Computational Linguistics and Chinese Language Processing
Vol. 24, No. 2, December 2019, pp. 63-78

基於訊息回應配對相似度估計的聊天記錄解構

Chatlog Disentanglement based on Similarity Evaluation Via Reply Message Pairs Prediction Task

劉至咸、張嘉惠*

ZhiXian Liu and Chia-Hui Chang

摘要

一般而言，為建立 Retrieval-based 聊天機器人，我們可以從聊天紀錄中來建立所需的問答配對（Question-Answer Pair），然而問答配對並非完全連續地呈現在聊天紀錄中，不同內容的問答配對可能互相穿插，而從互相穿插的訊息中分離出不同子題的會話任務稱為對話解構（conversation disentanglement）。現有的對話解構研究大多透過計算兩個訊息的相似度來解決問題，過去研究將問題定義為判斷兩則 Reddit 訊息是否屬於相同主題的對話，但其所提出的模型對於未見過訊息的效能很差。實務上我們發現即使是使用者，在沒有上下文的情況下，要單從兩個給定訊息，判定其是否屬於相同會話是非常困難的。但若我們的目標是預測兩則訊息是否為回覆關係，則使用者判斷的一致性效能相對的較高。因此在本篇論文中，我們使用 IRC 與 Reddit 資料集進行實驗，並使用聊天記錄進行對話解構。利用 Reddit 回覆標記合成的資料集提供大量訓練資料建立模型，最後透過 BERT 模型在新定義的回覆關係預測上獲得良好的效能。

Abstract

To build a Retrieval-based dialog system, we can exploit conversation log to extract question-answer pairs. However, the question-answer pairs are hidden in the conversation log, interleaving each other. The conversation task that separates different sub-topics from the interspersed messages is called conversation disentanglement. In this paper, we examined the task of judging whether two

* 國立中央大學資訊工程學系
Department of Computer Science and Information Engineering, National Central University
E-mail: zhixian989@gmail.com; chia@csie.ncu.edu.tw

Reddit messages belong to the same topic dialogue and found that the performance is worse if training and testing data are splitted by time. In practice, it is also a very hard task even for human beings as there are only two messages and no context. However, if our goal is to predict whether a message is a reply to the other, the problem becomes much easier to judge. By changing the way of data preparation, we are able to achieve better performance through DA-LSTM (Dual Attention LSTM) and BERT-based models in the newly defined Reply prediction task.

關鍵詞：對話解構、回覆關係預測、BERT 模型應用

Keywords: Chatlog Disentanglement, Reply Relation Prediction, BERT Neural Model

1. 緒論 (Introduction)

企業線上客服系統的需求提升，使得客服系統的人力成本增加，若能夠建立自動對話系統或聊天機器人（Chatbot）則可以有效降低人力成本。建立自動問答的 QA 系統或聊天機器人的模型可分為 Retrieval-based 與 Generation-based 兩種，Retrieval-based 的方法需要有預先定義的問答集，包含問題及對應的答案，適合特定領域的對話系統，因此我們希望能從對話記錄中抽取出所需要的大量問答配對，建立 Retrieval-based 的客服系統。從互相穿插的訊息中分離出相同主題的會話任務稱為對話解構（conversation disentanglement）。

典型會話訊息互相交錯的現象大多發生在多人聊天情境中，以圖 1 為例，表中為真實世界的 IRC 聊天室片段，其中 Thread 表示使用者正在參與的會話。由於會話訊息互相交錯，大部份使用者也會在訊息前加入回應的對象，以方便使用者區分不同會話的進行。

Speaker	Message	Thread
Elli	Any idea why 'passwd' would ask for a new password four times?	T77
Priscila	Elli: A hacked version.	**T77**
Melda	is there a way to get ls -l to print full path in each response?	T78
Arlie	Julietta, do whatever you want ... its an ethernet packet	T75
Leota	Jeanice: i had to replace most of the the startup scripts with just echo boo, the problem seemed to be with the kernel not being detected or something	T71
Melda	so it'll show /home/user/filename.jpg at the end of every single line?	T78
Elli	Priscila, yeah that could be a possbility except I just recompiled it from sources, thinking just that	T77
Elli	And still the same behavious	T77
Priscila	Elli: Hmmm. Weird indeed ..	**T77**
Julietta	Arlie, well, i can't, to actually inject valid packets i would need to modify the sockets state	T75
Jeanice	Leota: strange... and what errors did you get when you tried to compile?	T71
Priscila	Melda: ls -l /home/user/*	**T78**

圖 1. 真實世界聊天室的片段
[Figure 1. Conversation log in real-word chat room]

圖 2 為客戶（Client）與客服人員（Staff）兩人之間的實際對話紀錄，其中 Topic 表示訊息的內容分類，相同 TopicID 的訊息（Messages）為內容一致的會話子題（conversation），因此理想的問答配對應由相同 TopicID 的訊息所組成，如訊息 ID (2,8)、(4,6)與(5,9)的配對，不過我們更關注於客服人員如何回應客戶的描述配對(1,2)、(3,4)、(3,5)等，從圖中我們可以觀察到即使在兩人的對話中，也有不同問答子題同時進行著。

ID	Author	Messages	TopicID
1	Client	can't connect to nas, any suggestions?	T1
2	Staff	Do you know what the IP of the NAS was?	T2
3	Client	The device was setup by an admin who is no longer here. I renamed the machine and to setup a new IP and restarted. now it's not responding.	T1
4	Staff	Can you make sure the NAS is powered on?	T3
5	Staff	What is the model of the NAS?	T4
6	Client	yes it was.	T3
7	Client	but he did it incorrectly in my opinion.	T1
8	Client	he didn't set a static IP he set it for DHCP and made a reservation for it on the DHCP server.	T2
9	Client	ts-ec880u-rp	T4

圖 2. 客戶與客服人員實際問答紀錄

[Figure 2. Conversation log between client and staff]

現有的研究在處理對話解構問題時，使用的方法大多是將新進訊息與現有會話中的訊息逐一計算相似度，此相似度表示訊息屬於相同會話的可能性，最後利用計算的結果決定新進訊息是否開啟新的主題（會話），或是屬於現有的會話。Jiang 等於 IJCNLP2018 提出的連體分層卷積神經網路模型 SHCNN（Siamese Hierarchical Convolutional Neural Network）(Jiang, Chen, Chen & Wang, 2018)來進行相似度的計算，實驗結果顯示模型具有相當高的效能。但 Jiang 等人的研究在進行訓練資料與測試資料的分割時，採用隨機抽取訊息配對的方法，使得測試資料配對中的訊息，與訓練資料配對中的訊息重疊。因此他們所使用的實驗設計無法評估模型對於未來（未見過）訊息的效能表現，在後續的實驗中我們也證實 SHCNN 在預測未來訊息的效果非常差，因此無法運用於實務中。

為了瞭解「預測兩個訊息是否屬於相同會話」的難度，我們從測試資料中隨機選取數個樣本進行人工標記。在標記的過程中我們發現許多對話即使人類也無法有效分辨其是否來自同樣對話。其原因出在訊息配對中無上下文資訊，以致於難以猜測訊息所正在討論的主題。且人類在交談時較寫文章更為隨性，而使得這項任務更為困難。我們發現比起判斷兩個訊息是否來自相同會話，人們更擅長於判斷某個訊息是否正在回覆另一個訊息，這樣的「訊息回覆關係預測」在本文中稱為回覆預測任務，而「預測訊息是否屬於相同會話」在本文中稱為相同會話任務。

我們發現在回覆預測的任務中，人可以借助語意在沒有上下文的情況下判斷訊息的回覆關係。同時由於回覆預測任務與相同會話任務具有相同的輸入與輸出形式，使得回覆預測任務可以運用在現有的對話解構演算法中。實務上，各種模型在回覆任務中也普

遍有較好的表現，顯示出深度學習模型能夠處理這樣的問題。綜合來說，本文透過問題的重新定義，改善 SHCNN 對於未知對話解構任務的效能，並比較包括 BERT 等的數種類神網路架構。實驗結果顯示在相同會話任務與回覆預測任務中，我們觀察到 BERT 效能均優於其他模型的表現。

2. 相關研究 (Related Work)

現有的對話解構研究中，最常見的解決方法為建立模型來計算兩個訊息的相似度，用來判定一個新進的訊息是屬於已有的會話或是即將開始新的會話。然而訊息相似度的計算相當依賴訊息的表示方式，若我們能夠找出夠好的訊息表示方式，則能夠提升模型的效能，因此我們將分別探討對話解構與訊息表示兩個議題。

過去已有許多對話解構的研究，Allan 等人(Allan, 2002)以主題檢測和追蹤方式來決定新進的訊息屬於現有的會話或開啟新會話。根據 Shen 等人的研究(Shen, Yang, Sun & Chen, 2006)，訊息在相同的會話中，通常具有較高的相似度。Wang 等人(Wang & Oard, 2009)發現相同會話中的訊息通常具有類似的上下文，並以此來提高訊息相似度計算的效能。但由於相近的訊息會有部分重疊的上下文，這樣的作法會影響相似度計算，進而使的效果變差。Elsner 等人(Elsner & Charniak, 2008) (Elsner & Charniak, 2010) (Elsner & Charniak, 2011)則是透過語言特徵（linguistic features）來計算訊息的相似度。其中，上述的這些研究都採用詞袋（bag-of-words）的表示方式，而無法捕捉詞與詞間的關係。

Mehri 等人(Mehri & Carenini, 2017)採用隨機森林方式建立不同的分類器，並引入循環神經網路（RNN）與外部資料訓練「預測下個語句」的分類器（Next Utterance Classification）(Lowe, Pow, Serban & Pineau, 2015)。其中 in-thread 分類器會以 Same thread 分類器、Reply 分類器與 RNN 的輸出結果做為輸入，將訊息分類到所屬的會話中，不過此方法效能並不理想。

Jiang 等人(Jiang *et al*., 2018)採用連體分層卷積神經網路（SHCNN）進行訊息相似度的計算。其架構中包含兩個分層的卷積神經網路（CNN），分別捕捉訊息的低階與高階特徵，並串聯低階特徵與高階特徵作為訊息的表示（representation）向量。經過計算訊息表示向量的絕對值差值後，使用全連接層輸出兩個訊息屬於相同會話的機率。其實驗結果在 IRC 與 Reddit 資料有相當好的表現。但此模型對於未曾出現於訓練資料的訊息效能不佳，在實務上無法應用，因此我們需要其他更好的句表示（sentence representation）。

Devlin 等人(Devlin, Chang, Lee & Toutanova, 2018)提出了一種基於 Transformer 結構的深度學習模型 BERT（Bidirectional Encoder Representations from Transformers）。BERT 以 Masked 語言模型任務與接續／下句預測（next sentence prediction）任務進行預訓練，並將預訓練後的參數作為下游任務（downstream task）的初始值進行微調（Fine-Tuning）。這樣的方法在許多自然語言處理任務上獲得成功。

3. 問題定義與資料結構 (Problem Definition and Datasets)

此論文中所使用的資料集有三個，分別為 IRC、Reddit 與 QNAP 客服對話紀錄。其中 IRC 與 Reddit 為公開的資料集。IRC 資料集是由 Elsner 等人(Elsner & Charniak, 2008)所建立的人工標記資料。共約 6 小時的對話紀錄。每個標記者需要為每個訊息標記其所屬的會話，其中在相同會話裡的訊息所討論的內容是一致的，表示這些參與者正互相關注他們的對話中。Reddit 資料集是由 Reddit 文章中的評論組合而成的合成資料集，為 Jiang 等人(Jiang *et al.*, 2018)所提出的想法，將每條評論視為聊天訊息，並將同一則文章內的所有評論視為在相同會話裡的訊息。客服對話資料集為 QNAP 客服中心實際聊天紀錄，我們將不同討論內容內容視為相異的會話。

3.1 Reddit資料集與訊息配對的產生 (Message Pair Generation for Reddit Dataset)

我們參照 Jiang 等人(Jiang *et al.*, 2018)所提出方法來構建 Reddit 資料集。根據 Aoki 等人(Aoki *et al.*, 2006)的研究，小團體的社交對話中平均同時進行的會話為 1.79 個，而 Reddit 活躍的文章數與之相比可能多達數百篇。因此在合成 Reddit 資料集時，我們移除一些文章，使得任意時間內同時進行的會話數更貼近人實際交談的情況。此外，我們也刪減評論較多的文章，因為當越多的評論出現在文章中時，這些評論越容易偏離文章主題，我們希望在相同文章內的評論內容是一致的。

在相同會話任務中，我們需要從資料集中列舉訊息配對，但若列舉對話中所有可能的訊息配對，會導致產生的配對數量過於龐大，也會使得相異會話的訊息對數量遠多與相同會話的訊息對。因此我們只列舉時間間格 T 內的訊息配對。換句話說，對於任一訊息對（m_i, m_j），m_i 與 m_j所發表的時間t_i 與 t_j，其時間差$|t_i - t_j| < \mathrm{T}$。在此研究中我們將 T 設為一小時，與 Jiang 等人的實驗設置相同。

在回覆預測任務的準備資料中，我們對於 Reddit 中的每條評論都各自挑選五個評論作為配對，而五個評論中僅有一個為真實的回覆對象，其餘其他四個訊息從時間間隔 T 內的訊息中隨機挑選，這四個訊息有可能為來自相同主題或不同主題的評論。

3.2 訓練、驗證與測試資料的分割 (Training and Testing Data Splitting)

在分割訓練資料、驗證資料以及測試資料時，Jiang 等人(Jiang *et al.*, 2018)選擇從所有的聊天訊息對中，以隨機選取的方式，分割訓練資料、驗證資料與測試資料。但以隨機選取的方式切割資料存在著問題，如圖 3 所示，訊息在經過配對後同一訊息可能會出現在許多配對中，這意味著以隨機選取的方式進行資料分割時，測試資料有可能出現在訓練及驗證資料中。換言之，隨機選取方式分割出的測試資料，僅能驗證模型是否能對已知的訊息進行預測。相反的若依照訊息的發表時間分割測試資料（如圖 4 所示），則測試資料中的訊息對於訓練過的模型而言為未來的訊息。

在實驗中我們會對隨機選取以及依照發表時間兩種分割方式進行分析。在兩種分割方式中，皆以 80%作為訓練資料、10%作為驗證資料、10%作為測試資料的比例進行分割。在使用隨機選取分割方式的實驗中，僅以一半的訓練資料進行訓練（40%總資料量），以降低訊息重複出現在訓練資料及測試資料的影響。

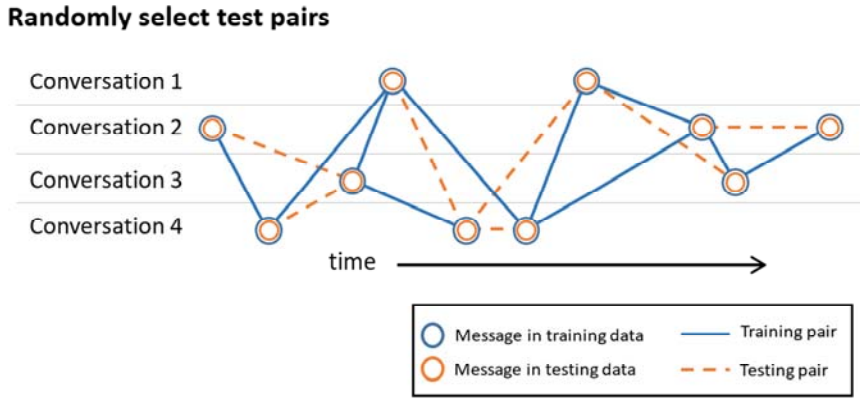

圖 3. 隨機選取的方式分割訓練與測試資料
[Figure 3. Training and Testing Splitting in Random]

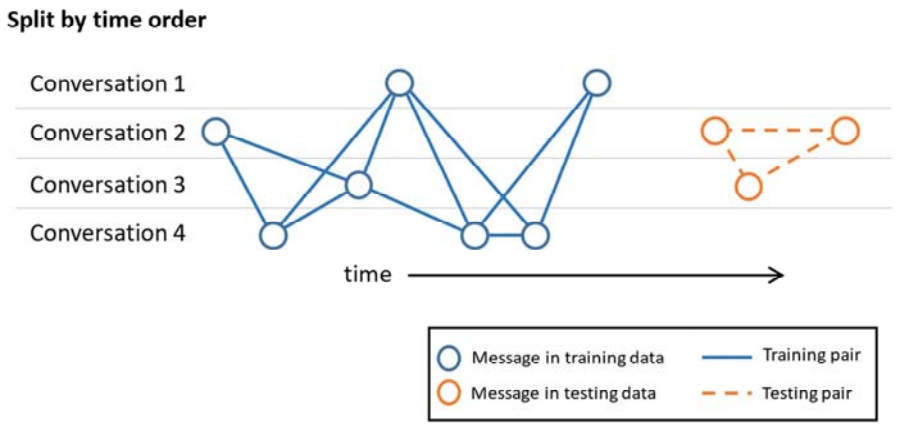

圖 4. 以發表時間的方式分割訓練與測試資料
[Figure 4. Training and Testing Data Splitting by Time]

在相同會話任務中，我們使用 IRC 與 Reddit 作為訓練及測試資料，表 1 呈現經過處理的資料數據。其中 Reddit 的資料為合成資料，因此有較多的合成資料，而 Reddit 的三個子討論版分別反映出不同的熱門程度。在回覆預測任務中，會話數量如表 2 所示，反映出 Reddit 三個子討論版間的原始資料比例。

表 1. 相同會話任務中的四個資料集分析
[Table 1. Datasets for Same Conversation Prediction Task]

資料集	Reddit			IRC
	gadgets	iPhone	politic	
會話數	468	529	6,197	159
訊息數	11,071	10,261	148,942	1,865
發言者數	6,387	4,506	28,365	183
平均同時進行的會話數	5.28	8.73	13.95	2.75
總配對數	487,695	507,226	4,492,361	79,682
相同會話配對數	118,889	111,145	1,226,863	5,390

表 2. 回覆預測任務中的三個資料集分析
[Table 2. Datesets for Reply Message Pair Prediction Task]

資料集	Reddit		
	gadgets	iPhone	politic
會話數	174	1,524	27,361
訊息數	6,007	35,286	800,619
發言者數	3,953	11,355	72,787
總配對數	19,169	103,179	2,423,900
真實回覆關係配對數	3,835	20,636	484,780

4. 模型架構 (Models)

在相同會話任務中，我們將接受兩個訊息m_1與m_2作為輸入，並預測訊息m_1與m_2來自相同會話的機率$P(same|m_1, m_2)$，其中訊息m為一文字符號序列。在回覆預測任務中，以訊息m_1與m_2作為輸入並輸出訊息m_2是訊息m_1的回覆的機率$P(m_2\ reply\ m_1|m_1, m_2)$，與相同會話任務同為二元分類問題，具有相同的輸入與輸出形式。因此所有的訓練例子可被表示為集合$\{(m_i, m_j, y)\}$，其中$y \in \{0,1\}$為對應的標記，表示為m_i與m_j是否來自相同會話或m_i是否為m_j的回覆對象。

對於任意的輸入訊息m可被表示為文字符號序列$m = \langle w_1, w_2, \dots, w_n \rangle$，而所有的文字符號將經過預訓練好的詞嵌入，轉換成固定維度d的向量（$w \in \mathbb{R}^d$）。若訊息中的詞不存在於預訓練好的詞彙中，則該詞將以未知詞符號（UNK）取代。

我們使用雙向 LSTM 做為訊息的編碼器，利用隱藏層，將輸入訊息m編碼成一個h維的向量z，表示為$z = \text{BiLSTM}(m)$，而這個向量z將被作為句子嵌入（sentence embedding）。注意配對中兩個不同訊息被編碼時，所用的編碼器是相同的，意即編碼兩個訊息的編碼

器權重共用（shared weight）。

輸入的訊息m_1與m_2分別被編碼成z_1與z_2後，採用二層全連接層作為最後的輸出層，計算方式如下：

$$P(same|m_1, m_2) = \sigma(\text{ELU}([z_1, z_2]^T W_0 + b_0)W_1 + b_1) \quad (1)$$

其中$W_0 \in \mathbb{R}^{2h \times h}$，$W_1 \in \mathbb{R}^{h \times 1}$，以上模型的設計如圖 3 中藍色線條所示。

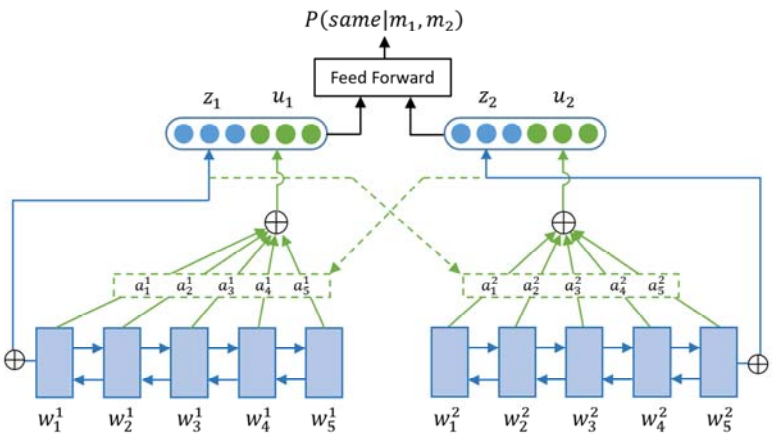

圖 5. 應用注意力機制的 BiLSTM 架構圖
[Figure 5. BiLSTM Model with dual attention mechanism]

Dual Attention LSTM:為更好的捕捉配對中兩個訊息的關係，我們加入互相關注的注意力機制。讓訊息m_1的 BiLSTM 輸出向量表示為e_i^1，其中$i = 1 \ldots L$，L為訊息長度，並給予另一個由m_2編碼成z_2的句子嵌入，用來計算m_1的注意力權重α_i^1及其表示向量u_1，計算方法如下：

$$\alpha_i^1 = softmax_i({z_2}^T e_i^1) \quad (2)$$

$$u_1 = \sum_{i=1}^{n} \alpha_i^1 e_j^1 \quad (3)$$

並且使用相同的方法計算u_2，其中u將作為z的額外訊息使用。注意力機制的計算如圖 3 綠色部分所示。此應用注意力專注的 BiLSTM 結構將作為我們所使用的第一個傳統深度學習模型，在實驗中將以 Dual-Attention LSTM (簡稱 DALSTM)表達。

CNN+LSTM: 我們將在前述的循環神經網路模型基礎上，但不使用注意力專注的情況下，加入單層卷積神經網路（CNN）作為我們所使用的第二個傳統深度學習模型，加入的卷積層會在輸入被送到循環神經網路前，先行對輸入進行編碼。首先，輸入的訊息m_1透過詞嵌入轉換將被表示為矩陣$M_1 \in \mathbb{R}^{d \times L}$（$d$為詞嵌入維度，$L$為訊息長度）。我們使用$h_c$個大小為$d \times k$ 的 kernel，因此卷積層將會輸出$h_c \times L$大小的 feature map，輸出的 feature map 會套用 Gated Linear Unit （GLU） (Dauphin, Fan, Auli & Grangier, 2016)作為非線性轉換單元（Activation function）並加上 Layer Normalization。最後$h_c \times L$的 feature

map 可被分為L個維度為h_c的向量，作為循環神經網路的輸入。同樣的，訊息m_2也將套用相同編碼方式，並共用權重。此 CNN 的編碼器如圖 6 所示，此模型在實驗中將以 CNN+LSTM 表示。

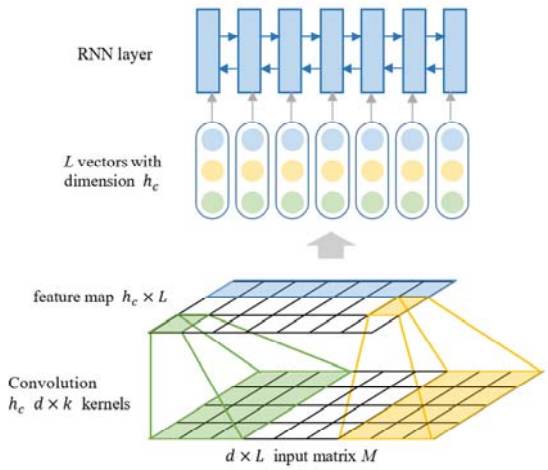

圖 6. CNN 編碼器架構圖
[Figure 6. CNN Decoder Structure]

BERT Model: 最後由於 Devlin 等人(Devlin et al., 2018)所提出的 BERT 模型在許多的下游任務獲得成功，因此我們使用已完成預訓練的 BERT 模型，在相同會話任務及回覆預測任務上進行 Fine-Tune。在輸入的部分，訊息m_1與m_2被放入相同輸入，並在當中插入適當的符號（token）。在輸出的部分，分類符號位置上的輸出將被用於最後的預測，此輸出向量z透過全連接層輸出最終的機率分數，計算方式如下列式子所示，其中$W_0 \in \mathbb{R}^{h\times h}$，$W_1 \in \mathbb{R}^{h\times 1}$：

$$P(same|m_1, m_2) = \sigma(\text{ELU}(z^T W_0 + b_0)W_1 + b_1) \quad (4)$$

我們使用交叉熵（cross-entropy）作為目標函式，並套用權重為λ的 L2 正規化，其中θ為參數集合，計算方式如下：

$$\sum_{(m_i,m_j,y)\in S}[y \cdot \log P + (1-y) \cdot \log(1-P)] + \lambda\|\theta\|^2 \quad (5)$$

5. 實驗與分析 (Experiments and Analysis)

在本章節中，我們將進行傳統深度學習模型與 BERT 模型效能的比較與分析。由於此研究中的實驗皆為二元分類，因此我們使用對相同會話（或真實回覆）的 F1-Measure 以及準確率（Accuracy）來進行評估。我們將相同會話配對及真實回覆配對的資料視為正向（Positive）資料，相異會話及非真實回覆資料視為負向（Negative）資料。

我們採用 Tensorflow (Google, 2015)做為模型建立的工具。在傳統深度學習模型部份，我們使用 GloVe (Pennington, Socher & Manning, 2014)模型，於 Common Crawl[]資料集上訓練大小寫區別的英文詞嵌入向量，維度$d = 300$，共 220 萬個單詞。雙向 LSTM 的 hidden size h 為 128，CNN 中所使用的 kernel 數量為$h_c = 128$個，其中 kernel 大小$k = 5$。傳統深度學習模型的所使用的批次量為 256、最大 epoch 為 40、初始學習率$2 \cdot 10^{-4}$。在 BERT 模型部份則使用大小寫區分的 BERT base 模型，12 層的 Transformer、hidden size 為 768、Multi-Head 數量為 12。文字最大長度L設定為 128，並截短超過此長度的文字；在 BERT 模型的輸入為兩訊息合計長度 128，並截短超出最大長度的部分；BERT 模型所使用的批次量為 32、最大 epoch 為 6、初始學習率為$2 \cdot 10^{-5}$。在所有的模型中所使用的最佳化器皆為 Adam、$\beta 1 = 0.9$、$\beta 2 = 0.999$，L2 權重衰減 0.01，並套用線性衰減於學習率，且在學習率前 30%迭代中（training step）使用 warmup，Dropout 設定為 0.1。在訓練的過程成中，相同會話配對與非相同會話配對（回覆關係配對與非回覆關係配對）的資料數量為 1:1。

在相同任務的訓練與測試資料分割方式中，我們將分別使用隨機選取分割與依照發表時間分割兩種方式進行評估；在回覆預測任務的訓練與測試資料分割方式中，我們只使用依照發表時間分割的方式進行評估，讓評估結果能更有效貼近實際應的情況。在兩種任務中，我們個別比較傳統深度學模型與 BERT base 模型，包含由 Jiang 等人所提出以 CNN 為基礎的 SHCNN 模型(Jiang *et al.*, 2018)、應用注意力機制的雙向 LSTM 以及基於 CNN 與雙向 LSTM 的 CNN+LSTM 模型。

5.1 相同會話任務 (Same Conversation Task)

表 3 為在相同會話任務中，模型在隨機選取分割與依照發表時間分割兩種方式的效能表現。隨機選取分割的測試資料，能夠評估模型對已知訊息的分類結果。若測試資料是依照訊息發表時間進行分割，則測試資料中訊息對訓練資料而言為未來訊息，這樣的評估方式較為合模型在現實世界中應用的情況。表 3 的下半部呈現所有的模型在預測未來的訊息時，效能皆大幅下降，顯示所有模型都無法預測未知的訊息。

為能更了解為何模型在進行相同會話預測表現很差的原因，以及人在處理相同任務時的表現，我們隨機從 Reddit 的測試資料中隨機挑選 500 個配對，並由 4 個人標記這份資料。結果如表 4 左側所示，我們發現人在執行相同會話任務時，表現不如預期。

標記資料的過程中我們瞭解到標記者在標記時，必須猜測配對中訊息正在討論的主題，然後判斷兩者討論的主題是否相同，而且在缺乏上下文的情況中，人們難以判斷這些訊息所討論的內容，因此我們猜測過去研究中，相同會話任務中模型效能表現不佳的原因，也可能來自於此。

表 3. 相同會話任務實驗結果(隨機分割 vs 時間分割)
[Table 3. Performance Evaluation for Same conversation Task (Splitting by Time or Randomly)]

資料集		Reddit						IRC	
		gadgets		iPhone		politic			
評估方式		F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy
隨機選取	SHCNN	0.779	0.887	0.608	0.816	0.639	0.770	0.805	0.967
	DALSTM	0.809	0.900	0.618	0.804	0.638	0.775	0.346	0.854
	CNN+LSTM	0.981	0.990	0.956	0.980	0.945	0.969	0.319	0.801
	BERT base	**0.988**	**0.994**	**0.994**	**0.994**	**0.996**	**0.970**	**0.888**	**0.985**
時間分割	SHCNN	0.253	**0.709**	0.318	**0.553**	0.411	0.553	0.039	**0.939**
	DALSTM	0.308	0.710	**0.417**	0.500	**0.441**	0.616	0.089	0.854
	CNN+LSTM	0.289	0.645	0.364	0.516	0.309	0.601	0.098	0.630
	BERT base	**0.522**	0.508	0.298	0.534	0.423	**0.667**	**0.223**	0.898

表 4. 四個人分別對相同會話任務與回覆預測任務的標記效能
[Table 4. Human Label Performance on Two Tasks]

	相同會話任務		回覆預測任務	
評估方式	F1	Accuracy	F1	Accuracy
標記者 1	0.105	0.728	0.548	0.860
標記者 2	0.352	0.763	0.800	0.930
標記者 3	0.263	0.732	0.590	0.875
標記者 4	0.638	0.796	0.703	0.865

5.2 回覆預測任務 (Reply Prediction Task)

當一個訊息為另一個訊息的回覆對象時，此兩訊息必然屬於相同會話，因此這樣的分類器仍然可以做為下游的對話解構任務。人工標記結果如表 4 右側所示，人們在預測下文是否正在回覆上文的訊息時，能夠更準確的進行預測。

在回覆預測任務的實驗中，只採用依照訊息發表時間分割訓練與測試資料的方法進行評估。表 5 為模型在相同會話任務與回覆預測任務中的效能表現，可以從中觀察到模型在回覆任務中的效能大多比相同會話好，且隨著訓練資料越多，其效能表現越高。

表 5. 相同會話任務與回覆預測任務中的模型效能表現
[Table 5. Same Conversation vs Reply Prediction]

資料集		Reddit					
		gadgets		iPhone		politic	
評估方式		F1	Accuracy	F1	Accuracy	F1	Accuracy
相同會話	SHCNN	0.253	**0.709**	0.318	**0.553**	0.411	0.553
	DALSTM	0.308	0.710	**0.417**	0.500	**0.441**	0.616
	CNN+LSTM	0.289	0.645	0.364	0.516	0.309	0.601
	BERT base	**0.522**	0.508	0.298	0.534	0.423	**0.667**
回覆預測	SHCNN	0.249	0.789	0.336	0.716	0.498	0.726
	DALSTM	0.343	0.690	0.455	0.690	0.594	0.792
	CNN+LSTM	0.260	0.646	0.413	0.688	0.561	0.782
	BERT base	**0.631**	**0.854**	**0.639**	**0.863**	**0.706**	**0.887**

5.3 切除實驗 (Ablation Test)

為了瞭解模型中局部的設計對效能影響，因此我們對 CNN+LSTM 與 LSTM 兩種模型進行了簡單的切除實驗分析。我們將以模型表現較好的環境進行分析。

由表 6 可以觀察到 CNN+LSTM 模型在 CNN 層使用 Gated Linear Unit 會比使用 ELU 會有更好的效能。而在 CNN 輸出套用了 Layer Normalization 的模型也有更好的效能。其中值得注意的是注意力機制在這個模型中對效能毫無幫助。

表 6. CNN+LSTM 模型切除實驗結果
[Table 6. Ablation Test on CNN+LSTM]

Reddit – gadgets （相同會話任務）		
評估方式	F1	Accuracy
CNN+LSTM	**0.981**	**0.990**
CNN+LSTM ELU	0.960	0.980
CNN+LSTM no norm	0.958	0.979
CNN+LSTM with attention	0.906	0.951

表 7. DALSTM 模型切除實驗結果
[Table 7. Ablation Test on DALSTM]

Reddit – politic （回覆預測任務）		
評估方式	F1	Accuracy
DALSTM	**0.594**	**0.792**
LSTM no attention	0.577	0.775

由表 7 可以得知應用了注意力機制的 LSTM 模型有更好的效能，但是對這樣的注意力專注設計僅能提升少許的效能。

5.4 客服人員聊天紀錄 (QNAP Conversation Log)

由於此研究中最初的動機源自於對客服聊天紀錄的對話解構任務，因此我們將在企業所提供的聊天紀錄中套用前述的實驗方法。

在資料準備的部分，由於缺乏大量訓練資料，只能尋求啟發式的方法進行資料標記。考量到兩人對話情境下，對話大多以一對一情況呈現，我們假設任一訊息的下個訊息即為回覆，並以此假設進行自動標記，標記數量如表八所示。由於這樣的假設與我們的實驗目的互相矛盾，我們可以得知這樣的自動標記必然有許多錯誤的資料，因此我們需要標記少量的資料來評估自動標記的品質。我們隨機抽取 60 則聊天紀錄進行人工標記後作為測試資料，標記數量如表 8 所示。透過測試資料的評估，我們觀察到在實際的兩人文字交談情境下僅有 0.801 準確率，換言之，在僅有兩人的文字交談環境中，還是有約有兩成的訊息是以交錯的方式呈現。

表 8. 聊天紀錄訓練資料與測試資料
[Table 8. QNAP Training & Testing Data]

資料集	QNAP Chat Log	
	Heuristic Labeling Training Data	Human Annotated Testing Data
會話數	1,937	60
訊息數	53,548	1,425
配對數	257,306	6,779

透過前一章的實驗經驗，我們在客服紀錄上進行回覆預測任務的訓練，以自動標記的資料作為訓練資料，並使用人工標記資料作為測試資料，實驗結果如表 9 中顯示，模型表現不如預期。

實驗的結果顯示在訓練資料品質不佳或訓練資料過少的情況下，模型無法學習到訊息間回覆的關係，且訊息在配對的過程有可能會放大錯誤標記的比例，所以在處理這樣的問題時，我們仍然需要人力對資料進行標記來產生足夠的訓練資料。

表 9. 模型於於人工標記測試資料上的結果
[Table 9. Model Performance on QNAP Human Annotated Test Data]

模型	CNN+LSTM		DALSTM		BERT		Heuristic Labeling	
評估方式	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy
效能	0.308	0.636	0.367	0.600	0.363	0.733	0.681	0.848

6. 結論 (Conclusion)

本論文提出使用回覆預測分類器取代對話解構中原有的相同會話分類器方法，使模型可以透過理解語意的方式進行對話解構的工作。相較於需要上下文支持來理解訊息內容主題的相同會話任務，回覆預測分類器可以僅透過訊息配對中的語意資訊進行分類，提高對話解構在實務上應用的可能性。

在回覆預測任務中，使用完成預訓練的 BERT 模型進行 Fine-Tuning 可以獲得良好的效能表現，即使在訓練資料較少的情況下，也能有不錯的效能。從聊天紀錄的實驗中可以得知，使用 BERT 進行 Fine-Tuning 時，下游任務所使用的標記需要有良好的標記品質，因此足夠的人工標記作為訓練資料仍然是必要的。

參考文獻 (References)

Allan, J. (2002). Introduction to Topic Detection and Tracking. In: Allan J. (eds) *Topic Detection and Tracking. The Information Retrieval Series*, vol 12. Springer, Boston, MA. doi: 10.1007/978-1-4615-0933-2_1

Aoki, P. M., Szymanski, M. H., Plurkowski, L., Thornton, J. D., Woodruff, A., & Yi, W. (2006). Where's the party in multiparty?: Analyzing the structure of small-group sociable talk. In *Proceedings of CSCW'06*, 393-402. doi: 10.1145/1180875.1180934

Dauphin, Y. N., Fan, A., Auli, M., & Grangier, D. (2016). Language modeling with gated convolutional networks. In arXiv preprint arXiv: 1612.08083.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In arXiv preprint arXiv:1810.04805.

Elsner, M. & Charniak, E. (2008). You talking to me? a corpus and algorithm for conversation disentanglement. In *Proceedings of ACL'08*, 834-842.

Elsner, M. & Charniak, E. (2010). Disentangling chat. *Computational Linguistics*, *36*(3), 389-409. doi: 10.1162/coli_a_00003

Elsner, M. & Charniak, E. (2011). Disentangling chat with local coherence models. In *Proceedings of ACL:HLT'11*, 1179-1189.

Google. (2015). TensorFlow, https://www.tensorflow.org/

Jiang, J.-Y., Chen, F., Chen, Y.-Y., & Wang, W. (2018). Learning to Disentangle Interleaved Conversational Threads with a Siamese Hierarchical Network and Similarity Ranking. In *Proceedings of NAACL '18*. doi: 10.18653/v1/N18-1164

Lowe, R., Pow, N., Serban, I., & Pineau, J. (2015). The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In arXiv preprint arXiv:1506.08909.

Mehri, S. & Carenini, G. (2017). Chat disentanglement: Identifying semantic reply relationships with random forests and recurrent neural networks. In *Proceedings of IJCNLP'17, 1,* 615-623.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of EMNLP '14*, 1532-1543. doi: 10.3115/v1/D14-1162

Shen, D., Yang, Q., Sun, J.-T., & Chen, Z. (2006). Thread detection in dynamic text message streams. In *Proceedings of SIGIR '06*, 35-42. doi: 10.1145/1148170.1148180

Wang, L. & Oard, D. W. (2009). Contextbased message expansion for disentanglement of interleaved text conversations. In *Proceedings of NAACL '09*, 200-208.