

國立東華大學資訊工程系
National Dong Hwa University
110 學年度大學部畢業專題研究報告
110 CSIE Undergraduate Project Report

基於自然語言處理與深度學習
實現自動化審文的發問平台



指導教授 Advisor： 顏士淨教授

專題參與人員 Team Member： 薛祖恩
徐向廷
張宸瑋
葉耀中

中 華 民 國 111 年 5 月 20 日

國立東華大學資訊工程學系

專題報告原創性聲明

National Dong Hwa University

Department of Computer Science and Information Engineering

Statement of Originality

本人鄭重聲明：

所呈交的專題報告是在指導老師指導下進行的研究工作及取得的研究成果。除文中已經註明引用的內容外，本報告不包含任何其他個人或集體已經發表或撰寫過的研究成果。對本文的研究做出重要貢獻的個人與集體，均已在文中以明確方式標明。若有違上述聲明，願依校規處分及承擔法律責任。

I hereby affirm that the submitted project report is the result of research under the supervision of my advisor. Except where due references are made, the report contains no material previously published or written by another person or group. All significant facilitators to the project have been mentioned explicitly. Should any part of the statement were breached, I am subject to the punishment enforced by the University and any legal responsibility incurred.

學號 Student No.	學生姓名 Name	親筆簽名 Signature
410821305	薛祖恩	
410821308	徐向廷	
410821312	張宸瑋	
410821319	葉耀中	

日期：Date 2022/5/20

計畫摘要 Abstract

Keywords: Word2Vec, BERT, NLP, CNN, Flask

自然語言處理以及深度學習的結合是一個非常值得探討的主題。我們認為社群以及社交網站等平台需要相關技術的應用；因此，我們決定基於自然語言處理以及深度學習來實作一個資訊工程類別的審文模型。

首先，我們使用爬蟲抓取各類網站上有關資訊工程的文章，以及下載來自 Kaggle 的非資工相關新聞以及文章；在做完資料蒐集過後，我們使用 **NLTK**_[6] 完成資料前處理，並使用 **Word2Vec**_[7]與 **Gensim**_[5]完成詞嵌入，最後再使用卷積神經網路以完成模型訓練；再將此模型運用在以 Flask 開發的網站上，製作出可以過濾非資工相關文章以及保留資工相關內容的平台功能，也完成 Q&A 平台「NDHU CSIEplus」並部署在雲端主機 Heroku 上。

我們發現，在 2018 年由 Google 開發的基於變換器的雙向編碼器表示技術（**BERT**_[4]），也是一個可以作為模型訓練之前的利器，在完成以 BERT 作為詞嵌入以及同樣使用卷積神經網路進行訓練的模型之後，我們比較了兩種不同模型版本並歸納出了最終的結論以及其優缺點。

目錄 Table of Contents

一、	前言 Introduction	1
二、	研究動機與研究問題 Motivation and Research Problem.....	2
三、	研究方法與步驟 Research Method.....	3
3.1	研究前知識.....	3
3.2	資料蒐集.....	4
3.3	資料前處理.....	4
3.4	Word2Vec 與 Gensim.....	5
3.5	基於 Word2Vec 之 CNN 模型訓練.....	8
3.6	基於 BERT 之 CNN 模型訓練.....	10
3.7	網站架設與模型串接.....	13
3.8	證明模型可用性.....	16
四、	結果及討論 Research Results an Conclusions.....	18
4.1	Word2Vec 以及 BERT 的比較.....	18
4.2	模型評價.....	19
4.3	結語.....	19
五、	參考文獻 References	20

圖目錄 List of Figures

(圖一) 類神經網路中 Perceptron 示意圖	3
(圖二) 卷積示意圖	4
(圖三) CBOW 示意圖	6
(圖四) Skip Gram 示意圖	6
(圖五) Word2Vec 權重計算	7
(圖六) Gensim - boy 與 girl 相似度	8
(圖七) Gensim - girl 與 mug 相似度	8
(圖八) 基於 Word2Vec 之 CNN 模型架構	8
(圖九) 基於 Word2Vec 之 CNN 模型 accuracy history	9
(圖十) 基於 Word2Vec 之 CNN 模型 loss history	9
(圖十一) 基於 Word2Vec 之 CNN 模型混淆矩陣	10
(圖十二) 遷移學習示意圖	11
(圖十三) BERT 動態單詞判斷	11
(圖十四) 基於 BERT 之 CNN 模型架構	11
(圖十五) 基於 BERT 之 CNN 模型 accuracy history	12
(圖十六) 基於 BERT 之 CNN 模型 loss history	12
(圖十七) 基於 BERT 之 CNN 模型混淆矩陣	13
(圖十八) 與資訊工程不相關之文章	14
(圖十九) 發佈失敗範例	14
(圖二十) 與資訊工程相關之文章	15
(圖二十一) 發佈成功範例	15
(圖二十二) NDHU CSIEplus 網站	16
(圖二十三) 基於 Word2Vec 之 CNN 模型 ROC 曲線	17
(圖二十四) 基於 BERT 之 CNN 模型 ROC 曲線	17

一、前言 Introduction

我們認為人工審文是一件非常消耗時間以及體力的事情，也因為許多匿名發文群組是由同學主動經營，但是經營優先序遠遠低於自身的課業、社團活動等等。常常會因為版主的個人時間分配，而讓有許多緊急而且重要的貼文沒有辦法在第一時間被群組的人知道。例如：寵物走失，如果沒有在第一時間發文，萬一發生了不幸，寵物們將極有可能錯過黃金救援時間，進而導致悲劇。

另一方面，貼文的品質也是需要被限制的，身為版主，絕對不會希望自己所經營的群組中有許多不符合板規、不合適、甚至是非法的貼文；站在開發者的角度，也不希望過多的違規文章佔據資料庫的儲存空間。例如：4chan 是一個言論自由極為開放的平台，也因此，有許多非法以及色情的文章出現在這個平台中，即使有志願管理員負責刪除這些貼文，但如果同時出現大量的違規文章，我們相信一般人是絕對無法迅速地下架所有文章的。

此外，在許多平台之中，官方會被動刪除貼文。例如：某極端組織在各大社群平台發佈血腥或宣傳恐怖主義的影片。如此殘忍的貼文若在收到大量的檢舉信件之後才刪除，對於那些看到的人們而言已經來不及了；如果可以在發文之前就知道文章不符合大眾社會觀感，將可以把傷害降到最低。

基於種種以上原因，我們決定製作一個審核資訊工程類別文章的模型，並建立一個平台，取名為「NDHU CSIEplus」，使東華的同學們能夠盡情的在平台上提問有關資訊工程的問題，同時使審文模型負責貼文品質以及維護平台環境的工作。

二、 研究動機與研究問題 Motivation and Research Problem

人工審文除了耗時、也會在有許多文章待審時對維護人員造成一定的壓力。我們認為，將文章審核交給模型進行處理將會大幅度減輕小編或版主的負擔，同時也可以加快社群或群組的更新速度。

現階段有許多社群平台，尚未有自動審核文章的機制；我們認為在各個版或是群組、社團中擴增此功能，將會對平台的成長而言為一大助力。

人工審文在維護人員有精神的時候，文章可發佈的準確率是可以被確定的；然而，隨著文章越來越多以及審文的時間越來越久，維護人員常常會出現疲憊的症狀，伴隨的是維護人員的精神狀態降低進而導致文章可發佈的準確率降低。

另一方面，使用模型做自動化審文，不僅可以維持高準確率，還可以全年無休、及時處理需要被審核的文章，並且隨著時間過去，也不會使模型判斷的準確率降低，模型也不需要休息。

然而，在研究模型自動化審文系統時，我們發現電腦無法直接透過文章來判斷單詞與單詞之間的關係，也無法判斷目標類別之關聯性。因此，我們需要找到一個能夠讓電腦分辨單詞間關係與目標類別關聯評分的技術，以確保電腦所認知的語言與人類認知的自然語言盡可能地完全一致。

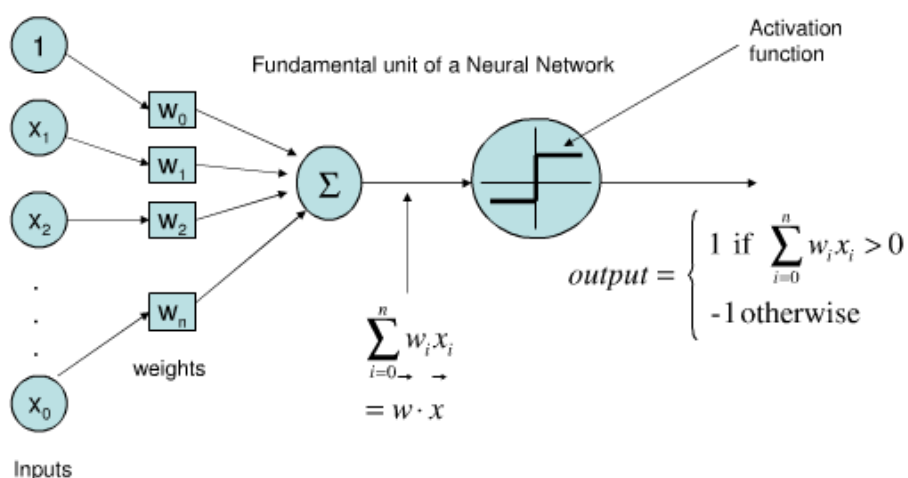
綜上所述，我們主要研究的問題是如何讓電腦在文章中判斷單詞間的關係以及判斷文章與資訊工程的關聯性；為了驗證我們的模型能夠在「NDHU CSIEplus」正確地處理以及判斷文章，實作網站的前後端開發與在網站上實現模型判讀也是必須的。

三、 研究方法與步驟 Research Method

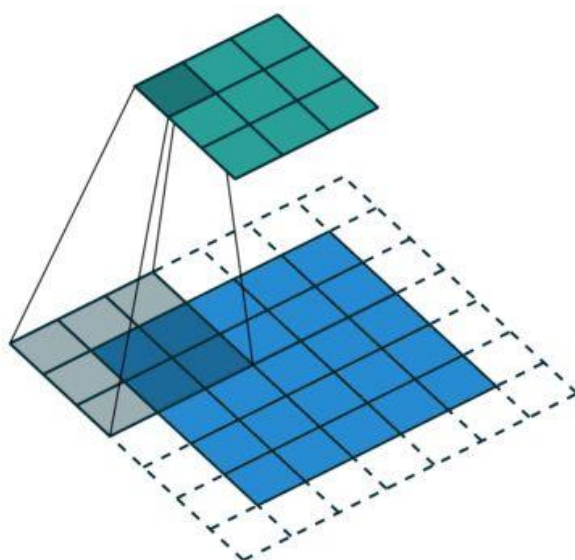
3.1 研究前知識

我們對於自然語言處理原先只有薄弱的知識，包括文字的前處理以及模型訓練的大致流程，也缺少實作的經驗，因此我們在網路上參考多篇的論文並且蒐集大量的資料，以整理出合適的演算法及軟體套件來完成一個自動審文系統。

卷積神經網路（Convolutional Neural Network）主要應用於影像辨識與自然語言處理等相關領域，為**類神經網路**[8]（見圖一）以及卷積層的組合架構。卷積神經網路透過「反向傳播」搭配「梯度下降」以調整模型權重進行監督式學習（Supervised Training）。在每一次進行**卷積（Convolution）** [9]時，為了提取同樣的特徵，同一層會共用權重與偏差（見圖二）。權重數量龐大會浪費計算資源，而在卷積層間嵌入池化層可以減少展開（Flatten）後全連接層（Dense Layer）的權重數量。池化層當中最廣泛被使用的是最大池化（Max Pooling），比起平均池化（Average Pooling）等池化方式而言，其運算效率較其他選項優越。



（圖一）類神經網路中 Perceptron 示意圖



(圖二) 卷積示意圖

訓練類神經網路模型時，需要避免過度擬合（Overfitting）以及低度擬合（Underfitting）的狀況。模型的收斂效果不佳時稱為低度擬合，發生於資料集特徵不明顯（過於隨機）、樣本數量過少或是模型本身的神經元層數不足；過度擬合發生於訓練資料特徵組合過於單調，導致模型出現「死背」的狀況而不具泛化能力，進而無法準確地應用於測試資料集。以上狀況可以透過隨機關閉神經元（Dropout）與批次正規化（Batch Normalization）來緩解上述問題。除此之外，大幅度擴增資料量也可以避免過度學習的發生。

3.2 資料蒐集

首先，我們從 Kaggle、Stack Overflow 等網站蒐集了與資訊工程相關的文章共約 120 萬筆，非資訊工程相關的文章共約 100 萬筆。

3.3 資料前處理

在自然語言處理（Natural Language Processing）當中，我們首先面臨到的問題就是：將人類能夠一目了然的文字轉換為電腦能夠解讀並且

計算的型態。大部分的學者皆選擇將每一個單詞或是文章段落轉換為一個矩陣，而這個步驟稱為字詞向量化或是「詞嵌入」（Word Embedding）。然而，在模型的訓練資料量受到侷限的狀況下，我們還需要在進行詞嵌入以前做一系列的前處理。

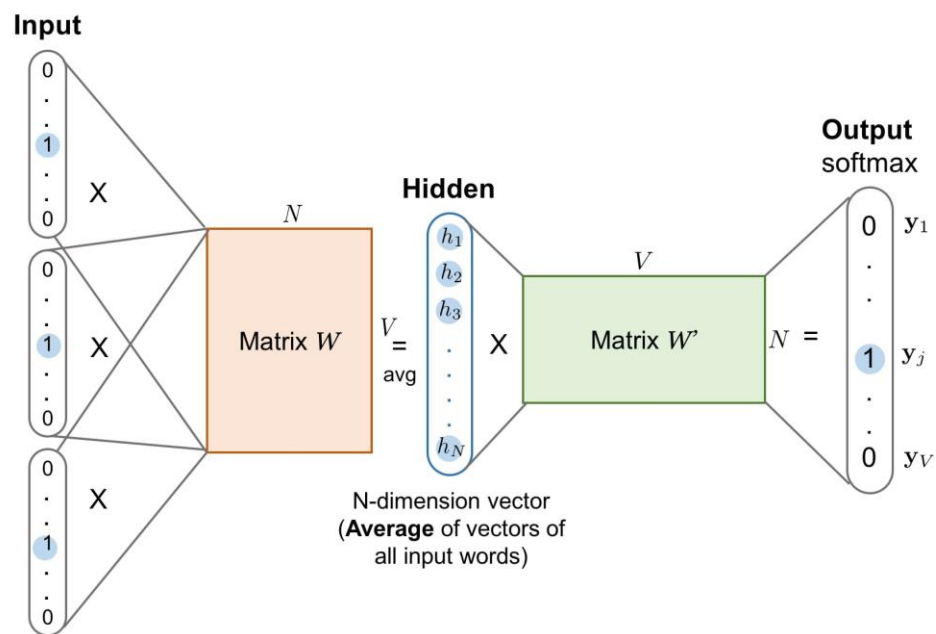
在前處理中，我們會使用到 NLTK 函式庫。首先，我們需要將一篇文章中的每一句話拆成一個又一個的符號（Token），這個步驟稱斷詞（Tokenization）。由於單詞的時態、單複數、停用詞（Stopword）對於文章整體所需傳達的大意影響不大，因此我們會刪除停用詞並且找尋他們其餘單詞的字根，然後一併將他們還原至最原始的型態；在 NLTK 中，我們需要找到每一個 Token 的詞性（Part-of-Speech Tagging），再根據其詞性進行「字形還原」（Lemmatization）。例如：「eating」與「ate」會變成「eat」，而「person」與「people」會被還原成「person」。接下來，我們打算使用卷積神經網路（Convolutional Neural Network）進行模型訓練；對於詞嵌入的實現，原本打算使用獨熱編碼（One-Hot Encoding），但是卻發現獨熱編碼有諸多特性以及缺點：

- 使用獨熱編碼會使大部分輸入層神經元為零，進而在訓練過程中導致神經元閒置，使計算資源過度浪費、以及使模型參數過多而導致模型過大，將因為 Heroku 的容量限制導致不易部署。
- 使用獨熱編碼將找不出單詞之間的關聯性，例如：「cat」、「dog」、「mug」，編碼為「100」、「010」、「001」，字義將完全無法被判斷。

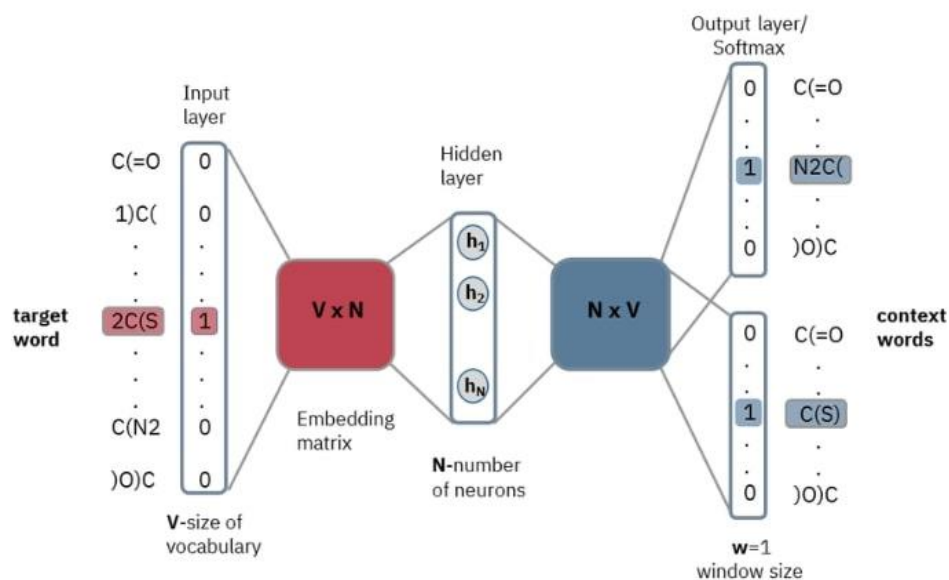
3.4 Word2Vec 與 Gensim

在發現獨熱編碼弊大於利之後，我們發現了一個更有效的技術，也就是使用 Gensim 函式庫以及 Word2Vec 模型架構進行詞嵌入。

Word2Vec 有其獨特的演算法，其演算法為：**CBOW (Continuous Bag-of-Words)** [1]以及 **Skip Gram**[2]，CBOW 透過上下文，也就是前後 n 個單詞，以預測出待預測的單詞（見圖三）；Skip Gram 則完全相反，以中間的單詞預測上下文前後 n 個單詞（見圖四）。兩個演算法都可以被分為三層 layer 的模型架構，第一層為 input layer，第二層為 projection layer（hidden layer），第三層為 output layer。

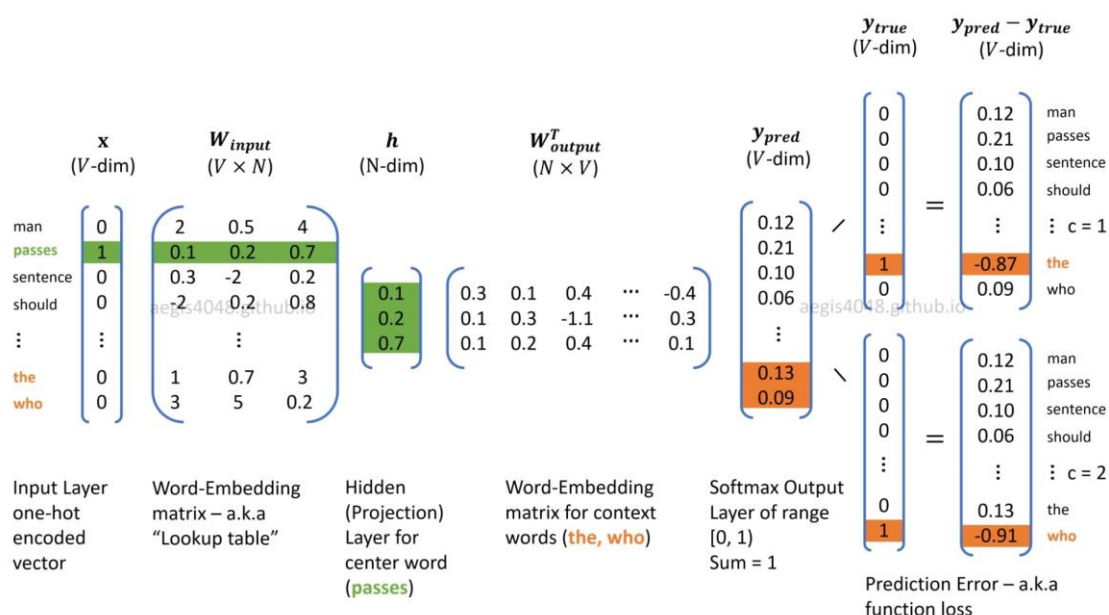


（圖三）CBOW 示意圖



（圖四）Skip Gram 示意圖

CBOW 架構有很多個平行的 input layer，每個 input layer 以及 output layer 的神經元數量皆為 token 個數；在 projection layer (hidden layer) 中，神經元數量為自訂的單字向量空間維度。Skip Gram 則與 CBOW 架構相反，input layer 與 output layer 對調，其餘架構相同。最後我們所使用的參數為 projection layer(hidden layer)到 output layer 之間的權重，用於詞嵌入使用[3]（見圖五）。



(圖五) Word2Vec 權重計算

Gensim 函式庫基於 Word2Vec 中的 CBOW 架構做實現，Gensim 中參數 window_size 為平行 input layer 的數量，vector_size 為 projection layer 中單詞轉換過後的單詞向量空間維度。Gensim 會將文章中出现過的每個單詞做獨熱編碼；也可以自訂門檻，將出現過一定次數的單詞納入單詞集合中。此外，Gensim 也可以將兩個字作為 input，其 output 為兩個單詞之間的相近程度；或將一個字作為 input，其 output 為 n 個相似度最近的單詞。在實作後發現，「boy」與「girl」相似度高達約 90%，原因是因為這兩個單詞容易被用在相同上下文之中（見圖六）；另一方面，「girl」與「mug」相似度約為 16%（見圖七）。

Similarity Score (boy, girl): 0.9053419828414917

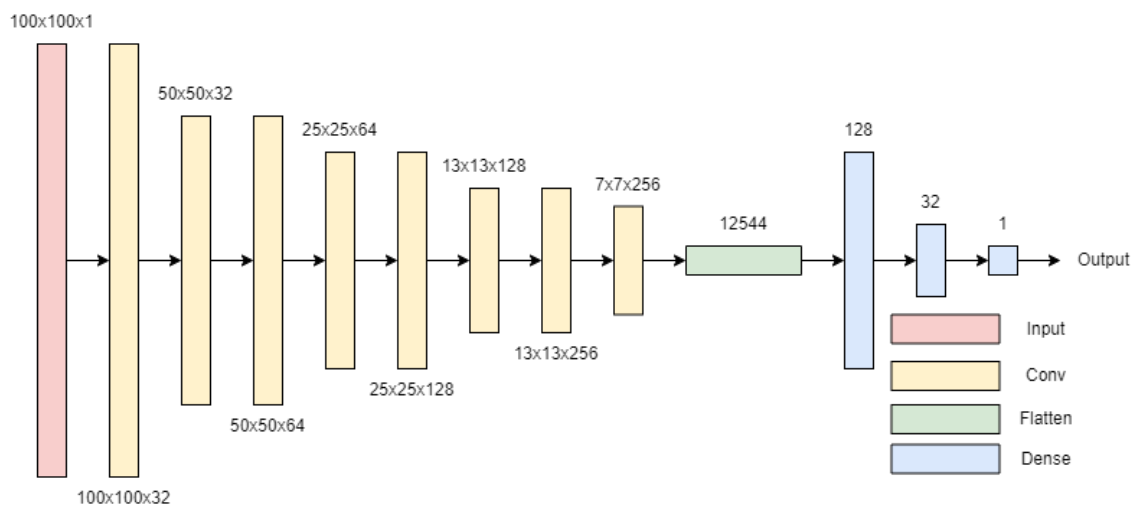
(圖六) Gensim - boy 與 girl 相似度

Similarity Score (girl, mug): 0.1603318601846695

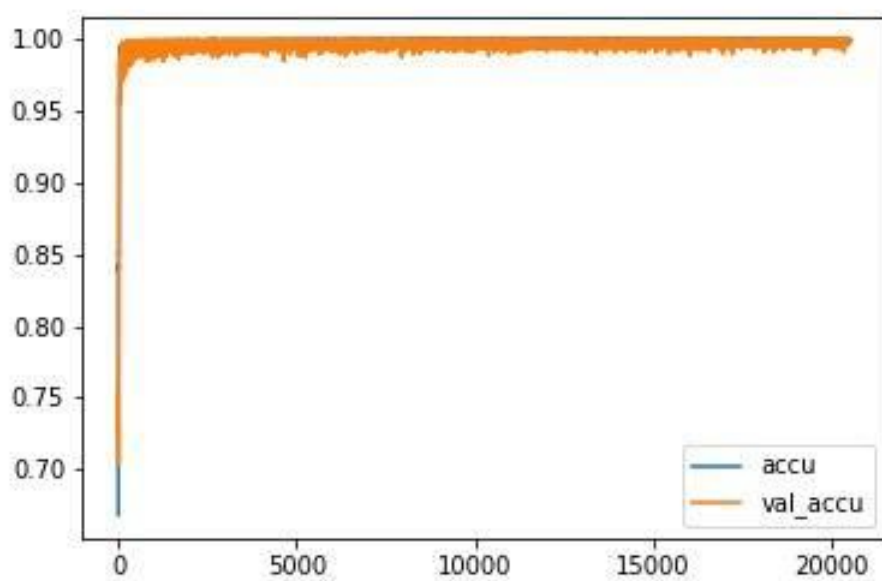
(圖七) Gensim - girl 與 mug 相似度

3.5 基於 Word2Vec 之 CNN 模型訓練

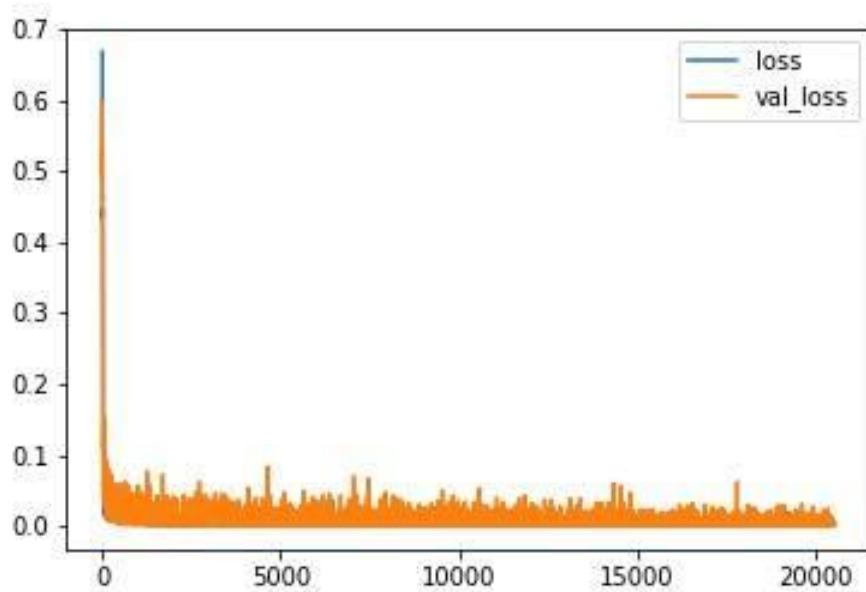
下一步是 CNN 的模型訓練，在進行多次嘗試之後，我們將 Gensim 中所設定的每個 vector_size 為 1×100 ，我們預估在自動化審文系統之待審文章中最多出現 100 個單詞，因此這個模型的 input layer 大小為 $100 \times 100 \times 1$ ，將每個矩陣當作一個「圖片」進行二元分類，在經過約 20000 個 epoch 過後將模型訓練出來。下圖為整體模型架構(見圖八)以及 Training history (見圖九、圖十)：



(圖八) 基於 Word2Vec 之 CNN 模型架構



(圖九) 基於 Word2Vec 之 CNN 模型 accuracy history



(圖十) 基於 Word2Vec 之 CNN 模型 loss history

在模型訓練之後，我們蒐集了測試資料集，資訊工程相關約 32 萬筆，非資訊工程相關約 11 萬筆，以下是基於此測試資料集的混淆矩陣（Confusion Matrix）並假設陽性為與資訊工程相關（見圖十一）：

Word2Vec Confusion Matrix	Positive (1)	Negative (0)
Positive (1)	TP 99.97% 323054 / 323166	FN 0.03% 112 / 323166
Negative (0)	FP 5.08% 5697 / 112236	TN 94.92% 106539 / 112236

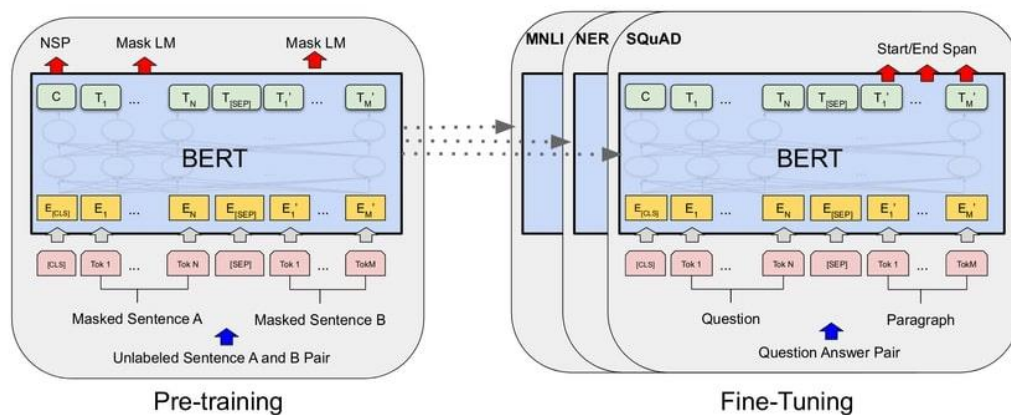
（圖十一）基於 Word2Vec 之 CNN 模型混淆矩陣

我們發現 Word2Vec 有一個致命的缺點，也就是 Word2Vec 無法判斷「一字多義」，即為 Static 的概念。例如：「Python」有「Python Language」與「蟒蛇」兩種意思，也就是當在文中出現「Snake」，模型很有可能會混淆並將此文章判定為與資訊工程相關（因為「Snake」與「Python」相關，「Python」與「資訊工程」相關）。

3.6 基於 BERT 之 CNN 模型訓練

在經過搜尋發現，Google 在 2018 年開發出了一款基於變換器的雙向編碼器表示技術（Bidirectional Encoder Representations from Transformers），簡稱為 BERT，此技術使用遷移學習（Transfer Learning）（見圖十二）以作為替代 Word2Vec 的功能。與 Word2Vec 相比，BERT 使用了較為動態的判斷，使 BERT 能夠以當下的單詞而判斷單詞之間的關聯性，即使是相同的單詞，也會因為上下文而使關聯性有所不同，

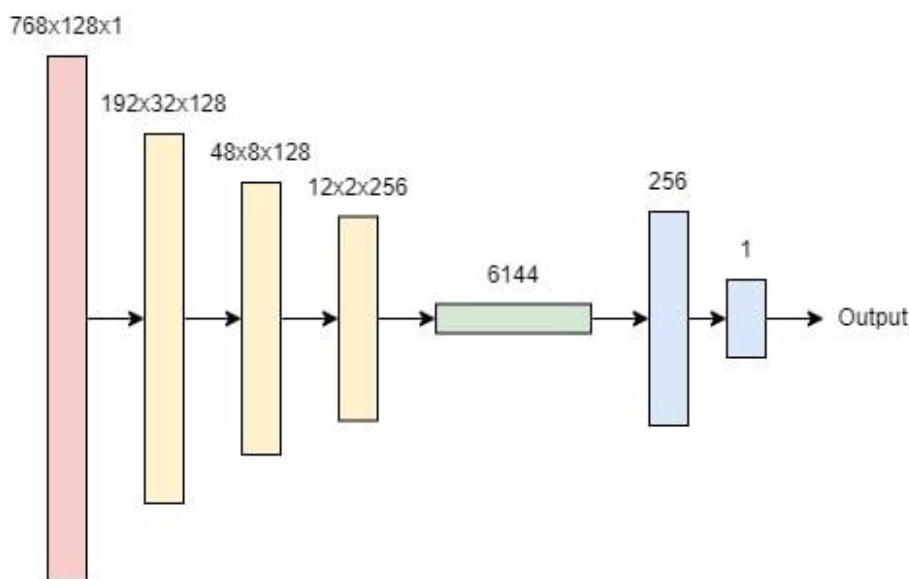
這是 Word2Vec 不能做到的。例如：「apple tree」與「banana tree」中的「tree」的關聯性為 89%，但是「apple tree」與「binary tree」中的「tree」的關聯性為 53%（見圖十三）。因此，我們也使用了基於 BERT 進行第二個版本的模型訓練（見圖十四）：



（圖十二）遷移學習示意圖

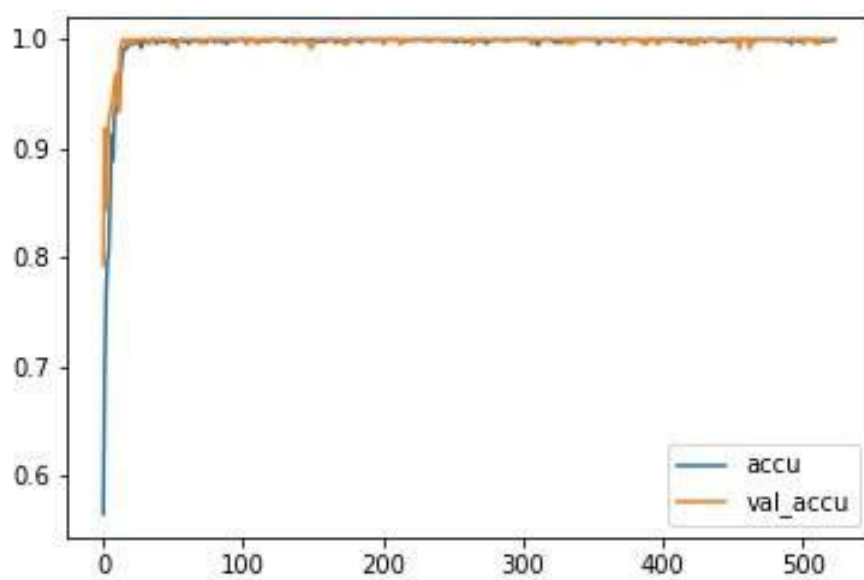
[[0.52670765]] 1: How are binary [trees] implemented in data structures? 2: Apples come from apple [trees].
[[0.52954316]] 1: How are binary [trees] implemented in data structures? 3: Bananas come from banana [trees].
[[0.8891155]] 2: Apples come from apple [trees]. 3: Bananas come from banana [trees].
[[0.7811103]] 1: How are binary [trees] implemented in data structures? 4: Which data structure is better? Stacks, heaps or [trees]?

（圖十三）BERT 動態單詞判斷

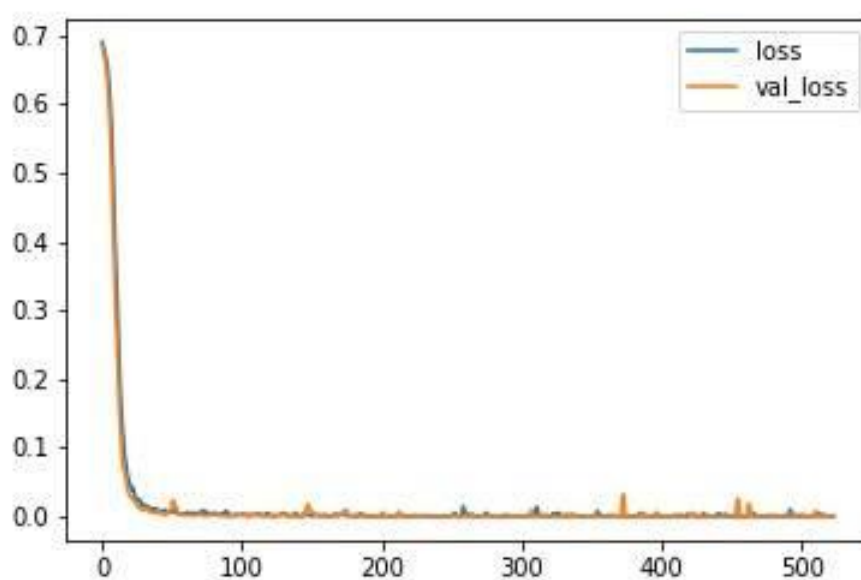


（圖十四）基於 BERT 之 CNN 模型架構

以下為基於 BERT 之 CNN 模型 Training history(見圖十五、十六)，
訓練了約 500 個 epoch 的結果：



(圖十五) 基於 BERT 之 CNN 模型 accuracy history



(圖十六) 基於 BERT 之 CNN 模型 loss history

我們使用了相同的測試資料集，以下是基於測試資料集的混淆矩陣並假設陽性為與資訊工程相關（見圖十七）：

BERT Confusion Matrix	Positive (1)	Negative (0)
Positive (1)	TP 99.72% 322261 / 323166	FN 0.28% 905 / 323166
Negative (0)	FP 1.24% 1397 / 112236	TN 98.76% 110839 / 112236

（圖十七）基於 BERT 之 CNN 模型混淆矩陣

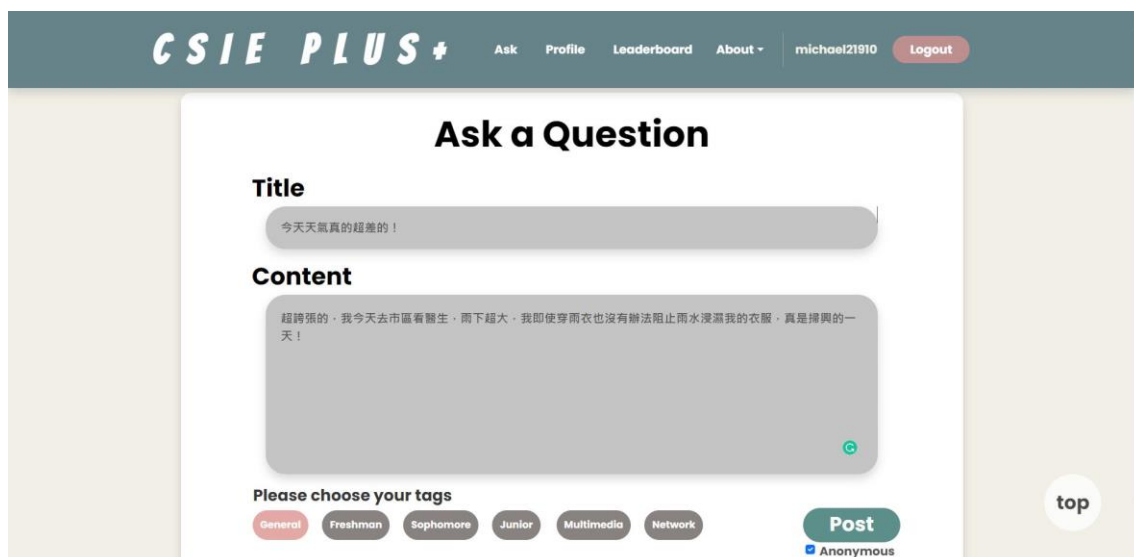
透過以上兩個混淆矩陣以及 Training history 可以發現，基於 BERT 之 CNN 模型雖然只訓練了約 500 個 epoch，但綜合表現而言，是優於基於 Word2Vec 之 CNN 模型的，並且兩模型收斂的差異程度也顯而易見。

3.7 網站架設與模型串接

我們打算將 Heroku 作為雲端主機並進行部署，因為基於 BERT 之 CNN 模型綜合準確率比基於 Word2Vec 之 CNN 模型高，因此我們打算使用基於 BERT 之 CNN 模型進行模型串接，卻在部署之後發現，記憶體的使用量遠遠超出 Heroku 所允許免費使用者的額度，因此，我們退而求其次，選擇基於 Word2Vec 的 CNN 模型做為部署到網站上的模型。

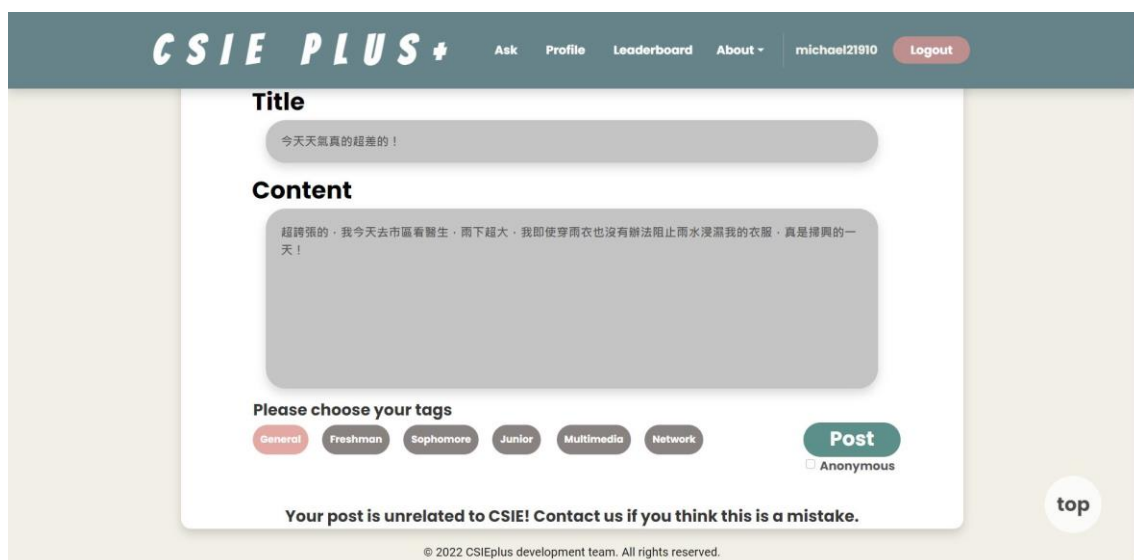
實現模型串接的方式為，使用者從前端以 HTTP POST Method 傳送資料到後端進行處理，在後端直接對文章內容進行資料前處理之後交給

基於 Word2Vec 之 CNN 模型做判斷，並將結果回傳以及將 threshold 設為 0.5，如果模型判斷之關聯度不小於 0.5，則發送成功（見圖十八、十九）；反之則跳出提示訊息，提示使用者必須發佈與資訊工程相關之文章（見圖二十、二十一）。



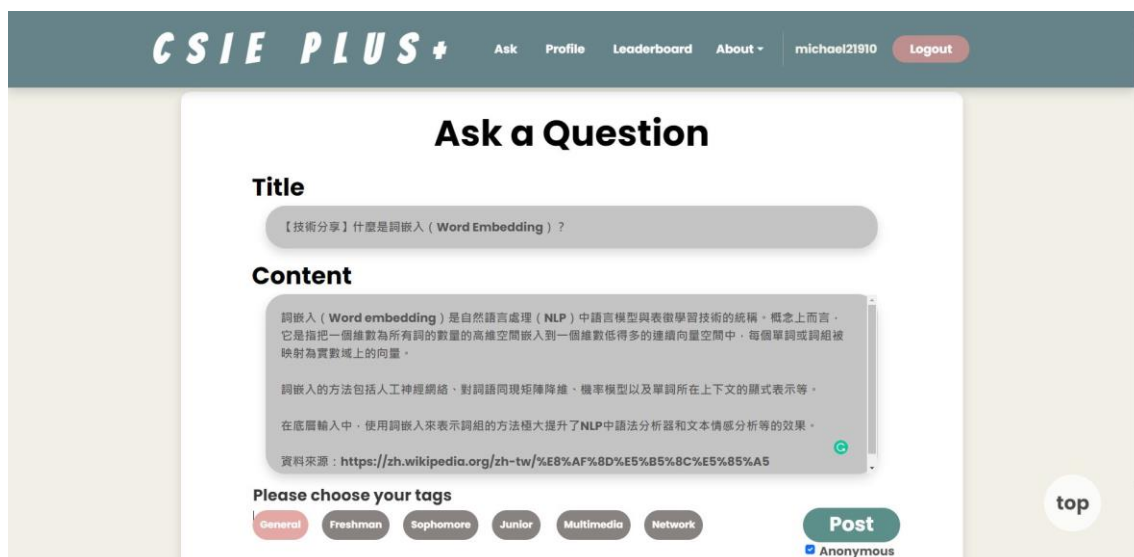
The screenshot shows the 'Ask a Question' interface on the CSIE PLUS+ website. The title field contains '今天天氣真的超難的！' and the content field contains '超誇張的，我今天去市區看醫生，雨下超大，我即使穿雨衣也沒有辦法阻止雨水浸濕我的衣服，真是掃興的一天！'. The 'Please choose your tags' section shows 'General' selected. The 'Post' button is visible, and the 'Anonymous' checkbox is checked. A 'top' button is on the right.

（圖十八）與資訊工程不相關之文章



The screenshot shows the same 'Ask a Question' interface as Figure 18, but with an error message at the bottom: 'Your post is unrelated to CSIE! Contact us if you think this is a mistake.' The 'Post' button is disabled, and the 'Anonymous' checkbox is unchecked. The 'top' button is on the right.

（圖十九）發佈失敗範例



(圖二十) 與資訊工程相關之文章

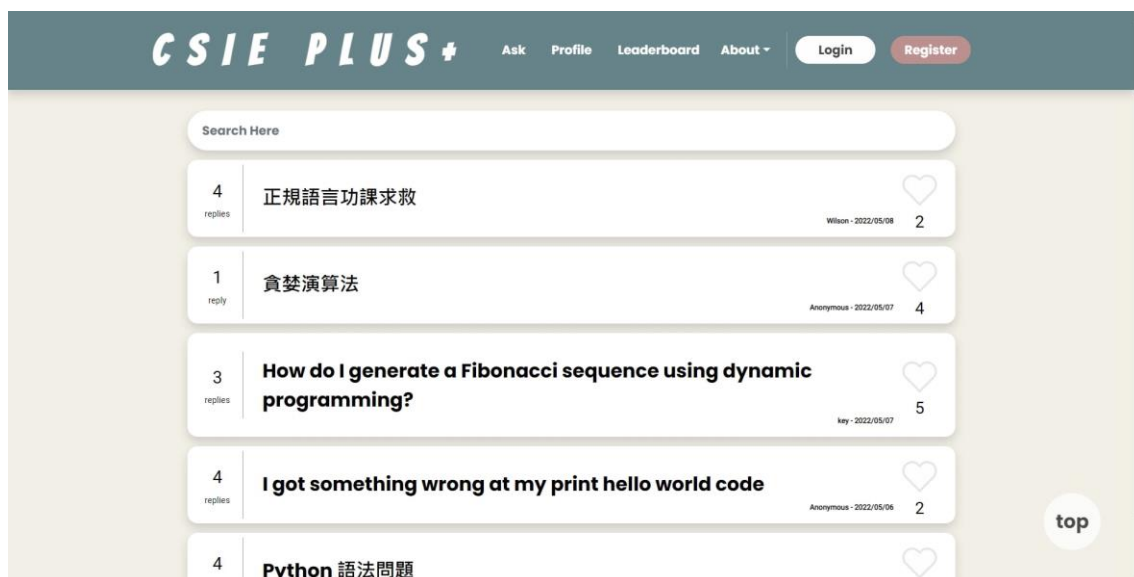


(圖二十一) 發佈成功範例

為了保持網頁的完整度，我們同時也建立了以下系統，以確保每位使用者的帳號安全以及保護網站不會輕易的被攻擊，包含：

- 登入系統
- reCaptcha Checkbox
- 信箱驗證系統
- 密碼加密 (SHA-256)

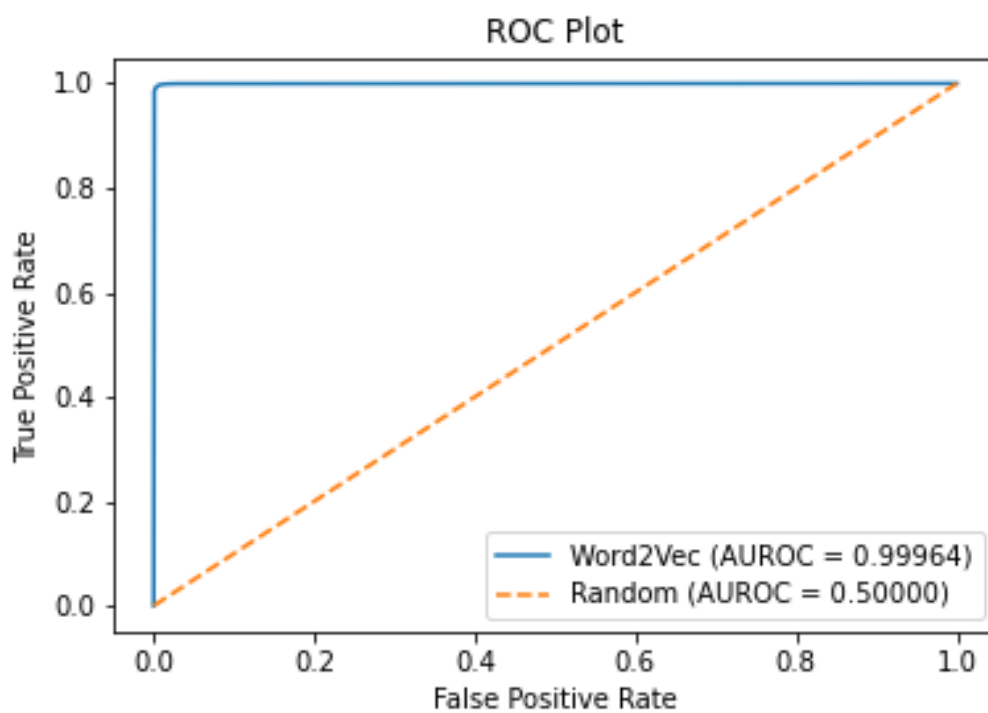
以上為模型串接的實現與網站基本功能之介紹，也是「NDHU CSIEplus」的部分樣貌（見圖二十二）。



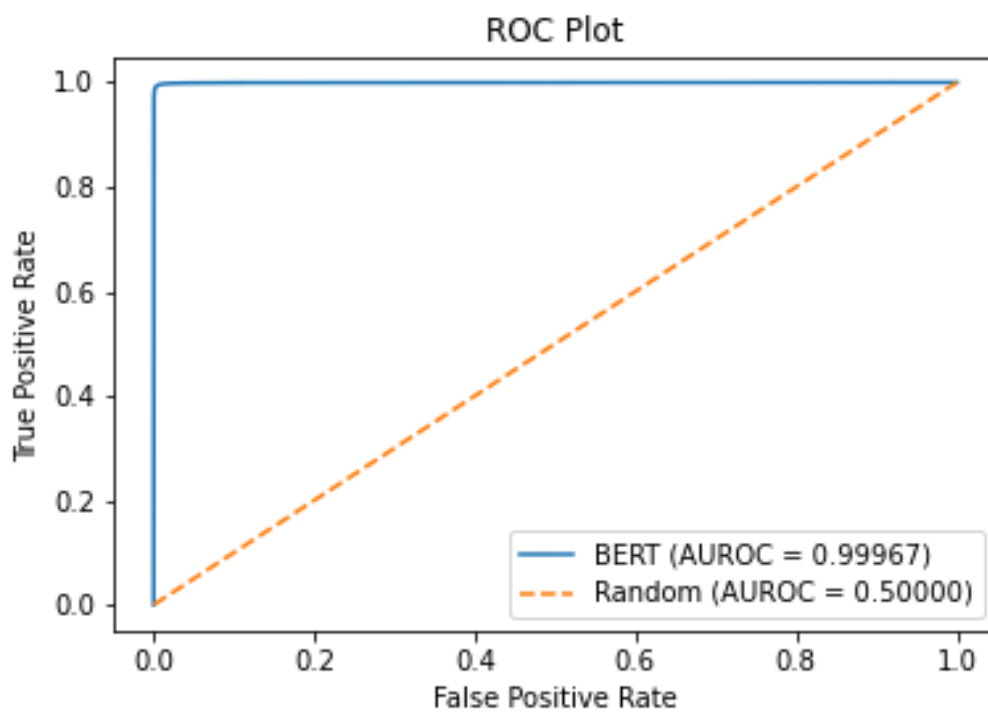
(圖二十二) NDHU CSIEplus 網站

3.8 證明模型可用性

為了證明基於 Word2Vec 之 CNN 模型以及基於 BERT 之 CNN 模型好壞，我們繪製了感受性曲線 (Receiver Operating Characteristic Curve)。結果顯示，不論是何種模型，都有傑出的表現 (見圖二十三、二十四)。基於 Word2Vec 之 CNN 模型的 AUROC 為 0.99964，基於 BERT 之 CNN 模型的 AUROC 為 0.99967。



(圖二十三) 基於 Word2Vec 之 CNN 模型 ROC 曲線



(圖二十四) 基於 BERT 之 CNN 模型 ROC 曲線

四、 結果及討論 Research Results and Conclusions

4.1 Word2Vec 以及 BERT 的比較

首先，從 Word2Vec 以及 BERT 的根本出發，對於 Word2Vec 而言，是沒有辦法分辨一字多義的單詞，因此，手術的「Operation」在被還原成「Operate」之後，極有可能與作業系統的「Operating」相互混淆，這是 Word2Vec 的缺點；另一方面，BERT 可以非常好地做到判斷一字多義的單詞，即使是少了一個標點符號，關聯度也會有所改變，更不用說是上下文了。

此外，關於容量大小，Word2Vec 所占的空間比 BERT 少非常多，在 BERT 當中分成 BERT-base 以及 BERT-large，即使是 BERT-base 也有高達一億一千多萬個參數；相對於 Word2Vec 而言，使用到的運算資源將減少非常多。

在圖九、十、十五以及十六中，展示了基於 Word2Vec 的 CNN 模型以及基於 BERT 的 CNN 模型之 training history graph，我們可以很輕易地發現，基於 Word2Vec 的模型在達到高達兩萬多個 epoch 之後仍然無法完全正確判斷文章的類別，但是 BERT 卻在 500 多個 epoch 左右就成功做到了這件事。

關於基於 Word2Vec 之 CNN 模型中，我們研究了 False Negative（非資訊工程文章判斷錯誤）的主要原因是：一字多義的單詞所造成的謬誤；在那些文章中多次提到許多與資訊工程相關的單詞，但是那些單詞的意思卻不是與資訊工程相關的意思，我們統整後發現極大多數的文章都有這樣子的趨勢。

4.2 模型評價

總體而言，我們對基於 BERT 所訓練出來的 CNN 模型評價為：雖然會花費掉非常多的運算資源，我們依然對於準確率感到相當滿意；另一方面，我們對基於 Word2Vec 所訓練出來的 CNN 模型評價為：雖然整體準確率不及 BERT，但是這樣的準確率還可以接受，也因為與 BERT 相比相當的輕量，使我們能夠成功開發出一個實際應用的平台，以供大家使用。

4.3 結語

隨著網路科技的發展以及社群媒體的普及，個人理念與知識的傳播無遠弗屆，不再受限於地理意義上的遠近，可是文章品質的保證與發文的便利性如同魚和熊掌一般，難以兼得。目前多數社群媒體以及問答平台均使用人工審核文章的方式以取得兩者之間的平衡，不過小編畢竟難以負荷龐大的文章數量，因此必須進行一場審文機制的工業革命，而自然語言處理與深度學習能夠用以解決此方面的問題。

自然語言處理與深度學習的應用價值是相當廣泛的，因此非常值得探討，而我們這項專題的目標就是深入研究這些領域的專業知識，並且實現一個自動審文系統。

為了確保模型的品質與準度，在實作專題的過程當中，我們經常翻閱相關書籍與其他學者的論文研究，除了吸取相關技術的實作方式以及原理之外，也從每一次的錯誤之中記取教訓並改善模型，使模型對於測試集之準確率越來越接近我們的期望。

五、 参考文献 References

(一)、 English

- [1] Weng, L. (2017, October 17). *Learning Word Embedding / Lil'Log*. Lilianweng.
<https://lilianweng.github.io/posts/2017-10-15-word-embedding/>
- [2] Öztürk, H., Ozgur, A., Schwaller, P., & Laino, T. (2020, February). *Exploring Chemical Space Using Natural Language Processing Methodologies for Drug Discovery*. Researchgate.
https://www.researchgate.net/publication/339013257_Exploring_chemical_space_using_natural_language_processing_methodologies_for_drug_discovery
- [3] Kim, E. (2019, May 6). *Demystifying Neural Network in Skip-Gram Language Modeling*. Aegis4048.
https://aegis4048.github.io/demystifying_neural_network_in_skip_gram_language_modeling
- [4] Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018, October 11). *BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding*. Arxiv.
<https://arxiv.org/pdf/1810.04805.pdf>
- [5] rare technologies. (2009). *Gensim Topic Modelling for Humans*. Radimrehurek.
<https://radimrehurek.com/gensim/models/word2vec.html>
- [6] Nltk team. (2001). *Natural Language Toolkit*. Nltk.
<https://www.nltk.org/>
- [7] Tensorflow. (n.d.). *Word2vec / Tensorflow Core*. Tensorflow.
<https://www.tensorflow.org/tutorials/text/word2vec>

(二)、中文

[8] Yeh, J. (2017, October 22). [資料分析&機器學習] 第 3.2

講：線性分類-感知器(*Perceptron*) 介紹. Medium.

<https://medium.com/jameslearningnote/%E8%B3%87%E6%96%99%E5%88%86%E6%9E%90-%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-%E7%AC%AC3-2%E8%AC%9B-%E7%B7%9A%E6%80%A7%E5%88%86%E9%A1%9E-%E6%84%9F%E7%9F%A5%E5%99%A8-perceptron-%E4%BB%8B%E7%B4%B9-84d8b809f866>

[9] 劉詩昆. (2018, July 31). 如何理解空洞卷積 (*dilated Convolution*) . 程式前沿.

<https://codertw.com/%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80/599845/>