中華大學資訊工程學系

100 學年度專題製作期末報告

網際網路熱門搜尋偵測-遊戲產業為例

專題生：B09602007 吳中峻
B09602090 陳雋峰
指導教授:周智勳　教授
專題編號:PRJ2011-CSIE-10035
執行期間 民國 100 年 3 月至 101 年 6 月

目錄

一、 專題簡介:

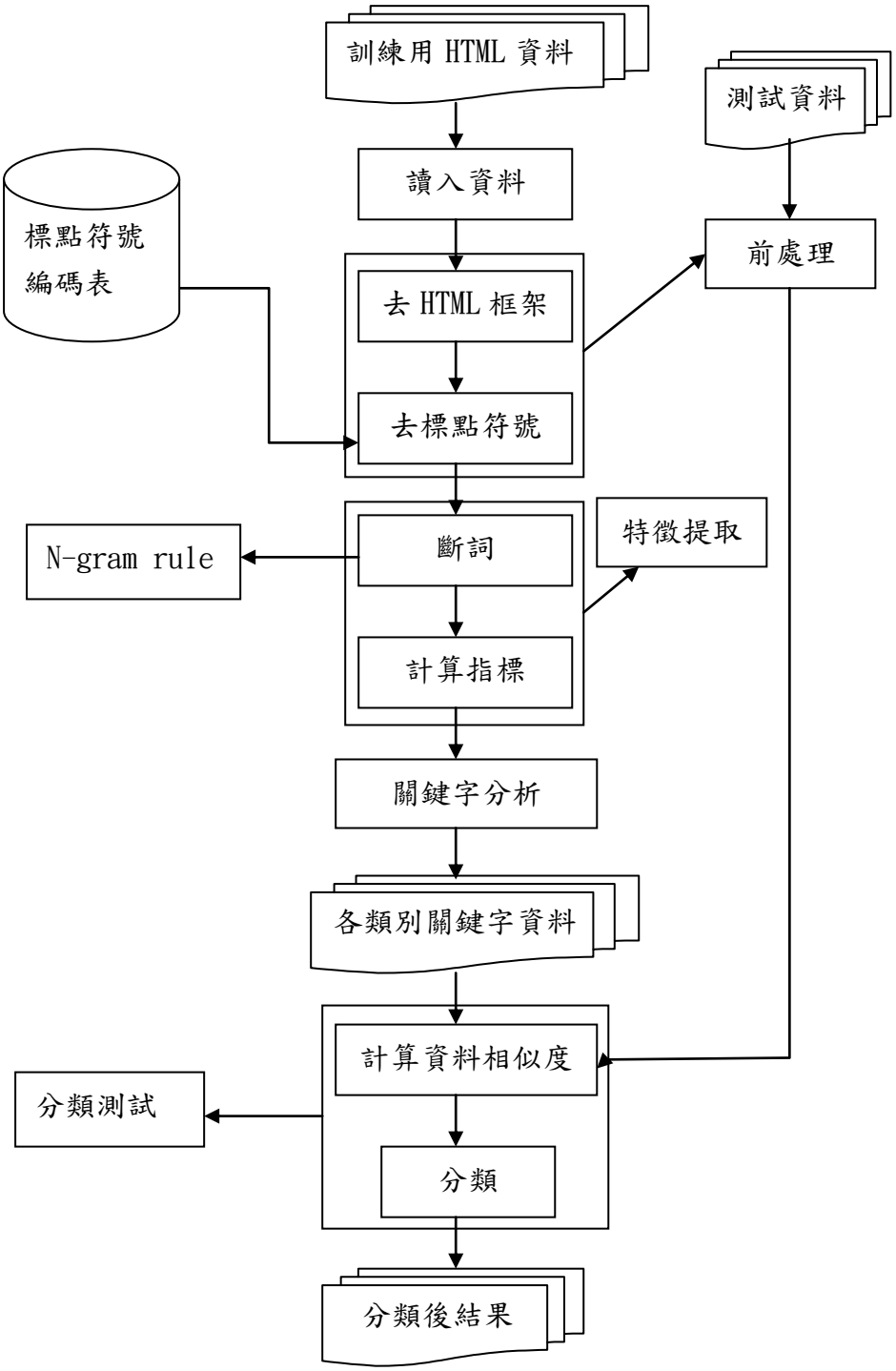
利用網際網路來搜尋資料，在現代社會是相當普及的行為，在這份專題中，希望能做到對網際網路蒐集來的遊戲相關文件與資料，進行自動分類，增加搜尋的準確度與效率。

網際網路在現今的社會中，扮演著不可或缺的角色，大量的訊息與資料在網際網路中，如何有效的利用網際網路進行資料搜尋，這份專題，希望能對大量的資料進行有效率的處理與分析，增加搜尋的效率。

這份專題主要鎖定在遊戲領域方面，首先在網際網路中收集遊戲領域相關資料，這些資料分為五大類別，對各類別的資料進行讀入的動作，同時將資料內的框架去除留下文字部分，文字部分經過斷詞處理，去掉標點符號以及虛詞後，對處理後的資料作 n-gram， n-gram 後的詞透過下面四項指標 term frequency，document frequency，conformity，uniformity ，算出相關數值後，透過數值分析得出各類別的關鍵字，即可利用關鍵字對文件進行自動分類，將文件放入正確的類別中。

二、　研究方法及步驟

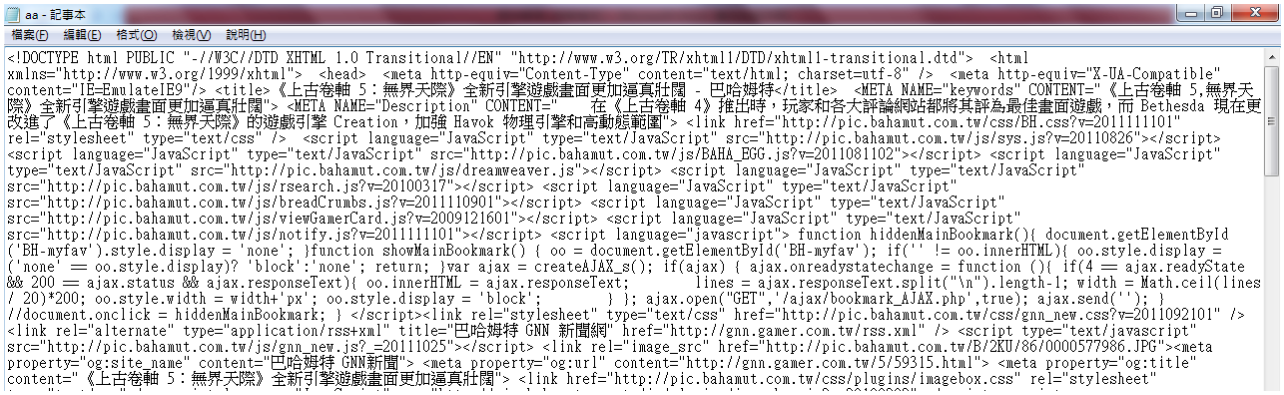
圖表 1



本次專題的研究方法及步驟如圖表 1 文件資料首先經由前處理，將 html 格式的資料中，屬於網頁框架的部分去除留下文字部分，斷詞將標點符號以及虛詞去除，同時經 n-gram 演算法處理後，配合詞庫選出關鍵字的候選詞，關鍵字分析經四項關鍵字評估指標選出合適的關鍵字，新進來的資料透過這些關鍵字便可自動分類到合適的資料類別。

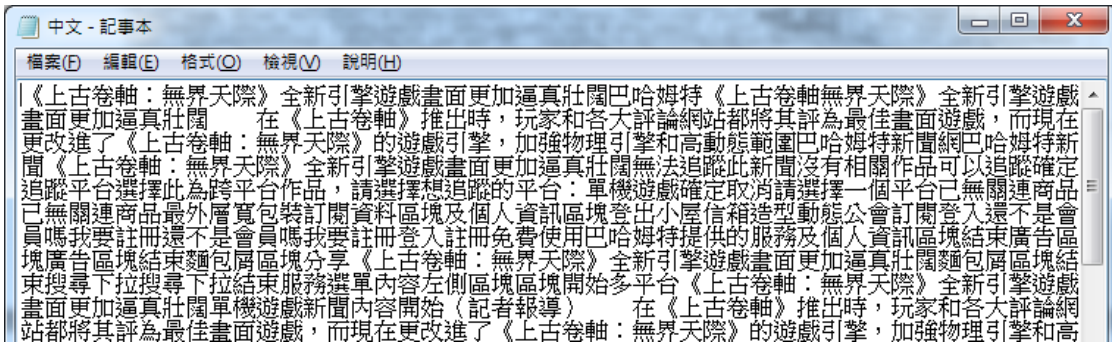
文件資料- 事先準備分類過的文件資料，讓程式獲得各類別的基礎資料，進而對未知的新資料進行自動分類處理。

下圖為剛讀入的 HTML 資料

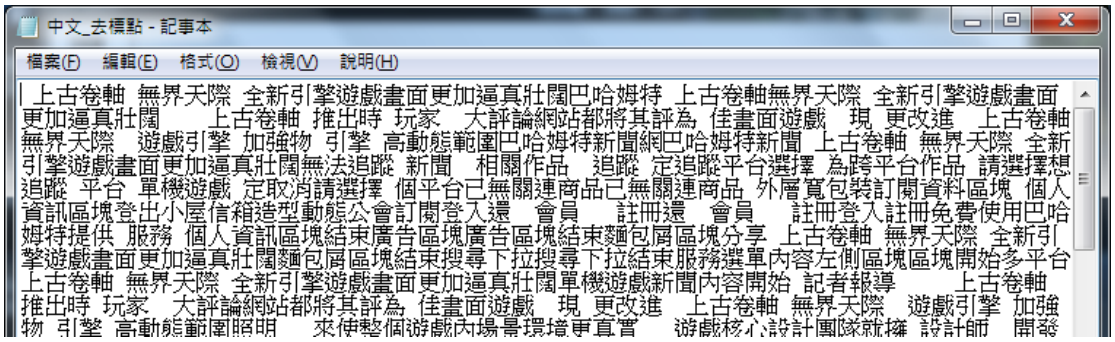


(1) 前處理

去 HTML 框架：將 HTML 文件中的數字、表格與框架移除，留下文字內容，處理後的文件如下圖所示。



去除標點符號：透過讀入的標點符號編碼表，將標點符號、虛詞、特殊字元去除，處理後的文件如下圖所示。



(2) 特徵提取

斷詞：在文件資料經過前處理後，文件中只剩下文字資料，對於英文而言，空白以及標點符號 能有效的區分出每一個詞彙，但是對於中文來說卻沒有明顯的一個分界可用於區分每一個詞彙，因此需要有一個有效的機制來對中文句子進行分析處理來區分每一個詞彙，在斷詞方面有很多的演算法廣為使用， long word priority rule , maximum matching rule , PAT tree-based rule , probabilistic learning rule and the n-gram rule ，在這份專題中使用的是 n-gram rule ，n-gram rule 的運作方式如下

Step 1.刪除標點符號。

Step 2.刪除虛詞，在這份專題中虛詞的範例如圖表 TABLE2。

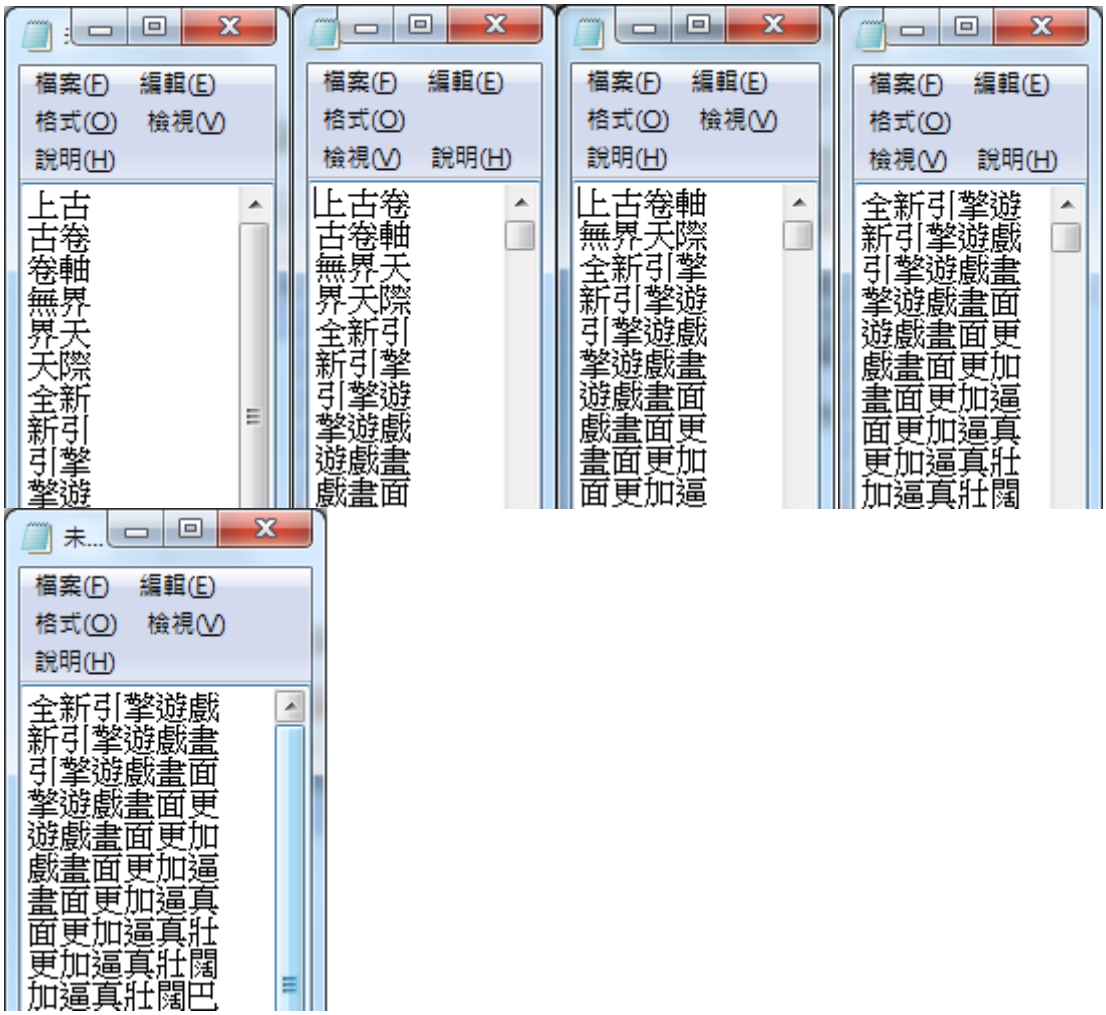
Step 3.透過 n-gram rule 整理出 2-gram 3-gram 4-gram 的詞彙，並刪除只出現一次的詞彙。

Step 4.保留 2-gram 詞彙, 3-gram 4-gram 透過詞庫提取需要的詞彙。詞庫中包含 43028 個 3-gram 與 4-gram 的常用詞彙。

TABLE 2. Function words.

了	也	不	之	什	及	且	乎	以	他	可	另	它	未
必	去	再	如	亦	各	在	她	有	此	而	何	但	你
即	否	吧	呀	我	沒	並	些	幾	最	僅	嗎	愈	理
該	嘛	確	和	啦	由	是	未	要	某	哇	咱	很	怎
一	二	三	四	五	六	七	八	九	十	百	千	萬	億

下圖為經處理後 2-GRAM、3-GRAM、4-GRAM、5-GRAM、6-GRAM 資料。



將各個做 N-GRAM 後僅出現一次的詞去除，並計錄出現過次數，處理後的文件如下圖所示。

關鍵字分析：處理後留下的關鍵字候選詞，經由下方四個演算法算出四個關鍵字特徵值：

k_i: candidate keyword
C_j: class j
t_{ij}: times of k_i appearing in C_j
m: the total number of candidate keywords
n: the number of classes
l_{ij}: the number of documents in class C_j containing candidate keyword k_i
l_j: the number of documents in class C_j
d_k: document k
tf_{ik}: the times a candidate keyword k_i appears in document d_k of class C_j

Term frequency TF_{ij}

$$TF_{ij} = \frac{t_{ij} / \sum_{i=1}^{m} t_{ij}}{\sum_{j=1}^{n} \left(t_{ij} / \sum_{i=1}^{m} t_{ij} \right)}$$

：計算該詞在該類別中出現的頻率相較於在所有類別中的比例。

Document frequency DF_{ij}

$$DF_{ij} = \frac{l_{ij}/l_j}{\sum_{j=1}^{n} (l_{ij}/l_j)}$$

：計算該詞在該類別中文件內含有該詞的比例相較於該詞在所有類別文件中的比例。

Uniformity U_{ij}

$$U_{ij} = -\sum_{k=1}^{l_j} q_{ik} \log q_{ik},$$

$$q_{ik} = \frac{tf_{ik}}{\sum_{k=1}^{l_j} tf_{ik}}$$

：計算該類別文件中該詞出現次數比例相較於在該類別中其他文件出現次數的比例。

Conformity CF_i

$$CF_i = -\sum_{j=1}^{n} d_{ij} \log d_{ij},$$

$$d_{ij} = \frac{l_{ij}}{\sum_{j=1}^{n} l_{ij}}$$

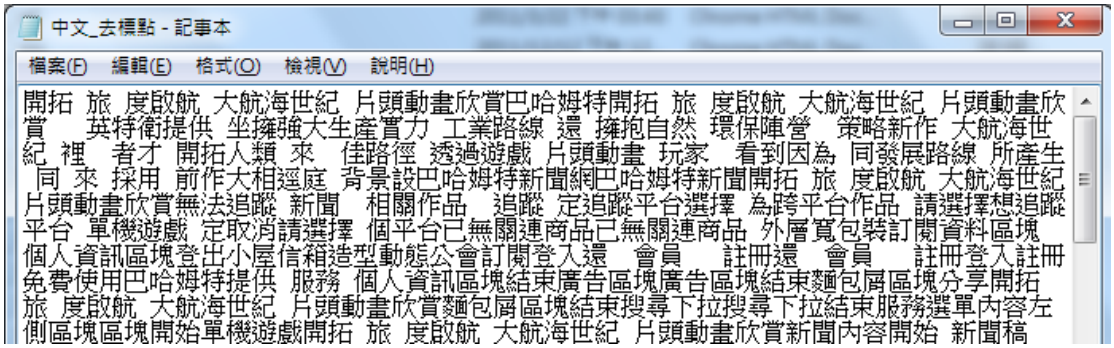
Conformity – 計算該類別文件中含有該詞的數量相較於所有類別文件中含有該詞的數量。

根據四個關鍵字特徵值演算法選出各類別代表關鍵字如下圖所示。

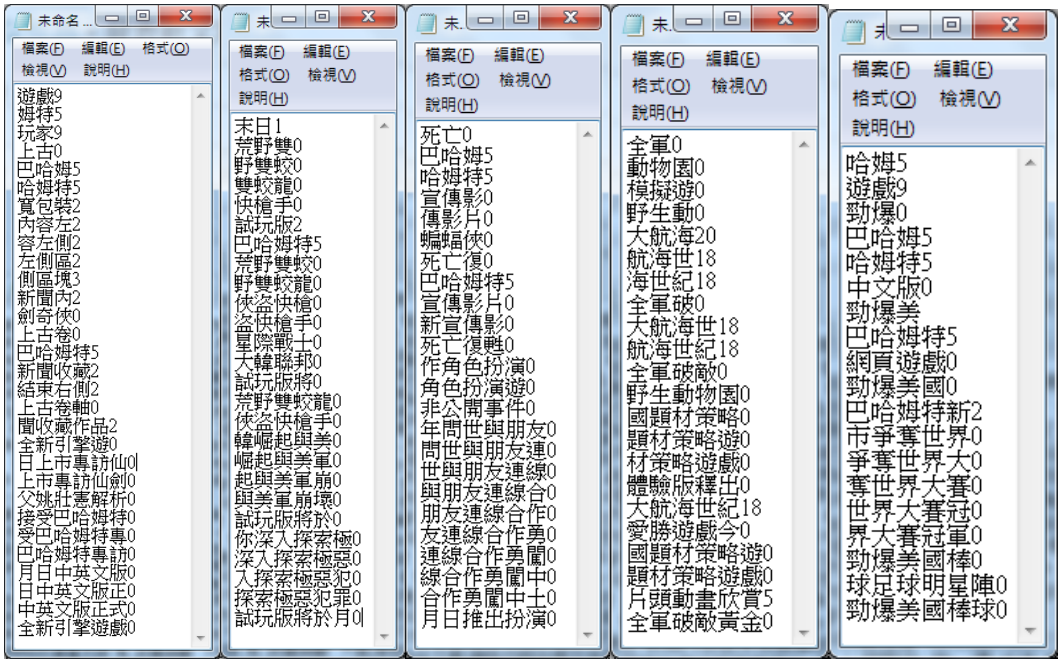
角色扮演類：射擊類：　動作類：　策略模擬類：　運動類：

(3) 分類測試

測試資料經前處理後如下圖所示。



計算相似度：計算文章在各類別出現的關鍵字數量，如下圖所示。



關鍵字的出現數量分別為：

角色扮演類:57　　射擊類:8　　動作類:15

策略模擬類:115　　運動類:31

從上面關鍵字的出現數量可以看出，策略模擬類的關鍵字出現數量最多有 115 次，因此最後將此文件分類到策略模擬類這個類別。

三、 成果

測試環境

作業系統:Microsoft Windows 7

CPU:Intel i5

記憶體:8G

硬碟:320G

開發環境：Visual Stuido 2010

開發語言：C++

訓練用文章量為各類別 10 篇，總共 50 篇。

以下為另外 50 篇文章讀入後分類的結果:

網頁分類＼判定結果	角色扮演類	射擊類	動作類	策略模擬類	運動類
角色扮演類	4	0	0	0	11
射擊類	1	6	0	0	4
動作類	1	0	6	0	1
策略模擬類	0	0	0	4	6
運動類	0	0	0	0	6

	角色扮演類	射擊類	動作類	策略模擬類	運動類	總數
讀入文章數量	15	11	8	10	6	50
判定正確數量	4	6	6	4	6	26
判定正確率	26.67%	54.55%	75.00%	40.00%	100.00%	52.00%

分類判定時，在誤判的情況下總是會將文章歸類到運動類。我們手動檢視運動類文章和選出來的關鍵字後發現，運動類遊戲文章比起其他類別，比較沒有頻繁出現的共通字，因此在篩選關鍵字時，選到了如巴哈姆特、遊戲…等在其他類別的文章也會頻繁出現的字為關鍵字，因此當讀入文件的重要辭彙出現頻率不夠時，便容易誤判至運動類。

四、心得

1.去掉網頁框架 - 這部分一開始採用，尋找"
"做為文字內容開始以及結束的關鍵字，但"
"在 HTML 中還有其他的特殊用法，該用法會導致程式在判定上產生失誤，無法正確的去除掉網頁框架。

解決方式 - 網頁的 HTML 語法多以英文撰寫，所使用的英文字皆可用 ascii 的編碼格式表示，使用<ctype.h>中的 isascii 指令，傳入的值若為 ascii 格式內的文字回傳非 0 的值，反之則為 0。

透過這個指令便能有效的將網頁框架去除，留下內文。

[程式範例 1]

2.去除中文標點符號(UTF-8 編碼格式) - 在除去中文標點符號，一開始對於編碼類型毫無概念，便認定了要用 big5 的編碼格式下去做，但無法去除掉標點符號。

解決方式 - 在查詢資料後發現，事實上多數的網頁，為了方便不同語言使用者之間的交流，採用 UTF-8 的編碼格式下去撰寫[參考文獻 1]，最早中文的編碼的格式皆為 big5 格式，而簡體中文則為 GB 格式，當設計出來的軟體開始外銷後便發現了問題，同一個字在不同的編碼格式中可能代表了別的字元，甚至在該編碼中並沒有這個字

元，便無法正常顯示在使用者的介面上，為了處理這個問題，就產生了 Unicode 俗稱萬國碼，在非 Unicode 環境下，由於不同國家和地區採用的字符集不一致，很可能出現無法正常顯示所有字元的情況。在這種情況下，一些非英語的歐洲語言編寫的軟體和文檔很可能出現亂碼。而將內碼表設定為相應語言中文處理又會出現問題， 因此有了 Unicode，內碼表技術現在廣泛為各種平台所採用。

在確認 HTML 資料為 Unicode - UTF8 格式後，對常出現的中文標點建立 UTF-8 格式的編碼表如表 1 所示，藉由編碼表得知那些字元為 UTF-8 格式下的中文標點，便能有效的將標點符號去除。

[程式範例 2]

3. 對中文做 N-gram 斷詞 - 將文章斷成 2-gram, 3-gram, 4-gram, 5-gram, 讀入詞庫做比對，在將各 gram 中沒出現在詞庫中的詞，在各 gram 中做出現次數統計，將只出現一次的詞去除，留下的便是候選詞。

[程式範例 3]

4. 在 MFC 程式中秀出處理好的文字檔案 - 遇到了 MFC 不支援 UTF-8 編碼的問題，導致顯示出來的文字檔案是亂碼，參考了網路上的資源[參考文獻 2]，透過 MultiByteToWideChar 將 UTF-8 編碼轉換成 unicode，再利用 WideCharToMultiByte 將 unicode 編碼轉換成 BIG5 碼，最後就能正

常的在 MFC 程式上秀出處理過後的文字檔案。

[程式範例 4]

參考文獻

1.http://www.unicode.org/cgi-bin/GetUnihanData.pl?codepoint=%E5%BE%97

表 1.UTF-8 的編碼對應表

{0xef, 0xbc, 0x8c}, // "，"

{0xe3, 0x80, 0x81}, // "、"

{0xe3, 0x80, 0x82}, // "。"

{0xcf, 0x86}, // "φ"

{0xcf, 0x87}, // "χ"

{0xcf, 0x88}, // "ψ"

{0xcf, 0x89}, // "ω"

[範例程式 1]

```
   for(int q=0;q<sizeof(data);q++) {
       if(isascii(data[q]))
       {
           /*english[en_counter] = data[q];
           en_counter++;*/
       }
       else
       {
           //printf("%c",data[i]);
           input[k]=data[q];
             data[q] = '\0';
           k++;
       }
   }
```

[範例程式 2]

```
int big5(int z)
{
     int j = 0;
     for(j=0;j<n+2000;j++)
    {
          test = false;
          for(int k=0;Big5Separator[k][0]!=0;k++)
          {
               if (Big5Separator[k][0] == (input[j]) && Big5Separator[k][1] ==
(input[j+1]) && Big5Separator[k][2] == (input[j+2]))
               {
                    //0xe3, 0x80, 0x80

                    test = true;
                    j = j + 2;
                    z++;
                    //z = z + 3;
               }
          }
```

```
        for(int k=0; special[k][0]!=0;k++)
        {
            if(special[k][0] == input[j] && special[k][1] == input[j+1])
            {
                test = true;
                j++;
                z++;
            }
        }

        if(test == false)
        {
            input2[z] = input[j];
            z++;
        }

    }
    return z;
}
```

[範例程式 3]

```
    for(j=0;j<=sizeof(input3);j++){
        if(input3[j]==0)
        {

       }
        else if(input3[j+1]==0){}
        else if(input3[j+2]==0){}
        else if(input3[j+3]==0){}
        else if(input3[j+4]==0){}
        else if(input3[j+5]==0){}
        else if(isascii(input3)){}
        else {
            two[k] = input3[j];
            two[k+1] = input3[j+1];
            two[k+2] = input3[j+2];
            two[k+3] = input3[j+3];
            two[k+4] = input3[j+4];
```

```
                two[k+5] = input3[j+5];
                two[k+6] = '\n';
                k=k+7;
                j=j+2;
            }
        }
```

[範例程式 4]

```
MultiByteToWideChar(CP_UTF8, 0, test3, 10000, unicode, 10000);//UTF-8 轉 Unicode
WideCharToMultiByte(950, 0, unicode, -1, cbig5, 10000, NULL, NULL);//Unicode 轉 Big5
```

五、致謝

感謝指導教授的細心指導，專題的製作上花了很長的一段時間，感謝老師在這之中的指引與督促，給予了我們很大的幫助，希望這段專題的製作經驗，能在未來給予我們幫助，指導教授周智勳老師在專題的製作過程中，適時的在關鍵點上，給我們指引方向，讓我們在專題的進度上快了很多，學長姐也給予了很多協助，這次的專題製作會是很珍貴的人生經驗。

六、參考文獻

1. Chih-Hsun Chou, Chang-Hsing Lee and Ya-Hui Chen, GA-Based Keyword Selection for the Design of an Intelligent Web Document Search System, THE COMPUTER JOURNAL, p1.~p3. , Vol. 52 No. 8, 2009

3. C How To Program 第五版 -作者：P.J.DEITEL,H.M.DEITEL

出版社: PEARSON 出版日期:2007 年 1 月 1 日

4. C++函式庫精華錄 作者：核心研究室：陳正凱，陳錦輝/編著

出版社：金禾 出版日期：2001 年 08 月 25 日