致理科技大學

資訊管理系專題報告

使用 NLP 技術製作 LINE 點餐系統

LINE Ordering System with NLP Technology

專題生：
（60910135）方鈺穌
（60910189）周奕丞
（60910199）洪千庭
（60910161）陳冠閔
（609101A6）陳幼貴

指導教授：陳瑋弘 老師

中華民國 113 年 5 月

致理科技大學

資訊管理系

畢業專題

使用ＮＬＰ技術製作ＬＩＮＥ點餐

一一二學年度

致理科技大學

授權書

本授權書所授權之專題報告在致理科技大學

112 學年度第 2 學期所撰寫。

專題名稱：使用 NLP 技術製作 LINE 點餐系統

本人具有著作財產權之論文或專題提要，授予致理科技大學，得重製成電子資料檔後收錄於該單位之網路，並與台灣學術網路及科技網路連線，得不限地域時間與次數以光碟或紙本重製發行。

本人具有著作財產權之論文或專題全文資料，授予教育部指定送繳之圖書館及本人畢業學校圖書館，為學術研究之目的以各種方法重製，或為上述目的再授權他人以各種方法重製，不限時間與地域，惟每人以一份為限。並可為該圖書館館藏之一。

本論文或專題因涉及專利等智慧財產權之申請，請將本論文或專題全文延至民國 113 年 6 月 31 日後再公開。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。

(上述同意與不同意之欄位若未勾選,本人同意視同授權)

✓同意　不同意

學生簽名：陳冠閔 陳幼貴 方鈺穌 洪千庭 周奕丞

(親筆正楷簽名)

指導老師姓名：陳瑋弘

(親筆正楷簽名)

中華民國 113 年 6 月

摘要

專題報告名稱：使用 NLP 技術製作 LINE 點餐系統　　頁數：46

校系別：致理科技大學資訊管理系

完成時間：112 學年度第 2 學期

專題生：方鈺穌 周奕丞 陳冠閔 洪千庭 陳幼貴

指導教授：陳瑋弘

關鍵詞：**NLP、點餐機器人、Line 機器人、機器學習、Azure**

為因應少子化及缺工問題並減少消費者的點餐等候時間，我們開發一個結合 Line 及 AI 的點餐機器人系統，使消費者更加便利同時也節省人力成本，達成雙贏。

此機器人是以 Azure CLU 用做模型訓練 + [Azure App severs + Azure BOT + Line bot 為後端]並以 Line 作為客戶端收發訊息。

具體流程是 LINE 端發出訊息後由 LINE BOT 先接收後透過 Webhook 傳送到建立在 Azure App severs 上面的 Azure BOT 服務並呼叫 Azure CLU 分析語句後返回相應語句，如此往復直至完成餐點的確認，然後將結果整合成一份訂單在發送到餐廳端，餐點做好時 Line Bot 再通知消費者取餐。

ABSTRACT

Thesis Title：LINE Ordering System with NLP Technology Pages：46

University：Chihlee University of Technology

Graduate School：Departmant of Information Management

Date：May, 2023 Degree：Master

Researcher：Yu-Wei Fang, Yi-Cheng Chou, Guan-Min Chen, Qian-Ting Hong, You-Gui Chen

Advisor：Wei-Hung Chen

Keywords：**NLP, ordering robot, Line robot, machine learning, Azure**

To address the challenges of a declining birthrate, labor shortages, and to reduce waiting times for consumers, we developed a ordering robot system that integrates Line and AI. This system not only provides greater convenience for consumers but also helps to save labor costs, achieving a win-win situation. The robot is trained using Azure CLU for model training + [Azure App servers + Azure BOT + Line bot as the backend], with Line used as the client for message sending and receiving. The specific process is that after a message is sent from the LINE client, the LINE BOT receives it first, then sends it to the Azure BOT service established on Azure App servers via Webhook, which calls Azure CLU to analyze the sentence and returns the corresponding response. This process repeats until the confirmation of the order is completed, then the result is integrated into an order and sent to the restaurant side. When the food is ready, the Line Bot notifies the consumer to pick up the food.

誌謝

在這裡要由衷感謝指導老師陳瑋弘的悉心指導與支持，您的專業知識和不懈努力讓我們受益匪淺。您的指導讓我們在專題製作的過程中獲得了很大的成長和進步，深深地感謝您的付出。

同時，也要感謝每一位專題組員的努力和貢獻。是你們的協作、合作和努力讓這個專題得以順利完成。我們共同攜手克服了種種困難，完成了專題報告，這將是我們共同的成就和回憶。再次感謝大家的努力和付出！

陳冠閔、周奕丞、洪千庭、陳幼貴、方鈺穌 謹致

致理科技大學 資訊管理 學士班

中華民國 113 年 6 月

目錄

圖目錄

表目錄

第壹章 緒論

本研究的主要方向是利用自然語言處理（NLP）技術來解決 LINE 餐點訂單的問題。對於餐廳而言，與傳統的電話點餐相比，LINE 點餐機器人具有許多優勢。首先，LINE 點餐機器人可以同時提供一對多的服務能力，不需要像電話點餐那樣一一接聽用戶的來電，從而大幅節省了電話費用和人力成本。同時，LINE 點餐機器人還能更有效地利用人員的使用率，因為一個店員可以同時處理多個用戶的請求，提高了效率。

第一節 研究背景

隨著科技的進步，走在路上幾乎每個人手上都會有一隻手機，一旦有時間就會拿著手機不停的滑，從台灣最熱門即時通訊 app 推薦排行報告中可以發現 LINE 是在台灣受眾最多的即時通訊軟體，當地用戶人數僅次於 YouTube 和 Facebook。

人氣度排名	即時通訊App	功能	安全性	界面	總評
1	LINE	4.5	4.5	4	4.3
2	Facebook Messenger	4	4	4.5	4.1
3	Telegram	5	4	4.5	4.5
4	WhatsApp	4.5	4.5	4.5	4.5
5	WeChat	4	3	4.5	3.8
6	Skype	4.5	3.5	4.5	4.1
7	Signal	4.5	5	4.5	4.6
8	Discord	4	4	4	4
9	Snapchat	4.5	4.5	4.5	4.5

圖 1-1 2023 台灣最熱門即時通訊 app

LINE 能在台灣收穫龐大用戶群體，除了受到它進入台灣市場的有利時機影響，原因還在於 LINE 使用體驗符合年輕人及老年人需求。

LINE BOT 擁有直覺和簡單的使用介面，使得使用者能夠輕鬆地與訂單機器人進行互動。這種便利的使用體驗有助於提高客戶滿意度，促進客戶的再次購買和口碑傳播。考慮到 LINE 使用者眾多的優勢，建立一個 LINE BOT 訂單機器人是一個具有吸引力且實用的方式，可為商家提供更便利的訂單管理和客戶服務。

LINE BOT 能夠實現即時互動，讓商家能夠快速回答客戶的問題並提供支持。

同時，LINE 的通知功能可以讓商家向客戶發送訂單狀態更新、促銷活動等重要信息，增加客戶的參與度和購買意願。

第二節 研究動機

不論是在平日上班日或者是假日放假時段，只要在電話點餐時，店家都需要有多一份人力去接聽以及記錄消費者所需的餐點。導致當店家在尖峰時段可能會有人手不足的問題，需要減少做餐及出餐速度去接聽電話。具體而言本專題研究動機有以下幾種：

動機一

隨著現在手機越來越普及，而每個人的手機內都有這各種不同類型 App，也有所謂的外送平台 如：Foodpanda、Honestbee、Uber Eat 等，而這些也替我們的生活帶來了更多的便利，想找什麼只需要拿出手機，點開 App 就能做到。但也因為 Line 在台灣的普及率很高，因此如果使用 Line Bot 進行開發， 也可以增加使用意願。

動機二

可以幫助店家減輕在於工作尖峰時段不必再多一個人力去處理電話點餐，同時也可以幫助消費者，在上班日的上班時段不用與其他也要進行電話點餐外帶的消費者進行電話的搶播。

動機三

減少消費者的點餐等候時間，使消費在餐點做好前可以安排自己的事情，餐點做好時 LINE BOT 再通知消費者取餐。也可以避免店裡的客人太多在店裡壅擠的等候取餐。

第三節 研究目的

本研究的目的是開發一個使用 NLP 技術製作 LINE 點餐系統的點餐系統，以改善餐廳在尖峰時段的人力短缺問題，同時提高消費者的點餐體驗和餐點等候時間的效率。本研究整理出的的研究目的有以下三種：

（一）系統開發：開發一個使用 NLP 技術製作 LINE 的點餐系統，使消費者可以透過 LINE 應用程式進行點餐、選擇餐點、修改訂單等操作。

（二）自動化訂單處理：設計系統能夠自動處理消費者的點餐指令，並接收訂單，減少人力處理的需求。

（三）即時互動與推薦功能：開發系統具備即時互動的能力，能夠回答消費者的問題、提供餐點推薦等個性化服務，增加消費者的參與度和滿意度。

第貳章 文獻探討

第一節 人工智慧聊天機器人

一、定義

聊天機器人是一個可以對自然語言輸入做出回應的程式，並試圖以模真人的方式進行對話，能夠與使用者進行自然語言交流並提供相關的回應和服務。聊天機器人的研究可以追溯到英國數學家和計算機科學家阿倫·圖靈（Alan Turing）的工作。在 1950 年，圖靈提出了一個著名的思想實驗，即所謂的「圖靈測試」（Turing Test），用來評估機器是否能夠表現出與人類相似的智能。

根據圖靈的想法，如果一個機器能夠在對話中表現得足夠像一個人，以至於無法被區分出是機器還是真實人類，那麼這個機器就可以被認為具有智能。這個想法促使了聊天機器人的研究，人們開始探索如何創建能夠進行自然語言對話的機器。隨著時間的推移和科技的發展，研究人員和工程師們利用人工智慧、機器學習和自然語言處理等技術來開發更先進的聊天機器人。這些技術使得聊天機器人能夠更好地理解和回應使用者的語言，並提供更智能化和有用的對話體驗。

因此，圖靈的工作對聊天機器人的研究起到了重要的啟發作用，他的想法和測試標準成為了評估聊天機器人智能程度的基準之一。

二、應用

聊天機器人是一種能夠模擬對話的人工智能應用程式，在各個領域都有廣泛的應用。它的使用可以提供許多重要的好處，並且能為不同的行業帶來創新和效率。

目前聊天機器人被廣泛用於社群網站及通訊軟體上，其提供服務以幫助使用 者達成各種目的，如提供資訊、社會互動、娛樂或將使用者導入其他連結或系統（Brandtzaeg & Følstad, 2017）。使用聊天機器人已在各個行業中逐漸成為提供 使用者幫助的常規功能，使用者可以透過聊天介面使用日常語言得到有意義的答案（Crutzen, Peters, Portugal, Fisser & Grolleman, 2011）。

第二節 NLP

一、定義

自然語言處理英文全名為 Natural Language Processing，是一門涉及計算機科學、人工智慧、語言學和語音學等多個學科的交叉領域。它的目的是為了使計算 機能夠理解、分析、生成和操作人類自然語言。 自然語言是人們在日常生活中使用的語言，它包括口語和書面文字，並且具 有很高的多樣性和複雜性。與其他形式的數據相比，自然語言具有很高的歧義性、 不確定性、主觀性和文化背景等方面的特點，這也是 NLP 面臨的挑戰所在。 NLP 的最新進展因其在語言建模方面的效率而受到廣泛關注。這些語言模型 正在各個行業中尋找應用，因為它們為實時、可靠和語義導向的文本分析提供了強大的機制 (Mishev, Gjorgjevikj, Vodenska, Chitkushev, Trajanov, 2020)。

自然語言處理包含的範圍相當廣泛，包括 : 斷詞 （word segmentation）、詞性標記（part-of-speech tagging）、專有名詞標記（name entity tagging）、詞義消歧（word sense disambiguation）、代名詞釋義（pronoum resolution）、句法剖析、文法比對、語意角色標註（semantic role labeling）、語意邏輯推論、自動音譯、機器翻譯、語音辨識、語音合成等（國家教育研究院 自然語言處理）。

以下為 NLP 常見的用途：

1. 語音識別：指將人類語音轉換為文字的過程。通過聲音訊號處理、語音特徵提取和語音識別模型訓練等技術，使得機器能夠理解人類的語音指 令或對話，將其轉化為可供計算機處理的文字訊息。語音識別在日常生 活中已經得到廣泛應用，例如語音助手、智慧家庭、電話自動化客服等 等。

2. 文本分類：指將文本資料按照預先定義好的分類方式進行歸類的過程。通過語言學特徵提取和機器學習等技術，使得機器能夠自動識別文本的 內容，將其歸入相應的類別中。文本分類在資訊檢索、文本分析、訊息 過濾、垃圾郵件分類、情感分析等領域有著廣泛應用。

3. 情感分析：指對文本、語音等資料進行情感語義分析的過程，通常是將文本的情感極性（如正向、負向、中性）進行自動標記或評分。情感分 析通過語言學知識表示、機器學習等技術，使得機器能夠識別和分析文 本中的情感內容，進而判斷該文本表達了何種情感或情感極性。情感分 析在市場研究、社交媒體分析、情感監測、商品評價分析等領域有著廣 泛應用。

4. 訊息擷取：指從非結構化的自然語言資料中自動抽取出所需的訊息的過程。通過語言學特徵提取、實體識別、關係識別等技術，使得機器能夠 自動從文本中識別出相關的實體、關係等重要訊息，並進行分類、歸納、綜合等處理，提煉出高度概括性的知識。訊息擷取在自動化報告生成、 自動化摘要生成、知識圖譜構建等領域有著廣泛應用。

5. 問答系統：指利用 NLP 技術，使得機器能夠理解和回答自然語言的問題。

問答系統的核心是自然語言理解和訊息檢索技術，通過理解問題、擷取 相關訊息、選取最佳答案等技術，使得機器能夠提供準確、高效的答案。 問答系統在智慧化客服、智慧化語音助手、智慧化搜索等領域都有著廣 泛的應用。

6. 機器翻譯：指通過 NLP 技術，將一種自然語言的內容轉換成另一種自然語言的內容。機器翻譯的過程主要包括語言分析、語言生成等步驟。通 過語言學特徵提取、統計機器翻譯、神經機器翻譯等技術，使得機器能 夠自動識別不同語言間的語義和結構差異，實現自動化的翻譯。機器翻 譯在國際交流、跨語言訊息檢索、多語言翻譯等領域都有著廣泛的應用。

7. 自然語言生成：指利用 NLP 技術，將非自然語言形式的訊息轉換成符合自然語言語法和語義的自然語言文本的過程。自然語言生成的過程主要 包括語義解析、句法分析、生成文本等步驟。通過機器學習、神經網路、 生成對抗網路等技術，使得機器能夠根據特定的語境和目標，生成符合 語法和語義的自然語言文本，例如自動寫作、自動翻譯等。自然語言生 成在智慧化客服、自動化寫作、自動化報告生成等領域都有著廣泛應用。

8. 詞向量表示：指將單詞映射為高維度實向量的過程，通過詞向量表示，可以將自然語言中的單詞轉換為計算機可以處理的形式。詞向量表示主 要通過分散式假設來實現，即認為上下文相似的單詞具有相似的詞向量 表示。詞向量表示的主要方法包括基於共現矩陣的方法、基於預訓練的 詞向量方法、基於上下文的方法等。其中，基於預訓練的詞向量方法是 目前最為常用的詞向量表示方法之一，通常是通過大量的無監督訓練來 學習單詞的詞向量表示。詞向量表示在 NLP 中有著廣泛的應用，例如語言模型、文本分類、情感分析等任務。(黃柏元，2022)

二、應用

在應用方面，NLP 也是許多學者在研究過程中所會運用到的一個領域，像是針對年輕運動員的心理訓練，以 NLP 對游泳運動員在焦慮管理和表現方面進行控制，藉此來改善運動員在比賽中的表現（Boughattas, Ben Salha, & Moella, 2022）、在醫療方面，幫助臨床放射科的醫生解釋成像的數據報告、圖像採集、等在放射學方面的當前和潛在應用（Donnelly, Grzeszczuk, & Guimaraes, 2022）。

近年來，自然語言處理在快速發展，研究範圍非常多元化，從簡單的文本分析，到各種的社交媒體分析。許多研究團體使用社交媒體的數據進行信息科學、信息檢索、網路科學、社會科學、社交媒體分析、心理學和與料庫語言學的相關研究，也有很多的企業利用在社交媒體上的留言與評論進行研究分析而因此受益。（洪麒盛，2019）

三、技術

自然語言處理（Natural language process, NLP），是語言資料處理的一個重要分支，也可以稱為自然語言理解，內容包含針對中文字的形音義以及詞彙、文句、文章的輸入、輸出、儲存進行識別、分析、理解、生成等多方面加工處理。

在中文自然語言當中，詞是具有意義的最小語言單位，各式的中文自然語言任務的實現都須建立在有效且正確的斷詞系統上，系統必須先能分辨出文中的詞才能進行分析或是預測，例如情感分析、文意分析，必須正確地完成中文斷詞功能與辨識詞義功能，而當斷詞功能完成後，當輸入的文章過長、包含的字詞量過多的情況下，還需要關鍵詞提取功能來完成擷取文章特徵功能（王君善，2019）。

另外，自然語言的處理技術對於句子結構以及字詞關係的分析有許多的幫助，就語意分析上的深淺，有分為所謂的的深度語意分析（Full Parsing）與淺度語意分析（Shallow Parsing）。深度語意分析意指更詳細的分析出句子中可能的主詞 （Subject）、述詞（Predicate）及受詞（Object）等資訊，而淺度語意分析則僅只有標記字詞詞性為動詞、名詞或是形容詞等，對於句子架構的分析不如深度語意 分析所提供資訊來的詳細，但是在速度上，淺度語意分析則快上許多。因此，處理速度與資訊的詳細程度是在分析過程中所要面臨的一項取捨。(江柏勳，2005)

第三節 LINE 應用

一、LINE 服務與技術

在台灣，無論哪個年齡層，幾乎人人都擁有一個 LINE 帳號，LINE 在台灣擁有高達 90%的驚人滲透率，全台 2300 萬人中，使用 LINE 的活躍用戶數就高達 2100 萬。LINE 的母公司是韓國上市公司，同時也是韓國最大入口網站 NAVER 集團，這些年，LINE 在台灣取得了成功的發展。

LINE 之所以能夠在短短的三年時間內迅速竄紅，除了因為 LINE 提供終生免 費的文字傳訊、語音訊息、語音通話、可以支援多達 100 人的群組聊天室、個人 主頁、動態消息之外，在 2012 年中旬推出的官方帳號服務從一開始的設定讓使用者可以透過 LINE 與自己喜愛的藝人互動，到後來也已經推廣到企業上，從好友之間的溝通到休閒娛樂，服務範圍從私人社群向企業網絡擴散，都使得 LINE 不再單純只是通訊軟體，而是正一步步進化成行動生活平台，深入個人生活與企業脈絡各個面向，形成新一代的文化現象（范揠容，2015）。

二、LINE 官方帳號應用與優勢

除了基本的通訊功能，LINE 也為企業與個人商家提供一個近用性高且個人化的「LINE 官方帳號」，可進行行銷推廣包含經營會員、傳送優惠訊息與客服問答以增加與客戶的互動之功能，並立即獲得回應和解決方案，這種個性化的服務體驗有助於建立良好的客戶關係，提高用戶滿意度和忠誠度。(李佩芸，2023)

LINE 應用程式提供了許多方便的功能，特別是在點餐方面，你可以申請 LINE 官方帳號，並利用這個帳號經營你的商家。根據 2022 年的最新數據，申請 LINE 官方帳號的總數已達到 241 萬。從零售、餐飲、醫療、美容美髮等產業的觀察中可以發現，越來越多的商家開始使用官方帳號來經營他們的顧客群。這些商家利用官方帳號提供各種服務，例如預約、訂位、點餐、詢問服務等等。

LINE 官方帳號的主要功能是，企業可以透過 LINE 官方帳號建立專屬的行銷管道，傳遞消息和發佈動態給每一個加入 LINE 官方帳號 好友的消費者，並且還可以上架貼圖來給消費者免費下載貼圖，但他的缺點是不能與消費者一對一聊天。LINE 創立社群行銷平台後，吸引了很多大型企業與品牌公司的目光，紛紛加入 LINE 企業官方帳號的行列，加入這項服務的主要目的就是為了透過這個管道增加品牌或產品的知曉與利潤，因此社群行銷也是 LINE 所著重的項目。

LINE 官方帳號有別於傳統的行銷廣告（例如 SMS、MMS）使消費者處於被動接收廣告無法由消費者自由決定選擇想接收的資訊，這是一種廣告訊息「推」的策略（Yang, Kim, and Yoo, 2013），這種策略因為沒有徵詢消費者的意願，強制發送廣告，容易造成消費者的困擾，使其成效不明顯，而現在的行動廣告不再像傳統行銷廣告一樣亂槍打鳥的方式發送廣告訊息，而是能更加精準的找到目標客群從「推」的策略逐漸變成「拉」的策略，使消費者透過自己選擇的方式主動參與接收廣告（Tsang, Ho, and Liang, 2004）。LINE-P 是以消費者自行主動搜尋企業官方帳號，並且自願加入其帳號為好友，LINE-P 才主動傳送訊息與消費者進行聯繫；消費者自主性的接受訊息是一種「拉」的策略，因此消費者對於廣告訊息的內容接受度較高（劉揚翊，2018）。

第參章 研究架構與方法

第一節 研究方法

本研究旨在探索如何利用 LINE Messaging API、Azure Bot Service 和 LUIS 應用程序來實現自然語言處理和用戶互動，下面是研究方法的具體步驟：

1. 使用用戶通過 LINE 客戶端向 LINE Messaging API 發送消息。
2. LINE Messaging API 對用戶進行身份驗證和授權，並將消息路由到 Azure Bot Service 應用程序。
3. Azure Bot Service 應用程序使用 LUIS 中間件將消息發送到 LUIS 應用程序進行自然語言處理。
4. LUIS 應用程序使用自然語言處理技術來了解不同用戶意圖和實體，並將結果返回給 Azure Bot Service 應用程序。
5. Azure Bot 服務應用程序根據 LUIS 分析結果執行相關的操作，例如返回相關信息或執行相關任務。
6. Azure App Service 應用程序處理 LUIS 分析結果，並根據結果執行相關的操作。
7. 最後，Azure Bot Service 應用程序使用 LINE Bot respone 將響應消息發送回 LINE Messaging API，並返回給用戶。

這些步驟構成了整個研究方法，通過結合 LINE Messaging API、Azure Bot Service 和 LUIS 應用程序，研究可以實現基於自然語言處理的用戶互動和響應系統。

第二節 研究架構或研究設計架構

本研究根據上節所分析研究之方法，主要重點在於使用者輸入資料以及分析關鍵字做為開發重點，本研究開發使用 LULS 作為後台雲端系統，搭配 LINE 為客戶端接收使用者輸入的資料，再於 LULS 雲端系統輸入關鍵字資料庫，加以分析使用者輸入資料，而以下圖 3-1 為本研究系統架構圖與各元件之說明。

圖 3-1 系統架構圖

有了上圖的系統架構圖之後，我們將整個系統架構流程嵌入 PDCA（Plan-Do-Check-Act）循環以實現持續改進時，可以在整個流程中引入 PDCA 的四個階段。以下是如何將 PDCA 整合到整個系統架構流程中並針對這四步驟逐一說明的示例：

PLAN (計劃):整個系統架構流程的開始，引入 PDCA 的計劃階段，在這個階段，收集有關系統性能、用戶反饋和需求的信息。

DO (執行):在 LINE Messaging API 中執行計劃中的改進措施，包括優化消息處理代碼、提高系統的穩定性等。

在 LUIS 應用程序中，執行計劃中的改進措施，以提高自然語言處理的性能和精確性。

CHECK (檢查): 在改進實施後，監控整個系統性能，收集數據以評估改進效果，還有分析用戶反饋，檢查系統是否符合計劃設定的目標。

ACT (行動): 根據 CHECK 階段的結果，決定是否繼續調整、優化或採取其他措施，如果改進效果不如預期，返回 PLAN 階段，重新計劃和調整。

如此一來，可以將整個系統架構流程嵌入了 PDCA 循環，以確保系統不斷改進並符合用戶需求，可以根據需要定期進行 PDCA 循環，以持續改善系統性能和功能，如下圖 3-2。

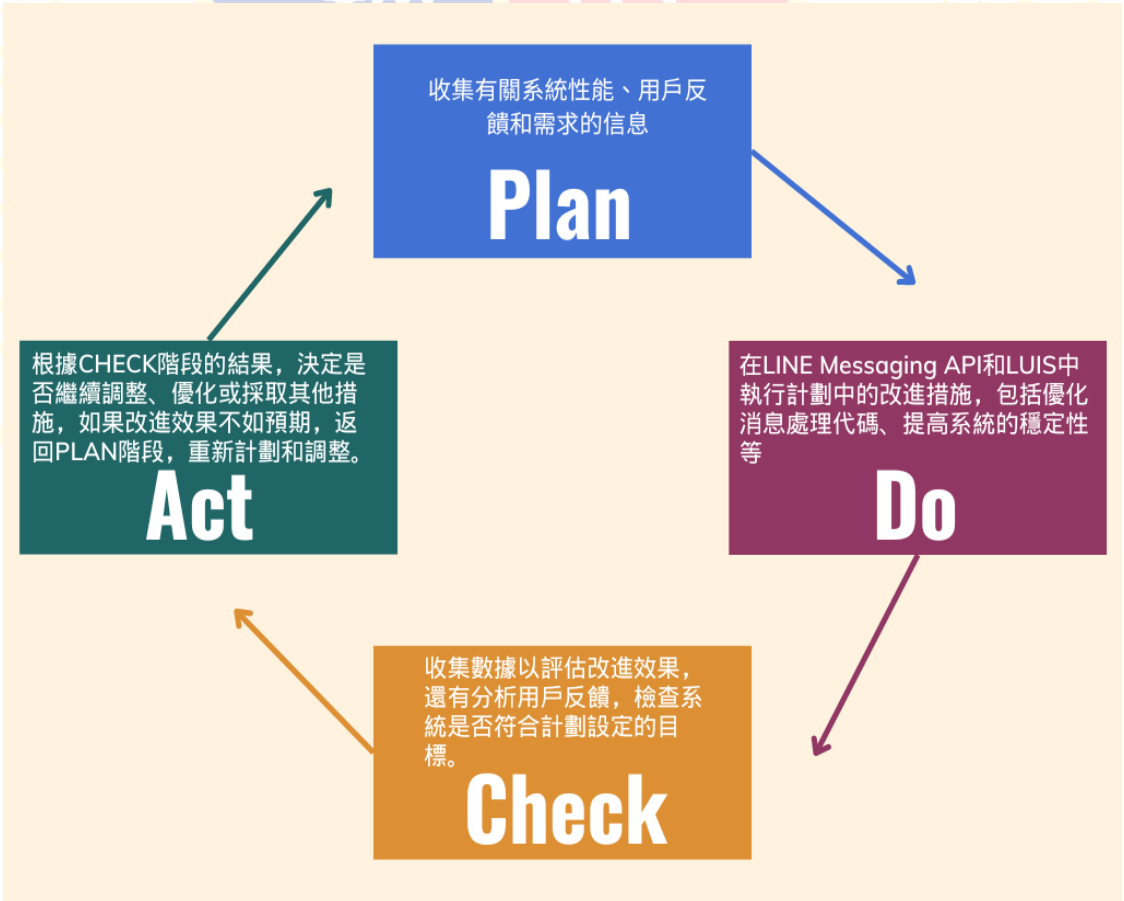


圖 3-2 PDCA 循環圖

第三節 研究工具

一、LINE 客戶端

本研究採用 Line 作為使用者的操作介面，透過使用者的 Line 帳號加入聊天機器人。使用者可以透過輸入機器人的 ID 或是掃描 QR Code 兩種方式加入，成功加入後，使用者可以在 Line 畫面中使用文字輸入的方式與機器人進行點餐，使用者可以直接在 Line 平台上完成點餐，無需額外下載其他應用程式或進行繁瑣的註冊程序。

二、Messaging API

Messaging API 包含兩種 API 形式，Push 和 Reply。

Reply API 是一個供聊天機器人（chat bot）用於回覆使用者訊息的 API，而 Push API 則是允許聊天機器人在任何時間主動將訊息傳送給使用者的 API。Webhook 是提供即時資訊的一種方法，透過 https 安全傳輸協定和 Webhook 伺服器以 JSON 格式進行即時訊息的傳送和接收，無需輪詢方式獲取即時資訊。

我們使用 Messaging API 讓資料在 bot 伺服器和 LINE 平台之間傳遞。使用者向聊天機器人發送訊息後，LINE 平台會將 webhook 事件傳送到 LULS 伺服器的雲端後台。然後，LULS 伺服器會根據 webhook 事件透過 LINE 平台回應使用者。所有這些傳送的請求都是以 JSON 格式透過 HTTPS 進行。

三、Azure Bot Service

Azure Bot Service 是一個功能強大且多元化的整合環境，專為開發和運行對話機器人而設計。作為一個綜合解決方案，它提供了廣泛的功能和服務，涵蓋從建立機器人對話服務到與各種對話頻道連接、測試對話功能，以及佈署和管理等多個方面，無論是在建立機器人對話系統還是在管理它們，Azure Bot Service 都提供了一個集成的平台。

四、LULS 後台系統

LULS（Language Understanding Intelligent Service）是由微軟提供的語言理解服務，旨在協助開發人員創建自然語言對話式應用程式。LUIS 基於自然語言處理（NLP）技術，利用機器學習算法訓練模型，以識別和解釋自然語言中的意圖和實體。我選擇使用 LUIS 進行本次研究的語言處理，主要是因為它與

Microsoft Azure 平台深度整合，該平台提供了多種服務，方便開發者在整合連接和操作時更加便利。

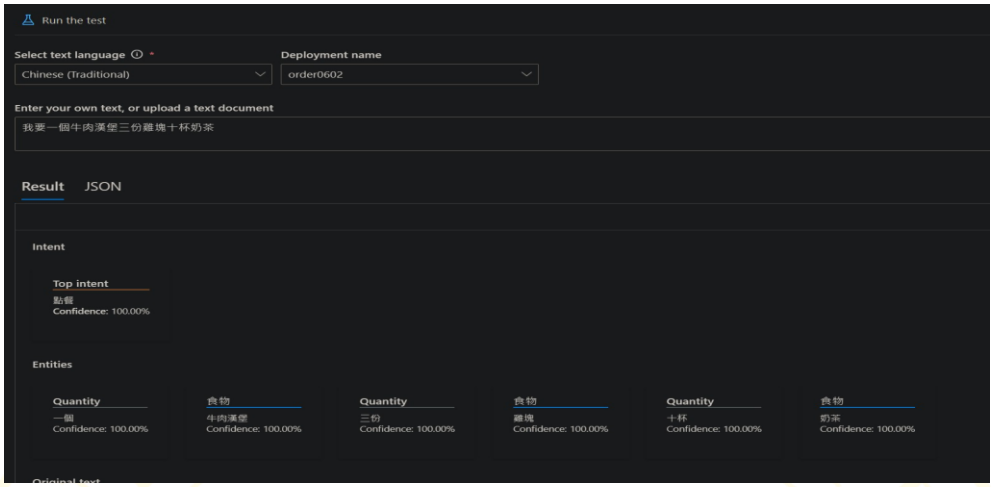


圖 3-3 LULS 後台系統圖

五、Azure Bot Service

Azure App Service 是一種雲託管服務，用於部署和擴展 Web 應用程序、API、移動文檔和函數應用程序。

在上述的系統架構中，Azure App Service 扮演著處理 LUIS 分析結果與執行相關操作的角色，當 LUIS 分析用戶輸入的意圖和實體後，Azure App Service 接收到 LUIS 的結果，從而了解用戶的意圖，Azure App Service 可以根據應用程序的邏輯和需求，執行與用戶理解相匹配的操作。這可以包括返回相關信息、執行特定任務、調用其他服務或處理業務邏輯等。

第肆章 系統實作

本章節為介紹本研究實作中的開發工具、環境、和實作的步驟，同時解釋 Line 聊天機器人系統的建立以及在開發的過程裡，遇到的問題和相關的解決方法， 並詳細介紹整個系統開發的流程，最後將於第五章節呈現本研究的實作成果，以及在第六章節中提出未來希望可以增加的功能。

第一節 系統開發環境

下表 4-1 為本研究在實作過程中所運用到的相關開發工具和開發環境整理，以及開發工具功能的簡單介紹。

表 4-1 開發工具與環境

項目	開發環境與工具	功能介紹
作業系統	Windows 10	作業系統
IDE	Visual Studio 2022	整合開發環境
程式語言	C# .NET6	開發
測試工具	Bot Framework Emulator	本地端測試
Web 服務	Azure App Service	將服務連接至網際網路
開發框架	Bot Framework SDK	開發所需模組
CLU	Language Studio	NLP
Bot	Azure Bot Service	Azure 機器人
Line	line developer	line 方面設定及連接

第二節 系統建置

本研究的主要目標是利用 Line 官方提供的機器人頻道，結合雲端平台 Azure 的強大功能，並且有別於其他系統需要特定連結導向外部網站進行點餐。這款點餐系統的獨特之處在於，它能夠在 LINE 的對話介面中實現快速而便捷的點餐操作。

在著手開發 Line Chat Bot 之前，我們必須進行一系列相關的設定工作，以確保整個項目能夠順利展開。這些設定工作在研究中擔任著關鍵的角色，因為它們確保了聊天機器人的建構和功能開發能夠順利進行，並最終實現我們所設定的目標。

在此過程中，我們將深入研究 Line 官方提供的機器人頻道的功能，並探索 Azure 雲端平台的優勢，以充分發揮其潛力。這將有助於我們建立一個具有高

效、靈活性和便利性的點餐系統，為用戶提供卓越的體驗。

總之，本研究不僅關注於點餐系統的實現，還注重在開發過程中所需的關鍵設定和配置，以確保整個項目能夠成功運行，並滿足用戶的需求。這項研究的成果將有助於擴展機器人應用領域，同時提供了一個具有應用前景的實用範例。

一、初始建立

建立點餐聊天機器人的第一個步驟在 line developer 裡新建頻道，在 Line Developer 平台上，新建一個頻道，然後創建一個 API 通道才能取得相關的權杖和密鑰。接著，設置一個 Webhook，以便從 Line 平台接收資訊，如下圖 4-1 至圖 4-4。

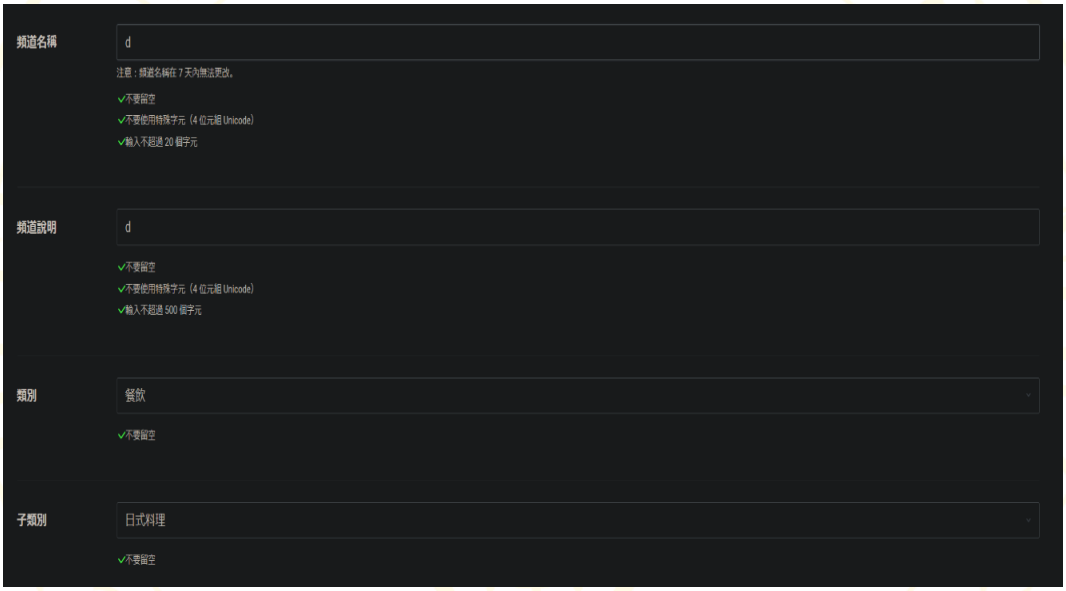


圖 4-1 在 line developer 裡新建頻道

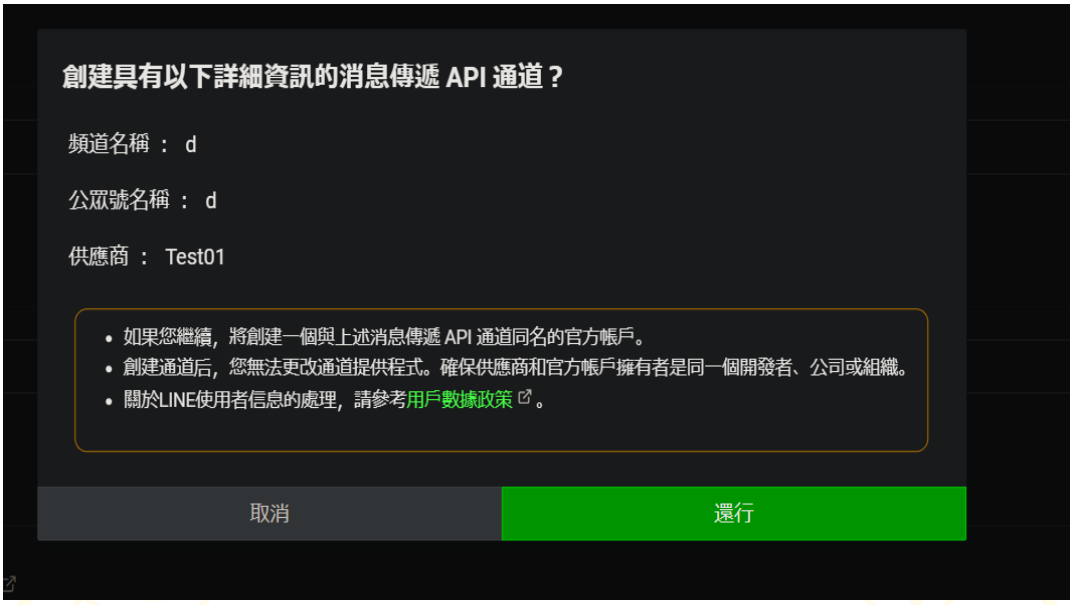


圖 4-2 在 line developer 裡創建 API 通道

圖 4-3 取得權杖及密鑰

圖 4-4 設置 Webhook

完成後，將相關資訊填入 Azure Bot 的設定頁面，進行身分識別的建立。這將使 Azure Bot 能夠與 Line 平台進行連接，如下圖 4-5 至圖 4-6。

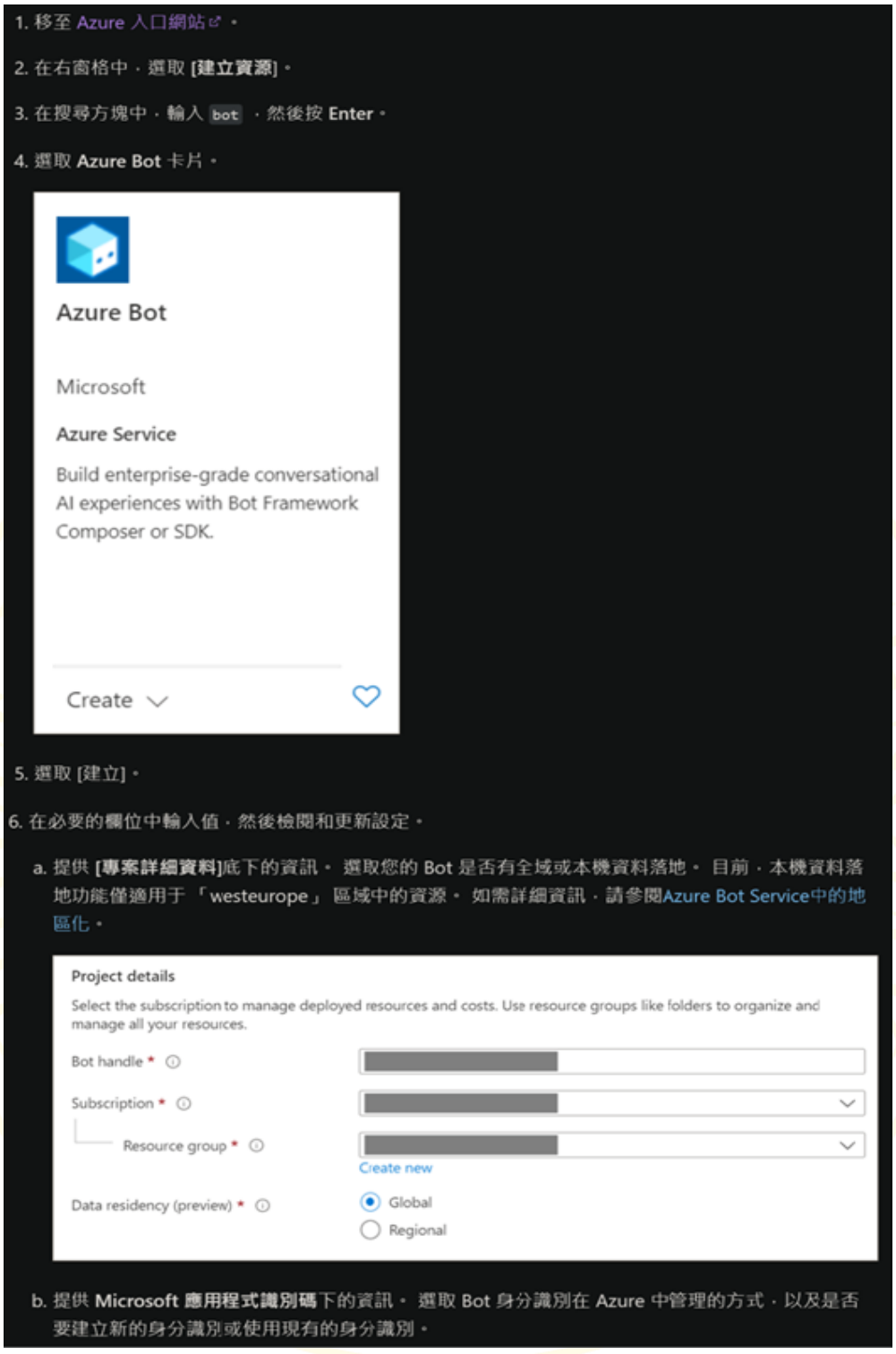


圖 4-5 將資訊填入 Azure Bot 設定頁面

圖 4-6 建立身分識別

接下來，前往 Line 官方的 Github 頁面下載範例程式碼，並根據範例進行必要的修改。這些範例程式碼將幫助理解如何與 Line 平台進行互動，從而更好的開發自己的機器人，如下圖 4-7 至圖 4-13。

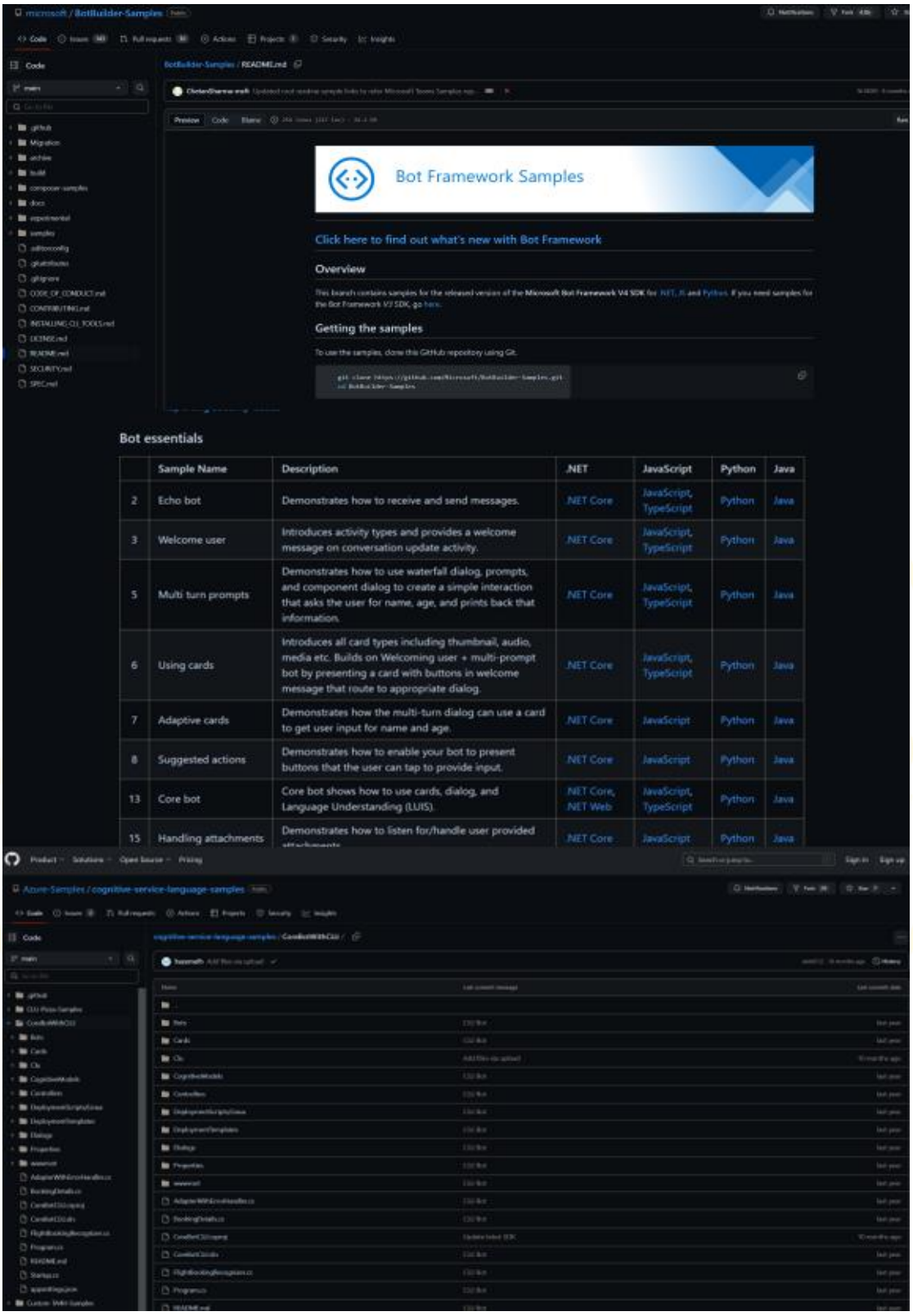


圖 4-7 到官方的 Github 下載範例

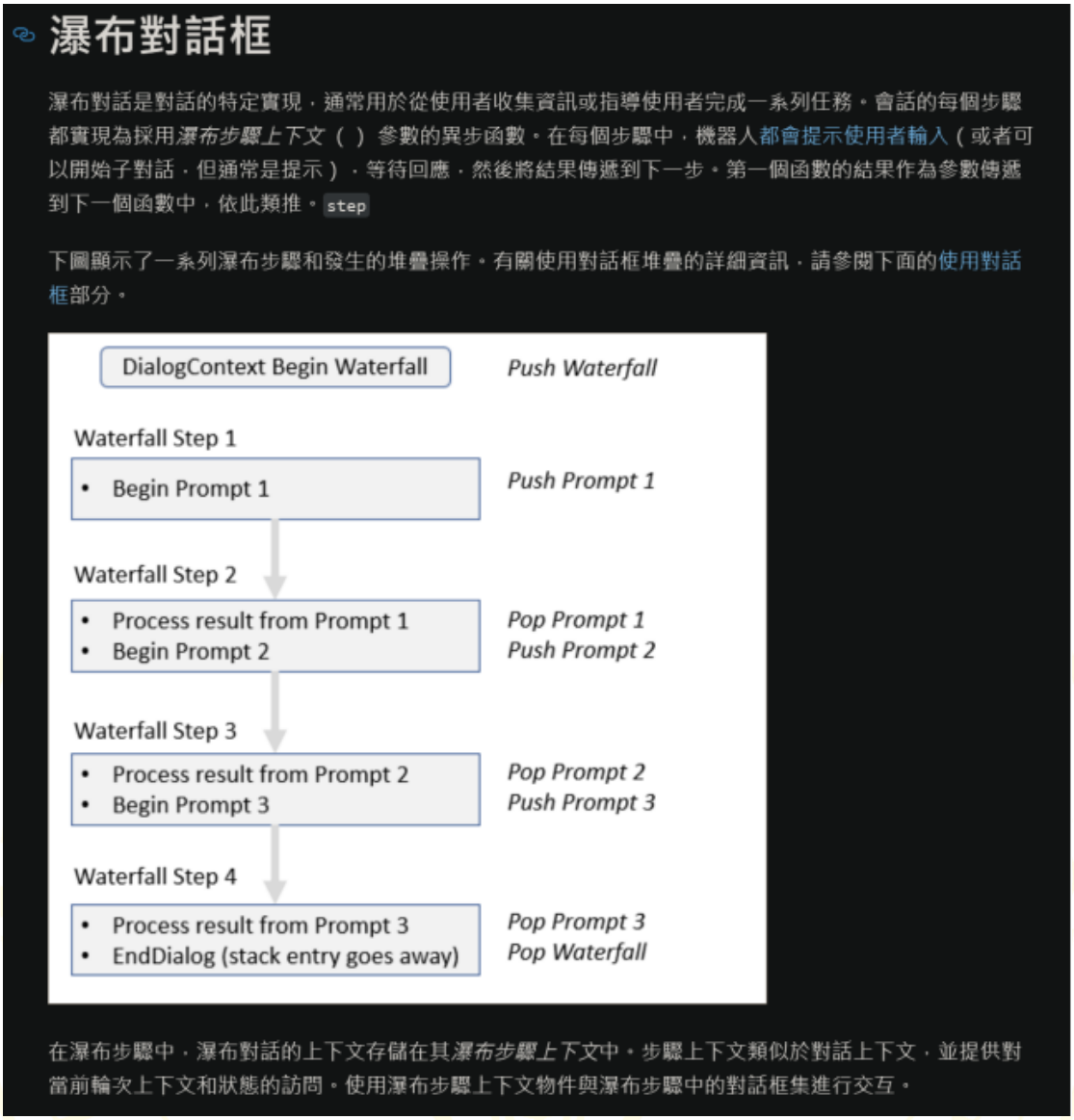
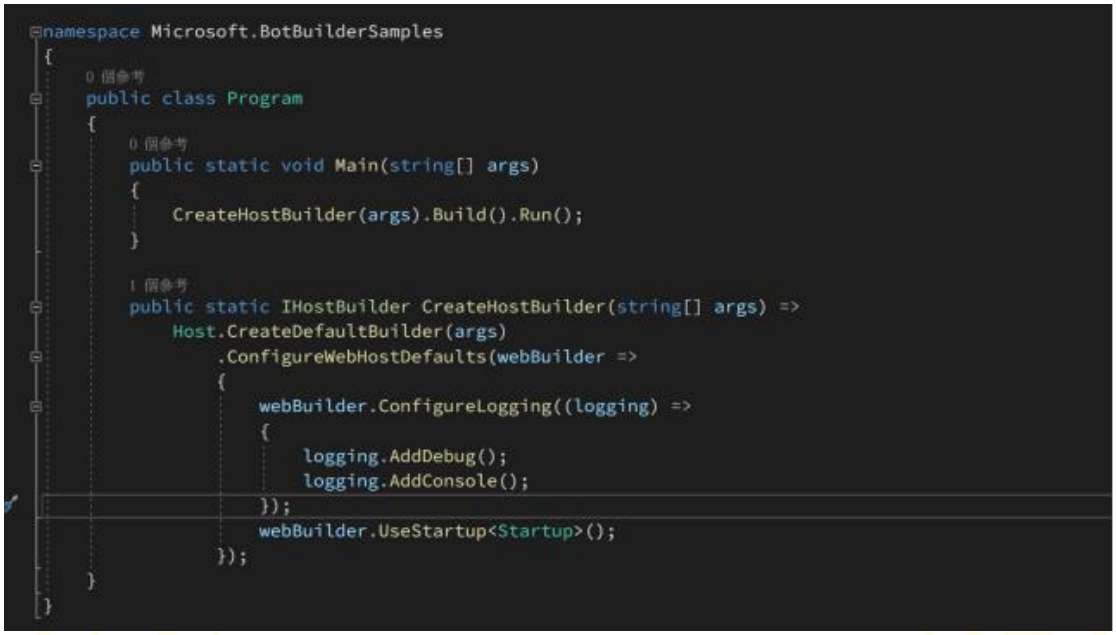


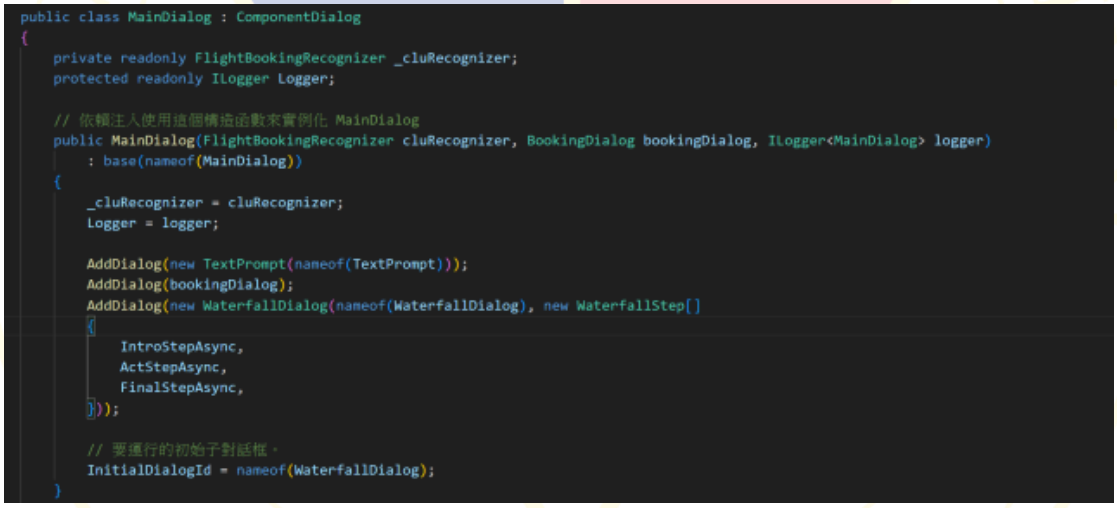
圖 4-8 了解原始設計並根據範例進行修改



```
namespace Microsoft.BotBuilderSamples
{
    0 個參考
    public class Program
    {
        0 個參考
        public static void Main(string[] args)
        {
            CreateHostBuilder(args).Build().Run();
        }

        1 個參考
        public static IHostBuilder CreateHostBuilder(string[] args) =>
            Host.CreateDefaultBuilder(args)
                .ConfigureWebHostDefaults(webBuilder =>
                {
                    webBuilder.ConfigureLogging((logging) =>
                    {
                        logging.AddDebug();
                        logging.AddConsole();
                    });
                    webBuilder.UseStartup<Startup>();
                });
    }
}
```

圖 4-9 確認主要啟動部分確定有無需要修改部分



```
public class MainDialog : ComponentDialog
{
    private readonly FlightBookingRecognizer _cluRecognizer;
    protected readonly ILogger Logger;

    // 依賴注入使用這個構造函數來實例化 MainDialog
    public MainDialog(FlightBookingRecognizer cluRecognizer, BookingDialog bookingDialog, ILogger<MainDialog> logger)
        : base(nameof(MainDialog))
    {
        _cluRecognizer = cluRecognizer;
        Logger = logger;

        AddDialog(new TextPrompt(nameof(TextPrompt)));
        AddDialog(bookingDialog);
        AddDialog(new WaterfallDialog(nameof(WaterfallDialog), new WaterfallStep[]
        {
            IntroStepAsync,
            ActStepAsync,
            FinalStepAsync,
        }));

        // 要運行的初始子對話框。
        InitialDialogId = nameof(WaterfallDialog);
    }
```

圖 4-10 沿用部分範例所有之實例然後加入自己創建之實例

```
    // 如果是第一次使用，請使用 FinalStepAsync 中提供的文本或默認值。
    var messageText = stepContext.Options?.ToString() ?? $"你今天想吃甚麼?\n比方說\"我要一份滷堡\"";
    var promptMessage = MessageFactory.Text(messageText, messageText, InputHints.ExpectingInput);
    return await stepContext.PromptAsync(nameof(TextPrompt), new PromptOptions { Prompt = promptMessage }, cancellationToken);
}
```

圖 肆-11 閱讀並找到最終回答的部分之程式碼並將其文字及變數修改成點餐的回答

```
namespace Microsoft.BotBuilderSamples
{
    /// <summary>
    /// An <see cref="IRecognizerConvert"/> implementation that provides helper methods and properties to interact with
    /// the CLU recognizer results.
    /// </summary>
    ///
    /// <總結>
    /// 一個 <see cref="IRecognizerConvert"/> 實現，提供輔助方法和屬性以與之交互
    /// CLU 識別器結果。
    /// </總結>
    3 個參考
    public class Order : IRecognizerConvert
    {
        4 個參考
        public enum Intent
        {
            Order,
            None
        }
```

圖 4-12 將意圖修改成與 CLU 模型的意圖一致

```
    1 個參考
    public class CluEntities
    {
        public CluEntity[] Entities;

        1 個參考
        public CluEntity[] GetmealcontentList() => Entities.Where(e => e.Category == "Food").ToArray();

        1 個參考
        public CluEntity[] GetQuantityList() => Entities.Where(e => e.Category == "Quantity").ToArray();

        0 個參考
        public string GetMeal() => GetmealcontentList().FirstOrDefault()?.Text;

        0 個參考
        public string GetQuantity() => GetQuantityList().FirstOrDefault()?.Text;

    }
}
```

圖 4-13 上面的參數修改完後連帶修改其他重要變數

```
        // 歡迎 CLU 並收集任何可能的預訂詳細信息。（注意 TurnContext 有對提示的響應。）
        var cluResult = await _cluRecognizer.RecognizeAsync<Order>(stepContext.Context, cancellationToken);
        switch (cluResult.GetTopIntent().intent)
        {
            case Order.Intent.Order:
                // 使用我們可能在響應中找到的任何實體初始化 BookingDetails。
                var bookingDetails = new BookingDetails()
                {
                    Quantity = cluResult.Entities.GetQuantity(),
                    Food = cluResult.Entities.GetMeal()
                };

                // 運行 BookingDialog，向其提供我們從 CLU 調用中獲得的任何詳細信息，它將填寫其餘部分。
                return await stepContext.BeginDialogAsync(nameof(BookingDialog), bookingDetails, cancellationToken);

            default:
                // 捕獲所有未處理的意圖
                var didntUnderstandMessageText = $"抱歉,我不明白你的意思 (intent was {cluResult.GetTopIntent().intent})";
                var didntUnderstandMessage = MessageFactory.Text(didntUnderstandMessageText, didntUnderstandMessageText, I
                await stepContext.Context.SendActivityAsync(didntUnderstandMessage, cancellationToken);
                break;
        }

        return await stepContext.NextAsync(null, cancellationToken);
    }


    private async Task<DialogTurnResult> FinalStepAsync(WaterfallStepContext stepContext, CancellationToken cancellationTo
    {
        // 如果子對話框（"BookingDialog"）被取消，用戶未能確認或意圖不是 BookFlight
        // 這裡的結果將為空。
        if (stepContext.Result is BookingDetails result)
        {
            // 現在我們有所有的預訂細節調用預訂服務。

            // 如果調用預訂服務成功，請告訴用戶。

            var timeProperty = new TimexProperty(result.Date);
            var travelDateMsg = timeProperty.ToNaturalLanguage(DateTime.Now);
            var messageText = $"您的餐點是{result.Food}{result.Quantity}";
            var message = MessageFactory.Text(messageText, messageText, InputHints.IgnoringInput);
            await stepContext.Context.SendActivityAsync(message, cancellationToken);
        }

        // 第二次使用不同的消息重新啟動主對話框
        var promptMessage = "還有甚麼我可以幫你的嗎?";
        return await stepContext.ReplaceDialogAsync(InitialDialogId, promptMessage, cancellationToken);
    }
}
}
```

圖 4-14 保留整體結構並且只更改必要部分

修改好程式碼後，到 CLU 部分來訓練以及布置模型，測試模型並記下連接需要的密鑰後，依照要求填寫設定資訊來連接各項必要服務，如下圖 4-15 至圖 4-19。

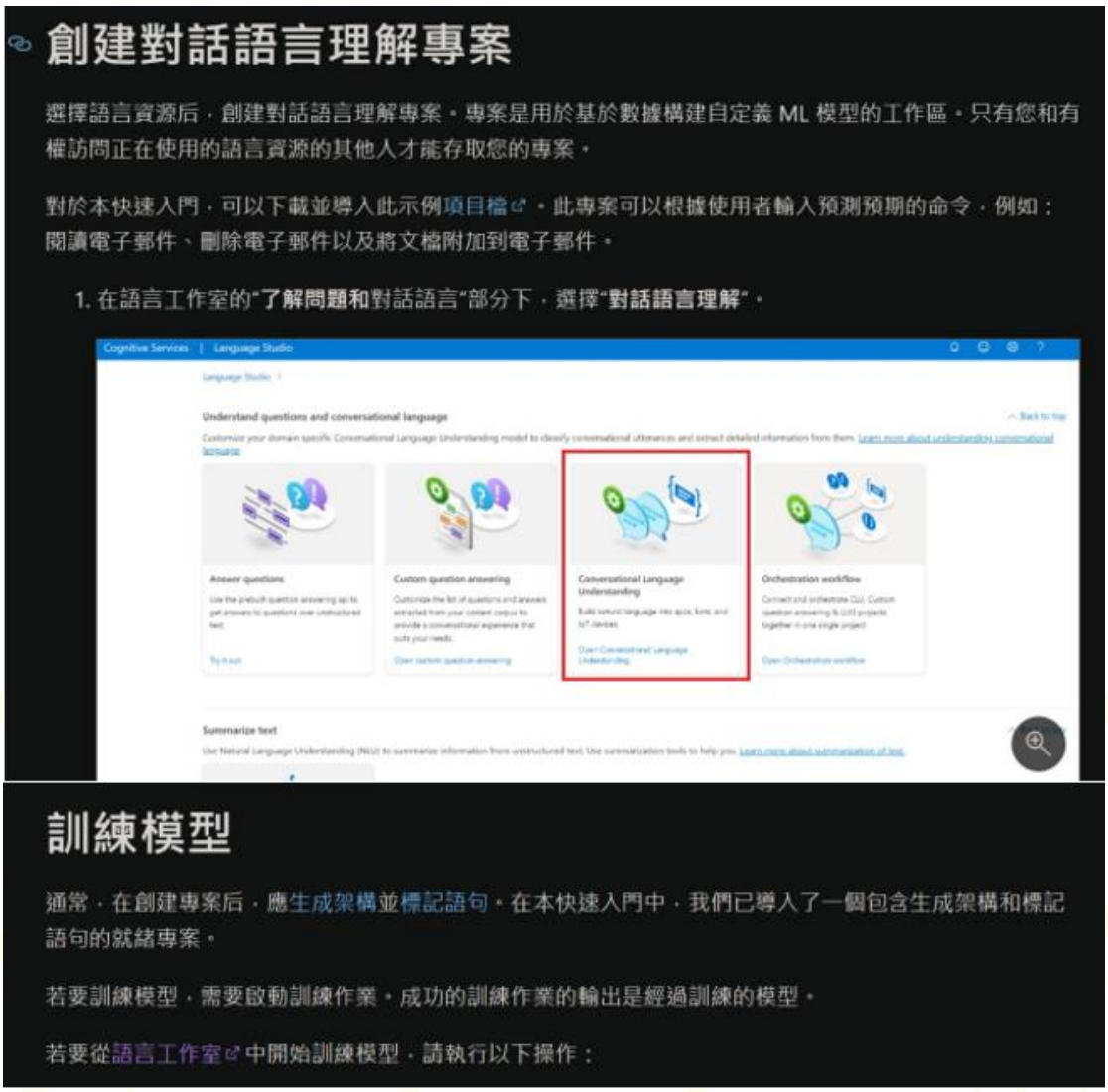


圖 4-15 到 CLU 部分來訓練及布置模型

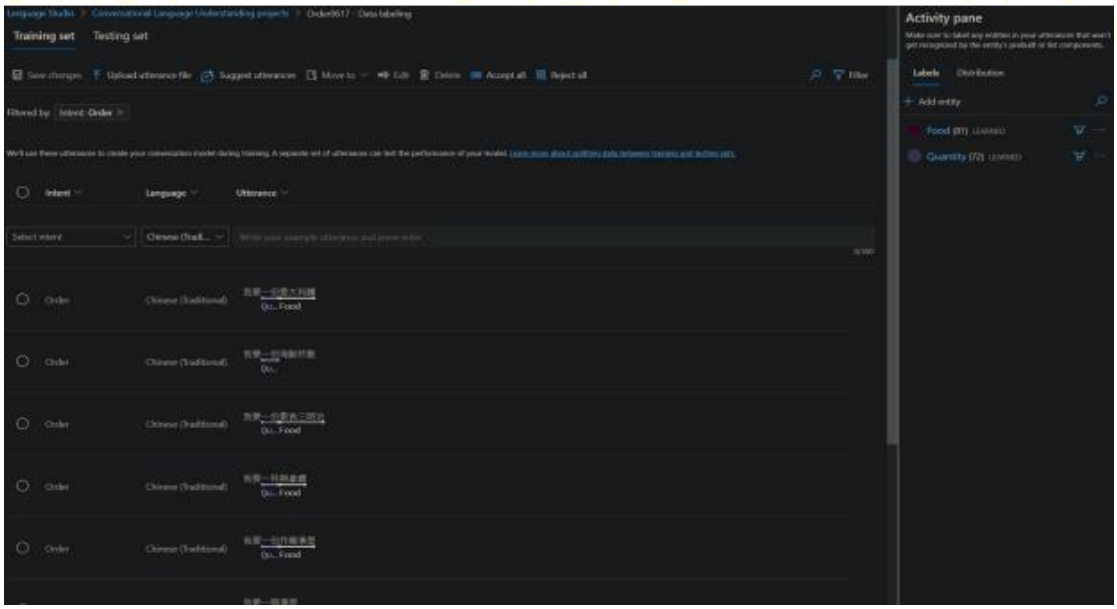


圖 4-16 放入訓練資料

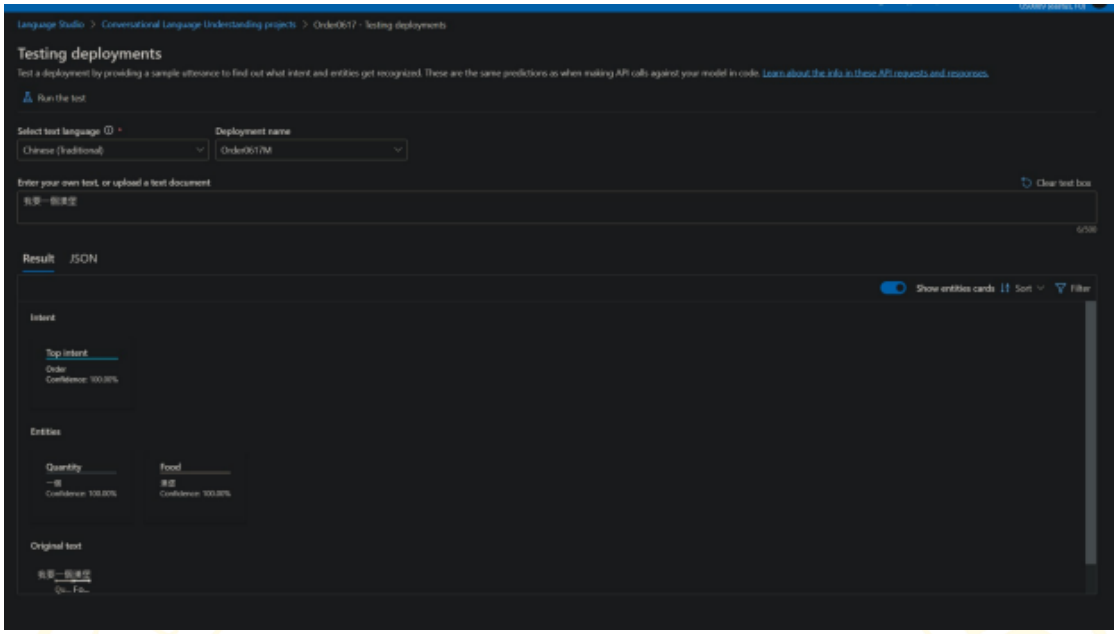


圖 4-17 測試模型

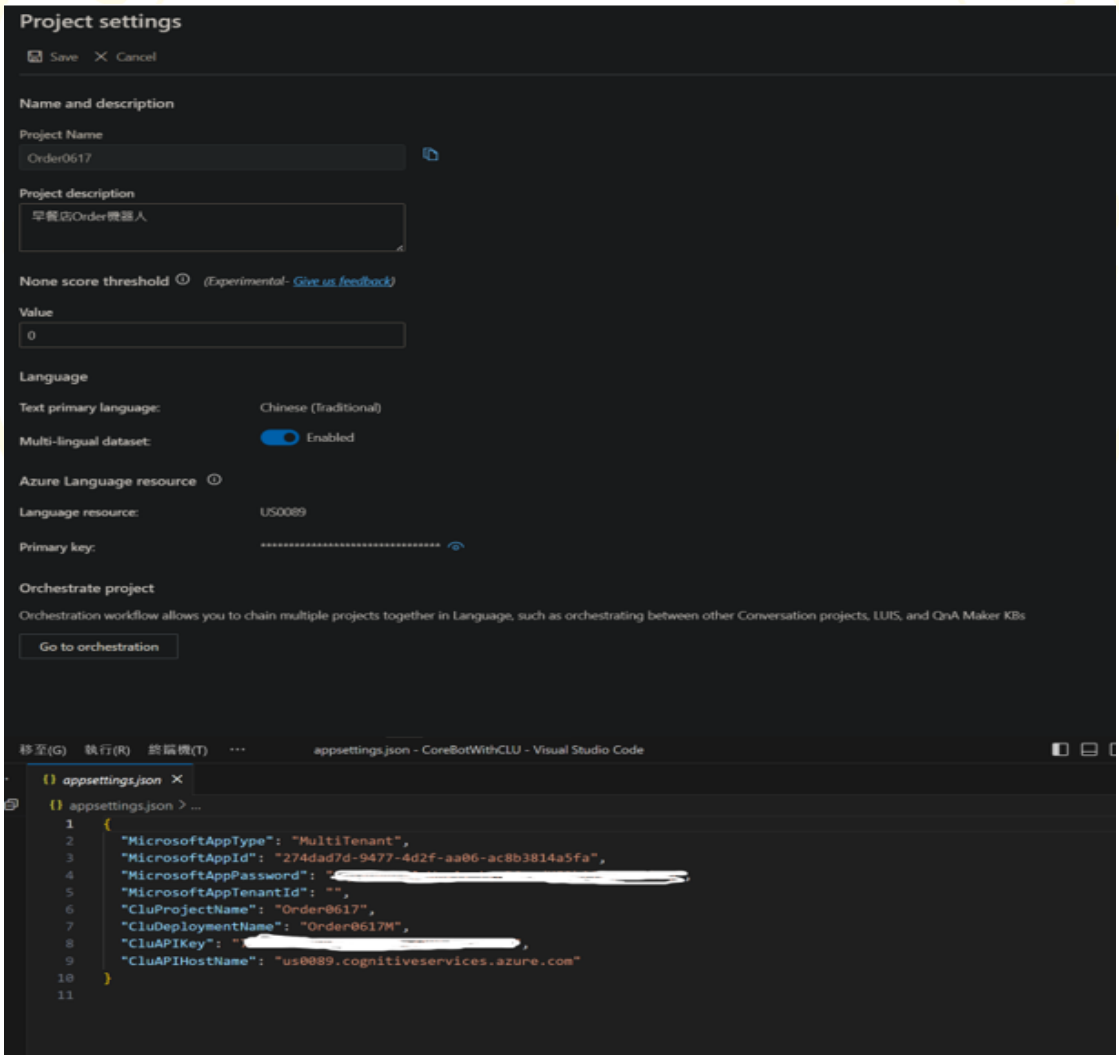


圖 4-18 連接需要的密鑰

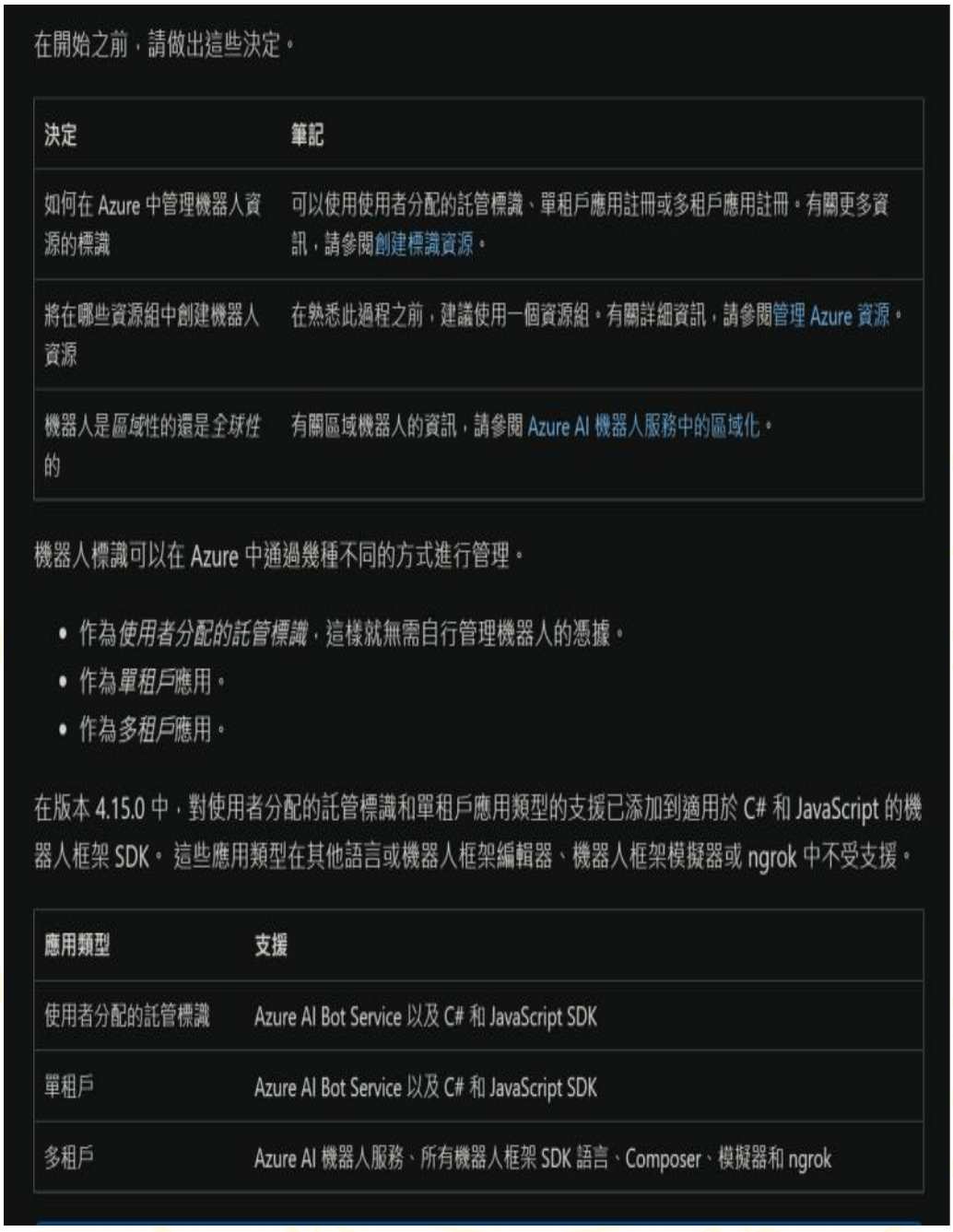


圖 4-19 依照要求填寫設定資訊來連接各項必要服務

連接各項必要服務之後將 bot 服務部屬到 app service 如圖 4-20 至圖 4-22。圖 4 -22 則為 line 端測試成功

圖 4-20 按照官方操作將 bot 服務部屬到 app service

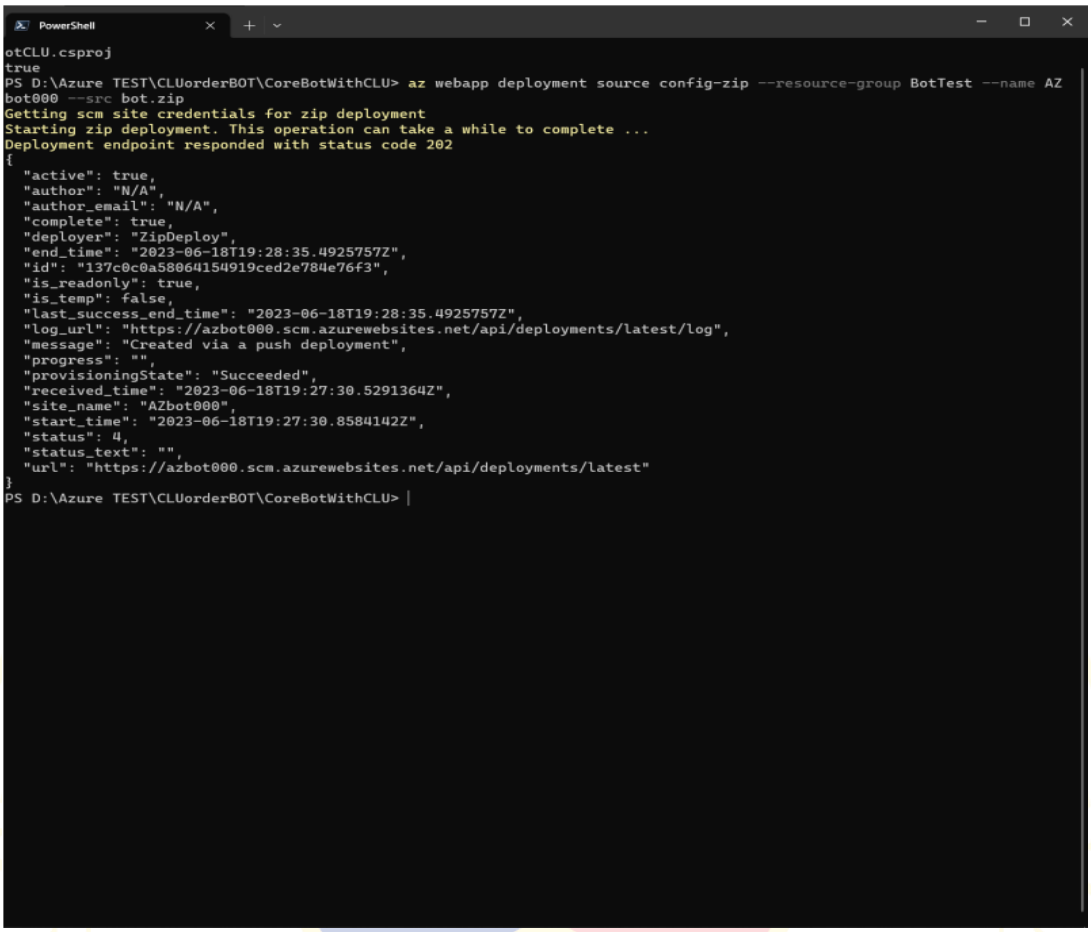


圖 4-21 部屬成功

圖 4-22 line 端測試成功

第伍章　結論與建議

第一節系統開發成果

基於測試結果，雖然系統的辨識率和功能在某些方面相對受限。儘管在複雜任務方面的表現似乎不夠穩定，但稍微制式化的輸入還是能夠完成線上點餐的任務。在某些特殊情況下，比如添加語氣詞、語助詞或介詞時，系統可能會出現錯誤的辨識結果。這可能與系統對於句子結構和語法的理解程度有關。在這方面，系統似乎需要更深入的語法和上下文理解，以避免對句子的詞彙使用產生混淆。然而，這樣的結果也並非出乎意料。自然語言處理技術的複雜性和多樣性意味著在某些情況下，系統可能會遇到困難。而系統的辨識率可能受限於其訓練數據的範圍和品質。在處理不同的語境時，系統可能需要更多的多樣性訓練數據，以提高對於語言的理解和處理能力。

總體而言，基於測試的結果，我們可以看到系統在某些方面的局限性。然而，這也是一個持續改進和優化的過程。通過更多的數據訓練和技術優化，系統的辨識率和功能可能會得到提升。同時，也提示用戶在使用時採取清晰的表達方式，以幫助系統更好地理解和回應。

最終，這樣的測試和反饋對於系統的改進至關重要，有助於使其能夠更好地滿足用戶的需求，提供更準確和穩定的服務。

第二節使用者問卷調查

本研究結果透過網路形式發放問卷邀請到 111 位使用者共同參與，其中有 71 位女性，40 位男性參加。年齡方面，在 20 歲以下的使用者有 38 位，20~40 的使用者為 41 人，40~60 的使用者為 30 人，而 60 歲以上的使用者有 2 人。下表 2 為本研究之問卷題目設計，以及圖 5-1 至圖 5-8 為個別題目結果分析圖。

表 5-1 問卷調查

題目編號	題目
一	平時是否有常使用網路點餐的習慣?
二	顧客先用網路訂餐，你認為是否能為店員節省時間?
三	這個 LINE 點餐機器人能確實幫助到我?
四	本機器人使用上簡單易懂且容易操作嗎?

五	對於目前本機器人自動回覆計數餐點功能是滿意的嗎?

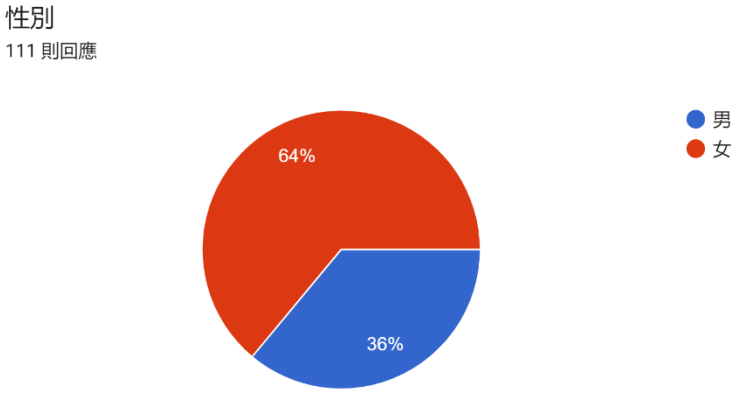


圖 5-1 參與問卷性別比例圖

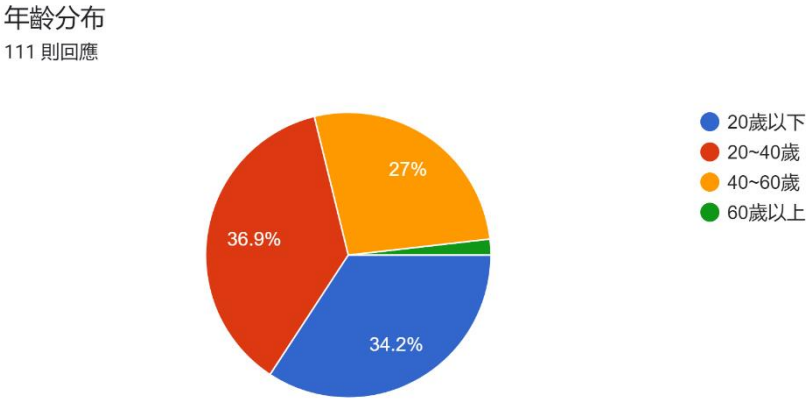


圖 5-2 參與問卷年齡分布比例圖

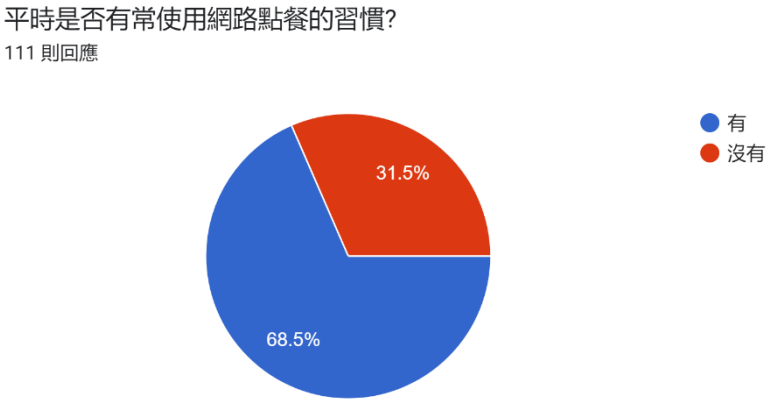


圖 5-3 平時是否有常使用網路點餐的習慣?

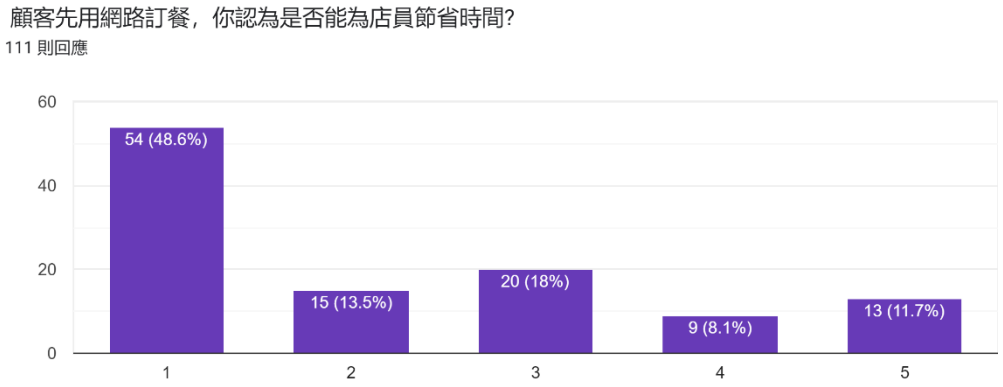


圖 5-4 顧客先用網路訂餐，你認為是否能為店員節省時間?

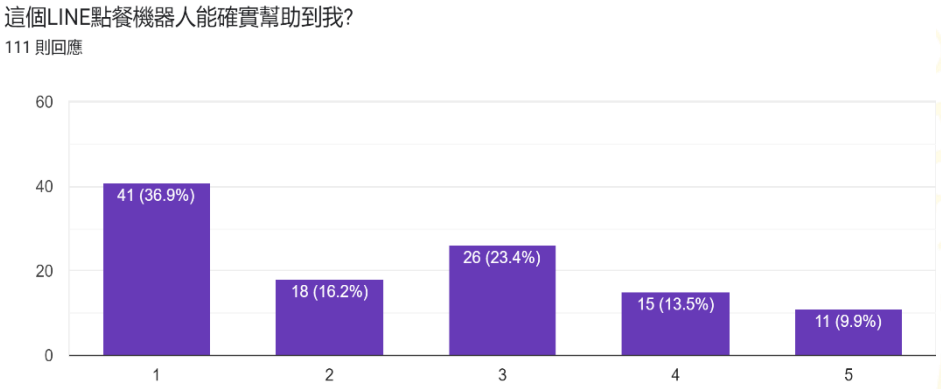


圖 5-5 這個 LINE 點餐機器人能確實幫助到我?

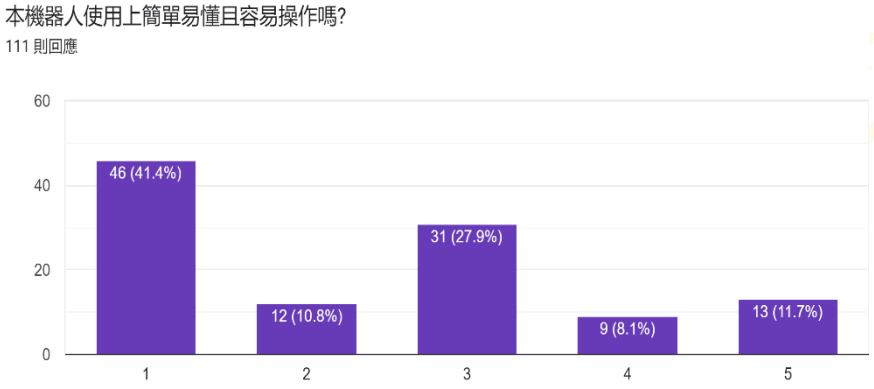


圖 5-6 本機器人使用上簡單易懂且容易操作嗎?

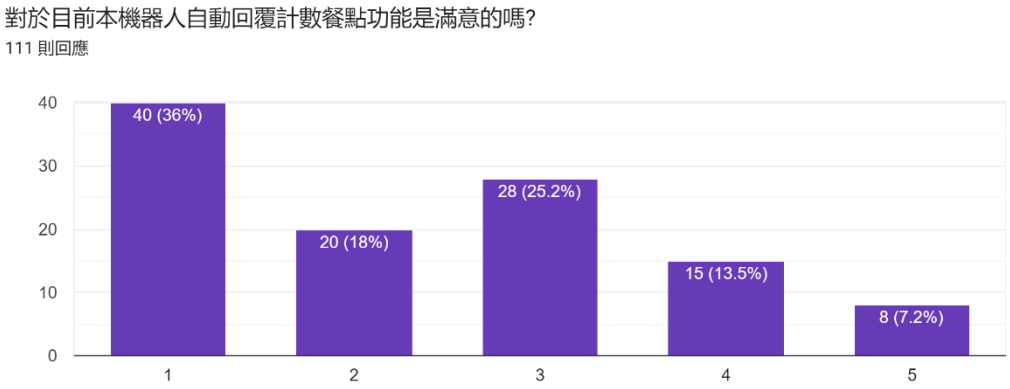


圖 5-7 對於目前本機器人自動回覆計數餐點功能是滿意的嗎?

第三節 系統未來發展

(一)使用更先進之模型:在專題實做部分大致完成後 微軟與 OPENAI 有深度的合作其中的 GPT-3 或 GPT-4 模型已經開放在 AZUR 平台試用 若是以後能將其整合進目前的系統勢必能在各方面都有很大的提升

圖 5-8 可選之更進階模型

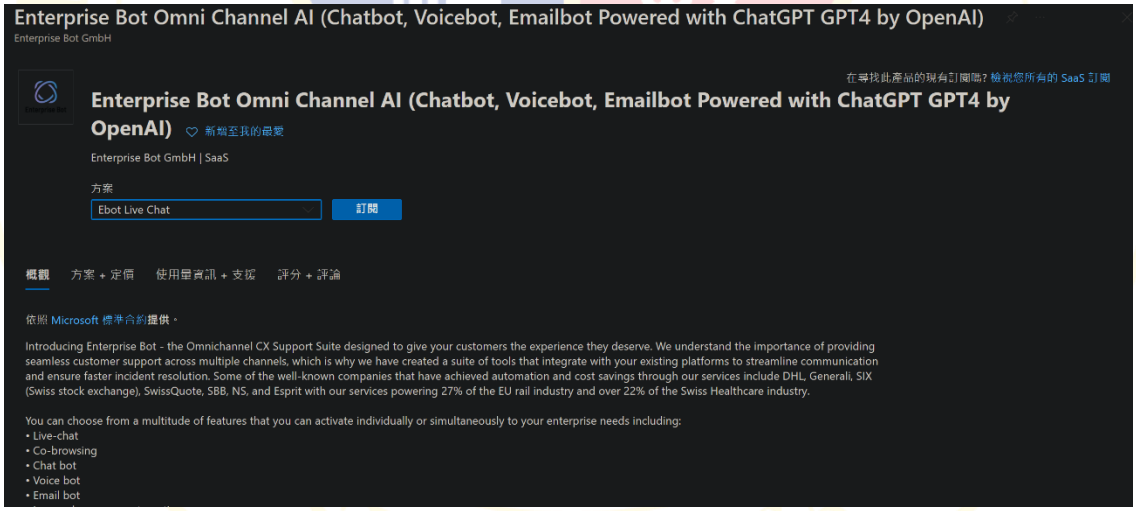


圖 5-9 可選之更進階模型

基於前述結果，將 ChatGPT 整合到系統中後，我們可以預見一系列潛在的未來發展可能性：

1. 提高語境理解： ChatGPT 的強大自然語言處理能力將有助於提高系統對用戶的語境理解。它可以更準確地識別用戶的意圖，並回應更複雜的查詢和對話，從而增強系統的功能性。

2. 改進對話流暢度： ChatGPT 可以改進對話的流暢度，使其更自然且具有連貫性。這將提升用戶體驗，使對話更具人性化。

3. 更高的語法和上下文理解：ChatGPT 對語法和上下文的理解能力較強，因此系統可能更容易處理複雜的句子結構和語法。這將有助於克服之前提到的語氣詞、介詞等可能導致的辨識錯誤。

4. 擴展功能範圍： ChatGPT 的整合可以使系統處理更多種類型的任務，包括回答問題、提供建議、執行任務等。這樣，系統可以應對更廣泛的用戶需求。

總之，整合 ChatGPT 後，系統的未來發展將朝著提高功能性、改進對話品質、增強用戶體驗以及個性化服務方向發展。這將使系統更適應多樣的用戶需求，提供更智能、準確和便捷的處理方式。然而，也需要不斷的優化和反饋，以確保系統持續進步並滿足用戶期望。

參考資料

英文文獻

Brandtzaeg, P. B., & Følstad, A. （2017）. Why people use chatbots. In International conference on internet science （pp. 377-392）.

Crutzen, R., Peters, G. J. Y., Portugal, S. D., Fisser, E. M., & Grolleman, J. J. （2011）. An artificially intelligent chat agent that answers adolescents' questions related to sex,drugs, and alcohol: an exploratory study. Journal of Adolescent Health, 48（5）, 514-519

K. Mishev, A. Gjorgjevikj, I. Vodenska, L. T. Chitkushev, D. Trajanov, （2020）"Evaluation of Sentiment Analysis in Finance: From Lexicons to Transformers," IEEE Access, vol. 8, pp. 131662-131682.

Yang, B., Kim, Y., & Yoo, C. （2013）. The integrated mobile advertising model: The effects of technology and emotion based evaluations. Journal of Business Research,66（9）,1345-1352.

Tsang M. M., Ho, S. C., & Liang, T. P. （2004）. Consumer attitudes toward mobile advertising: An empirical study. International Journal of Electronic Commerce, 8（3）, 65-78.

Boughattas, W., Ben Salha, M., & Moella, N. （2022）. Mental training for young athlete: A case of study of NLP practice. SSM - Mental Health, 2, 100076.

Donnelly, L. F., Grzeszczuk, R., & Guimaraes, C. V. （2022）. Use of Natural Language Processing （NLP） in Evaluation of Radiology Reports: An Update on Applications and Technology Advances. Seminars in Ultrasound, CT and MRI, 41 43（2）, 176-181.

中文文獻

洪麒盛（2019）。淡江大學資訊管理學系碩士論文。人工智慧情感對話機器人。

范揠容（2015）。世新大學資訊傳播學系碩士論文。智慧型手機通訊軟體的使用

者經驗之研究—以 LINE 為例。

李佩芸（2023）。國立臺北教育大學理學院數位科技設計學系玩具與遊戲設計碩士班 碩士論文。合作學習導入 LINE 聊天機器人 輔助自我調整學習－以網頁設計課程為例。

黃柏元（2019）。國立臺北教育大學理學院資訊科學系碩士論文。特定新聞對股價之研究。

王君善（2019）。國立中央大學網路學習科技研究所碩士論文。應用自然語言處理技術開發基於知識翻新理論之線上非同步合作論證平台與平台初步評估。

江柏勳（2005）。國立成功大學資訊工程學系碩士論文。基於自然語言處理技術之網路文件問答系統。

劉揚翊（2018）。東海大學企業管理學系碩士論文。探討資訊品質對 LINE 企業官方帳號的影響: 兼論官方帳號加入動機之調節效。