

實務專題報告（精簡版）

一、 專題主題

Asar 聊天機器人設計平台

二、 簡介

本專題旨在開發一款名為 Asar 的輕量級聊天機器人設計平台，將聊天機器人與樹莓派結合，提供一條龍的聊天機器人服務部署流程。開發者僅需一片微型單板電腦，就能輕鬆創造專屬於自己的聊天機器人。

三、 研究動機與目的

我們發現部署聊天機器人服務的流程相當複雜，且開發者需要掌握多種程式語言及專業技術才有能力完成，因此，本專題將開發一款名為 Asar，運作於樹莓派的輕量級聊天機器人設計平台，提供一個輕鬆部署聊天機器人服務的解決方案，Asar 有下列目標：

1. 簡化部署流程
2. 具備自然語言理解能力（NLU）的聊天機器人
3. 提供視覺化設計工具
4. 輕鬆控制樹莓派周邊設備及串接外部 API
5. 提高隱私，降低敏感資訊外流的風險

四、現有相關研究概況及比較

(一) 聊天機器人的 NLU 流程

根據 Rasa 論文[1]、Rasa Blog 文章[2]的研究，NLU 流程依序為斷詞、特徵化、實體提取、意圖識別、回覆決策，各個步驟的功能如下：

1. 斷詞 (tokenize)：將輸入的句子分割成單詞。
2. 特徵化 (featurize)：將單詞轉換成向量或數值，以便電腦進行處理和分析。
3. 實體提取 (entity extraction)：從句子中提取重要的訊息，例如：人名、地點。
4. 意圖識別 (intent detection)：判斷使用者在對話中的意圖或目的。
5. 回覆決策 (response selection)：根據使用者的意圖，做出相對應的回覆。

(二) 聊天機器人的設計模式

現行的商業聊天機器人開發平台，如 Google Dialogflow CX、Amazon Lex。皆採用以「邊與節點」為概念的視覺化設計模式來製作聊天機器人，概念包括流程圖、有向圖等。

(三) 視覺化程式語言

視覺化程式語言 (Visual programming language) 是一種以「方塊」為概念的程式語言，讓使用者透過圖形化元素進程式設計，相較於文字式程式語言更為直覺、簡單，因此有許多面向初學者的程式設計工具採用視覺化程式語言，例如 Scratch、MIT App Inventor。

五、系統設計與成果

(一) 系統架構

Asar 平台採用容器化的設計，實現快速部署。並依據功能分成四個子服務，包括前端網頁服務、後端 API 服務、聊天機器人服務、動作代理服務，系統架構如圖 1。

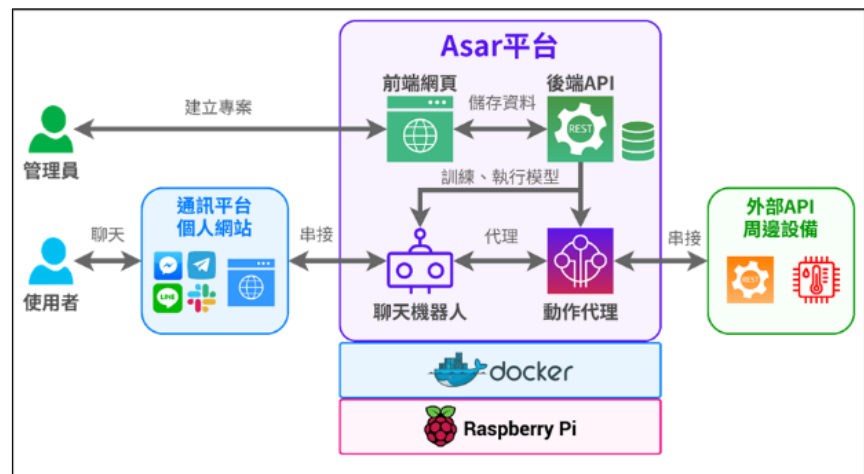


圖 1 Asar 系統架構

(二) 前端網頁服務

前端網頁服務使用 Vue 框架設計，採用 Prime Vue、Vue flow、Google Blockly 製作視覺化設計工具。設計工具包括：專案管理工具、訓練資料管理工具、視覺化劇本設計工具、視覺化程式設計工具、內建聊天室。

1. 專案管理工具：用於管理聊天機器人專案，提供一鍵訓練與執行對話模型的功能。

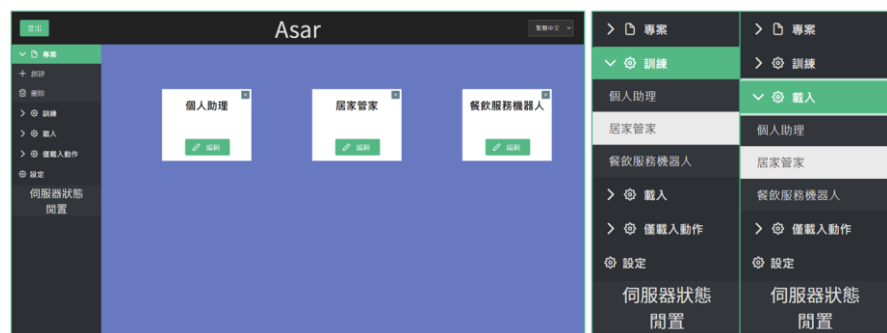


圖 2 專案管理工具

介面（左）；訓練與載入模型（右）

2. 訓練資料管理工具：提供人性化的介面，協助管理者建立訓練資料。

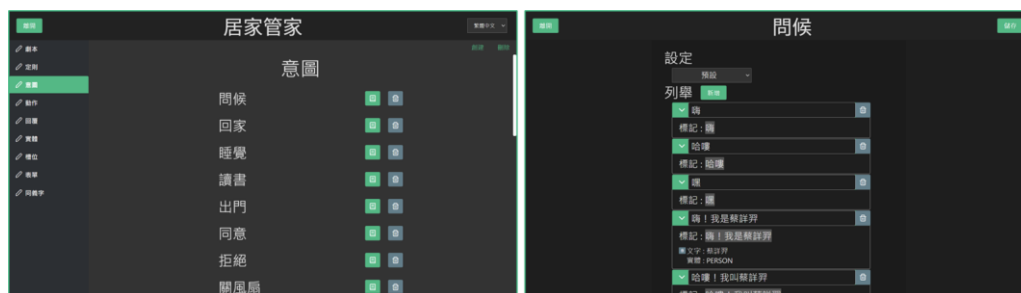


圖 3 訓練資料管理工具

選單 (左)；編輯器 (右)

3. 視覺化劇本設計工具：提供以流程圖為概念的設計工具，讓開發者能以直觀的對話流程思維來設計劇本。



圖 4 視覺化劇本設計工具

所有節點 (左)；劇本範例 (右)

4. 視覺化程式設計工具：提供以方塊為概念的程式設計工具，讓管理員能以簡單的方式來控制樹莓派的周邊設備、串接外部 API



圖 5 視覺化程式設計工具

所有程式方塊 (左)；程式範例 (右)

5. 內建聊天室：提供內建的聊天室，讓開發者能實時跟聊天機器人對話，檢查其回覆內容的準確性，並在必要時進行調整，以確保聊天機器人的行為符合預期。

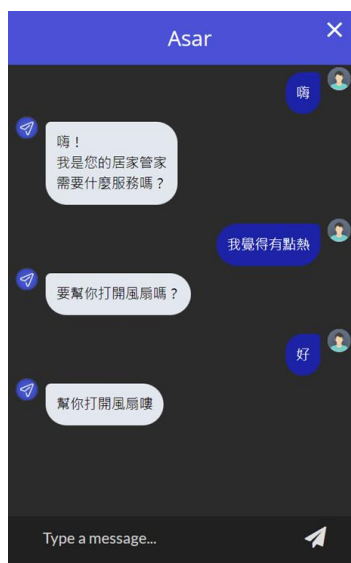


圖 6 內建聊天室

(三) 後端 API 服務

後端 API 服務使用 Flask 框架設計，SQLite 作為資料庫管理系統，並採用 Jinja 模板引擎來開發程式碼轉換工具。

本服務負責接收管理員從前端網頁提供的訓練資料，並儲存至後端檔案系統和資料庫。在訓練模型時，後端 API 服務會先將訓練資料轉換成特定的資料格式，再傳送給聊天機器人服務進行訓練。



圖 7 訓練資料轉換

轉換聊天劇本（左）；轉換程式方塊（右）

(四) 聊天機器人服務

聊天機器人服務使用 Rasa 機器學習框架開發，自然語言理解的流程設計如圖 8。斷詞、特徵化和實體提取分別使用中研院 CKIP Transformers 的斷詞器、ALBERT 語言模型[4]、實體提取器來處理；意圖識別和回覆決策分別使用 DIET (Dual Intent and Entity Transformer) [5]技術、TED Policy (Transformer Embedding Dialogue Policy) [6]技術來達成。



圖 8 自然語言理解流程

本服務提供了各大通訊平台的接口，簡化將聊天機器人整合至聊天室的步驟。此外，聊天訊息的推論完全在本地端執行，不依賴雲端運算，從而降低敏感資訊外流的風險。

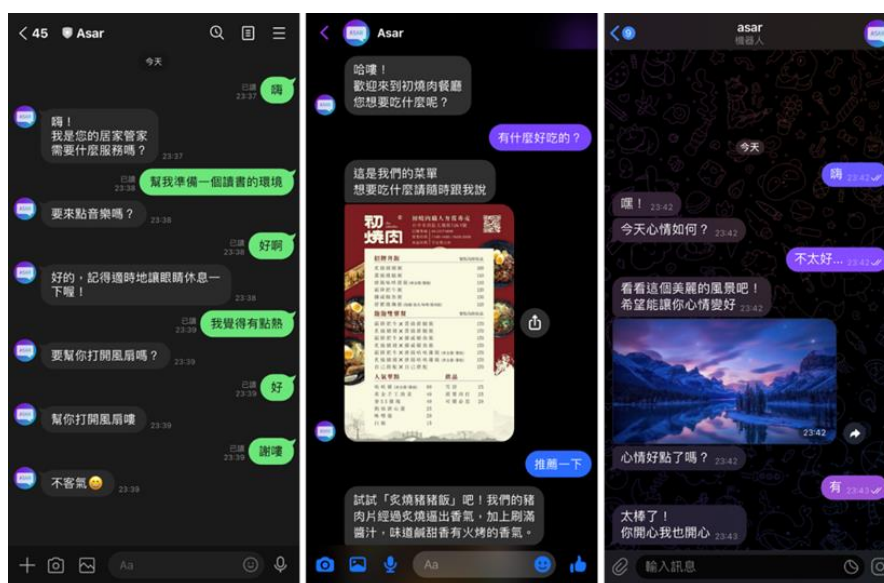


圖 9 各大通訊平台實例

LINE (左)；Facebook (中)；Telegram (右)

(五) 動作代理服務

動作代理服務基於 Rasa SDK 開發，Rasa SDK 屬於 Rasa 機器學習框架生態系的一部份，用於設計非對話型任務。非對話型任務包括呼叫外部 API、控制樹莓派周邊設備，如果使用者在聊天過程中試圖執行非對話型任務，聊天機器人服務會請求動作代理服務來代理執行。

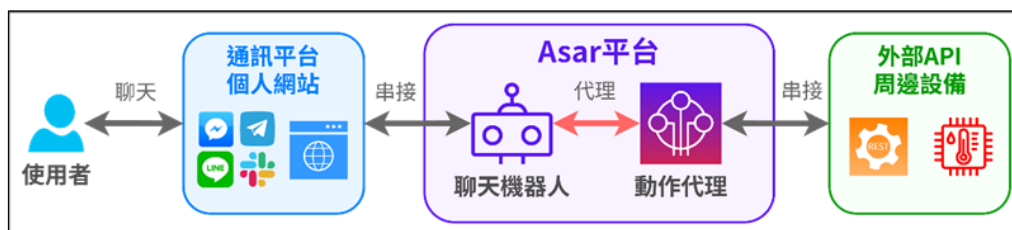


圖 10 動作代理流程

六、相關文獻

- [1]. Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. Rasa: Open Source Language Understanding and Dialogue Management. arXiv preprint arXiv:1712.05181. 2017.
- [2]. Vincent Warmerdam. Computing machinery and intelligence. Bending the ML Pipeline in Rasa 3.0. Rasa. <https://rasa.com/blog/bending-the-ml-pipeline-in-rasa-3-0/>. 2021.
- [3]. CKIP Transformers <https://github.com/ckiplab/ckip-transformers>
- [4]. Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942. 2019.
- [5]. Tanja Bunk, Daksh Varshneya, Vladimir Vlasov, and Alan Nichol. Diet: Lightweight language understanding for dialogue systems. arXiv preprint arXiv:2004.09936. 2020.
- [6]. Vladimir Vlasov, Johannes EM Mosig, and Alan Nichol. Dialogue transformers. arXiv preprint arXiv:1910.00486. 2019.
- [7]. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems. arXiv preprint arXiv:1706.03762. 2017.