中原大學 雲端計算平台實務

12/09 - 作業報告

Translate text and Translate Speech with Azure Cognitive Services

資訊碩一 11177035 林彥輝

授課教師：鍾武君 教授

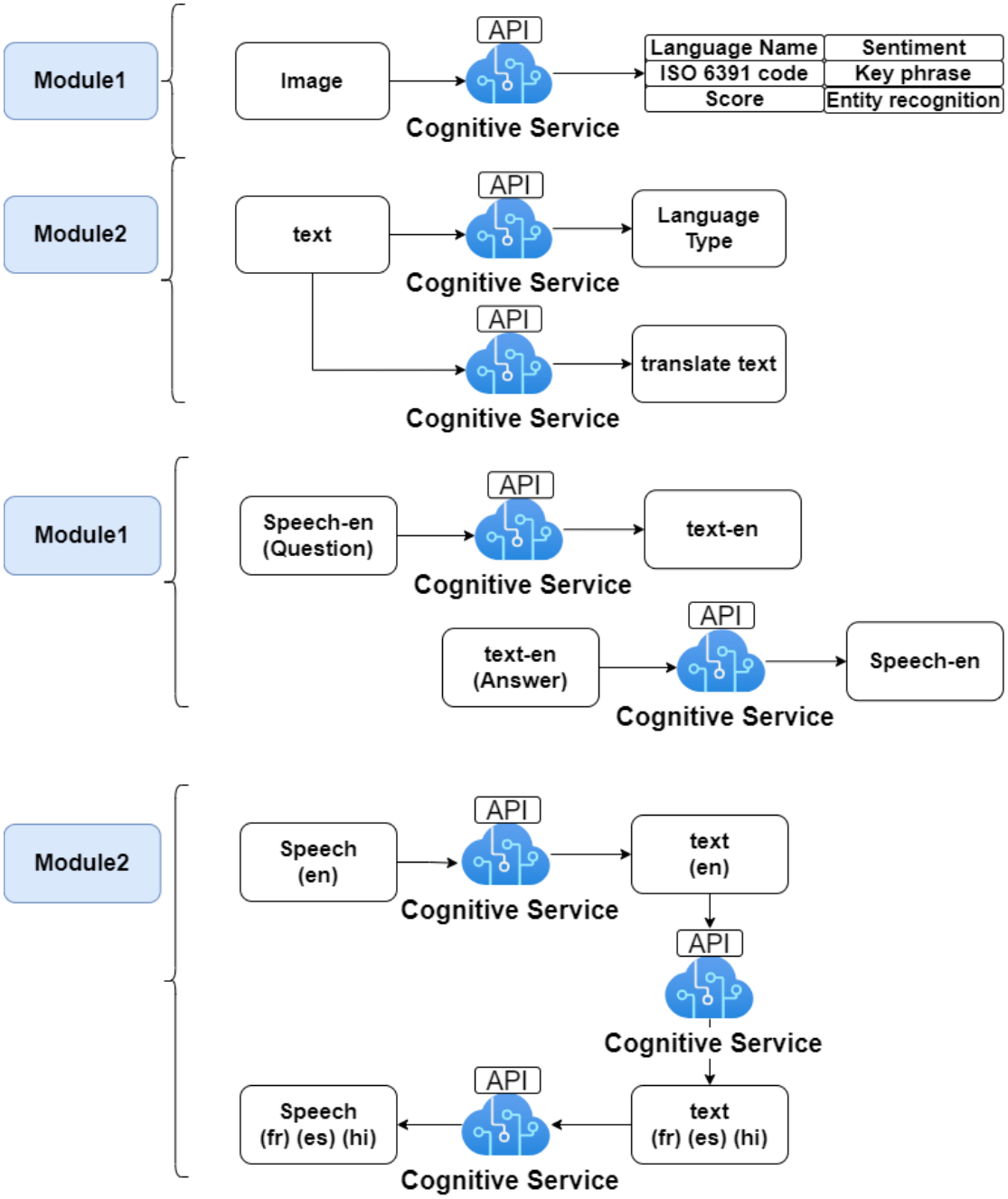
中華民國一一一年十二月

1. Model Intro

[Process and translate text with Azure Cognitive Services]

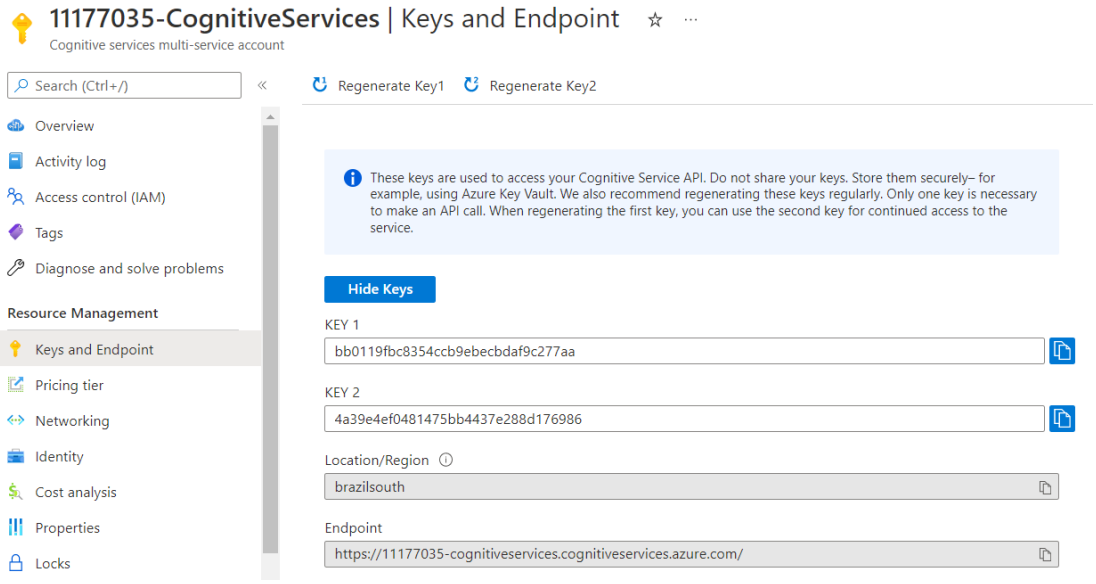
[Process and Translate Speech with Azure Cognitive Speech Services]

2. Summary Homework Assignment



Exercise Environment:

使用 AI-102 教程(https://github.com/MicrosoftLearning/AI-102-AIEngineer)。AI-102 與 AI-900 皆是 Azure 認證的課程，教程中包含各種 Learning Path 的路徑，共 24 個 Lab，此 Exercise 僅使用 05-analyze-text、06-translate-text、07-speech 以及 08-speech-translation 的 Lab。Lab 需要搭配 Azure Cognitive Services 使用。



實驗環境在本地環境，搭配 VSCode 的 IDE 實驗，使用方式只需而外在各 Lab 的.env 檔案填入 Endpoint 與 Key，就能被 Lab 中的 Python 程式碼所讀取。

```
COG_SERVICE_ENDPOINT=https://11177035-cognitiveservices.cognitiveservices.azure.com/

COG_SERVICE_KEY=bb0119fbc8354ccb9ebecbdaf9c277aa
```

實驗需要安裝以下套件才能在本地環境執行完整程式碼：

```
$ pip install azure-ai-textanalytics==5.1.0
$ pip install azure-cognitiveservices-speech==1.19.0
$ pip install playsound==1.2.2
```

(Process and translate text with Azure Cognitive Services)

Module 1: Extract insights from text with the Language service

此模組使用 Cognitive Service 分析文本，分析各種文本的屬性。

使用套件如下：

```
from dotenv import load_dotenv
import os
from azure.core.credentials import AzureKeyCredential
from azure.ai.textanalytics import TextAnalyticsClient
```

os.getenv() 讀取.env 檔案的 Endpoint 與 Key，AzureKeyCredential() 與 TextAnalyticsClient()建立 Azure 憑證以及 Cognitive Service 連線。

```
        # Get Configuration Settings
        load_dotenv()
        cog_endpoint = os.getenv('COG_SERVICE_ENDPOINT')
        cog_key = os.getenv('COG_SERVICE_KEY')

        # Create client using endpoint and key
        credential = AzureKeyCredential(cog_key)
        cog_client = TextAnalyticsClient(endpoint=cog_endpoint, credential=credential)
```

檔案預設執行位於'reviews'的內容，內容包含 5 個.txt 檔案，內容大致如下所示：包含正文、日期、地點。程式碼讀取檔案後會先逐行印出。

Good location and helpful staff, but on a busy road.

The Lombard Hotel, San Francisco, USA

8/16/2018

We stayed here in August after reading reviews. We were very pleased with location, just behind Chestnut Street, a cosmopolitan and trendy area with plenty of restaurants to choose from. The

Marina district was lovely to wander through, very interesting houses. Make sure to walk to the San Francisco Museum of Fine Arts and the Marina to get a good view of Golden Gate bridge and the city. On a bus route and easy to get into centre. Rooms were clean with plenty of room and staff were friendly and helpful. The only down side was the noise from Lombard Street so ask to have a room furthest away from traffic noise.

```
        # Analyze each text file in the reviews folder
        reviews_folder = 'reviews'
        for file_name in os.listdir(reviews_folder):
            # Read the file contents
            print('\n------------\n' + file_name)
            text = open(os.path.join(reviews_folder, file_name), encoding='utf8').read()
            print('\n' + text)
```

依序添加程式碼，首先添加 detect_language()偵測文本語言：返回 English, French 的結果。

```
# Get language

detectedLanguage = cog_client.detect_language(documents=[text])[0]

print('\nLanguage: {}'.format(detectedLanguage.primary_language.name))
```

review1.txt result:

```
review1.txt

Good Hotel and staff
The Royal Hotel, London, UK
3/2/2018
Clean rooms, good service, great location near Buckingham Palace and Westminster 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 o
f fish) with a Michelin Star. We had the taster menu which was fabulous. The rooms were very well appointed with a kitchen, lounge
, bedroom and enormous bathroom. Thoroughly recommended.

Language: English
```

review2.txt result: Language: English

review3.txt result: Language: English

review4.txt result: Language: English

review5.txt result: Language: French

添加 analyze_sentiment()偵測文本情感：返回 positive, negative, mixed 的結果。

```
# Get sentiment

sentimentAnalysis = cog_client.analyze_sentiment(documents=[text])[0]

print("\nSentiment: {}".format(sentimentAnalysis.sentiment))
```

review1.txt result: positive

```
review1.txt

Good Hotel and staff
The Royal Hotel, London, UK
3/2/2018
Clean rooms, good service, great location near Buckingham Palace and Westminster 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 o
f fish) with a Michelin Star. We had the taster menu which was fabulous. The rooms were very well appointed with a kitchen, lounge
, bedroom and enormous bathroom. Thoroughly recommended.

Language: English

Sentiment: positive
```

review2.txt result: Sentiment: mixed

review3.txt result: Sentiment: mixed

review4.txt result: Sentiment: negative

review5.txt result: Sentiment: positive

添加 extract_key_phrases()偵測關鍵詞：

```
        # Get key phrases
        phrases = cog_client.extract_key_phrases(documents=[text])[0].key_phrases
        if len(phrases) > 0:
            print("\nKey Phrases:")
            for phrase in phrases:
                print('\t{}'.format(phrase))
```

result of review1.txt, review2.txt, review3.txt:

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

```
Key Phrases:
        The Royal Hotel
        Tired hotel
        old hotel
        poor service
        United Kingdom
        room furnishings
        office rooms
        flight home
        British Museum
        London
        changing
        internet
        website
        1950
```

```
Key Phrases:
        Golden Gate bridge
        The Lombard Hotel
        The Marina district
        San Francisco Museum
        Lombard Street
        busy road
        Chestnut Street
        trendy area
        interesting houses
        Fine Arts
        good view
        bus route
        down side
        Good location
        helpful staff
        traffic noise
        USA
        We
        August
        reviews
        cosmopolitan
        plenty
        restaurants
        city
        centre
        Rooms
```

result of review4.txt, review5.txt(france):

```
Key Phrases:
        two queen size beds
        busy SIX lane street
        Golden Gate Bridge
        The Lombard Hotel
        Lombard street
        San Francisco
        early morning
        cotton balls
        Marina district
        good places
        walking distance
        late adults
        good hotel
        rooms
        USA
        Traffic
        night
        weekends
        Noise
        ears
        young
        budget
```

```
Key Phrases:
        hôtel agréable
        L'Hotel Buckingham
        Londres
        UK
        personnel
        chambres
```

添加 recognize_entities()偵測識別文本，識別分類如 Document[1]（14項）：Person, PersonType, Location, Organization, Event, Product, Skill, Address, PhoneNumber, Email, URL, IP, DateTime, Quantity.

```
        # Get entities
        entities = cog_client.recognize_entities(documents=[text])[0].entities
        if len(entities) > 0:
            print("\nEntities")
            for entity in entities:
                print('\t{} ({})'.format(entity.text, entity.category))
```

result of review1.txt, review2.txt, review3.txt:

```
Entities
        Hotel (Location)
        staff (PersonType)
        The Royal Hotel (Location)
        London (Location)
        UK (Location)
        3/2/2018 (DateTime)
        Buckingham Palace (Location)
        Westminster Abbey (Location)
        courtyard (Location)
        restaurant (Location)
        Indian (PersonType)
        West coast (Location)
        fish (Product)
        Michelin (Product)
        taster (PersonType)
        rooms (Location)
        kitchen (Location)
        lounge (Location)
        bedroom (Location)
        bathroom (Location)
```

```
Entities
        staff (PersonType)
        The Lombard Hotel (Location)
        San Francisco (Location)
        USA (Location)
        8/16/2018 (DateTime)
        August (DateTime)
        Chestnut Street (Address)
        restaurants (Location)
        Marina district (Location)
        houses (Location)
        San (Location)
        Francisco Museum of Fine Arts (Location)
        Marina (Location)
        Golden Gate bridge (Location)
        city (Location)
        centre (Location)
        Rooms (Location)
        room (Location)
        staff (PersonType)
        Lombard Street (Address)
        room (Location)
```

```
Entities
        hotel (Location)
        The Royal Hotel (Location)
        London (Location)
        United Kingdom (Location)
        5/6/2018 (DateTime)
        hotel (Location)
        since 1950's (DateTime)
        bit (Quantity)
        now (DateTime)
        one (Quantity)
        office rooms (Location)
        home (Location)
        British Museum (Location)
```

result of review4.txt, review5.txt:

```
Entities
        rooms (Location)
        The Lombard Hotel (Locatior
        San Francisco (Location)
        USA (Location)
        9/5/2018 (DateTime)
        Hotel (Location)
        Lombard street (Address)
        SIX (Quantity)
        Golden Gate Bridge (Locatic
        early morning (DateTime)
        night (DateTime)
        weekends (DateTime)
        rooms (Location)
        cotton balls (Product)
        the next day (DateTime)
        Rooms (Location)
        room (Location)
        two (Quantity)
        beds (Product)
        room (Location)
        four in (Quantity)
        room (Location)
        rooms (Location)
        hotel (Location)
        Marina district (Location)
        Presidio (Location)
        hotel (Location)
```

```
Entities
        hôtel (Location)
        Hotel Buckingham (Location)
        Londres (Location)
        UK (Location)
        hôtel (Location)
        Le (Quantity)
        personnel (PersonType)
        chambres (Location)
```

添加 recognize_linked_entities()釐清文檔中的詞句文意（借助 Wikipedia 網站）。

```
        # Get linked entities

        entities = cog_client.recognize_linked_entities(documents=[text])[0].entities

        if len(entities) > 0:

            print("\nLinks")

            for linked_entity in entities:

                print('\t{} ({})'.format(linked_entity.name, linked_entity.url))
```

result of review1.txt:

```
Links
	GOOD Music (https://en.wikipedia.org/wiki/GOOD_Music)
	Hotel (https://en.wikipedia.org/wiki/Hotel)
	The Royal Hotel (https://en.wikipedia.org/wiki/The_Royal_Hotel)
	London (https://en.wikipedia.org/wiki/London)
	Buckingham Palace (https://en.wikipedia.org/wiki/Buckingham_Palace)
	Westminster Abbey (https://en.wikipedia.org/wiki/Westminster_Abbey)
	India (https://en.wikipedia.org/wiki/India)
	West Coast Main Line (https://en.wikipedia.org/wiki/West_Coast_Main_Line)
	Michelin Guide (https://en.wikipedia.org/wiki/Michelin_Guide)
```

result of review2.txt:

```
Links
	The Royal Hotel (https://en.wikipedia.org/wiki/The_Royal_Hotel)
	London (https://en.wikipedia.org/wiki/London)
	British Museum (https://en.wikipedia.org/wiki/British_Museum)
```

result of review3.txt:

```
Links
	Lombardy (https://en.wikipedia.org/wiki/Lombardy)
	Hotel (https://en.wikipedia.org/wiki/Hotel)
	San Francisco (https://en.wikipedia.org/wiki/San_Francisco)
	Chestnut Street (Philadelphia) (https://en.wikipedia.org/wiki/Chestnut_Street_(Philadelphia))
	Marina District, San Francisco (https://en.wikipedia.org/wiki/Marina_District,_San_Francisco)
	Museum of Fine Arts, Boston (https://en.wikipedia.org/wiki/Museum_of_Fine_Arts,_Boston)
	Golden Gate Bridge (https://en.wikipedia.org/wiki/Golden_Gate_Bridge)
	Room (https://en.wikipedia.org/wiki/Room)
	Lombard Street (San Francisco) (https://en.wikipedia.org/wiki/Lombard_Street_(San_Francisco))
```

result of review4.txt:

```
Links
	Lombard, Illinois (https://en.wikipedia.org/wiki/Lombard,_Illinois)
	Hotel (https://en.wikipedia.org/wiki/Hotel)
	San Francisco (https://en.wikipedia.org/wiki/San_Francisco)
	Lombard Street (San Francisco) (https://en.wikipedia.org/wiki/Lombard_Street_(San_Francisco))
	Golden Gate Bridge (https://en.wikipedia.org/wiki/Golden_Gate_Bridge)
	Traffic (https://en.wikipedia.org/wiki/Traffic)
	Noise rock (https://en.wikipedia.org/wiki/Noise_rock)
	Room (https://en.wikipedia.org/wiki/Room)
	Marina District, San Francisco (https://en.wikipedia.org/wiki/Marina_District,_San_Francisco)
	Presidio of San Francisco (https://en.wikipedia.org/wiki/Presidio_of_San_Francisco)
	May (https://en.wikipedia.org/wiki/May)
```

result of review5.txt:

```
Links
	United Nations (https://en.wikipedia.org/wiki/United_Nations)
	L'Hôtel (https://en.wikipedia.org/wiki/L'Hôtel)
	Buckingham (https://en.wikipedia.org/wiki/Buckingham)
	London (https://en.wikipedia.org/wiki/London)
	United Kingdom (https://en.wikipedia.org/wiki/United_Kingdom)
```

(Process and translate text with Azure Cognitive Services)

Module 2: Translate speech with the speech service

此模組使用 Cognitive Service 翻譯文本，先偵測文本語言再翻譯至指定語言。使用套件如下：

```
from dotenv import load_dotenv
import os
import requests, json
```

os.getenv()讀取.env 檔案的 Endpoint 與 Key。

```
    load_dotenv()
    cog_key = os.getenv('COG_SERVICE_KEY')
    cog_region = os.getenv('COG_SERVICE_REGION')
    translator_endpoint = 'https://api.cognitive.microsofttranslator.com'
```

用於展示的文本置放於'reviews'的資料夾，內容大致如下所示：包含正文、日期、地點。程式碼讀取檔案後會先逐行印出。

Un hôtel agréable

L'Hotel Buckingham, Londres, UK

J'adore cet hôtel. Le personnel est très amical et les chambres sont confortables.

```
    # Analyze each text file in the reviews folder
    reviews_folder = 'reviews'
    for file_name in os.listdir(reviews_folder):
        # Read the file contents
        print('\n------------\n' + file_name)
        text = open(os.path.join(reviews_folder, file_name), encoding='utf8').read()
        print('\n' + text)
```

在 main()中，主要呼叫兩個 GetLanguage()偵測語言，再呼叫 Translate()翻文本轉換至"英文"。

```
        # Detect the language
        language = GetLanguage(text)
        print('Language:',language)

        # Translate if not already English
        if language != 'en':
            translation = Translate(text, language)
            print("\nTranslation:\n{}".format(translation))
```

GetLanguage()程式碼中，即展示如何透過 request()進行 post。定義 RestfulAPI 所需要的 header 與 body，使用 Translator 3.0，載入分析文本，呼叫

request()後得到分析結果，Document [2] 除了回傳語言外，還能回傳信心程度。

```
def GetLanguage(text):
    # Default language is English
    language = 'en'


    # Use the Translator detect function
    path = '/detect'
    url = translator_endpoint + path

    # Build the request
    params = {
        'api-version': '3.0'
    }

    headers = {
    'Ocp-Apim-Subscription-Key': cog_key,
    'Ocp-Apim-Subscription-Region': cog_region,
    'Content-type': 'application/json'
    }

    body = [{
        'text': text
    }]

    # Send the request and get response
    request = requests.post(url, params=params, headers=headers, json=body)
    response = request.json()

    # Parse JSON array and get language
    language = response[0]["language"]

    # Return the language
    return language
```

Translate()程式碼中，即展示如何透過 request()進行 post。定義 RestfulAPI 所需要的 header 與 body，載入分析文本，呼叫 request()後得到分析結果，與 GetLanguage()大致類似，Document [2] 寫道只需引入不同的參數就能得到相對應的 Response，以此展示例子額外引入'from'以及'to'的語言，就能在 Response 的欄位中就能擁有"translations"的值，內容即翻譯的結果。

```
def Translate(text, source_language):
    translation = ''

    # Use the Translator translate function
    path = '/translate'
    url = translator_endpoint + path

    # Build the request
    params = {
        'api-version': '3.0',
        'from': source_language,
        'to': ['en']
    }

    headers = {
        'Ocp-Apim-Subscription-Key': cog_key,
        'Ocp-Apim-Subscription-Region': cog_region,
        'Content-type': 'application/json'
    }

    body = [{
        'text': text
    }]

    # Send the request and get response
    request = requests.post(url, params=params, headers=headers, json=body)
    response = request.json()

    # Parse JSON array and get translation
    translation = response[0]["translations"][0]["text"]

    # Return the translation
    return translation
```

展示結果如下，文本中唯一的非英文文本成功的被翻譯為英文文本。

```
review5.txt

Un hôtel agréable
L'Hotel Buckingham, Londres, UK
J'adore cet hôtel. Le personnel est très amical et les chambres sont confortables.
Language: fr

Translation:
A pleasant hotel
The Hotel Buckingham, London, UK
Love this hotel. The staff is very friendly and the rooms are comfortable.
```

(Process and Translate Speech with Azure Cognitive Speech Services)

Module 1: Create speech-enabled apps with the Speech service

此模組使用 Cognitive Service 辨識語音（英語，詢問時間），並根據語音內容回答內容（英文，回答時間）。

使用套件如下：

```
from dotenv import load_dotenv
from datetime import datetime
import os
import azure.cognitiveservices.speech as speech_sdk
from playsound import playsound
```

os.getenv()讀取.env 檔案的 Endpoint 與 Key。使用 speech_sdk 建立 Cognitive Service 連線。

```
        load_dotenv()
        cog_key = os.getenv('COG_SERVICE_KEY')
        cog_region = os.getenv('COG_SERVICE_REGION')

        speech_config = speech_sdk.SpeechConfig(cog_key, cog_region)
        print('Ready to use speech service in:', speech_config.region)
```

main()中，呼叫 TranscribeCommand()辨識語音，確認語音為'what time is it?'，並呼叫 TellTime()回答當下時間，過程皆為英文對話。

```
        command = TranscribeCommand()
        if command.lower() == 'what time is it?':
            TellTime()
```

程式預定問答流程為：

1. "What time is it?"
2. "The time is 20:06."
3. "Time to end this lab."

TranscribeCommand()程式碼中，紅色框線部分為使用麥克風進行語音輸入，綠色框線部分為使用 time.wav 的語音檔輸入，內容為"What time is it?"，兩者擇一進行語音輸入。輸入的語音透過 recognize_once_async()進行非同步的語音辨識，根據 document [3]，可以輸入最多 15 秒的句子，並回傳辨識結果。speech.reason 為辨識種類[4]（NoMatch, Canceled, TranslatedSpeech）。以此 Lab 為例，RecognizedSpeech 代表語音辨識轉換為文本的 enum，接著進一步使用 speech.text 作為 command，並 return command。

```
def TranscribeCommand():
    command = ''

    # Configure speech recognition
    audio_config = speech_sdk.AudioConfig(use_default_microphone=True)
    speech_recognizer = speech_sdk.SpeechRecognizer(speech_config, audio_config)
    print('Speak now...')

    audioFile = 'time.wav'
    playsound(audioFile)
    audio_config = speech_sdk.AudioConfig(filename=audioFile)
    speech_recognizer = speech_sdk.SpeechRecognizer(speech_config, audio_config)

    # Process speech input
    speech = speech_recognizer.recognize_once_async().get()
    if speech.reason == speech_sdk.ResultReason.RecognizedSpeech:
        command = speech.text
        print(command)
    else:
        print(speech.reason)
        if speech.reason == speech_sdk.ResultReason.Canceled:
            cancellation = speech.cancellation_details
            print(cancellation.reason)
            print(cancellation.error_details)
    # Return the command
    return command
```

TellTime()程式碼中，首先呼叫 datetime()得到當下時間，並設定不同發聲語音。最後透過 SSML（Speech Synthesis Markup Language，語音合成標記語言）定義語音相關特徵，並且輸出 ”Time to end this lab!” 語音。

```
def TellTime():
    now = datetime.now()
    response_text = 'The time is {}:{:02d}'.format(now.hour,now.minute)



    # Configure speech synthesis
    # speech_config.speech_synthesis_voice_name = "en-GB-RyanNeural"
    speech_config.speech_synthesis_voice_name = 'en-GB-LibbyNeural' # change this
    speech_synthesizer = speech_sdk.SpeechSynthesizer(speech_config)


    # Synthesize spoken output
    responseSsml = " \
        <speak version='1.0' xmlns='http://www.w3.org/2001/10/synthesis' xml:lang='en-US'> \
            <voice name='en-GB-LibbyNeural'> \
                {} \
                <break strength='weak'/> \
                Time to end this lab! \
            </voice> \
        </speak>".format(response_text)
    speak = speech_synthesizer.speak_ssml_async(responseSsml).get()
    if speak.reason != speech_sdk.ResultReason.SynthesizingAudioCompleted:
        print(speak.reason)
```

(Process and Translate Speech with Azure Cognitive Speech Services)

Module 2: Translate speech with the speech service

此模組使用 Cognitive Service 及時翻譯語音（英語 → 法文、印地安語）。使用套件如下，與前章節相同：

```
from dotenv import load_dotenv
from datetime import datetime
import os
import azure.cognitiveservices.speech as speech_sdk
from playsound import playsound
```

os.getenv()讀取.env 檔案的 Endpoint 與 Key。使用 speech_sdk 建立 Cognitive Service 連線。

```
        cog_key = os.getenv('COG_SERVICE_KEY')
        cog_region = os.getenv('COG_SERVICE_REGION')
        # Configure translation
        translation_config = speech_sdk.translation.SpeechTranslationConfig(cog_key,
cog_region)
```

設定欲辨識語言（en-US）以及翻譯語言（法文 fr、西班牙文 es、印地文 hi）：

```
        translation_config.speech_recognition_language = 'en-US'
        translation_config.add_target_language('fr')
        translation_config.add_target_language('es')
        translation_config.add_target_language('hi')
        print('Ready to translate from',translation_config.speech_recognition_language)


        # Configure speech
        speech_config = speech_sdk.SpeechConfig(cog_key, cog_region)
```

程式執行首先請使用者輸入指令（欲翻譯的語言代碼），過程持續迭代直至使用者輸入'q'。接著根據使用者欲翻譯的語言執行 Translate()，Translate()程式碼接續請使用者輸入英文語音，再透過 speech sdk 翻譯出對應的語系語音。

```
        # Get user input
        targetLanguage = ''
        while targetLanguage != 'quit':
            targetLanguage = input('\nEnter a target language\n fr = French\n es = Spanish\n
hi = Hindi\n Enter anything else to stop\n').lower()
            if targetLanguage in translation_config.target_languages:
                Translate(targetLanguage)
            else:
                targetLanguage = 'quit'
```

Translate()程式碼中，紅色框線部分為使用麥克風進行語音輸入，綠色框線部分為使用 station.wav 的語音檔輸入，內容為”Where is the station?”，兩者擇一進行語音輸入。輸入的語音透過 recognize_once_async()進行非同步的語音辨識。

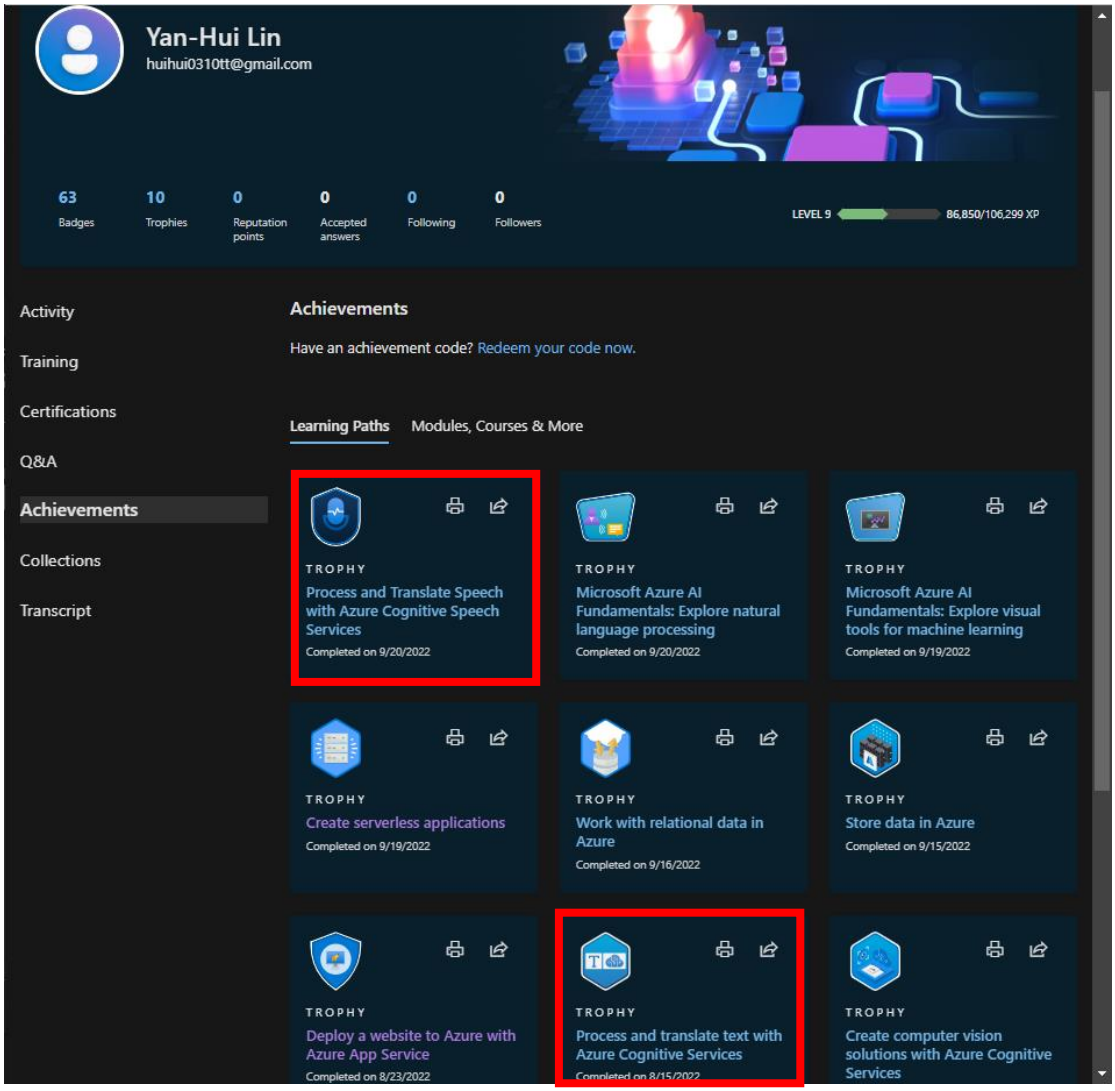
選擇聲音聲線後，SpeechSynthesizer()能進行語音合成，製作出經翻譯的語音，再經由 synthesizer.speak_text_async 輸出語音。

```
def Translate(targetLanguage):
    translation = ''
    # Translate speech
    # audio_config = speech_sdk.AudioConfig(use_default_microphone=True)
    # translator = speech_sdk.translation.TranslationRecognizer(translation_config,
audio_config)
    # print("Speak now...")
    # result = translator.recognize_once_async().get()
    # print('Translating "{}"'.format(result.text))
    # translation = result.translations[targetLanguage]
    # print(translation)

    audioFile = 'station.wav'
    playsound(audioFile)
    audio_config = speech_sdk.AudioConfig(filename=audioFile)
    translator = speech_sdk.translation.TranslationRecognizer(translation_config,
audio_config)
    print("Getting speech from file...")
    result = translator.recognize_once_async().get()
    print('Translating "{}"'.format(result.text))
    translation = result.translations[targetLanguage]
    print(translation)


    # Synthesize translation
    voices = {
            "fr": "fr-FR-HenriNeural",
            "es": "es-ES-ElviraNeural",
            "hi": "hi-IN-MadhurNeural"
    }
    speech_config.speech_synthesis_voice_name = voices.get(targetLanguage)
    speech_synthesizer = speech_sdk.SpeechSynthesizer(speech_config)
    speak = speech_synthesizer.speak_text_async(translation).get()
    if speak.reason != speech_sdk.ResultReason.SynthesizingAudioCompleted:
        print(speak.reason)
```

結果如下所示：選擇欲翻譯的語言代碼後，程式執行一連串的翻譯能製作出翻譯的語音。

```
Enter a target language
 fr = French
 es = Spanish
 hi = Hindi
 Enter anything else to stop

fr
Getting speech from file...
Translating "Where is the station?"
Où se trouve la station?
```

Take screenshots of Badges and Trophies



Learned from the Learning Path

透過這幾個 Module，了解 Cognitive Service 的 Language 以及 Speech 服務，程式碼調用許多的 SDK，使得開發人員能調用 RestfulAPI 使得自己的應用擁有這些服務。但重點是學習如何翻查底層的 Document，許多的調用方式以及回傳內容都需要進階去翻查底層技術手冊才能了解更多，才能進階套用在未來想開發的應用程式上。

3. Problems

過程皆順利完成，模組無 Bug 之處。

FeedBack

Module 上提供的 SDK 連結需要更新，程式碼也需要做更多的說明，例如 API 介紹，許多的 Function()及其回傳內容需要介紹。

Reference

[1] https://learn.microsoft.com/en-us/azure/cognitive-services/language-service/named-entity-recognition/concepts/named-entity-categories

[2] https://learn.microsoft.com/en-us/azure/cognitive-services/translator/reference/v3-0-translate

[3] https://learn.microsoft.com/en-us/python/api/azure-cognitiveservices-speech/azure.cognitiveservices.speech.speechrecognizer?view=azure-python

[4] https://learn.microsoft.com/en-us/javascript/api/microsoft-cognitiveservices-speech-sdk/resultreason?view=azure-node-latest