應用 kNN 演算法之文件分類平台實作

黃冠中
國立澎湖科技大學資訊工程學系

論文摘要

隨著科技日新月異，全球的資料量也逐年攀高，面對如此龐大的資料量，我們需要一個有效的管理方法，因此有人就提出了自動化文件分類的技術，本論文所要研究的並不是提出新的分類方法，而是整合過去所提出的自動化分類研究文獻，建立出一個較完整的自動化文件分類實驗平台，讓使用者能夠輕鬆在實驗平台上，設置各種實驗參數，並且輸出完整的實驗結果，從公平的角度下探討自動化文件分類的成效性。

關鍵詞：自動化文件分類(Automatic Rocchio Document Categorization)、KNN

一、緒論

1.1 研究背景

隨著近幾年來，科技網路不斷進步的關係，世界各地的新興文件以及資訊數量不斷的成長，不論是在網際網路(Internet)、數位圖書館(Digital Libary)、或是新聞(News)等方面都有驚人的數字。根據統計，從 1971 年開始，平均每 2.3 年，線上資料庫的數量就倍增，而這些線上資料庫內的資料，則以更快的速度急遽增加中[9]。很明顯的，面對這樣龐大的資料量，我們需要有效的辦法來管理，否則我們很難搜尋到需要的資料。

下圖是美國柏克萊大學所作出的全球資訊量的成長圖，由此圖我們可看出在 2003 年的時候全球的資訊量就已經高達(570 億 gigabytes)，到了今年我們更是難以想像資訊成長到了什麼地步，若再不能有效管理這些資訊的話，未來將會變「知識的垃圾掩埋場」(Knowledge Asset Landfills)，資訊的管理已經成為了當下刻不容緩的議題了。

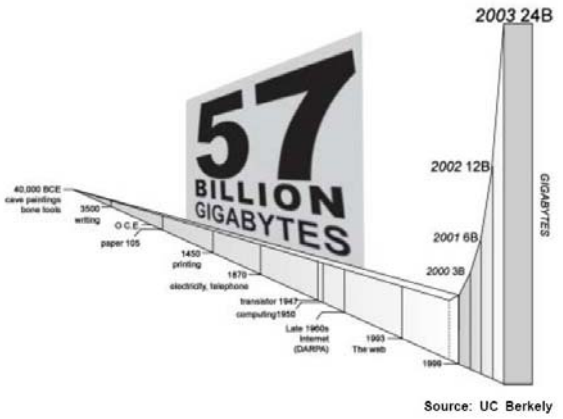

圖 1：資訊量的成長 [10]

在早期文件分類技術最普遍的方法就是知識工程，知識工程主要是以人工的方式進行分類，首先依據領域專家的知識定義每個類別的分類規則，當知識工程師和領域專家合力建立好分類規則之後，需要分類的文件即可依照這些分類規則進行分類，然而知識工程最大的缺點就是，需耗費大量的人力及物力資源在訂定分類規則和分類上，面對現今如此龐大的資料量，知識工程顯然無法應付『即時性』的需求，且人工的判斷法也可能不夠客觀，往往因為人們認知的不同而產生不一致性的現象。

1.2 研究動機

隨著資料量不斷的增加，早期的分類方法已經不敷使用了，因此在 90 年代興起了，機器學習(Machine Learning)模式，在機器學習模式中，會先由領域專家將一些文件是先分類好，接著機器學習模式會自動的從這些文件中找出每個類別的特徵，再利用這些特徵來做分類處理。本次研究就是希望能夠運用機器學習模式，建構出一個文件自動分類的平台，藉以解決人工分類法耗費人力物力資源的問題，且自動分類可以指定正確的類別給予這些文件，減少分類不一致性的問題。

雖然自動化文件分類，是近代學術研究中一個熱門的話題，但是一直都沒有一個較完整的實驗平台，能夠來測試自動化文件分類器的成效性，本論文所要研究的並不是提出新的分類方法，而是將過去所提出的自動化分類研究文獻收集整理，希望能夠把先前研究的相關文獻加以整合，建立出一個較完整的自動化文件分類實驗平台，和實驗參數設置介面，精靈式的參數設置介面，讓使用者能夠在實驗平台上，很輕鬆的設置各種實驗參數，並且輸出完整的實驗結果，讓使用者能夠知道在何種情況下，哪一種參數組合最好，期望本論文能夠在公平的角度下探討自動文件分類的成效性。在近代自動化文件分類的熱潮中，盡點綿薄之力。

二、文獻探討

在傳統的文件分類中，人們須先了解文章的大綱才能將其分類，這是相當高階的知識處理，然而在機器學習模式中，由於機器本身並不能了解文章的大綱，所以必須有一套方法，讓機器能夠自動的學習分類規則，[18]中指出機器學習模式中所要建立的並不是分類的規則，而是分類規則的自動建構者，目前最常用的方法為向量空間模式(Vector Space Model)。

向量空間模式的主要概念是由 G.Salton(1983)提出，簡單的來說就是先將待分類文件擷取出能夠代表該文件的特徵，然後將特徵轉換成為向量，再將其特徵向量與現存的類別特徵向量(Feature Vector)做相似度(Similarity)的比較，若相似度高於門檻值(Threshold)便將，待分類文件歸於該類別，其處理方法可分為四步驟討論，分別為前處理(Preprocessing)、特徵選取(Feature Selection)、相似度(Similarity)、分類方法。

2.1 前處理(Preprocessing)

為了使自動文件分類更為有效率，分類的結果更正確，我們通常會在特徵擷取前做一些處理，以去除雜訊(Noise)，使擷取出來的特徵更能代表該文件，稱之為前處理(Preprocessing)，以下是幾個常見的前處理方法。

刪除停用字(Stop-Word List)

在文件中有很多主詞、冠詞、介詞，還有其他常出現的慣用字，像是這一類的字它們本身並不具有任何的意義，但出現在一般文件的機率和頻率卻很高，如果我們能夠先將這一類無意義的字拿掉的話，便能大大的減少向量空間模式的維度，並且有效地增進機器學習的效率和準確性，而這些文字的集合我們稱為 Stop-Word List。

還原字根(Stemming)

在英文文件中有許多的名詞、動詞會以不同的型態出現像是單、複數型態，或者現在式、過去式…等，但實際上它們的字義都相同，如果不加以處理的話，在特徵擷取的過程中他們便會被視為不同的字，向量空間模型的維度便會大幅度的增加，對於這種情況必須制定一套規則來還原字根，本研究中採用 1980 年由 Martin Porter 提出 Porter Stemming algorithm[15]。

專有名詞（Proper noun）

針對特殊的文件，我們可以加重一些專有名詞的權限，藉以提高自動分類的準確性，以英文新聞文件為例，若出現人名、地名…等，字首為大寫的專有名詞(斷落第一個字除外)，像是出現「陳水扁」、「布希」我們幾乎可以認定此一文章為政治方面的報導，但是專有名詞通常出現的次數比較少，所以加重這些專有名詞的權重(Weight)，便有利於提高自動分類的準確性，本論文將研究此一方法的可行性。

2.2 特徵選取(Feature Selection)

特徵選取為整個自動化文件分類過程中最重要的一個步驟了，必須先由文件本身自動擷取出足以代表該文件的特徵，特特徵選取可以從文件表面的詞語(字、詞或片語)資訊獲得，通常有以下幾種：

字詞出現頻率 TF(Term Frequency,TF)

字詞出現頻率，指的是某一關鍵詞在某類文件中出現的次數，可用以下公式來表示：

$$W(d, t) = TF(d, t) \quad (1)$$

W(d,t)：文件 d 中出現關鍵詞 t 的權重。

TF 可以得到很高的召回率(Recall)，但並不精密。主要是因為如果關鍵詞，經常出現在各類的文件類別的話，那麼這些關鍵詞作為某一文件的特徵就會不明顯，因此最好能將出現在各文件類別中頻率較高的關鍵詞，從關鍵詞集(Trem Collection)之中移除，以提高召回率(Recall)[8]。

逆文件頻率(Inverse Document Frequency,IDF)

單一關鍵詞出現在不同文件中，出現此關鍵詞的文件數量稱之為文件頻率(Document Frequency,DF)，而逆文件頻率所要表達的是，若單一關鍵詞太普遍的出現在各個文件中的話，它所能突顯的意義就會相對的降低，若能夠集中同一類別的話，更能突顯其特徵，公式如下所示：

$$IDF(t) = \frac{N}{df(t)} \quad (2)$$

N 代表文件種數。

df(t)代表含有關鍵詞 t 的總數。

如果使用 IDF 來表達關鍵詞的特徵(Term Specificity)的話，則將可以提高召回率(Recall)。

同時強調字詞出現頻率及普遍性 TF × IDF

此一方法為自動化文件分類研究中最常被使用的，根據 G.Salton 等人的說法，單單只考慮 TF 或者 IDF 是不夠的，若使用 TF×IDF 則可以得到更好的特徵性。相對的由於必須同時考慮到 TF 和 IDF，若面對較龐大的資料量時，在資料計算方面，就沒有上述兩種方法來的來的快速，公式如下所示：

$$W(d, t) = TF(d, t) \cdot IDF(d, t) \quad (3)$$

W(d,t)：關鍵詞 t 在文件 d 中的權重。

TF(d,t)：文件 d 中出現關鍵詞 t 出現的頻率。

IDF(t)：逆文件頻率。

加權式逆文件頻率(Weighted Inverse Document Frequency,WIDF)

若依據 IDF 的定義，只論關鍵字出現的文件總數，不論文件出現該關鍵字次數，將會出現特徵分佈不合理問題。因為不管關鍵字出現在文件中幾次，結果所得到到 IDF 值都是 1，顯然並不合理。應該將其在該類文件中的出現頻率表現出來。定義如下:

$$WIDF(d, t) = \frac{TF(d, t)}{\sum_{i \in D} TF(i, t)} \quad (4)$$

TF(d,t)：代表關鍵字在 d 文件類別中出現的頻率。

$\sum_{i \in D} TF(i, t)$ ：i 代表 D 的文件集合的範圍內的各類文件。

由 M. Iwayama 的實驗中，可以得知，其召回率確實比 TF×IDF 提高了 4.4% [19]。

採用取平方的 IDF

由上述的內容得之，使用 $\log \frac{N}{df(t)}$ 會降低詞的特殊性，所以分類效果不甚理想，為了拉大特殊詞與常用詞之間的差距，可以採用 $\left[\log \frac{N}{df(t)}\right]^2$ 作為 IDF [11]。則權重定義可調整成：

$$W(d,t)=TF(d,t)\cdot\left[\log\frac{N}{df(t)}\right]^2 \qquad (6)$$

TF(d,t)：關鍵詞出現率。

df(t)：代表含有關鍵詞 t 的文件總數。

N：文件類別總數。

2.3 相似度(Similarity)

在特徵擷取完成之後，在過來就是要做各個文件向量的相似度的比較了，以下幾個是較常見的相似度的比較方法。

餘弦係數(cosine coefficient)

在向量空間中，最常被設計來計算相似度的就是餘弦係數(cosine coefficient)了，公式如下所示：

$$\text{cosine}(x,y)=\frac{\sum_{i=1}^{t}x_i y_i}{\sqrt{\sum_{i=1}^{t}x_i^2}\sqrt{\sum_{i=1}^{t}y_i^2}} \qquad (7)$$

x、y 為兩文件向量，x(x1, x2, x3,… xt)、y(y1,y2,y3,…yt)。

t 為 x 與 y 的維度。

由餘弦定理我們可以知道，若兩文件的維度比例皆相同，即兩互相向量平行，則其夾角為 0，兩向量的餘弦係數為 1，代表著這兩文件有極高的相似度，反之，當兩文件的維度比例不盡相同時，餘弦係數將降低，代表著兩文件並不相似。

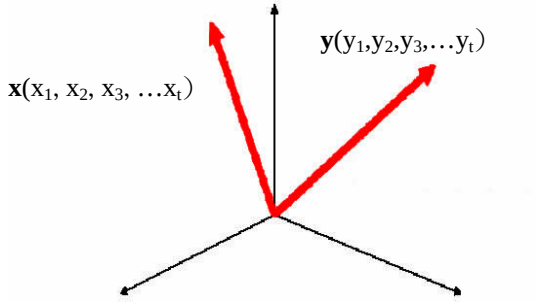

圖 2： 兩文件在空間向量中的表示

Jaccard 係數

在過去的研究中，Jaccard 係數是在衡量交易資料集（Transaction Data Set）時最為廣泛使用的相似度量測標準（Guha, Rastogi & Shim, 1998）。Jaccard 係數亦稱為 IOU 指標（Intersection Over Union measure），假設兩物件 a ,b 所屬交易之集合分別為 X 與 Y，則 Jaccard 係數分子為 X 與 Y 交集之大小，分子為 X 與 Y 聯集之大小。

$$\text{Jaccard}(x,y)=\frac{\sum_{i=1}^{t}x_i y_i}{\sum_{i=1}^{t}x_i^2+\sum_{i=1}^{t}y_i^2-\sum_{i=1}^{t}x_i y_i} \qquad (8)$$

x、y 為兩文件向量，x(x1,x2,x3,…xt)、y(y1,y2,y3,…yt)。

t 則為 x 與 y 的維度。

Dice 係數

$$dice(x,y)=\frac{2\sum_{i=1}^{t}x_i y_i}{\sum_{i=1}^{t}x_i^2+\sum_{i=1}^{t}y_i^2} \qquad (9)$$

x、y 為兩文件向量，x(x1,x2,x3,…xt)、y(y1,y2,y3,…yt)。

t 則為 x 與 y 的維度

2.4 分類方法

k-最鄰近法(k Nearest Neighbor,kNN)

k 最鄰近法在過去的研究文獻中，算是最常出現的一種分類法。主要是以向量空間的方式表示各篇文件的特性。當新進資料須判別其類別時，根據 k 個最接近新進資料的訓練資料的特性來預測出新進資料的類別。一般來說，k 最鄰近法的分類效果會較其他的分類器來的好，主要的原因在於它大量的訓練文件所產生的向量空間，但所花費的時間與空間卻過於龐大。由於 k 最鄰近法的計算方式是，利用待測文件與所有訓練文件作相

似度的計算，之後才能判斷出與其最接近的 k 篇訓練文件為何。因此每預測一篇待測文件的類別時，就必須做一次龐大的計算。Lam 與 Ho 觀察到 kNN 還是會受到雜訊文件（即分錯類別的訓練文件）的干擾，因此提出一個改良 kNN 的方法，實驗結果顯示其分類成效及分類速度都有提升，但訓練時的計算量卻變大[17]。圖 3 中表示當 k 值設定為 4，由圖中可以很清楚的看到，與文件 d1 相近的訓練文件中，最靠近 d1 的有 c1 類別中的 3 篇文章，和 c2 類別中的 1 篇，且已達 k 值門檻。

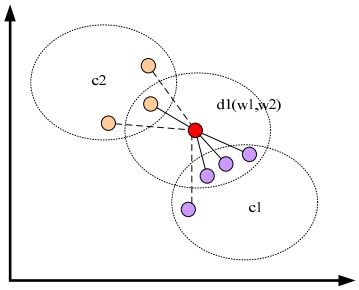

圖 3：kNN 判斷說明

kNN 會根據下列的公式挑選最合適的類別，公式如下：

$$y(x,c_j)=\sum_{d_i\in NN} Sim(x,d_i)y(d_j,c_j)-b_j \quad (10)$$

中心向量法

中心向量法與 kNN 非常的相似，一樣是在向量空間模型下，常出現的分類方法，其與 kNN 演算法最大的不同點在於，kNN 演算法是一種幾乎不需要訓練的分類方法，在有測試文件進來時，才讓測試文件對每一個訓練文件作比較，而中心向量法則是，在訓練階段的時候不斷的訓練分類方法，對每一個都產生出代表此類別的特徵向量，當有測試文件進來時，測試文件只需對每一個類別做比較，而不需要對每一文章都做比較。由下圖可以清楚的看到，當有測試文件 D1 進來時，D1 只對代表 c1 和 c2 類別的特徵向量做相似度的比較，而非針對所有的訓練文章做比較。由於圖中 c1 對 D1 的夾角，比 c2 對 D1 的夾角，來得要小，所以將 D1 推薦至 c1 類別中。

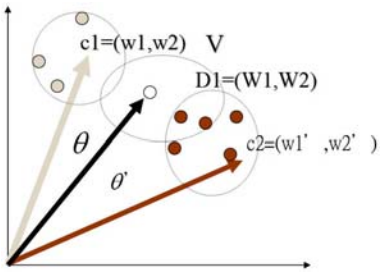

圖 4：中心向量法判別說明，此圖取自高自強,2004

2.5 門檻值(threshold)

Fabrizio Sebastiani (2002)指出，所謂的分類方法即是在歸納處理決定某個文件屬於某個分類的過程，由於半自動型的分類方法，會依據合適度排序合適的類別，但不做最後的決策，因此，必須藉由門檻值(Thresholds)來做分類的判斷，當未分類文件的相似度值高於門檻值時，則推薦文件到這個分類類別中，反之當未分類文件的相似度值低於門檻值時，將不推薦此文件到這個分類類別中。一般來說決定門檻值的方法，可以透過反覆實驗可以找出一個門檻值，使得半自動型分類器能具有較好的分類結果推薦。或者，也可以依據每個類別的特性決定各自的門檻值或只定義一個固定的門檻值，讓所有的類別都使用這個門檻值。

2.6 績效評估

表 1：文件數量分佈表

	分為該類	不分為該類
屬於該類別	a	b
不屬於該類別	c	d

在傳統的分類積效評估方式，必須先對每一個類別，將所有的文件將被劃分於如表 1 所示的四種狀況中，即屬於該類的文件，被系統正確分為該類的有 a 篇、沒被系統分為該類的有 b 篇；而不屬於該類的文件，被系統分為該類的有 c 篇、沒被系統分為該類的有 d 篇。對每個類別都做這樣的統計後，即可計算「正確率」(accuracy)、「精確率」(precision)、「召回率」(recall)，如下：

$$\text{accuracy} = \frac{(a+d)}{(a+b+c+d)} \quad (11)$$

$$\text{precision} = \frac{a}{(a+c)} \quad (12)$$

$$\text{recall} = \frac{a}{(a+b)} \quad (13)$$

其中正確率受 d 值影響很大，當 d 遠大於其他值時，不管有沒有正確分類，其「正確率」都接近 1。由於有這樣的不合理存在，這個評估方式盡可能不要用。而精確率與召回率不能單獨使用，理由是系統很容易做出高精確、低召回，或低精確、高召回的結果。為同時兼顧這兩個數據，經常再定義 F1=2PR/(P+R)，來比較不同系統的成效。如果同時有好幾個類別要一起考量，則有 micro-average 與 macro-average 兩種平均方法，定義如下：

$$\text{micro Precision} = \frac{\sum_i a_i}{\sum_i a_i + \sum_i c_i} \quad (14)$$

$$\text{micro Recall} = \frac{\sum_i a_i}{\sum_i a_i + \sum_i b_i} \quad (15)$$

$$\text{macro Precision} = \frac{1}{m}\sum_{i=1}^{m} \frac{a_i}{a_i + c_i} \quad (16)$$

$$\text{macro Recall} = \frac{1}{m}\sum_{i=1}^{m} \frac{a_i}{a_i + b_i} \quad (17)$$

其中註標 i 是指第 i 個類別，而 m 是類別的總數。

macro-average 考慮每個類別的成效後再做平均，因此容易受到大量的小類別影響。在展示成效時，經常這兩種平均數據都報告出來，以便分析比較。從數值上來看，通常 micro-average 都遠高於 macro-average。

三、研究方法

本論文主要是收集前人對於自動化文件分類，所做相關研究提出之分類方法加以整合，建立出一個自動化的實驗平台，研究步驟如下：

研究目的及範圍

與指導老師討論，決定研究的目的、及範圍，並規畫進行步驟與進行時程，以利進行。

相關資料收集、整理

由與本論文主要是收集，自動化文件分類的相關研究加以整合，建立出一個自動化的實驗平台，因此，需收集相關研究資料，先運用網際網路收集相關文獻，然後加以整理、分析，方便下一步驟進行。

系統功能設計與分析

依據收集來的相關文獻資料加以整合、分析，來決定自動化實驗平台，因該要具有的功能。

自動化系統平台建置

自動化分類平台的建置，主要運用 Borland C++Builder 6 撰寫程式，將上一個步驟中所設計的功能一一實做出來，將在下一章節詳述製作過程。

實驗與參數調整

事先設定實驗參數，並在實驗中，根據實驗結果進行調整。本論文將不斷地調整實驗參數，找出在各種條件下，分類方法的最佳參數組合。

結果評估

根據不同的實驗參數，加以檢討，來決定在那種條件下那種參數最合適，並記錄每次實驗過後的結果。

結論

依據最後的評估結果，來探討在何種條件

下，哪一種分類方法較具有顯著效果，讓使用者可以依據各種不同的條件採取最佳的分類方法。並期望此論文研究能夠做為未來相關研究的實驗平台。

四、研究成果

4.1 系統平台架構

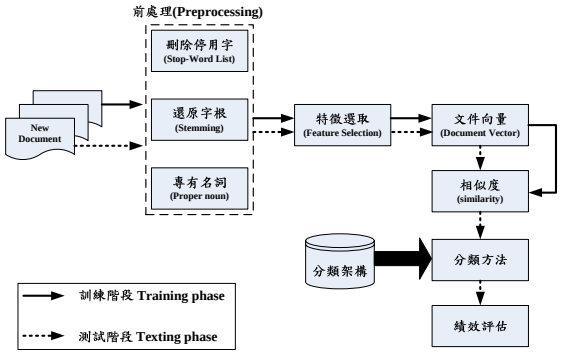

圖 5：自動化文件分類平台架構圖

圖 6：自動化文件分類平台

4.2 功能實作－前處理

圖形介面：

圖 7：前處理步驟圖形化介面

上圖為我們所設計的程式介面，使用者可以自由選擇是否對文章作 Stop-word list 或是否做 Stemming 的處理，輸入檔採用*.txt 的文字檔，目前使用路透社 21578 來做測試，檔案名稱 reut00001.txt~reut21578.txt
輸出檔部份，輸出處理完成的檔案，檔案名稱也是 reut00001.txt~reut21578.txt。

資料表：

在這個功能中我們使用到了三個資料表分別為 Stoplist、IrregularStem、keywordInfo 如下：

表 2：Stoplist 資料表

資料表名稱		Stoplist	
欄位名稱	型態	term	文字

此資料表為 Stop-word list 記錄著一些文章較常出現且無意義的字，若在文章中有這些字出現的話，可考慮刪除以縮減維度。

表 3：IrregularStem 資料表

資料表名稱		IrregularStem	
欄位名稱	型態	Variant	文字
		Original	文字

此資料表為 Stemming 處理用主要的目的是要將字尾是不規則變化的字還原，欄位 Variant 紀錄字尾為不規則變化的字，欄位 Original 則是還原後的字。

表 4：keywordInfo 資料表

資料表名稱		keywordInfo	
欄位名稱	型態	k_id	數字
		term	文字
		freq	數字
		d_freq	數字
		lastrd	數字

此資料表用於儲存，前處理完成的字，以供後續處理之用，欄位 k_id 的用處在於，給予每一個從文章中讀取出來的字，一個特定的 id(此 id 不可重複)、欄位 term 記錄此字詞、欄位 freq 記錄此字出現在所有文章的總次數、欄位 d_freq 紀錄有幾篇文章出現過此字、欄位 lastrd 記錄最後出現此字的文章 id。

在這邊要注意的是就算使用者不勾選 stop-word list 及 stemming，程式仍然會對文章做一些基本的處理，即是將大寫轉換成小寫，將非英文字字元轉換成空白字元，向式段落，數字，和一些特殊符號。

下方是 reut00001.txt 針對不同選項所輸出的結果:
輸入檔：

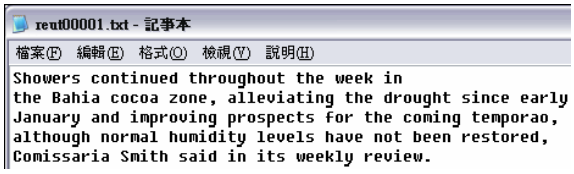

圖 8：輸入之文字檔

輸出檔：

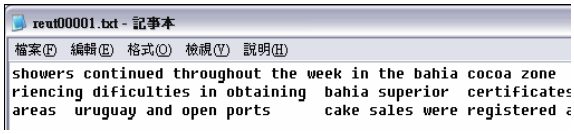

圖 9：不勾選 stop-word 及 stemming 輸出結果

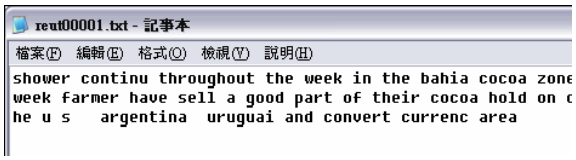

圖 10：不勾選 stop-word 勾選 stemming 輸出結果

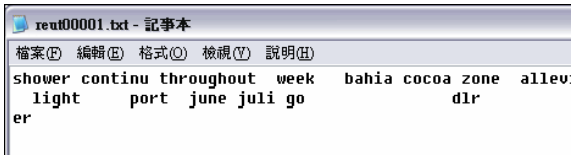

圖 11：勾選 stop-word 及 stemming 輸出結果

不難發現到，勾選 stop-word 一些較常出現的字就會被去除，像是 and、the

前處理的主要目的在於縮減文章向量的維度，以增進分類的速度，由上面可看出我們已經達成我們想要的結果，至於是否會得到較好的分類成效，我們留置後面討論。

4.3 功能實作－特徵擷取及文件向量轉換

圖形介面：

圖 12：特徵擷取圖形化介面

上圖為特徵擷取及文件向量轉換的圖形化介面，針對一些較常用的特徵擷取方法我們把它們加以整合，使用者可以很輕鬆的從上面的選項來選擇自己所需之特徵擷取方法，擷取完成後便將其轉換為文件向量輸出。

資料表：

在這個功能中我們只使用到了一個資料表 keywordInfo，來存取前處理完成的資料，在前處理中，我們已經將所有可能會出現在資料集裡面的字，都給予了一個特定的 id，以及計算了每個字的 DF 值，和記錄在資料集裡總共出現的次數。

輸入檔：

此處輸入檔為前處理完成後所輸出的文字檔，透過讀取每一篇文章，來擷取其特徵。

輸出檔：

將每個文字檔轉換成文件向量，一樣是存成 *.txt 的文字檔，檔案名稱也是為 reut00001~reut21578

4.4 功能實作－相似度的比較

圖形介面：

圖 13：相似度比較圖形化介面

上方為本次實驗所作之相似度的圖形介面，目前在文件分類中最常用來計算相似度的就是 cosine 了，Dice、Jaccard 也有人在使用，故也把們列入選項之中。

資料表：

表 5：news_info 資料表

資料表名稱		news_info	
欄位名稱	型態	newid	數字
		lewissplit	文字

欄位 newid 記錄著路透社 21578 中，每篇文章的編號、欄位 lewissplit 記錄此篇文章是屬於訓練集或者是測試集，在相似度比較中我們需將測試集的每一篇文章，跟訓練集的每一篇文章來做比較，可以藉由讀取此表來取得 newsid 方便做比較。

表 6：SimilarNews 資料表

資料表名稱		SimilarNews	
欄位名稱	型態	newid	數字
		s_newid	數字
		value	數字

此資料表用來記錄相似度比較完之後的最相似文章的 id 及其相似度。欄位 newid 紀錄測試集文章的 id、欄位 s_newid 用來記錄與此測試集文章最相似的訓練集文章 id，欄位 value 用來記錄此兩篇文章間的相似度。

輸入檔：

經過向量轉換之文字檔

五、結論

本論文所要研究的並不是提出新的分類方法，而是搜集過去所提出各種分類方法加以整合，希望開發一個較完整的自動化文件分類實驗平台，讓使用可能夠在實驗平台上，很輕鬆的設置各種實驗參數，並且輸出完整的實驗結果。

在程式開發的階段中碰到了許多問題，尤其是在剛開始要撰寫程式的時候，完全不知道要從何下手，後來慢慢的一步步把問題解決，當中看了許多別人寫的程式，並從中學習經驗，相信未來在撰寫程式時就會比現在要容易得多了。

六、參考文獻

[1]莊慧美，「以智慧型計算方法探討文件分類」，國立屏東科技大學，碩士論文，2000/7/26

[2]高志強，「組合自動化文件分類技術之研究-以專利文件分析為例」，國立中原大學，碩士論文，2004/7

[3]杜海倫，「以標題進行新聞自動分類」，國立清華大學，碩士論文，1999

[4]古倫維，「中英文新聞文件主題偵測方法之研究」，國立台灣大學，碩士論文，2000

[5]林政男，「以共現語詞為基礎的特徵選取在文件自動分類上之研究」，私立銘傳大學，碩士論文，2004/6

[6]賴榮滄，「中文郵件分類器之設計及實作」，私立逢甲大學，碩士論文，2002/6

[7]曾元顯，「文件主題自動分類成效因素探討」，私立輔仁大學，「中國圖書館學會會報」，2002/6，第 68 期，62-83 頁

[8]陳昭安「建構試題自動分類系統之研究－以MOCC術科試題為例」,國立台灣師範大學,2002
[9]楊允言,謝清俊,陳淑美,陳克健,「中文文件自動分類之研究」,第六屆計算語言學研討會論文集,p.217-233,1993
[10]王以誠,「知識內容分類（Taxonomy）的價值」,www.gorilla.com.tw,2003
[11] 林頌華,新聞標題自動分類,碩士論文,國立清華大學資訊工程系,民國 88 年 6 月
[12]Farbrizio S,"Machine Learning in Automated Text Categorization,"ACM Computing Surveys, Vol.34, NO.1, March 2002,pp.1-47
[13]Salton G.,McGill M.,"Introducion to Modern Information Retrieval,"McGraw-Hill New York,1983
[14]Salton G.,Buckley C.,"Term-weighting approaches in automatic text retrieval,"Information Process, man,24,5,1988,pp513-523.
[15] M. F. Porter, "An Algorithm for Suffix Stripping," Program 14 (3) 1980, pp. 130-137.
[16]Yiming Yang and Xin Liu, "A Re-Examination of Text Categorization Methods,"Proceedings of the 22nd Annual International ACM-SIGIR Conference onResearch and Development in Information Retrieval, 1999, Pages 42 – 49.
[17] Wai Lam and Chao Yang Ho, "Using a Generalized Instance Set for Automatic Text Categorization," Proceedings of the 21st Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1998, Pages 81 – 89.
[18] William B. Frakes, Ricardo Baeza-Yates, "Information Retrieval: Data Structure and Algorithm," Prentice Hall, 1992.
[19] Tokunaga, T. and M. Iwayama. 1994. Text Categorization Based onWeighted Inverse Document Frequency. Technical Report 0918-2802, Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan

附錄一

Port Stemming

我們使用 Stemming 的目的是為了增進搜尋的效率,將字尾的變化去除掉,例如,swimming -> swim,如此一來,可以將儲存的空間縮小,並且可以讓使用者所下的 query 更輕易地找到所需資訊。我們所採用的是 Porter 的演算法,以下概略介紹 Porter Algorithm 的步驟:

Step1:

將字尾有母音的 es、e、ed、y 替換掉,如 agreed --> agre。

Step2:

將字尾為 tional、fulness、iveness 等,替換成 tion、ful、ive 等等。

Step3:

將字尾為 icate、iveness、alize 等,替換成 ic、ive、al 等等。

Step4:

刪除剩餘的標準字尾,例如 al、ance、er、ic 等等。

Step5:

去除字尾沒有母音的 e。

這樣能將許多字母的變化型去除掉,減少資料儲存的空間,並且能搜尋出可能為使用者想要的資訊。