產品比對的研究

學生：楊博宇　　指導教授：吳毅成

國立交通大學電機資訊學院 資訊學程（研究所）碩士班

摘　　要

網路上產品資訊非常豐富且多樣，但是要找到自己真正需要的資訊卻不是件容易的事。一般的作法，是上各個相關網站收集資料，非常的耗費時間而且不方便。比較方便的作法是將資料存到資料庫，然後再利用查詢介面找到想要的資料。但是往往找到的是一堆相似的產品，還是需要人工判斷出相同的產品。所以本篇論文以產品資料比對，自動判斷相同產品為目標。

產品名稱、序號為辨識產品是否相同的重要條件。而比對兩個產品的名稱、序號，就像是兩個字串的比對。我們引用了最長共同子序列 Longest Common Subsequence (LCS)的概念，提出最長最多共同片段 Longest and Most Common Segments (LMCS)演算法，用來計算所有產品之間的分數，兩個產品之間分數越高代表兩個產品之間的相似度越高。並調整 LMCS 的計算權重，再以比對策略找到最相似的產品。經過調整後，回收率、精確度、相似度都可以達到 85%以上。

The Study of Product Matching

Student : Po-Yu Yang Advisor : Dr. I-Chen Wu

Degree Program of Electrical Engineering Computer Science
National Chiao Tung University

ABSTRACT

The product information is very rich and various on the web. It is difficult to find the information that we really need. The general way is to connect to all relevant website to collect product information. It is time-consuming and inconvenient very much. A more convenient way is to store the product information to the database, then utilize and inquire about interfaces to find the wanted information. Usually found a lot of similar products, and need to judge which products are the same products. Therefore, the goal of this thesis is to automatically judge that the same product by product name matching.

The products' name and serial number are important terms to judge same products. Comparing the name and serial number of products is like sequence comparison. We propose longest and most common segments (LMCS) algorithms which are based on longest common subsequence (LCS). LMCS used for calculating all products matching that higher score of LMCS to represent have higher similar degree. Adjust weight to calculate LMCS and use matching strategy in order to find the most similar products. After adjusting, the rates of recall, precision and similarity can be more than 85%.

誌　　謝

這篇論文能完成，首先需要感謝 吳毅成老師耐心且細心的指導以及朱正忠教授、留忠賢教授、陳隆彬教授提供寶貴意見。更要感謝 蘇瑞元學長不斷的給予幫助、輔導，還開發了 BODE 系統方便資料來源的萃取。還要感謝 徐健智、翁進富、汪益賢學長不時的給予關心，以及 林秉宏學長提供 String Matching 的研究心得。當然還有陪伴我度過漫漫歲月的 陳承駿同學，以及可愛的學弟們：自強、伯鈞、德彥、志祥、仕全、一芳、耀興、益洲、文峰。

感謝國家高速網路與計算中心的工作伙伴的體諒，更要感謝主管 洪淑惠副研究員的照顧。還要特別感謝家人的支持，我的父母、大哥、姊姊們的扶持，讓我一路走來倍感溫馨。因為有大家的祝福，讓我可以安心的完成學業。再次感謝大家。

謹以此論文獻給我的老婆 貴靜以及寶寶 凱丞，你們是我最重要的動力來源。

目　　錄

表 目 錄

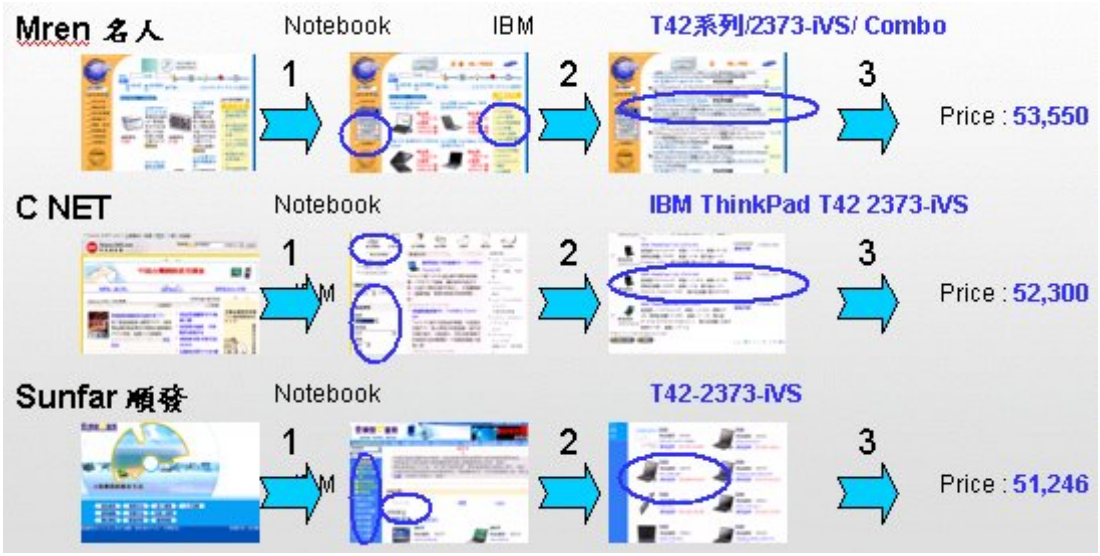
圖 目 錄

第一章 緒論

1.1 研究動機

網路上產品資訊非常豐富，但是要找到自己真正需要的資訊卻不是容易的事。雖然使用 Google, Yahoo 等知名的搜尋引擎，可以找到很多資訊。但卻因為資訊太多而造成使用者必須再次自行過濾出有用的資訊。例如當使用者想要購買 3C 產品時，除了逛街購物之外，最常採用的方法就是到知名購物網站尋找資訊瞭解產品的市場行情。

雖然很多知名 3C 網站，如：名人 3C[1]、C NET 科技資訊網[2]、順發 3C[3]、僑品電腦[4]、燦坤 3C[5]等網站都有方便使用者的查詢介面，可以幫助使用者快速找到網站內部相似的產品。但是不同網站之間，要找到相同產品就不是那麼的便利了。如圖表 1 所示，使用者要到各個網站查詢同一產品的資訊。假設要找 T42 系列 2373-iVS 的筆記型電腦，首先要連到名人網站首頁，點選“Notebook”產品類別後，接著選擇廠商類別“IBM”。然後一頁一頁的過濾找尋要找的產品，最後找到想要找的產品的價格。然後再連到 C NET 網站、順發 3C、僑品電腦[4]、燦坤 3C 等網站，並且重複同樣的找尋動作。還得將找到的資訊記錄下來進行比較，非常的耗費時間而且不方便。



圖表 1: 查詢各網站相同的產品

1.2研究目的

一般使用者要查詢不同網站的相同產品。其可能的作法是將各網站的產品資料收集起來，使用人工複製成 word 檔案或是將他們存到資料庫。然後再以人工目視過濾或者使用查詢介面，找出想要的資料。但是卻往往找出一堆相似的產品，還是需要再經過人工判斷哪些才是相同產品。

假設我們使用了資料庫的查詢介面，並且輸入查詢想要找的“T42 2373”筆記型電腦。查詢到的資料會如表格 1 所示，有很多相似的產品一起列出，因為它們都是“T42 2373”筆記型電腦。如果確定我們想要找的筆記型電腦為“T42 2373-iVS”，我們還得自己判斷、過濾其他不相同的產品，然後才能進行價格比較。如果系統可以自動顯示出相同的產品，將可以方便比對產品的價格。

如表格 2 所示，這四個網站的相同產品，其共同點是：它們都有“T42 2373-iVS”這個共同的產品資料序列。其中“2373-iVS”是產品的序號。由觀察發現，相同產品的比對與字串的比對有很大的關係。所以這篇論文以產品資料的比對，自動找出相同產品為目標。

表格 1: 相同產品不易辨識

產品名稱	價格	網站
T42 系列/2373-iVS/ COMBO	53,550	**名人**
T42 Pro 系列/2373-GRV/ Rambo	105,000	名人
IBM ThinkPad T42 2373-iVS PM 1.6G, 256MB	52,300	**C NET**
IBM ThinkPad X31 2672-iKV PM 1.5G, 256MB	41,888	C NET
IBM ThinkPad T42 2373-IVS D-1.6G / 56MB-DDR / 40G / COMBO / 14.1 吋	51,200	**僑品**
T42-2373-iVS	51,246	**順發**
T42-2373-IVP-COMBO	51,507	順發

表格 2: 產品比對與字串比對

產品名稱	價格	網站
T42 系列/2373-iVS/ COMBO	53,550	名人
IBM ThinkPad T42 2373-iVS PM 1.6G, 256MB	52,300	C NET
IBM ThinkPad T42 2373-IVS D-1.6G / 256MB-DDR / 40G / COMBO / 14.1 吋	51,200	僑品
T42-2373-iVS	51,246	順發

1.3 研究方法概述

文件比對在自動分類的應用上，如論文[6]所討論到的，以及 XML 文件分類如論文[7]所討論到的。通常會應用到中文的斷詞斷字，再針對具有分類價值的關鍵詞以向量模式、機率模式，和不同的分類比重方式來做自動分類實驗。其最主要的方式是建立關鍵詞的關聯性，依據關鍵詞的比重計算出比對後的分數來進行分類。

在音樂搜尋的應用上，如論文[8]所討論到的。利用音樂特徵建立索引機制，也是類似建立關鍵詞庫。在第二章的相關研究中，我們會介紹到論文[9]所提到的方法：使用 LCS 及 Longest Common Consecutive Subsequence (LCCS)混和的計算模式。是和我們所提出的方法比較有相關的方法。

在我們的實驗中最主要的不同點是：並沒有使用關鍵詞庫。直接針對產品資訊進行文字序列比對，而且正確找到相同產品。

在產品資訊中，產品名稱、序號是辨識產品是否相同的重要條件。比對兩個產品的名稱、序號，就像是兩個字串的比對。在字串比對的演算法中，最長共同子序列 Longest Common Subsequence (LCS) [10][11] 算是最知名也最常被應用的方法。如果純粹只使用 LCS 來計算所有產品之間的分數，並且藉此當作產品間相似度的判斷，將會有太多不相同的產品因為分數相同而誤判為相同產品。所以我們引用了 LCS 的概念，特別提出最長最多共同片段 Longest and Most Common Segments (LMCS) 演算法。並且使用 LMCS 計算所有產品之間的分數，兩個產品之間分數越高代表兩個產品之間的相似度越高。

雖然以LMCS的計算結果判斷產品相似度比以LCS計算結果來判斷產品相似度，其正確率好很多，但是還不夠令人滿意。因此我們又針對產生錯誤部分產品的特徵進行研究，發現產品名稱及序號有連續序列相同及組合的特徵。我們給予連續相同的序列較高的權重，而且相同的長度越長給予越高的權重。然後將所有產品以 LMCS 計算的比對結果全部紀錄下來。再以比對策略找到以 LMCS 方法計算後分數最高的產品，判斷該對產品為最相似的產品。另外針對序號通常包含字母、數字的特徵，調整 LMCS 的計算權重，找到準確度最高的權重值。以此方法所得到的實驗結果顯示：回收率 (Recall) 都可以達到 93%以上，準確度 (Precision) 也可達 70%左右。最後再調整 LMCS 的計算權重，回收率、精確度以及相似度都可以達到 85%以上。

1.4 論文大綱

本論文的綱要如下：

第一章　說明在各網站間比較相同產品的價格非常耗時且不方便，引發自動判斷相同產品的研究動機與目的，並簡要介紹整篇論文使用的方法以及針對實驗產生的錯誤加以改善。

第二章　介紹使用到的資料萃取工具 Browser-Oriented Data Extraction (BODE)[12][13] 系統、相關的演算法：最長共同子序列 (LCS)、以及 LCS 相關研究的論文：使用 LCS、最長共同連續子序列 Longest Common Consecutive Subsequence (LCCS) 混合計算兩字串的相似度。

第三章　詳細介紹本篇論文所使用的最長最多共同片段 Longest and Most Common Segments (LMCS) 演算法。並且介紹依據實驗結果，如何調整 LMCS 的計算權重，以改善實驗結果的正確率。以及如何運用經過 LMCS 計算後的分數，加上比對策略找到最相似的產品，並判斷其為相同產品。

第四章　以人工進行網站間產品的比對，並且將此經人工判斷所得到的數據當作驗證實驗結果好壞的判斷依據。針對實驗結果所產生的錯誤，進行問題分析。並修改實驗的方法，以得到更好的實驗數據。然後以回收率(Recall)、精確度(Precision)、相似度(Similarity)來評估實

驗的結果。

第五章　依據實驗結果所產生錯誤的問題加以討論，並提出未來可再深入研究的方向與工作。

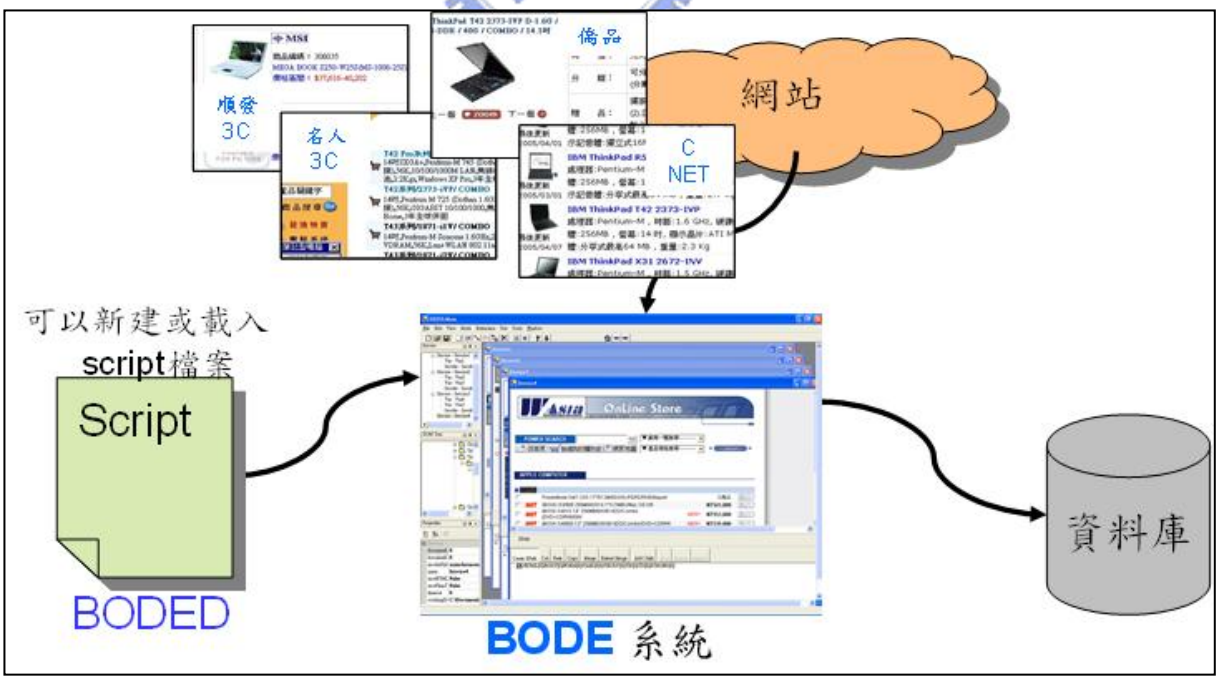
第二章 相關研究

在這一章，我們將介紹 BODE 系統，一個能自動萃取網頁資料並且儲存於資料庫的工具。介紹 LCS 演算法、LCS 的 Dynamic Programming 方法，以及 LCS 加上 LCCS 混合計算出相似度的應用。

2.1 網路資料萃取系統 (BODE)

Browser-Oriented Data Extraction (BODE)[12][13]系統是由本實驗室所研發的網路資料萃取系統。如圖表 2 所示。

BODE 系統可以連結到各個網站，模擬網頁瀏覽的方式，逐頁點選所要萃取的資料。並將所點選的資料以 XPath 的格式，存成 XML 格式的 Script 檔案。BODE 系統可以即時新增 Script 檔案來抓取網頁資料，也可以載入先前儲存的 Script 檔案，重複自動執行萃取網頁資料的動作。方便系統定時載入，執行資料萃取更新，並且儲存在資料庫中，方便分析。



圖表 2: BODE 系統

2.2 序列比對

序列比對最常應用在生物學領域[14] [15] [16] [17] [18] [19]，包括：DNA / RNA (脫氧核糖核酸 / 核糖核酸) 染色體和基因的組成部分的比對，以及氨基酸、蛋白質的序列比對...等等。因為越長的共同序列通常代表著結構或功能性越相似。LCS[10][11]則是序列比對方法中最知名且最常使用的方法之一。

2.2.1. 最長共同子序列 (LCS)

兩個產品的名稱、序號的比對，就像兩個字串的比對一樣。越相似的產品名稱、序號，有較長的共同名稱、子序列。

假設給予兩個序列：

X = [B, E, A, D, B, E, D]

Y = [B, E, D, E, D, B]

X, Y 的最長共同子序列為：LCS (X,Y) = [B, E, D, E, D]

LCS 的演算法如表格 3 所示：

表格 3: LCS 的演算法

$$C[i,j]=\begin{cases}0 & ,if \quad i=0 \quad or \quad j=0;\\ C[i-1,j-1]+1 & ,if \quad i,j>0 \, and \, Xi=Yj;\\ \max\{C[i,j-1],C[i-1,j]\} & ,if \quad i,j>0 \, and \, Xi\neq Yj;\end{cases}$$

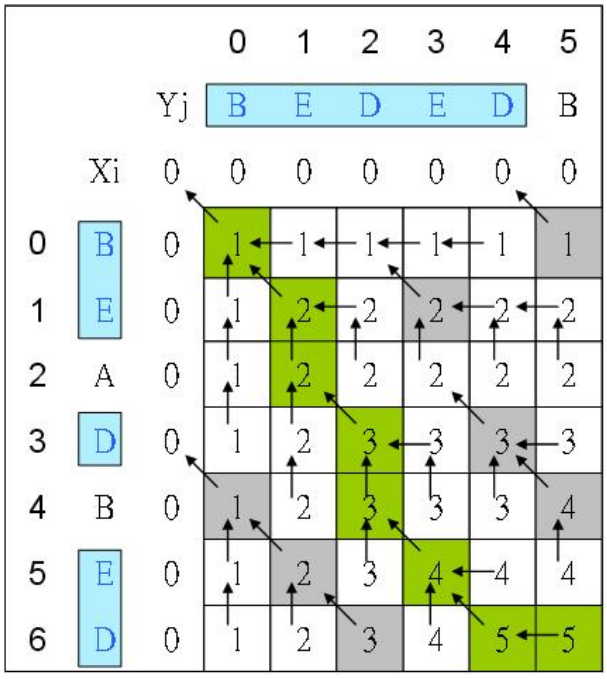
```
m ← length[X]
n ← length[Y]
for  i ← 1 to m
    do c[i, 0] ← 0
for  j ← 1 to n
    do c[0, j] ← 0

for  i ← 1 to m
```

```
do for j ← 1 to n
    do if Xi = Yj
        then c[i, j] ← c[i-1, j-1] + 1
             b[i, j] ← "↖"

       else if c[i-1, j] ≧ c[i , j-1]
            then c[i, j] ← c[i-1, j]
                 b[i, j] ← "↑"
            else c[i, j] ← c[i, j-1]
                 b[i, j] ← "←"
```

以 Dynamic Programming 方法實作 LCS 會得到如圖表 3 所示的結果。藉由 Dynamic Programming 方法，可以計算出 LCS 的分數、長度。並且紀錄其路徑，得知 LCS 的序列。



圖表 3: LCS 的 Dynamic Programming 結果

2.3 LCS 相關的研究

LCS 應用在聲音的比對上，如論文[9]所討論到的。先將使用者輸入的

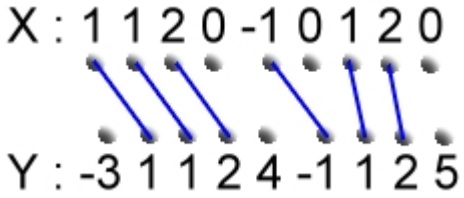
語音轉換成中介格式，再與資料庫中的中介格式音樂資料比對。實作出以語音查詢音樂的系統。

假設：使用者的語音轉換成中介格式為 X，資料庫中的中介格式音樂資料為 Y。

X = [1, 1, 2, 0, -1, 0, 1, 2, 0]

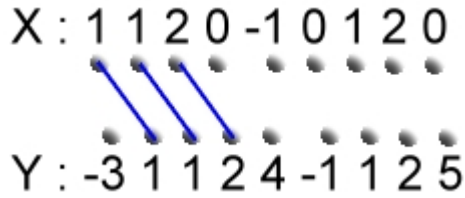
Y = [-3, 1, 1, 2, 4, -1, 1, 2, 5]

則 X,Y 的最常共同子序列為 6：LCS (X, Y) = [1, 1, 2, -1, 1, 2] = 6 ，如圖表 4 所示：



圖表 4: LCS (X, Y) = [1, 1, 2, -1, 1, 2] = 6

最長共同連續子序列(Longest Common Consecutive Subsequence) 簡稱 LCCS，是由 LCS 衍生而來，它的相同子字串必須是連續的。在這個例子裡，X,Y 的最常共同連續子序列為 3：LCCS (X, Y) = [1, 1, 2] = 3 ，如圖表 5 所示：



圖表 5: LCCS (X,Y) = [1, 1, 2] = 3

其混合的相似度評估公式(1)：

$$sim(X,Y) = \lambda \times \frac{LCS(X,Y)}{N} + \mu \times \frac{LCCS(X,Y)}{N} \quad \cdots\cdots\cdots\cdots \quad (1)$$

N : X, Y 序列的長度

λ : LCS 長度的權重

μ : LCCS 長度的權重

將經由公式(1)得到的計算結果設定一門檻值。將所有高於門檻值的結果，依分數高低順序傳回，當作查詢的結果。

第三章 產品名稱的比對演算法

3.1 最長最多共同片段 (LMCS)

我們所提出的演算法是以 LMCS 為基本，再針對產品資料序列的特徵加以改變。產品資料序列的特徵如：相同的產品都會有相同的產品序號。而產品序號都是連續的序列，且通常包含字母與數字。如表格 4 所示：

表格 4: 產品序號特徵

產品名稱	價格	網站
T42 系列/2373-iVS/ COMBO	53,550	名人
IBM ThinkPad T42 2373-iVS PM 1.6G, 256MB	52,300	C NET

產品名稱	價格	網站
T42 2373-iVS	53,550	名人
T42 2373-iVS	52,300	C NET

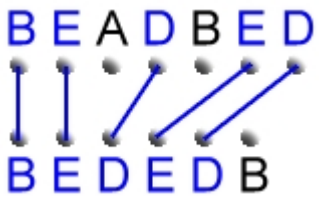
3.1.1. 共同片段 (Common Segments)

假設給予兩個序列：

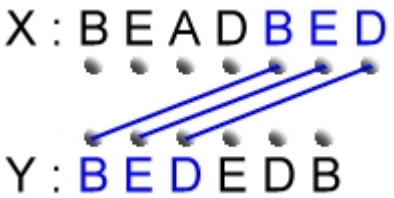
X = [B, E, A, D, B, E, D]

Y = [B, E, D, E, D, B]

其最常共同子序列 LCS (X,Y) = [B, E, D, E, D] = 5，如圖表 6 所示。最常共同連續子序列 LCCS (X,Y) = [B, E, D] = 3，如圖表 7 所示：

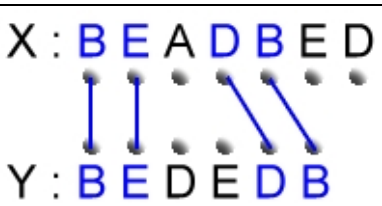


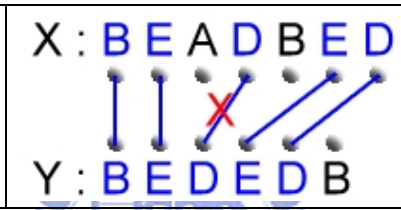
圖表 6: LCS (X,Y) = [B,E,D,E,D]

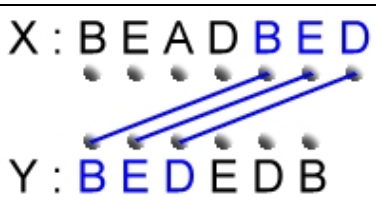


圖表 7: LCCS (X,Y) = [B,E,D]

其共同片段(CS)，CS (X,Y) = [BE, DB], [BE, ED], [BED]，如圖表 8 所示。LMCS 即是最長且最多的 CS 組合，在此例中，LMCS (X,Y) = [BE, DB] 或 [BE, ED]，LMCS (X,Y) = 4。







圖表 8: CS (X,Y) = [BE, DB], [BE, ED], [BED]

3.1.2. 為何使用 LMCS

為何使用 LMCS 呢？使用 LMCS 有什麼好處呢？我們以一個例子來說明：

假設我們要比對三個產品，X, Y, Z：

X: IBM ThinkPad T42 系列/2373-iVP/ COMBO

Y: IBM ThinkPad T42 2373-iVS PM 1.6G, 256MB

Z: IBM ThinkPad T42 2373-IVP PM 1.6G, 256MB

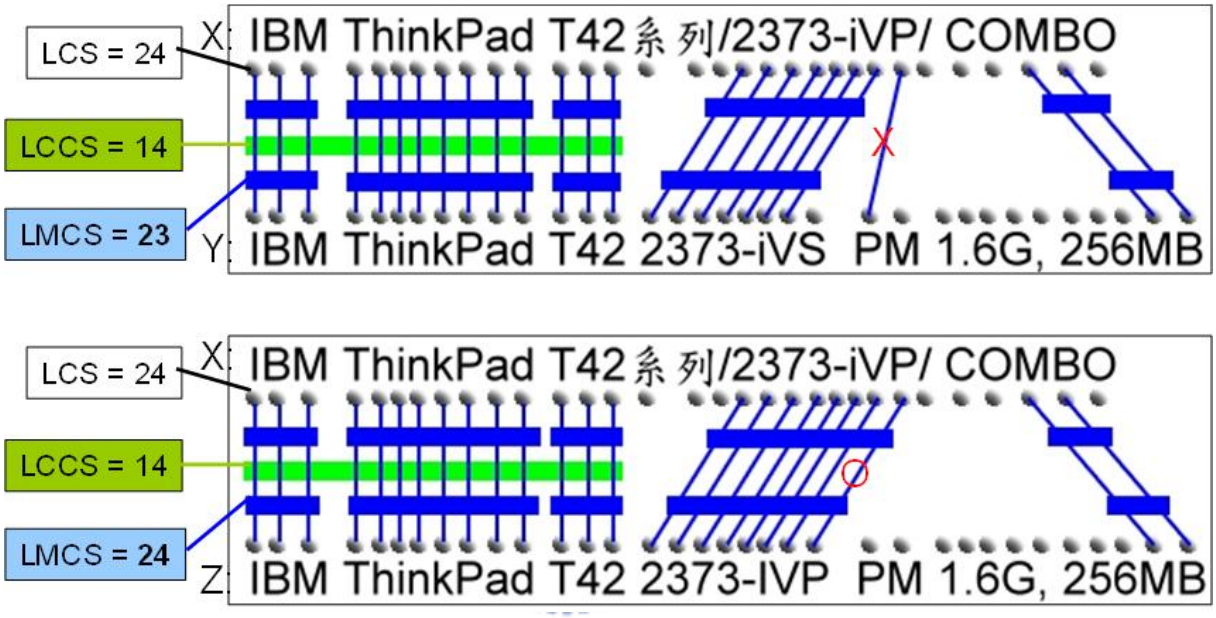
如圖表 9 所示，LCS (X,Y) = [“IBM ThinkPad T42”, “2373-iV”, “P”, “MB”] = 24，LCCS (X,Y) = [“IBM ThinkPad T42”] = 14；LCS (X,Z) = [“IBM ThinkPad T42”, “2373-iVP”, “MB”] = 24，LCCS (X,Z) = [“IBM ThinkPad T42”] = 14。

其中 LCS (X,Y) = LCS (X,Z) = 24, LCCS (X,Y) = LCCS (X,Z) = 14，

依據混合的相似度評估公式(1) 計算的結果：SIM (X,Y) = SIM (X,Z)。將無法正確判斷出 X, Z 才是相同的產品。

若是使用 LMCS 計算，LMCS (X,Y)= [“IBM ThinkPad T42”, “2373-iV”, “MB”] = 23；LMCS (X,Z)= [“IBM ThinkPad T42”, “2373-iVP”, “MB”] = 24。

LMCS(X,Z) > LMCS (X,Y)，依據分數越高代表產品相似度越高，X, Z 為最相似的產品，正確判斷 X, Z 為相同產品。



圖表 9: 使用 LMCS 計算，正確判斷 X, Z 為相同產品

3.1.3. LMCS 計算方法

LMCS 是由 LCS 衍生而來，它的特點是相同子字串必須是連續的片段。LMCS 的計算方法有三個步驟，如下：

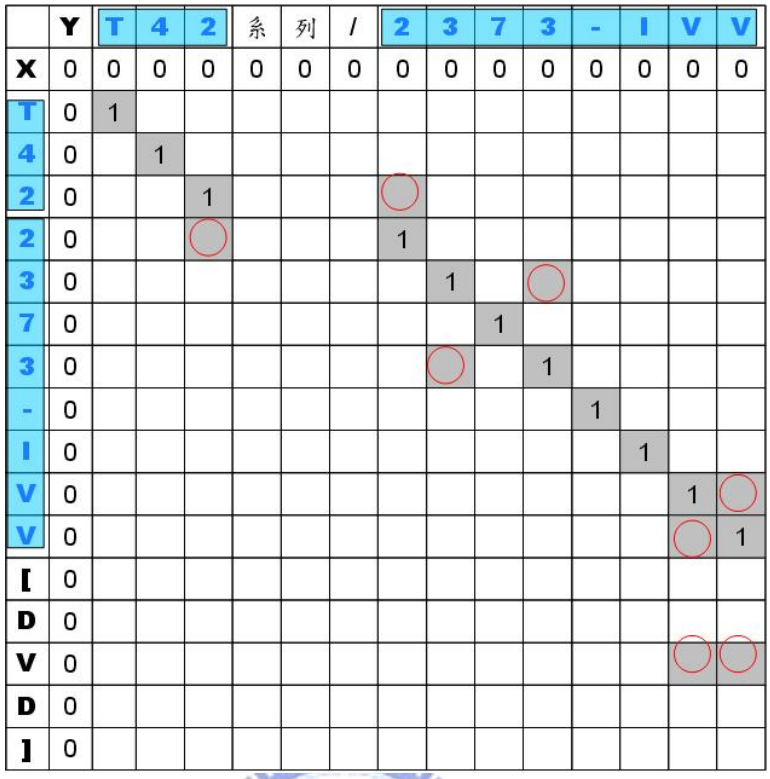
- 步驟 1：先以 LCS 的方法，求出所有相同字元的比對狀況。
- 步驟 2：將所有不連續的部分排除，不予以計算。
- 步驟 3：計算所有共同的連續片段 (CS) 的長度，並且將不重疊的所有 CS 加總。最後選擇總分最大的組合，就是所要求的 LMCS。

假設給予兩個產品名稱 X: T422373-IVV[DVD], Y: T42 系列/2373-IVV。

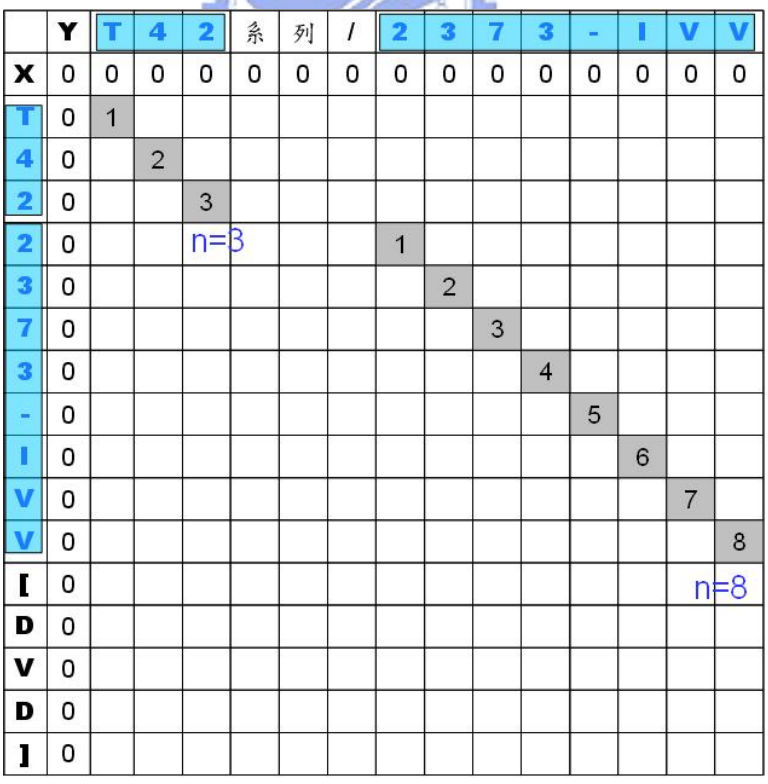
- 步驟 1：先以 LCS 的方法，求出所有相同字元的比對狀況。其結果表示如圖表 10，所有相同的字元都給予一點計分。
- 步驟 2：判斷所有 1 點計分的點右下斜角是否有連續，並且排除所有不連續的點。其排除的部分，如圖表 11 紅色圓圈的部分所示。
- 步驟 3：計算所有連續片段的長度。如圖表 12 所示，[T42]的連續長度為 3，[2373-IVV]連續的長度為 8。然後將[T42]的長度 3 加上[T42]字串後面的連續字串 [2373-IVV] 長度 8，得到最長且最多片段 LMCS = [“T42”,“2373-IVV”] = 11。

	Y	T	4	2	系	列	/	2	3	7	3	-	I	V	V
X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T	0	1													
4	0		1												
2	0			1				1							
2	0			1				1							
3	0								1		1				
7	0									1					
3	0								1		1				
-	0											1			
I	0												1		
V	0													1	1
V	0													1	1
[	0														
D	0														
V	0													1	1
D	0														
]	0														

圖表 10: 以 LCS 的方法求出所有相同字元的比對狀況



圖表 11: 紅色圓圈所示排除部分



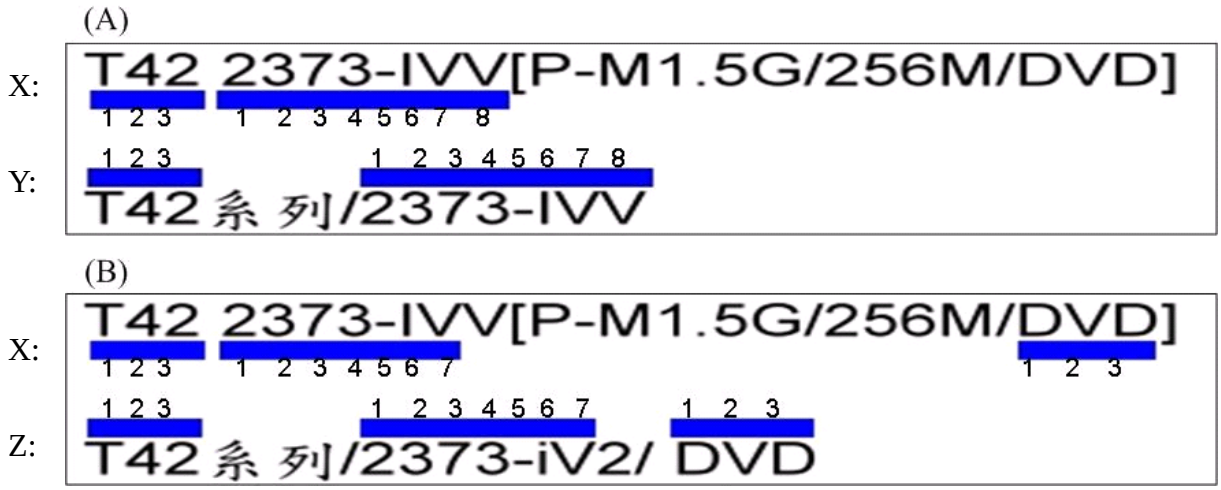
圖表 12: [T42]的連續長度為 3，[2373-IVV]連續的長度為 8。

LMCS = 3 + 8 = 11

3.1.4. LMCS 調整權重

為了讓產品序號在比對時能顯現其重要性，我們針對產品序號的特徵設計了一個權重計算方法。產品序號會是連續的、有次序性的且包含字母與數字的字串。所以我們設計了一個依據相同的字串長度越長，就給予越高的權重。並且在連續相同的字串中包含字母與數字的部分給予額外的權重加分。

以圖表 13 為例子，如果不給予額外的權重加分，則如圖 13(A)所示，LMCS(X,Y) = 3 + 8 = 11。如圖表 13(B)所示，LMCS(X,Y) = 3 + 7 + 3 = 13。則錯誤的判斷 X,Z 為最相似的產品。



圖表 13: LMCS 未加權重的計算

若連續相同的字串其長度為 n，則給予一個權重 W [n]。若連續相同的字串中包含字母與數字，則再加上第二個權重。

LMCS 的加權計算公式如公式(2)：

$$LMCS(X,Y) = \sum_{n \in Segment} W[n] \times W_2 \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (2)$$

假設：

$$W[n] = \frac{n \times (n+1)}{2} \quad \cdots\cdots (3)$$

$W_2 = 1$；若連續相同的字串包含字母與數字，則$W_2 = 2$。… (4)

將公式(3)、公式(4)代入公式(2)，得到 LMCS(X,Y) = W [3] + W [8]*2 = 6 + 72 = 78， LMCS(X,Z) = W [3] + W [7]*2 + W [3]= 6 + 56 + 6 = 68。

因為 LMCS(X,Y) 的分數較高，所以正確的判斷 X,Y 兩個產品的相似度較高。

3.1.5. 問題

我們所比對的資料是兩個網站的產品名稱、序列。每次比對都是由兩個網站各取一個產品，進行兩兩比對。每個產品都和另一個網站的所有產品比對過，且各有一個以 LMCS 計算出來的分數。剛開始做實驗時，我們假設每個網站不會陳列相同的產品兩次。亦即表示不同網站間，若有相同產品，也只會有一個，所以只保留一個比對的產品記錄。如此會造成 LMCS 分數相同時，會有漏失真正是相同產品的記錄，而造成比對結果錯誤。

為解決此一問題，我們採取了記錄所有比對的產品記錄。並且利用比對策略找出真正相同的產品。

3.2 比對策略

我們的比對策略如下：

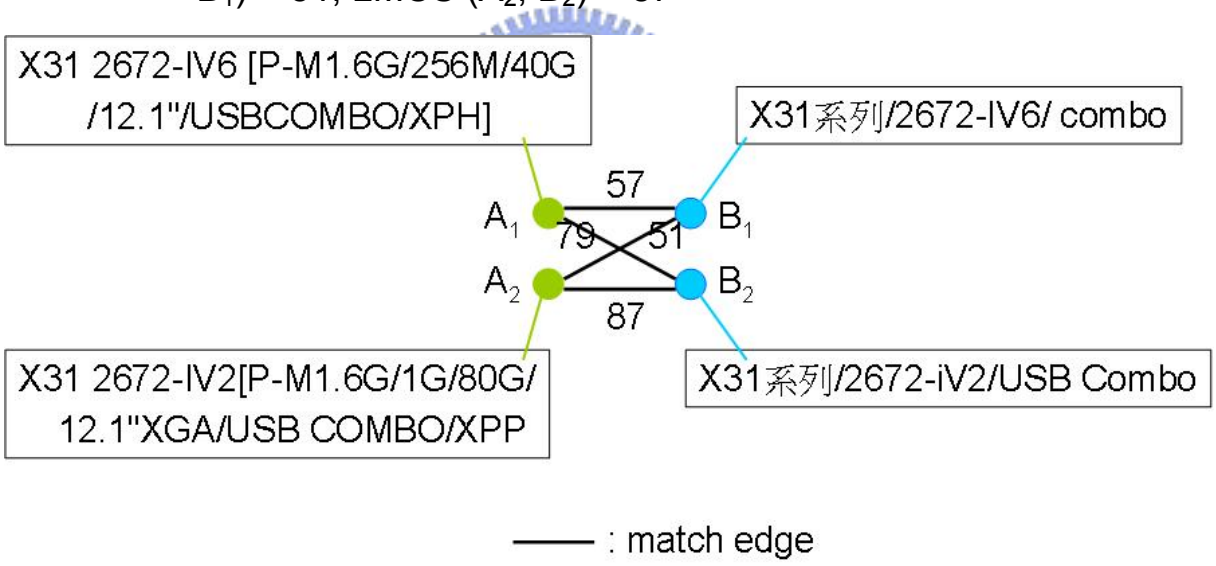
- 步驟 1：先計算所有產品的 LMCS 分數，並且將這些比對的計算結果記錄下來。
- 步驟 2：在這兩個網站的所有產品比對計算結果中，找尋分數最高的比對產品。
  - ①.這一對產品就是最相似的產品，並且標示它們是相同的產品。
  - ②.將這一對產品記錄起來，並移除所有和這對產品比對的紀錄。
- 步驟 3：重複步驟 2。並且設定門檻值，若 LMCS 分數低於門檻值，則

表示這一對產品為不相同的產品，並且停止繼續找尋分數最高的比對產品。

以下面的例子說明此一比對策略：假設有兩個網站A, B的四個產品A_1, A_2, B_1, B_2，如下：

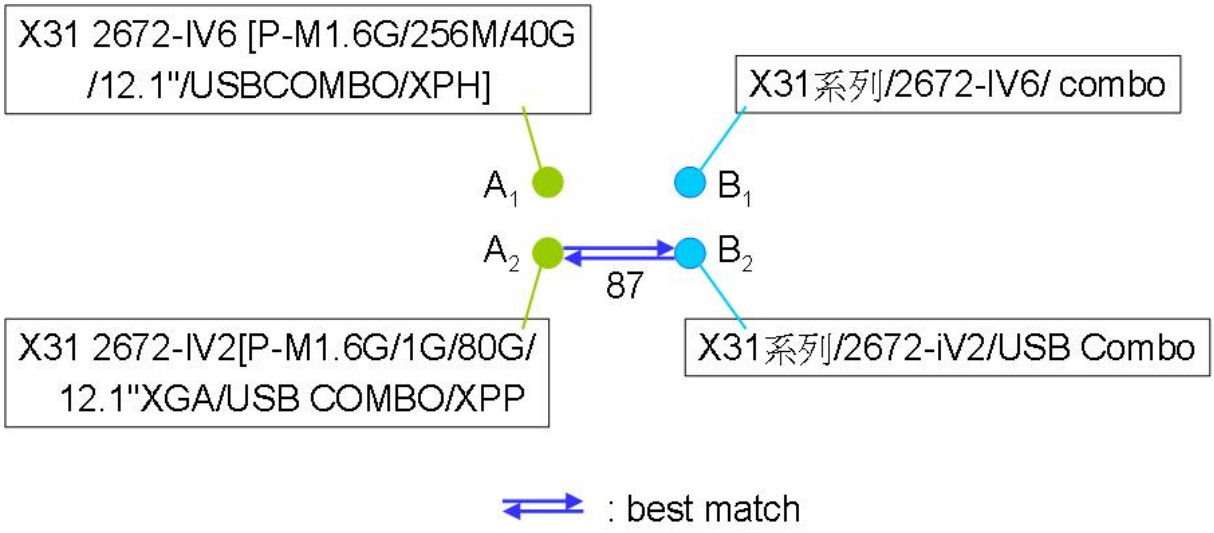
A_1	X31 2672-IV6 [P-M1.6G/256M/40G/12.1"/USBCOMBO/XPH]
A_2	X31 2672-IV2[P-M1.6G/1G/80G/12.1"XGA/USB COMBO/XPP
B_1	X31 系列/2672-IV6/ combo
B_2	X31 系列/2672-iV2/USB Combo

➤ 步驟 1：求出A_1, A_2與B_1, B_2，四個產品兩兩比對的LMCS分數。如圖表 14 所示，LMCS (A_1, B_1) = 57, LMCS (A_1, B_2) = 79, LMCS (A_2, B_1) = 51, LMCS (A_2, B_2) = 87

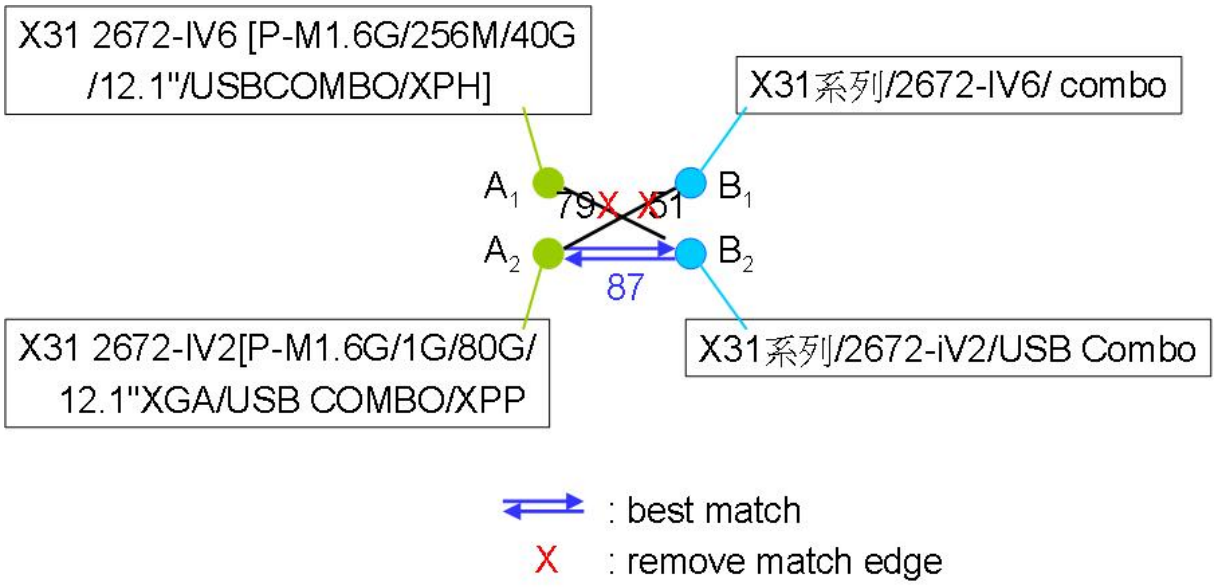


圖表 14: A_1, A_2與B_1, B_2 的LMCS分數

➤ 步驟 2：找出LMCS (A, B)的最高分。如圖表 15 所示， LMCS (A_2, B_2) = 87。接著，移除A_2, B_2的其他比對記錄：LMCS (A_1, B_2) = 79, LMCS (A_2, B_1) = 51。如圖表 16 所示。



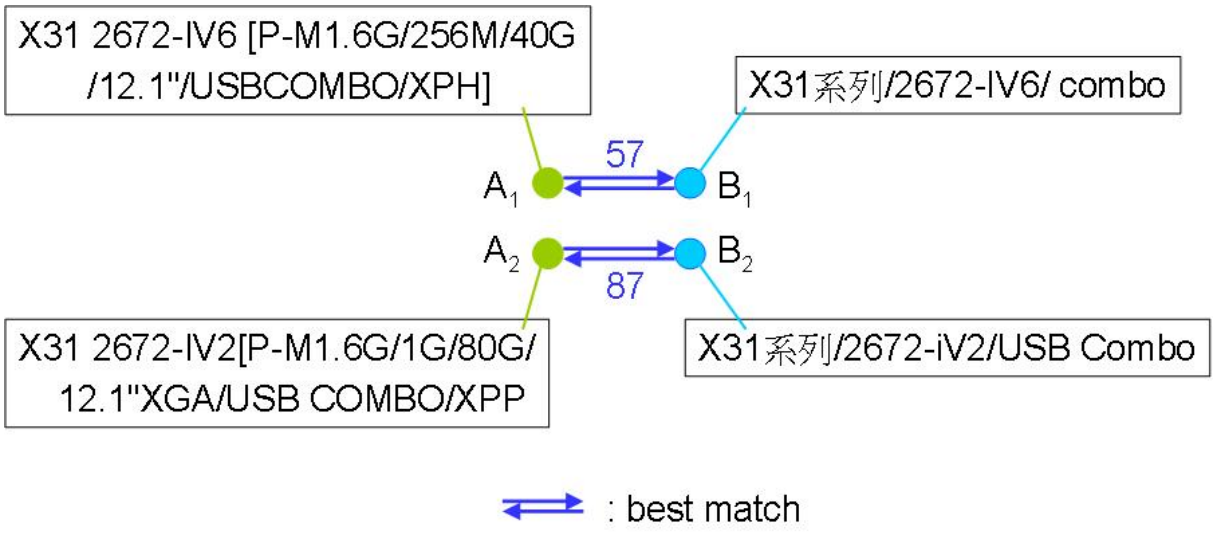
圖表 15: 找出最高分LMCS (A_2, B_2) = 87



圖表 16: 移除A_2, B_2的其他比對記錄

➤ 步驟 3：重複步驟 2，找到LMCS (A, B)的最高分：LMCS (A_1, B_1) = 57。如圖表 17 所示。

最後得到比對結果：(A_2, B_2) 是最相似的產品以及 (A_1, B_1) 是最相似的產品。


X31 2672-IV6 [P-M1.6G/256M/40G /12.1"/USBCOMBO/XPH]
X31系列/2672-IV6/ combo
57
A1
B1
A2
B2
87
X31 2672-IV2[P-M1.6G/1G/80G/ 12.1"XGA/USB COMBO/XPP
X31系列/2672-iV2/USB Combo
: best match

圖表 17: 得到比對結果

第四章 實驗結果分析與評估

4.1 實驗結果分析

產品分類的問題是一門很大的學問，我們不在這篇論文中討論，因此我們採用單一類別的產品資料來進行比對分析。在這篇論文中，我們選擇以筆記型電腦類別的產品資料當作比對分析的資料來源。並且以本實驗室所開發的 BODE 系統，將網頁的產品資料自動萃取並儲存到 MS SQL 資料庫中，再由資料庫將資料取出進行分析。

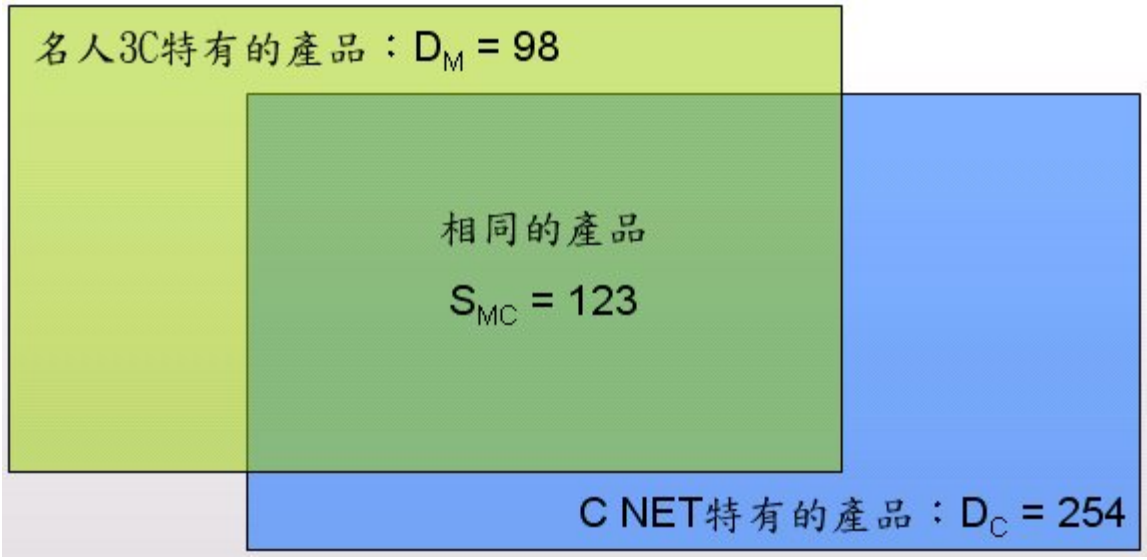
4.1.1. 產品比對分析-名人 3C 與 C NET

以名人 3C[1]、C NET 科技資訊網[2]進行產品資料的比對分析。

首先我們以人工比對的方式，分別計算出名人 3C以及C NET兩個網站在筆記型電腦這個產品類別的產品總數量，以及兩個網站之間相同與不相同產品的數量。我們訂定了一些標準定義，包括：S_{MC}表示名人 3C (Mren)與C NET 兩個網站間相同產品的數量、T_M表示名人 3C所有產品的數量、T_C表示C NET所有產品的數量、D_M表示名人 3C特有產品的數量、D_C表示C NET特有產品的數量，如表格 5 所示。然後將這兩個網站的相同產品與不同產品的分佈以圖表 18 來表示。

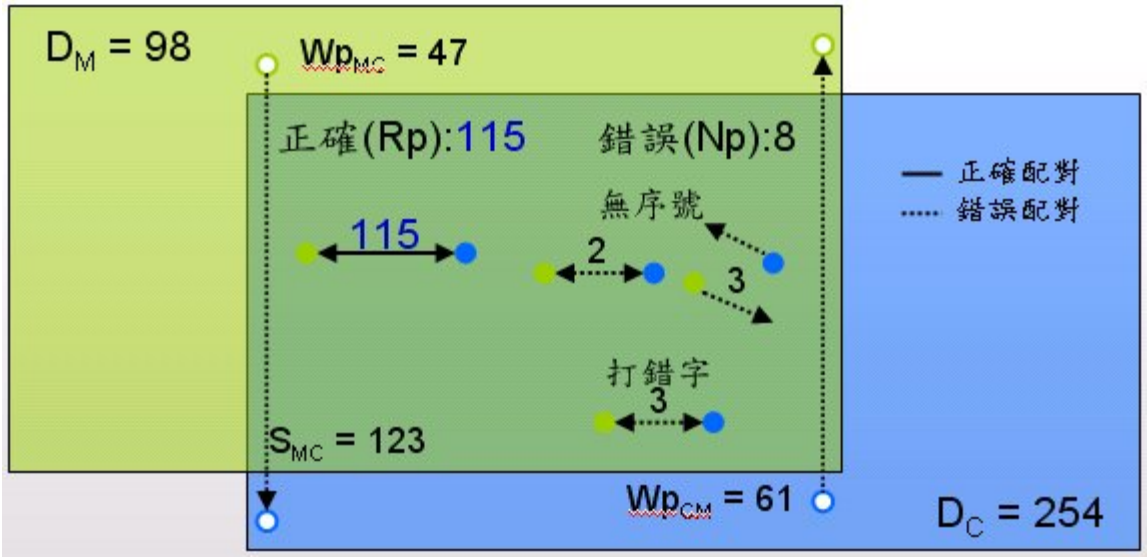
表格 5: 標準定義-名人 3C 與 C NET

名詞	定義
S_{MC}	以人工比對，名人 3C (Mren)與 CNET 網站間相同產品的數量
T_M	名人 3C (Mren)在筆記型電腦類別所有產品的數量
T_C	CNET 科技資訊網在筆記型電腦類別所有產品的數量
D_M	名人 3C (Mren) 特有產品的數量
D_C	CNET 科技資訊網特有產品的數量



圖表 18: 人工比對名人 3C 與 CNET 兩個網站的產品

接著以 LMCS 權重計算加上比對策略，比對名人與 C NET 兩個網站所有的筆記型電腦產品。所得到的結果如圖表 19 所示，有 115 項是正確判斷為相同產品。原本以人工判斷相同的產品數為 123，正確判斷出 115 項，達到 93%。在 8 項有相同產品卻判斷錯誤的部分，有 3 項產品是因為打錯字，導致 LMCS 計算分數較低而比對錯誤。有 5 項產品是因為其中一個網站是以產品序號代替產品介紹，而另一個網站卻無產品序號，而以規格描述產品，造成無法以字串比對正確判斷是否為相同產品。因此可以歸納出：產品資料的編輯是否正確，以及編輯的模式是否相同，會對比對的結果產生關鍵性的影響。



圖表 19: 名人與 C NET 網站的比對結果

4.2 實驗結果評估

前面的實驗都著重在是否有正確判斷出相同產品，現在對程式所跑出來的結果進行評估。包括：回收率、精確度、相似度的計算。

回收率 $$R = \frac{Rp}{Rp + Np} \quad \text{.....} \quad (5)$$

精確度 $$P = \frac{Rp}{Rp + Wp} \quad \text{.....} \quad (6)$$

相似度 $$sim = \frac{2PR}{P + R} \quad \text{.....} \quad (7)$$

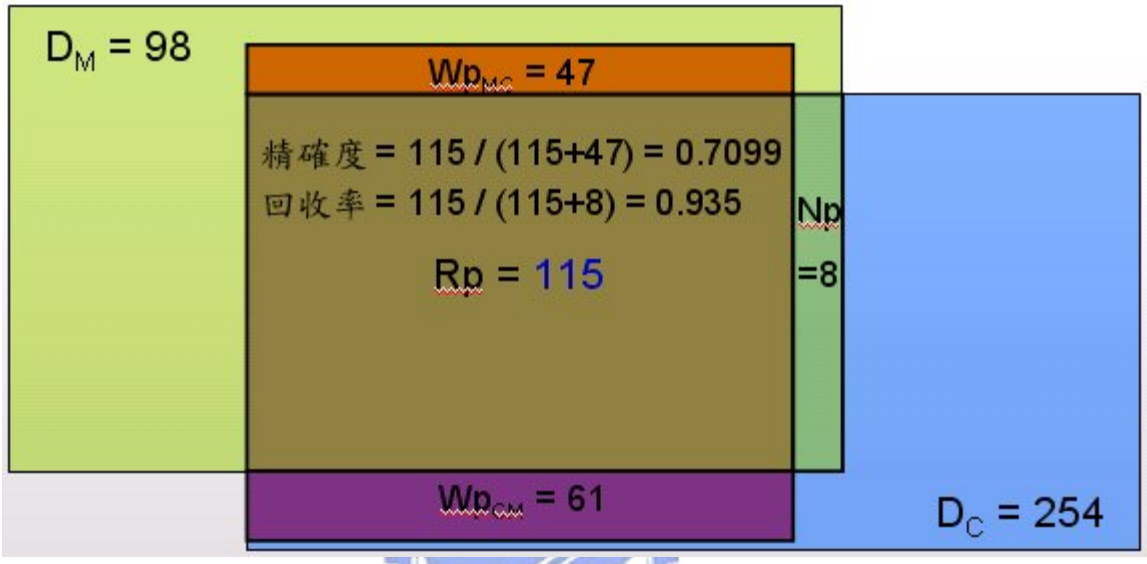
Rp: 程式正確判斷為相同產品的數量
Wp: 程式錯誤判斷為相同產品的數量
Np: 程式沒有判斷出來的相同產品數量

將名人和 C NET 網站產品的比對結果，以圖表 20 表示。程式正確判斷為相同產品的數量(Rp)為 115，程式沒有判斷出來的相同產品數量(Np)為 8，代入公式(5)，則回收率 R = 115 / (115 + 8) = 0.935。

其中兩個網站的程式錯誤判斷為相同產品的數量(Wp)並不相等，因為不同的產品只要其 LMCS 計算的分數相等，就同時判斷為另一個網站的相

同產品。程式錯誤判斷為相同產品的名人網站產品數量 WpMC 為 47，錯誤判斷為相同產品的 C NET 網站產品數量 WpCM 為 61。程式正確判斷為相同產品的數量(Rp)為 115，代入公式(6)，則精確度= 115 / (115 + 47) = 0.7099。

再將回收率 0.935、精確度 0.7099 代入公式(7)，得到名人 3C 比對 C NET 產品的相似度(M,C) = 0.806。



圖表 20: 回收率=0.935, 精確度=0.7099

4.2.1. 調整權重與最高相似度

我們利用修改公式的 W[n]值，分別設定：W[n] = n, $\frac{n\times(n+1)}{2}$, $\frac{3}{4}\times n^2$, n^2, $\frac{1}{4}\times n^3$, $\frac{1}{2}\times n^3$, $\frac{3}{4}\times n^3$, n^3, $\frac{1}{4}\times n^4$, $\frac{1}{2}\times n^4$, $\frac{3}{4}\times n^4$, n^4。

分別將其一一代入公式(2)，並經由程式計算結果，最後得到的最高的相似度如表格 6 所示：

W[n] = n 時，最佳相似度為 80.58608(%)，其精確度為 73.33333(%)，回收率為 89.43089(%)。

W[n] = n^2時，最佳相似度為 85.14056(%)，其精確度為 84.12698(%)，回收率為 86.17886(%)。

W[n] = n^3時，最佳相似度為 85.82996(%)，其精確度為 85.48387(%)，回收率為 86.17886(%)。

W[n] = n^4時，最佳相似度為 86.29032(%)，其精確度為 85.6(%)，回收率為 86.99187(%)。

權重調整與最高相似度的曲線圖變化，如圖 21 所示。調整權重得到最顯著的改善結果，是在將W[n] = n調整為n^2時。其所代表的意義是：相似字串的長度越長，給予越高的分數是有助於提高配對的精確度與回收率的。

在以上的實驗中，初始時回收率達到 90(%)，但是精確度卻都只有 70(%)。經過過濾分數較低的配對結果，可以觀察到規則：當兩個產品相似的字串越長，給予越高的分數，會有助於過濾出最高的相似度。以權重為W[n] = n^3時為例，保留所有配對結果的 77.9874214 (%)，可以得到最高相似度。權重為W[n] = n^4時，保留所有配對結果的 75.7575758 (%)，可以得到最高相似度。

權重為W[n] = n^2 與n^3時的差別，在於n^3的錯誤配對少了兩項。分別為：

NC4010 系列/PV177PA#AB0-BAG	HP Compaq nc4010 (PR610PA#AB0)

與

NC6000 系列/PQ245PA#AB0	HP Compaq nc6000 (PK351PA#AB0)

此兩項產品以實際人工配對是無相同產品，只是同系列產品。

權重為W[n] = n^3 與n^4時的差別，在於n^4的正確配對多了一項。

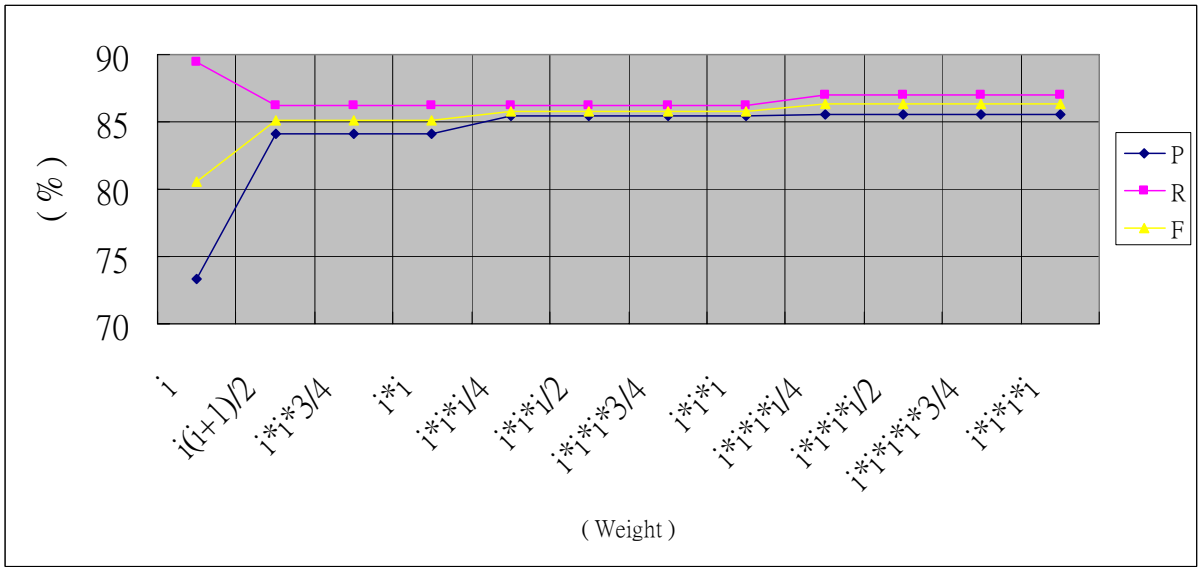
LifeBook L-2010	Fujitsu LifeBook L2010

其主要原因在於 L2010 與 L-2010 的差異。

表格 6: 權重調整與最高相似度-名人 3C 與 C NET

權重	精確度(%)	回收率	相似度	正確	錯誤
W[n] = n	73.33333	89.43089	80.58608	110(123)	40
$W[n]=\frac{n\times(n+1)}{2}$	84.12698	86.17886	85.14056	106(123)	20
$W[n]=\frac{3}{4}\times n^2$	84.12698	86.17886	85.14056	106(123)	20
W[n] = n^2	84.12698	86.17886	85.14056	106(123)	20
$W[n]=\frac{1}{4}\times n^3$	85.48387	86.17886	85.82996	106(123)	18

$W[n]=\frac{1}{2}\times n^3$	85.48387	86.17886	85.82996	106(123)	18
$W[n]=\frac{3}{4}\times n^3$	85.48387	86.17886	85.82996	106(123)	18
$W[n]=n^3$	85.48387	86.17886	85.82996	106(123)	18
$W[n]=\frac{1}{4}\times n^4$	85.6	86.99187	86.29032	107(123)	18
$W[n]=\frac{1}{2}\times n^4$	85.6	86.99187	86.29032	107(123)	18
$W[n]=\frac{3}{4}\times n^4$	85.6	86.99187	86.29032	107(123)	18
$W[n]=n^4$	85.6	86.99187	86.29032	107(123)	18



圖表 21: 權重調整與最高相似度曲線圖

4.2.2. 無法正確配對的原因

我們列出所有程式無法判斷的產品字串，並將其錯誤原因歸類，如表格 7 所示。序號與規格是表示：在兩個網站中，其中一個網站是以產品序號為主，沒有多餘的規格敘述。另一個網站則是沒有詳細的序號，而是以規格描述代替。打錯字類型，也就是因為編輯人緣的輸入錯誤所導致的程式配對錯誤。系列名稱較長類型，則是因為系列名稱的長度很長，導致給予權重加分過多而影響到配對的判斷，導致錯誤。

表格 7: 錯誤配對的原因歸類-名人 3C 與 C NET

錯誤類型	名人產品	C NET 產品
序號與規格	nc6230 系列 /DX607AV#AB0-004	HP Compaq nc6230
	nw8000 系列/PR629PA #AB0-PACK (By-Oder 3days)	HP Compaq nw8240
	NX6120 系列 /PQ999PA#AB0-GIFT	HP Compaq nx6120
	N203-B1C73H1/黑	Gigabyte N203 爵士黑
	NB-N512(銀) -N512B1A42H1	Gigabyte N512 715
打錯字	Satellite M30t 列 /PSM30L-71037	Toshiba Satellite M30 PSM30L-71037
	Satellite M30 系列 /PSM30L-71037M/ COMBO	Toshiba Satellite M30 PSM30L-7103M
系列名稱較長	TravelMate 4000 系列/ TM 4101LCi_U	Acer TravelMate 4101LCi-U

4.2.3. 產品比對驗證數據-名人 3C 與順發 3C

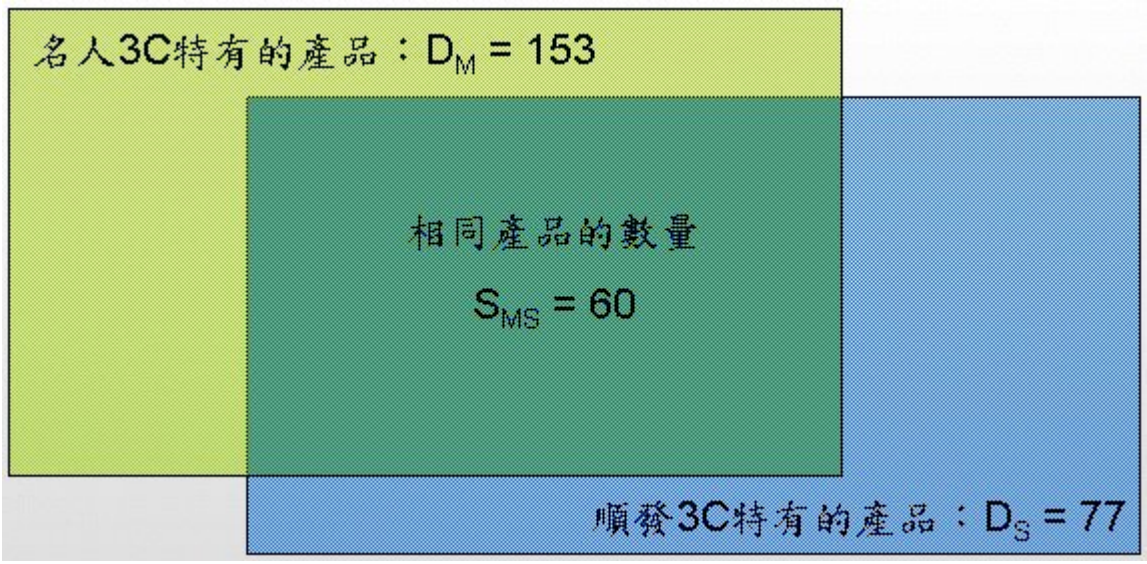
我們再以名人 3C 和順發 3C 網站的配對結果做驗證。

首先我們以人工比對的方式，分別計算出名人 3C以及順發 3C 兩個網站在筆記型電腦這個產品類別的產品總數量，以及兩個網站之間相同與不相同產品的數量。再訂定標準定義，包括：S_{MS}表示名人 3C (Mren)與順發 3C (Sunfar)兩個網站間相同產品的數量、T_M表示名人 3C所有產品的數量、T_S表示順發 3C 所有產品的數量、D_M表示名人 3C特有產品的數量、D_S表示順發 3C 特有產品的數量，如表格 8 所示。然後將這兩個網站的相同產品與不同產品的分佈以圖表 22 表示。

表格 8: 標準定義-名人 3C 與順發 3C

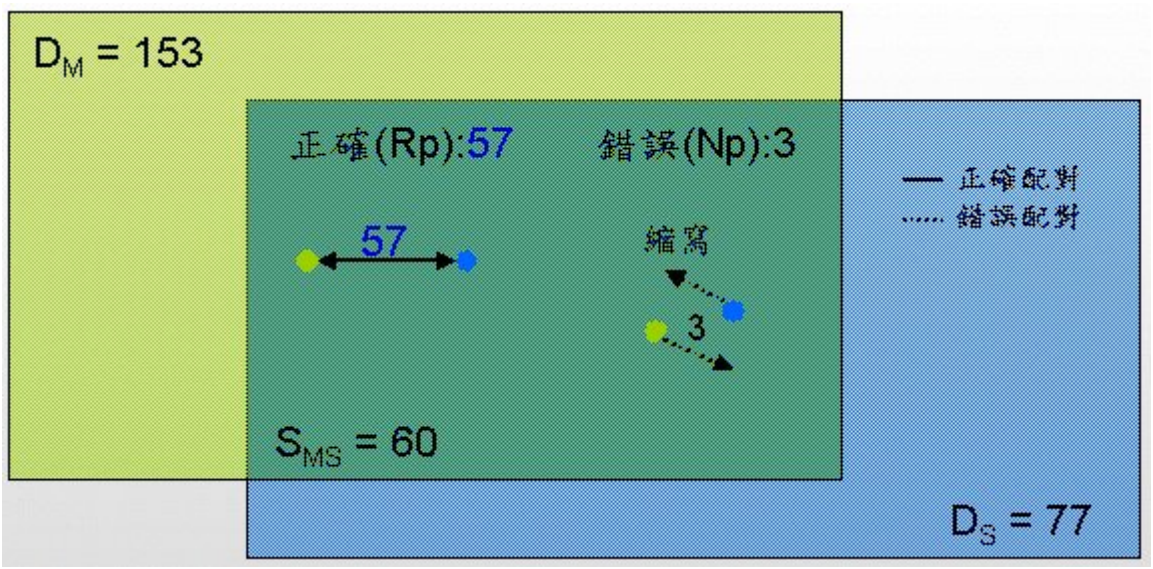
名詞	定義
S_{MS}	以人工比對，名人 3C (Mren)與順發 3C (Sunfar)間相同產品的數量
T_M	名人 3C (Mren)在筆記型電腦類別所有產品的數量

T_S	順發 3C (Sunfar)在筆記型電腦類別所有產品的數量
D_M	名人 3C (Mren) 特有產品的數量
D_S	順發 3C (Sunfar)特有產品的數量



圖表 22: 人工比對名人 3C 與 CNET 兩個網站的產品

接著以 LMCS 權重計算加上比對策略，比對名人與順發兩個網站所有的筆記型電腦產品。所得到的結果如圖表 23 所示，有 57 項是正確判斷為相同產品。原本以人工判斷相同的產品數為 60，正確判斷出 57 項，回收率達到 95%。在 3 項有相同產品卻判斷錯誤的部分，都是因為縮寫，導致 LMCS 計算分數較低而比對錯誤。



圖表 23:名人與順發網站的比對結果

其最佳回收率為 95% (57/60)，調整權重得到回收率降為 83%時，最高相似度為 80%。調整權重與最高相似度的結果如表格 9 所示。而不論權重如何調高都無法再增加最高相似度，顯見在配對的過程，兩個網站所販售的相同產品數量多寡，亦即其產品資訊相同的項目多寡，會直接影響到程式配對的結果。

表格 9: 權重調整與最高相似度-名人 3C 與順發 3C

權重	精確度(%)	回收率	相似度	正確	錯誤
$W[n] = n^3$	76.92	83.33	0.8	50(60)	15
$W[n] = n^4$	76.92	83.33	0.8	50(60)	15

錯誤配對的原因是因為使用產品系列名稱的縮寫，如表格 10 所示。共有三項產品錯誤配對，其中一項是只使用產品系列描述產品。另外兩項是因為將 Presario 系列名稱縮寫為 Pr，導致 LMCS 分數降低而配對錯誤。

表格 10: 錯誤配對的原因歸類-名人 3C 與順發 3C

錯誤類型	名人產品	順發產品
縮寫	N203-B1C42H1/紅	N203(紅)
	Presario B3800 系列 /B3801AP (PN761PA)	Pr B3801
	Presario M2200 系列 /PM2211AP (EE470PA#AB0)	Compaq PrM2211AP

第五章 結論與未來工作

5.1 討論

根據實驗的結果，可以歸納出產生錯誤判斷的幾個因素：

1. 資料編輯錯誤。

   資訊編輯人員的專業知識也很重要。如果專業知識不足，有些專業術語可能會看不懂、容易看錯，會產生編輯時的錯誤。也有可能是因一時疏忽打錯字。因為我們採取的比對方法是字串比對，當然必須要有相同的字串才會有較高的計分，也才能依據分數高低來判斷產品間相似度的高低。如果是打錯字或資料編譯錯誤，比對的計分都可能會變低，很可能因此就判斷錯誤。就像表格 6 的錯誤原因：打錯字。

2. 使用縮寫或產品代稱。

   很多廠商都會為特定產品系列命名，例如：IBM 的 ThinkPad 筆記型電腦系列，BenQ 的 JoyBook 筆記型電腦系列...等等。有的網站會將 ThinkPad 縮寫為 TP，將 JoyBook 縮寫為 JB。在同一網站內，有的用 JB，有的用 JoyBook 或是有些產品有註明 JoyBook 系列，有些沒有。以我們的比對方法計算，相同字串越長的產品，分數會越高，也就容易產生錯誤判斷。

3. 各網站的產品資訊編輯的模式不相同。

   有的以產品序號代替產品規格的介紹，有的卻是沒有產品序號而是以產品的簡要規格描述產品。也有的比較詳細描述產品，產品序號、規格都有。就像表格 6 的錯誤原因：無序號。如果產品的資料越詳細，編輯的方式、項目都相同，越能夠正確判斷出相同產品。相反地，如果不同網站的產品敘述，沒有相同的字串，就不適合使用 LMCS 方法了。

5.2 未來工作

本論文的實驗網站數量還不夠多，而且產品類別也不多。所以還不能夠代表所有產品類別的比對結果。

由於目前止考慮單一類別的產品，要如何讓產品比對後的結果應用到自動歸類，將是下一個目標。將來若能結合產品的特徵影像會有助於產品

的歸類，將有助於驗證產品比對後的判斷結果是否正確。

針對產品的代稱與縮寫，可以應用建立關鍵詞庫，將所有廠商的產品系列建立一關連表格，並設訂各關鍵詞之間的權重高低以表現其關連性的強弱。也可以針對關鍵詞設計一容錯機制，打錯字在嚴重性多低的範圍內都認定為同一關鍵詞。另外，如果能偵測產品資料的順序顛倒，更能增加產品比對的正確判斷率。

參考文獻

[1] 名人 3C，http://www.mren.com.tw
[2] C NET 科技資訊網，http://taiwan.cnet.com
[3] 順發 3C，http://www.sunfar.com.tw
[4] 僑品電腦，http://www.jpcl.com.tw
[5] 燦坤 3C，http://www.tkec.com.tw
[6] 楊允言，“文件自動分類及其相似性排序”，國立清華大學，資訊科學學系，碩士論文，民國 81 年
[7] 王常威，“以內容為基礎之 XML 文件分類方法之研究”，國立成功大學，資訊管理研究所，碩士論文，民國 92 年
[8] 陳秀娟，“利用數值索引作為音樂資料庫擷取之研究”，朝陽科技大學，資訊管理系碩士班，碩士論文，民國 90 年
[9] 李宜揚，“以聲音內容為主的音樂資料庫查詢系統”，國立清華大學資訊工程學系，碩士論文，1999
[10] D. S. I Iirschbcrg (1977) “Algorithms for the longest common subsequence problem, J.ACM 24(4) pp.664-675.
[11] L. Bergroth, H. Hakonen, T. Raita, “A survey of longest common subsequence algorithms”, SPIRE, pp.39-48, 2000.
[12] I-Chen Wu, Jui-Yuan Su, and Loon-Been Chen, "A Web Data Extraction Description Language and Its Implementation", The 29th Annual International Computer Software and Application Conference (COMPSAC 2005), Edinburgh, Scotland, July 2005.
[13] I-Chen Wu, Jui-Yuan Su, and Loon-Been Chen, "On the Web Data Extraction Model", The 17th International Conference on Software Engineering and Knowledge Engineering, Taipei, Taiwan, July 2005.
[14] D. Sankoff, and J. B. Kruskal (eds.). Time Warps, String Edits, and Macromolecules, The Theory and Practice of Sequence Comparison. Addison-Wesley, 1983.
[15] M. S. Waterman, “General methods of sequence comparison,” Bulletin of Mathematical Biology, vol. 46, no. 4, pp. 473-500, 1984.
[16] E. A. Cheever, D. B. Searls, W. Karunaratne, and G. C. Overton, “Using signal processing techniques for DNA sequence comparison,” Bioengineering Conference, 1989. Proceedings of the 1989 Fifteenth Annual Northeast, pp. 173-174, 1989.
[17] A.S. Deshpande, D.S. Richards, and W.R. Pearson. ``A Platform for Biological Sequence Comparison on Parallel Computers'. Computer Applications in the Biosciences, vol.7, pp.237-247,1991.
[18] Hide, W., Burke, J., and Davison, D. (1994). Biological evaluation

of , an algorithm for high performance sequence comparison. Journal of Computa- tional Biology 1, 199-215.
[19] Setubal, J. and J. Meidanis, Sequence comparison and database search. In Introduction To Computational Molecular Biology. 1997. 47-103.