



中原大學 雲端計算平台實務

12/10-作業報告

Microsoft Azure AI Fundamentals Explore
natural language processing

資訊四乙 10727211 林彥輝

授課教師：鍾武君 教授

中華民國一一〇年十二月

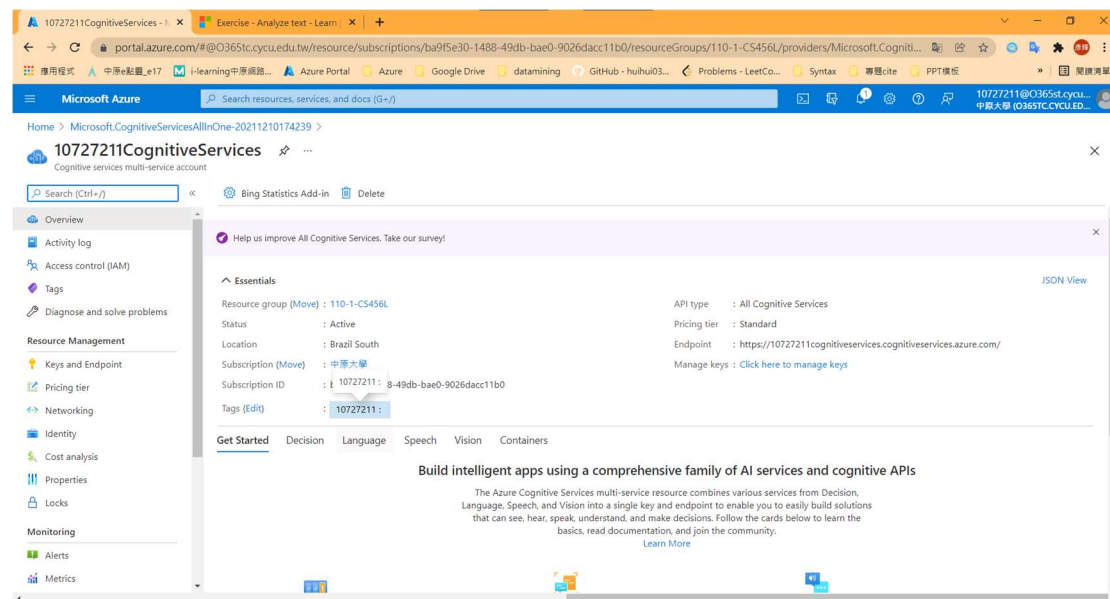
1. Learning Path Intro

Microsoft Azure AI Fundamentals Explore natural language processing
<https://docs.microsoft.com/en-us/learn/paths/explore-natural-language-processing/>

2. Summary Homework Assignment

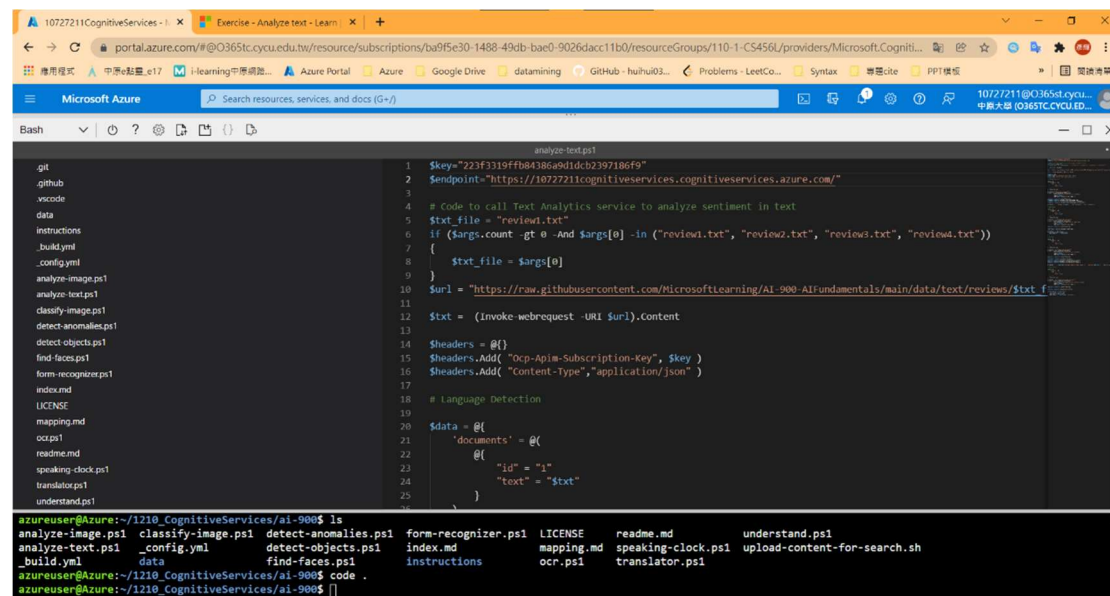
Model 1: Analyze text with the Language service

1. Create a Cognitive Services resource



Part1: 英文版網頁教程

1-2. config client application



1-3. Run code and get results

review1.txt

```
***Detecting Language***  
- Language: English  
- Code: en  
- Score: 0.99
```

```
***Finding Key Phrases***  
- 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
```

```
***Analyzing Sentiment***  
- A positive sentiment based on these scores:  
- Positive: 0.99  
- Neutral: 0.01  
- Negative: 0
```

```
***Identifying known entities***  
- 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
```

review2.txt

```
***Detecting Language***  
- Language: English  
- Code: en  
- Score: 1
```

```
***Finding Key Phrases***  
- 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
```

```
***Analyzing Sentiment***  
- A negative sentiment based on these scores:  
- Positive: 0  
- Neutral: 0.04  
- Negative: 0.96
```

```
***Identifying known entities***  
- 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
```

review3.txt

```
***Detecting Language***
- Language: English
- Code: en
- Score: 0.99
```

```
***Finding Key Phrases***
- 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
```

```
***Analyzing Sentiment***
- A mixed sentiment based on these scores:
- Positive: 0.86
- Neutral: 0
- Negative: 0.14
```

```
***Identifying known entities***
- 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\_\(San\_Francisco\)
```

review4.txt

```
***Detecting Language***
- Language: English
- Code: en
- Score: 1
```

```
***Finding Key Phrases***
- 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
  city
  TINY
  space
  family
  effort
  lots
  Presidio
  young
  budget
```


```
***Analyzing Sentiment***
- A mixed sentiment based on these scores:
- Positive: 0.47
- Neutral: 0.01
- Negative: 0.52
```

```
***Identifying known entities***
- 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
```

Part2: 中文網頁教程

2-2 View Review Documents

...	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 of 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.
	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 didn't work and had to come to one of their office rooms to check in for my flight home. The website says it's close to the British Museum, but it's too far to walk.
	review3.txt	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.



```
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 directly
off the Golden Gate Bridge. Traffic from early morning until late at night
especially on weekends. Noise would not be so bad if rooms were better insulated
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 picked 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 said, rooms are clean and they've made
an effort to update them. The hotel is in Marina district with lots of good places
to eat, within walking distance to Presidio. May be good hotel for young stay-up-
late adults on a budget
```

2-3 Get the Key and Endpoint for Cognitive Services resource




```
> ~
cog_key = '223f3319ffb84386a9d1dcb2397186f9'
cog_endpoint = 'https://10727211cognitiveservices.cognitiveservices.azure.com/'

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

[4] ✓ 0.2s Python

```
... Ready to use cognitive services at
https://10727211cognitiveservices.cognitiveservices.azure.com/ using key
223f3319ffb84386a9d1dcb2397186f9
```

2-4 Detect Language



```
... 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
```


2-5 Extract Key Phrases

1	review1.txt	
2		
3	Key Phrases:	
4	Good Hotel	
5	good service	
6	Clean rooms	
7	Royal Hotel	
8	great location	
9	Buckingham Palace	
10	Westminster Abbey	
11	fish	
12	West coast	
13	lounge	
14	bedroom	
15	enormous bathroom	
16	group	
17	kitchen	
18	London	
19	UK	
20	taster menu	
21	Michelin Star	
22	staff	
23	courtyard	
24		
25		
26	review2.txt	
27		
28	Key Phrases:	
29	old hotel	
30	Royal Hotel	
31	Tired hotel	
32	London	
33	United Kingdom	
34	room furnishings	
35	poor service	
36	British Museum	
37	website	
38	office rooms	
39	flight home	
40	internet	
41		
42		
43	review3.txt	
44		
45	Key Phrases:	
46	helpful staff	
47	Lombard Street	
48	Good location	
49	Chestnut Street	
50	Lombard Hotel	
51	Marina district	
52	traffic noise	
53	San Francisco Museum of Fine Arts	
54	good view of Golden Gate bridge	
55	trendy area	
56	USA	
57	city	
58	bus route	
59	busy road	
60	centre	
61	restaurants	
62	Rooms	
63	interesting houses	
64	reviews	

67	review4.txt	
68		
69	Key Phrases:	
70	rooms	
71	good hotel	
72	Lombard Hotel	
73	Lombard street	
74	late adults	
75	good places	
76	lane street	
77	young stay	
78	night	
79	early morning	
80	Marina district	
81	San Francisco	
82	USA	
83	Golden Gate Bridge	
84	walking distance	
85	queen size beds	
86	ears	
87	Traffic	
88	cotton balls	
89	city	
90	Presidio	
91	weekends	
92	budget	
93	day	
94	effort	
95	Noise	
96	space	
97	family	

2-6 Determine Sentiment

...	review1.txt : positive (0.9999973773956299)	
	review2.txt : negative (5.662441253662109e-07)	
	review3.txt : positive (0.9999995231628418)	
	review4.txt : negative (2.0623207092285156e-05)	

2-7 Extract Known Entities

```
1 review1.txt
2 - Location: The Royal Hotel (https://en.wikipedia.org/wiki/The\_Royal\_Hotel)
3 - Location: London (https://en.wikipedia.org/wiki/London)
4 - DateTime: 3/2/2018
5 - Location: Buckingham Palace (https://en.wikipedia.org/wiki/Buckingham\_Palace)
6 - Location: Westminster Abbey (https://en.wikipedia.org/wiki/Westminster\_Abbey)
7 - Location: India (https://en.wikipedia.org/wiki/India)
8 - Location: West Coast Main Line (https://en.wikipedia.org/wiki/West\_Coast\_Main\_Line)
9 review2.txt
10 - Location: The Royal Hotel (https://en.wikipedia.org/wiki/The\_Royal\_Hotel)
11 - Location: London (https://en.wikipedia.org/wiki/London)
12 - Location: London
13 - Location: United Kingdom
14 - DateTime: 5/6/2018
15 - DateTime: since 1950's
16 - DateTime: now
17 - Location: British Museum (https://en.wikipedia.org/wiki/British\_Museum)
18 review3.txt
19 - Location: Lombardy (https://en.wikipedia.org/wiki/Lombardy)
20 - Location: San Francisco (https://en.wikipedia.org/wiki/San\_Francisco)
21 - DateTime: 8/16/2018 |
22 - DateTime: August
23 - Location: Chestnut Street (Philadelphia) (https://en.wikipedia.org/wiki/Chestnut\_Street\_\(Philadelphia\))
24 - Location: Marina District, San Francisco (https://en.wikipedia.org/wiki/Marina\_District,\_San\_Francisco)
25 - Location: Marina
26 - Location: Golden Gate Bridge (https://en.wikipedia.org/wiki/Golden\_Gate\_Bridge)
27 - Location: Lombard Street (San Francisco) (https://en.wikipedia.org/wiki/Lombard\_Street\_\(San\_Francisco\))
28 review4.txt
29 - Location: Lombard, Illinois (https://en.wikipedia.org/wiki/Lombard,\_Illinois)
30 - Location: San Francisco (https://en.wikipedia.org/wiki/San\_Francisco)
31 - Location: Lombard Street (San Francisco) (https://en.wikipedia.org/wiki/Lombard\_Street\_\(San\_Francisco\))
32 - Location: Lombard
33 - Location: Golden Gate Bridge (https://en.wikipedia.org/wiki/Golden\_Gate\_Bridge)
34 - DateTime: from early morning
35 - DateTime: night
36 - DateTime: the next day
37 - Location: Marina
38 - Location: Marina District, San Francisco (https://en.wikipedia.org/wiki/Marina\_District,\_San\_Francisco)
39 - Location: Presidio of San Francisco (https://en.wikipedia.org/wiki/Presidio\_of\_San\_Francisco)
```

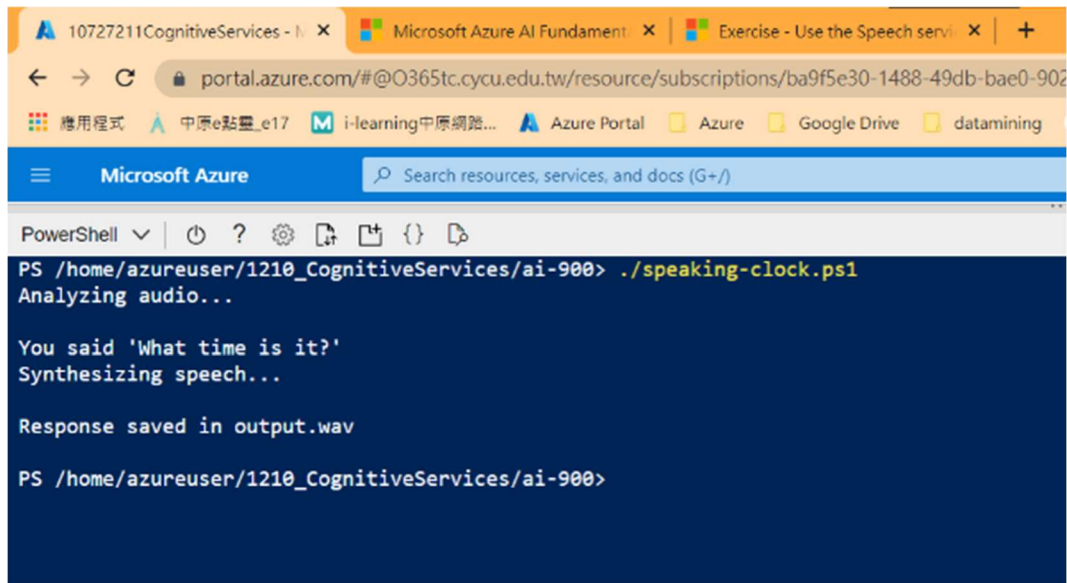
Model 2: Recognize and synthesize speech

Part1: 英文網頁教程

1-1. Get the Key and Location for your Cognitive Services resource

```
speaking-clock.ps1
1 $key="223f3319ffb84386a9d1dcb2397186f9"
2 $region="brazilsouth"
```

1-2. Speech recognition



Part2: 中文網頁教程

2-1. Get the Key and Location for your Cognitive Services resource

```
cog_key = '223f3319ffb84386a9d1dcb2397186f9'
cog_location = 'brazilsouth'

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

[1] ✓ 0.4s Python

... Ready to use cognitive services in brazilsouth using key 223f3319ffb84386a9d1dcb2397186f9

2-2. Speech Recognition

```
# Get spoken command from audio file
file_name = "light-on.wav"
audio_file = 'data/speech/light-on.wav'
# audio_file = os.path.join("data", "speech", file_name)

# Configure speech recognizer
speech_config = SpeechConfig(cog_key, cog_location)
speech_config.speech_synthesis_voice_name = 'en-US-ChristopherNeural'
audio_config = AudioConfig(filename=audio_file) # Use file instead of default (mic)
speech_recognizer = SpeechRecognizer(speech_config, audio_config)

# Use a one-time, synchronous call to transcribe the speech
speech = speech_recognizer.recognize_once()

# Play the original audio file
playsound(audio_file)

# Show transcribed text from audio file
print(speech.text)
```

[5] ✓ 7.7s Python

... Turn the light on.

2-3. Speech synthesis

```
# Get text to be spoken
response_text = 'Turning the light on.'
```

```
# Configure speech synthesis
speech_config = SpeechConfig(cog_key, cog_location)
speech_config.speech_synthesis_voice_name = 'en-US-ChristopherNeural'
speech_synthesizer = SpeechSynthesizer(speech_config)

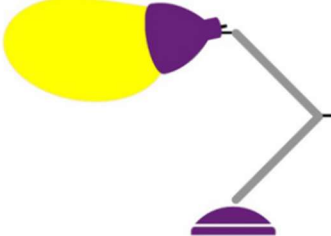
# Transcribe text into speech
result = speech_synthesizer.speak_text(response_text)

# Display an appropriate image
file_name = response_text.lower() + ".jpg"
img = Image.open(os.path.join("data", "speech", file_name))
plt.axis('off')
plt.imshow(img)
```

[6] ✓ 5.2s Python

... <matplotlib.image.AxesImage at 0x2a6eda2cb08>

</>



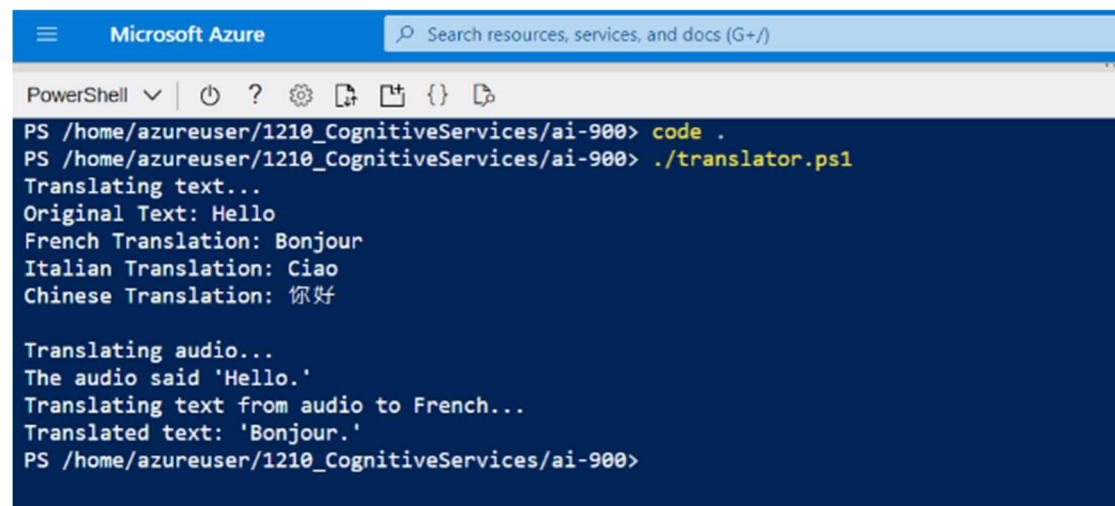
Model 3: Translate text and speech

Part1: 英文網頁教程

1-1. Get the Key and Location for your Cognitive Services resource

```
translatorps1
1 #Add your key here
2 $key="223f3319ffb84386a9d1dcb2397186f9"
3
4 #You need to add your resource location if you use a cognitive services resource
5 $location="brazilsouth"
6
7 #The endpoint is global for the Translator service, DO NOT change it
8 $endpoint="https://api.cognitive.microsofttranslator.com/"
9
10 #Text to be translated
11 $text="Hello"
12
13 # Code to call Text Analytics service to analyze sentiment in text
14 $headers = @{}
15 $headers.Add( "Ocp-Apim-Subscription-Key", $key )
16 $headers.Add( "Ocp-Apim-Subscription-Region", $location )
17 $headers.Add( "Content-Type","application/json" )
18
```

1-2. Speech Translation



```
Microsoft Azure Search resources, services, and docs (G+/)
PowerShell
PS /home/azureuser/1210_CognitiveServices/ai-900> code .
PS /home/azureuser/1210_CognitiveServices/ai-900> ./translator.ps1
Translating text...
Original Text: Hello
French Translation: Bonjour
Italian Translation: Ciao
Chinese Translation: 你好

Translating audio...
The audio said 'Hello.'
Translating text from audio to French...
Translated text: 'Bonjour.'
PS /home/azureuser/1210_CognitiveServices/ai-900>
```

Part2: 中文網頁教程

2-1. Get the Key and Location for your Cognitive Services resource



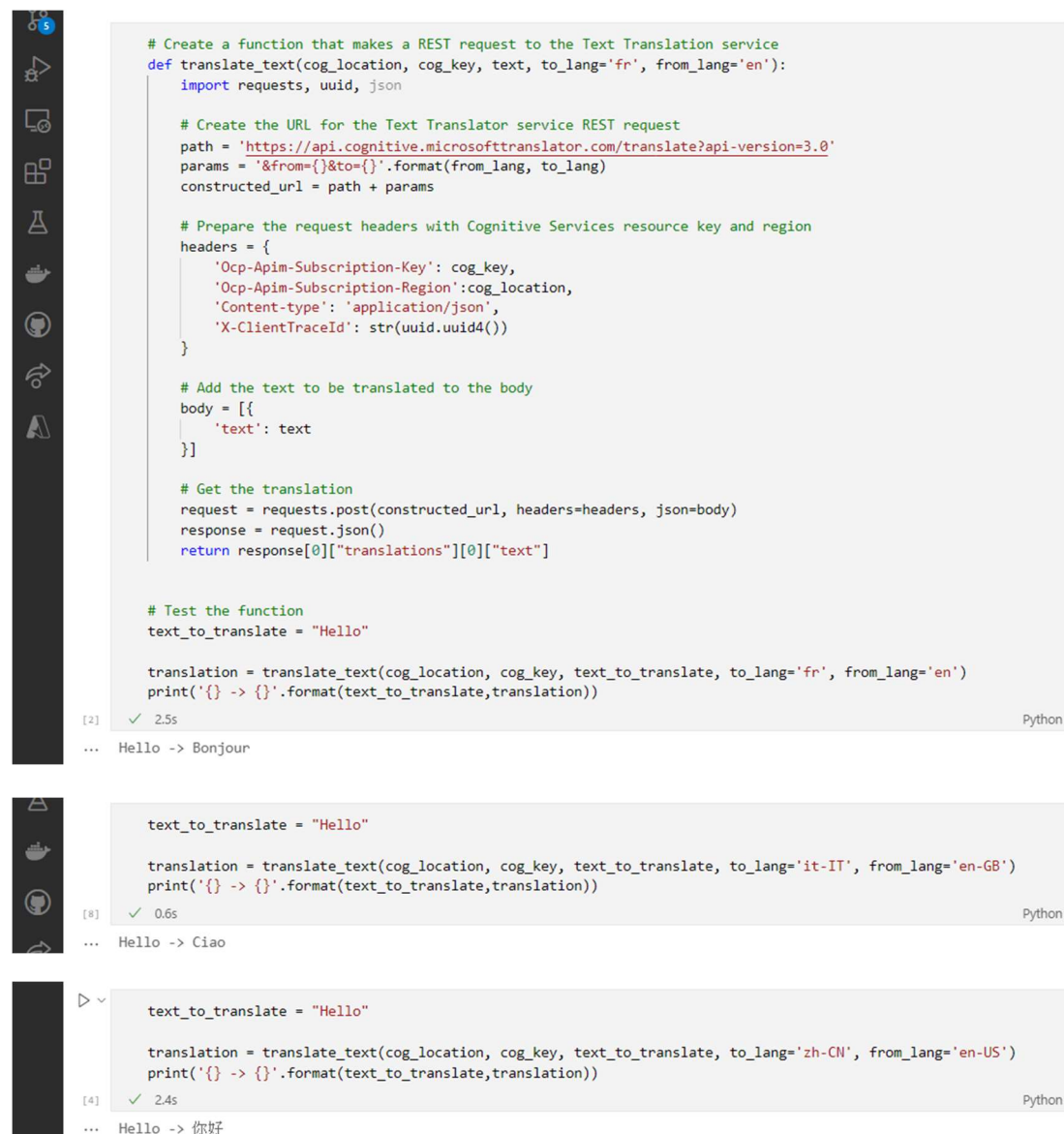
```
cog_key = '223f3319ffb84386a9d1dcb2397186f9'
cog_location = 'brazilsouth'

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

[1] ✓ 0.4s Python

... Ready to use cognitive services in brazilsouth using key 223f3319ffb84386a9d1dcb2397186f9

2-2. Translating Text



The image shows a Jupyter Notebook interface with a dark sidebar on the left containing various icons. The main area displays a Python script for translating text using the Microsoft Translator API. The script defines a function `translate_text` that takes `cog_location`, `cog_key`, `text`, `to_lang`, and `from_lang` as arguments. It constructs a REST request to the `https://api.cognitive.microsofttranslator.com/translate?api-version=3.0` endpoint, sets headers with the Cognitive Services resource key and region, and sends a POST request with the text in the body. The response is then processed to extract the translated text. The script is tested by calling `translate_text` with the text "Hello" and the target language "fr" (French).

```
# 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_lang='en')
print('{} -> {}'.format(text_to_translate, translation))
```

[2] ✓ 2.5s Python

... Hello -> Bonjour

```
text_to_translate = "Hello"

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

[8] ✓ 0.6s Python

... Hello -> Ciao

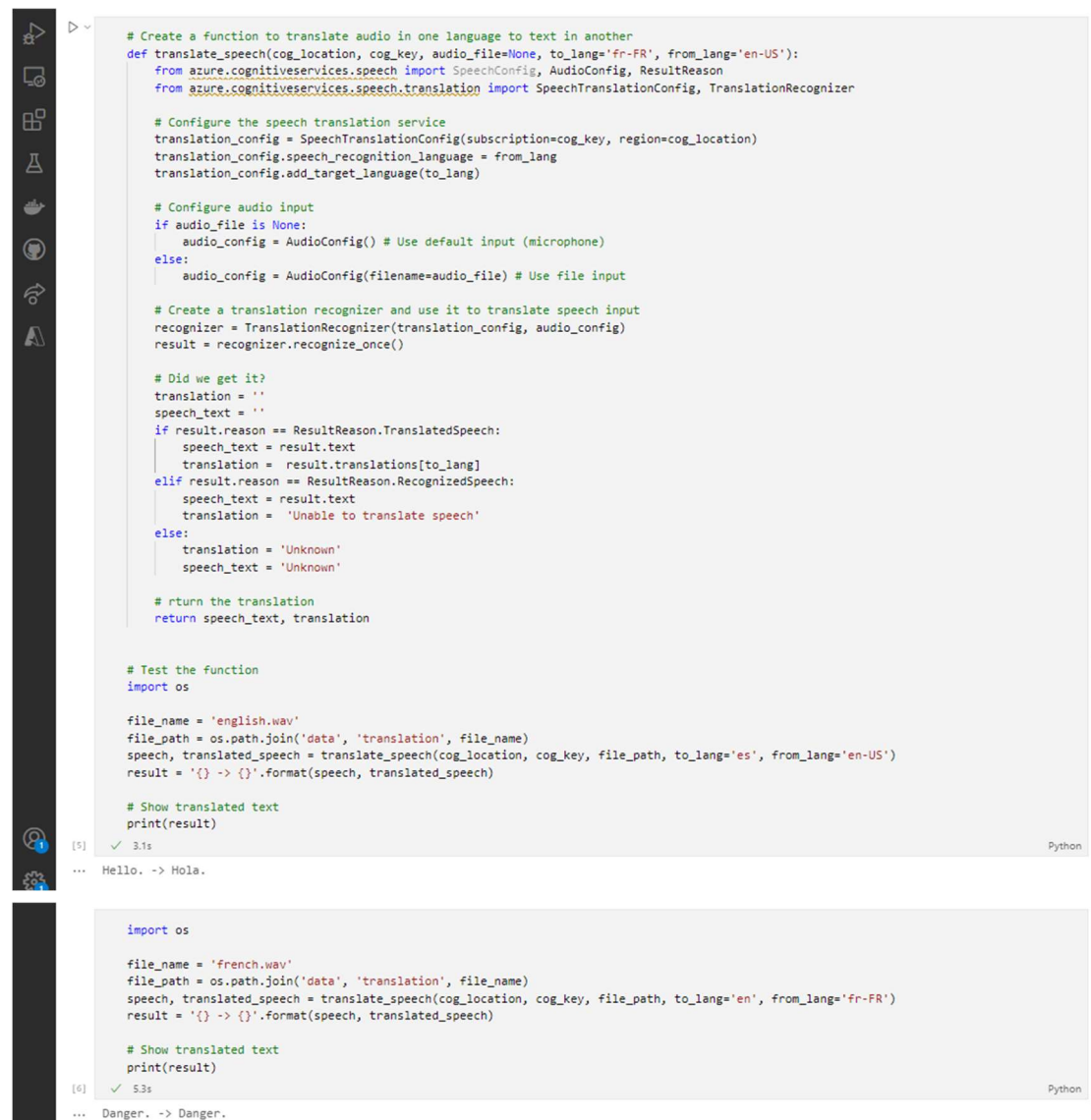
```
text_to_translate = "Hello"

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

[4] ✓ 2.4s Python

... Hello -> 你好

2-3. Speech Translation



The image shows a Jupyter Notebook interface with a dark sidebar on the left containing icons for various tools. The main area displays a Python script for speech translation using the Azure Cognitive Services Speech SDK. The script defines a function `translate_speech` that takes `cog_location`, `cog_key`, `audio_file`, `to_lang`, and `from_lang` as arguments. It configures the speech translation service, sets up audio input (either from a microphone or a file), creates a translation recognizer, and processes the input to return a translation. The script also includes a test section that uses a file named `english.wav` to demonstrate the translation from English to Spanish.

```
# Create a function to translate audio in one language to text in another
def translate_speech(cog_location, cog_key, audio_file=None, to_lang='fr-FR', from_lang='en-US'):
    from azure.cognitiveservices.speech import SpeechConfig, AudioConfig, ResultReason
    from azure.cognitiveservices.speech.translation import SpeechTranslationConfig, TranslationRecognizer

    # Configure the speech translation service
    translation_config = SpeechTranslationConfig(subscription=cog_key, region=cog_location)
    translation_config.speech_recognition_language = from_lang
    translation_config.add_target_language(to_lang)

    # Configure audio input
    if audio_file is None:
        audio_config = AudioConfig() # Use default input (microphone)
    else:
        audio_config = AudioConfig(filename=audio_file) # Use file input

    # Create a translation recognizer and use it to translate speech input
    recognizer = TranslationRecognizer(translation_config, audio_config)
    result = recognizer.recognize_once()

    # Did we get it?
    translation = ''
    speech_text = ''
    if result.reason == ResultReason.TranslatedSpeech:
        speech_text = result.text
        translation = result.translations[to_lang]
    elif result.reason == ResultReason.RecognizedSpeech:
        speech_text = result.text
        translation = 'Unable to translate speech'
    else:
        translation = 'Unknown'
        speech_text = 'Unknown'

    # return the translation
    return speech_text, translation

# Test the function
import os

file_name = 'english.wav'
file_path = os.path.join('data', 'translation', file_name)
speech, translated_speech = translate_speech(cog_location, cog_key, file_path, to_lang='es', from_lang='en-US')
result = '{} -> {}'.format(speech, translated_speech)

# Show translated text
print(result)
```

[5] ✓ 3.1s Python

... Hello. -> Hola.

```
import os

file_name = 'french.wav'
file_path = os.path.join('data', 'translation', file_name)
speech, translated_speech = translate_speech(cog_location, cog_key, file_path, to_lang='en', from_lang='fr-FR')
result = '{} -> {}'.format(speech, translated_speech)

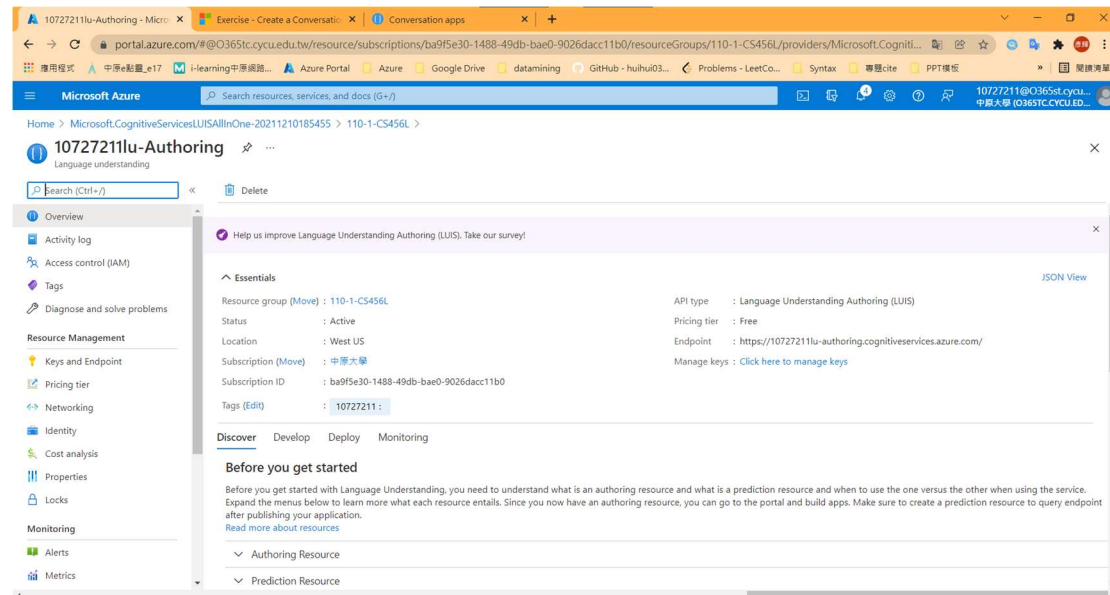
# Show translated text
print(result)
```

[6] ✓ 5.3s Python

... Danger. -> Danger.

Model 4: Create a language model with Conversational Language Understanding

1. Create a Conversational Language Understanding App



2. Create intents and entities

Create new app

Name *

HomeAutomation

Culture * ⓘ

English

Description

Simple home automation

Prediction resource ⓘ

10727211lu

Done

Cancel

Intents ⓘ


+ Create

+ Add prebuilt domain intent

≡ Rename

🗑 Delete

<div><input type="radio"/></div> <div>Name ↑</div>	Examples	Features
<div><input type="radio"/></div> <div>None</div>	0	<div>+ Add feature</div>
<div><input type="radio"/></div> <div>switch_off</div>	2	<div>+ Add feature</div>
<div><input type="radio"/></div> <div>switch_on</div>	2	<div>+ Add feature</div>

switch_on 

Machine learning features ⓘ

+ Add feature

Examples ⓘ

✓ Confirm all entities

📁 Move to ▾

🗑 Delete

⋮

☐ Example user input


Type an example of what a user might say and hit Enter.

☐ turn the fan on

de...

☐ turn the light on

device

switch_off 

Machine learning features ⓘ

+ Add feature

Examples ⓘ

✓ Confirm all entities

📁 Move to ▾

🗑 Delete

⋮

☐ Example user input

Type an example of what a user might say and hit Enter.

☐ turn the fan off

de...

☐ turn the light off

device

3. Train Language Model

The screenshot shows the Microsoft Bot Framework training interface. At the top, there's a navigation bar with 'DASHBOARD', 'BUILD', and 'MANAGE' tabs. The 'BUILD' tab is active. Below the navigation bar, there's a 'Train' button. A green notification box in the top right corner says 'Train completed' with a checkmark icon and the text 'Finished training. Your changes are now effective.' Below the notification, there's a table of intents. The table has columns for 'Name', 'Examples', and 'Features'. The first row is 'None' with 0 examples. The second row is 'switch_off' with 2 examples. The third row is 'switch_on' with 2 examples. Each row has a '+ Add feature' button.

Name	Examples	Features
None	0	+ Add feature
switch_off	2	+ Add feature
switch_on	2	+ Add feature

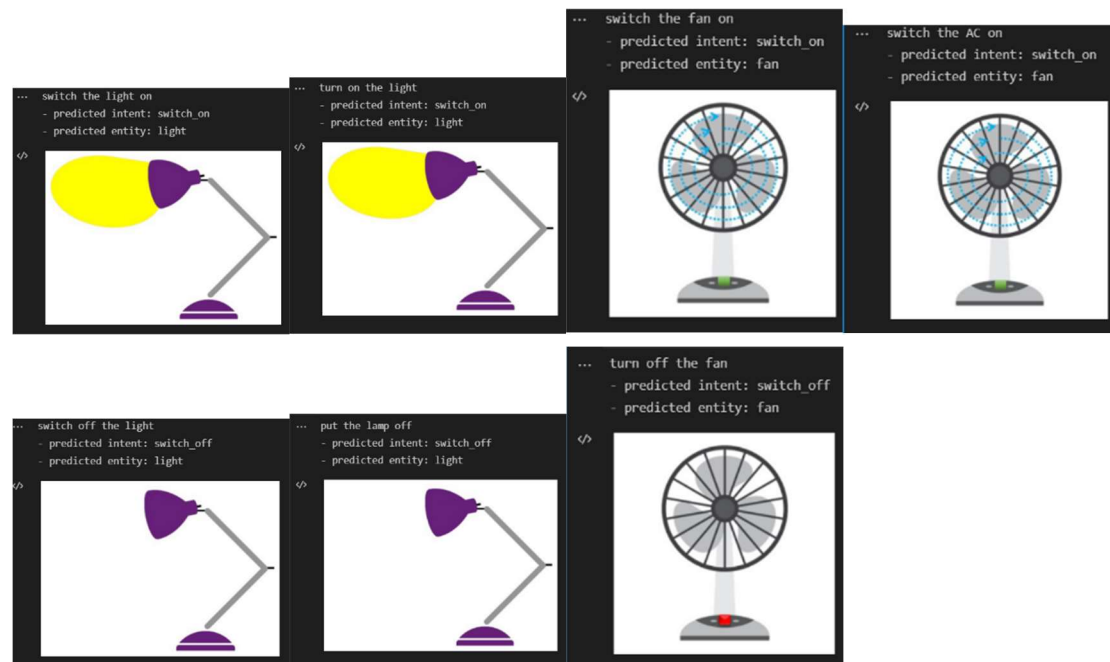
4. Test the Conversational

The two screenshots show the Microsoft Bot Framework testing interface. Both are labeled 'Version: 0.1'. The left screenshot shows the user input 'turn the light on' and the top-scoring intent 'switch_on (0.877)'. The right screenshot shows the user input 'turn the fan on' and the top-scoring intent 'switch_on (0.856)'. Both screenshots show 'ML entities' as 'No predictions', 'Composite entities' as 'No predictions', and 'Other entities' as 'device' (light/fan). The 'Top-matched patterns' section shows 'No matched patterns'.

5. Publish

The screenshot shows the Microsoft Bot Framework publishing dialog box. It has a title 'Choose your publishing slot and settings'. There are two radio buttons: 'Staging slot' and 'Production Slot'. The 'Production Slot' is selected. Below the 'Production Slot' radio button, it says 'Last Published: Not Published yet'. There are two toggle switches: 'Sentiment Analysis: Off' and 'Speech Priming: Off'. There is a link 'Change settings'. At the bottom, there are two buttons: 'Done' and 'Cancel'.

6. Command Input and show appropriate image

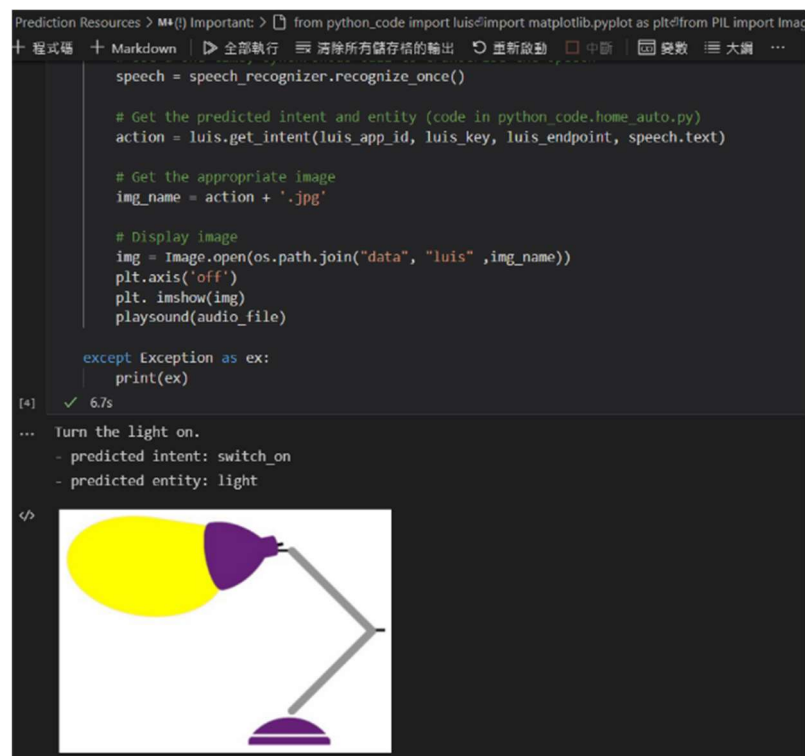


7. config Cognitive Service Resource

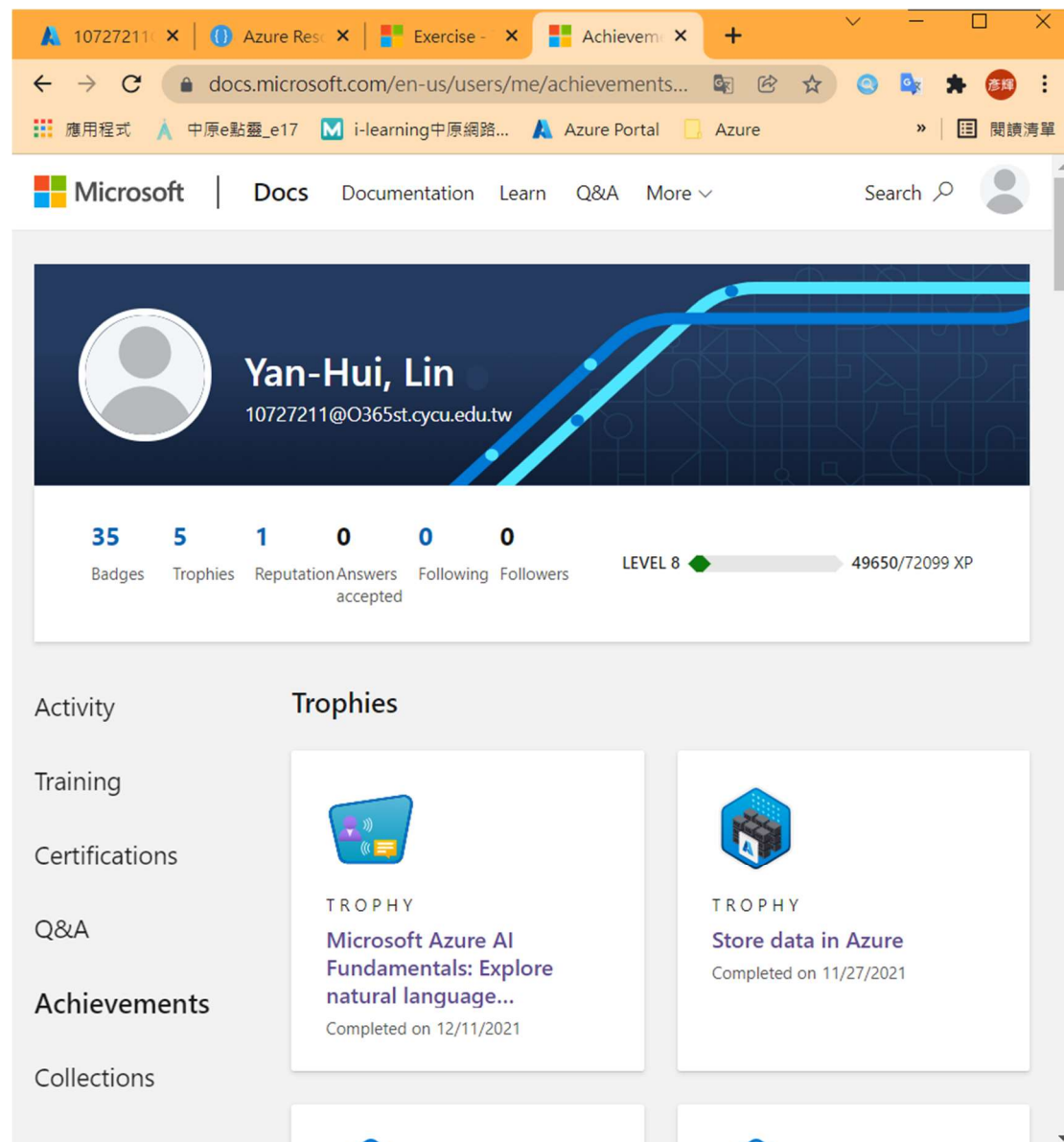
```
cog_key = '223f3319ffb84386a9d1dcb2397186f9'
cog_location = 'brazilsouth'

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

8. Speech Input and show appropriate image



Take screenshots of Badges and Trophies



Learned from the Learning Path

透過這次的 Learning Path 了解 Azure 在 AI、自然語言的服務，從文本分析、語音分析、翻譯、語言理解都能透過 Azure 的 API 簡易達成，且 API 的功能簡單易懂，無須在開發時擔心過多有關自然語言的機器學習問題。Azure 的 Cognitive Services 與 Language Understanding 能夠提供未來有關自然語言任務很大的協助。

Problems

1. (英文版教程)

<https://docs.microsoft.com/en-us/learn/modules/create-language-model-with-language-understanding/3-exercise-create-language-understanding-application>

教程中缺乏 `home-devices.ps1`，導致教程無法繼續完成。

2. (中文版教程)

實驗室無法開啟，解決方法為在本地環境下載 Github 範例程式碼並下載相關套件，安裝套件問題可直接至 stackoverflow 根據關鍵字找尋相關 azure 安裝指令。

3. DSVM

原先想透過 DSVM 的環境下載 Github 範例程式，但驗證卻遲遲無法通過，無法在 110-1-CS456L 的 ResourceGroup 中建立此資源。

FeedBack

中英文版教程建議要統一內容，在這次學習的過程發現中英文的教程內容差異甚大，希望將來可以修正此類問題。