

國立雲林科技大學資訊管理系

碩士論文

Department of Information Management

National Yunlin University of Science & Technology

Master Thesis

利用 NLP 建立 LSTM 模型進行情感分析

— 以年長者對話資料為例

Use NLP to Build an LSTM Model for Sentiment
Analysis- Taking Conversation Data from the Elderly as
an Example

林嘉維

Jia – Wei Lin

指導教授：陳重臣 博士

Advisor: Jong – Chen Chen , Ph.D.

中華民國 110 年 6 月

June 2021

國立雲林科技大學

碩士班學位論文考試委員會審定書

本論文係 林嘉維 君在本校 資訊管理系 所提論文

「利用 NLP 建立 LSTM 模型進行情感分析－以年長者對話資料為例」

合於碩士資格水準，業經本委員會評審認可，特此證明。

口 試 委 員 ： 陳重臣

陳昭宏

簡守維

陳重臣

指 導 教 授 ： 陳重臣

陳重臣

所

長：

莊煥榮

中 華 民 國

110 年

07 月 09

日

摘要

近年來，人口老化的趨勢越發嚴重，伴隨而來的就產生了許多問題及需求。而如何提升銀髮族的生活品質，是被一直關注的議題。隨著醫療技術的進步、資訊科技、人工智慧（AI）的快速發展，發展出許多新型態的醫療照護模式。針對「老年後沒人陪伴」這項議題，許多人提出建立對話機器人來跟老人進行交談，情感分析是一項能夠提升對話機器人在交談時的情感跟溫度的課題。

本研究想要探討的議題，就是想利用自然語言處理(NLP)建立一個以能夠將老人對話時說出來的對話內容進行情感判斷的機制，以便進行回覆的對話內容的篩選，讓對話機器人能夠回覆的更加具有情感以及溫度。

關鍵字：自然語言處理、對話機器人、NLP、情感分析



ABSTRACT

In recent years , the trend of population aging has become more and more serious , and many problems and demands have been generated along with it. How to improve the quality of life of the silver-haired people has always been a topic of concern. With the advancement of medical technology , the rapid development of information technology , and artificial intelligence (AI) , many new types of medical care models have been developed. In response to the issue of 'no one to accompany after old age' , many people have proposed to establish chatbot to talk to the elderly. Sentiment analysis is a topic that can improve the emotion and temperature of chatbot during conversation.

The topic that this research wants to explore is to use natural language processing (NLP) to establish a mechanism that can judge the content of the conversation spoken by the elderly during the conversation , so as to screen the content of the reply , so that the chatbot can reply. It's more emotional and warm.

Keywords: Natural Language Processing , Chatbot , NLP , Sentiment Analysis

目錄

摘要	i
ABSTRACT	ii
目錄	iii
表目錄	iv
圖目錄	v
第一章 緒論	1
1.1、研究背景	1
1.2、研究動機	1
1.3、研究目的	3
1.4、研究架構流程	4
第二章 文獻探討	5
2.1、自然語言處理	5
2.2、聊天機器人	5
2.3、情感分析	7
第三章 研究方法	8
3.1 研究設計	8
3.2 研究開發架構	9
3.3 研究工具	9
3.3.1、資料收集/處理	9
3.3.2、建立模型	12
第四章 資料分析	15
4.1、資料收集	15
4.2、資料處理	16
4.3、模型建立	19
4.4、輸出結果	22
第五章 結論	23
5.1 結論	23
5.2 建議	24
5.3 未來展望	24
參考文獻	25
中文文獻	25
英文文獻	28

表目錄

表 1、RNN 模式及應用	13
表 2、LSTM 各函數運算表示	14
表 3、人工標註情感分類標籤使用標準表	16
表 4、人工標註情感分類標籤後資料分類分佈表	16
表 5、LSTM 分類模型結果統計表	22



圖目錄

圖 1. 以人工智慧結合新科技的醫療保健概念	1
圖 2. 高齡化時程推估圖	2
圖 3. 三階段年齡人口變動趨勢推估圖	2
圖 4. 扶養比趨勢推估圖	2
圖 5. 研究架構流程圖	4
圖 6. 流水線式的任務導向型對話機器人	6
圖 7. 端到端架構的開放域對話機器人	6
圖 8. 任務型對話機器人架構圖	8
圖 9. 研究開發架構圖	9
圖 10. Word2Vec 兩種模型圖	11
圖 11. Doc2Vec 兩種模型	11
圖 12. 基本遞迴神經網路結構圖	12
圖 13. RNN 模型圖	12
圖 14. 標準 LSTM 結構圖	13
圖 15. 研究開發流程摘要圖	15
圖 16. 利用資料增強進行反向翻譯產生之資料圖	15
圖 17. 人工標註情感分類標籤檔案圖	16
圖 18. 進行 Jieba 後資料檔	17
圖 19. 使用 Word2Vec 訓練模型	17
圖 20. 使用 Doc2Vec 訓練模型	18
圖 21. 使用 Doc2Vec 模型訓練後取得的文本向量矩陣	18
圖 22. 使用 Tokenizer 模型進行訓練	18
圖 23. 使用 Tokenizer 模型取得文本向量矩陣過程	19
圖 24. 資料集切割程式碼	19
圖 25. 切割後訓練資料集	20
圖 26. 切割後測試資料集	20
圖 27. 建立 LSTM 分類模型	21
圖 28. LSTM 分類模型結構	21

第一章 緒論

1.1、研究背景

近年來，人口老化的趨勢越發嚴重，伴隨而來的就產生了許多問題及需求。而如何提升銀髮族的生活品質，是被一直關注的議題。隨著醫療技術的進步、資訊科技、人工智慧（AI）的快速發展，發展出許多新型態的醫療照護模式。

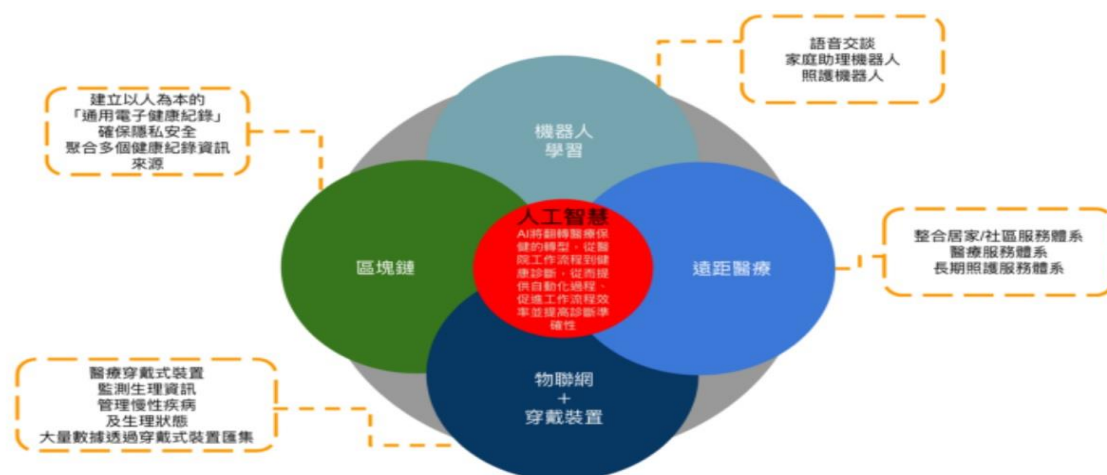


圖 1. 以人工智慧結合新科技的醫療保健概念
資料來源：陳鴻麒(2019)

1.2、研究動機

人口老化是全球都要面臨的問題。因為醫療技術的進步，所以能夠減緩死亡率以及延長預期餘命(life expectancy)，再加上生育率下降相關問題，導致了人口逐漸步入高齡化的結果。以台灣之人口老化的情況來說，1990 年的時候，65 歲以上的人口佔總人口的比例約為 6.2%，(即每 16 人中就有一位高齡人口)。而到了 2019 年，65 歲以上人口比例提升到了 15.28%（即每 6.5 人中就有一位高齡人口）。高齡人口比例的增加對於健康照護系統也帶來了極大的壓力。而高齡人口需要提供生活照護的需求也將會愈來愈大。(范光中&許永河，2010)

根據國家發展委員會之人口推估，如圖 2 所示，台灣已經在 1993 年成為高齡化社會(65 歲以上人口佔總人口比例達到 7%)，在 2018 年時，變成高齡社會(65 歲以上人口佔總人口比例達到 14%)，推估將會在 2025 年進入超高齡社會(65 歲以上人口佔總人口比例達到 20%)。

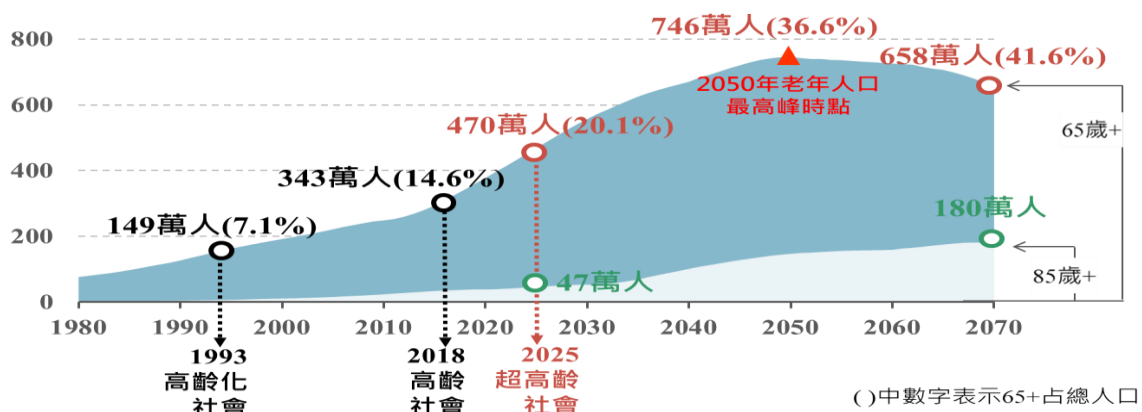


圖 2. 高齡化時程推估圖

資料來源：國家發展委員會(2020)

根據國家發展委員會之人口推估，如圖 3 所示，老年人口在 2017 年開始就超過幼年人口，預估在 2029 年時，老年人口將會達到幼年人口的 2 倍。預估 2070 年後，15-64 歲青壯年人口與 65 歲以上老年人口將非常相近。

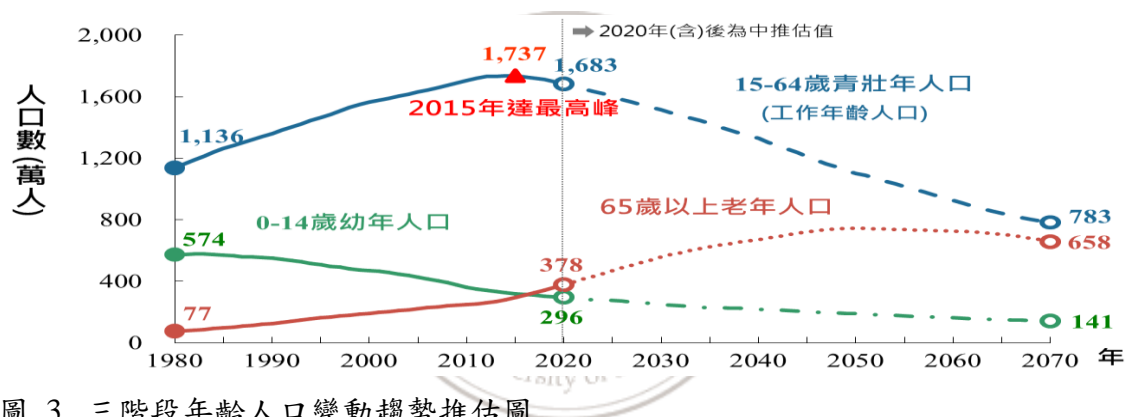


圖 3. 三階段年齡人口變動趨勢推估圖

資料來源：國家發展委員會(2020)

根據國家發展委員會之人口推估來看，扶養比將會越來越高，未來將會由 2020 年的 40.1% 突破至 2070 年的 100% 以上，未來青壯年人口的負擔將會越來越重。

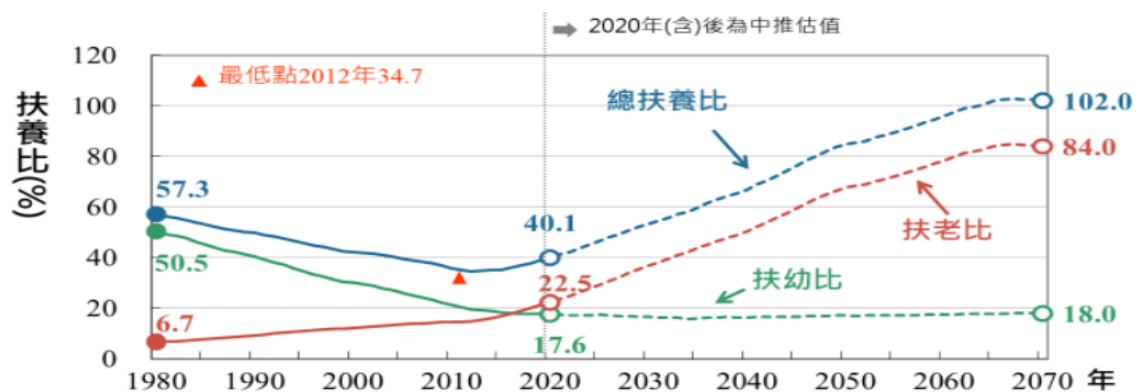


圖 4. 扶養比趨勢推估圖

資料來源：國家發展委員會(2020)

因為高齡化再加上少子化這些問題，政府提出了許多福利政策，像是在 2007 年，政府推出長照 1.0 的政策，不過由於長照 1.0 的預算太少、限制很多等問題，讓使用率非常低，導致政策的推動很緩慢。後來在 2017 年時，推出了長照 2.0 的政策，增加了很多的服務項目、服務範圍及補助，提供更多服務對象進行照護。(衛生福利部，2017)

人口老化的問題是全球性的，中國民政部在 2017 年時，提出智慧健康養老產業發展行動的計劃(2017-2020 年)，主旨是利用物聯網、雲計算、大數據、智能硬件等新一代的信息技術來讓智慧健康養老產業發展得更加快速。徐婷婷&張赫(2017)也在「養老邁入智慧時代，智能機器人能陪老人聊天、助休閒」中提到：「在中國心理衛生協會婦女健康與發展專業委員會常務副主任兼秘書長楊波看來，現在大多數老年人最怕的不是物質的缺乏，而是無人陪伴。」

1.3、研究目的

隨著科技的進步，衍生出許多新型態的醫療照護議題，目前常見的應用像是：智能健康病歷管理、醫療圖像處理(AI 輔助診斷)、早覺醫療(AI 輔助診斷)、醫藥研發...，而其中關於「老年後無人陪伴」這部分，更加是被重視的議題。現今因為人工智慧(AI)的高速發展，使得許多利用自然語言處理 NLP 建立出來的對話機器人面世，而對於如何讓對話機器人能夠表達出跟我們一般人聊天時所說的話，就需要考量到大部分一般人在聊天時，對於要說的話，會讓大腦進行篩選及判斷後才會講出來。而要讓電腦能夠像人一樣能夠流暢的與人進行對話，對話內容也同樣必須要經過邏輯的篩選以及判斷後，才能取得符合邏輯的回覆出來。情感分析是其中一項需要考量的因素，而要分析一個人的情感，需要考量到好幾個方面，才能進行判斷這個人的情感(例如：語音辨識(語速、聲音頻率、聲音大小)、臉部表情、說話內容...等)。

本研究想要探討的議題，就是想利用自然語言處理(NLP)建立一個以能夠將老人對話時說出來的對話內容進行情感判斷的機制，以便進行回覆的對話內容的篩選，讓對話機器人能夠回覆的更加具有情感以及溫度。

1.4、研究架構流程

本研究的研究架構流程如圖 5 所示，各階段詳細說明如圖：

1. 第一章 緒論

本章主要為本研究之研究背景、研究動機、研究目的。

2. 第二章 文獻探討

本章將整理本研究主要相關主題的文獻進行探討，主要分為三個項目，包括自然語言處理(Natural Language Processing，縮寫 NLP)、聊天機器人(ChatBot)以及情感分析(Sentiment Analysis)。

3. 第三章 研究方法

本章節主要描述本研究的資料收集、資料處理、開發流程以及研究工具介紹。

4. 第四章 資料分析

本章節描述進行資料分析的各個步驟與模型建置。

5. 第五章 結論

本章節主要為以本研究之分析結果進行結論以及未來展望。

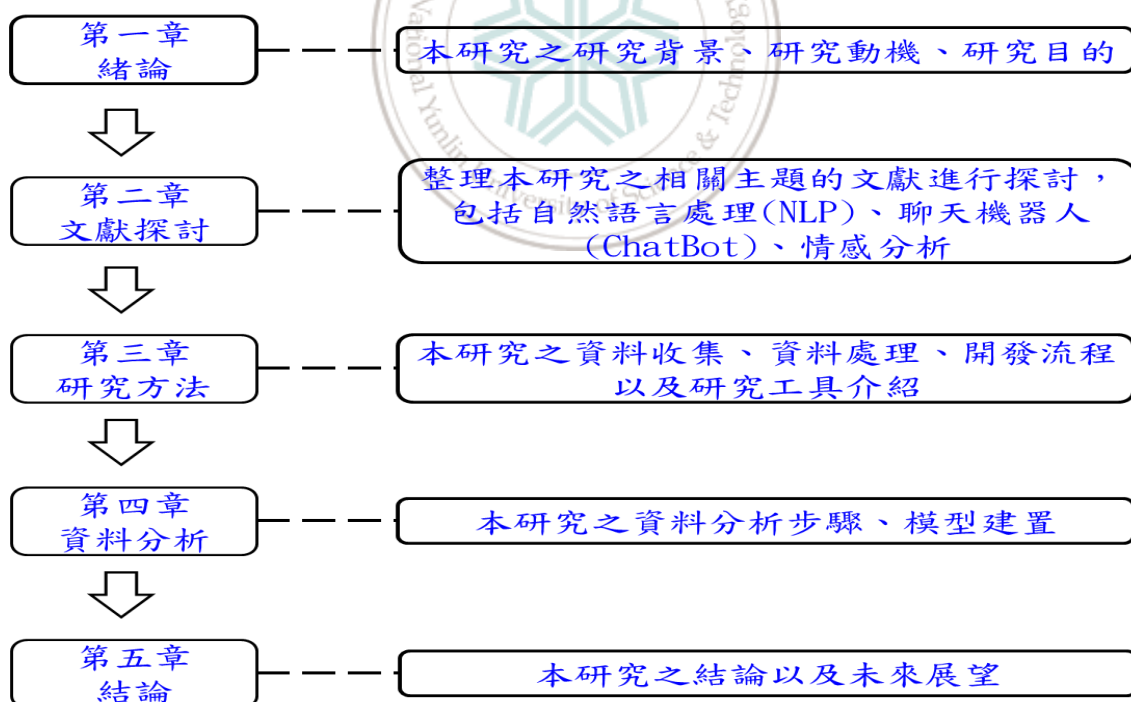


圖 5. 研究架構流程圖

資料來源：本研究整理

第二章 文獻探討

本研究是以自然語言處理技術建立出能夠加強應用於老人對話的聊天機器人為主的情感分析機制，因此將分為自然語言處理、聊天機器人以及情感分析三個小節來進行文獻探討。

2.1、自然語言處理

自然語言處理 (Natural Language Processing，簡稱 NLP)，是一門結合了人工智慧和語言學領域的學科。想要讓電腦能夠理解自然語言，分為自然語言理解 (Natural Language Understanding，簡稱 NLU) 以及自然語言生成 (Natural Language Generation，簡稱 NLG) 兩個部分。其中自然語言理解是指將自然語言的內容轉化成電腦能夠理解的形式；自然語言生成是指電腦生成數據並轉化為人類可以理解的自然語言形式。(每周科技詞，2019)

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、心理學和語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)

自然語言處理(NLP)是指：電腦能夠理解並且解釋人類書面以及口語的能力，主要是為了讓電腦能夠跟人類一樣，更加智能化的理解語言並且能夠使用語言來做表達，來彌補人類交流(自然語言)跟計算機理解(機器語言)之間的差異。目前已知 NLP 有了廣泛的應用領域，像是在信息提取、機器翻譯、文本生成、知識圖譜、智能問答、情感分析、對話系統等領域。(黃民烈&張岳，2021)

2.2、聊天機器人

顏理謙(2016)提出聊天機器人是一種經由文字、聲音、圖像或使用者對話介面來進行互動的電腦程式。只要輸入一張圖片或是關鍵字並設立條件，就會將你需要的服務或商品進行推薦。

聊天機器人的研究源自於電腦科學之父-圖靈(Alan Mathison Turing)在 1950 年發表的文章，他提出了一個提問：「機器能思考嗎？」，並開發了一項可以測試機器是否具有思惟的試驗-圖靈測試(Turing Test)進行佐證，測試標準是：一個人在未被告知的情形下與被測試者(AI 機器)進行對話，當測試人認為被測試者的應對答覆超過 30% 的回答他無法判別是機器還是人類所回答的，則通過圖靈測試，他認為通過圖靈測試的機器便擁有智慧。(李魚愷，2016)

最早的聊天機器人是於 1966 年，由麻省理工學院(MIT)的 Joseph Weizenbaum 發表的 Eliza 程式，劉秉鈞(2016)指出 Eliza 程式設計概念是以模仿為主要核心，

第一代設計以醫生為模仿對象，撰寫出一套虛擬看診系統電腦，角色設定為心理醫生，於臨床治療中透過提問讓病人回答，並將結果進行簡單的樣式比對，產生出電腦與人的交談感覺。

黃民烈(2021)提到說當前的對話系統主要有兩種類型，一種是流水線式的任務導向型，另一種是端到端架構的開放域。並指出當前的對話系統最大的問題有三種，分別是語義性、一致性以及交互性。其中語義性問題是希望對話系統能夠理解內容(content)、文本(context)和場景(scene)；一致性問題是希望對話系統能夠產生符合身分還有個性一致的對話；交互性問題是希望對話系統能夠跟用戶進行情感、情緒上的交流，得以實現交互性很強的對話。想要實現下一代對話系統需要具備並且結合「有知識，言之有物」、「有個性，能夠實現擬人化」、「在情感上有情感、有溫度，能夠做一些精細的情感類任務」這三項特點。

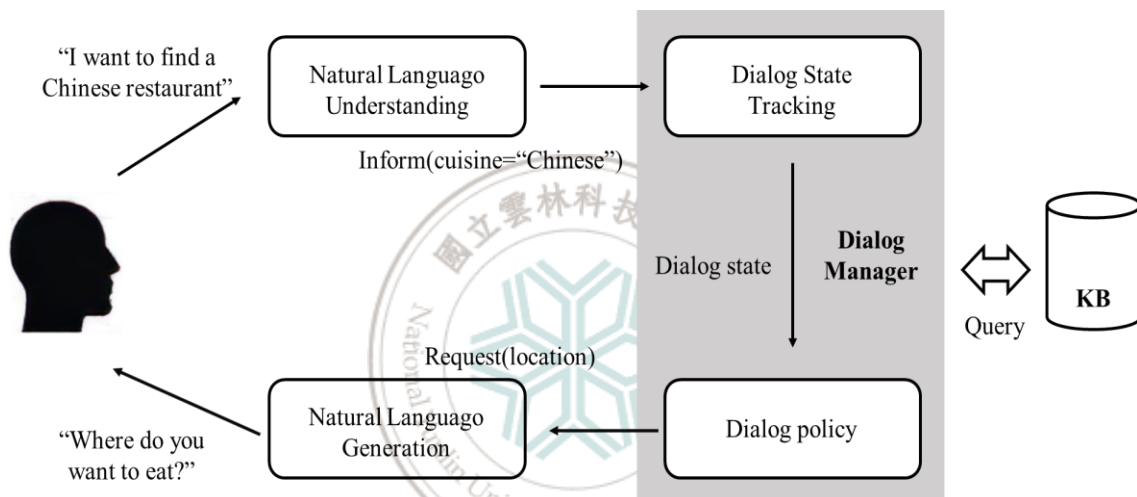


圖 6. 流水線式的任務導向型對話機器人

資料來源：黃民烈(2021)

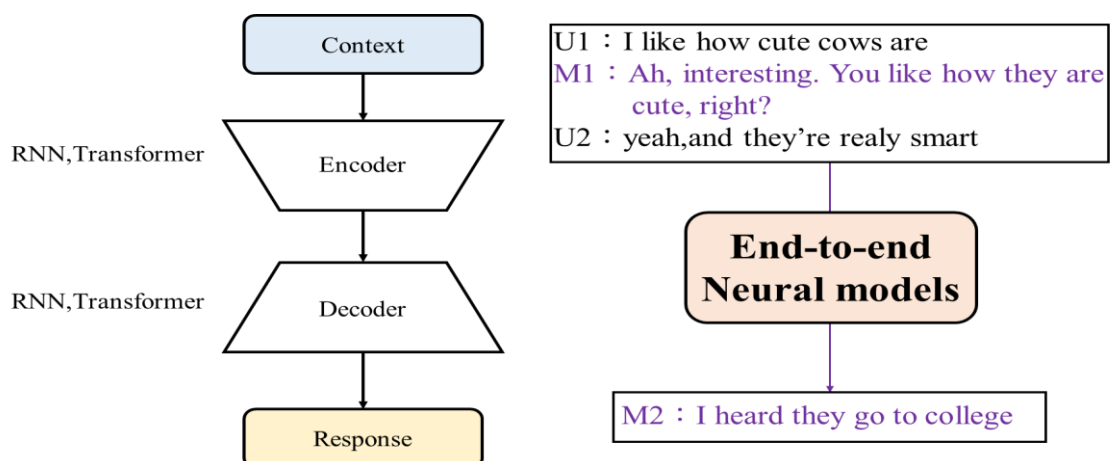


圖 7. 端到端架構的開放域對話機器人

資料來源：黃民烈(2021)

2.3、情感分析

情感依照價值的多寡，可以分成正向情感還有負向情感兩個部份，正向情感是指：人因為正向價值的增加或負向價值的減少而產生的情感，像是：愉快、信任、感激、慶幸等。負向情感是指：人因為正向價值的減少或負向價值的增加而產生的情感，像是：痛苦、鄙視、仇恨、嫉妒等。(itread01, 2019)

情感分析這一個詞彙最早是由 Das & Chen(2001)針對股票市場留言版上面的文本訊息，將其中的正面(positive)及負面(negative)的意見進行情感定義。

Cambria et al.(2013)指出情感分析(Sentiment Analysis)是用電腦運算方式來衡量文本的意見傾向、情緒反應等主觀成分的一種方法。

史丹佛大學的 Dan & Chirs(2012)提出情感分析(Sentiment Analysis)，又稱為傾向性分析、意見抽取(Opinion extraction)、意見挖掘(Opinion mining)、情感挖掘(Sentiment mining)、主觀分析(Subjectivity Analysis)，是對具有情感色彩的主觀性文字進行分析、處理、歸納以及推理的過程，像是從評論文字中，來分析出使用者對”數位相機”的變焦、價格、大小、重量、閃光、易用性等屬性的情感傾向。

陶冶中&簡睿志(2016)將情感分析總結為：包含觀點挖掘、評論挖掘、態度分析、意見挖掘、傾向性分析等一系列針對文本，所表達出的人所具有的潛在情感分析過程。並提出通常包含三個面向：1. 文本的主客觀分類：分出文本內容是主觀的評論還是客觀的陳述。2. 文本的極性分類(Polarity Classification)：判斷文本內容是正面的還是負面的分類。3. 文本情感傾向的強弱程度判斷：判斷文本情感是屬於強烈反對、反對、中立、支持、強烈支持哪一個等級，又稱為等級推理(Rating Inference)。

哈爾濱工業大學的劉挺(2015)提出：未來機器人肯定會擁有情感，情感是機器人擁有高級智能的一種體現。在自然語言處理領域，文本情感分析將是攻克機器人理解文字語言情感的一項重要技術，可以幫助機器人理解人類語言的情感，如識別出是喜悅、悲傷、憤怒等情緒，以便更好的進行語義理解，做出合適的應答。

第三章 研究方法

本研究主要為針對老人進行訪談的對話資料進行研究，資料收集部分為：針對雲林地區若干老人進行訪問，訪問時進行錄音，並轉換成逐字稿，取得資料量為 224 筆，因資料量偏少，所以利用資料增強-反向翻譯方法，將資料擴增至 896 筆，然後針對資料每句話來逐筆進行語句通順的處理，並再以人工方式來標註情感的分類標籤。資料處理部份為：利用 Jieba 對每句話去除停用字(Stop Word)後進行斷詞，再利用 Word2Vec 結合 Doc2Vec 以及 Tokenize 兩種方法取得每句話的文本向量。模型建立部分：將取得的資料集拆分為不同比例的訓練資料以及測試資料，並建立 RNN 中的 LSTM 分類模型，進行模型的訓練以及測試，取得分類準確率的輸出結果。

3.1 研究設計

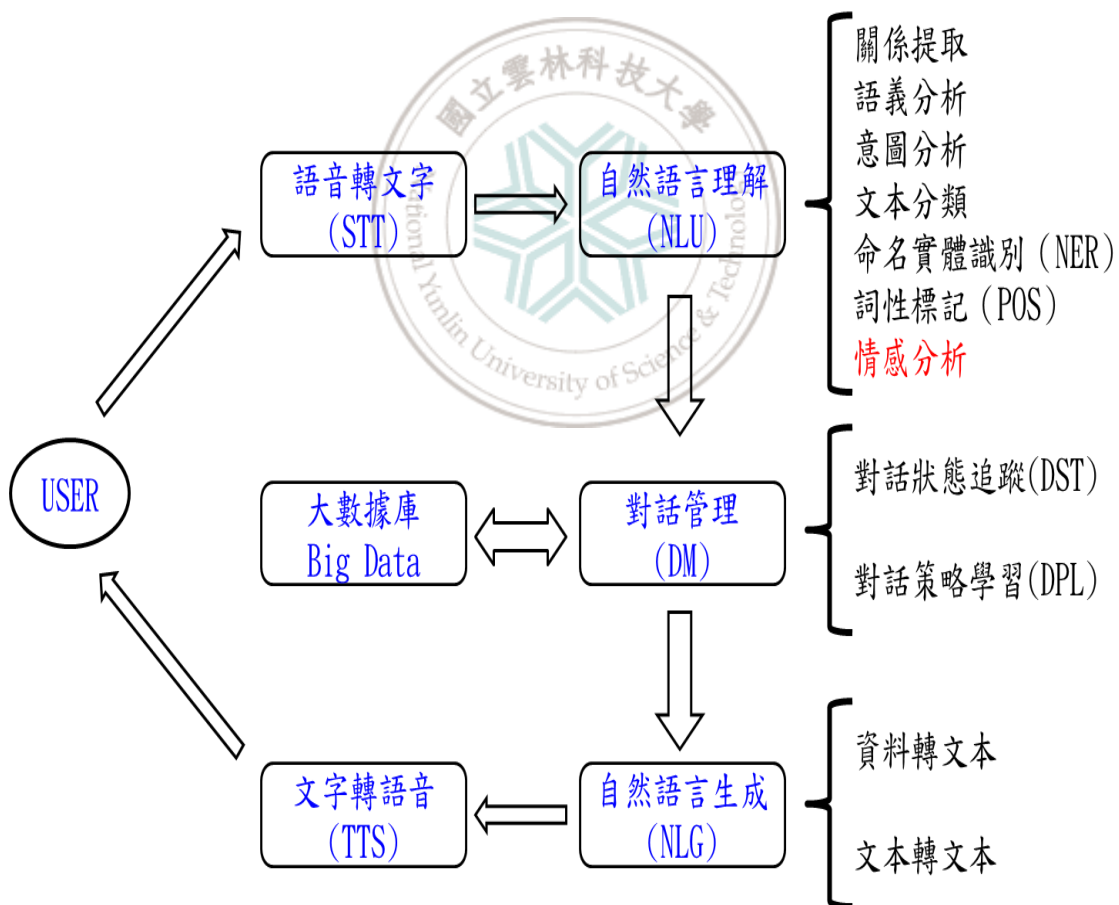


圖 8. 任務型對話機器人架構圖

資料來源：本研究整理

3.2 研究開發架構

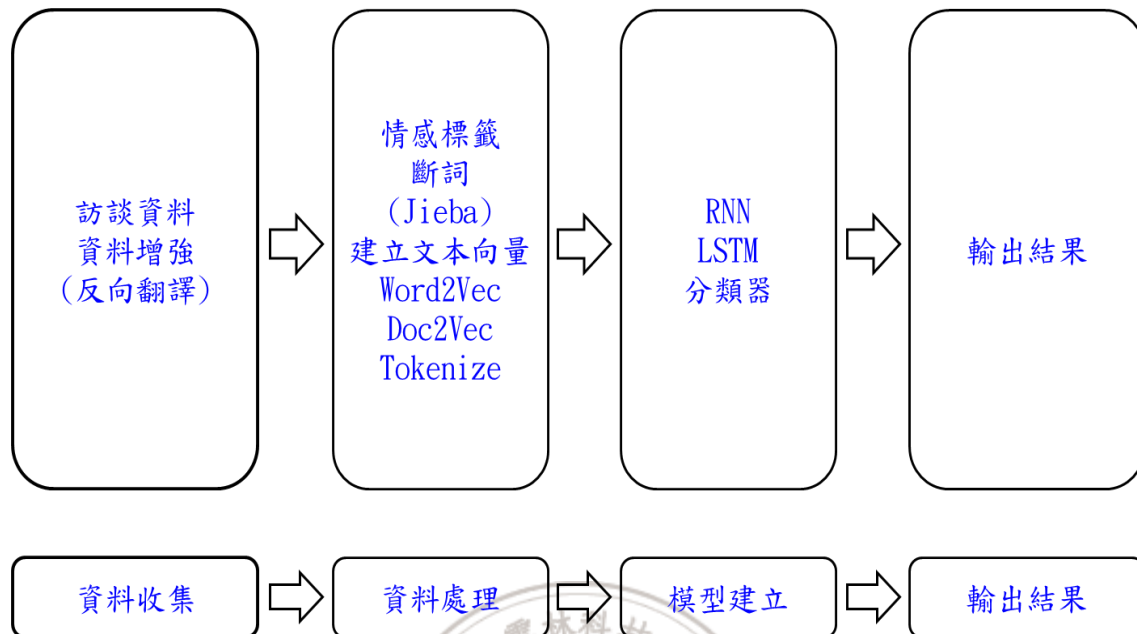


圖 9. 研究開發架構圖
資料來源：本研究整理

3.3 研究工具

使用到的研究工具主要為資料收集/處理及建立模型時所會使用到的技術及模型。針對以上使用到的技術及模型分為兩個小節進行技術介紹。

第一節將介紹進行資料收集及資料處理時，所用到的技術介紹，第二節將介紹本次研究使用的模型及技術原理。

3.3.1、資料收集/處理

1. Python

在維基百科中提到 Python 是一種被廣泛使用的直譯式、高階程式、通用型程式語言。Python 支援多種程式範式，包括物件導向、結構化、指令式、函數式和反射式程式。擁有動態型別系統以及垃圾回收功能，能夠自動管理記憶體的使用，並且擁有一個巨大而廣泛的標準庫。

2. 自然語言處理-資料增強(Data Augmentation)

Mars.Su(2020)指出當資料量不足時，如果還是要訓練機器學習模型，產生的效果會非常不好。所以後來有人提出了資料增強(Data Augmentation)的概念，就是為了應用於資料不足時，可以使用一些方法來將資料做增量的動作，進而能夠進一步的應用於訓練模型。

常見的中文 NLP Data Augmentation 方法大致上有：

1. 相似詞替換：利用相似的詞進行替換
2. Back Translation(反向翻譯)：利用翻譯軟體轉成外語在轉回來原本語系
3. Random Noise Injection：隨機新增、刪除、修改資料，當成新資料
4. Syntax-tree(語法樹)：假如原本語法是主動的，就轉成被動的語法
5. Pretrained model MLM(預訓練模型)：利用預先訓練好的模型

3. 中文斷詞-jieba

Jieba 是一個開源的中文斷詞函式庫，實作於多種程式語言，主要算法是使用 trie-tree 資料結構來進行字典配對，並將配對中剩下的單字以 HMM 模型 (Hidden Markov Model) 來組合產生出新詞，此外，Jieba 斷詞也提供使用者自定義詞典，可以提升斷詞效果。

4. 詞向量- Word2vec

Mikolov et al.(2013)提出 Word2vec，Word2vec 是很多用來產生詞向量的相關模型的集合。這些模型通常為淺層雙層的神經網路，用來訓練以重新建構語言學的詞文本。網路通常以詞來表現，並且需猜測相鄰位置的輸入詞，在 Word2vec 中詞袋模型假設下，詞的順序是不重要的。訓練完成之後，Word2vec 模型可以把每個詞映射到一個向量，來表示詞與詞之間的關係。該向量為神經網路的隱藏層。Word2vec 利用 Skip-grams 或連續詞袋 (CBOW) 兩種方法來建立神經詞嵌入。

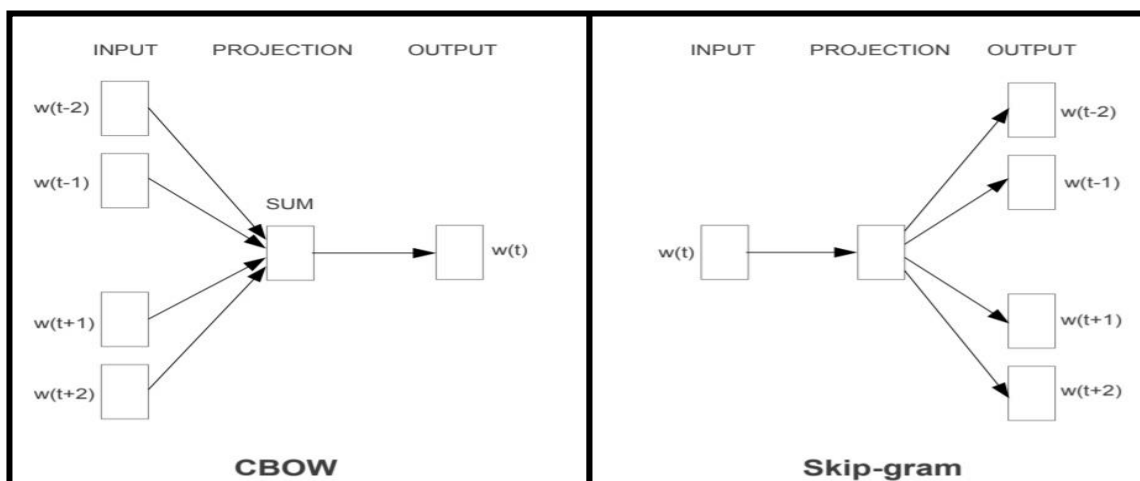


圖 10. Word2Vec 兩種模型圖

CBOW 根據上下文預測當前詞，Skip-gram 根據當前詞預測周圍的詞

資料來源：Mikolov & Chen & Corrado & Dean(2013)

5. 文本向量- Doc2Vec

Quoc & Mikolov(2014)提出 Doc2Vec，提出的主因是 Word2Vec 沒有考慮到單詞之間的順序關係。Doc2vec 是一種非監督式的算法，能將很長的一個文本(Ex：句子、段落或文檔)進行學習來取得固定維度的特徵向量表達，是 Word2Vec 的延伸，可以接受不同長度的句子來當作輸入進行訓練樣本。有兩種算法，一種是分佈記憶的段落向量(Distributed Memory Model of Paragraph Vectors，PV-DM)，像是 Word2Vec 的 CBOW 模型搭配段落向量進行預測，一種是分佈詞袋的段落向量(Distributed Bag of Words version of Paragraph Vector，PV-DBOW)像是 Word2Vec 的 Skip-gram 模型，差異在只使用段落向量進行預測。

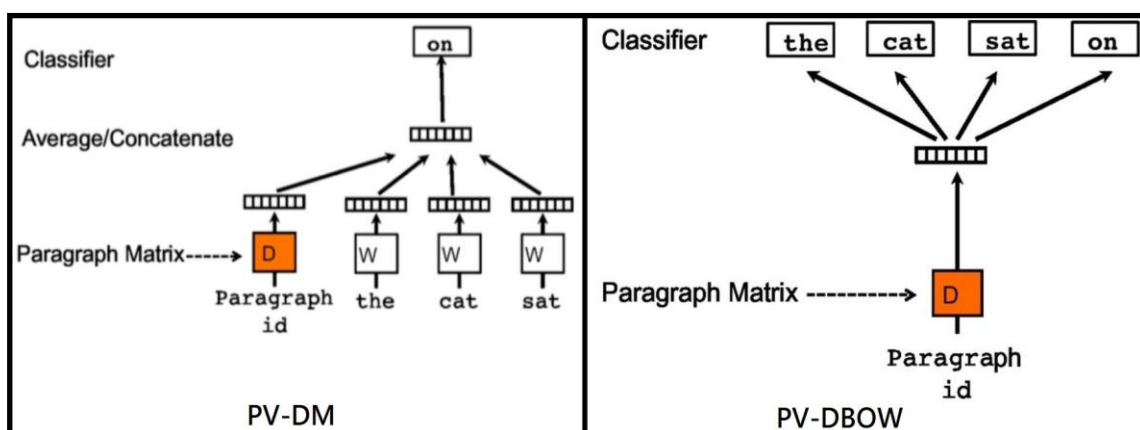


圖 11. Doc2Vec 兩種模型

PV-DM 模型及 PV-DBOW 模型

資料來源：Quoc & Mikolov(2014)

6. 文本向量-Tokenizer

Tokenizer 是一個用於向量化文本，或是將文本轉換為序列的類。是用來文本預處理的第一步：分詞。簡單來說，計算機在處理語言文字時，是無法理解文字的含義，通常會把一個詞(中文單個字或者詞組認為是一個詞)轉化為一個正整數，於是一個文本就變成了一個序列。

3.3.2、建立模型

1. 遞迴神經網路((Recurrent Neural Network , RNN)

遞迴神經網路(RNN)是神經網路(NN)的一種變形，主要特點是它會考慮過去的訊息，是一種有時間序列概念的神經網路，主要用來處理有關序列的問題 RNN 的每個神經元(不同時間點) 都能有獨立的輸入，經過運算後的輸出會當成下個時間點的輸入，是由不同時間點的神經元串連而成的。其中每個不同時間點的神經元都稱為一個 RNN Cell。

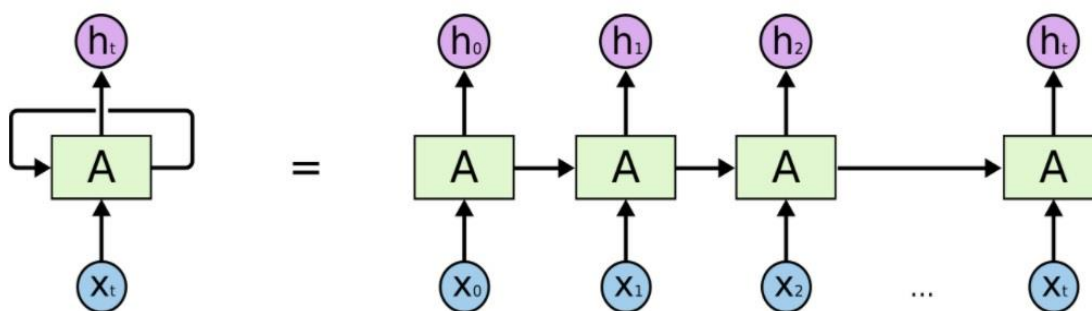


圖 12. 基本遞迴神經網路結構圖

資料來源：Olah(2015)

RNN 的模型可以依據輸入及輸出的數量，區分為五種模式，如下圖：

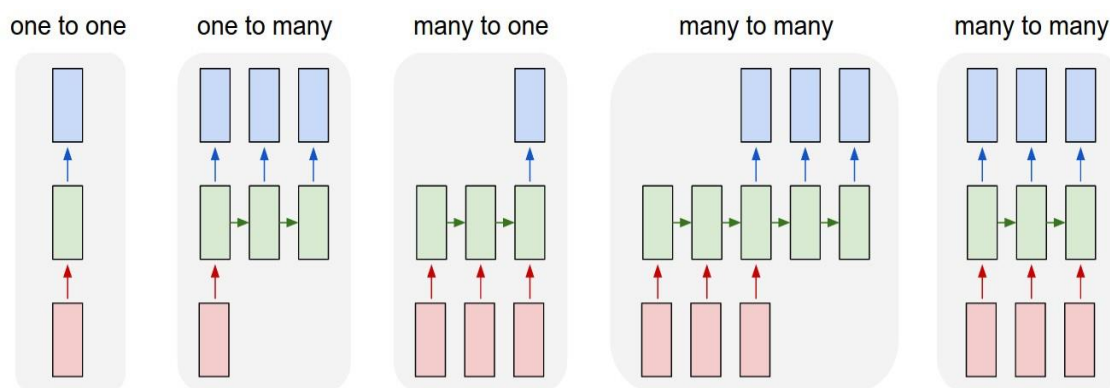


圖 13. RNN 模型圖

資料來源：Andrej(2015)

RNN 可以根據要輸入和輸出的數量，可以有很多種的應用及模式：

表 1、RNN 模式及應用

資料來源：mikechenx(2017)

模式	Input/output	應用舉例
一對一	固定長度的輸入 固定長度的輸出	圖像分類
一對多	單一輸入 多個輸出	影像標題(Image Captioning)、 文字摘要(Text Summarization)、生成文章
多對一	多個輸入 單一輸出	情感分析(Sentiment Analysis)
多對多	多個輸入 多個輸出	語言翻譯(Machine Translation)、對話機器人(Chatbot)
多對多	同步的多個輸入 同步的多個輸出	視訊分類(Video Classification)、影片字幕(Video Caption)、Slot Filling

2. 長短期記憶模型(Long Short Term Memory, LSTM)

遞迴神經網路(RNN)能處理時間序列的資料，但是因為很容易變成很深的神經網路，隨著神經網路層數的增多，會產生 Exploding Gradient(梯度爆炸)及 Vanishing Gradient(梯度消失)的問題，為了解決這些問題，Hochreiter & Schmidhuber(1997)提出了 LSTM(RNN 的變形)，他們在 RNN Cell 中導入了閥門的概念進行改進，能夠對 RNN Cell 的訊息進行流量控制。一開始的 LSTM 只有輸入門以及輸出門兩個閥門，由 Gers & Schmidhuber(2000)導入了遺忘門這項概念，才會形成目前常用標準的 LSTM。(Yu et al., 2019)

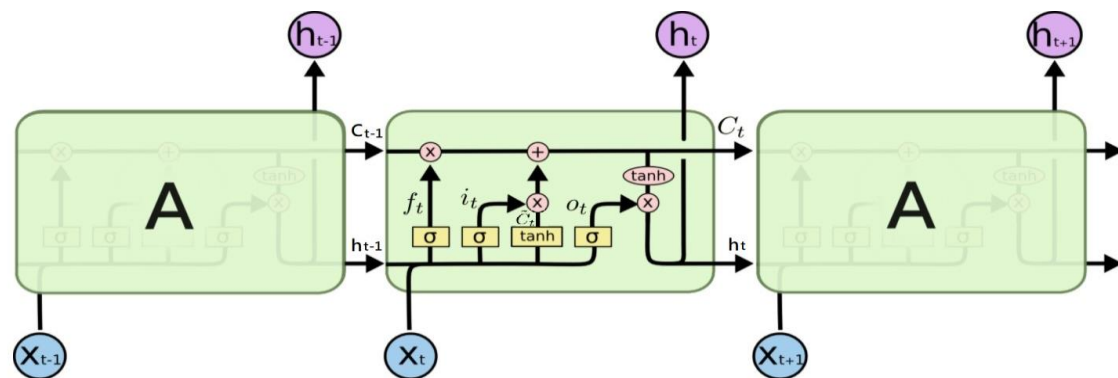


圖 14. 標準 LSTM 結構圖

資料來源：Olah(2015)

LSTM 主要由三個閘門(Gate) 及兩個 State 組成，各函數做的運算公式如下表：

表 2、 LSTM 各函數運算表示

資料來源：Olah(2015)

函數名稱	函式運算公式
Memory Cell state	$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$ $C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$
Hidden state	$h_t = o_t * \tanh(C_t)$
Forget Gate	$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$
Input Gate	$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$
Output Gate	$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$

LSTM 三個閘門的功能分別為：(蕭伯瑋，2018)

- 遺忘閘(Forget Gate)：控制多少比例的資訊要保留到下一個步驟的 memory cell。
- 輸入閘(Input Gate)：控制模型新讀入的資訊，有多少比例被寫入 memory cell 中。新讀入的資訊由這次運算被讀入的值 X_t 與上次運算的 hidden state h_{t-1} 計算而得。
- 輸出閘(Output Gate)：控制要從現在的 memory cell C_t 讀取多少比例，來計算 hidden state h_t 。

LSTM 主要是利用三個閘門(Gate)來控制這次進行運算的資料(X_t)跟上一次運算中輸出的 Hidden State 的輸出(h_t)的資料流量並決定下一次運算要使用多少，再來由 Cell State 對記憶的特徵進行調整，在來由 Hidden State 輸出這次運算出來的資料特徵，經由兩個 State 傳送每個運算之間的特徵輸出。

第四章 資料分析

本章節主要是講述研究開發流程，將會分成資料收集、資料處理、模型建立、輸出結果四個小節來進行，主要操作重點如圖 15。



圖 15. 研究開發流程摘要圖

資料來源：本研究整理

4.1、資料收集

資料收集主要利用以下兩種方法取得：

1. 老人訪談之逐字稿：224 筆
2. 利用反向翻譯的方法進行資料增強至：896 筆

review(原始)	中轉日再轉中	中轉英再轉中	中轉法再轉中
在我的一生中，我並不覺得那個階段特別重要。年輕的時候要努力掙錢養家糊口，老了有老年的聯苦。現在非常感謝樓主免費給我一套房子和政府的幫助。不然就想像小時候一樣，去當乞丐討飯了！現在，政府每個月補助6000元，又幫我安裝緊急電話，也不時有志願者來探望我，我很感激。我害怕政府沒錢給我，將這補助的6000元給取消掉。拜託你一定要把我這一句話轉告訴政府知道。	在我的一生中，我並不覺得那個階段特別引人注意。我年輕的時候，賺錢和吃老人，都很難。現在非常感謝樓主。免費給我一套房子，幫助政府。否則，像你小時候那樣乞討食物一定是在乞討！我現在很感謝政府每個月幫我拿6000元，設置緊急電話，志願者不時來。我最怕政府沒錢，給你6000塊錢。請取消，我想念你！請務必包括我的文本，以便政府可以知道。	在我的一生中，我並不覺得那個階段特別突出。年輕的時候要努力掙錢養家糊口，老了吃力不討好。現在非常感謝房東免費給我一套房子和政府的幫助。不然就想像小時候一樣當乞丐去討飯了！現在每個月政府補助6000元，安裝緊急電話，不時有志願者來探望。我很感激。我最怕政府沒錢，就給了6000元。取消，請想念你！請務必包括我的判決，以便政府知道。	在我的一生中，我認為這一步並不是特別重要。年輕的時候，為了生活，要拼命掙錢，老的也難吃。現在非常感謝樓主免費給我一套房子和政府的幫助。不然就想像小時候一樣當乞丐去討飯了！現在，政府每個月都幫我6000元，設置了緊急電話，不時有志願者來探望我。我很感激。我害怕政府沒錢，所以捐了6000元。取消，請想念你！請務必包括我的判決，以便政府知道。
我讓政府養，這間房子是我姊姊給我的，我的兩個女兒都在美國，她們自己都有小孩在養，自己用都不夠了，那有多的錢寄回來給我，所以我只能靠政府，不夠花也沒辦法，有多少就用多少，日子可以過就好。	我支持政府。我姐姐為我住在這所房子里。我的兩個女兒都在美國。他們必須撫養自己的孩子，這還不夠。如果你有錢，把它們寄回去。一世。我依靠政府。如果這還不夠，就沒有辦法花你擁有的那麼多，只要你能過上自己的生活。	我支持政府。我的大姐為我住在這所房子里。我的兩個女兒都在美國。他們必須撫養自己的孩子，他們需要的還不夠。如果他們有錢，請把他們還給我。我依靠政府。不夠用也沒辦法，有的就用，只要能過好自己的生活就行。	我支持政府。我姐姐為我住在這所房子里。我的兩個女兒在美國。他們必須撫養自己的孩子，而他們不需要。如果他們有任何錢，請把他們還給我。我依靠政府。不夠用也沒辦法，只要你能過上自己的生活，就盡量多用。
能夠有公教人員18%的優惠儲蓄，領月退休，感覺上好像有取之不盡的錢，用也用不完，還可以拿去當慈善、佈施給人、贊助別人，在子女面前，能享有尊榮感。當時若領全額退休金，可能錢會花光光，親朋好友可能會來向我借錢或是拿去亂投資，發生理財不當的情形，現在讓政府幫我理財，每半年給一次，衣食住行不缺，讓我沒有後顧之憂。	享受公務員和宗教從業人員18%的優惠儲蓄，每月領取退休金，讓他們覺得自己擁有取之不盡的財力，慈善事業，親戚贊助，在孩子面前，能享有尊榮感。當時，當我收到全額退休金時，我的親戚朋友來找我借錢或隨意投資，我的理財不合適，但現在政府每六個月幫助我一次理財。別擔心，你不缺衣食住行。	能夠享受18%的公益和宗教工作人員優惠儲蓄，每月領取退休金，感覺好像有取之不盡的財力，還可以做慈善，贊助親戚，在孩子面前享受榮譽。那時候，如果全額退款，我的錢可能會用完，親戚朋友會來找我借錢或隨意投資，理財不妥。現在政府幫我理財，每六個月提供一次。衣食住行不缺，我無後顧之憂。	能夠享受公共和宗教人員18%的優惠儲蓄，並按月領取養老金，給人的印像是財力無窮，還可以做慈善，贊助親戚，在孩子面前享受榮譽。那個時候如果全額退款，我的錢就會用完，親戚朋友會來找我借錢或隨意投資，財務管理不力，現在政府幫我管理一次財務每六個月「吃、穿、住、行」不缺，我無後顧之憂。

圖 16. 利用資料增強進行反向翻譯產生之資料圖

資料來源：本研究整理

4.2、資料處理

資料處理分為三個步驟：

1. 針對每句話逐筆進行語句通順處理，並且以人工方式標註情感分類標籤

表 4、 人工標註情感分類標籤使用標準表

資料來源：本研究整理

LABEL標準				情境							
2分量表	3分量表	5分量表		悲觀	消極	抱怨	不滿意	自卑	不滿	失望	無助
0	0	1	非常負面								
		2	偏負面								
人工判斷0/1	1	3	中等								
1	2	4	偏正面	樂觀	積極	讚美	滿意	驕傲	滿足	希望	感恩
		5	非常正面								

idx	label_2	label_3	label_5	review
0	1	1	5	在我的一生中，我並不覺得那個階段特別重要。年輕的時候要努力掙錢養家糊口，老了有老年的艱苦。現在非常感謝房東提供給我免費住，還有政府的援助。要不然就要像小時候一樣，去當乞丐討飯了！
1	1	2	5	能夠有公教人員18%的優惠儲蓄，領月退休，感覺上好像有取之不盡的錢，用也用不完，還可以拿去做慈善、佈施給人、贊助親人，在子女面前，能享有尊榮感。當時若領全額退休金，可能錢會花光光，親朋好友可能會來向我借錢或是拿去亂投資，發生理財不當的情形，現在讓政府幫我理財，每半年給一次，食衣住行不缺，讓我沒有後顧之憂。
2	1	2	4	現在，每個月都有政府的補助6000元，又幫忙安裝緊急電話，也不時有志工來探望我，我很感激。我最怕政府沒有錢給我，將這補助的6000元給取消掉。拜託你一定要把我這一句話轉告讓政府知道。
3	1	1	3	我讓政府養，這間房子是我姊給我住的，我的兩個女兒都在美國，她們自己都有小孩要養，自己用都不夠了，那有多餘的錢寄回來給我，所以我只能靠政府，不夠花也沒辦法，有多少就用多少，日子可以過就好。
4	0	1	3	我現在的身體狀況就是腸胃不好又貧血。從來沒有去過大醫院看病，最多就是去附近的健保診所買藥吃。我是長期慢性的病，所以也不需要去看什麼醫生，頂多就是掛號拿藥就好。
5	1	2	4	我很少去醫院看醫生，去年我的膀胱發炎，痛到受不了才叫我姪女帶我去醫院，我這一生住院過兩次，一次住一兩天，另一次好像住兩三天吧！好險有政府幫忙，所以住院沒花到什麼錢，若是沒有政府的照顧，應該錢就都沒了！
6	0	0	2	我的生活費用都靠政府的補貼救濟，一個月六千元，省吃節用，還可以過日子，不敢再要求什麼更多的，一天過一天，人能平安就好，政府其他的東西，我都不知道，眼睛也看不見，都沒興趣，也不想參加。
7	1	1	3	我今年拿不到老人津貼，不過明年就可以領了，因為我先生是從新光人壽退休的，他的存款利息三年可以領一次，今年政府在審查老人津貼時，剛好他的利息撥下來，超過政府規定的金額，所以我今年不能領，承辦人員說我明年就可以領老人津貼了。

圖 17. 人工標註情感分類標籤檔案圖

資料來源：本研究整理

表 3、 人工標註情感分類標籤後資料分類分佈表

資料來源：本研究整理

	224筆			896筆		
量表	2分量表	3分量表	5分量表	2分量表	3分量表	5分量表
非常負面	93	70	13	348	244	40
偏負面			57			204
中等		43	43		182	178
偏正面	131	111	81	548	470	354
非常正面			30			120

2. 使用 Jieba 對每句話進行斷詞



圖 18. 進行 Jieba 後資料檔

資料來源：本研究整理

3. 取得每句話的文本向量，使用以下兩種方法

1. 使用 Word2Vec 結合 Doc2Vec 的方法
2. 使用 Tokenize 的方法

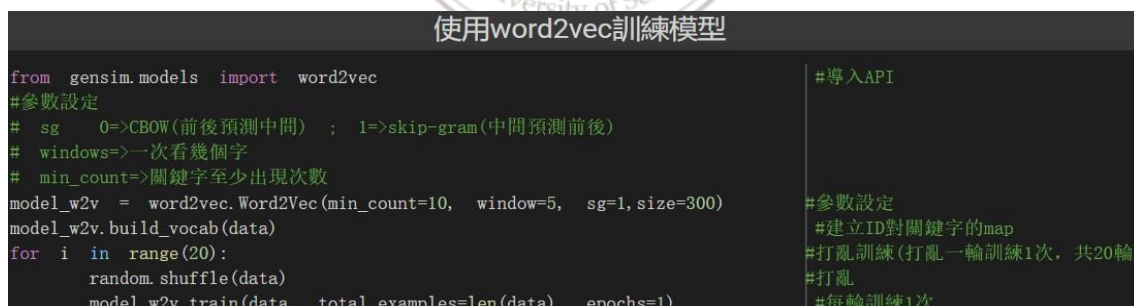


圖 19. 使用 Word2Vec 訓練模型

資料來源：本研究整理

使用Doc2Vec訓練模型

```

from gensim.models import Doc2Vec, doc2vec #導入API
#參數設定
# dm      => 0=>DBOW ; 1=>PV-DM(預設為1)
# min_count => 關鍵字至少需出現次數
# size     => 設定要取得的資料特徵維度，常用維度為100維(考慮神經網路太大)及300維(特徵較多，可較準確)
model_D2V = Doc2Vec(min_count=10, window=4, size=300) #參數設定
model_D2V.build_vocab(sentence_list) #建立ID對關鍵字的map
for i in range(20): #打亂訓練(打亂一輪訓練1次，共20輪)
    random.shuffle(sentence_list) #打亂
    model_D2V.train(sentence_list, total_examples=len(data), epochs=1) #每輪訓練1次

```

圖 22. 使用 Doc2Vec 訓練模型

資料來源：本研究整理

	idx	label_2	label_3	label_5	4	5	6	7	8	9	10	11	12	13	
	0	0	1	1	5	-0.038607	0.013380	-0.009881	-0.022745	-0.002026	0.033909	-0.076558	-0.003528	0.065077	-0.028328
	1	1	0	0	2	-0.034347	0.015868	-0.011205	-0.034692	-0.004747	0.051751	-0.063804	-0.000341	0.042565	-0.031807
	2	2	1	2	4	-0.024573	0.013333	-0.011961	-0.014086	-0.003554	0.028509	-0.034216	0.001148	0.048961	-0.022362
	3	3	1	2	5	-0.026233	0.016249	-0.027236	-0.032049	-0.006631	0.040867	-0.056262	-0.022253	0.039759	-0.042257
	4	4	1	2	4	-0.029560	0.019826	-0.012312	-0.014937	0.011481	0.030357	-0.061676	-0.031648	0.068328	-0.049956

	891	891	0	0	2	-0.026086	0.015168	-0.005176	-0.035141	0.002771	0.032621	-0.058275	0.014786	0.050371	-0.010307
	892	892	1	1	3	-0.041901	0.017655	-0.003136	-0.039908	0.000724	0.045676	-0.071151	0.010520	0.046840	-0.014966
	893	893	1	2	5	-0.021133	0.017802	-0.015958	-0.019877	0.000766	0.034669	-0.028920	0.010283	0.055590	-0.009857
	894	894	1	2	4	-0.026502	0.010337	-0.007992	-0.013556	-0.002041	0.019214	-0.049746	-0.005615	0.047645	-0.023782
	895	895	1	1	3	-0.013645	0.003798	-0.003438	-0.013165	-0.002230	0.013345	-0.017832	0.000734	0.016725	-0.008370
896 rows x 304 columns															

圖 21. 使用 Doc2Vec 模型訓練後取得的文本向量矩陣

資料來源：本研究整理

將句子做Tokenizer

```

from keras.preprocessing.text import Tokenizer #導入API函式庫
#設定參數
max_words = 5000 #最常使用的詞語數
max_len = 250 #最長句子數
#進行Tokenizer，並轉換為矩陣
tok = Tokenizer(num_words=max_words)
tok.fit_on_texts(X_2_train) #將結巴後的斷詞資料丟入Tokenizer做訓練
sequences_2 = tok.texts_to_sequences(X_2_train)
sequences_matrix_2 = sequence.pad_sequences(sequences_2, maxlen=max_len) #轉為矩陣，確保所有序列具有相同的形狀

#確認單詞數量
word_index = tok.word_index
print('共有 %s 個不相同的單詞(Token). ' % len(word_index))

共有 3388 個不相同的單詞(Token).

```

圖 20. 使用 Tokenizer 模型進行訓練

資料來源：本研究整理

切割後訓練資料									
<code>print(train)</code> <code>#訓練資料</code>									
	idx	label_2	label_3	label_5	...	300	301	302	303
293	293	1	2	4	...	0.007618	0.010156	0.000899	-0.010483
425	425	0	0	2	...	0.011684	0.013734	0.001519	-0.014240
657	657	1	2	5	...	0.008564	0.013487	-0.003728	-0.031722
789	789	1	2	5	...	0.007086	0.007275	0.001067	-0.010707
113	113	1	2	5	...	0.024172	0.024331	0.001033	-0.018127
...
835	835	1	2	5	...	0.015772	0.041958	-0.012558	-0.038636
192	192	1	2	4	...	0.006604	-0.000420	0.008592	-0.013419
629	629	1	2	4	...	0.024251	0.026836	0.007374	-0.017029
559	559	0	1	3	...	0.027735	0.019273	0.003953	-0.013599
684	684	1	2	4	...	0.011213	0.017697	0.000218	-0.010807
[761 rows x 304 columns]									

圖 26. 切割後訓練資料集

資料來源：本研究整理

切割後測試資料									
<code>print(test)</code> <code>#測試資料</code>									
	idx	label_2	label_3	label_5	...	300	301	302	303
145	145	1	2	4	...	0.016200	0.021322	-0.001999	-0.035639
60	60	1	2	4	...	0.033903	0.015506	0.017130	-0.000898
733	733	1	2	5	...	0.018730	0.018410	0.001949	-0.017146
31	31	0	0	2	...	0.020497	0.016541	0.003669	-0.016047
506	506	1	2	4	...	0.036464	0.022245	0.013295	-0.000838
...
727	727	1	2	5	...	0.012569	0.014152	0.003764	-0.014548
647	647	0	0	1	...	0.014521	0.015079	0.001255	-0.007463
620	620	0	0	2	...	0.016067	0.018194	-0.003538	-0.021232
501	501	0	0	2	...	0.007151	0.017132	-0.008917	-0.008018
362	362	1	2	5	...	0.033023	0.027119	0.001951	-0.006523
[135 rows x 304 columns]									

圖 25. 切割後測試資料集

資料來源：本研究整理

2. 建立 RNN 中的 LSTM 分類模型

建立LSTM神經網路模型

```
def RNN():
    inputs = Input(name='inputs',shape=[max_len]) # max_len = 250 => 最長句子數
    layer = Embedding(max_words,500,input_length=max_len)(inputs) # max_words = 5000 => 最常使用的詞語數
    layer = LSTM(300)(layer)
    layer = Dropout(0.5)(layer)
    layer = Dense(1,name='out_layer')(layer) # 最後一層參數值為類別數
    layer = Activation('sigmoid')(layer) # 二分類是'sigmoid', 多分類修改為'softmax'
    #layer = Activation('softmax')(layer) # 二分類是'sigmoid', 多分類修改為'softmax'
    model_2 = Model(inputs=inputs,outputs=layer)
    return model_2

model_2 = RNN()
model_2.summary()
model_2.compile(loss="binary_crossentropy", optimizer="adam",metrics=["accuracy"]) # 二分類
#model.compile(loss="categorical_crossentropy", optimizer="adam",metrics=["accuracy"]) # 多分類

# 模型訓練
h_2=model_2.fit(sequences_matrix_2,Y_2_train,batch_size=128,epochs=20,
                validation_split=0.2,callbacks=[EarlyStopping(monitor='val_loss',min_delta=0.0001)])
```

圖 27. 建立 LSTM 分類模型

資料來源：本研究整理

Model: "functional_1"

Layer (type)	Output Shape	Param #
inputs (InputLayer)	[(None, 250)]	0
embedding (Embedding)	(None, 250, 500)	2500000
lstm (LSTM)	(None, 300)	961200
dropout (Dropout)	(None, 300)	0
out_layer (Dense)	(None, 1)	301
activation (Activation)	(None, 1)	0
Total params: 3,461,501		
Trainable params: 3,461,501		
Non-trainable params: 0		

圖 28. LSTM 分類模型結構

資料來源：本研究整理

4.4、輸出結果

使用不同方法執行分類模型，將取得的準確率及 loss 率統計後如表 5：

1. 不同的資料筆數(原始資料 224 筆、資料增強後的 896 筆)
2. 不同比例的切割訓練/測試資料集方式
3. 不同分類量表(2 分量表、3 分量表、5 分量表)
4. 兩種取得文本向量的方式(Doc2Vec 以及 Tokenize)

表 5、LSTM 分類模型結果統計表

資料來源：本研究整理

訓練：測試	量表	資料筆數	原始資料(224筆)				資料增強後(896筆)			
		取向量方法	Doc2Vec		Tokenizer		Doc2Vec		Tokenizer	
		資料集	訓練	測試	訓練	測試	訓練	測試	訓練	測試
60 : 40	2分量表	accuracy	58.21%	58.89%	78.50%	51.85%	62.38%	62.67%	88.34%	85.19%
		loss	0.6875	0.6791	0.5477	0.6972	0.6633	0.6608	0.432	0.8323
	3分量表	accuracy	47.01%	51.11%	50.47%	40.74%	49.72%	47.91%	52.45%	51.85%
		loss	1.0792	1.0127	0.9726	1.0864	1.0444	1.0582	0.9498	2.1491
	5分量表	accuracy	35.82%	36.67%	42.99%	33.33%	36.31%	33.43%	99.30%	83.33%
		loss	1.4981	1.4586	1.4731	1.5329	1.4697	1.4856	0.0623	0.5915
65 : 35	2分量表	accuracy	57.24%	59.49%	69.83%	41.38%	62.03%	63.38%	65.16%	47.01%
		loss	0.7023	0.6777	0.497	0.7295	0.6625	0.6572	0.6192	0.7655
	3分量表	accuracy	49.66%	50.63%	55.17%	37.93%	49.31%	48.41%	100.00%	94.87%
		loss	1.036	1.0233	0.9205	1.1549	1.047	1.0491	0.0117	0.1833
	5分量表	accuracy	35.17%	36.71%	31.03%	17.24%	36.25%	33.70%	97.63%	78.63%
		loss	1.4838	1.4303	1.4748	1.5212	1.4799	1.4919	0.1548	0.6569
70 : 30	2分量表	accuracy	58.33%	58.82%	79.03%	56.25%	62.20%	63.20%	100.00%	95.24%
		loss	0.6739	0.6776	0.563	1.1437	0.6663	0.6585	0.0112	0.129
	3分量表	accuracy	48.08%	51.47%	50.81%	46.88%	48.80%	49.44%	84.83%	80.16%
		loss	1.0488	1.0221	1.023	1.0996	1.0505	1.0387	0.3655	0.4368
	5分量表	accuracy	37.82%	35.29%	38.71%	40.62%	35.09%	35.32%	99.80%	82.54%
		loss	1.4722	1.4596	1.4967	1.4987	1.4969	1.4733	0.0446	0.558
75 : 25	2分量表	accuracy	58.33%	58.93%	59.70%	58.82%	62.05%	63.84%	86.59%	81.48%
		loss	0.6906	0.6777	0.6362	0.6749	0.6681	0.6555	0.3979	0.5759
	3分量表	accuracy	48.81%	51.79%	58.21%	52.94%	48.81%	49.55%	100.00%	90.37%
		loss	1.0446	1.03	0.7701	1.0132	1.0536	1.0476	0.0073	0.3559
	5分量表	accuracy	33.33%	35.71%	38.06%	26.47%	34.82%	35.71%	96.83%	80.00%
		loss	1.5039	1.4739	1.5175	1.6623	1.4899	1.462	0.2225	0.6711
80 : 20	2分量表	accuracy	58.66%	57.78%	89.51%	63.89%	62.15%	63.89%	85.84%	84.03%
		loss	0.6851	0.6811	0.4808	0.631	0.6655	0.655	0.4956	0.5618
	3分量表	accuracy	46.37%	53.33%	51.05%	41.67%	48.04%	52.78%	99.83%	92.36%
		loss	1.0365	1.0192	1.05	1.0963	1.0527	1.0211	0.0163	0.2957
	5分量表	accuracy	30.73%	37.78%	36.36%	44.44%	33.94%	38.89%	100.00%	91.67%
		loss	1.4878	1.4702	1.4802	1.456	1.4932	1.4718	0.0128	0.3108
85 : 15	2分量表	accuracy	59.47%	52.94%	78.95%	57.89%	61.63%	58.52%	100.00%	96.73%
		loss	0.6828	0.695	0.5754	0.695	0.6705	0.6794	0.004	0.1182
	3分量表	accuracy	49.47%	50.00%	51.32%	55.26%	53.22%	48.15%	100.00%	96.08%
		loss	1.0455	1.0346	0.9866	1.0014	1.011	1.0565	0.0049	0.1155
	5分量表	accuracy	37.89%	35.29%	37.50%	36.84%	40.34%	34.07%	100.00%	92.16%
		loss	1.4526	1.4887	1.5206	1.5595	1.4451	1.4624	0.0044	0.3744
90 : 10	2分量表	accuracy	58.71%	60.87%	61.25%	46.34%	62.78%	60.00%	98.45%	97.53%
		loss	0.6936	0.6693	0.6824	0.6962	0.6634	0.6734	0.0375	0.1119
	3分量表	accuracy	49.25%	56.52%	71.88%	48.78%	48.88%	50.00%	86.18%	76.54%
		loss	1.0413	1.0278	0.67	0.9887	1.0491	1.0372	0.4828	0.7497
	5分量表	accuracy	38.31%	34.78%	33.75%	31.71%	33.62%	36.67%	100.00%	91.98%
		loss	1.4706	1.576	1.5226	1.6016	1.4904	1.4746	0.0213	0.2949

第五章 結論

5.1 結論

以往的對話機器人，關於對話內容的重點幾乎都放在讓對話的語意通順、能夠了解說話的意圖目的之類，對於情感分析的部分，大多數都著重在臉部辨識或是語音辨識這些部分，幾乎很少考量到對話內容的文本情感分析。

本研究針對不同的資料集、不同的文本向量取得方法、不同的分類標籤數，並利用不同的資料集切割方法進行驗證，來實驗出不同的模型來進行分析。

以輸出結果來看：

在 2 分量表部分，經過資料增強後進行訓練的準確率，幾乎都優於只使用原始資料進行訓練的準確率，表示使用較多的資料進行訓練，會比使用較少資料來進行訓練的效果還要好。不管是只使用原始資料進行訓練還是使用經過資料增強後的資料來進行訓練，利用 Tokenize 方法取得文本向量的準確率大多數都優於使用 Doc2Vec 方法取得文本向量的準確率。資料集切割成訓練資料以及測試資料來進行訓練的結果，使用的切割比例為使用 80%：20% 或是 85%：15% 所呈現的效果是最佳的。

在 3 分量表部分，經過資料增強後進行訓練的準確率，大多數都優於只使用原始資料進行訓練的準確率，表示使用較多的資料進行訓練，會比使用較少資料來進行訓練的效果還要好。不管是只使用原始資料進行訓練還是使用經過資料增強後的資料來進行訓練，利用 Tokenize 方法取得文本向量的準確率大多數都優於使用 Doc2Vec 方法取得文本向量的準確率。資料集切割成訓練資料以及測試資料來進行訓練的結果，使用的切割比例為使用 85%：15% 或是 90%：10% 所呈現的效果是最佳的。

在 5 分量表部分，經過資料增強後進行訓練的準確率，大多數都優於只使用原始資料進行訓練的準確率，表示使用較多的資料進行訓練，會比使用較少資料來進行訓練的效果還要好。不管是只使用原始資料進行訓練還是使用經過資料增強後的資料來進行訓練，利用 Tokenize 方法取得文本向量的準確率大多數都優於使用 Doc2Vec 方法取得文本向量的準確率。資料集切割成訓練資料以及測試資料來進行訓練的結果，使用的切割比例為使用 80%：20% 或是 85%：15% 所呈現的效果是最佳的。

5.2 建議

如果想要作與本研究相似的文本分類相關的題目的話，若是資料量也不多的話，會建議也參考資料增強的方法將資料量進行增加，進行資料增強後最重要的還是要以人工的方式將資料進行整理並進行標籤，畢竟要是資料的乾淨、完整、合理性...等條件會影響研究結果的正確性及呈現效果。而取得文本向量的方法，則是建議使用 Tokenize 的方法來取得，效果基本上都大幅優於使用 Doc2vec 的方法。資料的訓練資料集以及測試資料集的切割比例，大多進行機器學習的人都用 60%~90% 的訓練資料集來進行切割，但主要還是要看資料量來進行測試，以本研究 1,000 筆左右，最適合的比例是在 80%~85% 左右的訓練資料集來進行切割，畢竟想要呈現出不錯的效果，還是需要使用足夠數量的資料進行訓練才能跑出來。

5.3 未來展望

希望之後有機會能夠將此模型建立成一個模組，以便導入並加強對話機器人應用：即使用者說出一句話之後，利用語音轉換成一段文字後，能夠先經過本模組及其他情感分析相關模組來取得使用者當下情緒，再經由所取得的情緒分類來選擇比較適合於使用者當下情緒的回覆語料庫，或是選擇不適用於當下情緒的禁用語料庫，來進行回覆對話內容的篩選，讓對話機器人不會因為回覆的對話內容不洽當而影響使用者的體驗，而且能夠回覆的更加具有情感以及溫度。

參考文獻

中文文獻

- Chilan Yuk (2020). Keras 分詞器 Tokenizer. Retrieved. 2021/07/04. from. <https://zhuanlan.zhihu.com/p/138054335>
- Github. Jieba. Retrieved. 2020/11/27. from. <https://github.com/fxsjy/jieba>
<https://www.jiqizhixin.com/articles/2021-04-27-6>
- itread01(2019). 情感分類. Retrieved. 2021/07/03. From. <https://www.itread01.com/content/1547469491.html>
- iT 邦. Day 1: 自然語言處理簡介. Retrieved. 2020/11/27. from. <https://ithelp.ithome.com.tw/articles/10213115?sc=iThomeR>
- Mars.Su(2020). NLP Data Augmentation 常見方法. Coderbridge. Retrieved. 2021/07/03. from. <https://marssu.coderbridge.io/2020/10/26/nlp-data-augmenatation-%E5%B8%B8%E8%A6%8B%E6%96%B9%E6%B3%95/>
- Mikechenx(2017). Day 14: 循環神經網路(Recurrent Neural Network, RNN). iT 邦. Retrieved. 2021/01/09. from. <https://ithelp.ithome.com.tw/articles/10193469>
- weifanhaha(2020). RNN 進化了-LSTM 與 GRU. coderbridge. Retrieved. 2021/01/10. from. <https://zh-tw.coderbridge.com/series/2ec9cf0af3f74ed99371952f4849ae33/posts/2be54d939e0249f392eba159e5126e0b>
- 中華人民共和國民政部. 三部委關於印發《智慧健康養老產業發展行動計劃(2017-2020 年)》的通知. Retrieved. 2021/01/21. from. <http://www.mca.gov.cn/article/xw/tzgg/201702/20170215003313.shtml>
- 王鈞威, 簡聖倫, 陳義昆, 吳世弘. 基於 Seq2Seq 模型的中文文法錯誤診斷系統. 朝陽科技大學資訊工程系. 台中市.
- 朱晏呈(2019). Feedforward Neural Networks 於連續手勢辨識之研究. 國立臺灣師範大學資訊工程研究所碩士論文. 臺北市.
- 余新荃(2018). 以聊天機器人為主之長期照顧行動應用開發. 大同大學資訊經營學系(所)碩士論文. 臺北市.
- 李宏毅. 台灣大學-機器學習課程講義. Retrieved. 2021/01/09. from. <https://speech.ee.ntu.edu.tw/~tlkagk/courses ML19.html>
- 李魚愷(2016). 人工智慧應用於基本人機互動會話系統發展. 逢甲大學自動控制工程學系碩士論文. 台中市.
- 李勝凱(2018). 聊天機器人應用之探討—以南華大學資管系為例. 南華大學科技學院資訊管理學系碩士論文. 嘉義縣

每周科技詞. 自然語言處理. Retrieved. 2020/12/20. from.

http://c020.wzu.edu.tw/datas/upload/files/Documents/%E6%AF%8F%E5%91%A8%E7%A7%91%E6%8A%80%E8%A9%9E/%E6%AF%8F%E5%91%A8%E7%A7%91%E6%8A%80%E8%A9%9E_19_1080710_%E8%87%AA%E7%84%B6%E8%AA%9E%E8%A8%80%E8%99%95%E7%90%86.pdf

林玫君，劉立凡，李懿珍.(2016/05). 高齡化社會孤獨死之預防與因應. 社區發展季刊 153 期

洪麒盛(2019). 人工智慧情感對話機器人. 淡江大學資訊管理學系碩士論文. 新北市

范光中，許永河. 台灣人口高齡化的社經衝擊. 台灣老年醫學暨老年學雜誌 2010；5(3)：149-168.

<http://www.tagg.org.tw/DOWN/%E9%9B%9C%E8%AA%8C/5-3-%E5%85%A8%E6%96%87/5-3-1%E7%B6%9C%E8%AB%96-%E8%8C%83%E5%85%89%E4%B8%AD-p149-168.pdf>

徐婷婷&張赫(2020). 養老邁入智慧時代 智能機器人能陪老人聊天、助休閒. 人民網-人民日報. Retrieved. 2021/01/21. from.

<http://it.people.com.cn/BIG5/n1/2020/0212/c1009-31582655.html>

國家政策研究基金會. 人口推估. Retrieved. 2021/01/16. from.

https://www.ndc.gov.tw/Content_List.aspx?n=81ECE65E0F82773F

國家政策研究基金會. 少子化、高齡化 Q&A. Retrieved. 2020/11/27. from.

<https://www.npf.org.tw/13/8108>

張偉男，劉挺(2016). 聊天機器人技術的研究進展. 中國人工智慧通訊學會深度學習 234. (2018) 人人都能看懂的 GRU. 知乎. Retrieved. 2021/01/10. from.

<https://zhuanlan.zhihu.com/p/32481747>

深度學習 234. (2018) 人人都能看懂的 LSTM. 知乎. Retrieved. 2021/01/10. from.

<https://zhuanlan.zhihu.com/p/32085405>

陳人豪，潘博詠，林聖翔. 智能語音點餐系統. 雲林科技大學資訊工程系. 雲林縣.

陳峰瑛. 獨居老人的社會支持與生活適應之探究-以高雄都會區的獨居老人為例.

南華大學生死學系. 嘉義縣

陳鴻麒. 迎接高齡化時代，保健醫療少不了 AI (2019，4，9) Medium Retrieved. 2020/11/27. from.

<https://medium.com/@stevehcchen/%E8%BF%8E%E6%8E%A5%E9%AB%98%E9%BD%A1%E5%8C%96%E7%A4%BE%E6%9C%83-%E4%BF%9D%E5%81%A5%E9%86%AB%E7%99%82%E5%B0%91%E4%B8%8D%E4%BA%86ai-ca-c0952ca45>

陶治中，簡睿志(2016/12). 三元決策理論應用於社群媒體挖掘之情感分析-以 UBER 在臺營運話題為例. 運輸計劃季刊 第四十五卷 第四期 頁 301~330.

- 黃民烈&張岳 (2021). 像人一樣自然流暢地說話，下一代智能對話系統還有多長的路要走？ 機器之心. Retrieved. 2021/07/01. from.
https://www.sohu.com/a/463281456_129720
- 新華網融媒體未來研究院(2015). 劉挺:「文本情感分析」將成為攻克機器人「情商」難題的重要選項. 每日頭條. Retrieved. 2021/07/03. from.
<https://kknews.cc/media/6nzo8m.html>
- 詹峻陽. 人工智慧三大關鍵技術. 數位時代. Retrieved. 2020/12/20. from.
<https://www.bnext.com.tw/article/41534/3-key-techniques-of-ai>
- 維基百科. Python. Retrieved. 2020/11/27. from. <https://zh.wikipedia.org/zh-tw/Python>
- 維基百科. Word2vec. Retrieved. 2020/11/27. from.
<https://zh.wikipedia.org/wiki/Word2vec>
- 維基百科. 自然語言處理. Retrieved. 2020/11/27. from.
<https://zh.wikipedia.org/wiki/%E8%87%AA%E7%84%B6%E8%AF%AD%E8%A8%80%E5%A4%84%E7%90%86>
- 維基百科. 聊天機器人. Retrieved. 2020/11/27. from.
<https://zh.wikipedia.org/wiki/%E8%81%8A%E5%A4%A9%E6%A9%9F%E5%99%A8%E4%BA%BA>
- 趙陽洋、王振宇、王佩、楊添、張睿、尹凱(2020/10). 任務型對話系統研究綜述. 計算機學報. 第 43 卷第 10 期.
- 劉秉鈞(2016). 人工智慧於模擬人類對話系統設計. 逢甲大學自動控制工程學系碩士論文. 台中市.
- 劉冠麟(2020). 機器學習分類防疫新聞. 中央大學通訊工程學系碩士論文. 桃園市.
- 蔡炎龍, 林澤佑. 政治大學磨課師課程. Retrieved. 2021/01/09. from.
<http://moocs.nccu.edu.tw/course/172/intro>
- 衛生福利部(2017). 長照十年計畫 2.0 — 建立我國社區整體照顧模式, 佈建綿密照顧網. Retrieved. 2021/01/21. from.
<https://1966.gov.tw/LTC/cp-3636-42415-201.html>
- 蕭伯瑋(2018). 應用深度神經網路與集成學習於語音情緒辨識. 國立中山大學資訊工程學系碩士論文. 高雄市.
- 顏理謙(2016). 聊天機器人崛起, 下一個兵家必爭之地將是對話式商務. 數位時代. Retrieved. 2020/12/20. from.
<https://www.bnext.com.tw/article/42076/chatbot-conversational-commerce-is-coming>

英文文獻

- Abigail See , John Hewitt(2020). Natural Language Processing with Deep Learning CS224N/Ling284. Lecture 7: Vanishing Gradients and Fancy RNNs. Retrieved. 2021/01/09. From.
<http://web.stanford.edu/class/cs224n/slides/cs224n-2020-lecture07-fancy-rnn.pdf>
- Andrej Karpathy blog(2015). The Unreasonable Effectiveness of Recurrent Neural Networks. Retrieved. 2021/01/09. From.
<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>
- Cambria E , Schuller B , Xia Y , Havasi C. New avenues in opinion mining and sentiment analysis. IEEE Intelligent Systems. 2013;28(2):15–21.
- colah's blog. Understanding LSTM Networks. Retrieved. 2020/12/20. from.
<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Dan Jurafsky , Chirs Manning. 史丹佛大學自然語言處理課程-第七課”情感分析 (Sentiment Analysis)”. Retrieved. 2021/07/03. from.
- Das S. and Chen M. ”Yahoo! for Amazon : Extracting Market Sentiment from Stock Message Boards” , Proceedings of the Asia Pacific Finance Association Annual Conference (APFA) , 2001 , pp. 1367-1373.
- Dzmitry Bahdanau , KyungHyun Cho , Yoshua Bengio , 2014 , “Neural Machine Translation by Jointly Learning to Align and Translate” , arXiv preprint arXiv:1409.0473.
<https://class.coursera.org/nlp/>
- K. Cho , B. Van Merriënboer , C. Gulcehre , D. Bahdanau , F. Bougares , H. Schwenk , and ” Y. Bengio , “Learning phrase representations using rnn encoder-decoder for statistical machine translation , ” arXiv preprint arXiv:1406.1078 , 2014.
- Le Q , Mikolov T. Distributed representations of sentences and documents[C]//International Conference on Machine Learning. 2014: 1188-1196.
- Margaret Rouse(2017). What is a chatbot?. Techtarget. Retrieved. 2020/12/20. from.
<http://searchdomino.techtarget.com/definition/IM-bot>
- Olah , C. (2015). Understanding LSTM Networks. Retrieved. 2021/01/10. from.
<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Sutskever , I. , Vinyals , O. , and Le , Q. V. “Sequence to sequence learning with neural networks.” , Advances in neural information processing systems. 2014. p. 3104-3112.
- T. Mikolov , K. Chen , G. Corrado , and J. Dean , “Efficient estimation of word representations in vector space , ” arXiv preprint arXiv:1301.3781 , 2013.
- Techtarget (2017). What is chatbot? - Definition from WhatIs.com. Retrieved from.

<https://searchcrm.techtarget.com/definition/chatbot>

The 2019 Conference on Computational Linguistics and Speech Processing ROCLING 2019 , pp. 358-367

Yu , Y. , Si , X. , Hu , C. , & Zhang , J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural computation* , 31(7) , 1235–1270.
https://doi.org/10.1162/neco_a_01199

