

客語黑客松 AI API 說明

講師：陽明交大 陳又碩 博士



國立陽明交通大學

NATIONAL YANG MING CHIAO TUNG UNIVERSITY

Agenda 大綱

- 1 客語黑客松精神
- 2 客語AI 未來
- 3 客語AI 現況
- 4 雲端資源架構
- 5 API 說明



客語黑客松精神



友善客語環境
客語生活化
推廣客語文化



聯合國判定
客語「**嚴重瀕危**」，
挽救客語文化，
刻不容緩

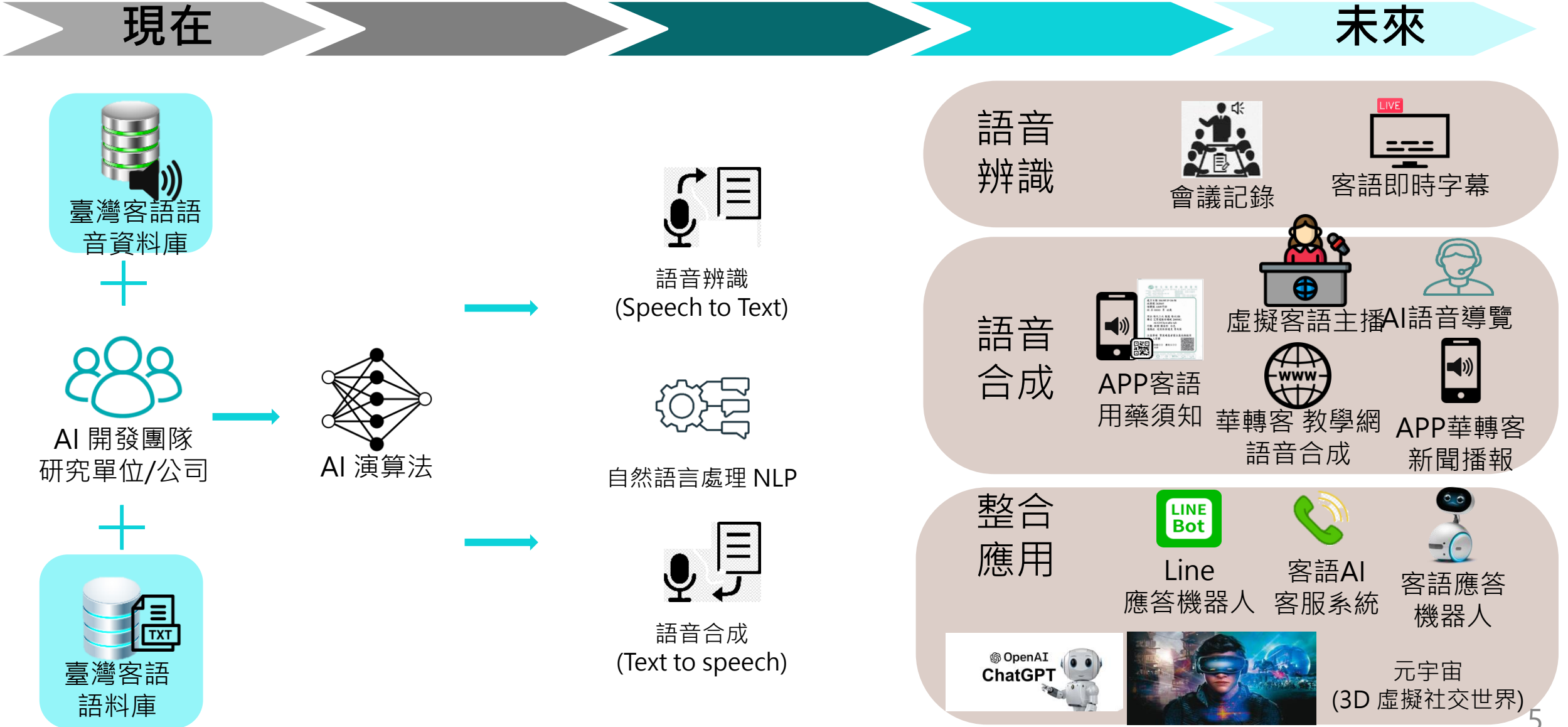


客語黑客松精神



- **AI與網路科技**是保存客語文化、推動客語普及之最佳工具。
- 收集**客語大數據**是客語AI大戰略最重要的第一步。
- 客委會 與 **華碩雲端** 前瞻布局客語大數據，進而**引導廣大AI資源投入**客語文化之發展
- 專案團隊超過**30年收語料與發展語音AI經驗**，也有**全台最大最完整的網路與GPU運算平台(TWCC)**。

客語大數據之未來- 客語語音資料庫及未來開放各界加值應用範圍



客語AI 現況

TTS (語音合成 , text to speech)



- 本資料庫為客語朗讀語料 (reading speech) ，是以原生客語文本，收集客語**四縣腔**、**海陸腔**一男一女，每人 **30** 小時的語料。
- 主要目標是要建置客語語音合成語料庫，作為研發人工智慧之基礎建設，尤其是針對基於深度學習之語音合成研究資料之要求。



語音合成錄音工作流程



設計錄音提示卡

- 從「台灣客語語料料庫」書面語料選擇合適內容
- 搭配生活常用口語語句，如地址、電話、數字
- 提示卡涵蓋所有客語發音

專業發音員錄音

- 四縣腔、海陸腔錄製男女聲各一名
- 每位發音人錄音30小時

語料校正

- 修正客語漢字及拼音，呈現錄音內容實際發音
- 標注語料特殊狀況

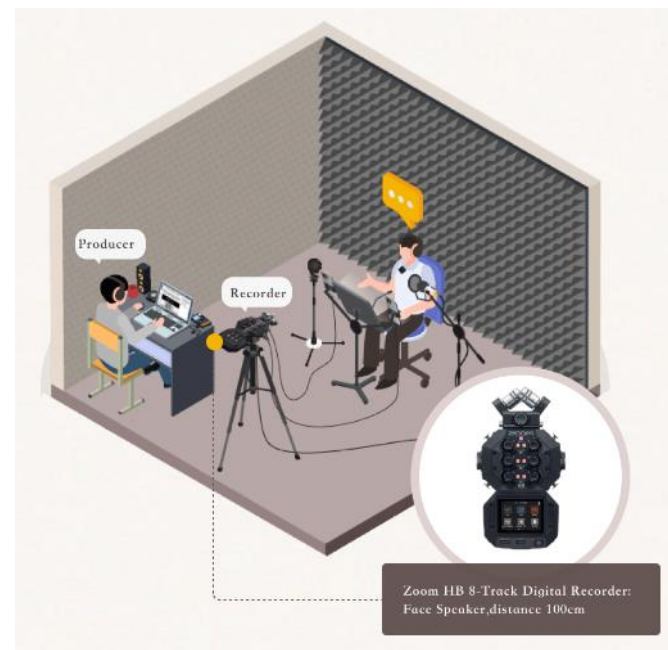
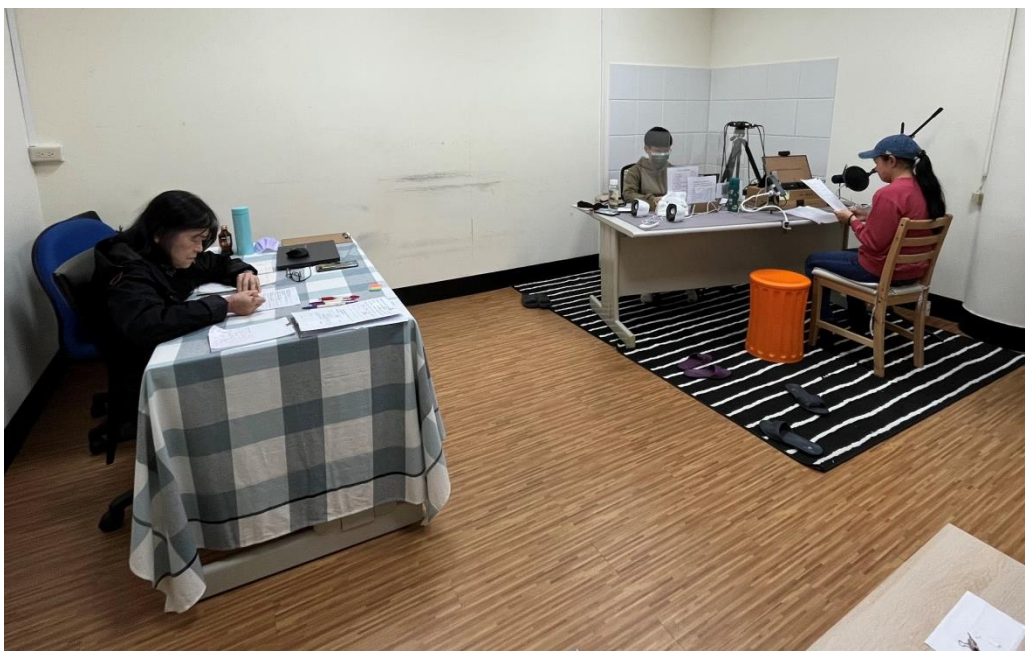
模型訓練

- 使用校正完成的音檔搭配文字訓練出語音合成模型

ASR (語音辨識 , Automatic Speech Recognition)



- 本資料庫為客語朗讀語料 (reading speech) ，是以原生客語文本，收集來自臺灣各地的**四縣腔**、**南四縣腔**、**海陸腔客語**語音，並同時以 **8 支** 麥克風進行錄製。
- 錄好的客語語音，經由兩次人工校正文本後，整理成可供語音辨認技術研究與開發使用之語音語料庫。



語音辨識錄音工作流程



設計錄音提示卡

- 從「台灣客語語料料庫」書面語料選擇合適內容
- 搭配生活常用口語語句，如地址、電話、數字
- 提示卡涵蓋所有客語發音

招募發音員

- 於客委會官網公告招募資訊，公開招募
- 薪傳師、客語教師組：錄製標準客語發音，作為核心語料
- 一般民眾組：錄製可以反應客語使用現況的客語發音，用以訓練適合一般人使用的語音辨認系統

發音員錄音

- 四縣腔、海陸腔分別錄製150名
- 每個腔調收錄300小時，每位發音人2小時
- 另收錄南四縣52位104小時

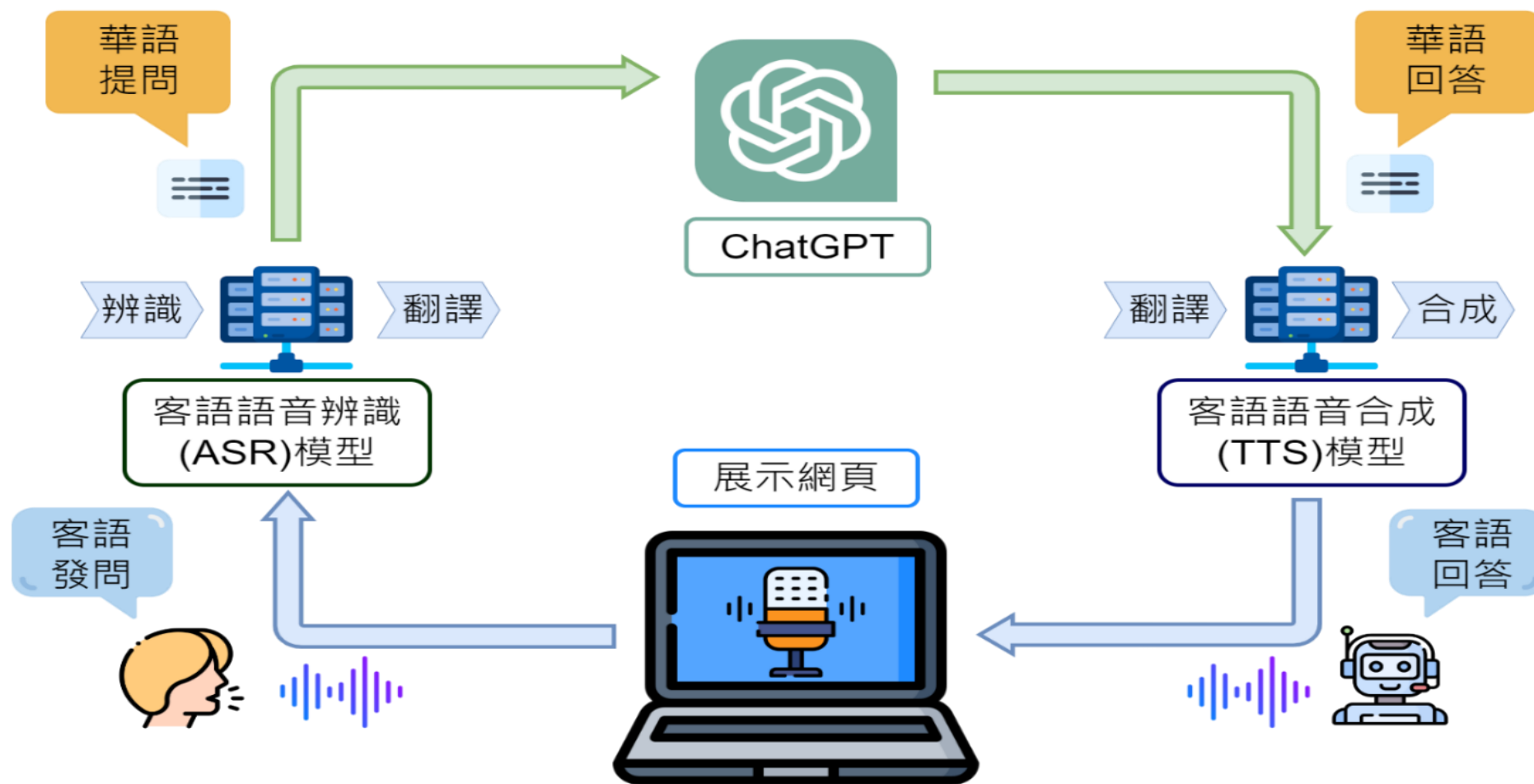
語料校正

- 修正客語漢字及拼音，呈現錄音內容實際發音
- 標注語料特殊狀況

模型訓練

- 使用校正完成的音檔搭配文字訓練出語音辨識模型

客語AI數位實際應用案例



客語機器人



NATIONAL YANG MING CHIAO TUNG UNIVERSITY

歡迎，來打嘴鼓

查詢結果

查詢結果-英譯

 您的問題

 您的問題-英譯

 錄音

CyberLink
PowerDirector

Copyright © 2023 Branda Technology Inc. All rights reserved. 廣聞科技

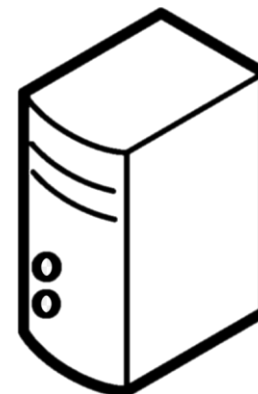
雲端資源架構



語音辨識 API



語音合成 API



雲端運算
資源
參賽團隊



參賽團隊
開發成果



API 說明



- Python Sample Code
- 上傳客語音檔，取回華語文

ASR API

```
# Replace with the actual URL
baseurl = 'http://172.27.8.42:6004' → baseurl = 'http://203.145.221.230:7004'
url = baseurl + '/run/predict'...

# load your audio file
with open('si-06-031.wav', 'rb') as file:
    audio_data = file.read()
    # encode to base64
    audio_base64 = base64.b64encode(audio_data).decode('utf-8')

audio_prefix = "data:audio/wav;base64,"

# to json format
audio_json = json.dumps({"fn_index": 0, "data": [None, {"data": audio_prefix + audio_base64, "name": "audio"}]})

# set header
headers = {'Content-Type': 'application/json'}

# send the request
response = requests.post(url, data=audio_json, headers=headers)

# check response
if response.status_code == 200:
    print(response.text)
else:
    print(f"Request failed with status code: {response.status_code}")
```

- Python Sample Code
- 上傳文字，取回客語音檔

TTS API

```
# Replace with the actual URL
baseurl = 'http://172.27.8.43:10101' → baseurl = 'http://203.145.221.230:10101'
url = baseurl + '/run/predict'...

text = "新年快到了，我們去買年菜好嗎？"

# to json format
audio_json = json.dumps({"fn_index": 0, "data": [text]})

# set header
headers = {'Content-Type': 'application/json'}

# send the request
response = requests.post(url, data=audio_json, headers=headers)

# check response
if response.status_code == 200:
    # Parse the JSON response
    json_response = json.loads(response.text)
    audiopath = json_response['data'][0]['name']
    # Send a GET request to download the file
    download_response = requests.get(baseurl + "/file=" + audiopath)

    if download_response.status_code == 200:
        # Set the path to save the downloaded file
        save_path = 'hakka_tts.wav' # Replace with the desired file path
        # Save the downloaded file
        with open(save_path, 'wb') as file:
            file.write(download_response.content)
        print(f"Download file and save to {save_path}")
    else:
        print(f"Request failed with status code: {download_response.status_code}")
```




Q & A

電話 / 來信詢問

張小姐

[\(02\)2898-7477 分機 86024](tel:(02)2898-7477)

jean.chang@asuscloud.com

線上詢問

<https://www.asuscloud.com/event-hakka/>