基於開源軟體部屬的即時手語生成

Real-time Multimodal Sign Language Generation Leveraging Open-source Frameworks

智慧機器科技 組

企劃書

Contents

1.計畫摘要

為了推動資訊平權並協助聽障者獲取日常語音資訊，我們開發了一套即時生成臺灣手語動畫的服務，讓使用者能夠即時將語音或文字內容轉換為符合臺灣手語語法的動畫，無需仰賴真人手語翻譯。

系統流程結合了語音轉文字(OpenAI Whisper)、語意檢索(使用 FAISS 與中文大語言模型 Qwen2-1.5B)、語意判斷與生成(開源 chat 模型） 以及動畫生成(Blender 動畫渲染)，打造出從語音輸入到手語動畫輸出的完整 pipeline model。

除此之外，我們部署服務到 Kserve 雲端原生平台部署系統，以實現模組化、可擴充的即時服務，具備彈性調度、容易維運的特性。未來，在持續擴增資料集後可廣泛應用於日常生活，補足現有資訊傳遞的斷層，讓手語翻譯服務能真正走入生活。

1.1 研究動機

隨著社會對資訊平權的重視逐年提升，重大記者會中逐漸安排了即時手語翻譯人員，讓聽障人士即便無法聽到聲音，也能透過手語理解資訊。然而，手語翻譯的服務多集中於特定場合，面對日常生活中大量的影音內容，聽障人士獲取資訊的權利依然受到諸多限制。因此，我們決定研發一套專為台灣手語語法設計的即時語音翻手語服務，以有效彌補現有的資訊傳遞缺口，進一步促進資訊平權的實現。

1.2 實際功能描述

為了方便操作，使服務更加使用者友善，我們製作了直覺性的網頁畫面供使用者輸入自然語言查詢。不論是文字輸入、錄音檔或是即時錄音都可以經過語音辨識與語義向量比對，從內建的語料資料庫中擷取語意最相符的資料，並播放的手語動畫展示。本系統不僅提供語言學習者實用工具，也為無障礙科技開創創新應用潛力。

1.3 作品與市場相關系統差異

台灣目前網路上關於本土手語翻譯資料並不多(不像其他國家有公開的完善資料庫)，電子資源就又更少了，公開的可以供查詢資源多半是辭典，且建立的時間久遠，基本不會更新，無法有效滿足生活中的溝通需求。在閱覽論文時也有看到非常多的研究希望透過機器學習在自然語言與手語間找尋規律，達到靈活翻譯境界，但因為資料不足等原因(在第二點會深入說明)而無法產生正確的手語表示，而宣告失敗。

我們在開發的系統的過程中整理現行字典的資料外，也加入了教育部手語書籍的資料，整合成一個規則一致、擁有所有目前可取得資源的電子台灣手語資料庫。在得知手語翻譯的困境後，為得做出最正確的自然語言轉手語翻譯，我們採用語意比對的方式產生需要的翻譯成果，確保輸出源於資料庫有根據的資料。除此之外，為了避免真人手語示範影片的隱私與不一致(不同人、背景等...)，我們一律將他們的手語演示改成統一的虛擬動畫。最後，因為手語資料不足是長期問題，我們將資料庫設計的方便資料擴增，可以開放使用者貢獻資料，並經由我們(管理員)驗證，不斷讓資料庫更加完整、完善，提供更多元的翻譯。

2.創意構思

2.1 理論基礎

我們運用了自然語言處理與語音辨識技術，開發了一套即時手語翻譯(生成)服務系統。使用者透過網頁前端輸入文字、上傳音訊檔案，或進行即時錄音，系統首先利用 Whisper 模型進行語音辨識，將音訊內容準確轉換為文字資料。

接著，這些文字會被傳到後端的語意檢索模組，先由 gte-Qwen-1.5B 向量模型將輸入句子編碼為語意向量，並與資料庫中預先向量化的手語資料進行語意相似度計算，篩選出語意最接近的前五筆資料。之後，我們再透過本地部署的 chat 語言模型，對這些相似候選句進行進一步語意精確比對與語義一致性驗證，以判斷是否存在與輸入語意完全相同的語句。

最後，若最終比對結果符合，我們即從資料庫中擷取對應的手語動畫檔案，傳回至前端展示；若無語意一致之句子，則回傳「沒有相似的句子」，確保翻譯品質與用戶體驗。

2.2 設計創新說明

如同前面在1.3點所說的，台灣目前網路上關於本土手語的電子資源並不多，大部分的查詢資源距今都已有十多年，而語法的統一也不過就是近年的事，之前的資源多有些過時，彼此間的語法也不統一。

在著手製作這套系統前，我們在搜尋相關資料中，不論國內外找到了許多手語翻譯系統，但卻都沒有成功普及於生活中。即便在開始注重資訊平權的現在，缺乏手語人才時，手語翻譯系統仍然未受到重用。為探究其原因，我們諮詢了台灣手語的專家，並得到了以下現在手語翻譯系統碰到的問題:

(1) 台灣手語還是一個「發展中的語言」，目前為止就連專家都還需要透過與聽障朋友交流發掘新的手語表示。

(2) 手語是一個具有空間性的語言，與平時所說的說話不同，相同的語句並不能通過替換主詞、名詞等擴增資料或以此類推。

舉例來說，「我和你吃飯」跟「我和她吃飯」是需要對話人物在空間中

的位置才可以表現的(前者需指向面向的人;後者需指向指名的人)。

再舉動詞的使用為例，光是「吃」一個字在手語中就有超過一千多種比法，並且還不是全部都有紀錄，根據食物的不同而異。

於是，為了應對上述問題，我們決定先以可以產生準確連貫的手語表示為目標。我們因此採取以下方法:

(1) 以語意比對先做到正確翻譯與動畫展現，而非直接用機器學習翻譯。機器學習的方法雖然成功的話泛用性會較廣，但以目前的狀況來說，產出的多是無法採用的手語輸出。

(2) 為了彌補資料不足的困境，我們將資料庫設計的容易加入新資料，不需因新資料的加入重新更新整個資料庫，並鼓勵有條件的使用者貢獻所擁有的手語資料，使開放服務更具使用性。

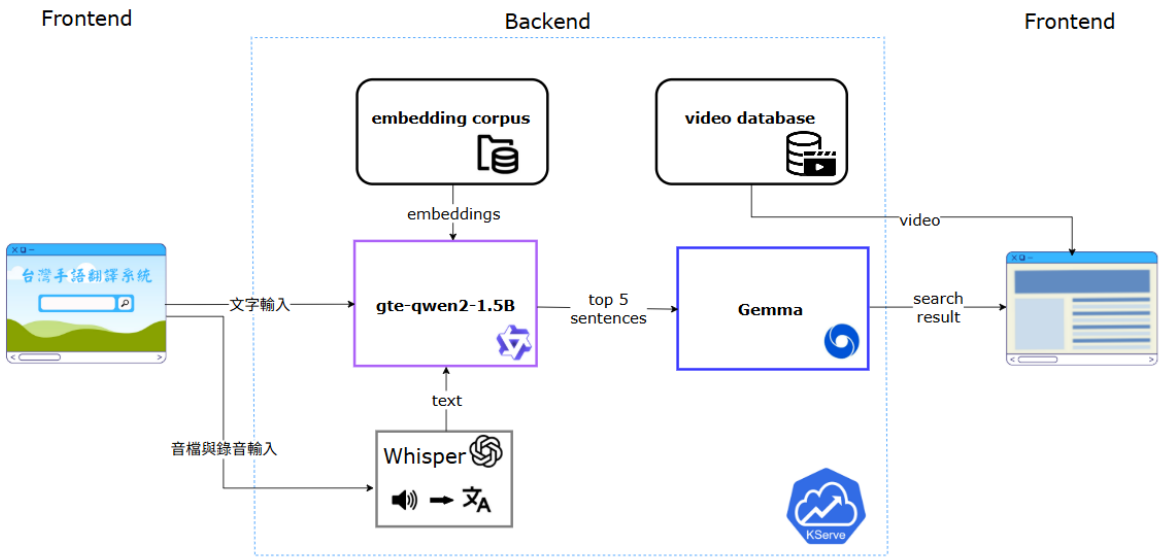
2.3 特殊功能描述

(1) 我們將真人手語影片經 Openpose 擷取手語動作關鍵點後， 再由 Blender 製作成虛擬手語動畫，避免了隱私與個人資料保護問題。

(2) 這個系統所使用的軟體為全部開源且獨立運作，適用於長久維護，並打包部署於 Kserve，支援資源的彈性調度，使服務的提供更加有效率。

3. 系統架構

3.1 架構說明

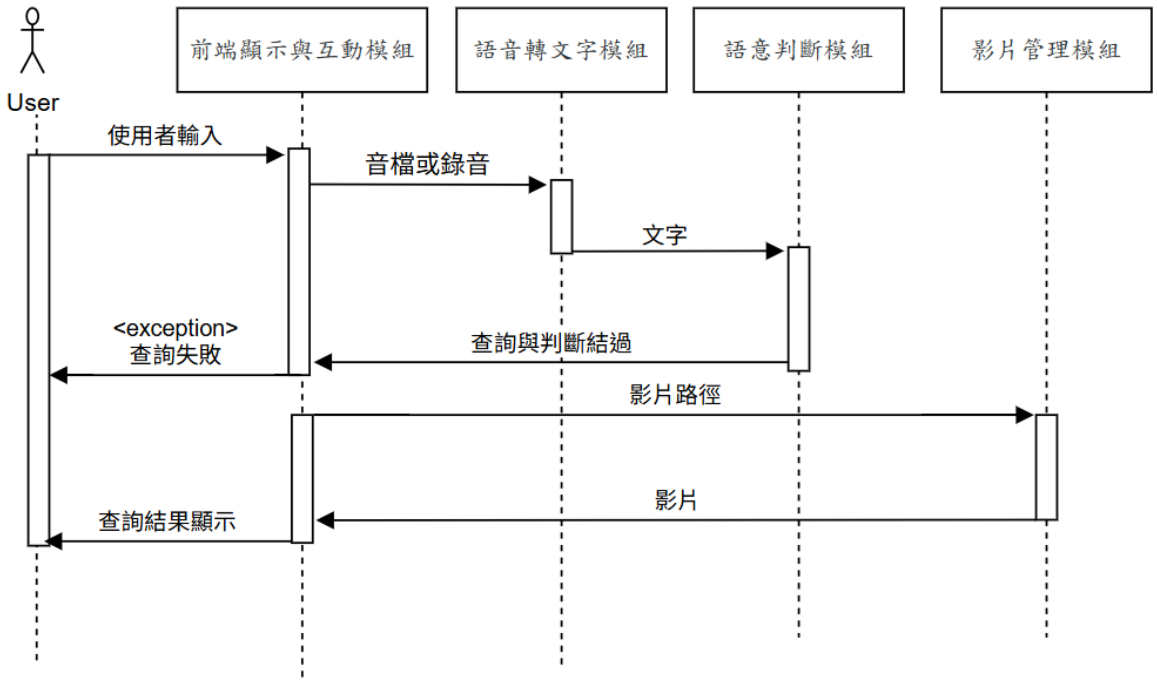


圖一、專案架構圖

圖一為我們系統的架構展示，根據功能可以分為以下幾個模組:

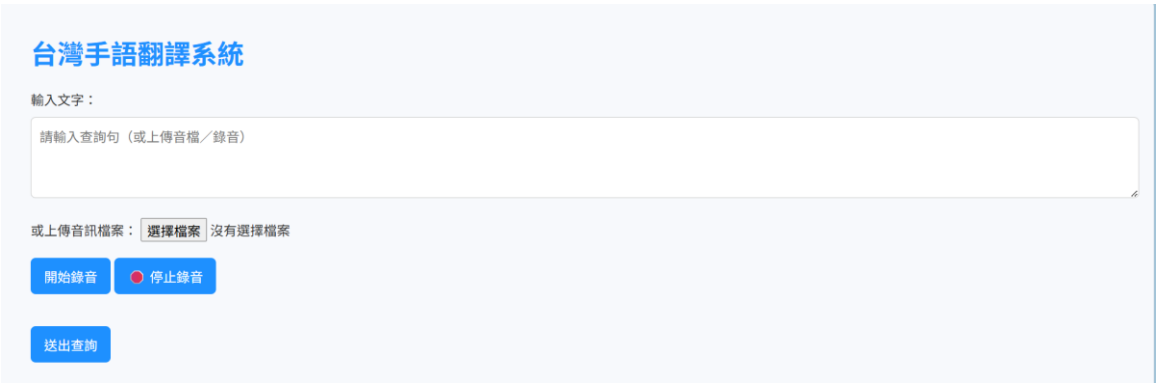
- 前端顯示與互動模組: 是使用者唯一直接接觸到系統的部分，會負責讓使用者輸入欲查詢的資料，並輸出手語影片或查詢失敗。
- 語音轉文字模組: 利用 Open AI 的開源軟體 Whisper 將音檔或使用者的錄音轉換為文字，並傳遞資料給語意判斷模組。
- 語意判斷模組: 將傳過來的內容切割處理，先經過 Qwen2-1.5B 做語意向量檢索，從我們的向量資料庫中找出前五名與查詢語句最相近的資料，再將五筆資料交給 Gemma 做精確判斷(相同與否)。
- 影片管理模組: 儲存 Blender 手語動畫，提供影片給前端顯示。

3.2 使用者互動流程圖



3.3 「人機介面設計」(UI) 與「使用者體驗」(UX) 設計

(1) 「人機介面設計」(UI)



我們將系統的使用者介面採用 HTML 與 CSS 製作，並透過 JavaScript 實現與後端的互動。主要操作流程集中於單一頁面，讓使用者能方便快速地輸入查詢內容並獲得手語翻譯結果。

使用者可透過三種方式進行查詢：直接輸入文字、上傳音檔和即時錄音。介面中設有清楚的文字輸入欄位、音訊上傳按鈕與錄音控制按鈕。當使用者完成輸入並點擊「送出查詢」後，系統會自動判斷輸入類型，並依據處理流程執行語音辨識與語意比對。

查詢結果會以直接顯示於頁面下方，包含語音辨識轉換後的文字內容以及對應的語意比對結果。若有符合的手語動畫資料，將會顯示查詢句與其對應的手語動畫；反之，若無相符資料，則顯示提示訊息，告知使用者無相似句子。

整體介面配色採用淡雅藍與灰白主色系，畫面簡單、可讀性高，適合所有使用者直覺操作與理解。

(2) 「使用者體驗」(UX) 設計

為了提升使用者的體驗，我們在設計過程中考慮了以下幾點：

i. **操作直觀簡單**：整體操作流程簡潔明瞭，使用者僅需透過單一網頁介面，即可完成文字輸入、音檔上傳或錄音，並進行查詢與獲得結果，無需額外安裝軟體或經過繁複設定。

ii. **快速回應與即時查詢**：系統接收到查詢後，會立即進行語音辨識與語意比對，並於處理完成後即時將結果顯示於網頁上，使用者可於短時間內獲得翻譯結果與動畫資訊，達成有效互動。

iii. **支援多模態輸入**：除了文字輸入，系統亦支援上傳音訊檔案與即時語音錄音功能，滿足不同使用情境下的查詢需求，使語音輸入成為更自然的操作方式。

iv. **清楚的結果呈現與輔助提示**：系統以區塊式卡片顯示辨識文字及語意比對結果，清楚標示每一句查詢句與其對應的動畫或系統回饋，若無相符資料亦會明確提示，避免使用者混淆或誤判結果。

4. 計劃管理

工作階段	工作日數	工作內容
1	7	問題統整、專家諮詢、需求調查
2	7	尋找相關技術、架構規劃

3	7	Whisper 測試與部署、手語資料收集
4	7	語意相似度辨識模型與向量化資料庫製作
5	7	語意相似度辨識模型測試
6	7	FAISS 檢索加速研究與測試
7	7	chat LLM 精準判斷製作與測試
8	7	UI 介面製作
9	7	串接前端、語音轉文字模組與語意判斷模組並測試
10	7	用 Openpose 擷取手語影片關鍵點
11	7	測試套用關鍵點到 Blender 虛擬角色
12	7	完成所有手語影片虛擬動畫化
13	7	測試前端抓取並呈現影片
14	7	建立服務 dockerfile、部署至 Kserve
15	7	測試服務在 Kserve 上的狀況與日常使用

周次		1	2	3	4	5	6	7	8	9	10
起始日期		5/13	5/20	5/27	6/3	6/10	6/17	7/1	7/8	7/22	8/5
工作階段	1	✔									
	2		✔								
	3			✔							
	4				✔						
	5					✔					
	6						✔				

	7							✔			
	8								✔		
	9									✔	
	10										✔

5. 軟體清單

5.1 作業系統環境

☐ Windows ☐ FreeBSD ☑ Linux
☐ MacOSX ☐ MacOS Classic ☑其他_Ubuntu___

5.2 主要開發程式語言

☐ Assembly ☐ C ☐ C++ ☐ Java ☐ Perl
☐ PHP ☑ Python ☐ Ruby ☐ .NET ☑ 其他_Javascript__

5.3 專案支援語言(可複選)

☑ 中文 ☑ 英文 ☐ 其他_____

5.4 開發環境

環境與組件	詳細說明
硬體	1.CPU：12th Gen Intel(R) Core(TM) i7-12650H 2.30 GHz 2.GPU：NVIDIA GeForce RTX3090 * 2
軟體	1.Python 2.Whisper 3.Openpose 4.Blender 5.Gemma 6.gte-qwen2-1.5 7.FAISS 8.Kserve
開發工具	1. Visual Studio Code 2. Jupyter Notebook

雲端平台	Jupyter Notebook Server

6. 其他

6.1 修改舊作參賽聲明

☑ 本專案開發之作品未使用團隊成員曾獲競賽獎勵之作品。

☐ 本專案開發之作品採用團隊成員曾獲競賽獎勵之作品，至少應有50%差異，請說明(參考切結書第十點之規定）。

6.2 經費補助

● 通過國科會大專生計畫獲得
  ○ 人事費: 肆萬捌千元
  ○ 業務費: 壹萬元

6.3 專案成果預定授權條款

本專案開發產品授權條款使用 CC BY-SA 宣告。

6.4 權力分配

☑依著作權法第 40 條之規定，由參賽學生與指導教授均等共有。

☐其他比例分配表，請說明