

法鼓文理學院佛教學系碩士論文

基於深度學習的佛教引用句推薦系統

Building a Buddhist Quotes Recommendation

System Based on Deep Learning Approaches



指導教授： 洪振洲博士

王昱鈞博士

研究生： 釋廣慧

中華民國一百一十一年七月

法鼓文理學院碩博士論文授權書

中華民國 110 年 10 月 13 日 110 學年度第 2 次教研會議修正通過

- 立書人（即論文作者）：釋廣慧（下稱本人） 學號：M109103
- 授權標的：本人於 法鼓文理學院（下稱學校）佛教學系（學系、碩士學位學程）
110 學年度第 2 學期之 碩士 博士 學位論文。

論文題目：基於深度學習的佛教引用句推薦系統

指導教授：洪振洲博士 王昱鈞博士

（下稱本著作，本著作並包含論文全部、摘要、目錄、圖檔、影音以及相關書面報告、技術報告或專業實務報告等，以下同）

緣依據學位授予法等相關法令，對於本著作及其電子檔，學校圖書館得依法進行保存等利用，而國家圖書館則得依法進行保存、以紙本或讀取設備於館內提供公眾閱覽等利用。此外，為促進學術研究及傳播，本人在此並進一步同意授權學校、國家圖書館、資料庫廠商等對本著作進行以下各點所定之利用：

一、對於學校之授權部分：

本人 同意 不同意（請勾選其一）授權學校，無償、不限期間與次數重製本著作並得為教育、科學及研究等非營利用途之利用，其包括得將本著作之電子檔收錄於數位資料庫，並透過自有或委託代管之伺服器、網路系統或網際網路向 學校校園內 校外位於全球使用之使用者（本點如前求勾選同意者，請勾選，並得複選）公開傳輸，以供該使用者為非營利目的之檢索、閱覽、下載及/或列印。

二、對於國家圖書館之授權部分：

本人 同意 不同意（請勾選其一）授權國家圖書館，無償、不限期間與次數重製本著作並得為教育、科學及研究等非營利用途之利用，其包括得將本著作之電子檔收錄於數位資料庫，並透過自有或委託代管之伺服器、網路系統或網際網路向館內及館外位於全球之使用者公開傳輸，以供使用者為非營利目的之檢索、閱覽、下載及/或列印。

三、對於資料庫廠商之授權部分：

本人 同意 不同意（請勾選其一）由學校將本著作有（無）償授權資料庫廠商（下稱該資料庫廠商或該廠商）進行以下範圍之利用：

- （一）該資料庫廠商得將本著作重製收錄於其所建置營運之特定數位資料庫（下稱該資料庫），並透過網際網路向全球訂購該資料庫之使用者公開傳輸，以供該使用者為非營利目的之檢索、閱覽、下載及/或列印。
- （二）該資料庫廠商不得再轉授權第三人將本著作重製收錄於其他資料庫或進行其他營利或非營利利用。但於台灣以外之海外地區，該廠商得委託當地之代理商或經銷商代為處理當地使用者訂購該資料庫事宜。
- （三）若該合作以有償方式進行，則資料庫廠商因本點授權利用本著作所取得之收益，應依該廠

商與學校授權契約支付本人合理權利金，支付標準由學校為本人利益而全權與該廠商議定。本人同意，上開權利金(以下請勾選其一)：

由資料廠商批次轉與學校，作為校務發展基金。

應給付本人，並由該廠商直接通知本人領取，且聯絡資料倘有不全、錯誤或異動而未書面通知，導致權利金無法給付，或收到廠商通知未回覆者，於次年3月31日後，自動將此筆款項由資料廠商批次轉與學校，作為校務發展基金。

(四) 本人保有隨時終止本點授權之權利，並於本人向學校辦理完成終止授權相關程序後，由學校通知該廠商將本著作自該廠商資料庫中刪除且不得再為其他形式之利用。但終止前已完成訂購之使用者，則視該使用者之訂購條件，由學校與廠商協商其提供及刪除時間。

四、本授權書第一點至第三點所定授權，均為非專屬且非獨家授權之約定，本人仍得自行或授權任何第三人利用本著作。

五、本授權書第一點至第三點所定授權對象，依各該點授權利用本著作時，均應尊重本人著作人格權及權利管理電子資訊等相關權利，不得以任何方式省略、增修或變更本人署名、本著作名稱、本著作內容及相關資料(包括本人原記載取得學位論文之學校全銜、書目等詮釋資料等)。第三點所定資料庫廠商亦應要求其代理商或經銷商遵守。

六、依本授權書第一點至第三點將本著作透過網際網路對外公開之時間(請勾選)：

於本授權書簽署日，均立即對外公開。

本人要求本著作應自民國__年__月__日起始得對外公開，故因本授權書第一點至第三點所定授權而發生得透過網際網路對校外、館外或對資料庫使用者之公開傳輸部分，亦應自該日起始生效力。

七、本授權書第一點至第三點分別所定各該授權對象，均應各自遵守其授權範圍及相關約定。如有違反，由該違反之行為人自行承擔一切法律責任。

八、本人擔保本著作為本人創作而無侵害他人著作權或其他權利。如有違反，本人願意自行承擔一切法律責任。

九、個資利用同意條款：本人同意，學校及國家圖書館為本授權書所定各授權事項目的範圍內(但勾選「不同意」者除外)得蒐集、處理及利用本人所提供之個人資料，學校並可將該等個人資料提供給包括國家圖書館及資料庫廠商在內之相關第三人在同一目的範圍內處理及利用。

研究生簽名： 釋廣慧

民國： 111 年 7 月 27 日

法鼓文理學院佛教學系碩士班

研究生學位論文口試委員會審定書

110 學年度第 2 學期

研究生：釋廣慧

題目：(中文) 基於深度學習的佛教引用句推薦系統

(英文) Building a Buddhist Quotes Recommendation

System based on Deep Learning Approaches

業經本委員會審議通過

學位論文口試委員

高照明

莊國材

指導教授

洪振坤

指導教授

王昱鈞

系主任

王昱鈞

中華民國 111 年 7 月 13 日

摘要

在撰寫文章時，我們時常會引經據典，引用一些名句以增強文章的說服力。所謂的佛教引用句指的是常見的佛教用語或經典名句，佛教引用句的哲學常給人心一種啟發作用，例如「凡所有相，皆是虛妄」、「應無所住，而生其心」「一切有為法，如夢幻泡影」等。因此於文章寫作時，如果能引用合適的佛教引用句，將可以讓文章更加生動有力。然而，對於一般人來說，要深入理解並熟知佛教引用句並不容易，此時如有一個推薦的方法，自動從文章的內容進行分析後，推薦合適的佛教引用句，寫作時更能得心應手。

在本文中，我們提出寫作時引用佛教引用句的推薦方法。我們嘗試解決這個吸引人的 AI 任務，並建立名言推薦的學習框架；首先，從網路上收集比較通俗的佛教引用句，並挖掘包含這些佛教引用句的文章，以建立一個用於實驗的資料集。再者，實驗以監督學習方式運用了詞嵌入(Embedding)、深度學習(Deep Learning)方法為這項任務建立模型，用以分析理解佛言名句特徵與佛教引用句上下文特徵之間的相關性進行判斷，最後輸出佛教引用句推薦的引用。實驗結果顯示，我們提出的方法對於這項任務是合適的且準確度達 91.48%。

關鍵字：佛教引用句、深度學習、中文寫作、自然語言處理、推薦

ABSTRACT

In writing an article, we often quote maxims or some famous sayings to enhance the persuasiveness of the article. Buddhist quotation is a vital and meaningful source of the maxims. The so-called Buddhist quotation refers to the common Buddhist expressions or famous classic sayings. The philosophy of famous Buddhist quotations often makes people enlightened. Therefore, when writing an article, if we can quote the appropriate Buddhist maxim, it will make the article more vivid and powerful. However, for the average person, understanding and being familiar with famous Buddhist quotations is not easy, so a recommended method is necessary to automatically analyze the article's content and then recommend the appropriate Buddhist quotations.

In this article, we propose and address the Buddhist quotation recommendation method to suggest famous Buddhist maxims. We adopt a deep learning approach to establish the quotation recommendation system. First, famous Buddhist maxims are selected, and the articles containing these Buddhist maxims are collected to construct a dataset for experimentation. Then, the supervised learning model comprised of word embedding and deep learning methods is established to generate a ranking list of appropriate Buddhist quotations based on the input context. The recommendation accuracy of our proposed method achieves 91.48%, that shows our method is effective for this quotation recommendation task.

Keywords : Buddhist quotes, Deep Learning, Chinese writing, NLP, Recommendation

誌謝

時光飛逝、歲月如梭，轉眼間兩年半的研究所生涯即將邁入尾聲，驀然回首總有釋懷的感覺，心中滿滿不捨，感謝讓我有所蛻變的一切。此篇論文的完成承蒙許多人的支持以及鼓勵。首先感謝我的指導教授洪振洲老師，啟發我一開始的論文構想，讓我一步一步學習及成長，感謝王昱鈞老師，細心及耐心的教學及技術輔導，兩位老師在論文撰寫期間給予無數的教導與督促，並且讓我從中學習到對於研究的執著與態度。在論文口試期間內，感謝莊國彬老師與高照明老師所給予的寶貴意見與指正，讓此篇碩士論文更臻於完善，在此表示深摯的謝忱。

在求學期間內，感謝同班同學們在研究上彼此的激勵與共同成長，很高興在我的人生旅途當中遇見你們，我會永遠珍惜這個緣分。

最感謝的是我的 師父^上悟^下行法師，由於您的支持，我研究所生涯中添增了豐富的色彩，在我低潮喪失動力之時總是能帶給我無窮盡的鼓勵，給予我心靈上的滋潤並且洗滌疲憊的身心。最後，感謝默默支持與關心我的台南極樂寺同參道友，讓我在求學過程當中無後顧之憂，忙碌之餘總是為我加油打氣，體諒我執意跑到異鄉讀書，我想我並沒有讓你們失望！在此分享這份喜悅給所有的人，我是幸福的，因為有你們！！

釋廣慧 謹致
法鼓文理學院佛教學系
中華民國 111 年 7 月

目次

一、研究動機與目的.....	1
二、相關文獻回顧.....	3
(一) 基於內容的引用推薦.....	3
(二) 監督學習式引用推薦.....	4
(三) 深度學習的引用推薦.....	4
三、問題定義.....	6
四、研究方法.....	8
(一) 資料搜集及清理.....	9
1 步驟一：候選佛教引用句清單.....	9
2 步驟二：佛教引用句的上下文.....	11
(二) 前置處理.....	13
1 Jieba.....	14
2 Tokenization.....	14
3 Zero padding.....	15
4 One-hot encoding.....	16
(三) 模型訓練與預測.....	16
1 Baseline 方法：機器學習分類器.....	17
2 深度學習模型.....	21

3 訓練及預測	28
(四) 實驗設計與結果.....	29
1 Baseline 方法：機器學習分類器.....	30
2 深度學習模型：LSTM 及 BiLSTM	31
3 BERT	40
4 綜合實驗結果	44
(五) 實際推薦.....	49
1 揀選文章	49
2 建立環境	49
3 前置處理	49
4 預測	49
五、結論.....	52
六、參考資料.....	53



表目錄

表 1 候選佛教引用句與結果標籤對應表.....	9
表 2 網站網址.....	13
表 3 某班學生數學英文成績	18
表 4 分類器模型準確度比較.....	30
表 5 Top1 Top3 Top5 推薦準確度及錯誤數.....	30
表 6 模型固定參數.....	31
表 7 LSTM 模型及參數實驗結果.....	33
表 8 BiLSTM 模型及參數實驗結果.....	34
表 9 兩組模型相同參數實驗比較.....	34
表 10 LSTM 預測錯誤數分析.....	35
表 11 預測資料中意義相近比較.....	36
表 12 訓練資料量少分析.....	39
表 13 LSTM Top1/Top3/Top5 推薦準確度及錯誤數	40
表 14 BERT 實驗結果.....	42
表 15 每一結果標籤預測錯誤率.....	42
表 16 BERT Top1/Top3/Top5 推薦準確度及錯誤數.....	43
表 17 SVM、LSTM、BERT 之 Top1/Top3/Top5 Accuracy	45
表 18 標籤 0、1、3、7、12 錯誤率	46
表 19 標籤 13、15 錯誤率.....	46

圖目錄

圖 1 TensorFlow 架構圖.....	8
圖 2 佛教引用句類編.....	10
圖 3 佛教引用句類編內容.....	10
圖 4 網頁搜尋文章範例.....	11
圖 5 html 文件標籤位置圖.....	12
圖 6 機器學習的分支.....	17
圖 7 KNN 之 K=3 分類法.....	18
圖 8 決策樹分類法.....	19
圖 9 SVM 決策邊界超平面.....	20
圖 10 機器學習分類器模型架構.....	21
圖 11 RNN 單元及其展開形式.....	22
圖 12 Sigmoid function.....	22
圖 13 長短期記憶單元(LSTM cell) 架構.....	24
圖 14 LSTM 模型架構及參數.....	25
圖 15 LSTM and BiLSTM 架構比對.....	26
圖 16 Transformer.....	26
圖 17 BERT Embedding 詞向量.....	27
圖 18 BERT 模型輸入輸出略圖.....	28
圖 19 K 折交叉驗證範例.....	29
圖 20 Embedding 輸出維度.....	32
圖 21 RNN 輸出維度.....	32
圖 22 LSTM 預測錯誤數.....	35
圖 23 標籤 0 錯誤數分析.....	37

圖 24 標籤 1 錯誤數分析.....	38
圖 25 標籤 13 錯誤數分析.....	39
圖 26 模型概要 (Summary)	41
圖 27 BERT 每一標籤預測錯誤數.....	43
圖 28 BERT Top1/Top3/Top5 錯誤數比對.....	44
圖 29 LSTM、BERT 錯誤數比較.....	46
圖 30 揀選文章實際測試結果.....	50



法鼓文理學院
Dharma Drum Institute of Liberal Arts

一、研究動機與目的

我們在寫作中，很常見的情況是想引用某人的陳述，像一句諺語、一些名人的陳述、佛學經文或是依據歷史典故等，使我們的文章更加優雅或令人信服。然而，有時我們非常渴望在文章某處引用名言佳句，但不知道應使用何種相關佳句來表達我們的想法，因為知道或記誦這麼多名言佳句並不容易，如果能有一個推薦系統為我們在寫作時適當推薦可利用的相關引用是件令人興奮之事。常見的佛教用語或經典名句的哲學常給人心一種啟發作用。現代社會，一般人一心追求物質生活的提升，因緣較好的物質生活追求到了，慢慢就形成奢侈，有的人因緣較差，始終無法在物質上得到滿足；如何才能使佛教在今天人們建設精神家園中發揮更好的積極作用？佛教是一種精神性的文化資源，能為人的精神世界提供資糧，它應該在人的精神世界發揮更大的作用。在學校校園、觀光景點、交通要道、候車站等，能夠把這些佛教引用句有關佛教積極、樂觀、正信、進步及多采多姿的一面呈現出來，扭轉一般人既有的成見。因此於文章寫作時，如果能引用合適的佛教引用句，將可以讓文章更加生動有力。

然而對於一般人來說，要深入理解並熟知佛教引用句並不容易，因此需要有一個推薦的方法，自動從文章的內容進行分析後，推薦合適的佛教引用句，此正是本研究的研究主題。譬如有一段文章「如果鬱悶肇因於對人、事、物的執著，若能常誦佛門名偈，則更是殊勝了得也。…，我自己鬱悶難當之際，便常用此偈語觀照眼前惱事，頓時，只覺執著心下降，寬恕心上升，心胸掙開了無形之束縛，一股芬芳清流悠然而至，於是乎，心得輕安自在，爭較心早得調伏矣！」，文章段落中保留「…」之處是我們希望能提供佛教引用句引用的位置，而我們希望推薦方法能對這篇文章段落推薦合適的佛教引用句，此處填上「一切有為法，如夢幻泡影」最適合，因為人、事、物是「一切有為法」，因執著而鬱悶難當，能觀照這一切「如夢幻泡影」，心就能輕安自在。

迄今為止，關於寫作的推薦，雖然已經有一些論文基於內容為主的分類與推薦系統，但並未見有對於佛教引用句的推薦做出研究。Tan et al. (2015)提出一種列表方法—基於線性特徵的模型來學習排名，用以推薦名言佳句，且證明其有用性，希望使用深度學習提高其性能。王皓儀 (2019)使用 RNN (循環神經網路) 之 LSTM 及 BiLSTM 模型推薦文句中適合的成語，結論中提到若有更完整的上下文資訊或許會有更適切的結果。本文正是聚焦於針對文章寫作進行佛教引用句之推薦，我們將建立並實驗一個深度學習模型來推薦根據上下文訊息特徵所分析預測的相關佛教引用句。

本論文所提供的主要貢獻在於：(一) 我們提出了一個新穎的佛教引用句推薦任務並討論任務的特殊性，並構建了一個任務的真實數據集。(二) 我們建議利用 RNN 深度學習架構進行排名以解決推薦任務。(三) 我們在一個真實數據集上進行了一系列實驗和評估，結果驗證了我們提出的方法的有效性和實用性。



二、相關文獻回顧

劉藹萍在《現代漢語》說明，引用是指在說話或寫作中引用現成的話，如詩句、格言、成語等，以表達自己思想感情的修辭方法。引用可分為明引暗引兩種。明引指直接引用原文，並加上引號，或者是只引用原文大意，不加引號，但是都注明原文的出處。暗引指不說明引文出處，而將其編織在自己的話語中，或是引用原句，或是只引大意。運用引用辭格，既可使文章言簡意賅，有助於說理抒情；又可增加文采，增強表現力。

引用其他人的諺語和著名的陳述可以提供支持，提供新的視角，或為一個人添加幽默感或文章論點。在機器學習中，寫作推薦佛教引用句可以通過考慮當前文本（稱為上下文）的各種特徵來完成。佛教引用句推薦可以看作是基於內容的推薦任務，其最密切相關的工作是基於內容的科學寫作引述推薦，因為這兩個任務都是基於文本內容進行推薦，所以一些引文推薦方法，如下所列各學者基於內容的引用推薦、監督學習式引用推薦、基於深度學習引用推薦，皆可以應用於佛教引用句推薦的參考。

(一) 基於內容的引用推薦

回顧推薦引用的歷史，Shaparenko, B., and Joachims, T. (2009) 解決引文的上下文和論文內容的相關性，並將語言模型應用於推薦任務。He et al. (2010) 提出了一種上下文感知方法，該方法測量出基於上下文在於引用上下文和一份文件之間的關聯，其核心思想是設計一種非參數概率模型，此模型可以測量引用上下文和文檔之間基於上下文的相關性，而且此方法可以有效地推薦上下文的引用。Lu et al. (2011) 進行了一項實驗，發現翻譯模型(translation model)可以提供比先進的方法更好的引用候選者。Tang et al. (2014) 針對中文論文中引用位置給定上下文推薦英文引文，

提出了雙語上下文引用嵌入算法，它是可以學習上下文和引用的低維聯合嵌入空間的方法。

(二) 監督學習式引用推薦

這些方法探索多個特徵來估算引文和上下文之間的相關性，並嘗試將這些特徵與監督學習方法結合起來。Bethard 和 Jurafsky (2010) 說明早期的方法，它利用線性分類器來學習一個推薦模型。Rokach 等人 (2013) 提出了一個引用推薦的監督學習框架，他們使用分類器來利用三個(general、author-aware、context-aware)特徵類型。早期的研究沒有使用學習來對引文推薦算法進行排名。Tan, J., Wan, X. and Xiao, J. (2015). 應用學習排序模型來整合多個特徵並顯示學習排名模型適用於這個任務，證明上下文信息在名言推薦任務中非常有用。

(三) 深度學習的引用推薦

Jiwei Tan, Xiaojun Wan and Jianguo Xiao,(2016).提出了一個基於 LSTM 的神經網路方法來推薦名言佳句任務。其直接學習上下文和佳句的分佈式含義表示，然後根據含義衡量相關性。特別是，試圖代表帶有特定嵌入的佳句中的詞組，根據語境、主題，甚至引用的作者偏好。在大型數據集上的實驗結果證明，提出的方法實現了最先進的性能它優於幾個強大的基線(Baseline)方法。

王皓儀(2019). 以深度學習應用於中文成語與同義成語寫作之推薦，使用中文成語例句語料庫來訓練 Word2Vec 模型，以獲得中文詞向量，接著再決定各中文成語在詞向量空間中的相似程度，以此作為推薦中文同義成語的依據。亦使用中文成語例句語料庫來訓練 LSTM 與 BiLSTM 模型，然後以此模型來推薦適合用於中文寫作時的成語。使用 LSTM 和 BiLSTM 模型推薦文句中適合的成語，當推薦一個

成語時其正確度甚至可以達到 50%。可見當輸入足夠的資料後深度學習模型確實很合適用來作自然語言處理。另外也發現此結果雖已比隨機猜測高出許多，但依然無法達到與人工推薦媲美，自然語言處理有一些固有的問題，譬如文句中語意的模糊性問題，所以一個句子有時尚無法正確判斷應使用何成語，需有包含上下文的語料來判斷，尤其是本研究所探討的成語部分。所以若有更完整的上下文資訊或許會有更適切的結果。

吳晨皓(2020). 以 BERT 與 GPT-2 分別應用於刑事案件之罪名分類及判決書生成，法學結合人工智慧為刑事案件分類實驗，輸入一段中文長文本案件片段內容，能給出一個案件類型的建議答案，並比較不同類型的預訓練模型影響及利用遷移學習的詞嵌入方式提升準確度。實驗結果顯示，刑事案件的分類任務上，若 BERT 搭配刑事判決書的預訓練模型，相較於 CapsNet-BiGRU、CNN-BiLSTM 等深度學習模型搭配以 Word2Vec 訓練的刑事判決書或中文維基百科之資料集的預訓練詞向量之方法，能更提高模型準確率與泛化能力，進一步提升遷移學習分類罪名的效果。

文句推薦的工作，實際上是自然語言處理領域常見的工作之一。因為文句的推薦，同時需考慮到上下文的訊息，因此適合使用 Recurrent Neural Network (RNN, 循環神經網路)來解決問題。本論文擬在深度學習模型之訓練上，我們採用 LSTM、BiLSTM 之 RNN 模型，以及 BERT(Bidirectional Encoder Representations from Transformers)模型架構，加上常用的機器學習分類器作為 Baseline(基線)方法進行實驗比較，以佛教引用句上下文資訊為訓練資料，設計佛教引用句推薦方法。

三、問題定義

本論文欲研究與探討的問題為如何運用深度學習方法，針對文章寫作進行佛教引用句之推薦。佛教引用句推薦屬於基於內容的推薦任務，我們將其定義為一分類問題。首先從佛學資料中檢索一般常用的佛教引用句作為候選佛教引用句集，擇定可供推薦之佛教引用句選項。方法之輸入為欲進行推薦之文本上下文之內容，方法之輸出為針對該上下文內容所建議之佛教引用句列表，其順序按照相關度自高至低排列。

為建構此推薦任務之分類方法，我們依前項佛教引用句蒐集其所含上下文之序列資料，將作為標準答案之標準佛教引用句及其上下文內容配對成為資料集，將資料集分割為訓練資料及測試資料，訓練資料用以訓練模型、測試資料用以測試、評估模型準確度，根據深度學習模型匹配的功能機率分數排名候選佛教引用句為推薦依據。

在本節中，我們介紹本論文中使用的詞彙概念：

(一)佛教引用句

是一組詞彙序列，使用在佛教引用句推薦任務裡面。

(二)上下文

在包含佛教引用句的文章中，詞組在佛教引用句之前發生的被命名為上文和佛教引用句後出現的詞組被命名為下文。佛教引用句的詞組是連接上文和下文。

(三)查詢的上下文

內容是由使用者提供，供推薦系統據以推薦相應的佛教引用句。

(四)候選佛教引用句

是可能與查詢上下文相關。在這項研究中，數據集中的所有佛教引用句都是視為候選佛教引用句，整組候選佛教引用句構成一個候選集。

(五)標準佛教引用句

標準(Golden Standard)佛教引用句屬於候選佛教引用句，在查詢上下文中是最佳答案。在這項研究中，查詢上下文的標準佛教引用句是與查詢上下文一起出現在真實文章中。因為它幾乎兩個完全相同的上下文不可能同時出現兩個不同的引用，我們假設每個查詢上下文都有一個並且只有一個標準佛教引用句。

(六)佛教引用句推薦

給定一組查詢上下文和一組候選佛教引用句，每次查詢上下文，系統要求返回佛教引用句的排名列表。



四、研究方法

本文提出一個基於深度學習(Deep Learning)的佛教引用句推薦方法，本方法將以 RNN 之 LSTM、BiLSTM 與 BERT 模型實驗，並採用 TensorFlow 為後端之 Keras 套件如圖 1 進行實作，且與基於傳統機器學習分類器之基線(Baselines)方法進行比較。

我們以 TensorFlow(是由 Google 開發)為後端引擎之 Keras 套件進行實作，Keras 是一種高階深度學習的程式庫，屬於開放原始碼，能夠運行在 TensorFlow 之上，建立深度學習模型，並進行訓練、評估準確性，且進行預測，如圖 1 所示。

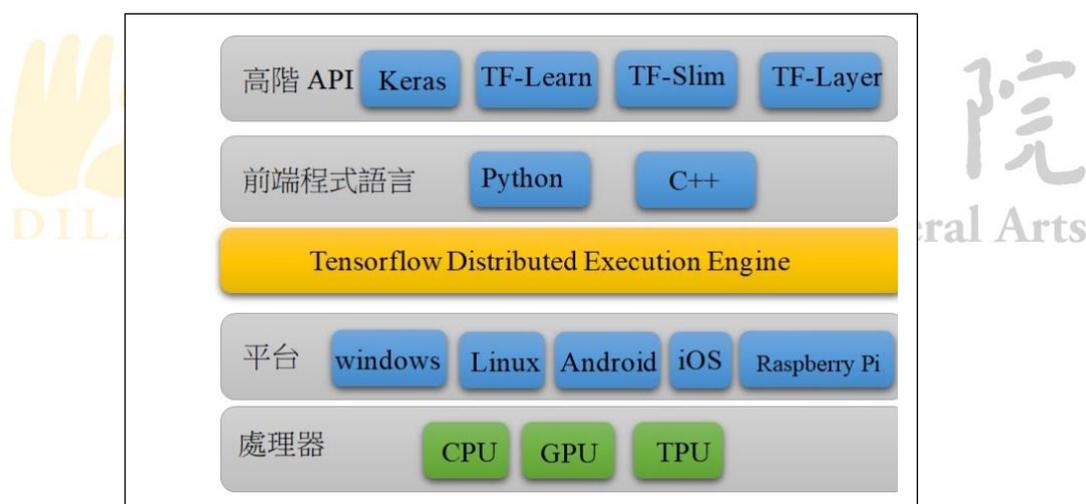


圖 1 TensorFlow 架構圖

(圖片來源Tensorflow+keras 深度學習人工智慧實務¹)

研究步驟詳述如下。

¹ <https://3.bp.blogspot.com/-NbdeXse6khhk/WXsDLTZO6-I/AAAAAAAAAb4/vRWXbbr1FAUdNaehcmYCTxOMgmMe8rvewCLcBGAs/s1600/image045.tif> (存取日期 2022/07/19)

(一) 資料搜集及清理

1 步驟一：候選佛教引用句清單

我們選取常見的、通俗的佛教引用句，共分為 18 類，並將每一類編制標籤 0~17 稱為結果標籤，在實驗時用以代表該候選佛教引用句，如表 1 所示，以字典形式儲存為候選佛教引用句。

表 1 候選佛教引用句與結果標籤對應表

編號	佛教引用句	文章數
0	一切有為法，如夢幻泡影	590
1	凡所有相，皆是虛妄	775
2	若真修道人，不見世間過	329
3	應無所住，而生其心	923
4	過去心不可得，現在心不可得	325
5	色不異空，空不異色	600
6	不生不滅，不垢不淨	445
7	不取於相，如如不動	328
8	如來者，無所從來	164
9	無有定法，如來可說	163
10	無有恐怖，遠離顛倒夢想	236
11	法尚應捨，何況非法	290
12	本來無一物，何處惹塵埃	518
13	一念悟時，眾生是佛	65
14	何期自性，本自清淨	182
15	不是風動，仁者心動	25
16	如來是真語者，實語者	164
17	是法平等，無有高下	243

搜尋常見的、通俗的佛教引用句，尚須考慮是否能搜尋到相當數量的上下文，必須有所選擇。因為經過參考淨行居士編輯《佛教引用句類編》如圖 2 所

示，雖然編列各類型佛教引用句然而少見通俗、可朗朗上口的名句，且缺乏相對的上下文資訊如圖 3 所示。



圖 2 佛教引用句類編

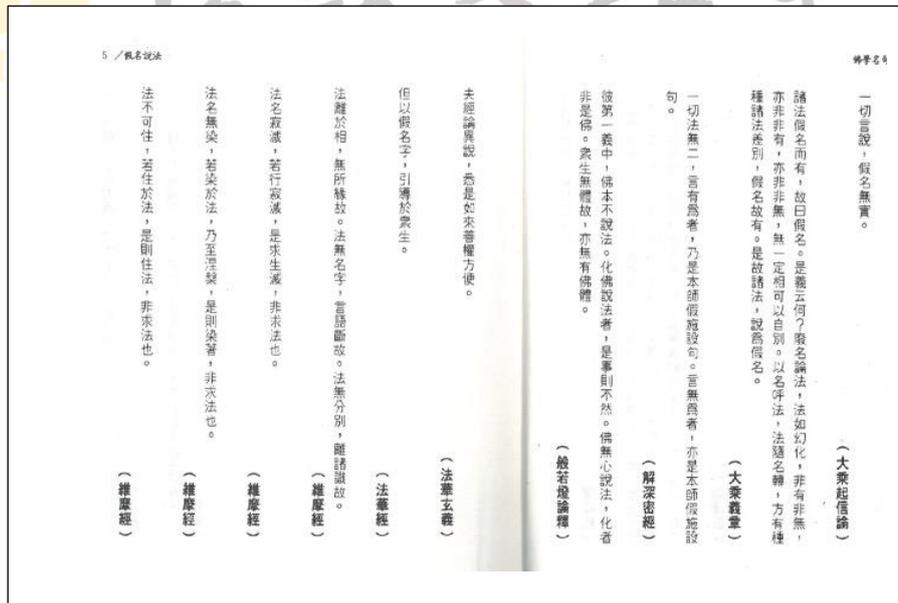


圖 3 句類編內容

2 步驟二：佛教引用句的上下文

候選佛教引用句字典編製完成後，開始撰寫爬蟲程式，依前項佛教引用句為鍵值，也就是標準佛教引用句，爬取各網站(如表 2 網站網址所列)文章資料，當檢索到所需資料如同圖 4 所示方式搜尋擷取所需要的文章資料。



圖 4 網頁搜尋文章範例

(1) 爬蟲技術(Web Crawler)

也稱為網路蜘蛛，主要是透過 URL(Uniform resource locator)，在遵循一定規則下，抓取網路訊息。工作原理如下：

- I. 首先設定爬取內容與方案。
- II. 由所設定的 URL 開始工作，如果獲得相關有用的訊息則儲存。
- III. 開始工作時可能會發現新的 URL，這時會篩選有用的 URL 放置在 URL 佇列。
- IV. 如果 URL 佇列是空的表示已經滿足此工作。

通常可以透過 requests.get() 取得的網頁內容，這是一個 Response 類型的物件。如果我們想把網頁的內容存到檔案裡面，怎麼做呢？當然我們可以透過 Response

物件的 text 屬性拿到所有的網頁內容，再把它們寫進檔案裡。下載網頁 html 檔案後，再從繁雜的 html 檔案內容中，擷取出我們感興趣的資料，而 BeautifulSoup 模組就是用來處理這件事很好用的工具，BeautifulSoup 解析上述下載之 html 文件時是用相對位置概念，如圖 5 所示，來解析文件內容，然後去除標籤儲存所需要之文字。

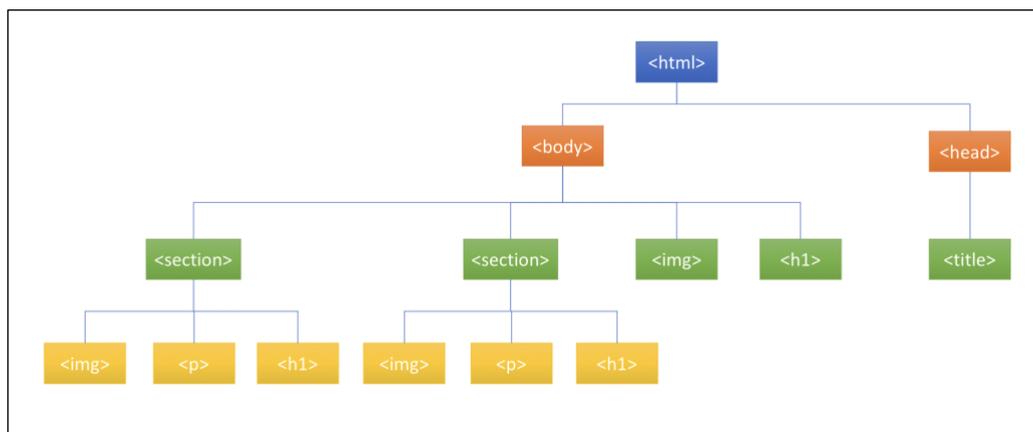


圖 5 html 文件標籤位置圖

(2) 擷取上下文

舉一例如下：從搜尋的文章中擷取含有標準佛教引用句之文章片段，「別人的，別人認不認真與我不相干。所以自己要想成就，首先要肯定，三藏十二部是諸佛菩薩教導我的。3. 我要發心作佛弟子，我要發願接受佛菩薩的教誨，別人學不學與我不相干，這樣你才能真正做到，像六祖惠能大師所說的，「若真修道人，不見世間過」。4. 學佛而不能成就，就是這個地方觀念顛倒了，「佛是教你的，不是教我的」。5. 天天責備別人，「你破戒了，你犯戒了」，完全沒有想到自己，起心動念都是找別人麻煩，這樣造了一身的罪業，最後的結果三途輪轉去了，虧還是」，並將該標準佛教引用句以 X 取代，儲存成檔案並標示其相對應的結果標籤。將該標準佛教引用句以 X 取代的目的，是避免每一筆訓練資料在深度學習模型中訓練的階段先看到有關的答案。

(3) 資料來源網站

資料來源網站有谷歌台灣、法鼓全集網頁版、般若文海網站、中台禪寺資訊網、聯合新聞網、佛光山資訊網、香光資訊網、台灣佛學網等等網頁，如表 2 網站網址所示。透過程式從抓取的文章中搜尋有佛教引用句之位置點，擷取前後約各 100 個字，文章長度大約 200 個字，再將文章以檔案儲存成含上下文的資料集。

表 2 網站網址

網站名稱	網址	擷取數量
Google 台灣	https://www.google.com.tw/	2719
法鼓全集網頁版	http://ccdd.omtf.pl/pc.htm	237
般若文海	https://book.bfn.org/	1325
中台禪寺	https://ctworld.org.tw/index.htm	481
聯合新聞網	https://udndata.com/ndapp/Index?cp=udn	332
佛光山資訊網	https://www.fgs.org.tw/	206
香光資訊網	http://www.gaya.org.tw/query/index.html	127
台灣佛學網	http://big5.xuefo.net/	257

(二) 前置處理

資料送入模型訓練之前，每一筆資料須預先處理，再儲存成訓練資料集。模型之訓練資料其輸入為上下文之語句，輸出為分類結果之標籤。為符合模型所需之資料格式，我們將輸入之上下文語句進行 Jieba 斷詞、Keras tokenization、以及 zero padding；針對輸出標籤則進行 one hot encoding，以下分別詳述前置處理之方法。

1 Jieba

Jieba 是一種中文斷詞工具，主要是透過詞典，在對句子進行斷詞的時候，將句子的每個字與詞典中的詞進行匹配，找到則斷詞，否則無法斷詞。

實際範例如次，我們利用 Jieba 進行中文分詞，經過分詞處理後之段落上下文，如底下字組序列，並將標準佛教引用句以 X 取代，此標準佛教引用句在候選佛教引用句中，如表 1 所示編號為 2：

[別人 認不認真 與 我不相干 所以 自己 要想 成就 首先要 肯定 三藏 十二部 是 諸佛 菩薩 教導 我的 我要 發心 作佛 弟子 我要 發願 接受 佛 菩薩 的 教誨 別人學 不學 與 我不相干 這樣 你 才能 真正 做到 像 六祖 惠能 大師 所 說的 X 學佛 而 不能 成就 就是 這個 地方 觀念 顛 倒了 佛 是 教 你的 不是 教 我的；天天 責備 別人 你 破戒 了 你 犯戒 了 完全 沒有 想到 自己 起心 動念 都是 找別人 麻煩 這樣 造 了一身 的 罪業 最後 的 結果 三途 輪 轉去 虧 還是] Institute of Liberal Arts

此上下文將做為機器學習及 AI 模型的主要資料集來源格式。為了便利後續處理，在此階段我們也會同時統計詞彙總數，並觀察詞彙分佈。

2 Keras Tokenization

Keras 提供 Tokenizer 模組，其功能類似字典，使用此字典將文字轉換成數字。因為在深度學習的模型中只能接受數字，所以必須將「上下文 list」轉換成「數字 list」。使用的步驟：

- I. 建立 Tokenizer 物件。
- II. 使用 `fit_on_texts()` 對要處理的序列資料做字典（對照表），並儲存在 Tokenizer 物件中，可用來對其他資料進行轉換或編碼。

4 One-hot encoding

一種特殊的位元組合，該位元組裏，僅容許單一位元為 1，其他位元都必須為 0。之所以稱為 one-hot 就是因為只能有一個 1 (hot)。若情況相反，只有一個 0，其餘為 1，則稱為 one-cold。在機器學習裏，也有 one-hot 向量 (one-hot vector) 的概念。在一任意維度的向量中，僅有一個維度的值是 1，其餘為 0。譬如向量 [001000000000000000]，即為 18 維空間中的一組 one-hot 向量。將類別性資料轉換成 one-hot 向量的過程則稱 one-hot 編碼 (one-hot encoding)。

在我們進行資料預處理過程中，程式將讀入每一上下文檔案文字序列，首先去除標點符號，然後將內容依所包含的佛教引用句，標示對應的結果標籤，用以作為訓練與驗證之用。參照佛教引用句與標籤對應表，所設定的標籤數字，將以數字 0~17 來代表。而為了要讓這些類別標籤與後續的深度學習訓練相容，我們以 one hot encoding 的方式，將這些類別標籤編碼再轉換為向量表達，讓模型可接受及輸入。

將結果標籤做 one hot encoding
例如「2」的 one hot encoding → [0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

再將所有上下文及相對的結果標籤資料集，以隨機函數打亂資料的排列順序，以便消弭因資料排列而可能產生的訓練誤差，然後將處理好的資料以檔案儲存為資料集。

(三) 模型訓練與預測

資料處理完成並將之儲存成檔案資料集，每一組態的模型訓練及測試資料，都從此資料檔案載入系統模型。確保每一次進行模型訓練及測試時使用相同的資料集。

1 Baseline 方法：機器學習分類器

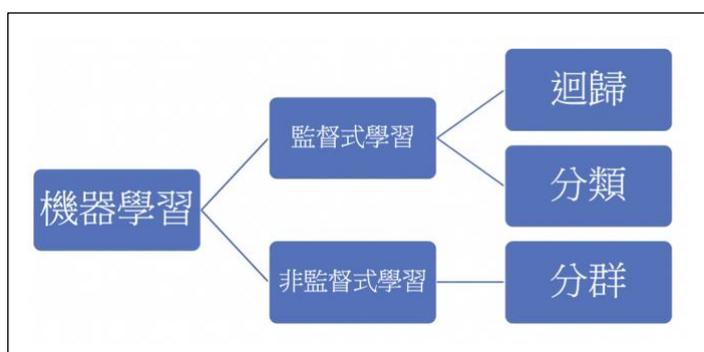


圖 6 機器學習的分支

機器學習的分支：分為監督式學習 (Supervised Learning)與非監督式學習 (Unsupervised Learning)，如圖 6 所示。監督式學習:資料都有相對結果標籤(Label)，這標籤是由人工標註，亦即人工先分類以便讓電腦學習之用，也是資料的答案，在輸出時用來判斷誤差時使用，這種預測比較精準。例如，解決迴歸(Regression)問題與分類(Classification)問題兩種。非監督式學習(Unsupervised Learning):資料沒有標籤，也就是沒有標準答案，機器自己尋找各種資料間的關聯與差異做出判斷。例如分群分析(Cluster Analysis)，常用的演算法是 K 平均分群演算法(K-means)。

在自然語言處理中，對於文章內容的分類方法，我們可以實驗機器學習的不同分類任務中，依照不同的演算法實驗解決相同任務的結果，與我們相關且最常見的有底下三種分類演算法：

(1) K-Nearest Neighbors (KNN) K 近鄰演算法

K-最鄰近的一種識別算法，透過周圍鄰居「Neighbors」來幫忙決定，尋找與這筆資料最近的 K 筆資料，K 值是使用者可以自行預先設定的一個整數，由這 K 筆資料的類別標籤協助判定。

表 3 某班學生數學英文成績

訓練資料序號	數學成績	英文成績	就讀班級
1	92	80	A
2	89	75	A
3	45	98	B
4	60	89	B
5	74	90	B
6	88	50	A
7	98	68	A
8	80	76	B

例如，假設某一個學校 8 名學生其數學、英文成績及其就讀班級，如表 3 所示。今要判定“紅色星號”這位同學，是屬於哪一班，在運用 KNN 方法中，如圖 7，將 K 值設為 3，由 A 班與 B 班中學生的英語與數學分數分佈中，判定“紅色星號”同學是屬於 A 班或 B 班，如圖 7 中圈起來之區域可清楚看出與星號位置最近的三筆資料分別是 A、A、B，按照 KNN 演算法的規則，3 個鄰居的最多數分類是 A，所以該位同學“紅色星號”就判定為來自於 A 班。

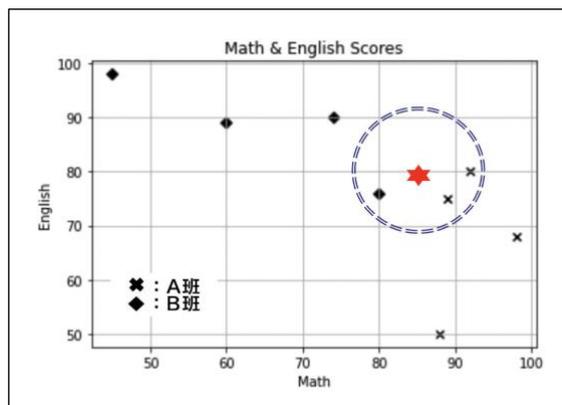


圖 7 KNN 之 K=3 分類法(圖片出處：李宏毅教授)

(2) Decision Tree (DT)決策樹

決策樹分類的分法是透過不斷進行二元分類來決定輸入最後對應的輸出，樹的中間節點 (Non-Leaf Nodes) 代表測試的條件，樹的分支 (Branches) 代表條件測試的結果，而樹的葉節點 (Leaf Nodes) 則代表分類後所得到的分類標記，也就是表示分類的結果。決策樹是應用廣泛的分類與預測工具，其是以樹狀圖為基礎的方法，具有讓使用者容易理解的優點。要建構決策樹，即需從資料中依序挑選適於用以分類的特徵，自根節點開始一步步建立中間節點，逐步建構完整的決策樹用以分類。在特徵的挑選上，一般會以資訊增益 (Information Gain) 作為特徵的選擇依據。決策樹學習之演算法會計算每一個特徵用以分類時所得到的資訊增益，優先挑選出可得到最大資訊增益的特徵作為根節點，依序挑選資訊增益次大者作為中間節點，如此重覆直至所有的資料皆能完成分類，即建構出完整之決策。圖 8 為決策樹之範例。

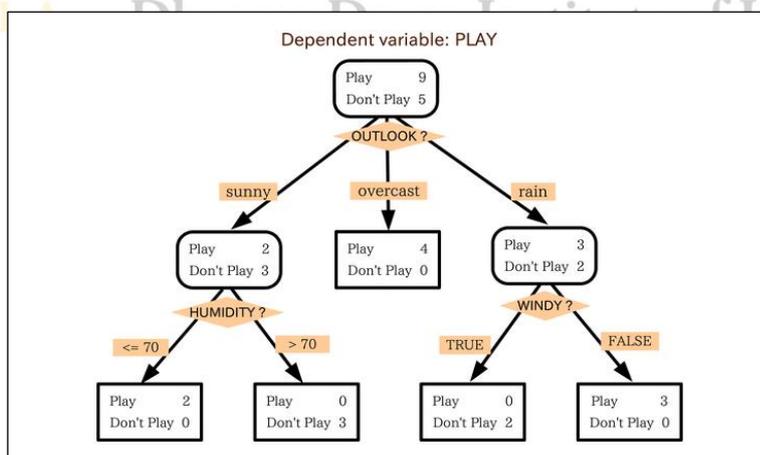


圖 8 決策樹分類法 (圖片出處：

https://commons.wikimedia.org/wiki/File:Decision_tree_model.png)

(3) Support Vector Machine (SVM) 支持向量機

支持向量機是一組用於分類、回歸和異常值檢測的監督學習方法。SVM 的分類方法是在類別資料間找出最佳的決策邊界(Decision boundaries)。它可以輕鬆處理多個連續和分類變量。SVM 可以在多維空間中構造一個超平面來分離不同的類，以迭代方式生成最優超平面，用於最小化錯誤，其核心思想是找到一個最大邊際超平面，將數據集最好地劃分為類，如果資料是二維的，超曲面將是一條曲線，如圖 9 所示。

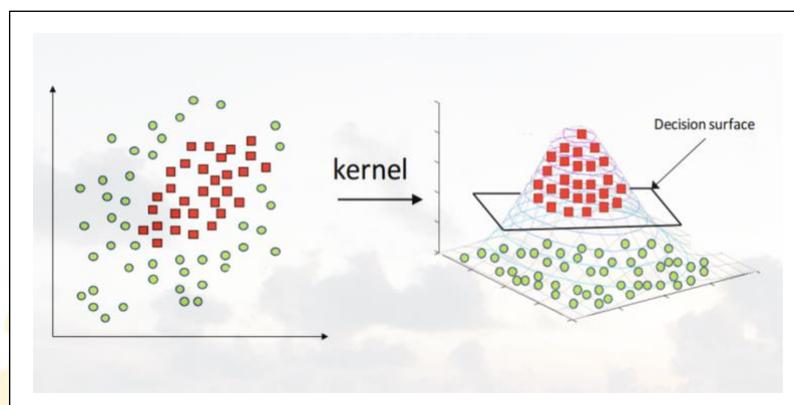


圖 9 SVM 決策邊界超平面

(圖片出處：<https://bbs.huaweicloud.com/blogs/252776>)

瞭解了機器學習分類器 KNN、DT、SVM 的理論背景，實作上我們將以 Python 程式語言引用 Scikit-learn 的分類器 (classifier) 以構建其實驗。模型框架如圖 10 所示，第一步：讀取我們前面已經前置處理完成並儲存的資料集，第二步：將讀入之每一筆資料文本中的詞語轉換為詞頻矩陣，矩陣元素 $a[i][j]$ 表示 j 詞在 i 類文本下的詞頻，並統計每個詞的 TF-IDF 權值。第三步：資料切分是分配訓練資料 80% 與測試資料 20% 比例。第四步：機器學習分類，我們使用 SkLearn 套用 KNN、DT、SVM 的 model 來預測佛教引用句資料集。

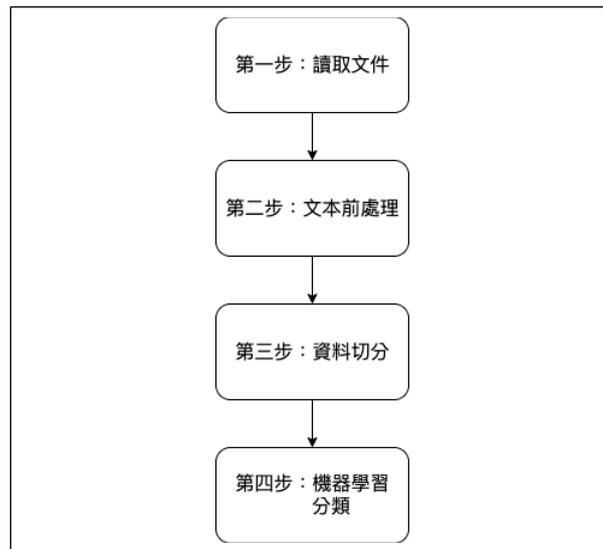


圖 10 機器學習分類器模型架構

2 深度學習模型

(1) RNN

Recurrent Neural Network 循環神經網路 (RNN) 是一種神經網路架構，它試圖利用順序輸入數據的信息，處理序列數據的能力對於文本處理相對重要，RNN 的理想目的是對序列中的每個元素執行相同的任務，輸出取決於前一個計算。圖 11 的左側是一個普通的 RNN 單元，右邊是它的展開形式。展開形式顯示完整輸入序列的網路。在圖 11 中， X_t 是時間 t 的輸入， h_t 是時間 t 的隱藏狀態。隱藏狀態就像網路的“記憶”。

h_t 是根據先前的隱藏狀態和當前步驟的輸入計算得出的，如：

$h_t = f(U_{X_{t-1}} + W_{X_t})$ 。U 和 W 是矩陣，合併當前輸入 X_t 和隱藏狀態 h_t 的影響。函數

f 是非線性的函數，例如 tanh 或 sigmoid。 y_t 是時間 t 的輸出， y_t 可以計算為： y_t

$= f(V_s)$ 。由於循環結構，RNN 可以處理任意輸入的長度，隱藏狀態使 RNN 能夠捕獲輸入序列的依賴關係。

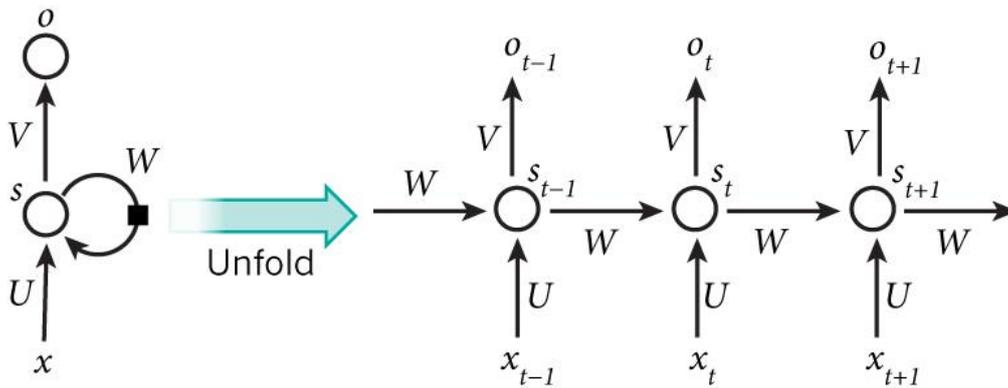


圖 11 RNN 單元及其展開形式

(圖片來源 <https://algotech.netlify.app/blog/text-lstm/>)

sigmoid function (S 型函數或稱乙狀函數) 具有特徵 “S” 形曲線或 S 形曲線的數學函數，因其函數圖像形狀像字母 **S** 得名，如圖 12 所示。其形狀曲線至少有 2 個焦點，也叫「二焦點曲線函數」。S 型函數是有界、可微的實函數，在實數範圍內均有取值，且導數恆為非負，有且只有一個拐點。S 型函數和 S 型曲線指的是同一事物。在一些學科領域，特別是人工神經網路中，S 型函數通常特指邏輯斯諦函數(logistic function)。

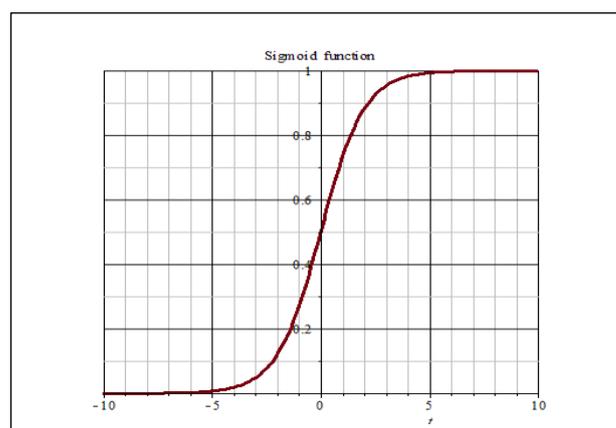


圖 12 Sigmoid function

(圖片來源 <https://www.twblogs.net/a/5b9fe7b92b71771a4da844c1?lang=zh-cn>)

神經網路 (Neural Network, NN)，在機器學習領域，是一種模仿生物神經網路的結構和功能的數學模型或計算模型，用於對函數進行估計或近似。神經網路由大量的人工神經元聯結進行計算。大多數情況下人工神經網路能在外界信息的基礎上改變內部結構，是一種自適應系統，通俗地講就是具備學習功能。現代神經網路是一種非線性統計性數據建模工具，神經網路通常是通過一個基於數學統計學類型的學習方法 (Learning Method) 得以優化，所以也是數學統計學方法的一種實際應用，通過統計學的標準數學方法我們能夠得到大量的可以用函數來表達的局部結構空間，另一方面在人工智慧學的人工感知領域，我們通過數學統計學的應用可以來做人工感知方面的決定問題，也就是說通過統計學的方法，人工神經網路能夠類似人一樣具有簡單的決定能力和簡單的判斷能力，這種方法比起正式的邏輯學推理演算更具有優勢。

(2) LSTM 與 BiLSTM

Long-Short term Memory (LSTM) 長期短期記憶，雖然 RNN 可以接受任意長度的輸入序列，但它通常無法捕獲依賴關係太長的輸入序列，太長的輸入序列也會導致梯度消失問題 (Vanishing gradient problem) 這一難題也是一種機器學習中的難題，出現在用梯度下降法以及反向傳播訓練的人工神經網路。在每一次訓練之迭代中，神經網路權重的更新值與誤差函數的偏導數成比例，然而在某些情形之下，梯度值可能會幾乎消失，致使權重沒辦法得到有效之更新，甚至可能導致神經網路完全無法繼續訓練。

LSTM 可以更好地解決難以捕捉長期依賴和梯度消失，已成功應用於許多任務，並取得了最先進的結果。其核心思想正是在循環神經單元中引入控制數據的閘門 (Gate)，閘門可以保持一個值如果閘門為 1，則為門控層，如果閘門為 0，則此值為零，特別是，正在使用三個閘門來控制是否忘記當前單元值 (忘記門 Forget Gate)，

無論是應該讀取它的輸入（輸入門 Input Gate）以及是否輸出新的單元值（輸出門 Output Gate）。在圖 13 中顯示了一個長短期記憶單元架構。

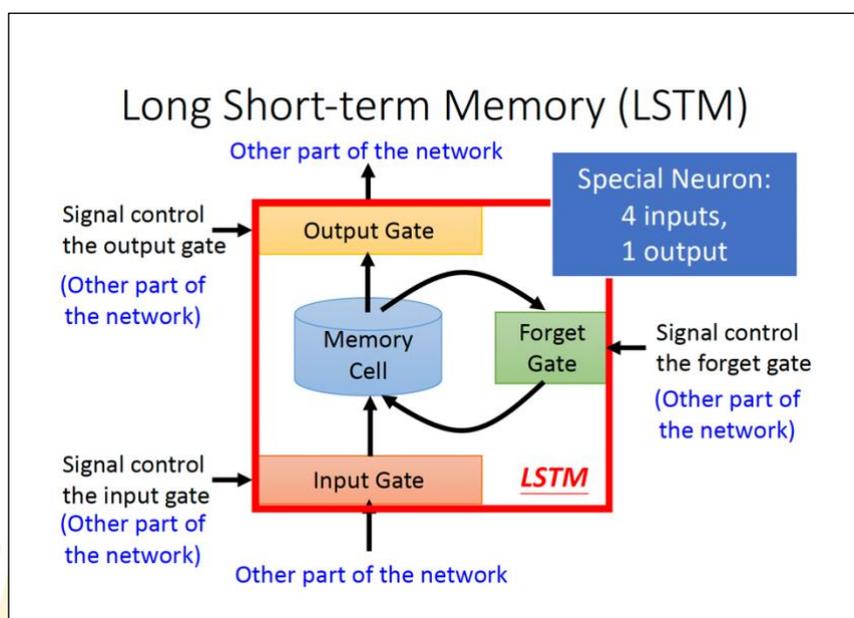


圖 13 長短期記憶單元(LSTM cell) 架構

（圖片出處：李宏毅教授）

我們採用 LSTM 與 BiLSTM 模型，其模型架構如圖 14 所示。

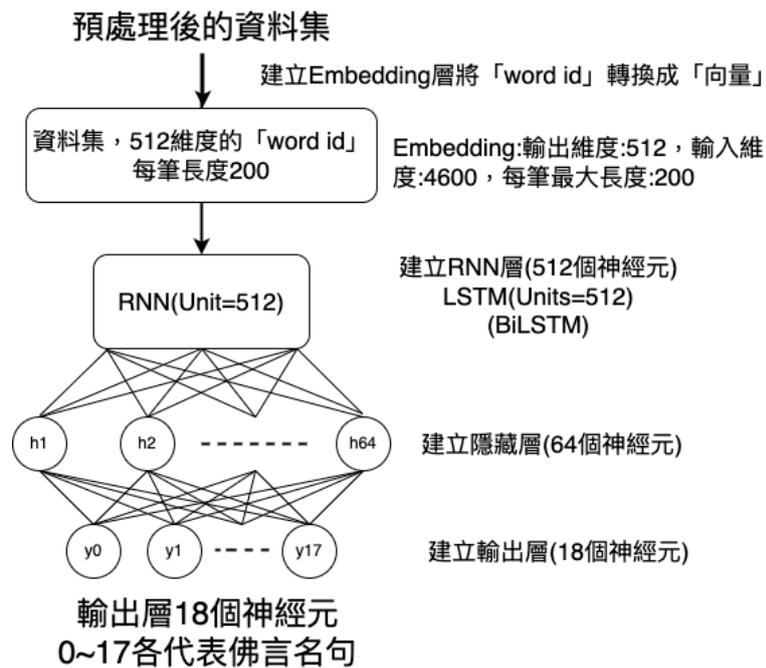


圖 14 LSTM 模型架構及參數

詞嵌入(word embeddings)是一種將文字映射到多維空間向量的語言處理技術，以空間向量方式模擬人類文字的語意。在本文實驗中，將「上下文」資料轉換為數字，可是數字無法與文字語意關聯，以轉換為向量的方式建立與文字的關聯性。

(3) BiLSTM

Bidirection Long-Shot Term Memory (BiLSTM) 雙向長短期記憶，雙向 LSTM 或 BiLSTM 是一種序列處理模型，由兩個 LSTM 組成，如圖 15 中右圖 BiLSTM 架構：一個在前向接收輸入，另一個在後向接收輸入。BiLSTM 主要功能為增加網路可用的信息量，改善上下文對演算法的可用性（例如，知道句子中緊跟的字詞和哪些單詞在前），相較於圖 15 中左圖 LSTM 架構。

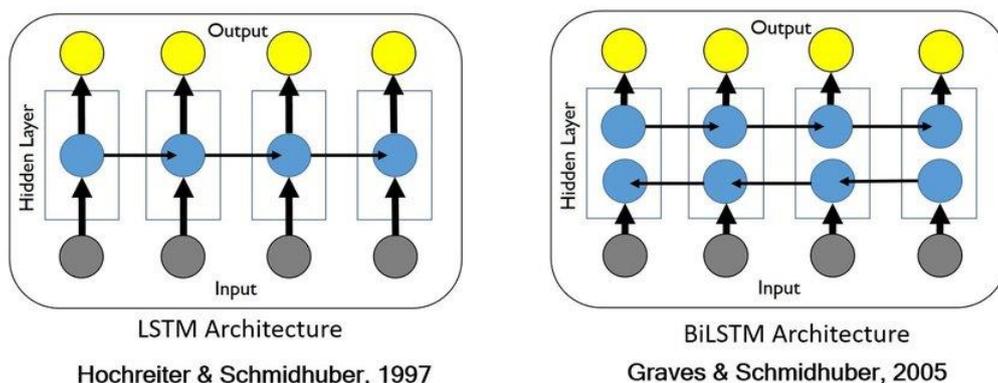


圖 15 LSTM and BiLSTM 架構比對

(4) BERT

BERT 全名為 **Bidirectional Encoder Representations from Transformers**，是美國科技公司 Google 於 2018 年所提出，已經成為近年來 NLP 在語言處理上必備的工具。在 Keras 亦可使用的 keras-bert 套件，可以直接讓我們在 Keras 裡面導入 BERT 模型。其利用大量無標註的文本以無監督的方式，訓練成的語言代表模型，其架構為 Transformer 中的 Encoder，如圖 16 所示。

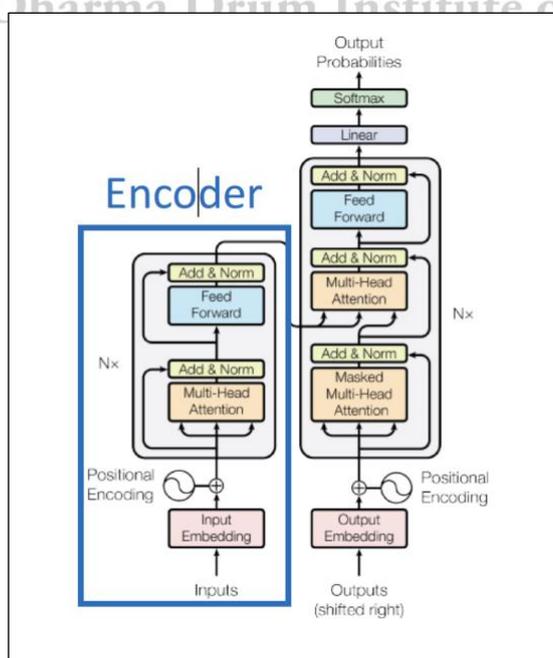


圖 16 Transformer

圖片出處：Attention is all you need(Google 2017)

一般詞嵌入中與上下文無關模型（如 word2vec 或 GloVe）為詞彙表中的每個單詞生成一個詞向量表示，因此容易出現單詞的歧義問題。BERT 考慮到單詞出現時的上下文。例如，詞「香」的 word2vec 詞向量在「我在廟裡拿著香拜拜」和「這房間的氣味很香」是相同的，但 BERT 根據上下文的不同提供不同的詞向量，詞向量與句子表達的句意有關，如圖 17 所示。

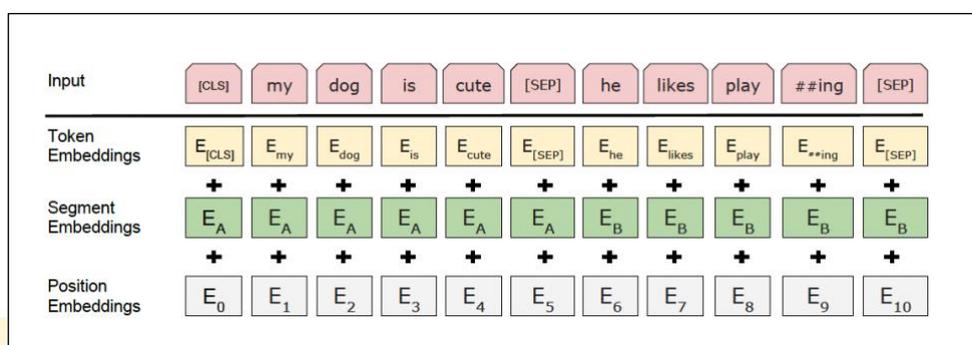


圖 17 BERT Embedding 詞向量

圖片出處：Attention is all you need(Google 2017)

BERT 模型框架如圖 18 所示，使用 BERT 一個很重要的概念就是「遷移學習」Transfer Learning，由於訓練 BERT 模型的成本相當高，一般應用之方式為取用別人預先已經訓練好的模型，將之遷移用在我們想要解決的問題之上。因此，模型訓練前我們需要來載入別人已經預先訓練好的預訓練模型。在 keras-bert 裡面已經提供了一些預訓練的 BERT 模型，可以直接載入。

我們必須一併載入進來字典與 Tokenizer，預訓練 BERT 模型在訓練時已經有將每一個字指定其對應的代號，建構出該模型的字典。keras-bert 裡面提供了一個 Tokenizer 的工具，它可以利用 BERT 的字典，建構出可以進行自動分詞的工具，也可以直接把文字轉換成對應的數字編碼，在轉換成向量輸入模型，如圖 18 示。

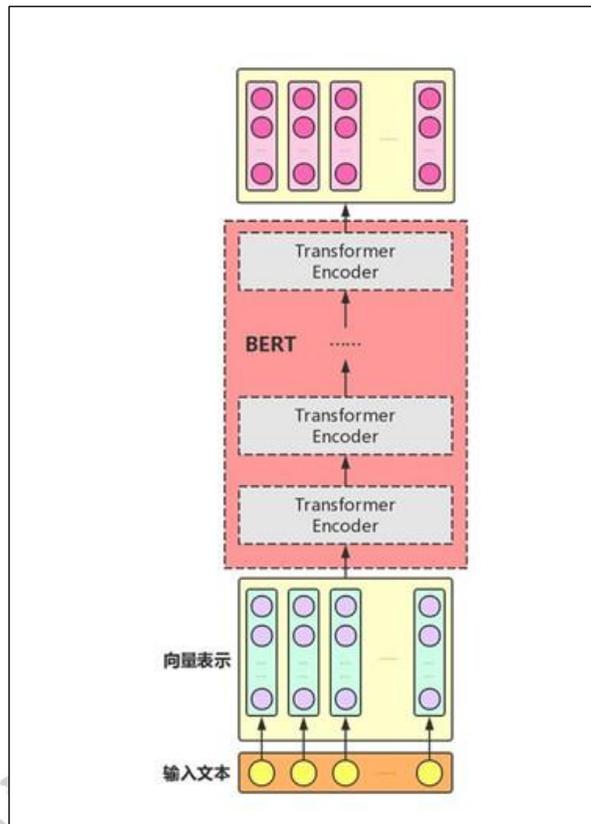


圖 18 BERT 模型輸入輸出略圖

(圖片出處：<https://www.cnblogs.com/gczz/p/11785930.html>)

3 訓練及預測

資料集裡經過分詞的段落文本經過 tokenizer 轉換為編碼做為模型之輸入，其最大之輸入長度為 200 字，而後接上詞嵌入層與 RNN 層，學習文本段落上下文之資訊，最後接上全連接層做為分類器之輸出，分類之預測結果做為佛教引用句推薦排序之依據。

(四) 實驗設計與結果

先以機器學習分類器實驗分類效果，再設計針對 LSTM、BiLSTM 及 BERT 各深度學習模型逐步進行實驗。在實驗評估上，我們將資料集依比例切分為訓練資料集與測試資料集，並採用十折交叉驗證(10-fold Cross-Validation)方法進行實驗。

所謂 K 折交叉驗證(k-fold cross validation)，是一種重復採樣過程，用於對有限的數據樣本上以評估機器學習模型。稱為 k 的參數，它指的是給定的數據樣本將被分成的組數。因此，通常稱為 k 折交叉驗證，K 值的設定可以由我們自己視需要而調整。

當選擇一個特定的 k 值時，它可以用來代替模型參考中的 k，例如 k=10 成為 10 折交叉驗證。以圖 19 為例：假設我們設定 K=10 成為 10 折交叉驗證，也就是將訓練集切割為十等份。亦即相同的模型要訓練十次，每一次的訓練都會在這 10 等份挑選其中 9 等份作為訓練資料，剩下 1 等份未參與訓練將其作為驗證集。

因此訓練 10 次後會有 10 個不同驗證集的 Error，此處的 Error 通常稱作 loss 也就是模型評估方式，再把這 10 次的 loss 加總起來取平均值當成最終的結果。



圖 19 K 折交叉驗證範例

(圖片來源:iT 邦幫忙 <https://ithelp.ithome.com.tw/articles/10279240>)

1 Baseline 方法：機器學習分類器

在 Baseline 部分，分別以常見的 KNN、DT 及 SVM 分類器實驗，並以 python 程式語言引用 Scikit-learn API 實作。

KNN、DT、SVM 模型經由相同訓練資料、測試資料（分割比例 80/20）實驗結果如下表 4 所示，KNN 模型約 63% 準確度、DT 模型約 77% 準確度、SVM 模型約 79.43% 準確度，相較之下 SVM 模型比其他兩中分類器有較高的準確度。

表 4 分類器模型準確度比較

分類器	KNN	DT	SVM
模型準確度%	63	77	79.43

從 SVM 模型以測試資料 1264 筆預測分類結果中，比較其推薦 1 句 (Top1) 佛教引用句的準確度 79.43%、推薦 3 句 (Top3) 佛教引用句的準確度 89.48%、推薦 5 句 (Top5) 佛教引用句的準確度 94.22%，及其個別的預測錯誤數，如表 5 所示。

表 5 Top1 Top3 Top5 推薦準確度及錯誤數

Top1		Top3		Top5	
ACC	errors	ACC	errors	ACC	errors
79.43%	260	89.48%	133	94.22%	73

2 深度學習模型：LSTM 及 BiLSTM

首先，我們對 LSTM 及 BiLSTM 模型在相同條件下做訓練、評估及預測，再針對兩者結果相互比較。

(1) 設定固定參數

在模型實驗中所設定的固定參數如表 6 所示：

表 6 模型固定參數

固定參數	設定
Optimizer(優化器)	adam
Loss(損失函數)	Categorical_crossentropy
Dense activation(啟動函數)	softmax
num_words 最大總詞彙數	4600(文章斷詞後編入詞庫的 詞彙數)
Maxlen 文章最大詞彙數	200
Units 分類結果數	18
Batch_size(批次處理數量)	32
Epochs(迭代次數)	20
隱藏層 Units	64

(2) 實驗中操縱變因

嘗試調整如下兩個參數，實驗出最佳典範。

I. 詞嵌入向量維度

如圖 20 中紅色字所示，調整 Embedding 輸出維度。此參數代表一個字詞用多少維度去表示，如果太小或太少，很多訊息沒法表達，資訊會 loss，太多太大則參數太多資訊會太分散，實驗至合適參數數字。

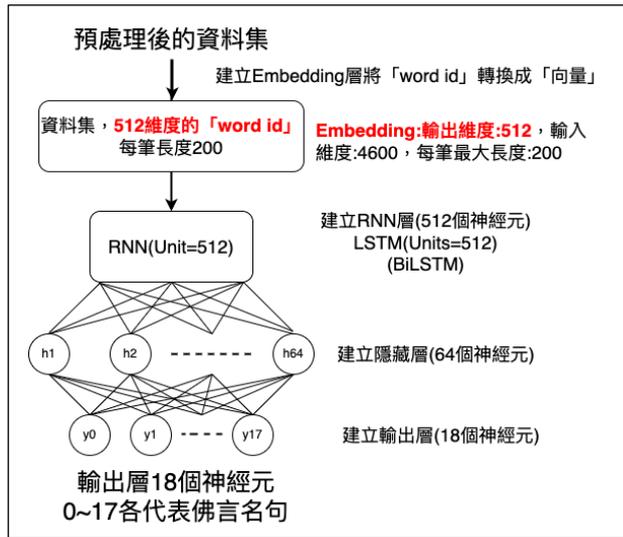


圖 20 Embedding 輸出維度

II. LSTM、BiLSTM 輸出維度

如圖 21 中藍色字所示，調整 LSTM、BiLSTM 輸出維度。此參數表示 RNN units 有多強的運算能力解析度，以解析前後文字，資料集數目不夠多時，Units 太大會有 Overfitting 情形，資訊會發散，且消耗資源愈大。

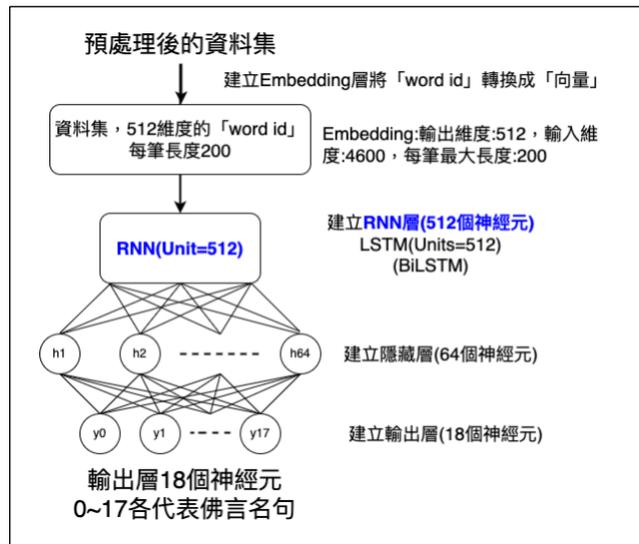


圖 21 RNN 輸出維度

(3) 實驗結果及分析

實驗結果及分析部分，使用 LSTM 及 BiLSTM 深度學習模型實驗中，我們交叉調整 LSTM、BiLSTM 輸出向量維度與詞嵌入向量維度，觀察在不同的設定下，相關兩種模型的表現，實驗結果如表 7 與表 8。

在表 7 中，主要是利用 LSTM 所建構的 AI 模型的效果彙整，根據表 7，我們觀察到 LSTM 輸出向量維度（簡稱 L 維度）32 時，提升詞嵌入向量維度（簡稱 O 維度）有助於提高平均準確率，L 維度 64 時，提升 O 維度亦有助於提高平均準確率，而 L 維度 512、O 維度 768 以上，平均準確率下降。再者，O 維度 512 的平均準確率大於，O 維度 768 的平均準確率。最優的平均準確率 81.87% 在 LSTM 輸出向量維度 512、詞嵌入向量維度 512。

表 7 LSTM 模型及參數實驗結果

O 維度 L 維度	32	64	128	256	512	768
32	70.55	71.26	73.65	73.16	75.88	72.86
64	72.00	75.57	76.05	77.17	78.18	78.52
128	73.79	74.94	76.24	77.86	79.25	80.31
256	72.78	77.12	78.22	77.86	78.98	79.70
512	76.29	78.84	78.92	79.06	81.87	81.75

表 8 BiLSTM 模型及參數實驗結果

O 維度 L 維度	32	64	128	256	512	768
32	70.63	73.44	73.51	73.38	74.26	73.47
64	71.67	75.30	75.55	75.96	76.24	76.71
128	70.71	75.19	76.08	75.88	77.64	79.58
256	71.15	74.40	77.23	78.38	80.16	81.24
512	67.23	75.19	78.68	78.44	81.68	80.97

綜合來說，藉由比較 LSTM、BiLSTM 兩組模型實驗結果及兩組模型相同參數實驗，如表 9 所示，比較結果分析：

兩組模型當詞嵌入向量維度及 LSTM、BiLSTM 輸出向量維度逐漸提高時，平均準確率亦逐漸提高趨勢。

在詞向量維度調整至 512、LSTM、BiLSTM 輸出向量為 512 時，兩者的平均準確率皆最高，LSTM 的 81.87% 高於 BiLSTM 的 81.68%，BiLSTM 模型於提升準確率並無顯著效益。

表 9 兩組模型相同參數實驗比較

模型	嵌入向量維度	LSTM 輸出維度	平均準確率
LSTM	512	512	81.87
BiLSTM	512	512	81.68
LSTM	512	768	81.75
BiLSTM	512	768	80.97

預測錯誤分析：再以 LSTM 測試結果分析其錯誤數及錯誤率如表 10 所示。我們做了相關實驗之後，結果還是有預測的錯誤，根據分析的結果，模組的錯誤率大概是 17.34%，不過，還是要從錯誤中去學習，17.34%錯誤來自於 1104 筆 case，由圖 22 裡顯示的就是每一個不同佛教引用句的錯誤數（橘色部分），藍色是他個別的總數。

表 10 LSTM 預測錯誤數分析

標籤	0	1	2	3	4	5	6	7	8
總數	590	775	329	923	325	600	445	328	164
錯誤數	107	158	40	157	10	48	25	80	22
錯誤率%	18	20	12	17	03	08	06	24	13
標籤	9	10	11	12	13	14	15	16	17
總數	163	236	290	518	65	182	25	164	243
錯誤數	44	26	114	111	30	44	9	11	68
錯誤率%	27	11	39	21	46	24	36	07	28



圖 22 LSTM 預測錯誤數

分析預測錯誤原因

I. 佛教引用句 18 類中，類別資料部分意義相近情形：

在各類別資料中如標籤 0、標籤 1、標籤 3、標籤 7、標籤 12 資料集中會有資料意義相近情形，如表 11 所示，使錯誤率普遍提升。列舉如：

表 11 預測資料中意義相近比較

標籤	0	1	2	3	4	5	6	7	8
總數	590	775	329	923	325	600	445	328	164
錯誤數	107	158	40	157	10	48	25	80	22
錯誤率%	18	20	12	17	03	08	06	24	13
標籤	9	10	11	12	13	14	15	16	17
總數	163	236	290	518	65	182	25	164	243
錯誤數	44	26	114	111	30	44	9	11	68
錯誤率%	27	11	39	21	46	24	36	07	28

標籤 0 代表「一切有為法，如夢幻泡影」，「一切有為法」是世出世間一切諸法包括起心動念的心，心也是有為法。夢如幻、泡、影三個字，是形容夢之不實。

標籤 1 代表「凡所有相，皆是虛妄」，凡所有相，就是一切有為法，虛妄的意思，是暫時的、臨時的，如夢幻、泡、影。

標籤 3「應無所住，而生其心」，應無所住，「無住」妙旨，全在「不取於相，如如不動」上，因為凡所有相皆是虛妄！把為什麼「不取於相，如如不動」的真相說出來。

標籤 7「不取於相，如如不動」：我們今天見色聞聲，六根接觸六塵境界就執著、就著相，外不著相，內不動心，他那個心就叫清淨心。

標籤 12「本來無一物，何處惹塵埃」，「本來無一物」就是無念，心有所住，就是妄念，六塵競起，如能一念不生，覺性現前，便是標籤 3 代表的「應無所住而生其心」。

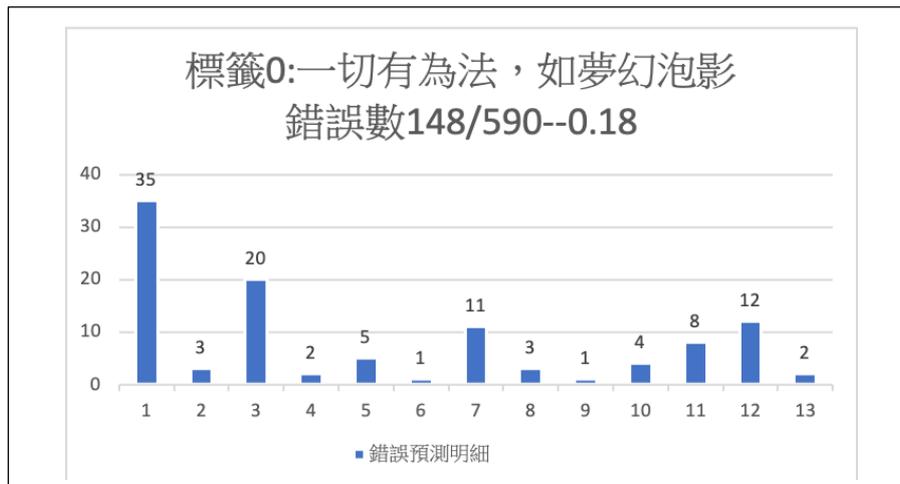


圖 23 標籤 0 錯誤數分析

由圖 23 中可以看出標籤 0「一切有為法，如夢幻泡影」的預測錯誤數分布在標籤 1、3、7、12 的比例相對較高。

範例一：「我們曾經被人家傷害過，或我們傷害過別人，或是我們自己傷害過自己，但過去都已經過去，金剛經說：X，我們就不要一直拿過去不如意的事情來懲罰自己，如果一直想到過去不如意的事情，這就是意根緣法塵，所以，在此提供大家人生幸福之道，不要拿自己的過失來懲罰自己，不要拿別人的過失來懲罰自己。」本段文章中 X 原本目標標籤是 0 代表「一切有為法，如夢幻泡影」，但是預測出的標籤是 1 代表「凡所有相，皆是虛妄」。

相對的由圖 24 中可以看出標籤 1「凡所有相，皆是虛妄」的預測錯誤數分布在標籤 0、3、7、12 的比例也相對較高。

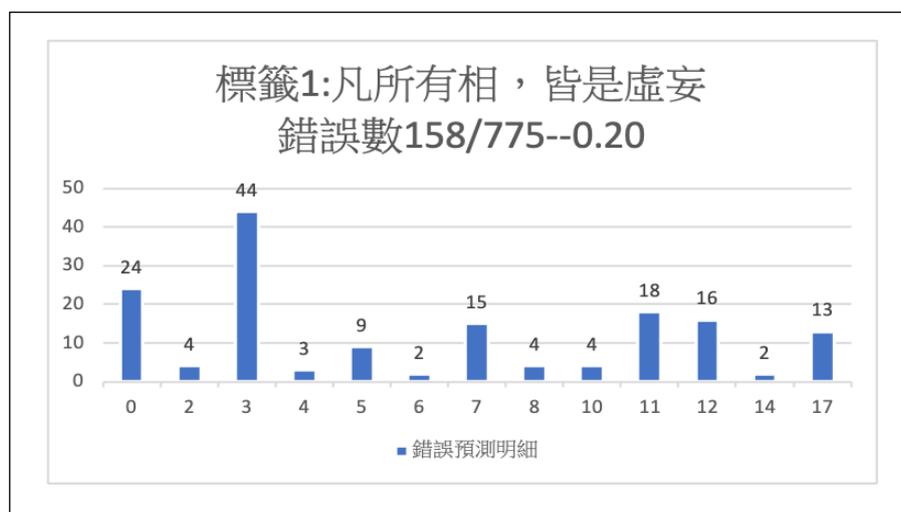


圖 24 標籤 1 錯誤數分析

範例二：「體悟到的般若智慧有多少，時時觀照空性，個性積極 的沈家楨居士，過去對於金剛經所說的 X 產生疑惑，如果大家都覺得世界上的一切都是虛妄不真，沒有永久性的 存在和價值，那麼人為什麼要努力、要奮鬥、要爭取上 進呢？直到再看金剛經時恍然大悟，為什麼佛陀在大徹大悟後，仍和常人一樣過著衣持鉢入城乞食的生活，他發現之前是理論上去認識金剛經，而把空錯誤解讀成消極態度」本段文章中 X 原本目標標籤是 1（凡所有相，皆是虛妄），但是預測出的標籤是 3（應無所住，而生其心）。

II. 訓練資料量少，模型訓練不足。

在 LSTM/BiLSTM 模型中，可能因為蒐集到可供訓練的資料量較少，以至於模型訓練後，測試資料預測的準確度低，導致錯誤數增加，如表 12 訓練資料量少分析所示，標籤 13 的資料總數 65 筆，錯誤率 46%，標籤 15 的資料總數 25 筆，錯誤率 36%，整體來說這兩類標籤模型預測錯誤率相對較高。分析標籤 13 的錯誤數中，如圖 25 所示，預測的錯誤對象比較分散。

表 12 訓練資料量少分析

標籤	0	1	2	3	4	5	6	7	8
總數	590	775	329	923	325	600	445	328	164
錯誤數	107	158	40	157	10	48	25	80	22
錯誤率%	18	20	12	17	03	08	06	24	13
標籤	9	10	11	12	13	14	15	16	17
總數	163	236	290	518	65	182	25	164	243
錯誤數	44	26	114	111	30	44	9	11	68
錯誤率%	27	11	39	21	46	24	36	07	28

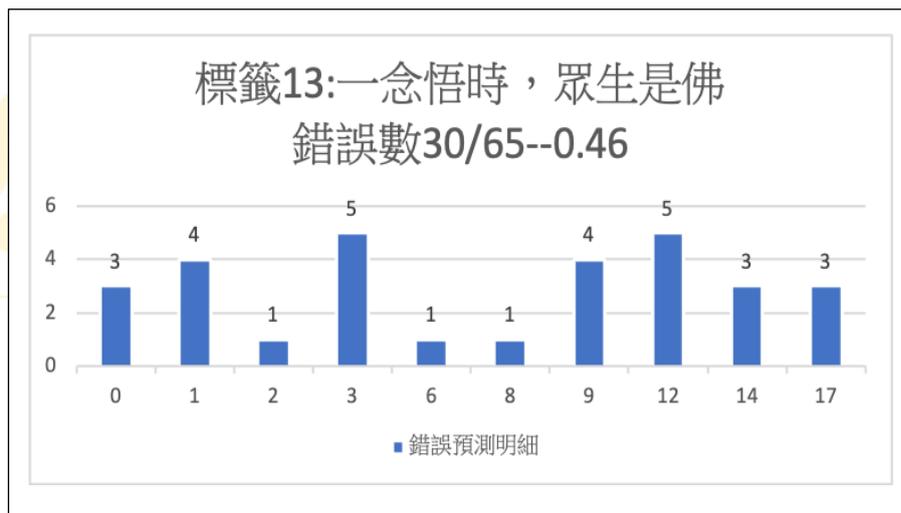


圖 25 標籤 13 錯誤數分析

範例三：「你若有自私心就不清淨，清淨也變成不清淨了，你若沒有自私心，一切都不是為我，自己就會清淨，因為你沒有貪心、沒有瞋心、沒有痴心、沒有我見，這就是悟，所以才說 X 有的人說，喔，所有的人都是佛，不錯，雖說眾生是佛，但是你要悟了，你若不悟，說是佛，那就像一個普通老百姓，自稱國王，這若教國王知道，一定把你抓去放到監獄裡，為什麼呢，你要造反！」本段文章中 X 原

本目標標籤是 13（一念悟時，眾生是佛），但是預測出的標籤是 12（本來無一物，何處惹塵埃）

我們再從 LSTM 模型預測分類結果中觀察推薦 1 句(Top1)佛教引用句的準確度 81.87%、推薦 3 句(Top3)佛教引用句的準確度 90.13%、推薦 5 句(Top5)佛教引用句的準確度 93.40%，及其個別的預測錯誤數，如表 13 所示。

表 13 LSTM Top1/Top3/Top5 推薦準確度及錯誤數

Top1		Top3		Top5	
ACC	errors	ACC	errors	ACC	errors
81.87%	1148	90.13%	628	93.40%	420

3 BERT

為了實作 BERT 模型，首先安裝 keras-bert 套件，接著，載入相關的模組，並設定一些參數，在 keras-bert 裡面已經提供了一些預訓練的 BERT 模型，可以直接載入。並且必須還要一併載入進來的是，預訓練 BERT 模型在訓練時，當然已經有將每一個字指定其對應的代號建構出該模型的字典。

keras-bert 裡提供了一個 Tokenizer 的工具，它可以利用 BERT 的字典，建構出可以進行自動分詞的工具。Tokenizer 也可以呼叫 encode()方法直接把文字轉換成對應的數字編碼。這個方法會回傳兩個清單，第一個是原本語句的每一個 token 的代碼，第二個是 BERT 需要的 segments 清單，這是用來表示語句的該個字是屬於第一句還是屬於第二句的。

接著我們建構出基於 BERT 的佛教引用句推薦模型，我們在 BERT 模型之上加上一個簡單的 Dense 層，用來輸出佛教引用句推薦的數值。整個模型概要如圖 26 所示。

```

Model: "model_4"
-----
Layer (type)                Output Shape                Param #   Connected to
-----
input_1 (InputLayer)        [(None, None)]            0         []
input_2 (InputLayer)        [(None, None)]            0         []
model_3 (Functional)        (None, None, 768)         101677056 ['input_1[0][0]',
                                                    'input_2[0][0]']
lambda (Lambda)             (None, 768)                0         ['model_3[0][0]']
dense (Dense)               (None, 18)                 13842     ['lambda[0][0]']
-----
Total params: 101,690,898
Trainable params: 101,690,898
Non-trainable params: 0

```

圖 26 模型概要 (Summary)

模型設計好後，就準備開始訓練。執行整個模型訓練過程使用 `model.fit()` 方法，在方法中設定每一個參數，包括訓練集輸入特徵、訓練集的標籤、`Batch_size`、`Epochs`、`Tokenizer Sentence Max Length` 等等參數。訓練/測試特徵資料切割比例採用 80/20 以及 10fold 交叉驗證(Cross Validation)。

實驗結果在 Train/Test Split Potion 80/20 部分(如

表 14 BERT 實驗結果所示)：

- i. 當 Batch Size 2, Epochs 5, Tokenizer Sentence Max Length 512 時，推薦的平均準確度 89.87%。
- ii. 當 Batch Size 10, Epochs 5, Tokenizer Sentence Max Length 512 時，推薦的平均準確度 90.27%。
- iii. 當 Batch Size 10, Epochs 5, Tokenizer Sentence Max Length 300 時，推薦的平均準確度 90.58%。

另外模型評估、驗證部分，再以 10fold Cross Validation 實驗，設定的參數 Batch Size 10, Epochs 5, Tokenizer Sentence Max Length 300，結果顯示推薦的平均準確度 91.48%。

表 14 BERT 實驗結果

Train/Test Split Portion	Batch Size	Epochs	Tokenizer Sentence Max	Evaluated (%)
80/20	2	5	512	89.87
	10	5	512	90.27
	10	5	300	90.58
10 fold CV	10	5	300	91.48

在 10fold Cross Validation 中，將全部資料集分成 1:9 的分割，十分之一為測試資料，十分之九為訓練資料，循環十次，讓整份資料集成員都有機會成為測試資料，這是 Cross Validation 方法的特點。在模型的每一 fold 中，模型經訓練後，將一份測試資料送進 model.predict() 方法中算出每一筆資料預測結果的機率，經 10fold 後就可推算出全部資料每一筆的預測推薦佛教引用句結果機率排列。再以此機率分數排名，推算每一結果標籤預測後錯誤數量如圖 27 所示及錯誤率如表 15 所示。

表 15 每一結果標籤預測錯誤率

標籤	0	1	2	3	4	5	6	7	8
總數	590	775	329	923	325	600	445	328	164
錯誤數	79	68	26	91	1	25	12	45	5
錯誤率%	13.4	8.8	7.9	9.9	0.3	4.2	2.7	13.7	3.0
標籤	9	10	11	12	13	14	15	16	17
總數	163	236	290	518	65	182	25	164	243
錯誤數	17	10	50	49	9	15	0	4	36
錯誤率%	10.4	4.2	17.2	9.5	13.8	8.2	0.0	2.4	14.8

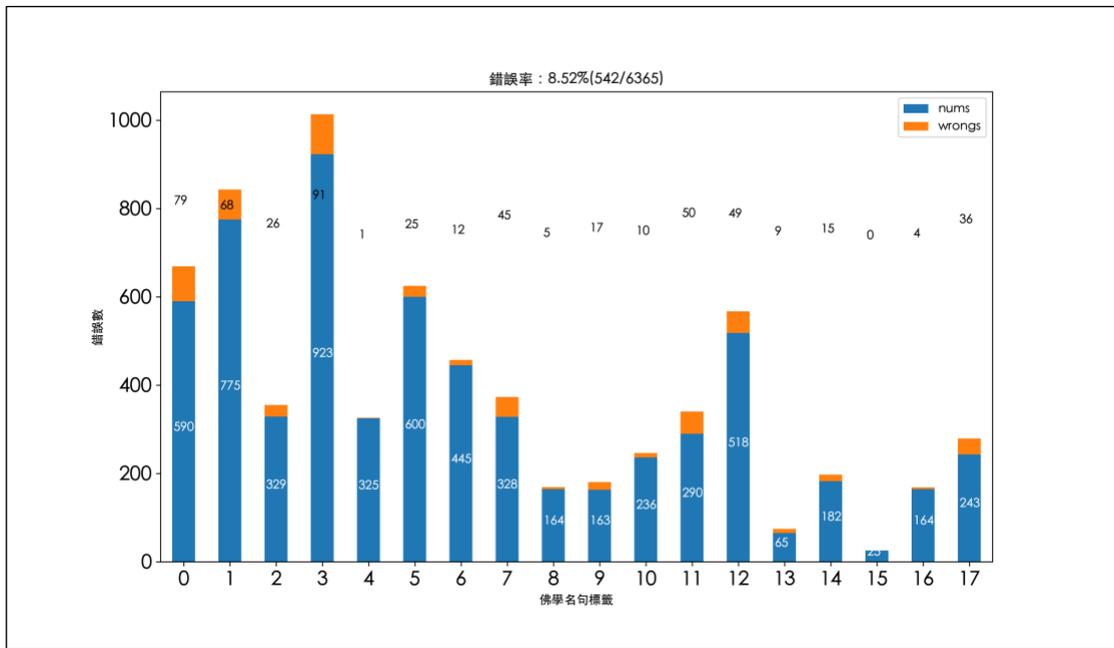


圖 27 BERT 每一標籤預測錯誤數

我們再從 BERT 模型預測分類結果中計算其推薦 1 句、推薦 3 句、推薦 5 句結果如何，如表 16 所示。顯示其推薦 1 句(Top1)佛教引用句的準確度 91.48%、推薦 3 句(Top3)佛教引用句的準確度 96.56%、推薦 5 句(Top5)佛教引用句的準確度 97.97%，及其個別的預測錯誤數如圖 28 所示。

表 16 BERT Top1/Top3/Top5 推薦準確度及錯誤數

Top1		Top3		top5	
ACC	errors	ACC	errors	ACC	errors
91.48%	91	96.56%	26	97.97%	16

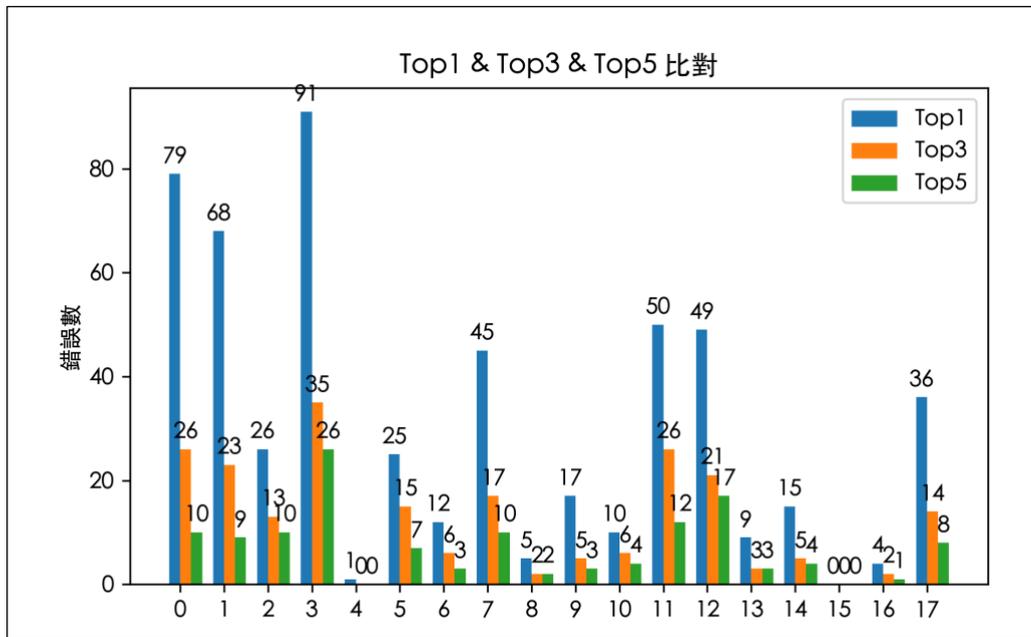


圖 28 BERT Top1/Top3/Top5 錯誤數比對



4 綜合實驗結果

整體實驗經由機器學習分類器 KNN、DT、SVM 及深度學習 LSTM、BiLSTM 及 BERT 模型的推薦方法，得出實驗結果數據，底下我們以機器學習分類器準確度較佳的 SVM 及深度學習 LSTM、BERT 之間比較其 Top1、Top3、Top5 準確度如表 17 所示。

- (1) 由表中顯示出 Top1 時準確度，深度學習 BERT 91.48% 及 LSTM 81.87% 優於機器學習 SVM 79.43%，以深度學習模型比較則 BERT 優於 LSTM 及 BiLSTM。
- (2) 再分析 Top3 時也就是推薦機率最高的 3 句候選佛教引用句各項準確度，SVM 準確度由 79.43% 提升至 89.48%，提升了 10.05%。LSTM 準確度由 81.87% 提升至 90.13%，提升了 8.26%。BERT 準確度由 91.48% 提升至 96.56%，提升了 5.08%。

- (3) 觀察在 Top5 時，也就是推薦機率最高的 5 句候選佛教引用句各項準確度，SVM 準確度由 89.48% 提升至 94.22%，提升了 4.74%。LSTM 準確度由 90.13% 提升至 93.40%，提升了 3.27%。BERT 準確度由 96.56% 提升至 97.97%，提升了 1.41%。
- (4) SVM、LSTM、BERT 三者準確度在 Top5 已經達 93% 以上，相對於 Top3 準確度提升已經減緩，由此判斷候選佛教引用句語意相似情形在 Top3 推薦時相對於 Top1 確實在提供較多正確答案得到大幅度提升。當然 Top5 時可提供更多正確答案。

表 17 SVM、LSTM、BERT 之 Top1/Top3/Top5 Accuracy

Classifier		Top1 (%)	Top3 (%)	Top5 (%)
ML	SVM	79.43	89.48	94.22
DL	LSTM	81.87	90.13	93.40
	BERT	91.48	96.56	97.97

- (5) 在深度學習模型中 BERT 推薦候選佛教引用句準確度相對於 LSTM 及 BiLSTM 高出將近 10%，我們試著分析改進了哪些項目。首先，回顧前面實作 LSTM 實驗時我們分析了哪些問題是致使產生錯誤的原因。

I. 類別資料間意義相近原因

分析 LSTM 在各類別資料中如標籤 0、標籤 1、標籤 3、標籤 7、標籤 12 資料集中會有資料意義相近情形，如表 18 所示，使錯誤率普遍提升。

表 18 LSTM 標籤 0、1、3、7、12 錯誤率

標籤	0	1	2	3	4	5	6	7	8
總數	590	775	329	923	325	600	445	328	164
錯誤數	107	158	40	157	10	48	25	80	22
錯誤率%	18.1	20.4	12.2	17.0	3.1	8.0	5.6	24.4	13.4
標籤	9	10	11	12	13	14	15	16	17
總數	163	236	290	518	65	182	25	164	243
錯誤數	44	26	114	111	30	44	9	11	68
錯誤率%	27.0	11.0	39.3	21.4	46.2	24.2	36.7	6.7	28.0

II. 訓練資料量少，模型訓練不足原因

在 LSTM/BiLSTM 模型中，可能因為蒐集到可供訓練的資料量較少如標籤 13、標籤 15，以至於模型訓練後，測試資料預測的準確度低，導致錯誤數增加，如表 19 所示。



表 19 LSTM 標籤 13、15 錯誤率

標籤	0	1	2	3	4	5	6	7	8
總數	590	775	329	923	325	600	445	328	164
錯誤數	107	158	40	157	10	48	25	80	22
錯誤率%	18.1	20.4	12.2	17.0	3.1	8.0	5.6	24.4	13.4
標籤	9	10	11	12	13	14	15	16	17
總數	163	236	290	518	65	182	25	164	243
錯誤數	44	26	114	111	30	44	9	11	68
錯誤率%	27.0	11.0	39.3	21.4	46.2	24.2	36.7	6.7	28.0

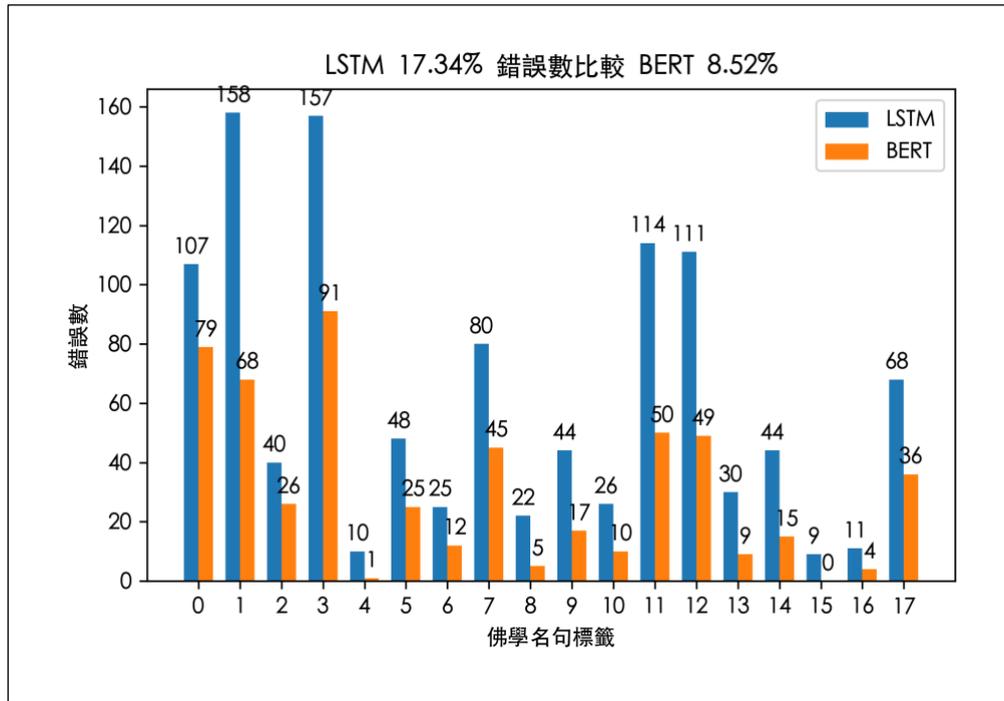


圖 29 LSTM、BERT 錯誤數比

DILA Dharm Drum Institute of Liberal Arts

再者，LSTM、BERT 兩者整體錯誤率比較分析如所示，錯誤率由 LSTM 17.34% 降至 BERT 8.52%，由表 20 顯示，LSTM、BERT 兩者在標籤 0 錯誤率 18.1% 降至 13.4%、標籤 1 錯誤率 20.4% 降至 8.8%、標籤 3 錯誤率 17.0% 降至 9.9%、標籤 7 錯誤率 24.4% 降至 13.7%、標籤 12 錯誤率 21.4% 降至 9.5%，顯示出 BERT 模型確實大幅改進 LSTM 缺點中資料間意義相近問題。且標籤 13 錯誤率 46.2% 降至 13.8%、標籤 15 錯誤率 36.7% 降至 0%，顯示出 BERT 模型亦確實大幅改進 LSTM 訓練資料量少使模型訓練不足準確度低的問題。

表 20 LSTM 及 BERT 錯誤率比較

標籤		0	1	2	3	4	5	6	7	8
總數		590	775	329	923	325	600	445	328	164
錯誤數	LSTM	107	158	40	157	10	48	25	80	22
	BERT	79	68	26	91	1	25	12	45	5
錯誤率%	LSTM	18.1	20.4	12.2	17.0	3.1	8.0	5.6	24.4	13.4
	BERT	13.4	8.8	7.9	9.9	0.3	4.2	2.7	13.7	3.0
標籤		9	10	11	12	13	14	15	16	17
總數		163	236	290	518	65	182	25	164	243
錯誤數	LSTM	44	26	114	111	30	44	9	11	68
	BERT	17	10	50	49	9	15	0	4	36
錯誤率%	LSTM	27.0	11.0	39.3	21.4	46.2	24.2	36.7	6.7	28.0
	BERT	10.4	4.2	17.2	9.5	13.8	8.2	0.0	2.4	14.8

(五) 實際推薦

實驗結果已經證實深度學習的 BERT 模型為目前實驗最佳方法，所以我們就以 BERT 模型為主已選取一段文章來實際測試看看其推薦佛教引用句的效果如何。

1 揀選文章

首先選取測試的文章為：「大智慧的人，他的心非常的安詳，他過著世界太平的日子。應該無所著，生起清淨心。為什麼？世間，不值得我們執著。不值得我們這樣痛苦，煎熬自己。這是很愚痴的人，但問題是習氣難改。」

2 建立環境

建立模型的環境載入 keras-bert 相關模組、預訓練中文 Base BERT 模型、字典與 Tokenizer，最重要的是呼叫我們辛苦訓練、建立的模型並且呼叫已經儲存起來的權重檔案（model load_weighted）。

3 前置處理

送入模型預測之前當然要將測試的文段做前置處理，也就是先經由 BERT Tokenizer 編碼方法處理，並且用 numpy 轉換成 array 格式才能送給 predict 方法做預測。

4 預測

環境建立好之後，我們看看模型對於這段文章推薦的佛教引用句到底輸出的預測是什麼？經由 predict 方法，BERT 模型會對輸入文章推薦出每一類標籤的相對

機率，如圖 30 所示，第一個機率屬於標籤 0、第二個機率屬於標籤 1、...以此類推。也可以用 numpy 的 argsort 方法轉換為結果標籤且以機率由小至大排序。

所以此文章經由模型推薦的佛教引用句由標籤的排序可以看出，第一名是標籤 3，也就是「應無所住，而生其心」，第二名是標籤 0「一切有為法，如夢幻泡影」，第三名是標籤 11「如筏喻者，法尚應捨，何況非法。」

```
1 sentiment_text = '大智慧的人，他的心非常的安詳，他過著世界太平的日子。應該無所著，生起清淨心。'
2
3 ids, segments = tokenizer.encode(sentiment_text, max_len=512)
4
5 my_X_test = [np.array([ids]), np.array([segments])]
6
7 predict = model.predict(my_X_test)
8 LabelRank = np.argsort(predict)
9 print(predict)
10 print(LabelRank)

[[9.6177295e-02 2.8489102e-02 1.0365417e-03 7.9962951e-01 2.5823817e-03
 5.6395256e-05 2.5041523e-05 2.1959697e-03 6.8125874e-03 1.4742593e-03
 2.8604770e-04 5.4539654e-02 3.7794972e-03 9.1733108e-04 9.8394428e-04
 6.0082297e-04 1.1074905e-04 3.0287652e-04]]
[[ 6  5 16 10 17 15 13 14  2  9  7  4 12  8  1 11  0  3]]
```

DILA Dharma Drum Institute of Liberal Arts

圖 30 揀選文章實際測試結果

「應無所住，而生其心」，「應無所住」就是不執著，不在乎自我的利害得失；「而生其心」就是以無私、無我的智慧處理一切事物。也就是「應生無所住心」，生什麼心？生清淨心，清淨心就是般若智慧，生智慧心。確實最符合本段「應該無所著，生起清淨心」的意旨。所以是我們所稱「標準佛教引用句」，正確答案。

「一切有為法，如夢幻泡影」，凡是人、事、物任何可以表現出來的現象，夢---如一場夢、幻---虛幻、泡---泡沫、影---影子，知道一切相都是暫時有的假相。雖然也是勸我們不要執著假相，卻不如「應無所住，而生其心」貼切於原意。

「如筏喻者，法尚應捨，何況非法。」，意思是佛法像渡人的舟筏一樣，過了河就沒有用了，目的在教我們世間和出世的佛法都不能執著，才可得到真正的快樂。不只在「法尚應捨」，更要注意的是「何況非法」。

分析結果證實，BERT 模型確實推薦了標準佛教引用句，也就是最佳答案。



五、結論

我們針對佛教引用句推薦問題，提出一基於深度學習之推薦方法，並評比傳統的機器學習分類器方法的準確度。在機器學習分類器 KNN、DT、SVM，實驗顯示 KNN 模型 63%準確度、DT 模型 77%準確度、SVM 模型 79.43%準確度，SVM 模型優於 KNN、DT 模型。深度學習 LSTM、BiLSTM 及 BERT 模型，實驗顯示 LSTM 模型 81.87%準確度，BiLSTM 模型 81.68%準確度，BERT 模型 91.48%，BERT 模型明顯優於 LSTM、BiLSTM 模型，及機器學習分類器 SVM 模型。

再者，分析深度學習模型推薦之錯誤率，BERT 模型確實改善（一）類別資料部分意義相近情形。（二）訓練資料量少，模型訓練不足。使模型推薦的準確度大幅提升，且實驗證明有完整且適度上下文資訊配合合適的深度學習模型，能學習並推薦適用的佛教引用句，其準確率可達 91.48%。

本論文所提供的主要貢獻在於：（一）我們提出了一個新穎的佛教引用句推薦任務並討論任務的特殊性，並構建了一個任務的真實數據集。（二）我們建議利用 RNN 深度學習架構進行排名以解決推薦任務。（三）我們在一個真實數據集上進行了一系列實驗和評估，結果驗證了我們提出的方法的有效性和實用性。

佛經語言是非常珍貴且豐富的語料庫，雖目前僅揀選 18 種名句進行推薦，效果稍顯侷限，然實驗結果已證明本方法之有效性，未來可再新增更多佛教引用句或非佛教領域之引用句以擴大資料集數量，並應用不同的深度學習模型持續精進，拓展應用層面以實現更廣泛的推薦方法，且可進行寫作的支援系統與互文性（intertextuality）之相關研究，期能於各種類引用句推薦及佛學研究與推展上創造更多實用價值。

六、參考資料

(一) 中文專書

1. Francois Chollet 著、葉欣睿譯（2019），《Deep Learning 深度學習必讀-Keras 大神帶你用 Python 實做》，台北市：旗標科技。
2. 林大貴（2017），《TensorFlow+Keras 深度學習人工智慧實務應用》，新北市：博碩文化。
3. 齋藤康毅著、吳嘉芳譯（2017），《Deep Learning：用 Python 進行深度學習的基礎理論實作》，台北市：碁峰。
4. 劉藹萍（2016），《現代漢語》，重慶大學出版社。

(二) 中文期刊論文

1. 王皓儀(2019)，《深度學習應用於中文成語與同義成語寫作之推薦》，國立臺北科技大學碩士論文。
2. 吳晨皓（2020），《BERT 與 GPT-2 分別應用於刑事案件之罪名分類及判決書生成》，國立高雄科技大學碩士論文。

(三) 西文期刊論文

1. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin. (2017). "Attention is all you need." Advances in neural information processing systems, 30.
2. Benyah Shaparenko, Thorsten Joachims. (2009). "Identifying the original contribution of a document via language modeling." the 32nd international ACM

- SIGIR conference on Research and development in information retrieval July, Pages 696–697.
3. Bethard Steven and Jurafsky Dan. (2010). "Who should i cite: learning literature search models from citation behavior." In Proceedings of the 19th ACM international conference on Information and knowledge management, 609–618.
 4. Lior Rokach, Prasenjit Mitra, Saurabh Kataria, Huang Wenyi, Lee Giles. (2013). "A supervised learning method for context-aware citation recommendation in a large corpus." INVITED SPEAKER: Analyzing the Performance of Top-K Retrieval Algorithms 1978.
 5. Qi He, Jian Pei, Daniel Kifer, Prasenjit Mitra, Lee Giles. (2010). "Context-aware citation recommendation." In Proceedings of the 19th international conference on World wide web, 421–430. ACM.
 6. Tan Jiwei, Wan Xiaojun, Xiao Jianguo. (2015). "Learning to recommend quotes for writing." In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (Austin Texas, January 25–30, 2015). AAAI'15. AAAI Press, USA, 2453-2459.
 7. Tan Jiwei, Wan Xiaojun, Xiao Jianguo. (2016). "A Neural Network Approach to Quote Recommendation in Writings" Conference on Information and Knowledge Management Indianapolis Indiana USA October 24 - 28, ACM.
 8. Xuewei Tang, Wan Xiaojun, Xun Zhang. (2014). "Cross-language context-aware citation recommendation in scientific articles." In Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval, 817–826. ACM.

9. Yang Lu, Jing He, Dongdong Shan, Hongfei Yan. (2011). "Recommending citations with translation model." In Proceedings of the 20th ACM international conference on Information and knowledge management, 2017–2020. ACM.

(四) 網路資源

1. Ahmad husain. (2019, November 12). TEXT CLASSIFICATION WITH LSTM. ALGORITMA TECHNICAL BLOG. <https://algotech.netlify.app/blog/text-lstm/> ◦
2. Chris mccormick. (2016, April 19). Word2Vec Tutorial - The Skip-Gram Model. Chris McCormick. <http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/> ◦
3. Cloudmile. (2020, December 1). 人工智慧、機器學習和深度學習是什麼？. CloudMile Inc. https://www.mile.cloud/zh/resources/blog/What-is-artificial-intelligence-machine-learning-deep%20learning_29 ◦
4. Ikala cloud. (2019, December 23). 常見監督式機器學習演算法-機器學習兩大學習方法. IKala Cloud. <https://ikala.cloud/supervised-learning-classification-regression-algorithms/> ◦
5. It 邦幫忙. (2021, October 8). 交叉驗證 K-Fold Cross-Validation. IT 邦幫忙. <https://ithelp.ithome.com.tw/articles/10279240> ◦
6. It 邦幫忙. (2021, September 24). 決策樹(Decision Tree). IT 邦幫忙. <https://ithelp.ithome.com.tw/articles/10271143> ◦
7. Jason brownlee. (2017, October 4). How to Use Word Embedding Layers for Deep Learning with Keras. Machine Learning Mastery. <https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/> ◦

8. Keras-cn. (2021, November 30). Keras:基於 Python 的深度學習庫. Read the Docs. <https://keras-cn.readthedocs.io/en/latest/#keras>。
9. Leemeng. (2019, July 10). BERT:理解上下文的語言代表模型. LeeMeng. https://leemeng.tw/attack_on_bert_transfer_learning_in_nlp.html。
10. Pandas. (2022, June 23). Pandas Documentation. Pandas. <https://pandas.pydata.org/docs/index.html>。
11. Tensorflow. (2022, May 18). Basic Text Classification. TensorFlow. https://www.tensorflow.org/tutorials/keras/text_classification?hl=zh-tw。
12. Vinkin tsang. (2019, December 23). [深度學習]自然語言處理---基於 Keras Bert 使用. CSDN. <https://blog.csdn.net/zwqjoy/article/details/103671088>。
13. Weifanhaha. (2020, February 28). Transformer-變出近代的強大模型. CoderBridge. <https://tw.coderbridge.com/series/2ec9cf0af3f74ed99371952f4849ae33/posts/5c495ca5e46e40bc98ff623e87919c9a>。
14. 一個處女座的程序猿. (2021, March 28). SVM 算法簡介. 華偉雲. <https://bbs.huaweicloud.com/blogs/252776>。
15. 內容行銷副理 brett grossfeld. (2021, November 2). 機器學習與深度學習. Zendesk 部落格. <https://www.zendesk.tw/blog/machine-learning-and-deep-learning/>。
16. 光彩照人. (2019, November 3). 圖示詳解 BERT 模型的輸入與輸出. 博客園. <https://www.cnblogs.com/gczzr/p/11785930.html>。

17. 自學成功道. (2021, September 1). Python 字典 (dictionary) 基礎與 16 種操作. 自學成功道. <https://selflearningsuccess.com/python-dictionary/>。
18. 李宏毅教授. (2015, September 18). What Is Machine Learning, Deep Learning and Structured Learning? 13. Hung-Yi Lee. http://speech.ee.ntu.edu.tw/~tlkagk/courses_MLSD15_2.html。
19. 阿新. (2018, November 25). Numpy. Random.Shuffle 打亂順序函式. 程式人生. <https://www.itread01.com/content/1543159084.html>。
20. 張育唐 陳藹然. (2011, April 18). 準確度和精確度. 科學 Online. <https://highscope.ch.ntu.edu.tw/wordpress/?p=24512>。
21. 鐘聲吻晚霞. (2021, January 12). 在 M1 的 Mac 上配置 Python 數據科學環境. 知乎. <https://zhuanlan.zhihu.com/p/343607875M1>。

