Text Analytics

In [1]:
```
pip install azure-ai-textanalytics --pre
```

```
Requirement already satisfied: azure-ai-textanalytics in d:\python\lib\site-packages (5.1.0b3)
Requirement already satisfied: six>=1.6 in d:\python\lib\site-packages (from azure-ai-textanalytics) (1.15.0)
Requirement already satisfied: azure-core<2.0.0,>=1.4.0 in d:\python\lib\site-packages (from azure-ai-textanal
ytics) (1.9.0)
Requirement already satisfied: msrest>=0.6.0 in d:\python\lib\site-packages (from azure-ai-textanalytics) (0.
6.19)
Requirement already satisfied: azure-common~=1.1 in d:\python\lib\site-packages (from azure-ai-textanalytics)
(1.1.26)
Requirement already satisfied: requests>=2.18.4 in d:\python\lib\site-packages (from azure-core<2.0.0,>=1.4.0-
>azure-ai-textanalytics) (2.24.0)
Requirement already satisfied: certifi>=2017.4.17 in d:\python\lib\site-packages (from msrest>=0.6.0->azure-ai
-textanalytics) (2020.6.20)
Requirement already satisfied: requests-oauthlib>=0.5.0 in d:\python\lib\site-packages (from msrest>=0.6.0->az
ure-ai-textanalytics) (1.3.0)
Requirement already satisfied: isodate>=0.6.0 in d:\python\lib\site-packages (from msrest>=0.6.0->azure-ai-tex
tanalytics) (0.6.0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in d:\python\lib\site-packages (from re
quests>=2.18.4->azure-core<2.0.0,>=1.4.0->azure-ai-textanalytics) (1.25.9)
Requirement already satisfied: idna<3,>=2.5 in d:\python\lib\site-packages (from requests>=2.18.4->azure-core<
2.0.0,>=1.4.0->azure-ai-textanalytics) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in d:\python\lib\site-packages (from requests>=2.18.4->azure-
core<2.0.0,>=1.4.0->azure-ai-textanalytics) (3.0.4)
Requirement already satisfied: oauthlib>=3.0.0 in d:\python\lib\site-packages (from requests-oauthlib>=0.5.0->
msrest>=0.6.0->azure-ai-textanalytics) (3.1.0)
Note: you may need to restart the kernel to use updated packages.
```

Using SDK

In [1]:
```
key = "5256fa5f05d144f691b6b7e607d21f99"
endpoint = "https://textanalyticsdgdfz.cognitiveservices.azure.com/"
```

In [2]:
```
# Authenticate the client

from azure.ai.textanalytics import TextAnalyticsClient
from azure.core.credentials import AzureKeyCredential

def authenticate_client():
    ta_credential = AzureKeyCredential(key)
    text_analytics_client = TextAnalyticsClient(
            endpoint=endpoint,
            credential=ta_credential)
    return text_analytics_client

client = authenticate_client()
```

Language Detection

In [8]:
```
def language_detection_example(client):
    try:
        documents = ["Ce document est rédigé en Français."]
        response = client.detect_language(documents = documents, country_hint = '')[0]
        print("Language: ", response.primary_language.name)

    except Exception as err:
        print("Encountered exception. {}".format(err))
language_detection_example(client)
```

```
Language:  French
```

Named Entity Recognition

In [19]:
```
def entity_recognition_example(client):

    try:
        documents = ["Smart phone are improved a lot in recent years and Apple tops the most"]
        result = client.recognize_entities(documents = documents)[0]

        print("Named Entities:\n")
        for entity in result.entities:
            print(entity)

    except Exception as err:
        print("Encountered exception. {}".format(err))
entity_recognition_example(client)
```

```
Named Entities:

{'text': 'Smart phone', 'category': 'Product', 'subcategory': None, 'offset': 0, 'confidence_score': 0.53}
{'text': 'Apple', 'category': 'Organization', 'subcategory': None, 'offset': 51, 'confidence_score': 0.47}
```

Key Phrase Extraction

In [21]:
```
def key_phrase_extraction_example(client):

    try:
        documents = ["Smart phone are improved a lot in recent years and Apple tops the most"]
        response = client.extract_key_phrases(documents = documents)[0]

        if not response.is_error:
            print("\tKey Phrases:")
            for phrase in response.key_phrases:
                print("\t\t", phrase)
        else:
            print(response.id, response.error)

    except Exception as err:
        print("Encountered exception. {}".format(err))

key_phrase_extraction_example(client)
```

```
        Key Phrases:
                 lot
                 recent years
                 Apple
                 Smart phone
```

Using REST API

In [22]:
```
import requests
# pprint is used to format the JSON response
from pprint import pprint
```

In [23]:
```
import os

subscription_key = "5256fa5f05d144f691b6b7e607d21f99"
endpoint = "https://textanalyticsdgdfz.cognitiveservices.azure.com/"
```

In [26]:
```
language_api_url = endpoint + "/text/analytics/v3.0/languages"

documents = {"documents": [
    {"id": "1", "text": "This is a document written in English."},
    {"id": "2", "text": "Este es un document escrito en Español."},
    {"id": "3", "text": "这是一个用中文写的文件"},
    {"id": "4", "text": "This is a document written in English 这是一个用中文写的文件"}
]}
```

In [27]:
```
headers = {"Ocp-Apim-Subscription-Key": subscription_key}
response = requests.post(language_api_url, headers=headers, json=documents)
languages = response.json()
pprint(languages)
```

```
{'documents': [{'detectedLanguage': {'confidenceScore': 0.99,
                                     'iso6391Name': 'en',
                                     'name': 'English'},
                'id': '1',
                'warnings': []},
               {'detectedLanguage': {'confidenceScore': 1.0,
                                     'iso6391Name': 'es',
                                     'name': 'Spanish'},
                'id': '2',
                'warnings': []},
               {'detectedLanguage': {'confidenceScore': 1.0,
                                     'iso6391Name': 'zh_chs',
                                     'name': 'Chinese_Simplified'},
                'id': '3',
                'warnings': []},
               {'detectedLanguage': {'confidenceScore': 0.61,
                                     'iso6391Name': 'zh_chs',
                                     'name': 'Chinese_Simplified'},
                'id': '4',
                'warnings': []}],
 'errors': [],
 'modelVersion': '2020-09-01'}
```

Analyze sentiment

In [28]:
```
sentiment_url = endpoint + "/text/analytics/v3.0/sentiment"

documents = {"documents": [
    {"id": "1", "language": "en",
        "text": "I really enjoy the watching AVENGERS movie. Even i like DC but to screenplay i prefer less watc
    {"id": "2", "language": "es",
        "text": "Este ha sido un dia terrible, llegué tarde al trabajo debido a un accidente automobilistico."}
]}
```

In [31]:
```
headers = {"Ocp-Apim-Subscription-Key": subscription_key}
response = requests.post(sentiment_url, headers=headers, json=documents)
sentiments = response.json()
pprint(sentiments)
```

```
{'documents': [{'confidenceScores': {'negative': 0.0,
                                     'neutral': 0.0,
                                     'positive': 1.0},
                'id': '1',
                'sentences': [{'confidenceScores': {'negative': 0.0,
                                                    'neutral': 0.0,
                                                    'positive': 1.0},
                               'length': 43,
                               'offset': 0,
                               'sentiment': 'positive',
                               'text': 'I really enjoy the watching AVENGERS '
                                       'movie.'},
                              {'confidenceScores': {'negative': 0.23,
                                                    'neutral': 0.48,
                                                    'positive': 0.29},
                               'length': 60,
                               'offset': 44,
                               'sentiment': 'neutral',
                               'text': 'Even i like DC but to screenplay i '
                                       'prefer less watching that'}],
                'sentiment': 'positive',
                'warnings': []},
               {'confidenceScores': {'negative': 0.93,
                                     'neutral': 0.05,
                                     'positive': 0.02},
                'id': '2',
                'sentences': [{'confidenceScores': {'negative': 0.93,
                                                    'neutral': 0.05,
                                                    'positive': 0.02},
                               'length': 92,
                               'offset': 0,
                               'sentiment': 'negative',
                               'text': 'Este ha sido un dia terrible, llegué '
                                       'tarde al trabajo debido a un accidente '
                                       'automobilistico.'}],
                'sentiment': 'negative',
                'warnings': []}],
```

```
 'errors': [],
 'modelVersion': '2020-04-01'}
```

In [32]:
```
keyphrase_url = endpoint + "/text/analytics/v3.0/keyphrases"

documents = {"documents": [
    {"id": "1", "language": "en",
        "text": "I really enjoy the watching AVENGERS movie. Even i like DC but to screenplay i prefer less watc
    {"id": "2", "language": "es",
        "text": "Este ha sido un dia terrible, llegué tarde al trabajo debido a un accidente automobilistico."}
]}
```

In [33]:
```
headers = {"Ocp-Apim-Subscription-Key": subscription_key}
response = requests.post(keyphrase_url, headers=headers, json=documents)
key_phrases = response.json()
pprint(key_phrases)
```

```
{'documents': [{'id': '1',
                'keyPhrases': ['screenplay i', 'watching AVENGERS movie', 'DC'],
                'warnings': []},
               {'id': '2',
                'keyPhrases': ['trabajo debido',
                               'tarde',
                               'dia terrible',
                               'accidente automobilistico'],
                'warnings': []}],
 'errors': [],
 'modelVersion': '2020-07-01'}
```

Identify Entities*(Used to determine category type)

In [35]:
```
entities_url = endpoint + "/text/analytics/v3.0/entities/recognition/general"

documents = {"documents": [
    {"id": "1", "language": "en",
        "text": "I really enjoy the watching AVENGERS movie. Even i like DC but to screenplay i prefer less watc
    {"id": "2", "language": "es",
        "text": "Este ha sido un dia terrible, llegué tarde al trabajo debido a un accidente automobilistico."}
]}
```

In [36]:
```
headers = {"Ocp-Apim-Subscription-Key": subscription_key}
response = requests.post(entities_url, headers=headers, json=documents)
entities = response.json()
pprint(entities)
```

```
{'documents': [{'entities': [], 'id': '1', 'warnings': []},
               {'entities': [{'category': 'DateTime',
                              'confidenceScore': 0.8,
                              'length': 6,
                              'offset': 13,
                              'subcategory': 'Duration',
                              'text': 'un dia'},
                             {'category': 'DateTime',
                              'confidenceScore': 0.8,
                              'length': 5,
                              'offset': 37,
                              'subcategory': 'TimeRange',
                              'text': 'tarde'},
                             {'category': 'Quantity',
                              'confidenceScore': 0.8,
                              'length': 2,
                              'offset': 63,
                              'subcategory': 'Number',
                              'text': 'un'},
                             {'category': 'Event',
                              'confidenceScore': 0.64,
                              'length': 25,
                              'offset': 66,
                              'text': 'accidente automobilistico'}],
                'id': '2',
                'warnings': []}],
 'errors': [],
 'modelVersion': '2020-04-01'}
```

In [ ]: