpsession.cognitiveservices.azur
e.com/ (https://nlpsession.cognitiveservices.azure.com/) using key Your Key
```

In [4]:

```
!pip install azure-ai-textanalytics
!pip install azure-cognitiveservices-language-textanalytics
```

```
Requirement already satisfied: azure-ai-textanalytics in d:\anaconda\anacond
a\lib\site-packages (5.1.0)
Requirement already satisfied: six>=1.11.0 in d:\anaconda\anaconda\lib\site-
packages (from azure-ai-textanalytics) (1.15.0)
Requirement already satisfied: azure-common~=1.1 in d:\anaconda\anaconda\lib
\site-packages (from azure-ai-textanalytics) (1.1.27)
Requirement already satisfied: msrest>=0.6.21 in d:\anaconda\anaconda\lib\si
te-packages (from azure-ai-textanalytics) (0.6.21)
Requirement already satisfied: azure-core<2.0.0,>=1.14.0 in d:\anaconda\anac
onda\lib\site-packages (from azure-ai-textanalytics) (1.17.0)
Requirement already satisfied: certifi>=2017.4.17 in d:\anaconda\anaconda\li
b\site-packages (from msrest>=0.6.21->azure-ai-textanalytics) (2020.6.20)
Requirement already satisfied: requests-oauthlib>=0.5.0 in d:\anaconda\anaco
nda\lib\site-packages (from msrest>=0.6.21->azure-ai-textanalytics) (1.3.0)
Requirement already satisfied: requests~=2.16 in d:\anaconda\anaconda\lib\si
te-packages (from msrest>=0.6.21->azure-ai-textanalytics) (2.24.0)
Requirement already satisfied: isodate>=0.6.0 in d:\anaconda\anaconda\lib\si
te-packages (from msrest>=0.6.21->azure-ai-textanalytics) (0.6.0)
Requirement already satisfied: oauthlib>=3.0.0 in d:\anaconda\anaconda\lib\s
ite-packages (from requests-oauthlib>=0.5.0->msrest>=0.6.21->azure-ai-textan
alytics) (3.1.1)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
d:\anaconda\anaconda\lib\site-packages (from requests~=2.16->msrest>=0.6.21-
>azure-ai-textanalytics) (1.25.11)
Requirement already satisfied: idna<3,>=2.5 in d:\anaconda\anaconda\lib\site
-packages (from requests~=2.16->msrest>=0.6.21->azure-ai-textanalytics) (2.1
0)
Requirement already satisfied: chardet<4,>=3.0.2 in d:\anaconda\anaconda\lib
\site-packages (from requests~=2.16->msrest>=0.6.21->azure-ai-textanalytics)
(3.0.4)
Requirement already satisfied: azure-cognitiveservices-language-textanalytic
s in d:\anaconda\anaconda\lib\site-packages (0.2.0)
Requirement already satisfied: azure-common~=1.1 in d:\anaconda\anaconda\lib
\site-packages (from azure-cognitiveservices-language-textanalytics) (1.1.2
7)
Requirement already satisfied: msrest>=0.5.0 in d:\anaconda\anaconda\lib\sit
e-packages (from azure-cognitiveservices-language-textanalytics) (0.6.21)
Requirement already satisfied: requests-oauthlib>=0.5.0 in d:\anaconda\anaco
nda\lib\site-packages (from msrest>=0.5.0->azure-cognitiveservices-language-
textanalytics) (1.3.0)
Requirement already satisfied: requests~=2.16 in d:\anaconda\anaconda\lib\si
te-packages (from msrest>=0.5.0->azure-cognitiveservices-language-textanalyt
ics) (2.24.0)
Requirement already satisfied: isodate>=0.6.0 in d:\anaconda\anaconda\lib\si
te-packages (from msrest>=0.5.0->azure-cognitiveservices-language-textanalyt
ics) (0.6.0)
Requirement already satisfied: certifi>=2017.4.17 in d:\anaconda\anaconda\li
b\site-packages (from msrest>=0.5.0->azure-cognitiveservices-language-textan
alytics) (2020.6.20)
Requirement already satisfied: oauthlib>=3.0.0 in d:\anaconda\anaconda\lib\s
ite-packages (from requests-oauthlib>=0.5.0->msrest>=0.5.0->azure-cognitives
ervices-language-textanalytics) (3.1.1)
Requirement already satisfied: idna<3,>=2.5 in d:\anaconda\anaconda\lib\site
-packages (from requests~=2.16->msrest>=0.5.0->azure-cognitiveservices-langu
age-textanalytics) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in d:\anaconda\anaconda\lib
```

```
\site-packages (from requests~=2.16->msrest>=0.5.0->azure-cognitiveservices-
language-textanalytics) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
d:\anaconda\anaconda\lib\site-packages (from requests~=2.16->msrest>=0.5.0->
azure-cognitiveservices-language-textanalytics) (1.25.11)
Requirement already satisfied: six in d:\anaconda\anaconda\lib\site-packages
(from isodate>=0.6.0->msrest>=0.5.0->azure-cognitiveservices-language-textan
alytics) (1.15.0)
```

Language Detection

In [11]:

```
import os
from azure.cognitiveservices.language.textanalytics import TextAnalyticsClient
from msrest.authentication import CognitiveServicesCredentials

# Get a client for your text analytics cognitive service resource
text_analytics_client = TextAnalyticsClient(endpoint=cog_endpoint,
                                            credentials=CognitiveServicesCredentials(cog_ke

# Analyze the reviews you read from the /data/reviews folder earlier
language_analysis = text_analytics_client.detect_language(documents=reviews)

# print detected language details for each review
for review_num in range(len(reviews)):
    # print the review id
    print(reviews[review_num]['id'])

    # Get the language details for this review
    lang = language_analysis.documents[review_num].detected_languages[0]
    print(' - Language: {}\n - Code: {}\n - Score: {}\n'.format(lang.name, lang.iso6391_nam

    # Add the detected language code to the collection of reviews (so we can do further ana
    reviews[review_num]["language"] = lang.iso6391_name
```

```
Review1.txt
 - Language: English
 - Code: en
 - Score: 1.0

Review2.txt
 - Language: English
 - Code: en
 - Score: 1.0

Review3.txt
 - Language: English
 - Code: en
 - Score: 1.0

Review4.txt
 - Language: English
 - Code: en
 - Score: 1.0

Review6.txt
 - Language: English
 - Code: en
 - Score: 1.0
```

Key Phrase extraction

In [12]:

```
# # Use the client and reviews you created in the previous code cell to get key phrases
key_phrase_analysis = text_analytics_client.key_phrases(documents=reviews)

# print key phrases for each review
for review_num in range(len(reviews)):
    # print the review id
    print(reviews[review_num]['id'])

    # Get the key phrases in this review
    print('\nKey Phrases:')
    key_phrases = key_phrase_analysis.documents[review_num].key_phrases
    # Print each key phrase
    for key_phrase in key_phrases:
        print('\t', key_phrase)
    print('\n')
```

```
Review1.txt

Key Phrases:
         Good Hotel
         good service
         Clean rooms
         Royal Hotel
         great location
         Buckingham Palace
         Westminster Abbey
         fish
         West coast
         lounge
         bedroom
         enormous bathroom
         group
         kitchen
         London
         UK
         taster menu
         Michelin Star
         staff
         courtyard


Review2.txt

Key Phrases:
         old hotel
         Royal Hotel
         Tired hotel
         London
         United Kingdom
         room furnishings
         poor service
         British Museum
         website
         office rooms
         flight home
         internet
```

```
Review3.txt

Key Phrases:
         old hotel
         Royal Hotel
         Tired hotel
         London
         United Kingdom
         room furnishings
         poor service
         British Museum
         website
         office rooms
         flight home
         internet


Review4.txt

Key Phrases:
         rooms
         good hotel
         Lombard Hotel
         Lombard street
         late adults
         good places
         lane street
         young stay
         night
         early morning
         Marina district
         San Francisco
         USA
         Golden Gate Bridge
         walking distance
         queen size beds
         ears
         Traffic
         cotton balls
         city
         Presidio
         weekends
         budget
         day
         effort
         Noise
         space
         family


Review6.txt

Key Phrases:
         average radius
         Solar System
         times
         gas giant
         Jupiter
         Earth
         Saturn
         planet
```

https://docs.microsoft.com/en-us/azure/cognitive-services/text-analytics/how-tos/text-analytics-how-to-sentiment-analysis?tabs=version-3-1 (https://docs.microsoft.com/en-us/azure/cognitive-services/text-analytics/how-tos/text-analytics-how-to-sentiment-analysis?tabs=version-3-1)

Sentiment Analysis

In [13]:

```
# Use the client and reviews you created previously to get sentiment scores
sentiment_analysis = text_analytics_client.sentiment(documents=reviews)

# Print the results for each review
for review_num in range(len(reviews)):

    # Get the sentiment score for this review
    sentiment_score = sentiment_analysis.documents[review_num].score

    # classifiy 'positive' if more than 0.5,
    if sentiment_score < 0.5:
        sentiment = 'negative'
    else:
        sentiment = 'positive'

    # print file name and sentiment
    print('{} : {} ({})'.format(reviews[review_num]['id'], sentiment, sentiment_score))
```

```
Review1.txt : positive (0.9999973773956299)
Review2.txt : negative (5.662441253662109e-07)
Review3.txt : negative (5.662441253662109e-07)
Review4.txt : negative (2.0623207092285156e-05)
Review6.txt : positive (0.5)
```

Entity Extraction

In [14]:

```
# Use the client and reviews you created previously to get named entities
entity_analysis = text_analytics_client.entities(documents=reviews)

# Print the results for each review
for review_num in range(len(reviews)):
    print(reviews[review_num]['id'])
    # Get the named entitites in this review
    entities = entity_analysis.documents[review_num].entities
    for entity in entities:
        # Only print datetime or location entitites
        if entity.type in ['DateTime','Location']:
            link = '(' + entity.wikipedia_url + ')' if entity.wikipedia_id is not None else
            print(' - {}: {} {}'.format(entity.type, entity.name, link))
```

```
Review1.txt
 - Location: The Royal Hotel (https://en.wikipedia.org/wiki/The_Royal_Hotel)
 - Location: London (https://en.wikipedia.org/wiki/London)
 - DateTime: 3/2/2018
 - Location: Buckingham Palace (https://en.wikipedia.org/wiki/Buckingham_Pal
ace)
 - Location: Westminster Abbey (https://en.wikipedia.org/wiki/Westminster_Ab
bey)
 - Location: India (https://en.wikipedia.org/wiki/India)
 - Location: West Coast Main Line (https://en.wikipedia.org/wiki/West_Coast_
Main_Line)
Review2.txt
 - Location: The Royal Hotel (https://en.wikipedia.org/wiki/The_Royal_Hotel)
 - Location: London (https://en.wikipedia.org/wiki/London)
 - Location: London
 - Location: United Kingdom
 - DateTime: 5/6/2018
 - DateTime: since 1950's
 - DateTime: now
 - Location: British Museum (https://en.wikipedia.org/wiki/British_Museum)
Review3.txt
 - Location: The Royal Hotel (https://en.wikipedia.org/wiki/The_Royal_Hotel)
 - Location: London (https://en.wikipedia.org/wiki/London)
 - Location: London
 - Location: United Kingdom
 - DateTime: 5/6/2018
 - DateTime: since 1950's
 - DateTime: now
 - Location: British Museum (https://en.wikipedia.org/wiki/British_Museum)
Review4.txt
 - Location: Lombard, Illinois (https://en.wikipedia.org/wiki/Lombard,_Illin
ois)
 - Location: San Francisco (https://en.wikipedia.org/wiki/San_Francisco)
 - Location: Lombard Street (San Francisco) (https://en.wikipedia.org/wiki/L
ombard_Street_(San_Francisco))
 - Location: Lombard
 - Location: Golden Gate Bridge (https://en.wikipedia.org/wiki/Golden_Gate_B
ridge)
 - DateTime: from early morning
 - DateTime: night
 - DateTime: the next day
 - Location: Marina
 - Location: Marina District, San Francisco (https://en.wikipedia.org/wiki/M
arina_District,_San_Francisco)
 - Location: Presidio of San Francisco (https://en.wikipedia.org/wiki/Presid
io_of_San_Francisco)
Review6.txt
 - Location: Jupiter
 - Location: Earth
```

Language Understanding (LUIS)

A natural language understanding (NLU) AI service that allows users to interact with your applications, bots and IoT devices by using natural language.

Language Detection Demo: https://aidemos.microsoft.com/luis/demo (https://aidemos.microsoft.com/luis/demo)

LUIS: https://azure.microsoft.com/en-in/services/cognitive-services/language-understanding-intelligent-service/#overview (https://azure.microsoft.com/en-in/services/cognitive-services/language-understanding-intelligent-service/#overview)

Translating Text

https://docs.microsoft.com/en-us/azure/cognitive-services/translator/language-support (https://docs.microsoft.com/en-us/azure/cognitive-services/translator/language-support)

In [2]:

```
cog_key = 'Your Key'
cog_location = 'centralindia'

print('Ready to use cognitive services in {} using key {}'.format(cog_location, cog_key))
```

```
Ready to use cognitive services in centralindia using key Your Key
```

In [19]:

```
# Create a function that makes a REST request to the Text Translation service
def translate_text(cog_location, cog_key, text, to_lang='fr', from_lang='en'):
    import requests, uuid, json

    # Create the URL for the Text Translator service REST request
    path = 'https://api.cognitive.microsofttranslator.com/translate?api-version=3.0'
    params = '&from={}&to={}'.format(from_lang, to_lang)
    constructed_url = path + params

    # Prepare the request headers with Cognitive Services resource key and region
    headers = {
        'Ocp-Apim-Subscription-Key': cog_key,
        'Ocp-Apim-Subscription-Region':cog_location,
        'Content-type': 'application/json',
        'X-ClientTraceId': str(uuid.uuid4())
    }

    # Add the text to be translated to the body
    body = [{
        'text': text
    }]

    # Get the translation
    request = requests.post(constructed_url, headers=headers, json=body)
    response = request.json()
    return response[0]["translations"][0]["text"]


# Test the function
text_to_translate = "Hello"

translation = translate_text(cog_location, cog_key, text_to_translate, to_lang='fr', from_l
print('{} -> {}'.format(text_to_translate,translation))
```

```
Hello -> Bonjour
```

In [20]:

```
text_to_translate = "Hello"

translation = translate_text(cog_location, cog_key, text_to_translate, to_lang='it-IT', fro
print('{} -> {}'.format(text_to_translate,translation))
```

```
Hello -> Ciao
```

In [21]:

```
text_to_translate = "Hello"

translation = translate_text(cog_location, cog_key, text_to_translate, to_lang='zh-CN', fro
print('{} -> {}'.format(text_to_translate,translation))
```

```
Hello -> 你好
```

In [22]:

```
text_to_translate = "गनेशन क्रिकेट का शौकीन था । अंग्रेजी कक्षा के बाद जब वह खेल के मैदान के लिए दौड़ता

translation = translate_text(cog_location, cog_key, text_to_translate, to_lang='en-US', fro
print('{} -> {}'.format(text_to_translate,translation))
```

```
गनेशन क्रिकेट का शौकीन था । अंग्रेजी कक्षा के बाद जब वह खेल के मैदान के लिए दौड़ता था । खे
ल का मैदान दूर होने की वजह उसे हमेशा देर हो जाती थी । उसके पुराने मित्रों ने उसे राजी कर लि
या कि वह अगले दो दिन अपनी कक्षा में ना जाये ताकि वह महत्वपूर्ण मैच का अभ्यास कर ले । उस
के प्रशिक्षक भी उसकी सहायता को सहमत हो गये कि वह कक्षा अध्यापक को बोल देंगे । लेकिन प्र
शिक्षक ने उसे धोखा दिया और गनेशन को बिना पूर्व अनुमति के दो दिन अनुपस्थित रहने के लिए द
ण्ड दिया गया । -> Ganesan was fond of cricket. After the English class when he
ran to the playground. He was always late because the playground was far awa
y. His old friends persuaded him not to go to his class for the next two day
s so that he could practice the important match. His trainers also agreed to
help him speak to the class teacher. But the trainer cheated him and Ganesan
was punished for two days absent without prior permission.
```

Moving Forward: https://gentle-pebble-08cfc3110.azurestaticapps.net/#about (https://gentle-pebble-08cfc3110.azurestaticapps.net/#about)

https://vivekraja98.medium.com/literature-text-translation-audio-synthesis-using-microsoft-azure-cognitive-services-5e35add0c79e (https://vivekraja98.medium.com/literature-text-translation-audio-synthesis-using-microsoft-azure-cognitive-services-5e35add0c79e)

In [ ]: