小說推薦系統

目錄

研究動機與目的

研究方法

系統實驗

效能評估

結論

參考文獻

研究動機與目的

得益於資訊發達，現代人做選擇常常依賴於網路推薦，不僅看電影要先看 評分，吃餐廳吃平價好的，小說自然也不例外。但是，隨著時間發展，各 種評論與推薦卻又變得更加商業化，常常搜尋「推薦小說」，所得的結果都 是各種奇怪的部落格，出版商推薦的也都是注重於該書的商業價值，想真 正的找到有用的推薦都只能去各大論壇搜尋。

而小說推薦時常常都是主觀撰寫的推薦文，常常得要多看好幾篇才能找到 「好像真的挺好看」的小說，更有時要跑遍各大論壇才能找到真正想看的 小說，費時又費力。

這時就希望有個程式能幫忙蒐集各大論壇的討論，簡化親自搜索的時間， 並從這些討論中，判斷該書是否推薦。

研究方法

Bert

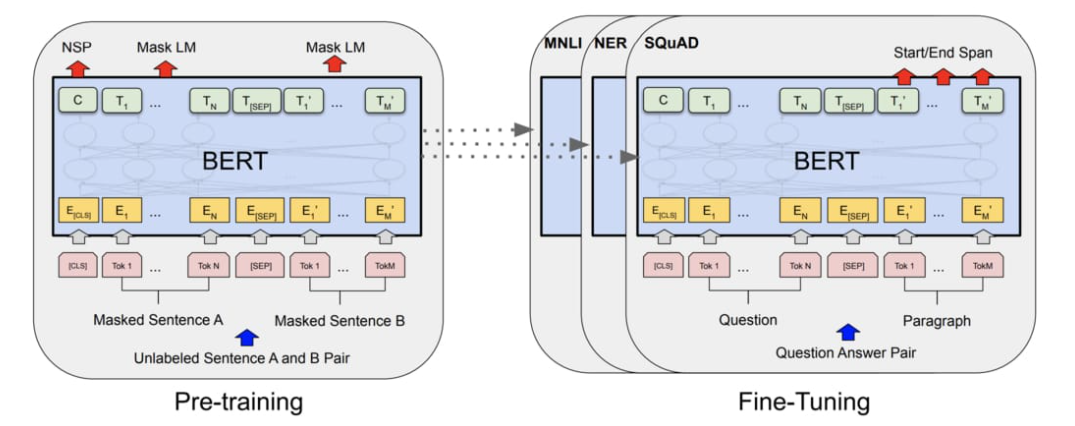
是Google以無監督的方式利用大量無標註文本煉成的語言代表模型， 其架構為Transformer 中的Encoder，我們利用bert進行句子情緒預 測。

原理：

利用兩階段遷移學習

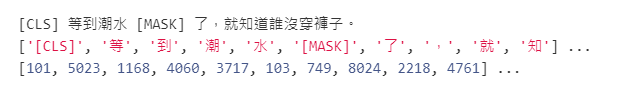
第一階段：先以 LM Pretraining 的方式預先訓練出一個對自然語言 有一定「理解」的通用模型

第二階段：再將該模型拿來做特徵擷取或是fine tune下游的（監督 式）任務

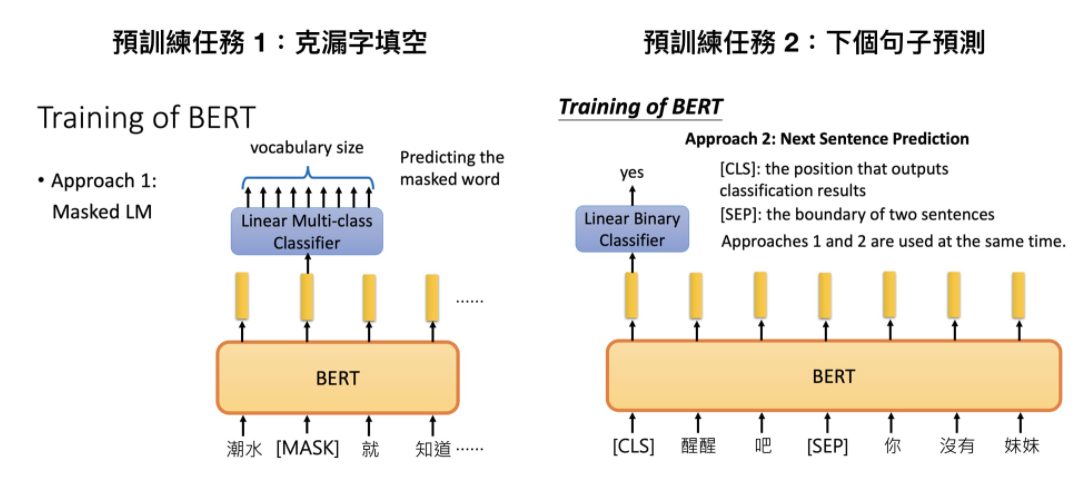


第一階段流程

將資料進行編碼，並加入特殊的token

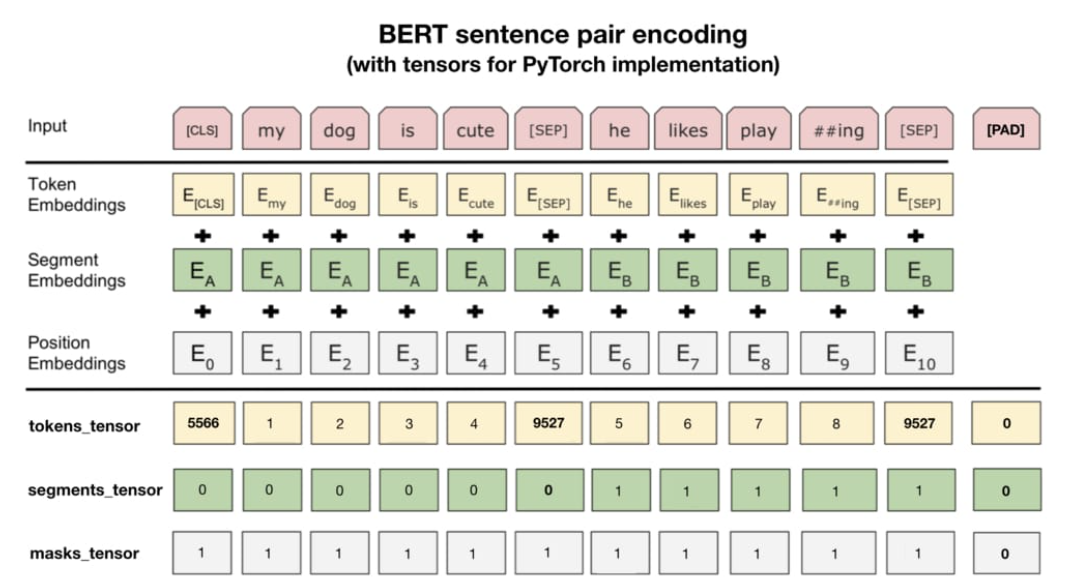


進行預訓練任務，為克漏字填空和下個句子預測

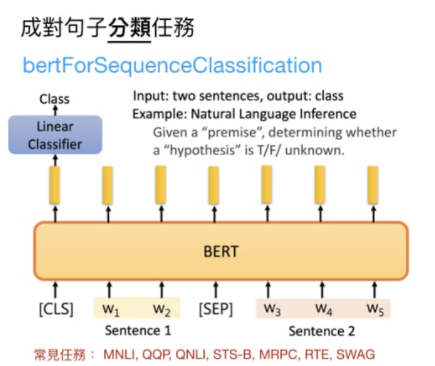


第二階段流程

準備文本數據並轉為bert相容的輸入格式



加入新layer做下游任務，像是加入新的線性分類器

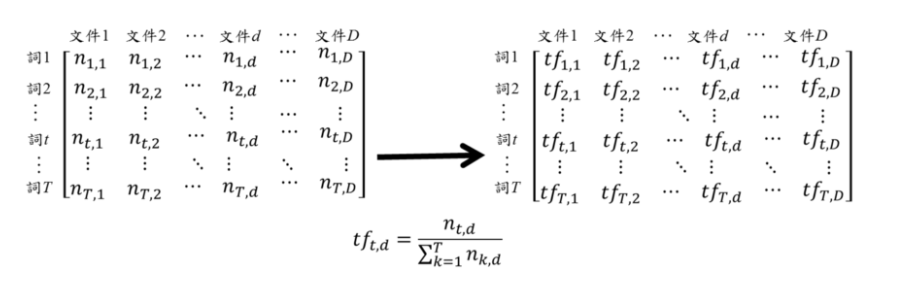


TF-IDF

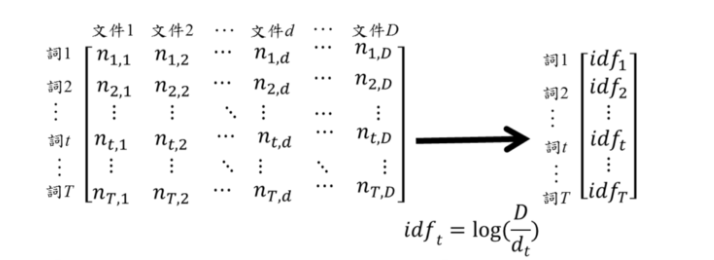
是一種用於資訊檢索與文字探勘常用的加權技術，用來評估單詞對於文件 的集合或詞庫中一份文件的重要程度，我們利用TF-IDF萃取關鍵字

原理：

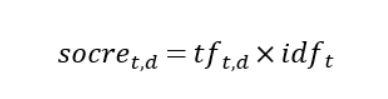
TF：處理每一個「文件」中所有「詞」的問題，也就是統計文章中詞 的出現頻率，我們將所有的「詞」在每個文件中都各自標準化，讓 「詞」在各自「文件」中出現只剩下頻率



IDF：處理每一個「詞」在所有「文件」中的問題，也就是統計詞在每 篇文章出現的頻率，「詞」在越多「文件」中出現，代表相對應的IDF 會比較小，也就是這個「詞」可能沒什麼用



最後計算分數，判斷是否為關鍵字



系統實驗

我們整體的系統架構分為三個部分

1. 取得該小說的評論

2. 分析評論之情緒(正面 or 負面)

3. 找出該小說的關鍵字(ex:戀愛、奇幻)

第一部分

我們使用python的網路爬蟲模組beautifulsoup，來爬取ptt論壇的資料



圖 爬蟲程式碼

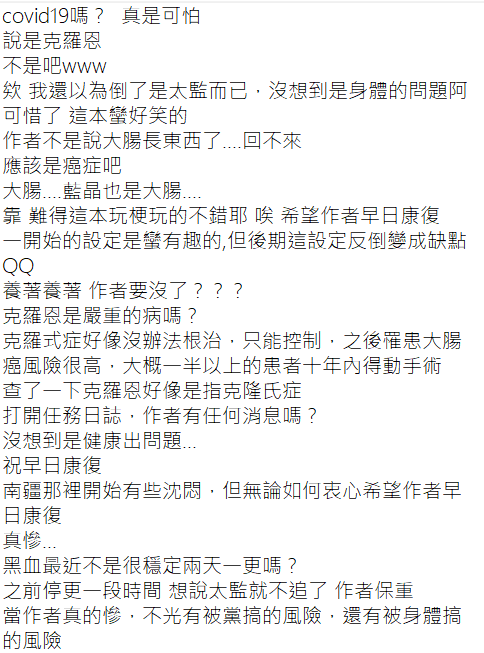


圖 小說ptt留言文字檔

第二部分

使用自製的模型，對評論進行情緒預測，輸入一個句子，會輸出正面或反 面

模型訓練：

首先，我們先準備我們訓練的資料，訓練資料總共為9600筆，而測試 資料為1200筆

接著我們對資料進行編碼，變為符合bert的輸入格式，總共會回傳4 個tensor ，input\_id為編碼後的句子，label為資料的正反情緒，0 為負面，1為正面。

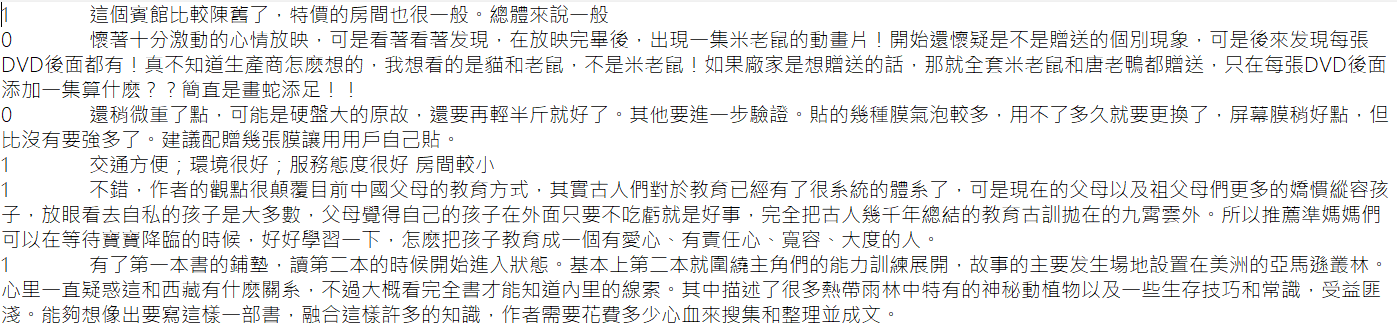


圖 訓練資料

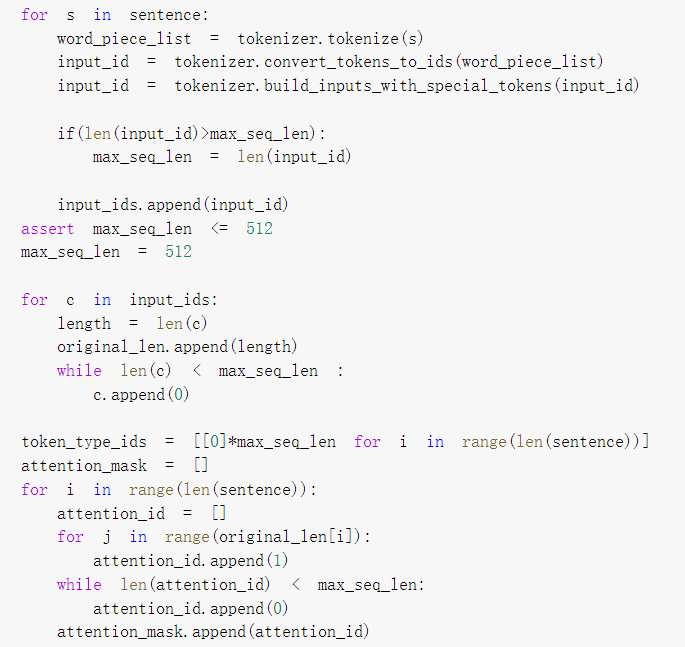


圖 資料編碼程式碼

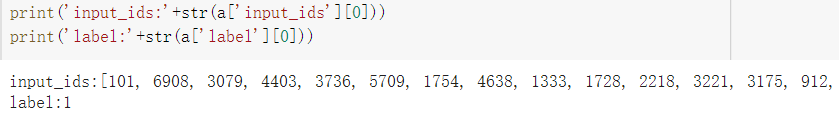


圖 編碼結果

開始準備訓練我們的模型，我們利用bert fine-tuing模型- BertForSequenceClassification進行訓練，此模型的in\_features為 768、out\_features為3，dropout為0.1，而我們總共會訓練3次，並記 錄loss和準確率，我們可以看到，最後訓練的準確率為97.396。

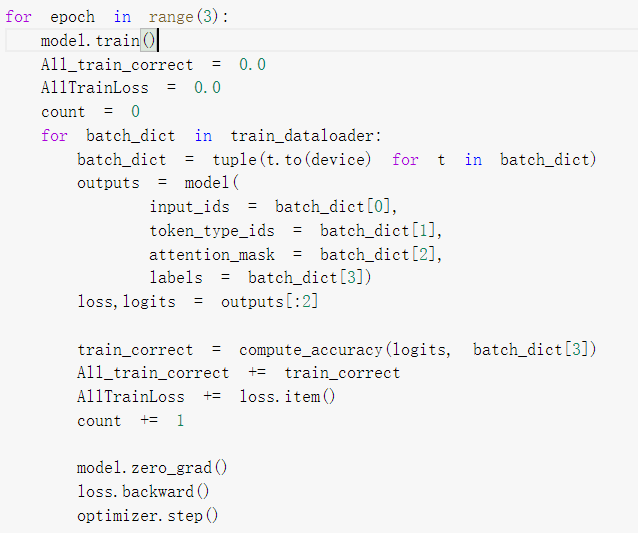


圖 訓練過程

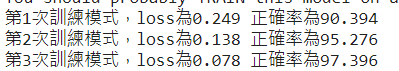


圖 訓練結果

進行預測：

我們將爬蟲取得的資料利用上面的方式進行編碼後，放入模型，就會 預測該評論是正面還是反面，最後統計總評論數和正面評論，計算出 分數，最高為5，最低為0



圖 預測程式碼

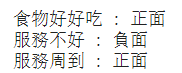


圖 預測範例

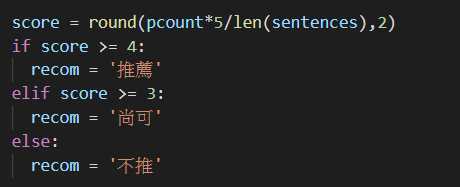


圖 評分系統

第三部分

使用TF-IDF演算法得出每則評論的關鍵字，我們使用jieba套件內建的函 式-jieba.analyse來取得，最後輸出出現次數最多的字詞

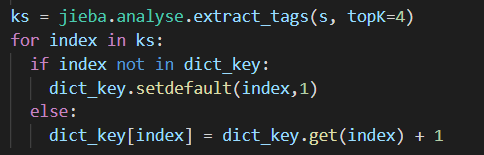


圖 關鍵字程式碼

效能評估

Bert模型效能

經過最終的訓練後，我們模型的正確率為97.396，準確率算是滿高 的，但對於ptt留言預測，我們人工比對後，發現有以下2個問題

一：ptt留言常常會出現只有人才會懂的流行用語，像是母湯、小朋 友才做選擇之類的，模型辨識的效果就不好

二：ptt的評論有時會斷為好幾行，使得模型讀入的評論被切斷，預 測的結果也會不準

關鍵字評估

因為用的是專業團隊提供的語料包，所以效能方面不會有問題，但對 於ptt整體留言來說，討論範圍過大而且用語過雜，關鍵字的出現頻 率往往只有1次到3次，因此我們評估萃取該小說之關鍵字在ptt上 有難度。

整體評估：

我們的目標為找到真正好看的小說，因此從ptt來爬取評論，事實 上，ptt 的留言通常都比較負面，也會和其他人爭吵，導致小說的評 分被拉的很低。雖說整體的評分偏低，但以相對的分數來說還是能找 到較好看的小說。

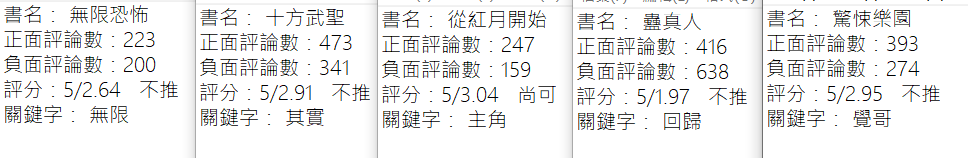


圖 部分預測結果



圖 網頁顯示結果(評分區間為0-1)

結論

本次專題的結果雖沒有預想中理想，但模型在預測情緒上，準確率是相當 高的，如果能使用專門對應ptt留言的訓練集來訓練，相信預測效果會更 加準確。

在應用方面，此系統不單單僅能應用在小說上，更可以推廣到美食或者影 視相關的領域。也能在選舉時，觀察各大論壇對於政見支持與否，來調整 競選團隊策略或發言，以便於更有效的取得民眾支持。

若以不同角度分析，還能向下分析到單一帳號的發文、回復的傾向與種 類，來做到個人化廣告或交友推薦的應用。

參考資料

<https://medium.com/pyladies-taiwan/%E6%B7%B1%E5%BA%A6%E5%AD%B8%E7%BF%92%E6%96%B0%E6%89%8B%E6%9D%91-pytorch%E5%85%A5%E9%96%80-511df3c1c025>

<https://leemeng.tw/attack_on_bert_transfer_learning_in_nlp.html>

<https://www.youtube.com/watch?v=UYPa347-DdE&ab_channel=Hung-yiLee>

<https://github.com/pengming617/bert_classification>

<https://leemeng.tw/attack_on_bert_transfer_learning_in_nlp.html>

<https://github.com/UDICatNCHU/udic-nlp-API>