國立虎尾科技大學資訊工程系

專題製作報告

利用文字探勘與分類器技術

建置社群推文檢索工具

參與成員：張國榮　呂嘉哲

簡孝齊　高承泰

指導教授簽名：_____

中華民國　　年　　月　　日

目錄

圖目錄

國立虎尾科技大學資訊工程系專題報告

題目：利用文字探勘與分類器技術

建置社群推文檢索工具

指導老師：黃建宏

專題參與人員：張國榮、高承泰、呂嘉哲、簡孝齊

班級：四資工三乙

一、摘要

本專題利用網路爬蟲自動擷取網站上的文章資料作為語料庫 訓練分類器，利用此分類器自動判斷輸入文章是否屬於旅遊文章，如果是的話屬於三類別下的哪一類別後，即可配合語料庫進行 TF-IDF 特徵詞分析、並可看到關鍵字的重要性。

中文關鍵字：

網路爬蟲（web crawler）：

利用程式去自動瀏覽各式網站並抓取網站內容的工具。透過爬蟲程式可以大幅提升收集資料的速度與自由度。

單純貝氏分類器：

貝氏分類器是一個基於機率統計的監督式學習演算法。貝氏分類器需要先知道各個類別所具有的特徵（在本專題即為特徵字），並假設所有的特徵之間具有條件獨立的情況。因此可以利用條件機率相乘的方法，計算該篇文章屬於某個類別下的機率。根據貝氏定理，其公式為：

$$p(C|F1, \cdots, Fn)$$

其中 C 為類別，$(F1, \cdots, Fn)$為該類別特徵字之集合。而由於單純貝氏分類器假設各個特徵具有條件獨立的情況，故上述公式又可以寫為：

$$p(C)\prod_{i=1}^{n} p(Fi|C)$$

通過計算文章於各個類別下的機率並取出最大值，即可推估輸入文章可能屬於的類別。

基因演算法（英語：genetic algorithm (GA)):

是計算數學中用於解決最佳化的搜尋演算法，是進化演算法的一種。進化演算法最初是借鑑了進化生物學中的一些現象而發展起來的，這些現象包括遺傳、突變、自然選擇以及雜交等等。基因演算法通常實現方式為一種電腦模擬。對於一個最佳化問題，一定數量的候選解（稱為個體）可抽象表示為染色體，使種群向更好的解進化。傳統上，解用二進位表示（即 0 和 1 的串），但也可以用其他表示方法。進化從完全隨機個體的種群開始，之後一代一代發生。在每一代中評價整個種群的適應度，從當前種群中隨機地選擇多個個體（基於它們的適應度），通過自然選擇和突變產生新的生命種群，該種群在演算法的下一次疊代中成為當前種群。

英文關鍵字：

bert:

起源於預訓練的上下文表示學習，包括半監督序列學習(Semi-supervised SequenceLearning)，生成預訓練 (Generative Pre-Training)，ELMo 和 ULMFit。與之前的模型不同，BERT 是一種深度雙向的、無監督的語言表示，且僅使用純文字語料庫進行預訓練的模型。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「水分」的 word2vec 詞向量在「植物需要吸收水分」和「財務報表裡有水分」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。

BertTokenizer:

分詞器，做一些基礎的大小寫、unicode 轉換、標點符號分割、小寫轉換、中文字元分割、去除重音符號等操作，最後返回的是關於詞的陣列。

TF-IDF（Term Frequency－Inverse Document Frequency）：

是一種用於自然語言處理的一種統計方法。用以評估一個字對於一個語料庫中的其中一份文章的重要度。TF（Term Frequency）代表字詞的重要性隨著它在檔案中出現的次數增加，其公式為：

$$\mathbf{tf}_{i,j}= \frac{n_{i,j}}{\sum_k n_{k,j}}$$

表示單詞 在語料庫中的一篇文章 的出現次數，而 則是在該篇文章中，所有出現單詞的次數總和。IDF（Inverse Document Frequency）則是讓詞隨著它在語料庫中出現的頻率成反比下降。其公式為：

$$idf_i \mathrm{idf_i}= \log\frac{|D|}{|\{j:t_i \in d_j\}|}$$

其中$|D|$表示整個語料庫中所含的文章數量，， $|\{j:t_i \in d_j\}|$則表示語料庫中包含單詞的文章個數。故計算 TF×IDF 可找出在文章之於整個語料庫中較為重要的幾個關鍵字，而在本專題我們亦使用 TFIDF 來作為貝氏分類器的分類特徵。但考慮到在不同的類別下同樣的單詞可能會具有不同的 TFIDF 權重，故我們將 TFIDF 的 IDF 值對應到的整個語料庫的文章數量，改為對應到該類別下所含的文章數量，以更利於找出文章中的特徵字。

二、專題緣由與目的

專題緣由：

由於現在 3C 產品的普及，人人皆有一機，整天待在家裡不出門了解台灣的美實在太可惜了，台灣的景點與美食都還沒吃遍玩透，怎麼可以說沒地方玩了呢。而現今網路資料量龐大，若是不把對台灣付出貢獻的旅遊文章加以統整，也太對不起台灣這個寶島了。為了讓台灣使用者更方便的瀏覽旅遊相關資料，我們希望透過抓取網路上的文章加以統整，經過學習過後的分類器，整理出旅遊各個標籤分類，進而從網路上提高獲取旅遊資料的效率。

目的:

1.自動文章分類：

使用者自訂或爬到的文章可透過旅遊文章自動分類，快速標籤該篇文章屬於台灣娛樂、台灣美食、台灣文化的哪一類別。確定該篇文章的所屬類別後，便可利於後續分析。

2.音訊檔分類：

使用者可透過上傳音訊檔(.wav)來分析出文字，並藉此判斷該影音是否屬於旅遊，如果是又屬於哪一類別，並且可以得到該音訊檔的的文本，可以更快速替該影音檔如:MP4、MPEG、API……等等的影音檔上字幕，以達到節省人工費時的上字幕過程。

3.特徵字萃取：

進行特徵字的萃取可讓使用者明白該篇文章或語料庫具有哪些特別重要的關鍵字。除可做為訓練之特徵外，結合網路爬蟲程式可以讓使用者明白當前新聞的趨勢與熱門關鍵字。

4.基因演算法分詞:

透過一連串的基因演算法的交配與突變過程，去產生斷句，並且可以不需要預先設定只有那些詞才會被切出來，只需要有足夠多的文章使其產生斷句的權值，讓基因演算法去計算其適應值便可以產生斷句。

專題製作過程：

本專題的預處理流程與應用功能分別如圖 2-1、圖 2-2 所示：

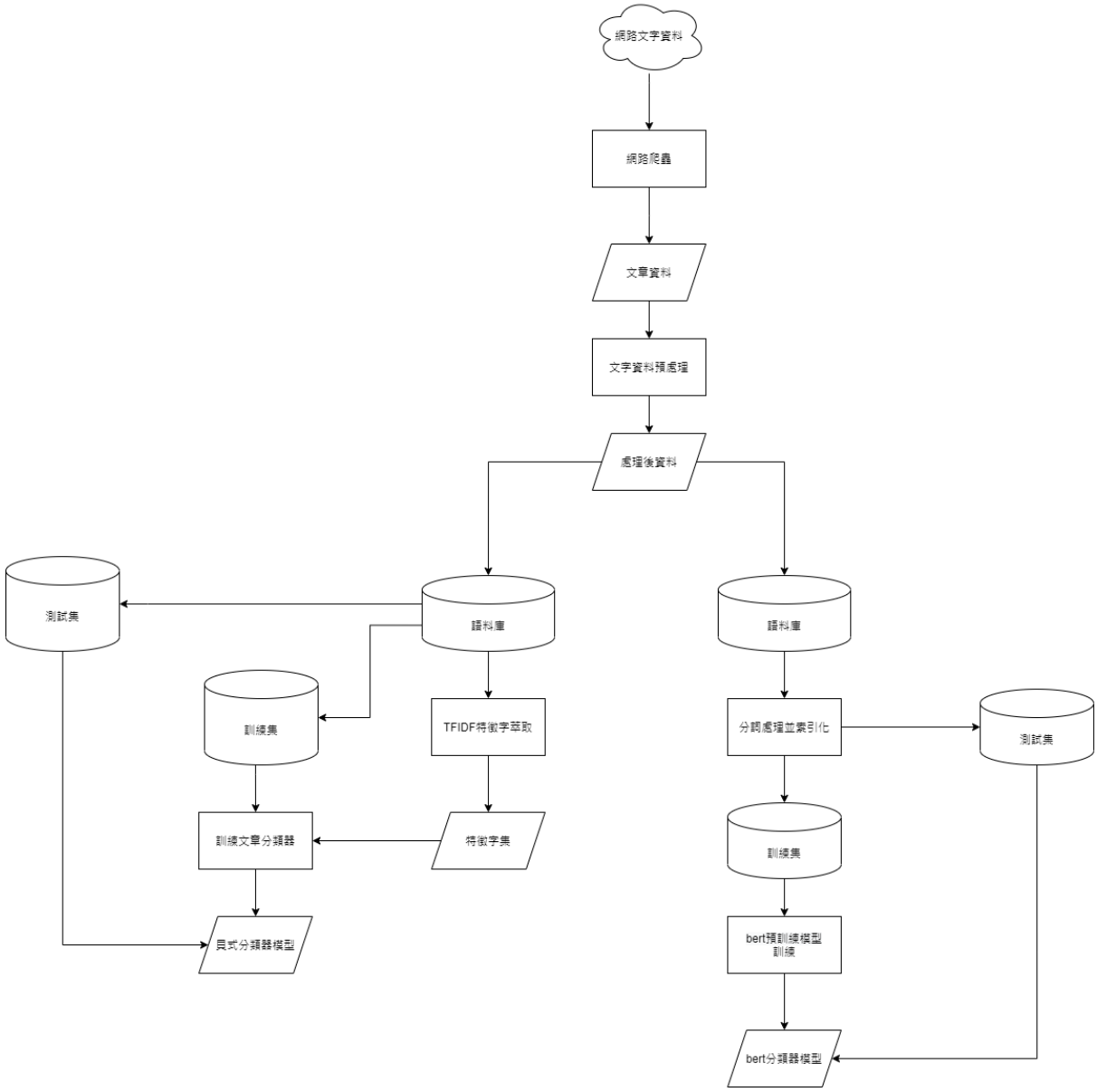

圖 2- 1 模型預處理步驟

其中我們主要需要處理的環節包含：

- 網路爬蟲程式
- 語音辨識
- 文字資料預處理

- 貝氏分類器的特徵字萃取
- 基因演算法
- Bert 分詞處理
- GUI 設計

網路爬蟲程式

本專題取自 ETtoday 網站（https://travel.ettoday.net/）提供所提供的中文旅遊資料集。ETtoday 是台灣重要的新聞網站之一，其特點在於該公司從未印製過紙本報導而全由數位媒體刊載新聞，因此具有相當豐富且利於我們使用的新聞文章可作為我們的資料集。

我們使用爬蟲程式擷取的檔案格式為 beautifulsoup，將其解析後，爬蟲程式連續對每篇文章的網址進行存取。

某些網站設有反爬蟲機制阻礙連接，因此設置虛擬代理人欺騙以便連續爬取，爬蟲程式將爬取的文章以陣列方式儲存，陣列一儲存新聞標題、陣列二儲存新聞文章內容，之後以類別為區隔，將每個類別的新聞轉換為資料結構並輸出為 xlsx 格式。

此外，我們有定時爬取我們需要的文章，將他存放在同一個目的地，讓我們的系統可以持續精進。

語音辨識

讓使用者上傳一個影音檔(.wav)，系統將辨識該影音檔並顯示出該影音檔的說話內容，且可以看它是否屬於旅遊，如果是的話又屬於旅遊三大類別中的哪一個類別。

文字資料預處理

將爬蟲程式取得的資料依照類別歸納成一個個的 xlsx 檔後，貝氏分類器的部分就必須將文字資料進行去除不需要的文字的動作。在此階段使用 jieba 中的中文字分割成一個個的 Token，並用正規表示式去除標點符號，英文以及數字，僅留下中文 Token 作為訓練材料。

基因演算法的部分則是在去除不必要的文字後，標點符號，數字，英文等等的，而在遇到這些情況式進行分段，並且將每一篇文章的每一個分段進行兩字組、

三字組、和四字組的切割並記錄下出現幾次以及長度以做為他們的權值，來讓基因演算法計算適應值。

貝氏分類器的特徵字萃取

在專題製作時使用 TF-IDF 作為特徵字的依據，可是由於資料量達不到一定的數量，所以訓練後得到的成功率只有 50 出頭，後面去直接進行人為的特徵字新增才往上提到了 60%。

基因演算法設計

在製作基因演算法時，由於沒有模組可以直接使用，所以是參考網路上找到的流程圖來進行設計，當時經過了多次的調整參數，並且經過多次使用確定好這樣不會是一個窮舉法的設計，每次的結果都是局部最佳化產生的結果，並且有慢慢因為突變的過程來突破局部最佳化，才確定基因演算法在於中文切字的應用確實是可行的。

Bert 分詞處理

讀取檔案轉換成 csv 檔並用 transformers 中的 BertTokenizer 對讀取的文章進行分詞，並轉化成 2 種序列號。

token_tensor:代表該文字的索引值

segments_tensor:代表文章的界線，用 0 和 1 做區別，此專題只對一篇文章進行分類，故都是 0。

將兩種序列號放進模型以進行預測。

GUI 設計

為方便檢視專題成果，本專題透過圖形視窗介面呈現，GUI 以簡約舒適的風格設計為主軸，讓按鍵一目瞭然，且設有防呆機制大幅減少因使用者不當使用行為導致程序終止的機率。此外，本專題分析時會花費一些時間導致圖形視窗介面凍結，所以會有進度條表示程式正常運作中。

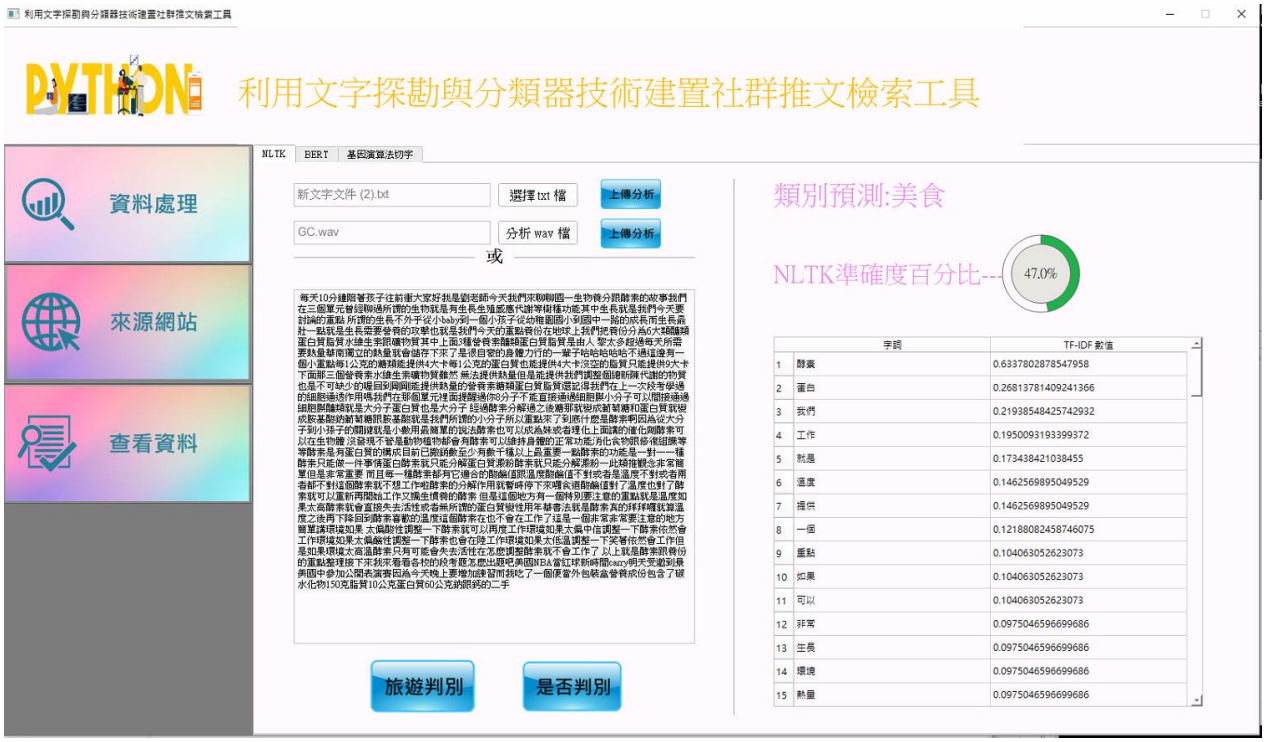

圖 2- 2.2GUI NLTK 分類介面呈現

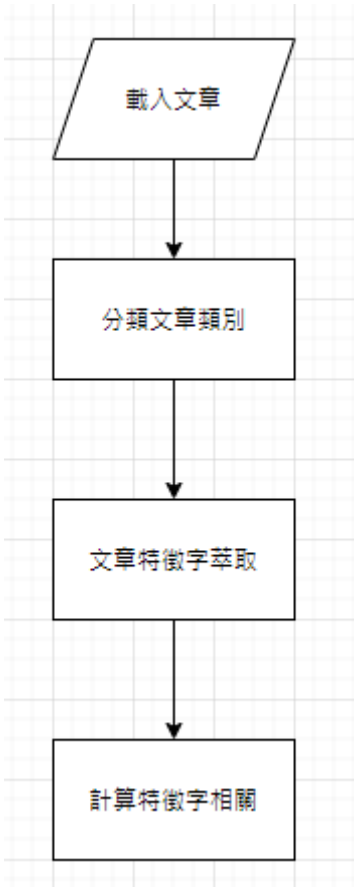

圖 2- 3 GUI NLTK 分類流程圖

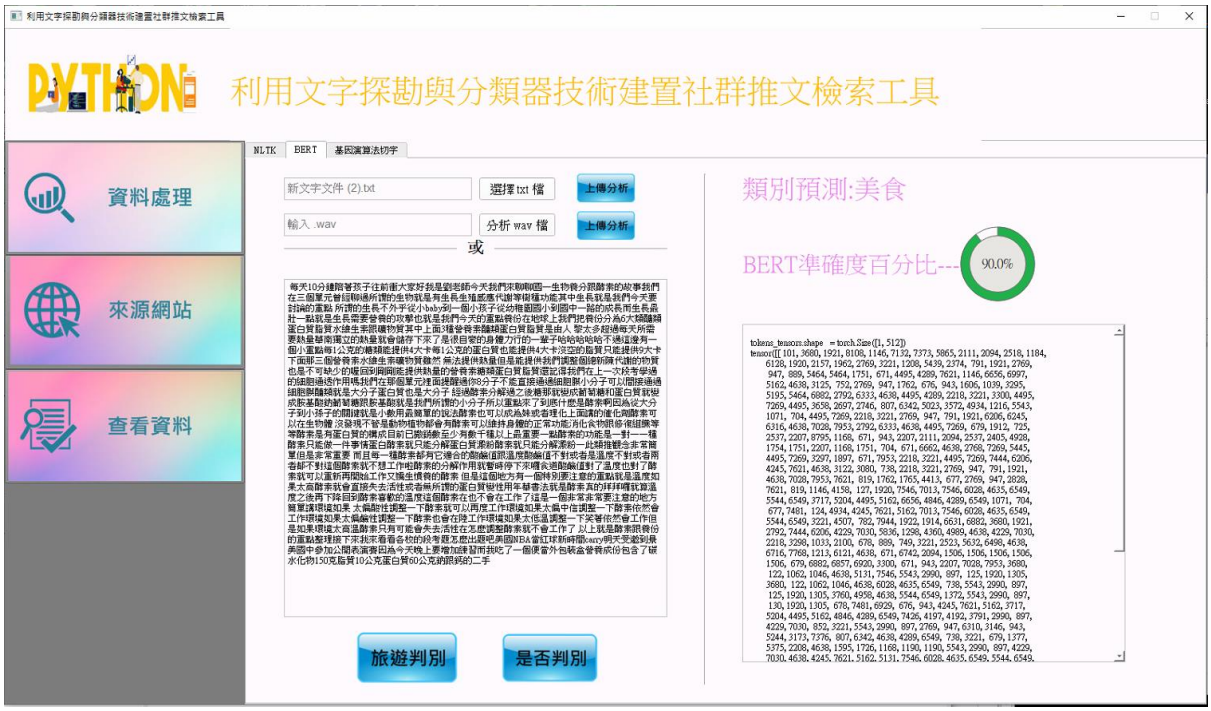

圖 2- 4 bert 分類介面呈現

圖 2- 5 bert 分類流程圖

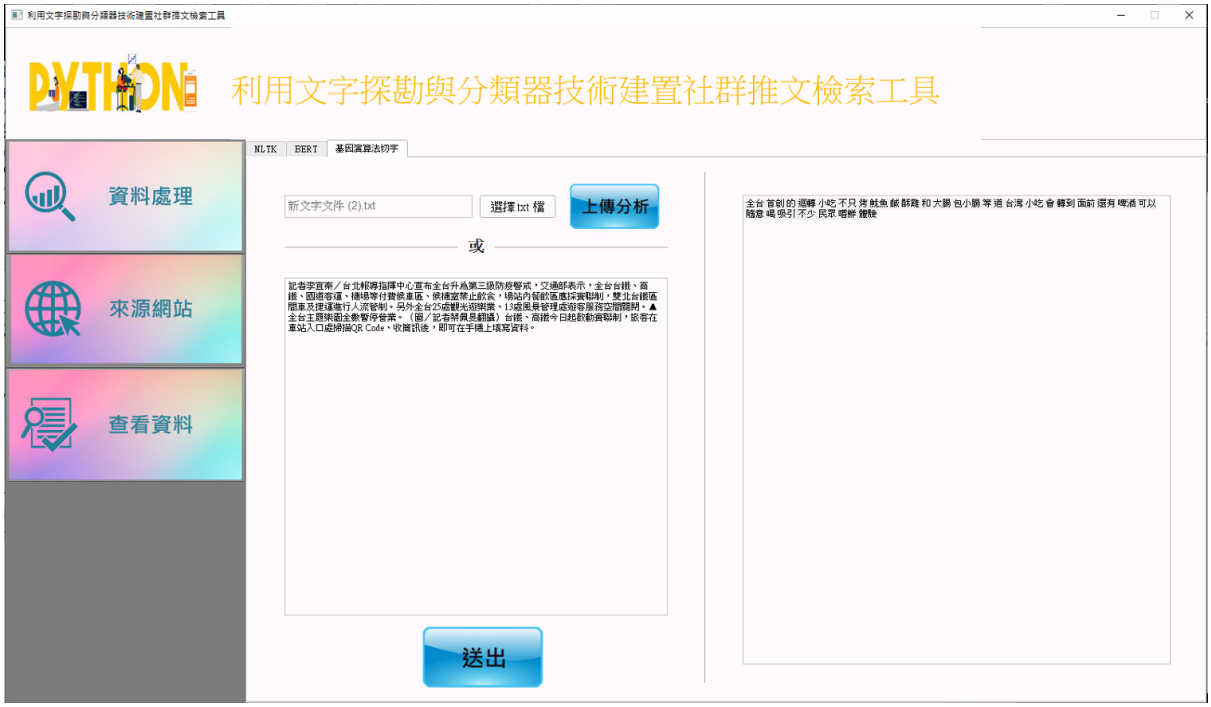

圖 2- 6 基因演算法分詞呈現

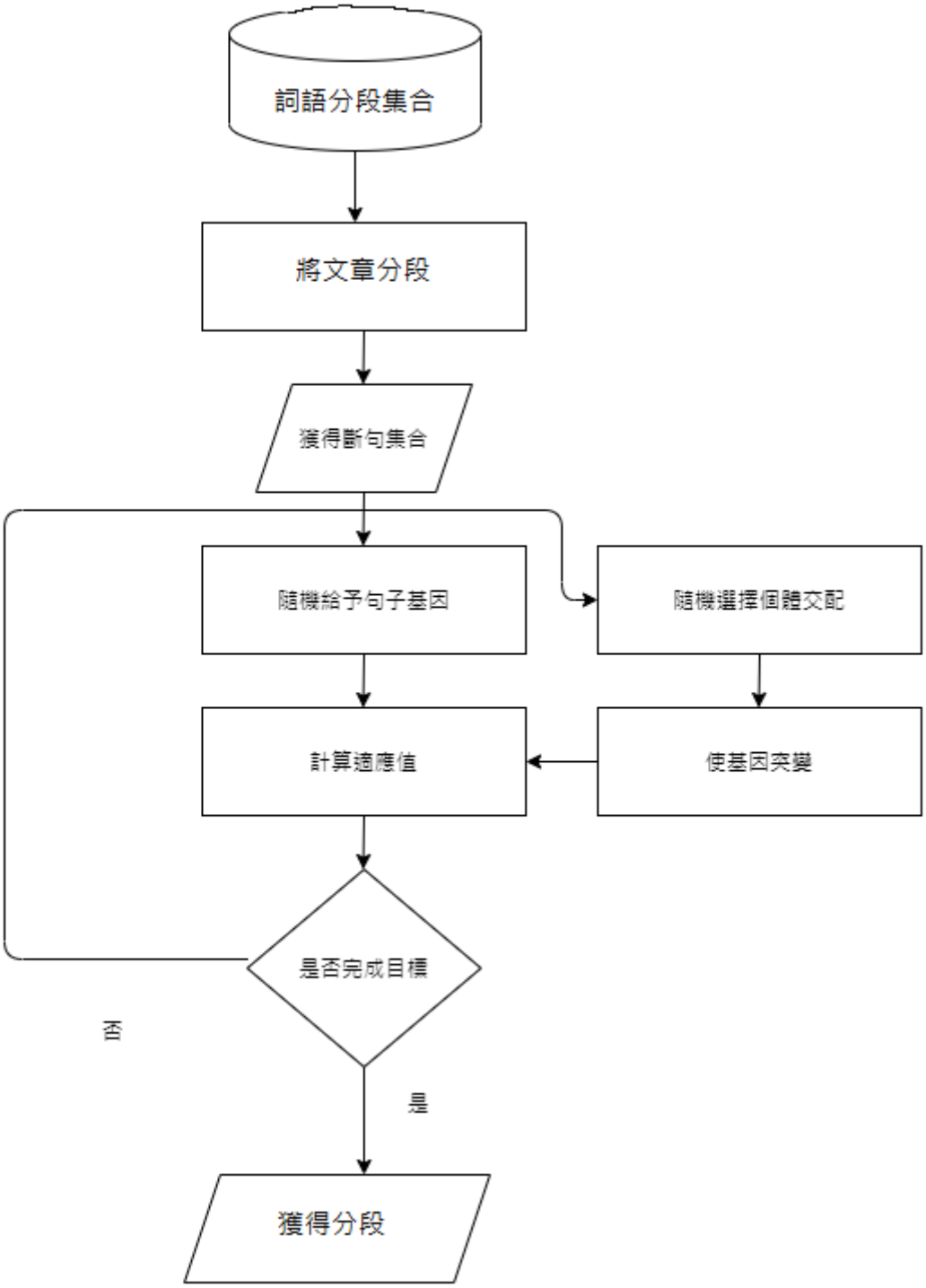

圖 2- 7 基因演算法流程圖

三、結論

本專題我們用三個類別來進行分類，但是本專題的核心程式 都採用了物件導向模組化的方法進行撰寫，具有很高的擴充性與可攜性， 也因此可以輕鬆地應用到其他類型的分類分析任務上。

bert 與 NLTK 的準確率比較:

三類(美食、文化、娛樂)分類:92% vs 57%　　單一類(美食):90% vs 47%

單一類(文化):93% vs 68%　　單一類(娛樂):92% vs 53%

是否分類:96% vs 70%　　單一類(是):94% vs 98%　單一類(否):99% vs 42%

從上面的數據可以得知 bert 的強大，幾乎每個判斷準確率都達到 90%以上，跟 NLTK 一相比，除了在單一類(是)的分類比較優秀，其他都被 bert 比下去了，甚至有些連一半的準確率都不到。

經過本專題後我們除了學習到基礎的機器學習、自然語言處理、bert 的簡單應用與統計方法的應用外，也明白了如何處理在實作上述的操作時所遇到的諸如程式錯誤、特徵篩選、過度訓練，資料存儲失敗等等的問題，雖然不能做到最好，但也是一個十分寶貴的經驗。

四、團隊分工

專題統籌與進度規劃：張國榮
bert 分類器的建置：高承泰
貝氏分類器的建置：張國榮
基因演算法分詞的建置：張國榮
網路爬蟲程式：簡孝齊
GUI 介面撰寫與設計：簡孝齊
專題文件撰寫：呂嘉哲 簡孝齊 高承泰
專題海報設計：簡孝齊 呂嘉哲 高承泰
資料庫存取：呂嘉哲
影音處理系統：呂嘉哲

五、參考文獻

[1]Van de Cruys, Tim.(2011)“Two multivariate generalizations of pointwise mutual information.” In Proceedings of the Workshop on Distributional Semantics and Compositionality, pp. 16-20. Association for Computational Linguistics.

[2] Salton G. and McGill M.J. .(1983). "Introduction to modern information retrieval", McGraw-HIII Book company.

[3] George H. John and Pat Langley (1995). "Estimating Continuous Distributions in Bayesian Classifiers." Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. pp. 338-345. Morgan Kaufmann, San Mateo.

[4] Hand, D. J.; Yu, K.(2001)"Idiot's Bayes — not so stupid after all? ".International Statistical Review.2001, 69 (3): 385–399.

[5] Russell,S and Norvig, P.(1995) “Artificial Intelligence: A Modern Approach 2nd. Prentice Hall.” 2003

[6]黃仁鵬、張貞瑩 .（2014）. 運用詞彙權重技術於自動文件摘要之研究。中 華民國資訊管理學報，第二十一卷，第四期，頁 391-416

[7]Python 使用 MySQL 資料庫的教學與安裝-
https://www.maxlist.xyz/2018/09/23/python_mysql/

[8]進擊的 BERT:NLP 巨人之力與遷移學習
https://leemeng.tw/attack_on_bert_transfer_learning_in_nlp.html

[9]BeautifulSoup 開發網頁爬蟲
https://www.learncodewithmike.com/2020/02/python-beautifulsoup-web-scraper.html

[10]AJAX 動態載入網頁的爬取秘訣
https://www.learncodewithmike.com/2020/10/scraping-ajax-websites-using-python.html