

# 機器學習技術在自動辨識大學程式設計課程的應用

## Applying Machine Learning Techniques to Automatic Classification of University Programming Courses

陳嘉葳  
國立政治大學資訊科學系  
will8467@gmail.com

李蔡彥  
國立政治大學資訊科學系  
li@nccu.edu.tw

### 摘要

在教育部高教深耕計畫下，全國各大專院校開設了許多程式設計相關課程供學生修習。然此類課程的開設單位、師資來源、學分數等均為後續政策研究的重要議題。目前各大專院校的課程開設情況，係收錄在全國大專院校課程網<sup>1</sup>中，此資料庫內容雖豐富，但目前並無有效的方式可以辨識一門課是否為程式設計相關課程。本研究的目的是希望能根據現有的課程網資料，建立一套自動化課程辨識的機制，以將辨識的結果提供給後續研究之用。本研究嘗試使用機器學習（包含深度學習）技術等人工智慧演算法，根據課程網中的課程大綱資訊，辨識程式設計相關課程，並與人工以關鍵字搜尋方式所得的結果進行比較。實驗結果顯示，深度學習演算法 Bert 對課程大綱深度學習的精準度及各項指標皆遠超越其他機器學習模型或人工關鍵字篩選，在課程辨識的二分類問題上有非常好的表現，說明電腦已能取代人工關鍵字篩選的方法，對課程資訊進行有效的辨識及分類。

**關鍵詞：**人工智慧、程式設計、自動辨識、深度學習、機器學習、自然語言處理

### 1. 研究動機及目的

隨著資訊科技應用的蓬勃發展，掌握數位科技的技術發展成為各國政府施政的重點，其中培育具有運算思維素養的下一代人才更是重要。我國在 108 課綱中已正式將運算思維納為資訊教育的核心，程式設計課程也成為中學的必修課。另外，教育部也在大學的高教深耕計畫中，將邏輯思考與程式設計能力的訓練，列為重點指標之一，並希望在 108 學年度能有 50% 以上的學生有機會修習到程式設計課程。然各大學如何達成此目標，是否有足夠的資源投入，以及是否能達到期望中的學習成效，均需進一步研究分析。然而，如何掌握這些程式設計課程開設的資訊，是進一步研究的前提。目前各大學雖有各自認定程式設計課程的機制，但此資訊尚未能有有效收集的機制。以收納全國大專院校課程資訊的課程網為例，目前尚無程式設計課程的標記，雖可透過關鍵字進行資料庫查詢，但準確度仍有待商榷。

為有效辨識並搜集程式設計課程相關資訊，以作為後續研究的依據，我們嘗試透過人工智慧

中機器學習的方式，從已知為程式設計課程的資料集中學習此類課程的特徵，以期發展出一個有效的分類器，取代過往以手動搜尋關鍵字查找課程的方法。這樣的技術除了可以提高課程分類辨識的準確性外，也可降低資料收集的成本，未來並有可能可以用於各類課程發展的研究上，作為各類人才培育供需研究的基礎，並可發展為課程推薦的系統。

本研究的核心目標在於使用最新的自然語言處理（Natural language processing）技術及機器學習演算法，瞭解是否能取代手動關鍵字查找課程的方法，並分析各類機器學習演算法的效果，找出最適合的方法，做為未來資料收集系統開發的依據。

### 2. 相關研究回顧

隨著數位時代資訊爆炸與人工智慧技術發展，近年許多研究使用機器學習與深度學習技術對各項領域議題進行分類預測及探索，目的在於將繁瑣和巨量的資訊歸納整理，找出歸納的原則及圖像。因此，此「訊息分類」問題亦格外受到重視。過往許多研究([1][2][5][7][8][9])的主要任務皆是將訊息進行分類後，套用在系統上或著建立一套分析或推薦機制。[1][2]將分類（主題分類）問題帶入社群媒體領域，使用機器學習分類的演算法針對留言或文章內容進行分類，應用在推薦系統和主題辨認的應用中；[7][8][9]則是在訊息分類上的研究，例如施旭峰針對 2009 莫拉克颱風的事件，使用自然語言處理對災情訊息進行分類，並建立能判斷災情階段以及嚴重程度的機制，使災情動員能更加有效的運作[8]；陳世榮等針對新聞及社會科學類文本進行分類標記，使用自然語言技術對公投相關新聞進行分類標記[9]；以及，林頌堅使用 LDA 及分類回歸樹對期刊內容進行主題式分類[7]。然而，對於課程分類或程式設計課程分類的相關研究，目前較無相關研究或發展，原因可能在於資料的缺乏、以及需要更加完整的課程紀錄資料。

關於訊息分類問題的技術層面，許多機器學習以及新的深度學習技術被開發，用來解決相關問題。如[6]，以機器學習模型對 Twitter 上的發文內容（tweet）進行分類，目的在於找出「假帳戶」（Spam account），使用 Random forest、Logistic Regression 等多種方法，並用 Accuracy、Precision、

<sup>1</sup> <http://ucourse-tvc.yuntech.edu.tw/>

Recall、F-Score 進行評估；另外，關於神經網絡等深度學習技術，在近年成為資料科學家開發以及應用上的主流，其中最有名氣的無非是 Google 在 2019 年所開發的 Bert 自然語言深度學習模型 (Bidirectional Encoder Representation from Transformers) [3]，能雙向辨別上下文文意，因為經過字詞與文意相關的填空訓練，對於訊息分類的任務能更加精準，且非常適合二分類任務，如[4]使用 Bert 對中文新聞資料進行主題預測，增加新聞推薦系統在推薦上的準確率。參考上述研究，本研究的模型及分析比較分成三個方向：傳統機器學習 (Random forest、Logistic Regression 等)；深度學習模型 (Bert)；人工使用關鍵字篩選程式設計課程。我們將比較這三類方法在「程式設計課程辨識」上的效能差異，尤其探討 Bert 深度學習模型是否能課程辨識上有效取代人工方法，並進一步探討 Bert 在本研究任務的表現上是否能超越傳統機器學習模型。

國外有針對課程分類與推薦的相關研究，如 [5] 欲透過個人化資料打造課程推薦分類機制，本研究希望延伸此類研究，延伸 [7][8][9] 針對訊息和內容分類的概念，使用「課程大綱」做為內容分類的判準，放入機器學習及深度學習模型學習，期望能有系統性地對程式設計課程進行整理、找出分類模式、精準預測，未來放入更多新課程進入模型時，能有效、快速的從大量課程中找出程式設計課程，能即時地進行推薦或放入資料庫進行整理。

### 3. 資料處理

#### 3.1 資料來源與清理

本研究資料來源為兩部分，我們使用 Python pandas 套件清理資料。

第一，程式設計課程資料集：我們透過教育部取得部分期間技職體系學校所開設的程式設計課程，作為辨識訓練的標準答案 (Ground Truth)。此資料集涵蓋 106 學年度第一學期及第二學期、以及 107 學年度第一學期，共計三個學期由各校提供之程式設計課程 (共 7132 門) 的資料集，資料欄位包含學期、學校名稱、系所名稱、課程名稱等。

第二，課程網資料集：從「大專校院課程資源網」以人工方式下載所有科技職體系學校在 106-1、106-2、107-1 這三個學期內各系所所開設的所有課程，除學期、學校名稱、系所名稱、課程名稱外，尚包含課程大綱、男女修課人數等欄位。此資料集共計有 25 萬 6602 筆課程資料。

#### 3.2 資料合併及標準答案標記

在資料處理上，我們將程式設計課程資料集作為判斷該課程「是否為程式設計課程」的標準答案，對課程網資料集的課程進行比對。依照

「學期、學校名稱、系所名稱、課程名稱」四個欄位當作判準欄位，找出課程網資料集中所有包含在程式設計資料集中的課程，標記成「1」（表示該課程為程式設計課程），其餘課程標記成「0」（表示該課程為非程式設計課程），去除關鍵欄位（如校系、課程大綱等）為遺漏值之資料後，共有 25 萬 6419 筆課程資料，如表 1 所示，程式設計課程在清理後資料之比例分布如下：

表 1 「是否為程式設計課程」欄位比例分佈 (課程網資料集)

是否為程式設計課程	統計數量
1：程式設計課程	7326(2.9%)
0：非程式設計課程	249093(97.1%)

我們以此建立標準答案欄位，放入機器學習、深度學習等模型中進行學習和預測。

#### 3.3 課程大綱修正

本研究以自然語言處理技術進行模型建構，主要以「課程大綱」當作模型輸入層，預測該課程是否為程式設計課程；然部分課程缺乏有效或有意義的課程大綱（例如只包含該校課程查詢網址或無實質內容），我們將此類課程的課程大綱替換為課程名稱，成為修正後的課程大綱內容。

#### 3.4 資料修正過程

在資料集的比對過程中，發生許多同一堂課因系所名稱、課程名稱在兩筆資料集使用不一致導致比對失敗的情況。系所名稱不一致的狀況，有些是因縮寫的關係，如「資工系」與「資訊工程學系」、「電子系」與「電子工程系」等；另外有些是系所名稱不在大學課程網當中，如院級課程（外語學院、商學院等）。課程名稱上亦會有相同課程，紀錄名稱卻有些許差異的情形，如「程式設計（一）」與「程式設計 1」、「互動式多媒體設計實務」與「互動多媒體設計實務」，一到兩個字在紀錄上的差異、用阿拉伯數字還是國字數字紀錄、全半形標點符號及紀錄中多按一個空白鍵等原因都會造成比對上的困難。因此本研究反覆過濾程式設計課程資料集中比對失敗的名單，手動對上述狀況進行修正，將程式設計課程資料集中約 98% 的課程標記在課程網資料集中，但仍有 135 筆（約 1.9%）資料沒辦法與課程網任一堂課進行對應，這些課程比對失敗的原因在於課程網資料中本身並不包含這些課程，對課程網依照校系所有課程進行查詢時，亦沒有發現與這些課程相似的課程名稱。最後，我們只能將這些課程過濾刪除。

## 4. 實驗方法及研究成果

### 4.1 實驗方法

本研究探討能否以機器學習預測的方式增進程式設計課程的比對效能。因此我們將分為人工查找（關鍵字）與自然語言處理（機器學習與深度學習）兩種方法進行查詢分類，並評估成效：

第一，**人工查找**：將課程網資料放入資料庫，手動下關鍵字查找程式設計相關課程，再比對有多少課程真實為程式設計課程。人工篩選所使用關鍵字包含：程式、運算思維、APP、創客、網頁設計、資料分析、Labview、計算機程式、HTML、Python、XML、Web、人工智慧與機器學習、人機互動、Scratch、R 語言、VBA、資料庫系統、嵌入式系統、C 語言、Java、Unity。

第二，**機器學習模型**：我們使用 Logistic Regression、Decision Tree、Random Forest 等機器學習模型及 Google Bert 自然語言分類模型進行預測，模型的輸入（x）與輸出（y）如下方公式(1)所示，我們將課程大綱轉換為詞向量（word vector；或詞嵌入，word embedding）並使用以上模型中不同的機制抽取特徵進行學習、預測。前三種方法的輸入層皆用相同方法將課程大綱轉換成詞向量：使用 Python gensim 中 word2vec 套件，載入 CKIP 繁體中文詞向量模型（使用中央通訊社文本訓練而成），進行詞向量轉換；而 Google Bert 則是用內建已訓練好的模型進行詞向量轉換的工程。

$$y_{\text{是否為程式設計課程}} = x_{\text{課程大綱}} \quad (1)$$

### 4.2 使用模型

#### 4.2.1 Logistic Regression（羅吉斯回歸） [11]：

在機器學習中，羅吉斯回歸依舊為運算成本極低、具備一定準確性之分類模型，成為本研究作為對照的模型之一。透過下方公式(2)，在不同的輸入（input）之下，找出其輸出層的類別為「1」的機率，使用 Python Sklearn 套件實做，分類器使用 LogisticRegression，將 solver 參數調整成「lbfgs」，其餘參數為預設。

$$y_{\text{是否為程式設計課程}} = \frac{e^{f(\text{課程大綱詞向量})}}{1 + e^{f(\text{課程大綱詞向量})}} \quad (2)$$

#### 4.2.2 Decision Tree（決策樹）與 Random Forest（隨機森林） [12]：

決策樹與隨機森林法皆為樹狀結構演算法，透過對課程大綱的詞向量做資訊量提取，找出分類規則對所有課程進行分類，建立樹狀結構，進而判斷，找出程式設計課程。而隨機森林做為決

策樹衍生的模型，透過多數投票機制找出最適合對程式設計課程進行預測之模型，進而套用在資料分類上。以上兩種演算法亦使用 Sklearn 實現，因資料龐大，減少不必要的運算成本，故在決策樹的參數調整上，以「entropy」為主要提取資訊的判準，最深深度設為3層；在隨機森林中，除最深深度設為3層之外，使用10顆樹進行最適模型的判斷。

#### 4.2.3 SVM（support vector machine，支援向量機） [10]：

支持向量機(SVM)為經典的機器學習演算法，其原理在於對於一個二維平面的所有點，依照給定特徵（課程大綱的詞向量）將平面切割成兩個群集，不斷迭代，直到找到 margin 為最大、兩類別差異最大的平面為止（如圖1中 margin 的範圍），善於處理二分類任務，本研究使用 Sklearn 套件中 SVC 分類器進行實現，使用「linear」作為 kernel 進行分類，其他參數為預設。

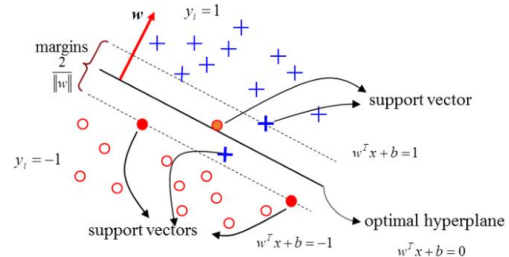


圖1 SVM 運算機制解釋圖（資料來源：[10]）

#### 4.2.4 Google Bert [3]：

全名為 Bidirectional Encoder Representation from Transformers，由 Google 開發的詞向量深度學習模型。Bert 是透過兩階段訓練而成的模型，第一階段的「Pre-training」為「非監督式」學習，透過克漏字填空（Masked LM）、預測下一個子句（Next Sentence Prediction）兩項機制訓練詞向量模型，與傳統 word2vec 等方法比較起來，差異在於 Bert 的訓練方式為「雙向」，於詞向量訓練時考量上下文，因此能提高辨識語句順序及意義的準確度；第二階段的「Fine-tuning」為「監督式」學習，使用第一階段訓練好的詞向量模型對文本進行特徵提取。Bert 透過 Token Embeddings 將句子轉換成張量、使用 Segment Embedding 判斷兩子句是否相似、以及利用 Position Embedding 分辨相同用詞在不同位置的意義（如圖2所示），藉以上機制準確地針對大量文本進行分類。因不同課程之課程大綱敘述語句有所差異，使用 Bert 模型，能將複雜且零散的課程大綱化繁為簡進行程式設計課程的分類工作，加上將上下文順序考量進模型的機制能解決課程大綱內容歧異度大之問題，故本文將 Bert 作為深度學習分析的主軸，與一般機器學習的詞向量訓練模型（羅吉斯回歸、隨機森林等）及手動篩選等方法進行比較。本研究使用 Pytorch

實現，考量課程大綱多半為同時夾雜中英文，且英文特徵如 Python、app 等字彙會是判斷關鍵，故 Pre-training 與 Fine-tuning 兩階段皆使用「bert-base-multilingual-cased」模型。（其他參數：優化器選用 AdamW，learning rate 為 2e-5，epsilon 為 1e-8）

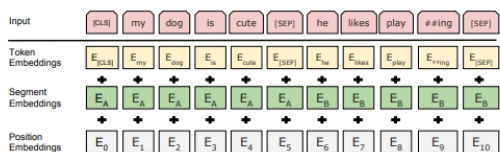


圖2 Token Embeddings、Segment Embedding、Position Embedding 運作流程圖[3]

### 4.3 評估方法

針對手動篩選關鍵字及各項自然語言處理模型，我們皆以 Accuracy、Precision、Recall、F1 Score 四項評估指標進行精準度判斷，分類結果的混淆矩陣（如圖3）及四項指標的計算公式如下：（手動計算及使用 Sklearn 套件進行計算）

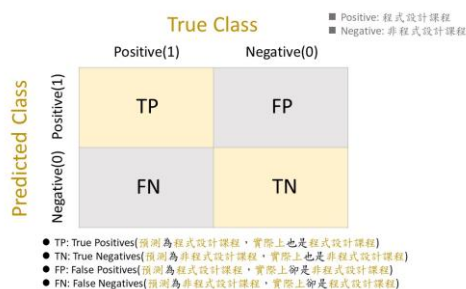


圖3 混淆矩陣 (Confusion Matrix)

- Accuracy :  $(TP + TN) / (TP + TN + FP + FN)$ ，預測正確率，只考慮是否正確預測目標實際的類別。
- Precision :  $TP / (TP + FP)$ ，精確率，預測為「程式設計課程」的資料，有哪些真實為程式設計課程（有多少課程不是真正的程式設計課程）。
- Recall :  $TP / (TP + FN)$ ，召回率，真實為「程式設計課程」的資料，有多少被預測為程式設計課程（有哪些沒有被預測為程式設計課程）。
- F1 Score :  $2 * ((Precision * Recall) / (Precision + Recall))$ ，綜合考量 Precision 和 Recall 後計算的指標。

### 4.4 實驗結果

本研究成果的分析著重在：「人工篩選」、及「傳統機器學習」（Logistic Regression、Decision tree、Random Forest、SVM）、及「深度學習」

（Bert）等方法在批量分辨程式設計時的效能。人工篩選結果如表2：

表2 人工篩選結果

年度	篩選結果	評估指標
106 學年度	<ol style="list-style-type: none"> <li>該年度課程數：256,963 門</li> <li>該年度程式設計課程：4,440 門</li> <li>用 4.1 節關鍵字查找之課堂數：4,183 門</li> <li>共有 2,276 門課為標準答案中程式設計課程</li> <li>Confusion Matrix TP : 2,276 FP : 1,907 (4,183-2,276) FN : 2,164 (4,440-2,276) TN : 250,616 (256,963)</li> </ol>	<ol style="list-style-type: none"> <li>Accuracy: 0.984</li> <li>Precision: 0.544</li> <li>Recall: 0.513</li> <li>F-score: 0.528</li> </ol>
107 學年度第一學期	<ol style="list-style-type: none"> <li>該年度課程數：127,848 門</li> <li>該年度程式設計課程：2,692 門</li> <li>用 4.1 節關鍵字查找之課堂數：2,264 門</li> <li>共有 1,252 門課為標準答案中程式設計課程</li> <li>Confusion Matrix TP : 1,252 FP : 1,012 (2,264-1,252) FN : 2,164 (2,692-1,252) TN : 250,616 (127,848-3,704)</li> </ol>	<ol style="list-style-type: none"> <li>Accuracy: 0.981</li> <li>Precision: 0.553</li> <li>Recall: 0.465</li> <li>F-score: 0.505</li> </ol>

以深度學習與傳統機器學習所獲得之結果如表3：

表3 機器學習各模型的預測結果

使用模型	資料集及驗證程序	交叉驗證預測結果 (平均)
Google Bert	<ol style="list-style-type: none"> <li>母體資料</li> <li>交叉驗證：5-fold cross validation</li> </ol>	<ol style="list-style-type: none"> <li>Accuracy: 0.985</li> <li>Precision: 0.899</li> <li>Recall: 0.819</li> <li>F-score: 0.854</li> </ol>
Google Bert (sampling)	<ol style="list-style-type: none"> <li>抽樣資料 (程式設計：非程式設計 = 1 : 1)</li> <li>交叉驗證：5-</li> </ol>	<ol style="list-style-type: none"> <li>Accuracy: 0.927</li> <li>Precision: 0.928</li> <li>Recall: 0.927</li> <li>F-score: 0.927</li> </ol>

	fold cross validation	
Random Forest	1. 母體資料 2. 交叉驗證： 5-fold cross validation	1. Accuracy: 0.934 2. Precision: 0.578 3. Recall: 0.6508 4. F-score: 0.600
Decision Tree	1. 母體資料 2. 交叉驗證： 5-fold cross validation	1. Accuracy: 0.971 2. Precision: 0.486 3. Recall: 0.500 4. F-score: 0.493
Logistic Regression	1. 母體資料 2. 交叉驗證： 5-fold cross validation	1. Accuracy: 0.974 2. Precision: 0.829 3. Recall: 0.594 4. F-score: 0.643
SVM	1. 母體資料 2. 交叉驗證： 5-fold cross validation	1. Accuracy: 0.029 2. Precision: 0.51 3. Recall: 0.500 4. F-score: 0.029

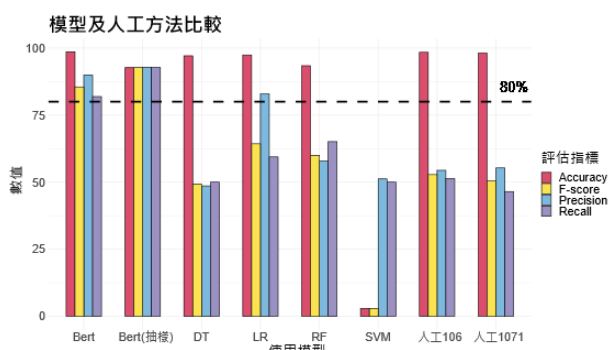


圖4 各類模型辨識效果比較

從表2、表3以及圖4中可以看出，除 SVM 外，不論人工篩選、機器學習、深度學習，在 Accuracy 的表現皆落在0.927到0.985左右（在此數值，人工篩選的數值為0.98，大於所有機器學習的模型）。Accuracy 普遍偏高的主要原因在於 True Positive 的程式設計課程數在整體課程總數的比例較低，因此需與其他指標一併比較方具意義。關於 F-score，同為使用母體資料下，最高為 Bert 的0.854，遠大於機器學習模型中最高的0.642（Logistic Regression），且勝過以人工篩選方式得到的 F-score（106學年度的0.528及107第一學期的0.505），因 F-score 同時考量 Precision 和 Recall 的情況，可以說不論是針對程式設計課程進行預測、還是針對非程式設計課程進行預測，Bert 深度學習的方式在本研究欲探討的目標有最好的結果，且精準度在一定的水準上，當使用程式設計課程與非程式設計課程比例為1:1的抽樣資料時，F-score 有高達0.92的表現；在使用程式設計課程占比約3%的母

體資料進行預測時，F-score 也具有0.854的高標準，表示即使在 Positive（程式設計課程）與 Negative（非程式設計課程）比例相當失衡且在課程大綱描述長短不一致等因素下，Google Bert 仍能表現出0.9以上的預測精確率，證明深度學習的預測方式能有效取代人工篩選關鍵字的模式、甚至勝過傳統機器學習模型，在程式設計課程查找及辨識系統的應用上，深具應用價值以及可能性。

## 5. 結論及未來展望

在大數據與機器學習領域，二分類的問題在資訊判別的應用中經常被討論，本研究針對「程式設計課程自動辨識」使用多項方法嘗試預測、比較並找出最佳解，預測結果為使用 Google Bert 模型能精確地針對程式設計課程進行預測，也顯示出在未來將此概念以及相關實作，實際應用在課程推薦或自動課程辨識系統的潛在可能性。

本研究在資料處理的環節碰到許多困難以及限制：1、各校在透過不同管道登記資料時，在校名、系所名、課程名稱上皆有些許誤差，因此本研究必須透過手動方式校正嘗試修正此問題，但仍有135門課程無法對應上；2、因本研究主要使用課程大綱做為自動判斷時使用欄位，碰到遺漏值以及些許學校在課程大綱上只附上網址或無內容時，本研究僅能將這些課程之課程名稱替代其課綱。雖最後本模型的精準度高，但長期而言，如能蒐集到完整的課程大綱進行分析，應有助於進一步提升準確度；3、本研究使用的標準答案（ground truth）侷限於技職體系學校，並無一般大學資料。

本次研究結果說明了在特定領域的課程辨識上（程式設計領域課程），「電腦辨識能取代人工篩選」的確有其可能性，並能將此次成果作為未來在其他類課程自動辨識領域上的基石，期許在未來，能延伸至各系統開發，如課程自動推薦系統等。

## 6. 參考文獻

- [1] D. B. Kurka, A. Godoy, F. J. Von Zuben, "Using Retweet Information as a Feature to Classify Messages Contents," in *WWW '17 Companion: Proc. of the 26th International Conf. on World Wide Web Companion*, April 2017, pp.1485-1491, <https://doi.org/10.1145/3041021.3053904>
- [2] I. Ronen, I. Guy, E. Kravi, M. Barnea, "Recommending social media content to community owners," in *SIGIR '14: Proc. of the 37th International ACM SIGIR Conf. on Research & Development in Information Retrieval*, July 2014, pp.243-252, <https://doi.org/10.1145/2600428.2609596>
- [3] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, "BERT: Pre-training of Deep Bidirectional

- Transformers for Language Understanding,” 2019. arXiv preprint arXiv:1810.04805
- [4] J. Liu, C. Xia, X. Li, H. Yan, T. Liu, “A BERT-based Ensemble Model for Chinese News Topic Prediction,” in *BDE 2020: Proc. of the 2020 2nd International Conf. on Big Data Engineering*, May 2020, pp.18–23. <https://doi.org/10.1145/3404512.3404524>
- [5] Y. Park, "Recommending Personalized Tips on New Courses for Guiding Course Selection," in *ACM SE '17: Proc. of the SouthEast Conf.*, April 2017, pp. 172–174. <https://doi.org/10.1145/3077286.3077567>
- [6] Z. Alom, B. Carminati, E. Ferrari, “Detecting Spam Accounts on Twitter,” in *Proc. of 2018 IEEE/ACM International Conf. on Advances in Social Networks Analysis and Mining (ASONAM)*, 2018. Available: <https://ieeexplore.ieee.org/document/8508495>
- [7] 林頌堅，利用文字內容主題特徵與機器學習方法探討 MIS 相關期刊在 ISI 資料庫的主題分類。教育資料與圖書館學，52卷3期 (2015 / 09 / 01)，2015, pp. 269 - 298。
- [8] 施旭峰，〈災難事件下新媒體資訊傳播方式分析與自動化分類設計——以八八風災為例〉。國立政治大學資訊科學系碩士論文，2013。
- [9] 陳世榮，社會科學研究中的文字探勘應用：以文意為基礎的文件分類及其問題。人文及社會科學集刊，27(4)，2015, pp.683-718。
- [10] 機器學習-支撐向量機(support vector machine, SVM) 詳細推導。[Online]，來源：<https://reurl.cc/Q33yoo>
- [11] 機器 / 統計學習：羅吉斯回歸 (Logistic regression)。[Online]，來源：<https://reurl.cc/D6gD5E>
- [12] Machine learning 下的 Decision Tree 實作和 Random Forest(觀念)(使用 python)。[Online]，來源：<https://reurl.cc/EznD50>