巨量資料分析研討專題

資管研一 邱鈺珍

# 研究議題

電影評論情感分析(sentiment analysis)

使用技術：深度學習(Deep Learning)/自然語言處理(Natural Language Processing)

前言：

* NLP定義

依據維基百科NLP的定義，自然語言處理（NLP）是計算機科學以及人工智慧的子領域，專注在如何讓計算機處理並分析大量（人類的）自然語言數據。NLP 常見的挑戰有語音辨識、自然語言理解、機器翻譯以及自然語言的生成。

一張含有 文字, 電池 的圖片

自動產生的描述

* NLP相關應用



* 演算法

之前自然語言的處理主要是使用【循環神經網路】(Recurrent Neural Network, RNN)，這幾年在演算法有長足的進展，從簡單的RNN、LSTM、GRU開始，後來有【注意力】(Attention)的補強，之後又有 Transformer 出現，相繼有 BERT、ELMO、XLNet、GPT ...等，甚至 BERT 也支援中文了，最近**的Simple Transformers** 更是值得注意的趨勢 。

* NLP術語

NLP 需要解決的第一個問題是將文字實例的集合轉換成矩陣形式，其中每一列都是文字實例的數值表示，即向量。

**分詞**是NLP的基礎任務，分詞就是將句子、段落、文章這種長文本，分解為以字詞為單位的資料結構，方便後續的處理分析工作。

舉例如下所述:

token(符號)：包括單詞和標點

tokenization(分詞)：我是台灣人->['我', '是', '台灣人']。

分詞的原因是因為要將複雜問題轉化為數學問題，而詞是一個比較合適的細微性。

中英文分詞有3個典型區別：

1. 分詞方式不同，中文更難；
2. 英文單詞有多種形態，需要詞性還原和詞幹提取；
3. 中文分詞需要考慮細微性問題。

根據 GitHub 上的 star 數：

(1)中文分詞工具排名如下

Hanlp/Stanford分詞/ansj分詞器/哈工大LTP/KCWS分詞器/jieba/IK/清華大學THULAC/ICTCLAS。

(2)英文分詞工具排名如下：Keras/Spacy/Gensim/NLTK。

* 概念說明

文字型態的資料，是沒有辦法如同數值資料一樣進行 Forward Propagation 以及 Backward Propagation 的，是沒有辦法微分的，簡單來講，是沒辦法進行運算並且訓練權重網路的。為了要解決這個問題，我們需要將『文字』轉換成『數值』。最簡單的方法，就是將每個『相異字』(Character) 轉換成一個特定的數字。這樣一來，我們便可以將機器學習相關的技術應用在文字領域了。

不過，只是單純地轉換 Character，有時是得不到較好的結果的。對中文而言，有時使用『詞』作為句子裡的基本元件單位會更恰當；另外，只是轉成一個『數字』也很難表現出中文詞彙的多樣性，轉換成『向量』通常效果會更好一點。

因為 Neural Network 是以矩陣運算為主，必須固定長度，所以，訓練前先調整輸入的句子，過長就截掉，過短就補空白。(padding)

# 研究方法

本研究使用50000筆電影評論的資料集，分成訓練資料80%及測試資料20%。透過資料預處理、詞嵌入、模型預測等步驟來完成此系統。本研究基本流程如圖2.1，分為預處理、詞嵌入、預測模型。

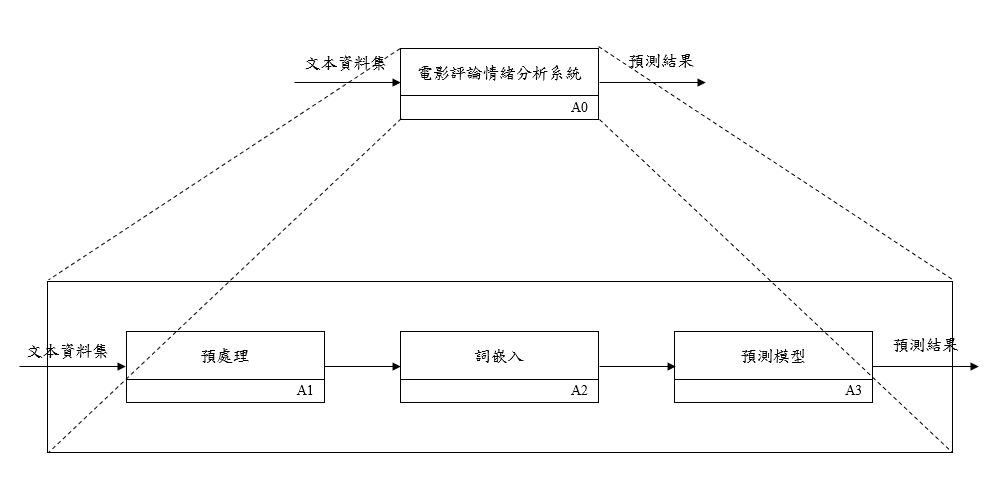


圖2.1 電影評論情緒分析系統架構圖

1. 預處理

使用pandas讀取 DataSet(CSV格式)將50000筆原始文本資料載入。

所謂的預處理，即是將原始資料(raw data)中遺漏或不需要的資料去除的過程。將這些不能使用的資料去除後，我們還要將剩下的資料進行整理。

本專題(程式<一般作法>)會進行以下處理

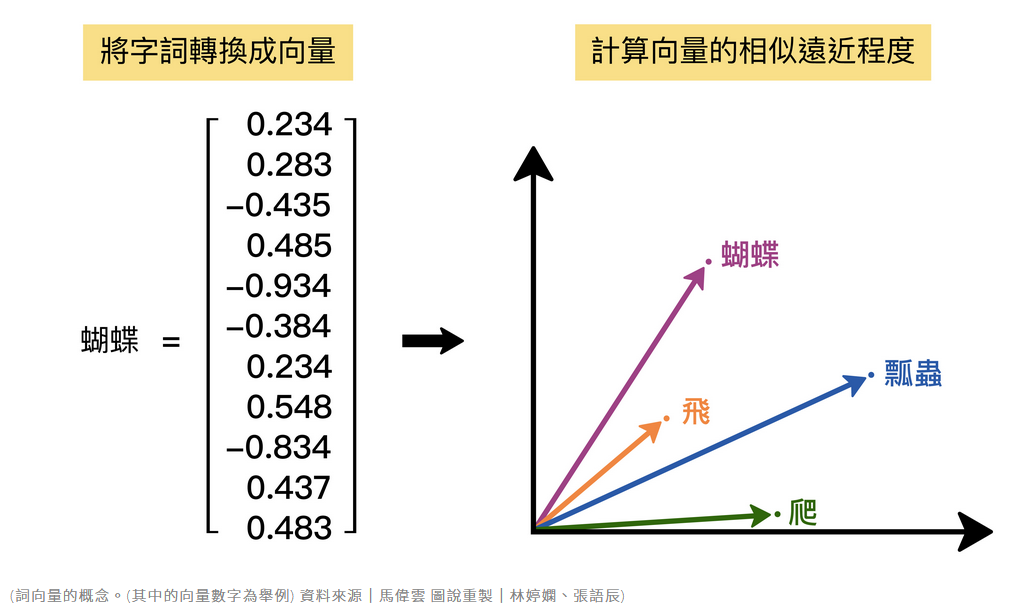
1. 判斷文本是否有非字符串格式
2. 剔除該行是NULL值
3. 英文字母 大寫 轉換成 小寫
4. 去掉英文的特殊字符(PUNCTUATIONS)-> !"#$%&\'()\*+,-./:;<=>?@[\\]^\_`{|}~
5. 刪除停用詞(ex: “is”、“the”、“and”、“so”、“my”等)
6. 詞幹/詞條提取(Stemming and Lemmatization)
7. 透過[詞雲]了解詞之分佈(如下圖)
8. 計算每一行文本的字數分佈圖
9. 拆分資料集(訓練集80% 測試集20%)
10. 再次確定每條文本轉換成一個字符串
11. 透過KERAS的Tokenizer功能將文字分詞化且轉成數字

一張含有 文字 的圖片

自動產生的描述

1. 詞嵌入

將編碼後的文本列表進行詞嵌入(Embedding)處理，並輸出的資料經過處理，轉換為向量形式。

詞向量示意圖：

1. 預測模型

準備訓練及測試資料

1. 訓練資料80% (40000筆)
2. 測試資料20% (10000 筆)

**專題程式分為兩種技術分別撰寫：**

**(1) 一般作法: keras.Sequential model/Tokizer/Embedding**

**(2) 最新做法: Simple Transformers/Bert**

**本研究建構的兩種深度學習預測模型，研究方法分述如下。**

模型一之架構及參數如圖2.2~2.3，總共架設4層。模型一起初的輸入為10000個長度為150的詞向量組合，經過一個embrdding層(依據Keras規範必須放在第一個位置)、一個全局平均池化(GlobalAveragePooling1D)層。接著進入Dense層，Activation Function 設定為 relu。最後進入最後Dense層，Activation Function 設定為sigmoid。

一張含有 文字 的圖片

自動產生的描述

圖2.2 程式架構(模型一)

一張含有 文字 的圖片

自動產生的描述

圖2.3 架構summary圖(模型一)

模型二採用Simple Transformers 框架，其創建模組的程式架構如圖2.4，訓練模組的程式架構如圖2.5。

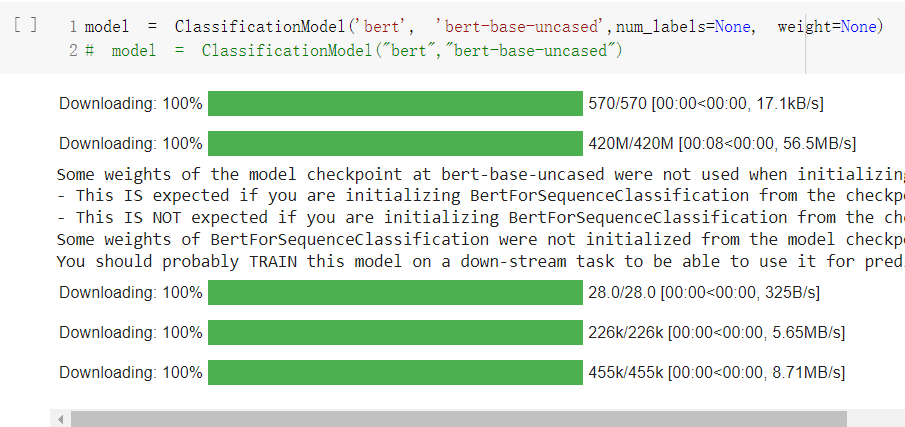


圖2.4 創建模組圖(模型二)



圖2.5 訓練模組圖(模型二)

# 資料來源

1. 資料來源

資料來源是 IMDB\_Dataset.csv (<https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews>)

一張含有 文字 的圖片

自動產生的描述一張含有 桌 的圖片

自動產生的描述

1. 資料集描述

關於此數據集：

IMDB 數據集包含用於自然語言處理或文本分析的 5萬筆電影評論。

這是一個用於二元情感分類的數據集，包含比以前的基準數據集更多的數據。此資料集包含兩個欄位，包含欄位名稱分別是review(電影評論)和sentiment(情緒正負面)。

本資料集包含25000筆正評電影評論；25000筆負評電影評論。



有關更多數據集信息，請通過以下鏈接，<http://ai.stanford.edu/~amaas/data/sentiment/>。

# 具體分析結果

本專題程式分為兩種技術分別撰寫：

1. 一般作法: keras.Sequential model/Tokizer/Embedding
2. 最新做法: Simple Transformers/Bert

**本研究建構的兩種深度學習預測模型，分析結果分述如下4.1~4.3節。**

1. 模型一分析結果

如圖2.6所示，本模型共執行了10個Epoch，最後訓練資料準確度0.9534、測試資料準確度0.8765。

比較執行9個Epoch和10個Epoch後的測試準確度(val\_accuracy)，可以發現10個Epoch(0.8765)比9個Epoch(0.8798)的結果還要低。然而，觀察執行9個Epoch和10個Epoch後的訓練準確度(accuracy)，可以發現10個Epoch(0.9534)比9個Epoch(0.9498)的結果還要高。這代表執行到10個Epoch時，可能已經有過擬合的情形發生。

圖2.7～圖2.8是**繪製訓練和驗證準確度/訓練和驗證損失**，以紅線代表訓練資料、藍線代表測試資料。如圖2.7顯示訓練資料和測試資料，在不同Epoch下的準確度曲線變化。圖2.8，顯示了訓練資料和測試資料，在不同Epoch下的Loss曲線變化。

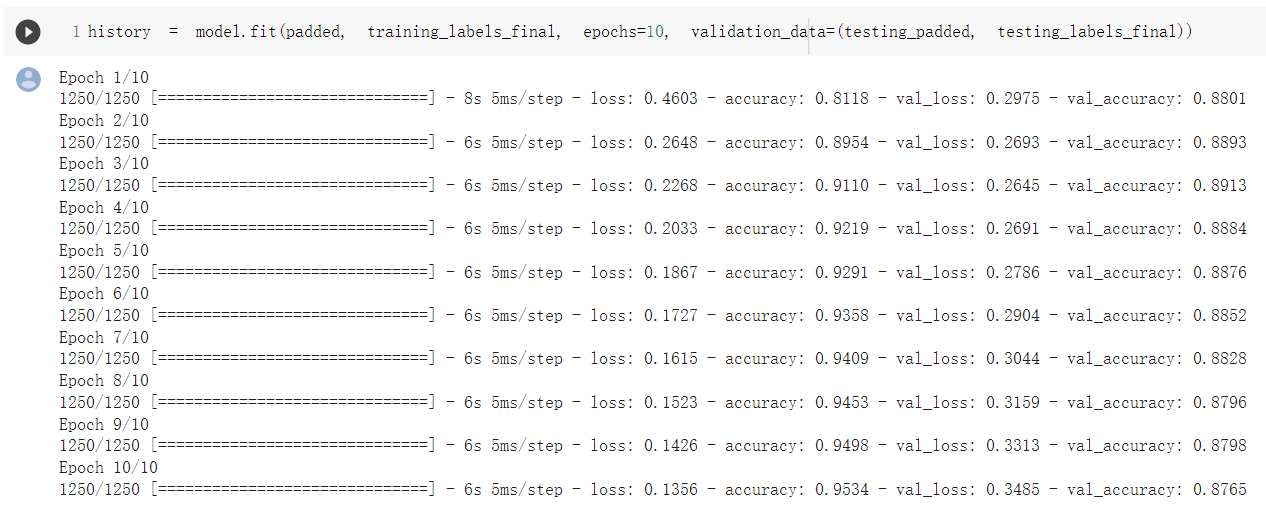


圖2.6 訓練模型/ metrics 評估準確率方法圖

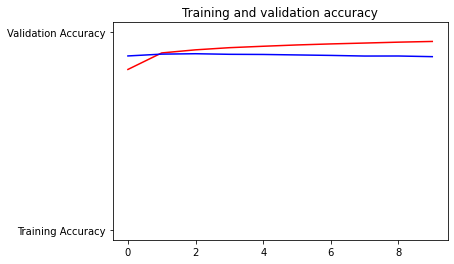


圖2.7 Epoch \_ accuracy 曲線圖

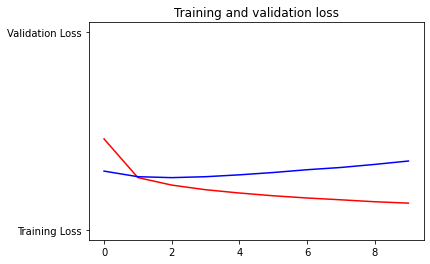


圖2.8 Epoch \_Loss曲線圖

1. 模型二分析結果

如圖2.9 ~ 2.10所示，此模型有0.8825的準確度。

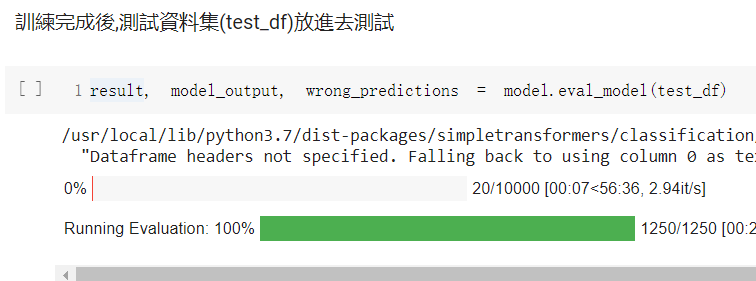


圖2.9 訓練完成後，測試資料集(test\_df)放進去測試

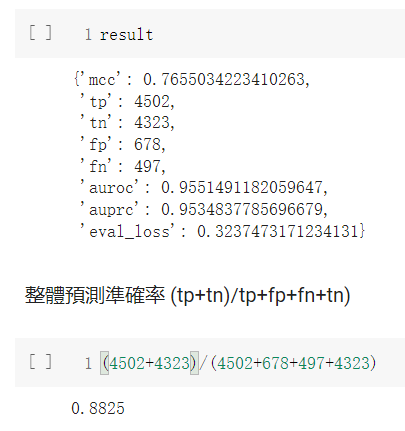


圖2.10 預測成果圖

1. 討論

|  |  |  |
| --- | --- | --- |
|  | 模型一 (一般作法) | 模型二 (Simple Transformers) |
| 測試資料準確度 | 0.8765 | 0.8825 |
| 執行Epoch次數 | 10 | 1 |

由本研究得到的數據可知兩個模型準確率差不多。但比較兩個模型，可得知其差異如下：

1. 一般做法需要經過資料字串處理進行分詞等重要作業
2. Simple Transformers無須進行資料分詞
3. NLP最重要的作業流程之一，就是分詞，尤其對於華人而言，中文分詞又是大工程。但是由4.1、4.2節預測結果得知，透過一段程式，模型二得到的成果與模型一差異不大。而且Simple Transformers不用分詞，節省不少程式撰寫時間，達到簡化的效果。
4. Simple Transformers近來是一個很強力、方便的架構，最早提出是在 Google的[《Attention Is All You Need》](https://arxiv.org/abs/1706.03762)這篇論文中。對於此項技術可以再觀察了解。

# 相關實務應用

本研究可以做為學術交流用途，或者提供相關產業參考。電影產業相關公司(電影製片及發行、線上串流媒體隨選視訊平台)可以透過自然語言處理技術，提供客戶更良好的視聽體驗。電影產業公司，可以在電影上映後，蒐集網路評論。利用模型進行情緒分析，藉此了解受眾對該電影看法的走向。這可以幫助電影產業相關公司進行決策。以線上串流媒體隨選視訊平台為例，若該電影得到的評論傾向正面，可以保留或開始提供該影片；若該電影得到的評論傾向負面，可以不要提供該影片到平台上，若該影片已經在平台上，則可以縮短影片提供觀看的期間，並於到期時下架該影片。電影公司除了參考內部專業人士的意見，搭配此自然語言處理技術輔助，可以選擇更符合客戶期望的電影提供。藉此滿足多數客戶的需求，產生商業價值。舉例來說：好的行銷策略搭配更優質的服務，可以增加已有客戶的忠誠度，並成為吸引潛在客戶的誘因。

# 實作程式連結(GitHub)

**程式連結如下，有兩個版本。****version\_1是採用一般作法，version\_2是採用Simple Transformers的作法：**

* <https://github.com/Jenstudy/movie_project/blob/master/version_1/JJ_1.ipynb>
* <https://github.com/Jenstudy/movie_project/blob/master/version_2/JJ_2.ipynb>
* 注：由於version\_1.ipynb (一般作法/使用模型一) 的程式過大，可能在連結(<https://github.com/Jenstudy/movie_project/blob/master/version_1/JJ_1.ipynb>)

無法檢視。請改用以下連結：<https://nbviewer.org/github/Jenstudy/movie_project/blob/master/version_1/JJ_1.ipynb>