中文訊息傳遞服務對話系統之建構

葉丞鴻 **Cheng-Hung Yeh**
中央大學資訊工程學系
yeh110522095@g.ncu.edu.tw

張嘉惠 **Chia-Hui Chang**
中央大學資訊工程學系
chia@csie.ncu.edu.tw

摘要

任務型導向對話 (TOD) 系統面臨著語料收集、標記和模型架構及訓練等挑戰。過去使用 Wizard-of-Oz (WOZ) 方法進行語料蒐集，透過人與人互動標記以訊息傳遞爲主的對話語料。然而，使用 WOZ 方法時需要同時產生自然語言對話及標記對話中提及的槽值，這會影響整體資料集品質且難以迅速建立對話語料。本研究提出專注於訊息傳遞的 messageSGD 語料集，利用 Schema-Guided Dialogue (SGD) 產出對話的框架，再由標記人員進行改寫，加速語料庫的生成。另外我們也使用 T5 模型和 Instruction Prompt 建置 NLU、DST、DPL、NLG 四個任務模型，分別達到 91.36、80.08、70.54 及 78.18 的 F1-Score。透過本研究，我們能夠以較少資源快速建立對話系統，並期望提供額外的對話系統建置方法。

關鍵字：任務導向對話系統、語料建構
Keywords: Task-orient dialogue, Corpus construction

1 Introduction

在過去，業界專注於建構任務型導向對話系統（task-oriented dialogue systems），以幫助完成特定任務，例如飛機航班預訂 (Seneff and Polifroni, 2000) 或公車訊息 (Raux et al., 2005)。而隨著智慧型系統及虛擬助理的普及，建構可跨不同應用領域處理任務的對話系統變得越來越重要。

任務導向對話系統，也稱爲目標導向對話系統（goal-oriented dialogue system），主要通過與使用者之間的自然語言交互來執行特定任務。依據 (Chen et al., 2017) 的研究分類，對話系統本質必需理解人類語言時的歧義；整合第三方服務和對話環境；最後，產生自然和引人入勝的回覆。現有任務導向對話系統將以上問題分爲四個子任務來解決，如圖1所示。自然語言理解（Natural Language Understanding, NLU）解析使用者的話語（utterance），了解這句話的需求及意圖（intention）。對話狀態追蹤（Dialogue State Tracking, DST）則記錄 NLU 模組所分析的對話意圖和對話中的實體（entity）與槽值（slot value），以利 TOD 系統將使用者所提到的資訊輸入資料庫進行查詢。最後再由對話策略學習（Dialogue Policy Learning, DPL）及自然語言生成（Natural Language Generation, NLG）將資料庫回傳的查詢結果轉換爲系統的回覆策略（policy）並以自然語言回饋給使用者。

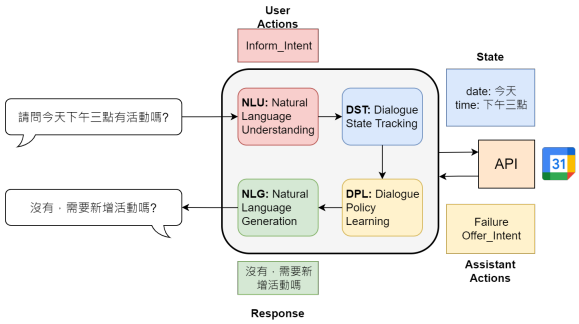

Figure 1: 任務導向系統架構圖

而隨著 ChatGPT 及 InstructGPT(Ouyang et al., 2022) 的發佈，大幅降低了建構對話機器人的難度，也拉近了整合任務導向對話與開域對話（Open-domain dialogue）的距離。透過大量的文本資料和 RLHF（Reinforcement Learning from Human Feedback）進行訓練，ChatGPT 能更加的了解如何根據使用者的輸入輸出更恰當的回覆。爲了解決任務導向對話需要串接外部 API 的問題，ChatGPT 也提供了上百種的外掛程式（Plugin），用來與市面上的各種應用進行互動。

本論文以綱要引導（Schema-Guided）的方式進行機器對機器（machine-to-machine）的對話蒐集，通過定義清楚的對話綱要和對話模擬器（Dialogue Simulator）來快速的產生對話。而標記人員只需要將模擬器產生出的對話做對話改寫即可，如此便可大幅降低語料標記上所需要的時間及成本。

在實驗部份，我們以 T5（Text-To-Text Transfer Transformer）(Raffel et al., 2020) 爲基底架設如圖1的四個模組，並整合爲一完整的對話系統。我們爲了使各個模組在對話的解析及回覆的決策能有更進一步的提升，我們還在各模型的訓練階段加入了提示學習（Prompt Learning），使模型能依據各個任務的定義來更了解該如何對資料進行解析並輸出。準確度在對話理解的任務上，相較於無提示的 76.03，以提示學習進行微調可提升到 83.36。對話生成中的 BLEU-Score 也從 25.89 提升至 32.43。爲了能讓對話狀態追蹤及對話決策能對對話做更精確的解析，我們透過增加 in-context 來讓 DST 的準確度從 41.65 提升至 51.69。

雖然現在能透過 ChatGPT 快速開發各個智慧助理，但由於使用該服務需要依照生成的字詞數量（tokens）來計費，故長期下來也會形成不少的營運及維護成本。因此透過自動標記方法來快速蒐集任務對話語料，訓練任務導向對話系統仍是減少建構對話系統的時間及成本的重要方式。本研究以電子郵件和通訊軟體等訊息領域爲範例，希望建立一自動標記方法來快速蒐集任務對話語料，減少建構對話系統的時間及成本。

2 Related Work

當前語料蒐集可以分爲 Machine-to-Machine 及 Human-to-Human 二種蒐集方式。這二種方式都需要模擬眞實的人機互動對話情境。在 Human-to-Human 的語料蒐集方式中，Wizard-of-Oz 是目前最爲流行的方法。而在 Machine-to-Machine 語料蒐集方式中，Schema-Guided Dialogue 則被廣泛使用來模擬對話和進行資料蒐集。本章將對 Wizard-of-Oz 和 Schema-Guided Dialogue 兩種語料蒐集方法的演進進行回顧和探討。

2.1 Wizard-of-Oz

爲了建立任務導向對話的溝通語料庫，過去常使用 Wizard-of-Oz(WOZ) 方法 (Kelley, 1984)。這種方法需要一人扮演機器角色，另一人扮演人類角色，進行特定情境和任務的對話，以收集人機對話的語料。最早的語料庫是 ATIS (Hemphill et al., 1990)，用於航班口語理解任務。後來改進的 WOZ2.0 (Wen et al., 2017) 建立了餐廳訂位的任務語料。在標記方面，系統需記錄使用者對話狀態和意圖，並標記自身的對話。儘管這些語料奠定了任務導向對話研究的基礎，但仍有多領域和跨領域對話的限制。

爲增加對話的複雜性和多樣性，MultiWOZ(Budzianowski et al., 2018) 採用了類似的方法來擴充語料。MultiWOZ 使用基於模板的方式，結合資料庫綱要中的槽生成任務敘述，以幫助對話標記人員更好地理解對話主題和任務目標。使用者角色的人員根據生成的任務進行對話，而系統角色的操作人員則對使用者的要求進行資料庫查詢並回報結果。

MultiWOZ 提供了詳細的任務描述，使對話語料更具體，協助研究人員進行更有效的對話研究。然而，由於 MultiWOZ 使用 Amazon Mechanical Turk 進行人工標記，標記一致性仍然存在問題。因此，CrossWOZ(Zhu et al., 2020) 提供了中文對話語料，使用自動標記來標記對話意圖和狀態，並強調使用者每輪對話中選擇的領域是相互依賴的，以增強模型對上下文的理解，從而減少對話標記的不一致性。

(葉丞鴻 et al., 2022) 使用 Wizard-of-Oz 方式，仿照 CrossWOZ 的方法，建立了 messageWOZ 語料集，如圖2所示，以深入了解中文訊息服務的對話語料收集方式。該語料集包含了涉及電子郵件、行事曆和通訊軟體三個服務的使用者互動對話。我們聘請兩位標記人員，一人擔任使用者提出需求，另一人擔任助理解決需求。使用者根據目標提出需求，助理則查詢資料庫並以自然語言回饋結果，對話持續進行直到目標完成。我們使用 INFORM、REQUEST、SOM 和 SELECT 等規則進行對話行爲標記，引入跨領域對話，使模型理解不同但相關的槽值。

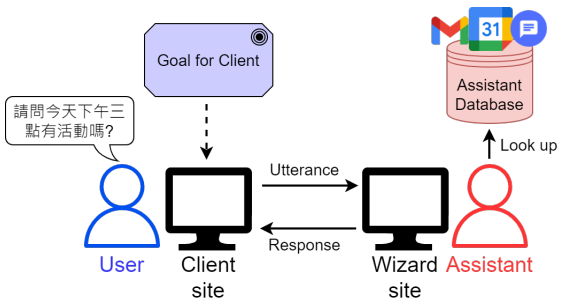

Figure 2: messageWOZ 資料蒐集方法

2.2 Schema-Guided Dialogue

爲了降低對話語料的標記成本，研究人員嘗試了機器對機器的方法。這些方法使用對話代理取代人類來擔任使用者端和助理端的角色，完成特定任務對話。對話代理可以使用傳統的機器學習架構或深度學習模型。例如，M2M(Shah et al., 2018) 利用自動化框架和自我對話 (self-play) 機制建立對話代理，在餐廳和電影院情境中模擬客户與服務人員的對

話。該方法從資料庫綱要 (schema) 中提取槽位 (slot) 和槽值 (slot value)，並將槽值隨機抽樣插入預設的對話模板，生成初步的任務型對話。最後，通過眾包 (Crowdsourcing) 的方式對對話進行人工改寫，以提高其眞實度。

另一種方法是使用 SGD 語料 (Rastogi et al., 2020)，這是一種符合語音助理需求的 Schema-Guided Dialogue 的大量對話語料。SGD 是目前世界上最大的任務對話語料，包含各種領域和大量的對話數據。SGD 使用資料庫 API(Application Programming Interface) 獲取綱要，每個綱要都包含服務、意圖和槽值。SGD 使用兩個機率自動機 (Probabilistic Automaton) 作爲系統和使用者，構建對話模擬器 (Dialogue Simulator)，並使用對話大綱和對話模板生成對話。由於大綱已包含每輪對話的意圖和槽值，使用 SGD 可以減少對話狀態和意圖的標記成本。

我們發現 Wizard-of-Oz 人對人對話蒐集方法很少提供資料庫或 API 查詢結果。同時，在定義對話行爲時，我們需要使用自動標記規則，因此無法涵蓋過於複雜的行爲，也導致對話行爲標記的數量不足。鑒於上述困境，我們的研究採用綱要引導的方式收集與訊息相關的服務，建立了 messageSGD 的任務導向對話語料。我們希望透過機器對機器的自動標記方法，能夠快速建立語料集並搭建完整的對話系統。

3 messageSGD

我們參考了 (Rastogi et al., 2020) 的方法，以綱要引導對話 (Schema-Guided Dialogue) 自動蒐集語料。但由於 SGD 資料集並未提供對話模擬器架構等相關資訊，我們將分析 SGD 語料中的對話行爲與綱要，添加訊息服務的對話語料。

3.1 服務及意圖

服務 (service)，或稱爲領域 (Domain)，是指對話系統所提供的功能或服務。在過去的對話系統中常見的服務有餐廳訂位、購物、旅遊景點查詢等。本研究爲了更好地掌握相關領域的語言使用情況，我們將服務分爲郵件（Mail）、行事曆（Calendar）和通訊軟體（Message）。

對話意圖 (intention) 代表了使用者在與系統進行對話時想要達成的目標或意圖。不同的服務領域可能有不同的對話意圖。例如，在郵件服務中，可能會有發送、查看郵件等意圖。在行事曆服務中，可能會有新增、查詢活動等意圖。而在通訊軟體服務中，可能會有發送訊息、查看聊天紀錄等意圖。了解使用者的對話意圖可以幫助對話系統更好地理解和回應使用者的需求，提高對話的效果和效率。

3.2 綱要建構

綱要建構是根據使用者需求和各個 API 定義服務綱要，讓對話代理能夠存取和改寫資料庫中的資料。在 (Rastogi et al., 2020) 所提的方法中，綱要清楚地定義了對話語料的本體 (ontology)，也就是定義了對話中會使用到的服務 (service)、插槽 (slot) 和意圖 (intent)。我們依據常用的郵件、行事曆和通訊軟體 API 定義了資料庫綱要，如表1所示。表中每個服務的第二列爲該服務會使用到的插槽，我們明確定義了每個插槽的敘述，並爲每個插槽添加權重，讓對話代理了解插槽間的優先權。第三列則爲該服務所擁有的意圖，每個意圖都有事務性 (transactional) 標籤。事務性意圖如添加活動、寄送郵件等非查詢的行爲 (在表中以粗體表示)，可協助對話代理存取和使用不同特性的意圖。

Mail Domain
Recipient, Subject, Sender, Content,
Copy recipient
SendMail, FindMail
Calendar Domain
Name, Date, Time, Participant,
Content, Location
AddEvent, LookupEvents
Message Domain
Contact, Group, Message
SendMessage, FindMessage

Table 1: 綱要中所使用到的插槽及意圖

3.3 對話行爲

相較於期望助理達成使用者目的的對話意圖，對話行爲 (Dialogue actions) 著重於助理和使用者在對話過程中所採取的行動，即對話系統在理解使用者的對話意圖後，根據系統的設計和能力所執行的動作。對話行爲可以包括問答、確認、請求資訊、提供資訊、提醒、建議等。我們參照 (Rastogi et al., 2020) 來定義更多樣的對話行爲。在進行對話交互時，我們以（對話行爲、服務、插槽、槽值）對話元組表示，使對話代理輸出能夠格式化。在表中可得知，INFORM、CONFIRM、OFFER、OFFER_INTENT 及 INFORM_COUNT 爲告知類的對話行爲，當代理輸出該行爲時，代理需標記當時所提及之插槽及槽值。而 REQUEST 爲請求類的行爲，需標記提及之插槽。其餘行爲由於未有任何告知及請求訊息，故插槽及槽值皆留空。

3.4 資料庫建構

在任務導向對話系統中，話語會被解析爲對話狀態並輸入至資料庫進行查詢，而系統會依據查詢結果去決定回覆策略。而由對話代理組成的 Machine-to-Machine 語料蒐集方法中，助理代理也必須藉由存取資料庫來決定下一個狀態的對話行爲。由於眞實的資料難以取得，故研究人員常以網頁爬蟲搭配統計抽樣來使資料庫逼近眞實的環境，爲了使資料庫更趨近現實的情境，本研究在政府資料開放平台 (Open Data) 抓取 1408 個活動並建構行事曆資料，通訊軟體則爬取 Line OpenChat 中的 4896 條訊息來建立服務資料庫，至於電子郵件服務，我們爬取 1926 則 PTT 網路論壇上的文章來模擬信件資訊。

3.5 對話模擬器

根據 SGD 的對話模擬器框架，本研究所使用的對話模擬器由用户和助理二個代理組成，而此二代理皆由機率自動機 (probabilistic automaton) 來互相溝通並轉移彼此的對話行爲。

在開始對話模擬前，模擬器會將各個服務初始化並選擇一個當前服務綱要的意圖，且初始化助理行爲 (*assistant_actions*) 爲 GOODBYE，使用户代理能夠進行第一輪的交互。開始進行交互時，用户代理會根據系統行爲進行狀態轉移，並回傳當前輪次的對話行爲序列 (*user_actions*)。與用户代理相同，助理代理也須根據用户的對話行爲進行狀態轉移，生成一個助理行爲序列。待二個對話代理皆完成對話交談後，我們會將目前輪次的對話大綱更新到對話歷史中，同時也會檢查本輪次的大綱是否重複出現在對話歷史中，若曾在歷史中被提及，則會讓代理重新進行當前輪次的交談。

二個對話代理將會持續進行交談直到 GOODBYE 行爲再次出現在 *assistant_actions* 中，即可完成一次完整的對話模擬。在整個對話生成過程中，我們會檢查對話歷史，以確保每輪生成的對話行爲、槽及槽值是唯一且不重複的。一旦生成的對話包含在歷史中，我們就會再次生成對話，直到產生一個全新的對話。最終，我們會將產生的對話轉換爲自然語言文本，以便進一步處理。

3.6 對話改寫

透過對話模擬器，我們可以獲得對話大綱。爲了方便標記人員使用大綱進行對話改寫，我們定義了任務敘述模板，將對話行爲轉換爲機械式對話，接著進行以下對話改寫步驟：(a) 模擬器生成對話大綱後，我們使用模板轉換時間類的槽值，使其更符合口語表達。(b) 根據對話行爲和槽位設計不同的對話模板，並將槽值插入模板中，形成機械式對話。(c) 將機械式對話交給標記人員進行改寫。

對話改寫的範例如圖3所示。我們根據前一章所定義的各個行爲描述來建構對話模板。對於需要告知槽值的行爲，如 INFORM、INFORM_COUNT 和 OFFER，我們將模板中的槽位標籤替換爲當時抽樣到的槽值。而對於需要提及某一槽位的行爲，如 REQUEST、INFORM_INTENT，我們在模板中加入該槽位的相關訊息。

獲得機械式對話後，我們設計了一個改寫系統，並聘請了兩位工讀生進行改寫。我們要求標記人員將原本分爲兩句的對話改寫爲一句，同時保持原始文意不變。我們還要求人員對每組大綱進行連貫性評分，以評估這種對話生成方法的品質。

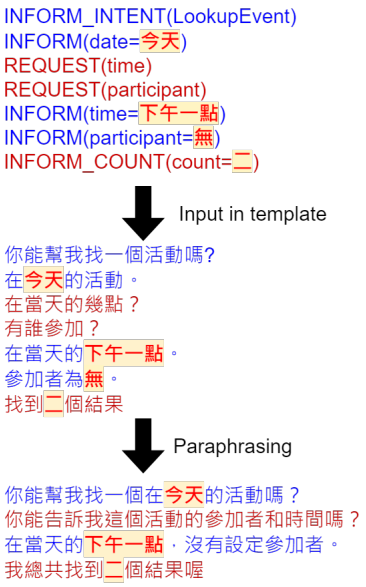

Figure 3: 對話改寫示意圖

4 統計與分析

本節比較 messageSGD 和 messageWOZ 這兩個資料集在數量統計方面的差異。我們分析了兩個資料集中的對話數量、對話輪次、平均對話輪次，並依據每組對話的輪次來得出 messageSGD 的平均行爲數量，並提供了相關結果和比較，如表2所示。

根據分析結果，在完整資料的比對上，雖然二者語料集在一組對話中的平均對話輪次相近，但在使用者及助理的平均行爲數量上（Avg. u-acts & Avg. s-acts）皆明顯高於 messageWOZ，表示此方法產生之資料提供足夠多的資訊，讓模型理解及學習如何回覆現實生活上使用者各種可能的要求。

Dataset	messageSGD			messageWOZ
	Single	Multiple	ALL	ALL
Dialogues	383	212	595	339
Turns	3634	4436	8070	4714
Avg. Turn	9.72	21.02	13.80	13.90
Avg. Acts	7.43	16.25	10.61	8.00
Avg. u-acts	3.49	7.65	4.99	3.06
Avg. s-acts	3.71	8.35	5.38	4.96

Table 2: messageSGD & messageWOZ 比較表

圖4爲 messageSGD 的對話行爲分佈，在圖中我們也可得知除了 INFORM、REQUEST 和 GOODBYE 等行爲外，其餘對話行爲皆平均涵蓋在各組對話中。而 OFFER、INFORM_COUNT 及 CONFIRM 針對不同事務性意圖的對話行爲也反映了使用者 INFORM 行爲的數量。

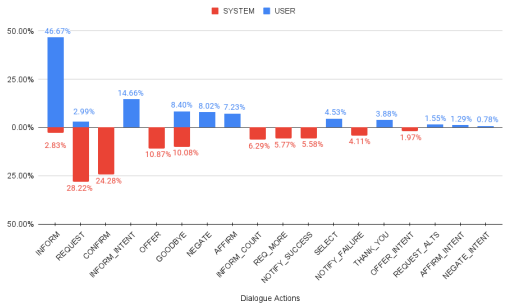

Figure 4: messageSGD 對話行爲標籤分佈

我們分析了 messageSGD 的對話輪次分佈和對話行爲分佈，如圖5所示。結果顯示，在我們的數據集中，單領域的對話平均有 9.72 個回合，多領域的對話平均有 21.02 個回合。此外，除了特定的對話行爲外，其他對話行爲均勻地分佈在各組對話中。

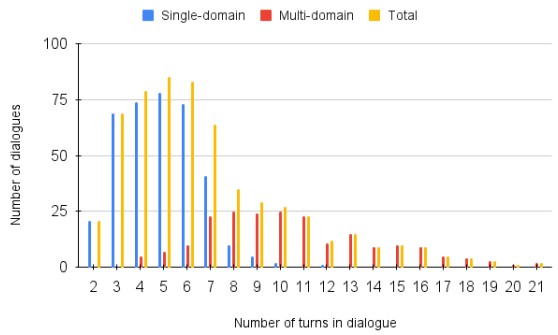

Figure 5: messageSGD 對話輪次分佈

由於我們使用對話模擬器進行對話大綱的生成，對話使用者和助理的行爲難免會有不連貫的疑慮。因此我們讓標記人員進行對話改寫時，也順便對當前改寫的整組對話進行連貫性評估，圖6爲標記人員對單領域、多領域及完整語料進行評估的平均分數。由圖可觀察到，當單領域的對話輪次增加，整組對話會越不連貫。

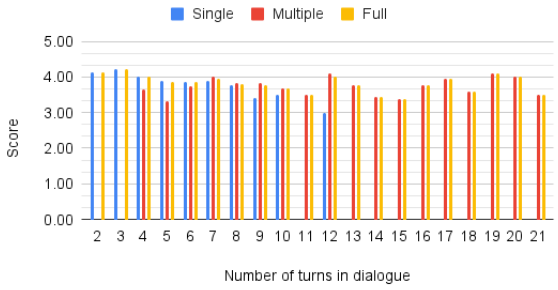

Figure 6: messageSGD 連貫性評估

本節比較了 messageSGD 和 messageWOZ 這兩個資料集在數量統計上的差異。我們發現這兩個資料集在對話數量、對話輪次、平均對話輪次和行爲數量等方面存在顯著差異。研究者在使用這些資料集進行對話系統相關研究時應該考慮這些差異的影響。

5 任務導向對話系統

以下章節將會由資料前處理開始分別介紹我們是如何建構任務導向的對話系統的，以及設定各個模組間的輸入輸出，使整個系統的四個 TOD 任務能夠正確的完成任務，並能依據使用者話語給予適當的回覆。

5.1 資料前處理

在 messageSGD 及圖7中，所有的對話行爲和對話狀態皆以字典及串列的資料格式進行儲存，但由於 T5 爲以 Transformer 爲基底的文字對文字（Text-To-Text）模型，我們必須將輸入輸出轉換爲序列格式，才能使各個任務的 T5 模型進行學習。我們使用 (Zhu et al., 2022) 提出的方法來將語料中的標記資料進行序列化，如表3所示。序列化的對話行爲格式爲 [行爲][服務]([槽][槽值],...)，序列化的對話狀態格式爲 [服務]([槽][槽值],...)，若同一輪對話中包含多個行爲及狀態，我們則用分號分隔不同的行爲或不同服務的狀態。

5.2 Model Tuning

由於序列化結構性資料，本文採用了 (Zhu et al., 2022) 的方法，使用 mT5（multilingual T5）(Xue et al., 2020) 模型來微調四個 TOD 任務。mT5 是一種多語言的預訓練模型，具有強大的自然語言處理能力，可以應用於包含中文在內的 101 種語言和任務。圖7展示了這四個 TOD 任務的輸入和輸出的對照關係。爲了能讓模型能更了解和更精確的處理任務

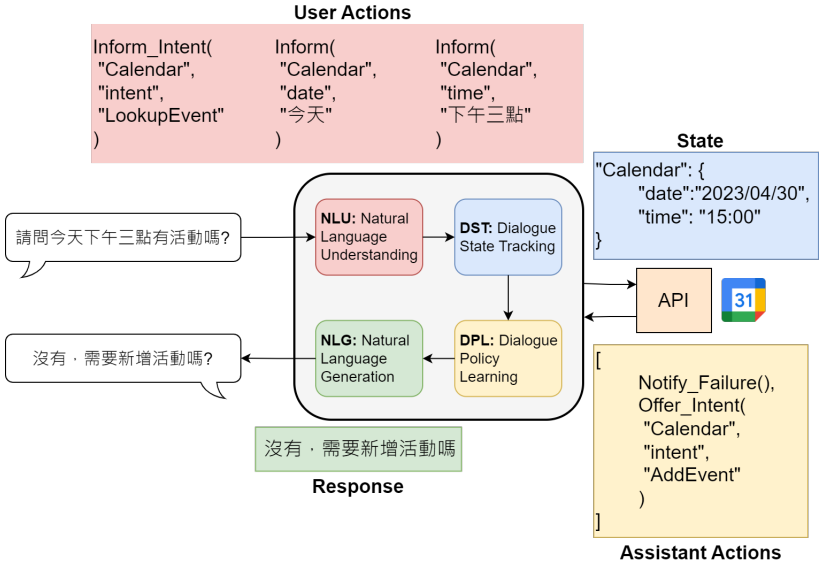

Figure 7: TOD 系統在 messageSGD 訓練下的輸出範例

user: 請問今天下午三點有活動嗎?
system: 我沒有找到任何結果
user: 那明天下午一點有嗎? 謝謝
DA-U: [INFORM][Message]([date][明天], [time][下午一點]); [THANK_YOU][ ]([ ][ ])
State: [Message]([date][明天],[time][下午一點], [intent][LookupEvent])
DA-S: [INFORM_COUNT][Calendar]([count][1])
system: 我有找到一則活動喔

Table 3: 序列化對話行爲和狀態

導向的對話，我們參考了 InstructDial(Gupta et al., 2022) 的方法。InstructDial 爲一個用於對話的指令調整框架，使用 48 種不同的對話任務進行訓練，爲增加模型在不同對話任務上的跨任務泛化能力，該方法在每個任務的輸入之前添加了任務定義 (指令)、特殊標記，使模型能依據定義了解當前所要執行的任務，也能根據特殊標記了解各個輸入的用途。

本研究更換了 InstructDial 的基底模型，使用 PromptCLUE(?) 來分別對四個 TOD 任務進行微調。PromptCLUE 是一個基於 T5 的生成式預訓練模型，它使用了千億中文 token 的語料，累計學習了 1.5 萬億中文 token，並且在數百種不同類型的 NLP 任務上進行了 Prompt 任務式訓練。它具有較好的零樣本學習能力和少樣本學習能力，可以自定義標籤體系和採樣方式，支持理解、生成和抽取等多種任務。我們將每個模組需要的輸入資料整理成表4的格式進行微調:

自然語言理解任務爲對話行爲的預測，其輸入是對話的上下文 (Context)，即對話歷史，包含了對話中先前助理通知、用户的請求等信息。而輸出是用户當前的對話行爲 (User Dialog Act，簡稱 DA-U)，它是對用户的發言進行解析和分類，以便整個系統能夠理解用户的意圖和需求。

其次，對話狀態的輸入輸出映射。對話狀態追蹤任務是依據對話的上下文來紀錄對話狀態 (State)。對話狀態包含了有關對話進展和系統内部知識的信息，它被用於跟踪和更新對話中的重要參數和資訊。

爲策略生成輸入包括對話狀態 (State)、對話的上下文 (Context) 和資料庫的查詢結果 (DB)。我們期望該模型能夠決定下一步系統的對話行爲序列 (System Dialog Act，簡稱 DA-S)，在完整的 TOD 系統中，它指示對話系統在對話中做出合適的回應和行爲。

最後，自然語言生成任務的輸入包括助理的對話行爲 (DA-S) 和對話的上下文 (Context)，而輸出是系統生成的回應 (Response)。這一任務旨在生成符合對話情境和要求的自然語言回應。

6 指標與結果

本研究使用 Text-to-Text 模型進行任務導向對話，將對話行爲、對話領域和槽值皆進行序列化，並將同一輪次的對話視爲一個元組 (對話行爲、對話領域、槽和槽值)。

6.1 評估指標

在評估 NLU、DST 與 DPL 模型時，我們除了分開計算對話行爲、對話領域、槽和槽值的召回率、精確度及 F1-score 外，我們也將預測元組(Predicted tuples)與正確答案(Golden

Task	Standard Input	Instruction Prompt
NLU	$U_{t-3}\backslash nU_{t-2}\backslash nU_{t-1}\backslash nU_t$	對話理解: 依據對話歷史預測對話行爲。 $[HISTORY]$ $U_{t-3}\backslash nU_{t-2}\backslash nU_{t-1}\backslash nU_t$ $[QUESTION]$ 對話行爲:
DST	$U_{t-3}\backslash nU_{t-2}\backslash nU_{t-1}\backslash nU_t$	對話狀態追蹤: 依據對話歷史預測對話狀態。 $[HISTORY]$ $U_{t-3}\backslash nU_{t-2}\backslash nU_{t-1}\backslash nU_t$ $[QUESTION]$ 對話狀態:
DPL	$State\backslash nU_{t-3}\backslash nU_{t-2}\backslash nU_{t-1}\backslash nDB_{result}$	對話決策: 依據對話狀態、對話歷史和資料庫結果來決定系統的對話行爲。 $[STATE]$ $State[HISTORY]$ $U_{t-3}\backslash nU_{t-2}\backslash nU_{t-1}$ $[DATABASE]$ DB_{result} $[QUESTION]$ 系統行爲:
NLG	$DA-S\backslash nU_{t-3}\backslash nU_{t-2}\backslash nU_{t-1}\backslash nU_t$	回覆生成: 依據對話行爲及對話歷史來產生回覆。 $[ACTION]$ $DA-S$ $[HISTORY]$ $U_{t-3}\backslash nU_{t-2}\backslash nU_{t-1}$ $[QUESTION]$ system:

Table 4: Standard 與 Instruction Prompt 輸入資料比較表

tuples）進行評估，計算全面的 F1（overall F1）與正確率（Accuracy）。

由於本研究所使用的對話模型皆產生序列輸出，我們需計算模型預測的文本是否與參考文本相同，故我們使用精準匹配度（Exact Match，EM）來評估序列生成模型的效能。Exact Match 是一種二元指標，衡量模型生成的文本與參考文本之間是否完全相同，即計算預測文本與參考文本相同的比例。

在 NLG 任務中，模型輸入爲對話歷史，輸出爲助理回覆，由於這種任務爲類似機器翻譯的文本對文本生成任務，我們採用 BLEU(Papineni et al., 2002) 和 BERTScore(Zhang* et al., 2020) 來進行評估。BERTScore 是一種用於評估自然語言生成任務的指標，例如文本摘要和機器翻譯。它通過比較預測文本和參考文本的上下文嵌入（embedding）來計算。嵌入是使用預先訓練的 BERT 模型生成的。兩個句子之間的相似度是通過預測文本和參考文本嵌入之間的餘弦相似度來衡量的。

6.2 實驗結果

我們將 messageSGD 使用 k 折交叉驗證（k-fold cross-validation）的方式分爲 5 個資料集進行訓練、驗證和測試。每個驗證和測試集包含 100 組對話，其餘對話用於訓練。每個輸入資料包含最近的 5 輪對話（上下文大小 =5）以進行模型訓練和預測。每個任務訓練 10 個 epochs，訓練和測試的批次大小設定爲 10。

我們使用標準輸入和指令提示兩種方法來訓練四個 TOD 任務，並進行比較，如表5所示。從比較結果中可以看出，在自然語言理解任務中，對話行爲、領域和槽值的預測效果優於標準輸入方法。正確率和精確匹配度也顯示出指令提示方法能夠幫助模型更好地理解當前任務的執行。

在對話狀態追蹤任務中，指令提示的輸入方法仍然優於標準輸入。儘管效果有所提升，但正確率和精確匹配度並不突出。我們觀察到 DST 模型的輸入和輸出後發現，由於語料中的每個槽值都是非類別型的，即沒有固定的數值或數值範圍，因此模型難以準確提取每段對話中的狀態資訊。

策略學習任務的效果與 DST 類似。儘管對話領域的 F1-score 略低於標準輸入方法，但其他指標仍優於後者，能夠根據當前對話提供更適當的回覆策略。由於策略回覆受當前對話情境的影響而產生多變性，因此在整體 F1-score 和準確率等綜合指標上的效果並不理想。

在自然語言生成部分，指令提示方法在 BLEU Score 上比標準輸入方法高出 6.54。然而，在 BERT-Score 上的差異不大，這表明經過指令提示微調的模型使得回覆更接近參考文本，但對於 BERT 等預訓練模型來說，兩種方法所生成的回覆差異不大。

在對 T5 模型進行微調的過程中，我們發現對話歷史的多寡對各個對話任務的效能有顯著影響。以 fold-1 的資料集進行實驗時，我們加入了 5、10 和 20 輪次的對話歷史，觀察每個任務的效能變化，如表6所示。

在對話理解任務中，我們觀察到隨著輸入對話的增加，T5 模型在該任務上的分類表現明顯下降。這是因爲對話理解任務需要根據相鄰的對話上下文進行行爲分類和槽填充，當輸入歷史過多時，模型的判斷容易混淆，導致錯誤的預測。

對於對話狀態追蹤任務，模型不僅需要根據

NLU	F1	Acc	EM
STD	84.23	76.03	74.34
Instruct	89.37	83.36	82.05
DST	slot F1	Acc	EM
STD	70.40	38.73	35.54
Instruct	70.52	41.44	39.66
DPL	F1	Acc	EM
STD	66.85	54.55	53.13
Instruct	69.88	56.93	55.07
NLG	BLEU	BERT-F1	
STD	25.89	78.54	
Instruct	32.43	78.45	

Table 5: TOD 任務結果比較

當前對話更新對話狀態的資訊，還必須追蹤和保留先前對話中提及的資訊。增加輸入對話歷史的長度可以幫助模型更好地掌握需要追蹤和記錄的對話歷史資訊。但加入的對話歷史過多時，模型可能因爲對話歷史資訊過多而產生混淆，導致效能下降。

在過去的回覆策略生成任務中，研究人員未將對話歷史視爲模型的參考。然而，本研究認爲對話的決策往往會受到過去對話提及的資訊影響。我們在本研究中比較了加入對話歷史的影響。實驗結果證明，增加對話歷史的輪次能更好地使模型做出適當的決策。

對話生成任務需要根據對話歷史和當前助理行爲將助理行爲轉換爲自然語言。與對話理解任務相似，輸入的參考歷史越多，模型越容易混淆，產生與當下對話情境不符的回覆。表中更清楚地展示了對話生成任務中對話歷史對模型的影響。隨著加入的對話歷史數量增加，BLEU 和 BERT-Score 也相應降低。

根據本章節的分析結果，我們比較了 messageSGD 和 messageWOZ 這兩個資料集在數量統計方面的差異。結果顯示，相較於 messageWOZ，messageSGD 在對話行爲數量上有明顯的優勢，提供了更豐富的資訊，讓模型能夠更好地理解和學習回覆現實生活中使用者各種可能的要求。

在實驗中我們使用不同的輸入方法進行四個對話任務的比較，還觀察了輸入對話歷史的多寡對模型性能的影響。結果顯示，使用指令提示的輸入方法在自然語言理解（NLU）、對話狀態追蹤（DST）和策略學習（DPL）任務上的效能優於標準輸入方法。

NLU	F1	Acc	EM
context-5	**91.36**	**86.50**	**85.63**
context-10	89.08	83.16	81.71
context-20	89.64	84.33	83.02
DST	Slot F1	Acc	EM
context-5	71.28	41.65	39.04
context-10	**80.08**	**51.69**	**47.90**
context-20	78.82	44.41	46.73
DPL	F1	Acc	EM
context-5	70.54	**58.13**	56.03
context-10	70.79	57.83	**57.10**
context-20	**70.98**	57.69	56.52
NLG	BLEU	BERT-F1	
context-5	**29.40**	**78.18**	
context-10	27.24	78.04	
context-20	21.17	76.50	

Table 6: 各 TOD 任務在 context 多寡之差異

7 結論

本研究使用綱要引導建立了一個任務導向的語料蒐集環境。通過綱要引導建立的對話模擬器提供了一種有效的方式來生成任務導向的語料，該語料可用於訓練和評估智慧助理系統。與人工對話和標記相比，我們比較了 messageSGD 和 messageWOZ，發現透過綱要和模擬器生成的語料能夠產生多樣化的對話場景，並能夠控制對話的內容和流程，從而提高對話系統的性能和適應能力。

在實驗部份，我們使用 T5 爲基底的文字對文字模型來建構對話系統。我們在模型的輸入文本中加入每個任務的定義及特別標記，使對話系統能在訓練參數較少的情況下，也能更準確的分析對話。

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2):25–35.

Prakhar Gupta, Cathy Jiao, Yi-Ting Yeh, Shikib

Mehri, Maxine Eskenazi, and Jeffrey Bigham. 2022. InstructDial: Improving zero and few-shot generalization in dialogue through instruction tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 505–525, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

J. F. Kelley. 1984. An iterative design methodology for user-friendly natural language office information applications. *ACM Trans. Inf. Syst.*, 2(1):26—41.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8689–8696.

Antoine Raux, Brian Langner, Dan Bohus, Alan W Black, and Maxine Eskenazi. 2005. Let's go public! taking a spoken dialog system to the real world. In *in Proc. of Interspeech 2005*. Citeseer.

Stephanie Seneff and Joseph Polifroni. 2000. Dialogue management in the mercury flight reservation system. In *ANLP-NAACL 2000 Workshop: Conversational Systems*.

Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. 2018. Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871*.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Qi Zhu, Christian Geishauser, Hsien-chin Lin, Carel van Niekerk, Baolin Peng, Zheng Zhang, Michael Heck, Nurul Lubis, Dazhen Wan, Xiaochen Zhu, et al. 2022. Convlab-3: A flexible dialogue system toolkit based on a unified data format. *arXiv preprint arXiv:2211.17148*.

Qi Zhu, Kaili Huang, Zheng Zhang, Xiaoyan Zhu, and Minlie Huang. 2020. CrossWOZ: A large-scale Chinese cross-domain task-oriented dialogue dataset. *Transactions of the Association for Computational Linguistics*, 8:281–295.

葉丞鴻, 李聿鎧, and 張嘉惠. 2022. 多領域任務導向用户語音助理對話收集系統. TAAI.