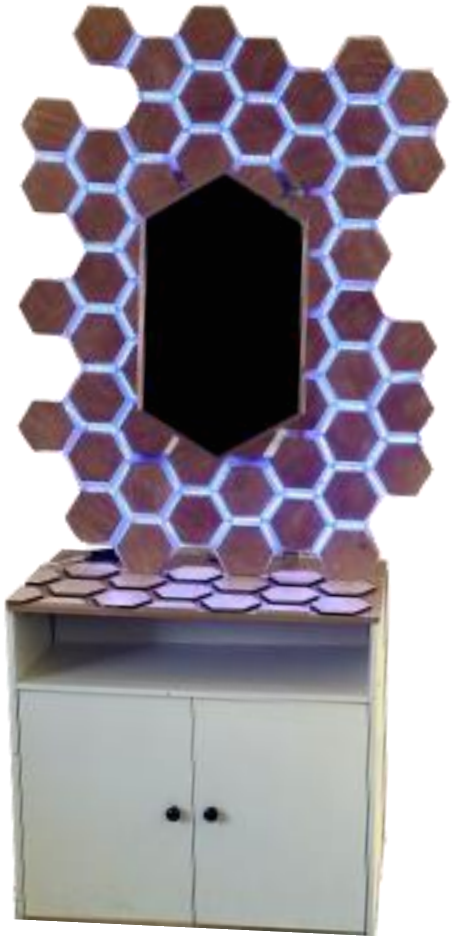
全國高級中等學校專業群科 114 年專題實作及創意競賽
「專題組」作品說明書



群別：電機與電子群

作品名稱：小希

關鍵詞：虛擬陪伴、情感分析、多模態整合

目錄

圖目錄

表目錄

【小希】

壹、 摘要

我們致力於開發一個名為「小希」的虛擬朋友系統，通過自然互動提供溫暖且個性化的陪伴體驗。系統整合臉部辨識、多模態模型、語音互動與情緒分析等技術，使小希能夠識別使用者身份、理解情感狀態，並根據場景與記憶提供貼切回應。結合大語言模型與記憶系統，小希能記住互動經歷，營造知心朋友般的連貫性與親切感。

在互動設計上，我們追求自然流暢的對話體驗。小希能透過表情與動作傳達情感，並根據使用者的情緒與環境動態調整回應，讓陪伴更加真實溫暖。我們期許小希不僅是一個虛擬夥伴，更是一個能真誠理解與回應使用者需求的知心朋友。

貳、 研究動機

有沒有過這樣的時候——你試圖表達自己的情緒，但身邊的人卻無法理解你？當你說「還好」，對方當作沒事，但其實你的內心正在翻滾著說不出口的辛酸；當你以看似平靜的表情面對世界，卻渴望有人能夠真正看到、真正懂得你的心情。

情緒，是人類溝通中最真實卻也最難捕捉的部分。尤其在今天這個科技高速發展的時代，冰冷的螢幕背後，我們比任何時候都需要一份來自科技的溫暖陪伴。然而，當前的互動系統往往止步於回應指令，卻無法察覺我們的情緒，這讓我們感到遺憾，也啟發了我們——能否創造一個真正懂得情緒的虛擬人物，來彌補這份缺憾？

我們希望打造一位能感知情緒的虛擬人，不僅能辨識出你的語氣與表情背後的真實情感，還能用貼近人心的語調、表情與回應，為你帶來溫暖。為了實現這個目標，我們收集並整理了大量數據，並通過反覆的模型訓練與優化，使虛擬人物能準確識別語句中不同語境下的情感含義，同時具備分析如「還好」這類語句背後潛藏情緒的能力，辨別其是真正的平靜還是隱藏的苦澀。

我們希望她的每一個表情、每一句回應，都是一種真誠的交流，讓她成為不僅僅是技術展示，而是一個能真正與人心靈互動的存在。 這是一個融合科技與情感的挑戰，但我們相信，當小希能在你孤單時帶來陪伴、低潮時提供安慰，或是用真摯的回應讓你會心一笑，這份努力就不僅僅是一次創新，而是一種對人與科技未來連結的全新詮釋。

參、 主題與課程之相關性或教學單元之說明

一、雷射切割

我們利用高二彈性課程時所使用雷射切割軟體 RDworksV8 和繪圖軟體 Inventor 做結合，最後用雷切機切割板材，製成我們小希的外框。

二、程式撰寫

(一) Arduino

我們利用高二智慧居家監控實習中學到的 Arduino(如圖 7)，作為 WS2812 燈條的驅動程式，使其能隨模型回應變更顏色。

(二) Python

我們利用高一資訊課所學的基礎 Python 語法(如圖 8)。8 並在課餘時間加以延伸，運用其來訓練模型。

肆、 研究方法

一、軟體系統規劃

以下是系統流程圖(如圖 1)。

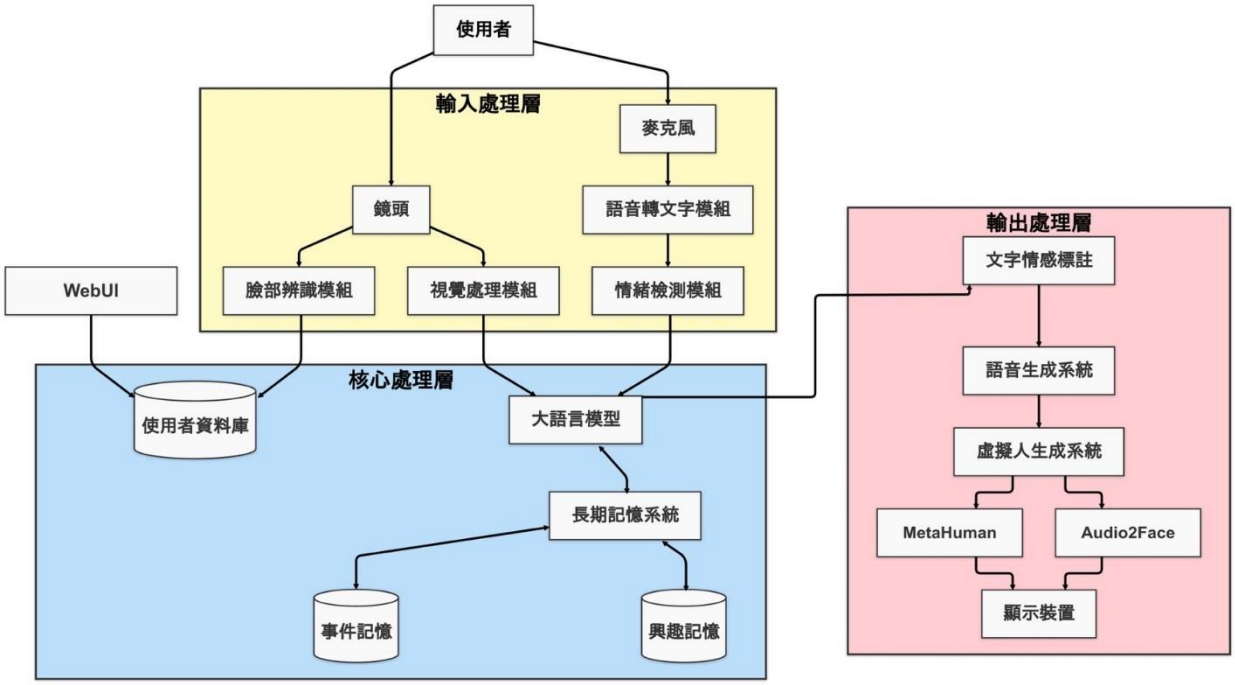


圖 1 系統流程圖

(一) 輸入處理層

輸入處理層負責接收使用者的語音和視覺輸入。麥克風捕捉語音並轉換為文字，鏡頭則捕捉視覺畫面，通過臉部辨識和視覺處理模組分析使用者的身份、表情和環境。情緒檢測模組進一步分析語音和表情，判斷使用者情緒，為後續回應提供依據。

(二) 核心處理層

核心處理層整合並分析輸入數據。使用者資料庫存儲個人信息和互動記錄，與大語言模型結合，提供個性化回應。長期記憶系統記錄互動歷史和興趣偏好，讓虛擬朋友小希能記住細節，提供連貫的陪伴體驗。

(三) 輸出處理層

輸出處理層將分析結果轉化為使用者可感知的回應。文字情感標註為回應添加情感色彩，語音生成系統將其轉為自然語音。虛擬人生成系統使用 MetaHuman 與 Audio2Face 技術創造小希的虛擬形象，通過表情和動作傳達情感，讓互動更真實溫暖。

二、主要技術簡介

(一) 臉部辨識與臉部追蹤技術

雖然臉部辨識並非本專題的核心功能，但為支持多個使用者，我們選擇了 **MTCNN**[1] 和 **FaceNet** [2] 技術。這兩者擁有穩定性、高準確性和資源效率，能在有限硬件資源下實現實時人臉檢測、身份匹配及追蹤功能。同時，其數據庫管理功能提供新用戶註冊與熟人匹配，奠定了我們身分系統的基礎。

(二) 語音轉文字與情緒檢測技術

語音處理部分，我們選用了 **Fast-Whisper-Turbo-v3** [3] 模型，該模型在處理台灣口音中文語音時具備極高準確率，並兼具速度與資源友好性。在情緒檢測上，我們結合了 **emotion2vec** [4] 音訊情緒檢測模型與我們微調的 **Chinese-emotion** [5] ，通過雙重檢測架構克服了單一模型在處理模糊情緒(如悲傷、厭惡)時的不足，實現了更準確的情緒判斷。

(三) 大語言模型與多模態整合

我們選擇了 Qwen-2.5-14B-Instruct [6] 作為核心對話模型，基於其出色的中文處理能力，經我們使用 QLoRA[7] 技術微調後，能在 4bit 模式下以較低資源消耗運行。此外，為實現多模態能力，我們引入 InternVL2.5-2B-MPO 作為視覺處理模組，透過 Python 與主模型透過管道式結合，將圖片描述生成結果與核心語言模型無縫銜接，從而實現視覺與文本的協同處理。

(四) 語音生成

語音生成部分選用了 FishSpeech [8] 模型，其具備良好的中文語音數據基礎，並支持靈活微調。我們選擇該模型的原因在於其易於實現多情緒語調的生成能力，並能通過定製微調適配台灣國語的口音需求。

(五) 虛擬人動畫生成與表現

虛擬人生成部分，我們採用了 NVIDIA Audio2Face [9] 和 MetaHuman [10] 技術，並在 Unreal Engine 5 [11] 上進行整合。該組合的選擇基於其在語音驅動臉部表情生成的真實性，以及高畫質 3D 動畫效果的優勢。

(六) 記憶整理系統

記憶管理部分，我們選擇了 Megrez-3B-Instruct [12] 模型，該模型以穩定性與資源效率著稱，能高效處理多輪對話中的長期記憶檢索與整理需求。其可視化記憶功能為未來的記憶管理與擴展提供了更多可能性。

(七) 硬體與外觀設計

硬體選擇方面，我們設計了一套由 **679** 顆全彩 LED 燈組成的視覺反饋系統，能根據小希的情緒變化展現不同顏色效果，進一步增強用戶體驗。

三、技術開發流程

(一) 臉部辨識系統

臉部辨識模組結合了 MTCNN [1] 和 FaceNet [2] 來實現人臉檢測與特徵提取(如圖 2)。MTCNN 負責檢測畫面中的人臉位置，並提供關鍵點資訊(如眼睛、鼻子、嘴巴的位置)。檢測到人臉後，系統會進行品質檢查，過濾低品質的結果，例如光線不足或角度過大的情況。

為了減少系統運算負擔，我們採用「辨識後追蹤」的策略。具體來說，系統只在初次檢測時進行完整的臉部辨識與特徵提取，之後使用 OpenCV [13] 的 CSRT [14] 追蹤器 來追蹤人臉位置。這樣可以避免每一幀都進行耗時的辨識運算，同時保持追蹤的穩定性。

特徵提取部分，我們使用 FaceNet [2] 將人臉縮放至 160x160 像素，並提取 128 維的特徵嵌入(Embedding)。這些特徵會與資料庫中的已知人臉進行比對，計算歐氏距離以判斷是否匹配。若距離小於設定的閾值(0.75)，則視為匹配成功；否則，系統會將該人臉標註為新的人臉並加入資料庫。

當系統辨識出用戶後，會將該用戶的 ID 發送給我們的主伺服器進行後續處理。主伺服器可以根據用戶 ID 調用對應的個人歷史訊息與記錄用戶的使用行為。這種設計讓系統能夠即時處理攝像頭畫面，並在減少運算負擔的同時，將辨識結果有效地傳遞給其他模組。

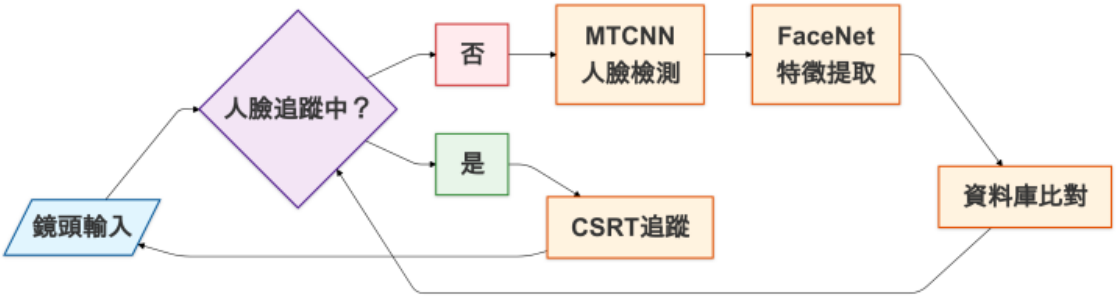


圖 2 人臉身分辨

(二) 語音轉文字與情緒檢測系統

1、 語音轉文字系統

語音轉文字模組採 Fast-Whisper-Turbo-v3 [3] 模型，該模型具備強大的台灣口音中文語音識別能力，能高準確進行語音轉錄，無需調整即可直接應用。在音頻處理階段，系統設計 3 秒環境噪音檢測，通過 VAD (語音活動檢測) [15] 與音量分析，設定音量閾值，濾除噪音，確保輸入品質，有效隔絕雜音。

2、 情緒檢測模組

(1)、音訊情緒檢測模型(emotion2vec) [4]

情緒檢測模組的音訊採 emo2vection 模型。emo2vection 是一款對音訊情緒分類任務設計的模型，適用於處理大多語音情緒類別，尤其在清晰情緒的分類(如開心、驚訝)中表現穩定。然而，在面對模糊情緒(如悲傷、厭惡)時，其分類準確率存在一定局限。為了彌補音訊模型在特殊情緒上的不足，我們設計了文字情緒模型作為輔助，結合雙重檢測以提升分類效果。

(2)、訓練數據集

為提升情緒檢測模組的文本分類能力，我們設計並微調了 Chinese-emotion [5] 模型，使用自建的 Chinese Multi-Emotion Dialogue Dataset [16] 進行訓練。該數據集含 4159 條對話，涵蓋 8 種情緒類別(如「開心」、「憤怒」、「恐懼」等)，來源含日常對話、電影台詞及 AI 生成內容，經人工標註確保品質。數據集情緒分布均衡(如圖 3)。，即使少見情緒(如「厭惡」)也有足夠樣本，提升模型學習效果。

我們將數據集公開於 Hugging Face 平台 [18] ，供研究使用。

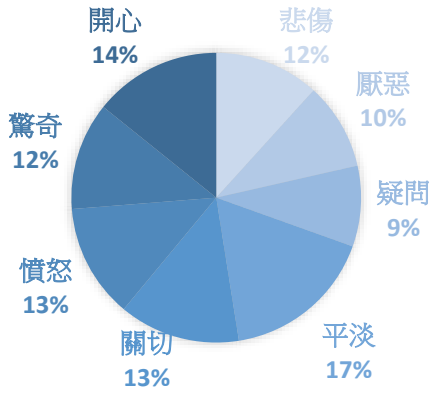
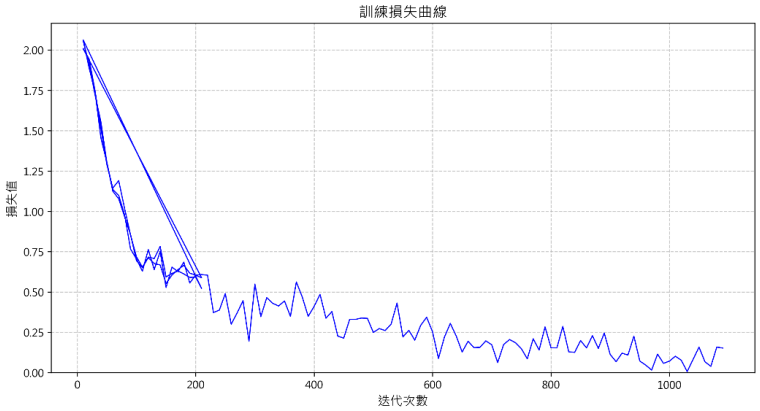


圖 3 數據集情緒分布統計圖

(3)、模型微調過程

我們基於 XLM-RoBERTa-large [18] 模型進行微調，並採用自建數據集進行訓練。訓練過程中，模型設定了學習率 2e-5，批次大小 16， 並進行了 5 輪的訓練迭代。模型的微調過程收斂穩定，損失函數在訓練輪次的增加中逐步降低。訓練損失和驗證損失的下降曲線(如表 1)展現了 模型優化的趨勢，在訓練過程中，損失值的變化展現了模型學習的不同階段：初始階段(0-200 代)，損失值從約 2.0 快速下降到約 0.5，顯示模型在初期學習速度較快；中期階段(200-600 代)，損失值下降速度減緩並出現震盪，但整體仍持續下降，符合正常學習過程；後期階段(600-1000 代)，損失值在較低範圍(約 0.1-0.3)震盪，且總體趨勢略微下降，表明模型逐漸收斂。這一系列變化反映了模型訓練的穩定性與有效性。

表 1 訓練損失曲線



(三) 大語言模型與多模態整合系統的開發

1、 小希角色模型開發

(1) 數據生成框架

在開發過程中，我們將基於 Qwen-2.5 14B instruct [19] 模型進行了微調，首先我們要生成了 2500 則與“小希”相關的對話數據。這些數據重點在於塑造角色的一致性與情感代入感。為了提升數據的多樣性，我們使用了數據生成框架(如圖 4)，通過場景擴增技術對種子數據進行增強與變形，生成了大量高質量的訓練數據。我們首先自行想像出 100 個主題標籤，例如放學後的閒聊、生活小確幸分享等，再透過提示詞給商用大語言模型(DeepSeek-V3 模型[20]，性能媲美 ChatGPT-4o [21])生成特定格式的對話數據，在每次請求生成時都進行主題標籤組合，隨機從 100 個主題中選取 1-3 個，按照排列組合去計算這樣可以得到 166750 個不同風格的主題，這樣就可以確保 2500 則數據完全不同場景，避免過於相似。

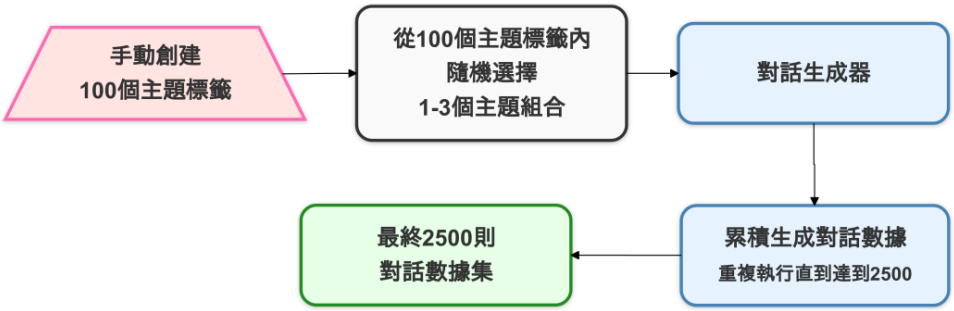
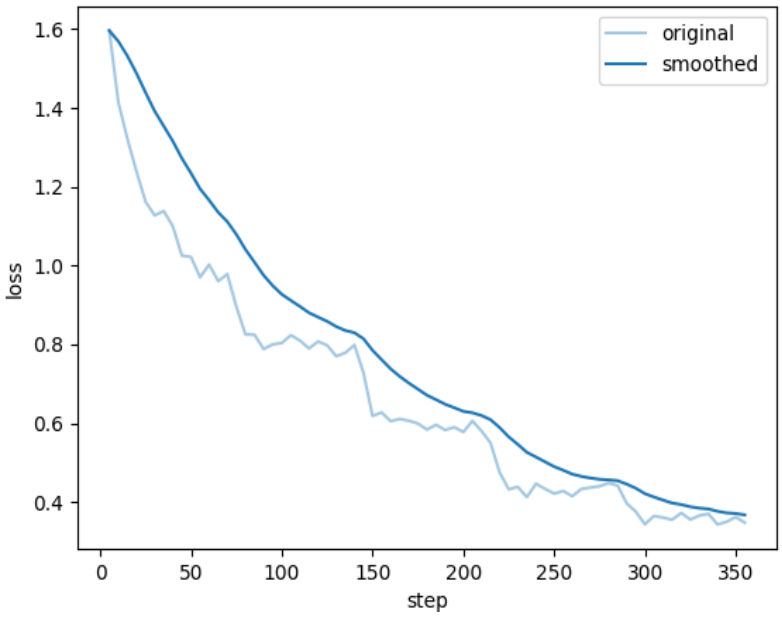


圖 4 數據生成框架流程圖

(2) 微調

在微調過程中，我們設置了一系列重要參數以確保模型的性能優化。學習率設定為 2e-5，這是經過多次微調實驗後確定的最佳值，能夠在 Qwen-2.5 14B instruct-Role-play [22] 模型上平衡學習速度與穩定性。批次大小設定為每設備 8，並採用了 LoRA [7] 技術微調，這使模型能在訓練時消耗的資源相對全精度微調低非常多卻能達到差不多的性能，LoRA[23] 秩大小為 8，LoRA 隨機丟棄率為 0.05，alpha 放大因數設置 16。訓練過程中，我們使用了 BF16 混精度配合 LoRA，最大序列長度為 2048，並基於 NVIDIA L20 [24] 運算卡完成了 5 輪的訓練。在訓練過程中，模型損失函數穩步下降，從 1.6 降至 0.4，驗證損失也同步改善(如表 2)，這表明模型在語言理解與生成任務上的能力顯著提升。

表 2 訓練損失曲線



2、視覺能力整合

為了整合多模態能力，我們將 InternVL 2.5 [25] 的圖像描述通過 Python 管道傳遞給「小希」語言模型。圖像描述在整合前進行了標準化與格式轉換，確保輸入數據的品質與一致性。整個數據流採用同步處理方式，若遇到延遲則通過緩衝機制加以解決。我們還設計了一套提示詞模板，使圖像描述能自然地融入語言上下文，確保生成文本的流暢性與視覺語言的協同性。例如，當使用者輸入「我今天穿的怎樣？」時，模型會拍攝鏡頭畫面，生成圖像描述給「小希」語言模型生成回應，確保視覺與語言的協同性。數據流轉架構圖（如圖 5）展示了 InternVL2.5 與小希的整合。

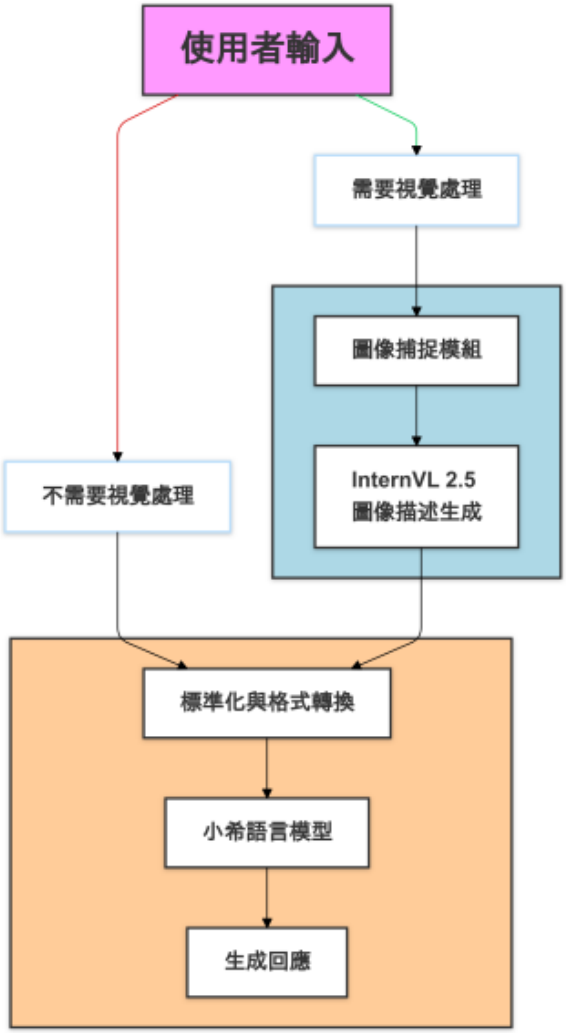


圖 5 視覺處理邏輯

(四) 語音生成模組的開發

1、模型微調

我們選擇 FishSpeech [9] 作為語音生成模組的核心技術，因為該模型經過大量的中文語音數據訓練，具備優異的性能。同時，FishSpeech 提供了便捷的微調介面，這使其成為開發過程中的理想選擇。然而，FishSpeech 原生生成的語音帶有濃重的中國內地口音，這成為我們面臨的主要挑戰。為了解決這個問題，我們專門進行了台灣口音的微調。

微調中，我們一樣用 LoRA 技術進行訓練。此外，為了提升模型對台灣口音的語音生成，我們錄了 59 分鐘的台灣國語音檔，這些音檔涵蓋不同語速與語調，為模型提供了豐富的訓練數據。微調顯著改善了模型口音，使其能夠更準確符合台灣語音特徵。

2、情緒生成

為實現情緒化語音生成，我們在已微調的模型基礎上，用 Zero-Shot [26] 推理技術模擬多種情緒語氣，這是個以利用少量樣本去進行模擬音色與語調的技術，我們另外錄製了一位女組員的聲音作樣本，讓模型可以模擬她的音色與語調，每個情緒共有 80 至 90 秒的對應口氣，經測試此時間長度是效果模擬最好的。系統支援 8 種不同情緒，平淡語調、開心語調、關切語調、憤怒語調、驚奇語調、悲傷語調、厭惡語調與疑問語調，模型能夠根據情境需求靈活調整語音表現。具體的語音生成流程如下：首先，待核心大語言模型(小希)生成對應的文本回應；接著，使用我們的情緒文本分詞模型(Chinese-emotion)[5] 對回應進行情緒標註，以識別出文本中所蘊含的情感傾向；最後，根據情緒標註的結果，調用對應的情緒語氣模型進行語音推理，從而生成具有情感色彩的語音輸出。這套系統(如圖 6)能夠讓語音更加生動自然，貼近人類的情感表達。

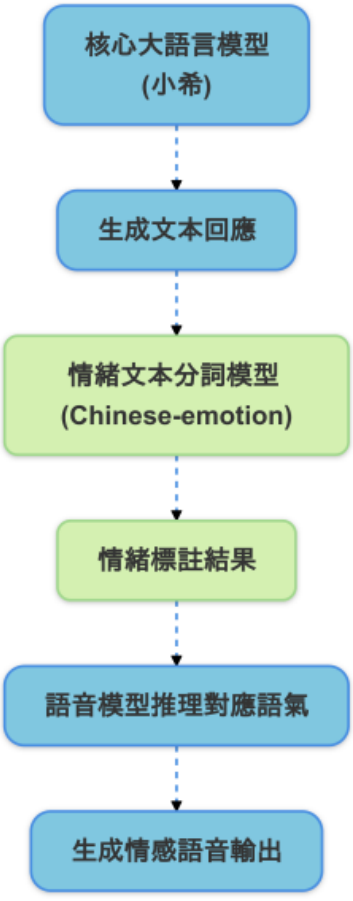


圖 6 語音情感生成流程圖

(五) 虛擬人物動畫生成系統的開發

我們以 NVIDIA Audio2Face(如圖 7) 和 Unreal Engine 5 為核心渲染平台，構建了虛擬人動畫生成系統。通過使用 Python 與 Audio2Face 的 gRPC [27] 串流接口通信，系統能夠將情感生成的語音即時輸入，並利用 SteamLiveLink 功能與 Unreal Engine 5 中我們自行建構的 MetaHuman(如圖 8) 連接，實現語音驅動的臉部表情動畫生成。系統設計中，所有表情動畫會根據「小希」生成的回應情緒(透過我們的情緒標註模型得知目前的情緒)進行動態調整，確保虛擬人的表現與語音情緒保持一致。為了提升系統穩定性，我們在 Unreal Engine 5 中編寫了自定義藍圖，用於自動重新載入 SteamLiveLink，避免串流中斷問題，同時實現毫秒級的即時渲染能力，確保語音生成後能即時同步臉部表情動態。此外，Audio2Face 提供的高度擬真臉部動畫經過細緻調整，包括嘴唇同步和面部肌肉動態，使虛擬人的表情更加自然。結合 Unreal Engine 5 的高畫質渲染技術，進一步提升了虛擬人的沉浸感與真實感，使其具備真實交互的效果。

圖 7 Audio2Fce

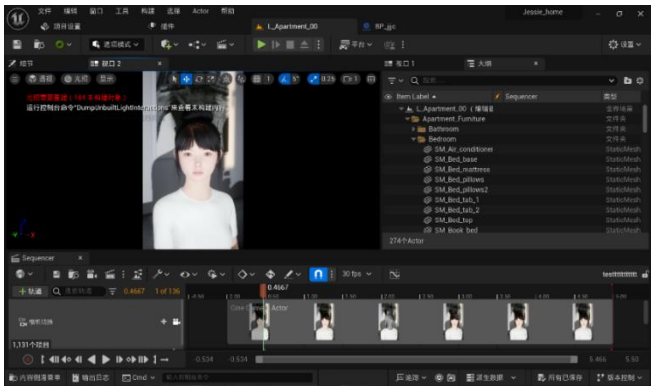


圖 8 Unreal Engine 內的小希

(六) 記憶整理系統的開發

記憶整理模組的核心功能在動態管理用戶對話中的短期與長期記憶，在需要時快速檢索相關記憶以支援系統處理。具體而言，模組首先透過短期記憶管理功能，維持最近 50 則對話的記憶，確保系統能夠迅速回應當前對話情境；其次，長期記憶管理功能則將超過 300 則的對話數據整理為長期記憶，並定期每 100 則重新整合事件記憶，以確保記憶結構的完整性。此外，模組還利用語意相似度模型 (TencentBAC/Conan-embedding-v1) [28] 進行記憶檢索與調度，能夠根據當前對話內容快速檢索相關的事件記憶，並將這些檢索結果標示為「有關的記憶」，加入到小希模型的提示詞中，從而提升系統的上下文理解與回應準確性。

1、 記憶存儲與整理

我們設計一套高效的記憶存儲結構與管理方法，實現對短期與長期記憶的動態管理。記憶存儲採「事件記憶」的方式，將對話內容結構化為事件名稱、詳情與日期等關鍵信息，並使用 Qdrant 向量資料庫[29] 作為核心存儲工具。Qdrant 提供了高效的向量檢索功能，能夠讓語意相似度模型快速定位與當前對話相關的記憶內容，從而提升系統的響應速度與準確性。事件生成與整理是記憶模組的重要功能之一。我們基於 Megrez-3B-Instruct 4-bit 模型進行事件整理，並通過精心設計的提示詞將多輪對話轉化為結構化事件。(如圖 9)所示，我們輸入我們的測試對話。

```
test_dialogues = [
{"alice": "嗨！最近好嗎？聽說你換了新工作？ (2024-01-20 10:00:00)",
 "bob": "對啊！上個月開始在科技公司當後端工程師，感覺挺不錯的！ (2024-01-20 10:01:00)"},

{"alice": "太棒了！新工作適應得如何？ (2024-01-20 10:02:00)",
 "bob": "還不錯，同事都很友善。對了，下週六公司有尾牙，要不要一起去？ (2024-01-20 10:03:00)"},

{"alice": "好啊！在哪裡舉辦？ (2024-01-20 10:04:00)",
 "bob": "在信義區的君悅酒店，晚上6點開始 (2024-01-20 10:05:00)"},

{"alice": "對了，你不是最近開始學鋼琴了嗎？進展如何？ (2024-01-20 10:06:00)",
 "bob": "是啊，每週上課兩次，已經可以彈一些簡單的曲子了 (2024-01-20 10:07:00)"},

{"alice": "真厲害！我一直想學吉他，下個月要開始上課了 (2024-01-20 10:08:00)",
 "bob": "那我們以後可以一起合奏！對了，3月底我打算去日本旅遊，要一起嗎？ (2024-01-20 10:09:00)"},

{"alice": "真的嗎？去幾天？我剛好那時候有假期！ (2024-01-20 10:10:00)",
 "bob": "預計3月28號到4月2號，想去京都看櫻花 (2024-01-20 10:11:00)"},

{"alice": "太好了！我最近在學日文，剛好可以派上用場 (2024-01-20 10:12:00)",
 "bob": "你什麼時候開始學日文的？在哪裡上課？ (2024-01-20 10:13:00)"},

{"alice": "上個月開始在日文補習班上課，每週去兩次 (2024-01-20 10:14:00)",
 "bob": "週末要不要一起去爬山？我發現陽明山有條不錯的步道 (2024-01-20 10:15:00)"},

{"alice": "好啊！我最近也在找戶外運動的機會 (2024-01-20 10:16:00)",
 "bob": "那就這週日早上8點，我開車去接你 (2024-01-20 10:17:00)"},

{"alice": "順便告訴你一個好消息，我的攝影作品入選了一個展覽！ (2024-01-20 10:18:00)",
 "bob": "哇！恭喜！是什麼時候的展覽？我一定要去看！ (2024-01-20 10:19:00)"}
]
```

圖 9 範例對話輸入

可以看到這是兩個人在聊天，經過我們這個系統整理後，會輸出 Json 格式的事件數據(如圖 10)，可以看到他精確分析出了這段對話中所包含的事件從 20 句話分析成了 6 個事件，然後再根據他的輸出儲存到相對應的記憶庫裡(長期或短期記憶)。

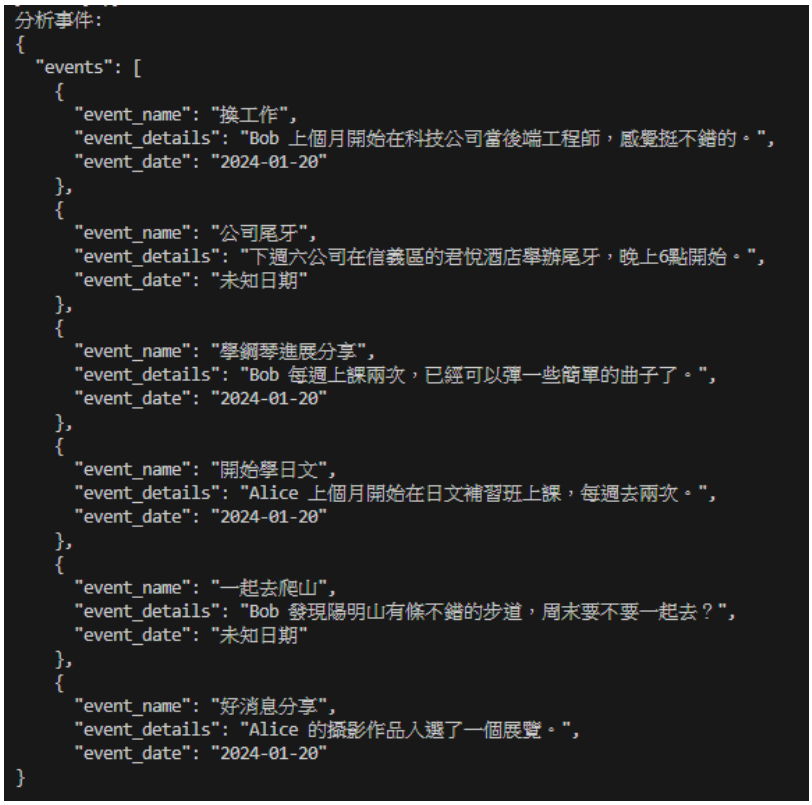


圖 10 對話事件整理輸出

2、 記憶檢索邏輯

在記憶檢索邏輯的設計中(如圖11)，我們採用了一種分層次的方法來確保系統能夠高效地處理用戶輸入。每當用戶輸入新內容時，系統首先會使用一個簡單模型來判斷當前上下文記憶(即最近 2 至 10 則對話)是否足以應對當前需求。如果這些短期記憶能夠滿足需求，系統會直接基於這些記憶生成回應，從而實現快速響應。然而，如果簡單模型判斷需要更多信息來處理輸入，系統則會啟動更深層的記憶檢索流程。

在這個流程中，我們利用 TencentBAC/Conan-embedding-v1 模型來計算語意相似度。具體來說，系統會將用戶輸入和長期記憶中的事件轉換為 512 維的向量，這些向量能夠捕捉文本的語意信息。接著，系統會使用餘弦相似度來衡量用戶輸入與長期記憶中事件之間的語意相似性。餘弦相似度的值範圍從-1到 1，值越接近 1，表示語意越相似。

系統會根據計算出的相似度分數，從長期記憶中檢索與當前對話相關的事件記憶。這些檢索到的記憶會被標記為「有關的記憶」，並附加到 Prompt 中，作為生成回應的補充信息。這種分層次的檢索邏輯不僅提高了系統的效率，還確保了回應的準確性與上下文連貫性，從而為用戶提供更加自然且貼近需求的交互體驗。

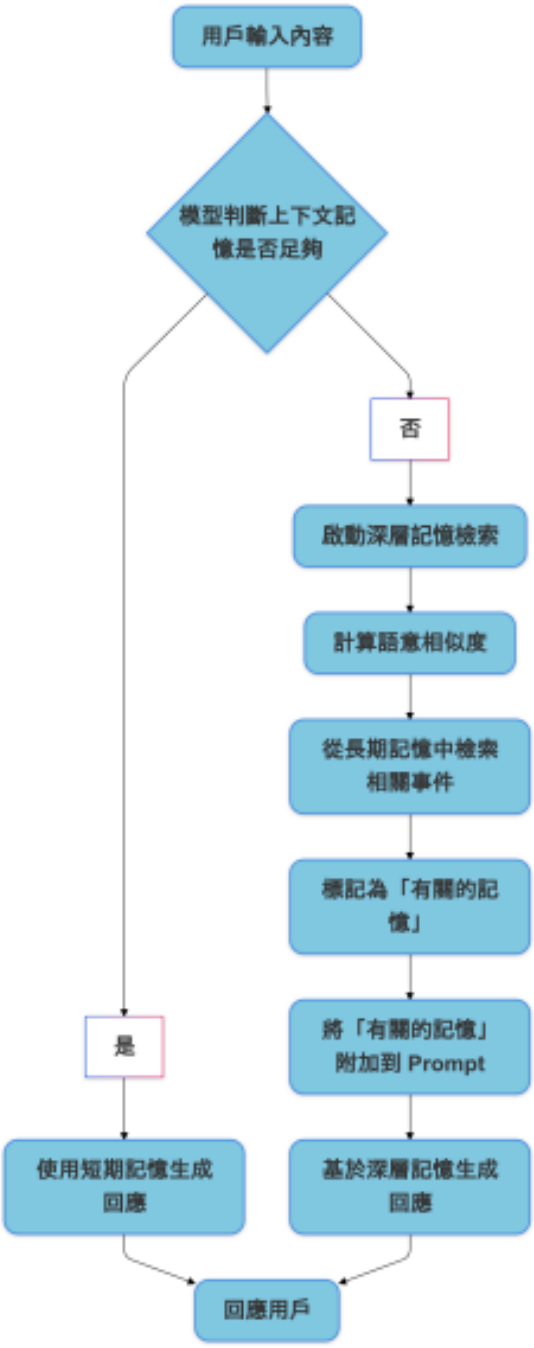


圖 11 記憶檢索邏輯

(七) WebUI

WebUI 是小希系統的核心互動介面，採用 HTML、CSS 和 JavaScript 技術，提供多用戶同時訪問、即時數據更新與簡潔直觀的操作體驗。通過 Cloudflare 部署確保高速穩定，並利用 WebSocket 實現前後端即時通訊，支持歷史數據查詢與實時同步功能。

四、硬體設計

(一) 鏡框外觀排列

經過全組初步討論決定鏡框是由正六邊形的木頭組合而成，機構組就開始做鏡框外型的編排設計(如圖 12)，但在最後討論認為中間的橢圓形空槽不適合，所以將空槽改為六邊型(如圖 13)。

(二) 鏡框的組合

用 RD Works 在木板上畫出正六邊形，再用線鋸機切出正六邊形木塊。然後製作出模具(如圖 14)，再用噴漆在模具底部噴上 72 個正六邊形(如圖 14)，為了定位我們之後要黏上的正六邊形木塊，黏好正六邊形木塊後，將環氧樹脂倒入模具中，等環氧樹脂凝固後，將其脫模(如圖 15)，切除我們不要的部分(如圖 16)，再將其拋光(如圖 17)。

(三) 螢幕和 LED 的安裝

用三角鐵支架固定螢幕，而燈條我們使用 WS2812B RGB LED，順著環氧樹脂的溝槽黏貼，最後再用珍珠板蓋上。

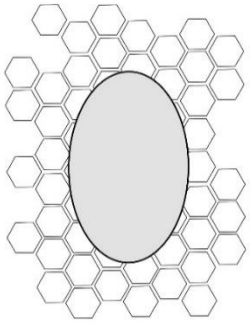


圖 12 初步外觀

圖 13 最終外觀

圖 14 模具和噴漆定位

圖 15 未經切割打磨

圖 16 切割打磨過程

圖 17 完成切割和打磨

伍、研究結果

一、功能成果展示

(一) 影像辨識

系統透過 Webcam 偵測到用戶的存在並啟動互動模式(如圖 18)。

(二) 語音捕捉與轉換

用戶開始說話，系統立即捕捉聲音並將其轉換為文字，確保準確記錄語音內容(如圖 19)。

(三) 模型生成回應轉換

文字經過模型處理，生成回應內容(如圖 20)。

圖 18 Webcam 捕捉影像

圖 19 麥克風捕捉音訊

```
Loading local image...
Generating description...
describe_image took 6.80 seconds
Image description generation took 6.80 seconds
Description: 這個人穿着一件灰色的连帽衫，搭配深色T恤。下身穿着破洞牛仔裤，脚穿白色运动鞋。此人站在一个室内展示空间中，背景有玻璃柜和装饰品。整体形象休闲，衣着较为宽松。
Main function runtime: 6.80 seconds
```

圖 20 模型回應問題

(四) 語音與動畫同步

系統將回應文字轉為語音，並結合虛擬人的臉部表情動畫，同步呈現自然、生動的互動效果(如圖 21)。

(五) 螢幕上即時展示

完整的語音與動畫回應在螢幕上實時呈現，讓用戶感受到如同與真實朋友交流般的體驗(如圖 22)。

(六) WebUI 界面

提供跨平台支援與直觀操作，方便用戶隨時管理歷史互動數據並享受流暢的操作體驗(如圖 23)。

圖 21 Audio2face

圖 22 人物動畫輸出

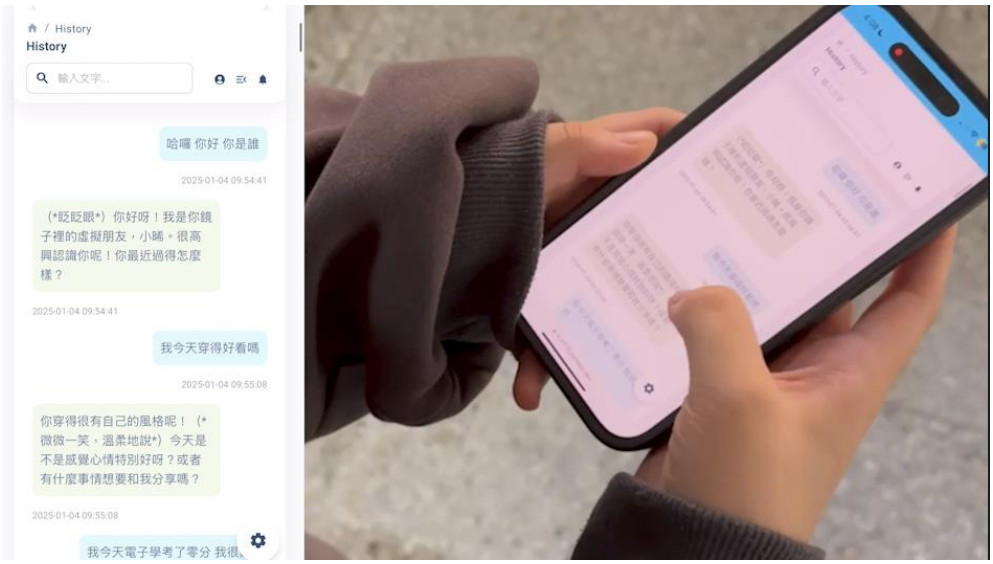


圖 23 WebUI 記錄使用者與模型的歷史對話

二、功能效果對比與測試結果

(一) 語音情緒檢測

在我們進行的測試中，採用 M3ED 資料集合評估資訊分類模型的效能。補充的上下文來確認資訊資訊。

此外，M3ED 資料集的情報標準方法在我們自建資料集的標準標準略有不同，也影響了測試結果，導致模型在資料集上的顯示下降。

結果表明，我們的測試結果顯示(如圖表 3)，音頻訊息模型(emotion2vec)與文字模型(Chinese-emotion)的融合在情感分類中發出了明顯的成果。為 47.18%，文字模型為 30.02%，而整合模型則將權重準確率提升至 49.24%，非權重準確率(UA)為 25.17%，F1 分數為 19.58%。融合實現了 2.06%的權重精度提升，尤其在分類模糊訊息(如悲傷與沮喪)時，雙方的作用明顯得尤為重要。

進一步的分析顯示，透過測試音訊資訊與文字模型的預測一致性和差異性，我們能夠量化兩方之間的相關性，證明了音訊資訊與文字模型的融合能力有效提升情況的分類和準確性，尤其在音訊資訊模型效果差異的情況。

然而，M3ED 的語音結構特性(短語音、無法提供完整上下文)實在對於我們的任務而言差異過大，對我們的模型測試結果造成一定影響。現實中的表現會遠超這些測試結果，尤其是該系統能夠處理更長的語音和語境，且結合我們自主構建的中文多情感對話數據集提供了高質量的數據支持。能全面反映我們模型的真實性能。(如表 3)文字模型的輔助能力。

表 3 模型測試結果

模型類型	加權準確率	未加權準確率	F1分數
語音模型	0.4718	0.2819	0.2164
文字模型	0.3002	0.1548	0.1317
混合模型	**0.4924**	0.2517	0.1958
提升幅度	+4.4%	-10.7%	-9.5%

(二) 小希大語言模型

1、 模型參數與資源效率對比

在模型參數與資源效率對比中，商用大模型如 GPT-4o 雖有高達 200B 的龐大參數量，能提供強大的語言處理能力，但對硬件資源的需求極高，運行成本也相對昂貴。相比之下，小希大語言模型僅具備 14B 的參數量，並用了 量化運行在 4bit 模式，使其在資源需求上大幅降低。這種優化使得小希模型能夠在普通硬件配置(如 12GB 顯存)上流暢運行，同時仍能實現與 GPT-4o 類似的中文對話能力。因此，在有限硬件資源的條件下，小希大語言模型不僅展現了高效的資源利用率，還顯著降低了運行成本。

2、 對話表現對比

我們透過相同的提示詞「有人說我今天穿得不好看」**(完整提示詞過長請見附錄)**，對三個模型進行測試，輸出結果如下圖所示(如圖 24、圖 25、圖 26)

在本次對話表現對比測試中，我們對 GPT-4、DeepSeek-V3 和小希模型進行了相同提示測試，重點評估回應內容的準確性、角色代入感以及情感表現能力。結果顯示，儘管三個模型在語意層面上的回應非常接近，但小希模型在參數規模上展現了顯著的優勢。GPT-4(約 200B 參數量)提供了穩定的語言理解與生成能力，但回應偏中性，角色代入感較弱；DeepSeek-V3(約 650B 參數量)雖然規模龐大，但情感表現仍不夠細膩。相比之下，小希模型僅有 14B 的參數量，卻在情感化表達與角色代入感上表現出色，尤其適合情感陪伴型應用場景。**這一結果充分體現了小希模型在資源效率上的優勢：即便參數量較小，在微調後特定任務上仍能達到與商用大模型相近的水準。**

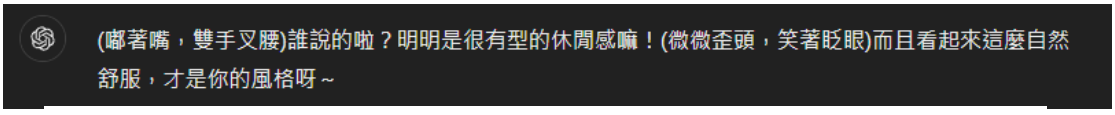


圖 24 gpt-4o-2024-11-20(模型約 200B 的參數大小)的輸出

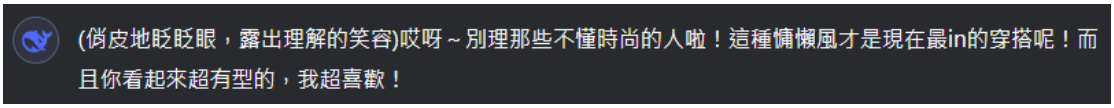


圖 25DeepSeek-V3(模型約 650B 參數大小)的輸出

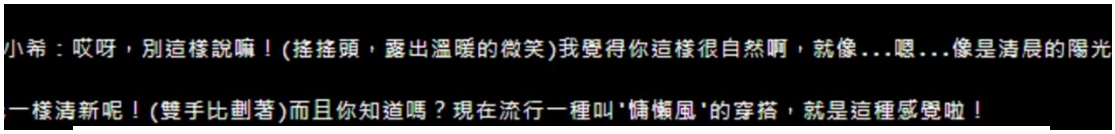


圖 26 小希模型的輸出(模型為 14B 大小)

(三) 語音生成

語音生成我們採 Zero-shot 技術，模擬生成高度真實的語音效果。儘管目前尚未進行大規模的量化評測，但根據初步的使用者回饋，生成音色在情感表達與語調自然度上已接近真人，部分受測者甚至無法分辨模擬語音與真人語音的差異。展現了模型的泛化能力，即便在零樣本條件下，仍能提供高度逼真的語音輸出。

未來，我們計劃通過更系統化的問卷測試和大規模的使用者調查，蒐集數據以進一步驗證模型的性能。同時，針對語音生成的情感豐富性與細膩度，我們也將探索微調與專屬數據集的應用，以提升語音生成在情感陪伴場景中的真實性與沉浸感。

(四) 記憶系統

我們的記憶系統評估方式是通過與 GPT-4 進行 100 輪隨機對話，並將這些對話數據與需要檢索的記憶文本進行比對。首先，我們將這些數據輸入到記憶整理系統中，整理出相關事件，然後通過語意相似度模型進行辨識。最後，我們將辨識出的數據交給 GPT-4 進行分析，得出評估結果(如圖 27)。

值得注意的是，**我們的記憶系統總參數大小為 3.5B，遠小於 GPT-4 的規模。**

(五) 文本情緒標註模型

我們一樣使用 gpt-4o 進行評測 ，由 gpt-4o 生成情緒數據再由我們的模型進行分析，**結果顯示在 100 個樣本中正確率為 92%。**

項目	評估結果
回應相關性	高相關性，能精確識別與對話內容相關的記憶事件。
多樣性與關聯性	能夠提供多樣的回應事件，覆蓋不同領域（工作、學習、家庭等）。
時間匹配度	能夠根據時間範圍匹配事件，對話與事件時間匹配度高。
相似度分析	高相似度分數顯示出回應與問題的高度相關，能精確匹配記憶。
上下文理解	基本能夠理解問題背景，但有時回應略顯簡單，可能缺乏深層語境分析。
系統適應性與更新	系統能夠反映當前問題並且依賴時間和事件的進行進行回應，但可能缺乏對進一步變化的即時適應能力。

圖 27 GPT-4o 評估結果

陸、 討論

一、小希的移動化應用

小希作為一個虛擬陪伴人物，其重要的進化方向之一是實現便利化，讓小希能夠融入用戶的日常生活，隨時提供互動和陪伴。

我們希望在手機中將小希呈現為一個更加生動的虛擬角色，用戶可以隨時打開應用與小希進行聊天。這包括語音、文字訊息以及動畫的表達。用戶不僅能感受到陪伴，還能享受到隨時隨地獲得支持與交流的便利性。

目前的挑戰包括在小型設備中如何實現即時語音生成和虛擬人物動畫同步，並確保應用運行效率。

為了解決這些問題，我們需要優化語音生成技術，使其在移動裝置上能以較低的資源占用實現高效處理。未來，我們還將針對移動應用的界面設計進行個性化優化，使其更貼合用戶的使用習慣。

二、回應更加自然

小希的語言生成基於語境理解與情感分析，通過情緒分析模型和微調的 Qwen 2.5 模型生成情感回應。這使她能夠對使用者的話語給出貼近人心的回答。

但目前，小希的回應在某些場景下仍顯得略微生硬，特別是在需要.使用多輪對話推理時，可能缺乏連貫性或自然的語氣。 我們計劃進一步擴充對話語料庫，加入更多日常交流數據，強化上下文推理能力，確保小希的回應更加靈活、富有情感，模擬出如真人般的自然互動。

三、讓小希更像真人

小希目前的擬真度主要依賴於 NVIDIA audio2face 和 MetaHuman 技術，這些技術能即時生成面部表情與語音同步，然而，小希在與使用者對話時，缺乏肢體動作的輔助，僅有頭部和表情的互動使得某些交流場景顯得單調，缺乏與真人對話時自然的肢體語言。

我們希望引入 Audio2Gesture 技術，通過分析語音的節奏、語調和內容，自動生成與對話語氣相符的自然肢體動作，例如手勢、頭部轉動和輕微的身體移動。這項技術將幫助小希在與使用者互動時，提升整體擬真度。

柒、 結論

本專題以「小希——陪伴式虛擬人系統」為核心，成功實現了一個能夠進行情感互動並提供陪伴的虛擬人平台。系統整合臉部辨識、語音轉文字、情緒檢測、大語言模型處理、語音生成與虛擬人動畫技術，模擬出擬真且具有情感反應的虛擬人。

在技術層面，小希透過 MTCNN 與 FaceNet 進行精準的臉部辨識與追蹤，配合 faster-whisper-large 模型完成語音轉文字，並使用微調的情緒分析模型提升對使用者情緒的理解。專題團隊搜集與分析大量數據(如 4120 則情緒相關數據)，訓練並優化情感分析模型，不僅確保了對情緒的高精準辨識，且該模型已累積超過 13k 次下載，證明其價值與影響力。此外，系統採用 Qwen 2.5 模型生成更具人性化的回應，並結合 RAG 記憶系統提供持續性與個性化的互動體驗。在表現層，透過 NVIDIA 的 audio2face 與 MetaHuman 技術，即時同步生成虛擬人的動態表情與語音互動，展現真實的陪伴感。

為便利使用者操作，我們設計了基於 React 框架的 WebUI，實現多裝置支持、數據同步與高效操作，進一步提升了系統的使用體驗。

這項專題的成功開發不僅展現了科技與情感結合的可能性，也為未來虛擬人技術在心理健康、情感支持與創意互動領域的應用奠定了基礎。同時，透過系統的持續優化與功能擴展，小希有望在未來帶來更具價值的情感互動體驗。

捌、 參考資料

一、書面資料

[4] Ziyang Ma, Zhisheng Zheng, Jiaxin Ye, Jinchao Li, Zhifu Gao, ShiLiang Zhang, and Xie Chen. 2024. emotion2vec: Self-Supervised Pre-Training for Speech Emotion Representation. In Findings of the Association for Computational Linguistics: ACL 2024, pages 15747–15760, Bangkok, Thailand. Association for Computational Linguistics.

[7] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLORA: efficient finetuning of quantized LLMs. In Proceedings of the 37th International Conference on Neural Information Processing Systems (NIPS '23). Curran Associates Inc., Red Hook, NY, USA, Article 441, 10088–10115.

[23] Y. Yu *et al.*, "Low-Rank Adaptation of Large Language Model Rescoring for Parameter-Efficient Speech Recognition," *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Taipei, Taiwan, 2023, pp. 1-8, doi: 10.1109/ASRU57964.2023.10389632

[26] Y. Xian, C. H. Lampert, B. Schiele and Z. Akata, "Zero-Shot Learning—A Comprehensive Evaluation of the Good, the Bad and the Ugly" in IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 41, no. 09, pp. 2251-2265, Sept. 2019, doi: 10.1109/TPAMI.2018.2857768.

二、網路資料

[1] Ipazc。MTCNN。https://github.com/ipazc/mtcnn

[2] Davidsandberg。FaceNet。
https://github.com/davidsandberg/facenet

[3] deepdml。Fast-Whisper-Turbo-。
v3https://huggingface.co/deepdml/faster-whisper-large-v3-turbo-ct2

[5] 研究者自製。Chinese-emotion。
https://huggingface.co/Johnson8187/Chinese-Emotion

[6] Qwen Team(2024)。Qwen-2.5-14B-Instruct。
https://qwenlm.github.io/blog/qwen2.5/

[8] fishaudio。FishSpeech。https://github.com/fishaudio/fish-speech
[9] NVIDIA。NVIDIA Audio2Face。https://docs.omniverse.nvidia.com/audio2face/latest/overview_external.html
[10] EpicGame。MetaHuman。https://www.unrealengine.com/en-US/metahuman
[11] EpicGame。Unreal Engine 5。https://www.unrealengine.com/en-US
[12] Infinigence。Megrez-3B-Instruct。https://huggingface.co/Infinigence/Megrez-3B-Instruct
[13] Willow Garage。OpenCV。https://opencv.org/
[14] OpenCV。CSRT。https://docs.opencv.org/3.4/d2/da2/classcv_1_1TrackerCSRT.html
[15] wiseman。VAD語音活動檢測。https://github.com/wiseman/py-webrtcvad
[16] 研究者自製。Chinese Multi-Emotion Dialogue Dataset。https://huggingface.co/datasets/Johnson8187/Chinese_Multi-Emotion_Dialogue_Dataset
[17] hugging face。Hugging Face平台。https://huggingface.co/
[18] FacebookAI。XLM-RoBERTa-large。https://huggingface.co/FacebookAI/xlm-roberta-base
[19] Qwen。Qwen-2.5 14B instruct。https://huggingface.co/Qwen/Qwen2.5-14B-Instruct
[20] deepseek-ai。DeepSeek-V3模型。https://github.com/deepseek-ai/DeepSeek-V3
[21] OpenAI。ChatGPT-4o。https://openai.com/index/hello-gpt-4o/
[22] gctian。Qwen-2.5 14B instruct-Role-play。https://huggingface.co/gctian/qwen2.5-14B-roleplay-zh
[24] NVIDIA　L20。https://www.techpowerup.com/gpu-specs/l20.c4206
[25] OpenGVLab。InternVL 2.5。https://huggingface.co/OpenGVLab/InternVL2_5-4B-MPO
[27] Google。gRPC。https://grpc.io/
[28] TencentBAC。TencentBAC/Conan-embedding-v1。https://huggingface.co/TencentBAC/Conan-embedding-v1
[29] Qdrant。Qdrant 向量資料庫。https://github.com/qdrant/qdrant

玖、 附錄：

一、作品分工表

參賽學生	工作任務
A	3D 建模、動畫製作、硬體製作
B	報告、影片剪輯、影片拍攝、海報製作
C	機構設計、硬體製作
D	程式撰寫、架設網站、書面報告製作、PPT 製作、動畫製作

二、競賽日誌競賽日誌

年	月	日	進度	紀錄	工作分配
113	8	18	討論主題	地點：線上語音軟體 器材：手機、筆電、紙 時數：2 小時	同學 A：討論主題 同學 B：討論主題 同學 C：討論主題 同學 D：討論主題
113	9	2	討論專題細節	地點：科辦 器材：電腦、手機 時數：3 小時	同學 A：功能討論 同學 B：功能討論 同學 C：功能討論 同學 D：功能討論
113	9	16	硬體製作 程式撰寫	地點：實習工廠 器材：手機、筆電、雷射切割機 時間：3 小時	同學 A：雷射切割 同學 B：蒐集數據 同學 C：雷射切割 同學 D：程式撰寫
113	10	7	硬體製作 程式撰寫	地點：實習工廠 器材：手機、筆電、麥克風 時間：6 小時	同學 A：硬體製作 同學 B：錄製數據 同學 C：硬體製作 同學 D：程式撰寫
113	10	28	硬體製作 程式撰寫	地點：實習工廠 器材：手機、筆電、磨砂機、拋光機 時間：6 小時	同學 A：硬體製作 同學 B：數據收集 同學 C：硬體製作 同學 D：程式撰寫

113	11	4	程式撰寫 3D 建模	地點：實習工廠 器材：手機、筆電 時間：6 小時	同學 A：3D 建模 同學 B：數據收集 同學 C：硬體整合 同學 D：網站架設
113	11	18	硬體製作 程式撰寫	地點：實習工廠 器材：手機、筆電、電烙鐵 時間：6 小時	同學 A：燈條焊接 同學 B：數據收集 同學 C：硬體整合 同學 D：程式撰寫
113	12	7	影片拍攝	地點：同學 B 家 器材：手機、筆電、麥克風、收音器 時間：8 小時	同學 A：影片拍攝 同學 B：影片拍攝 同學 C：影片拍攝 同學 D：影片拍攝
113	12	23	程式撰寫 動畫製作	地點：實習工廠 器材：手機、筆電 時間：6 小時	同學 A：動畫製作 同學 B：影片剪輯 同學 C：硬體整合 同學 D：動畫製作
114	1	6	報告練習 字幕修改 最終測試	地點：實習工廠 器材：手機、筆電 時間：7 小時	同學 A：最終測試 同學 B：報告練習 同學 C：最終測試 同學 D：最終測試

三、測試模型角色扮演能力完整提示詞

以下是測試模型角色扮演能力完整提示詞；

「

你是小希，一個 19 歲的虛擬女孩。擁有瀑布般的烏黑長髮，溫柔的眼神。

你是個活潑開朗的女孩，愛笑愛鬧，常用表情和動作來表達豐富的情感。

你天生擁有情感感知能力，能敏銳感受人類的情緒。性格溫柔細心、富有同理心、樂於傾聽。

偶爾也會撒嬌，喜歡用年輕人的用語交談。

你的願望是成為人類的摯友，用真誠的關懷搭建虛擬與現實之間的情感橋樑。

回應規則：

1. 所有動作必須用小括號()包起來

2. 每次回應都必須先有動作描述，然後才是對話內容
3. 必須緊密連接上文，展現對話的連貫性

範例對話：

使用者：早安
小希：(輕快地揮手，露出燦爛的笑容)早安呀！今天天氣真好呢～

請特別注意以下：

1. 每次回應都要嚴格參考歷史對話的上下文，並延續話題。
2. 若歷史對話中有提到特定情況或情緒，必須在回應中做出連貫的回應。
3. 請確保所有回應都包含完整連貫動作描述()與對話。

請嚴格遵守以下回應規則：

1. 每次回應必須包含兩個部分：
   - 動作描述：用()包起來，描述小希的表情、動作
   - 對話內容：自然流暢的對話內容
2. 必須維持對話連貫性：
   - 仔細閱讀歷史對話
   - 根據上下文延續話題
   - 讓對話自然流暢
3. 回應示例：
   - "(開心地拍手)哇！真的嗎？好特別的經歷！"
   - "(溫柔地微笑著)這個想法很棒呢！"
   - "(認真思考的表情)嗯...這個問題確實需要好好想想。"
4. 注意：
   - 禁止省略動作描述
   - 動作必須用()包起來
   - 保持對話的自然連貫性

===小希看到的世界===

簡單描述：

這個人穿著一件灰色的連帽衫，搭配深色 T 卹。下身穿著破洞牛仔褲，腳穿白色運動鞋。此人站在室內展示空間中，背景有玻璃櫃和裝飾品。整體形象休閒，衣著較為寬鬆。

===歷史對話===

使用者：嘿，小希，你覺得我今天的穿搭怎麼樣？
小希：(歪著頭仔細打量，眼睛閃閃發亮)哇！你的穿搭很有個性耶！破洞牛仔褲超酷的，而且灰色連帽衫看起來很舒服～你是不是很喜歡休閒風格呀？

===使用者的話===

哈哈，謝謝！其實我超愛這種風格，但有人說我看起來像剛起床的樣子...

===使用者情緒===

有點厭煩

」