應用於電子商場營業類別查詢之語音辨識系統

陳建宏、黃英峰、陳榮貴

中華電信研究所 應用科技研究室
桃園縣楊梅鎮 326 民族路 5 段 551 巷 12 號（應科 5857）
Tel: (03)4245857 Fax:(03)4244167
Email: {cch, engfong, jkchen}@ms.chttl.com.tw

摘 要

本文針對網際網路(World Wide Web)上的資料庫查詢，提供除了鍵盤輸入外，另一個更便捷迅速的語音查詢方式。在現階段, 本系統主要是應用在電子商場營業類別查詢，但也適合其他類似的網際網路上之資料庫查詢。本系統有兩大優點：首先，我們提出一個架構在網際網路上之主從式(Client-Server)語音辨認系統架構，將大量的資料及語音辨認運算放在伺服器端，解決在瀏覽器端進行語音辨認時需要大量資料下載及計算能力不足的問題。 再者，我們也提出一個快速部分匹配(Fast Partial Matching, FPM) 語音辨認演算法，此演算法適合對已經過整理的資料庫，以標題辨認的方式查詢，且容許不完整和含有多餘內容的語音輸入，增加使用者的方便性。本系統對上網查詢生活資訊，提供了一個更方便迅速的查詢方式。

關鍵詞：語音辨認(Speech Recognition) 部分匹配(Partial Matching)
主從式(Client-Server)

1. 前言

目前網際網路(internet)蓬勃發展，特別是 WWW(World Wide Web)不但應用在網際網路，連公司內部的企業內網路(intranet)也都使用瀏覽器(browser)的方式來操作。我們相信未來個人電腦會成為每個家庭的必備家電用品，而音效卡和麥克風會成為個人電腦的必備週邊設備；同時，上網查詢生活資訊將是日常生活中常會去做的事。網路上很多應用都會使用到資料庫的查詢, 當可查詢的資料項目數目多到網頁無法列表顯示出來時，一般使用者便須敲入所要查詢的字串，由文字檢索系統去查出最可能的幾個答案，再由使用者挑選正確的答案。如果網站的資料庫能提供語音查詢功能，將使得上網查詢變得更輕鬆，甚至不用學電腦（中文輸入）也可以查。

一般的檢索方式主要有兩類，一類是所謂的全文檢索，例如 GAIS 的 WWW 檢索系統[1]，這樣的系統是在文章內容中搜尋，使用者輸入的字串基本上沒有什麼限制，若以語音來輸入，必須用聽寫機的方式來輸入任意字串，此種方式雖然應用的範圍較廣，但語音辨認的效果較難得到令人滿意的結果；另一類則是針對標題進行搜尋，例如 HiNet 上的智慧型電子商場(IEM), 也就是 HiGo 所提供的商品查詢[2]，即以營業項目或廠商名稱為搜尋對象，以查得相關廠商的超連結。此類應用通常使用在已經過整理的資料庫上的查詢，因為詞彙量有限，甚至可以用詞彙辨認的方式來作語音辨認，以目前的技術，此類的語音辨認的效果已可達實用的階段。

本文的重點是介紹我們如何在 WWW 的環境上建立一套實用的語音查詢系統，現階段將查詢的範圍設定為經過整理的資料庫的標題，例如人事資料庫的人名；問題題庫的題目等。主要內容包括：描述如何以主從式的方式在 WWW 上架構這樣的語音查詢系統；同時考慮到使用者語音輸入時的方便性，這裡我們也探討容許不完整語音輸入的辨認技術。本系統可應用的範圍包括在企業內網路(intranet)上的人事資料查詢，分機號碼查詢；或網路上的股票查詢，天氣查詢，颱風天是否上班的查詢，以及時刻表查詢等等，舉凡此類資料庫的查詢皆是。本文以 HiGo 的商品查詢為我們的實際應用系統，瀏覽器則使用 PC 及 Netscape。

2. 系統描述

一般而言此類系統所要查詢的內容，並不在網頁的內容中，且通常是動態的，隨時在改變，並且其數量也不小。若要下傳到瀏覽器這端來進行辨認，要花不少時間，不是很實際的做法。再考慮到瀏覽器端的 PC，可能不是每部機器的效能都可滿足語音辨認的需求。因此我們決定將辨認器分為兩部分，前端處理放在瀏覽器端，辨認器的主要部分則放在伺服器端，也就是在網路上以主從式的方式來建立我們的辨認系統。使資料庫及主要運算都在伺服器上，故可同時解決大量資料下傳及運算能力不足的問題。

下面幾節是系統方塊圖以及語音辨認技術的介紹：

2.1. 系統方塊圖

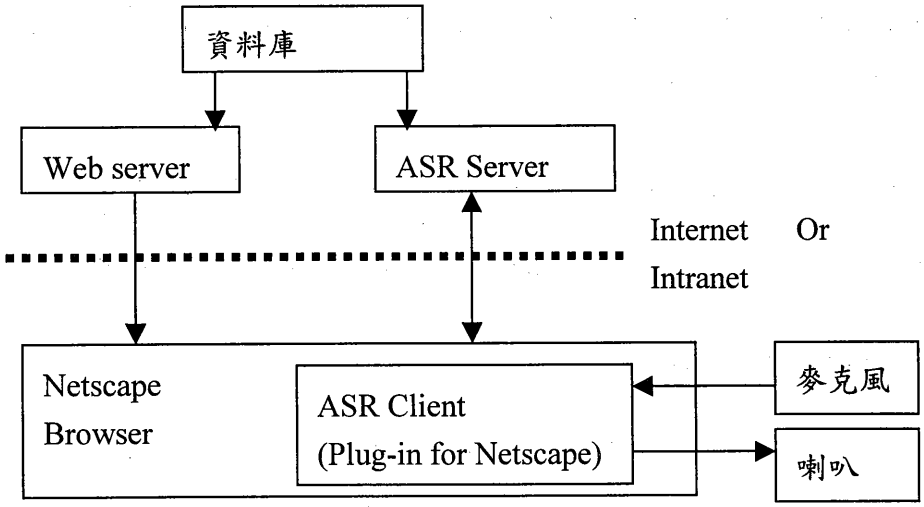


圖 一.不含防火牆的系統網路架構圖

圖一是此系統的方塊圖，運作方式描述如下:首先 Netscape 瀏覽器連上 Web Server，下載我們的首頁，此首頁包含 ASR(Automatic Speech Recognition) Server 的網路位置，及各項設定值。瀏覽器根據此首頁啟動先前已經安裝好的 Plug-in(ASR Client)，ASR Client 提示使用者輸入語音，經錄音並初步求取特徵值，ASR Client 再經由網路將特徵值傳送至語音辨認伺服器(ASR Server)，語音辨認伺服器根據傳來的資料選擇適當的辨認範圍，進行語音辨認，辨認結果連同超連結資料再送回 ASR Client，使用者可在 ASR Client 所顯示的畫面上得到所要的查尋的結果，或經使用者確認後點選連結到相對應的網頁去，進而從此網頁得到結果，或繼續查詢直到所要的結果得到為止。

至於與瀏覽器結合的部分，本階段我們選擇目前相當普遍的組合—Windows 95(或 NT) 搭配 Netscape。因為必須使用到低階的錄放音功能，我們選擇以 Plug-in 的方式來撰寫 ASR Client 端。此 Plug-in 基本上是一個 C 語言所寫成的動態連結程式庫(DLL)，只能在 X86 的 PC 上執行，所以目前我們的系統也只能在 X86 的 PC 上執行。但低階的錄放音功能是用 Windows 提供的低階函式介面撰寫，故任何在 Windows 上裝妥的音效卡皆可正常使用。

2.2. 主從式的網路架構

視 ASR Client 和 ASR Server 之間是否有防火牆相隔，系統的架構也相對的有所不同。茲分成企業內網路(intranet) 和網際網路(internet)來討論:

2.2.1.企業內網路

ASR Client 和 ASR Server 在網路上，以 TCP/IP 的協定，用 TCP mode 的 SOCKET 互相連接。在企業內網路的應用下，通常有快速的網路傳輸速度，而且 ASR Client 可以直接連接到 Server，中間不需透過防火牆，如圖一所示。為了獲得最佳的使用效果，也

就是最快的反應速度，我們訂定了 ASR Client 和 ASR Server 間的協定，使他們可用交談式的方式來傳遞資料。當 ASR Client 端一邊在錄音、求取語音特徵參數時，即可一邊將語音特徵參數往 ASR Sever 送。等錄音完成，ASR Client 於很短的延遲時間後即可收到 ASR Server 送回的辨認結果。

2.2.2.網際網路

在網際網路的應用下，必須考慮到一般公司均會加裝防火牆。對於 WWW 而言通常就是裝一個 Proxy，使所有的瀏覽器要下載公司外的首頁資料都必須透過 Proxy。如此我們在公司內的 ASR Client 便無法直接連接到在公司外的 ASR Server，那麼如何讓 ASR Client 可以將特徵值送到 ASR Server，並取回辨認結果呢？一個解決的方法是依據 HTTP(Hypertext Transfer Protocol)協定[3]，使用其 POST 的方式。依此法 ASR Client 連接上 Proxy 並將要傳送給 ASR Server 的資料一次全部傳送給 Proxy，Proxy 便會將資料轉送給 ASR Server；且將 ASR Server 送回的結果轉送給 ASR Client。這個方法有一個限制是：資料必須一次全部送出去，才等送回的結果。因此 ASR Client 的做法改為先完成錄音並求得全部特徵值，才將特徵值 POST 給 ASR Server，然後等待辨認結果回來。同時，ASR Server 端也必須配合。為了使原來的 ASR Server 能夠同時應付企業內網路和網際網路的需要，ASR Server 保持不變，另外多執行了一個模擬的 CGI(Common Gateway Interface)，簡稱 SCGI。SCGI 模擬 HTTP Server 上的 CGI 接收 ASR Client POST 過來的資料，再以交談式的方式和 ASR Server 連接，並傳遞資料給 ASR Server 及取得辨認結果。然後 SCGI 再以 POST 的方式把辨認結果傳回給 Client。系統方塊圖如圖二所示：

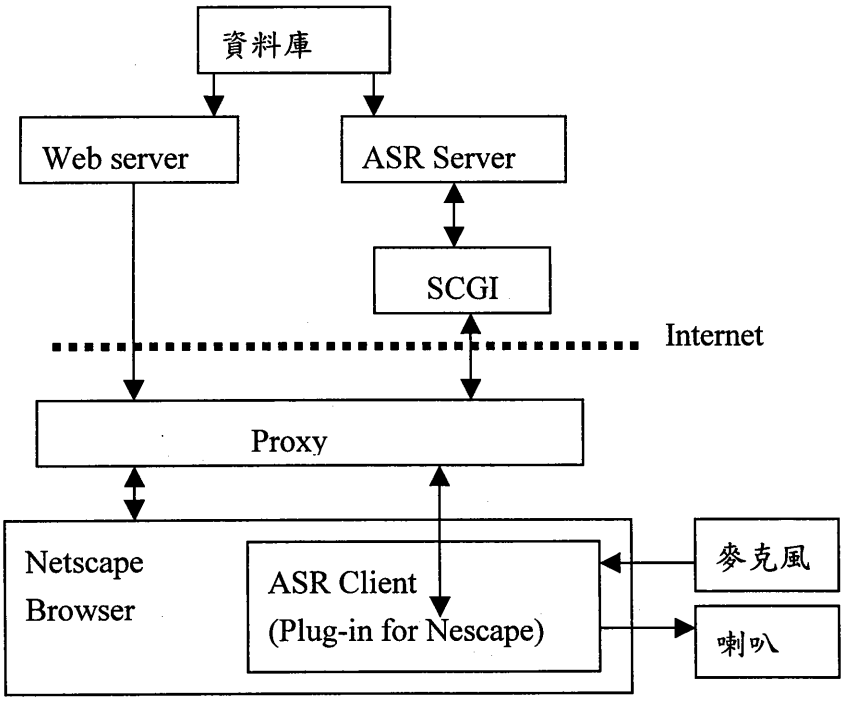


圖 二. 含有防火牆的系統網路架構圖

2.3. 語音辨認

在語音辨認的部分，雖然是標題式的辨認，但使用者往往不知道確切的文字內容，在此情況下，要正確無誤念出整個標題，實屬不可能，最常碰到的問題少掉一些或有多餘的內容，或兩者兼有，例如：

標題：　　電視電腦遊樂器
語音輸入：電視遊樂器

或

標題：　　電腦組件
語音輸入：電腦週邊設備

為了讓系統更好用，必須配合使用者的習慣，給使用者最少的限制同時又能得到良好的正確率。為此，我們採用了「部分匹配(Partial Matching ,PM)」的觀念，以解決上述問題。

本系統查詢的標題大約有 3000 個, 字數從 1 到 11 個字都有。下面幾節我們首先介紹本系統所使用的語音特徵向量及語音模型；然後在語音辨認方法方面, 分別介紹部分匹配(Partial Matching, PM)演算法以及快速部分匹配(Fast Partial Matching, FPM)演算法。

2.3.1.語音特徵參數及語音模型之介紹

本系統的語音是麥克風輸入，取樣頻率為 16KHz。語音特徵參數向量由 25 個係數組成，分為 12 個 Mel-scale 反頻譜係數(cepstrum)、12 個一階微分的反頻譜係數、一個一階微分能量值。

語音模型採用目前自動語音辨認系統(Automatic Speech Recognition, ASR)最流行之隱藏式馬可夫模型(Hidden Markov Model, HMM)[L.R. Rabiner, 1989]，每個狀態中之機率函數採用 Gaussian mixture density， 混合(mixture) 的數目為 4。每一個國語音節包含子音部分及母音部分，子音部分的狀態數(state)為 3，母音部分的狀態數(state)為 5。

2.3.2.部分匹配(Partial Matching, PM)演算法

本法與一般的詞彙辨認類似，但為了克服多字或少字的困難，須對每一標題構建新的 HMM 網路，然後將輸入語音與每一個標題的 HMM 網路做最佳化匹配，求出屬於該標題的分數來，以分數最高的幾個標題當做候選標題，再由使用者去挑選。接下來是我們第一種方法的介紹：

1. 對每一個標題找出相對應的音節模型($S_1S_2...Sn$)，配合 Garbage Model 組合出如下的 HMM 網路:

圖三. 有順序限制的標題HMM網路圖

這個網路使標題中的任一音節都可被跳過，而且在任何音節之間都可加入 Garbage model，但被跳過的音節都必須按其在標題中的順序出現。

2. 對上述 HMM 網路用輸入語音資料去比對出維特比路徑(Viterbi Path)，得到對數相似度(log likelihood)來求該標題的分數(score)。對於一個標題的分數求法如下：

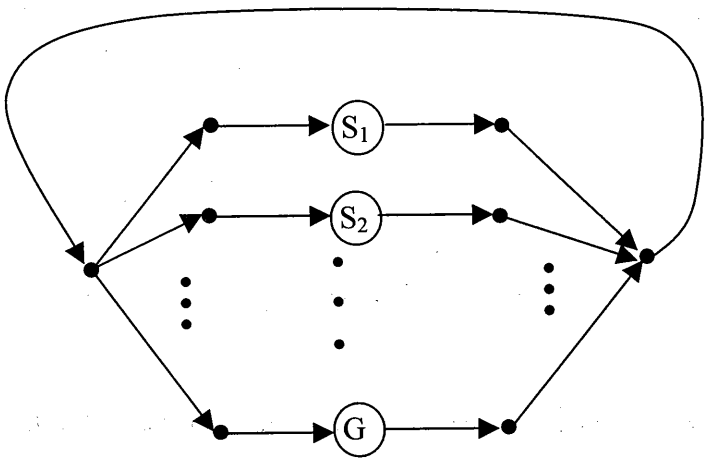
$$\text{Score} = \sum_{t}^{T} \log(f(O_t \mid \lambda_{St}))$$

其中 o_t 是一個音框的特徵向量，St 是指最佳路徑中 t 音框所對應的狀態(state)，λ_{St} 則是該狀態的模型參數。以相同做法為每一標題求得分數。

3. 找出分數最高的前幾名來。

在我們的系統中 Garbage Model 的選擇，使輸入語音匹配到正確音節的部分對該正確音節模型所得到的對數相似度(log likelihood)，會高於對 Garbage Model 所得到的對數相似度(log likelihood)高。故匹配到越多音節的標題，整體分數也越高。

選用圖三的 HMM 網路結構的原因是基於我們認為:一個有意義的短詞裡的音節組合，一般在念的時候也會按其固定順序。這個網路有這個順序限制，對有意義的短詞有較好的辨認效果。但對於標題中有兩個以上的短詞，使用者念的時候又沒有一定順序時，將造成只有其中一個短詞被匹配到。為了改善這個缺點，可將 HMM 網路改成如圖四的結構：



圖四. 無順序限制的標題 HMM 網路圖

圖四的結構使不按順序的輸入也都可被匹配到。另外為了減少因為網路限制太鬆，使不正確的標題也容易匹配到高的分數的問題，可在最佳路徑的音節連接到另一音節的地方，根據此兩音節在該標題中是否前後相鄰來給於不同的加分。假設最佳路徑中有 M 個音節($S_1S_2...S_M$)，此標題的整體分數計算方式如下：

$$\text{Score} = \sum_{t}^{T} \log(f(O_t \mid \lambda_{St})) + \sum_{l=1}^{M-1} Q(S_l, S_{l+1})$$

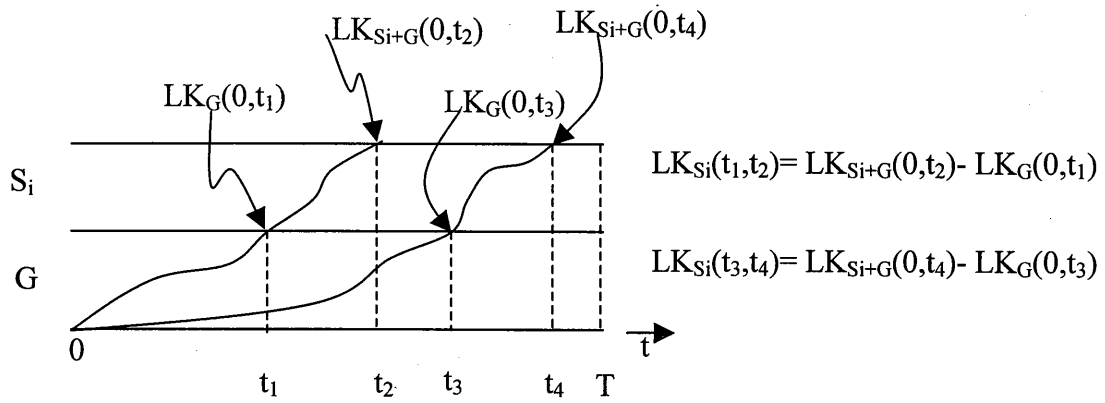
$$Q(S_l, S_{l+1}) = \begin{cases} q_0, \text{當} S_l \text{和} S_{l+1} \text{在標題中是前後相連} \\ q_1, \text{當} S_l \text{和} S_{l+1} \text{在標題中不是前後相連} \end{cases} \quad \text{且 } q_0 > q_1$$

2.3.3.快速部分匹配(Fast Partial Matching, FPM)演算法

在 2.3.2.節中所提的 PM 方法因為要對每一標題去計算分數，當所要查詢的標題數目增大時，計算時間會成比例增加。當數目多到一個程度時，系統的反應速度便不符合實際使用需求。在這節中我們提出一個快速部分匹配演算法，以預處理的方式，避免一個一個比對。因此，大幅提昇比對速度的目的。此演算法分四個步驟：步驟 1. 從輸入語音找出 N 個最可能的音節區段(Syllable islands); 步驟 2. 根據這些音節以快速查詞的方法，找出有這些音節出現的 L 個標題來；步驟 3. 用步驟 1 所得到的 likelihood 來對在步驟 2 中找出來的標題進行快速評比(fast ranking)，將 L 個標題再縮小範圍到 K 個標題；(4). 若有需要可再對這 K 個標題進行仔細評比(detail ranking)。 4 步驟分述如下：

步驟 1. 尋找音節區段**(Syllable islands)**

在這裡要把輸入語音對國語 414 個音節作比對(warping)，每一個音節都要找出 n 個音節區段(Syllable islands)。這些音節區段彼此重疊的音框數不可超過設定值，比如說須少於音節區段音框數的四分之一。通常一個 標題中出現的相同音節數目不會多於 3 個，因此 n 等於 3 到 5 之間即可。如圖五. 所示，以 level-building 的方式將 Garbage model(G)和音節模型(S_i)疊起來，進行維特比(viterbi search)比對，計算出每一個時間點的累積對數相似度(log likelihood)值，並記錄每條路徑從 Garbage model(G)跳到音節模型(S_i)的時間點：



圖五. Garbage model(G)和音節模型(S_i)的 level-building 之比對路徑圖

圖中 $LK_{Si+G}(0,t_2)$表示包含 Garbage model 及 syllable model，從 0 比對到 t_2 所得到的累積對數相似度(log likelihood)；$LK_{Si}(t_1,t_2)$ 表示單單 syllable model，從 t_1 比對到 t_2 所得到的累積對數相似度(log likelihood)；$LK_G(t_0,t_2)$則表示單單比對 Garbage model 得到的累積對數相似度(log likelihood)。音節區段的資訊包括起始時間(t_1)、終止時間(t_2)以及該區段對音框數做過正規化的 log likelihood rate($LR_{Si}(t_1, t_2)$)。參考[Seiichi Nakagawa,1997]，$LR_{Si}(t_1, t_2)$的求法如下：

$$LR_{Si}(t1,t2)=\frac{LK_{Si}(t_1,t_2)-LK_G(t_1,t_2)}{t_2-t_1}$$

因為 $LK_{Si}(t_1, t_2)= LK_{Si+G}(0, t_2)- LK_G(0, t_1)$，且 $LK_G(t_1, t_2)= LK_G(0, t_2)- LK_G(0, t_1)$，所以

$$LR_{Si}(t1,t2)=\frac{LK_{Si}(0,t_2)-LK_G(0,t_2)}{t_2-t_1}$$

接著我們要求得 t_2=0 到 T 的所有 LR_{Si}，從中挑出 n 個的步驟如下：

1. 首先挑出具有最大 $LR_{Si}(t_1, t_2)$值的那個音節區段。
2. 從其餘音節區段挑出次大的音節區段，並測試其與其他已挑中的音節區段是否重疊的音框數不超過設定值，若是則選用；否則捨棄。
3. 重複步驟 2 直到選到 n 個音節區段或沒音節區段符合為止。

然後從所有音節，共有 414*n 的音節區段, 中挑出最大的 N 個來，做為下一步驟使用。

步驟 2. 快速查標題

這裡是利用本所研發的快速查詞演算法[Eng-Fong Huang,1994]，來查出包含上一步驟所求出的音節之所有標題來。此演算法僅使用音節編號，並沒有用到任何其他的 likelihood 值，其運算速度相當快，即使輸入的音節區段數目多到 50 個，而待搜尋的標題數目多達數萬的情況下，此部份的搜尋時間也只佔整個語音辨認的極小部分的時間。因此標題數目多寡雖會影響此步驟的時間，但幾乎不影響整體的辨認時間。假設輸入的音節編號有：

ㄉㄧㄢ，ㄋㄠ，ㄑㄧ，ㄘㄞ...

會被搜尋到的標題可能是：

電腦遊戲　　11‥

電腦配件及耗材　　11‥‥1

其中「11‥」和「11‥‥1」代表該標題有音節出現的位置。

步驟 3. 快速評比**(fast ranking)**

根據步驟 2 所找出的標題，我們可在較少量的範圍，對這些標題評分(scoring)，再選出最後可能的候選標題。因為對一個標題而言，出現在其中的音節區段可能有相互重疊的情形發生，必須先進行音節區段的時間匹配，將互有重疊而分數較低的音節區段剔除。最後才根據匹配到的音節區段來計算標題分數。

標題分數的求法，我們實驗過以下三種方法，假設匹配到此標題的音節區段有 P 個：

(1). 音節區段的 LR(likelihood rate)除以其音框數作正規化，然後所有匹配到的音節區段的正規化 LR 再平均：

$$\text{score} = \frac{\sum_{i}^{P} \frac{LK_{Si}(t_1^i, t_2^i) - LK_G(t_1^i, t_2^i)}{t_2^i - t_1^i}}{P} \qquad (1)$$

(2). 音節區段的 LR 除以其音框數作正規化，然後所有匹配到的音節區段的正規化 LR 全部加起來：

$$\text{score} = \sum_{i}^{P} \frac{LK_{Si}(t_1^i, t_2^i) - LK_G(t_1^i, t_2^i)}{t_2^i - t_1^i} \qquad (2)$$

(3). 音節區段的 LR 不作正規化，直接把所有匹配到的音節區段的 LR 全部加起來：

$$\text{score} = \sum_{i}^{P} LK_{Si}(t_1^i, t_2^i) - LK_G(t_1^i, t_2^i) \qquad (3)$$

在我們初步的實驗中，發現第三個方式有最好的辨認結果。我們的推論是：在我們這樣的查詢工作中，有越多音節及音框匹配到的標題，越有可能是所要的標題。從(3)式可看出匹配到的音節數以及每一音節所含的音框數越多，累積的 score 就越高。最後按 score 高低找出前 K 個標題。

步驟 4. 仔細評比(detail ranking)

步驟 3 找出的前 K 個標題，已是最終所要的答案了。但若有需要，可再用前述的 PM 演算法再對這 K 個標題進行仔細評比。

2.3.4.語音辨認方法的結論

快速部分匹配(Fast Partial Matching, FPM)演算法利用多段式的做法將搜尋的範圍逐漸縮小。原本用 PM 需要數分鐘才能算完的計算，用 FPM 卻可即時(real time)完成。FPM 主要計算時間是花在音節區段的產生，這部分的運算時間和標題數目大小無關；而其他步驟所花的時間雖和標題數目大小有關，但因在 FPM 中所佔的時間比例相當小，故整體而言 FPM 的計算速度幾乎不受標題數目多寡所影響。

以實際使用的觀點來看，也就是假設使用者不會故意輸入很多多餘的語音，從初步實驗結果，可看出本系統有不錯的表現。因為本系統尚未完全完成，故本文尚未能提供完整的實驗數據。

3. 結語和展望

本文針對網際網路上的資料庫語音查詢系統，提出一個架構在網際網路(World Wide Web)上之主從式(Client-Server)語音辨認系統架構，解決在瀏覽器端進行語音辨認時需要大量資料下載及計算能力不足的問題。 同時我們也提出一個快速部分匹配(Fast Partial Matching, FPM) 語音辨認演算法，此演算法適合對已經過整理的資料庫，以標題辨認的方式查詢，且容許不完整及含有多餘內容的語音輸入。本系統對上網查詢生活資訊，提供一個更方便迅速的查詢方式。

本系統尚有許多工作未完成，除了將這些工作做得更完整之外，未來我們將朝兩個方向繼續努力:一是提高產生音節區段的準確度；另一個則是在標題的評比(ranking)方面加入詞庫及新詞的知識，以提高整體語音辨認的正確率。

致謝：

在此感謝中華電信研究所王所長、鄭副所長、張光耀主任及劉繼謚總計畫主持人對本項研究的支持與幫助，並感謝同仁在相關問題討論上給予的協助與建議。

參考文獻：

1. http://gais.cs.ccu.edu.tw/cwww2.html 。
2. http://higo.hinet.net/chinese2.htm 網頁中的商品查詢。
3. <draft-ietf-http-v11-spec-07> 可從 http://www.ics.uci.edu/pub/ietf/http/ 取得。
4. Eng-Fong Huang, Chian-Hung Chen and Hsiao-Chuan Wang, "New Search Algorithms for Fast Syllable Hypothesization and Lexical Access in a Large-Vocabulary Mandarin Polysyllable Word Recognizer", Computer Processing of Chinese and Oriental Languages, Vol. 8, No. 2, Dec 1994, pp. 211-225.
5. L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recogniton",Proc.IEEE, 77 (2):257-286,February 1989.
6. Seiichi Nakagawa, Konstantin P. Markov, "Speaker Verification Using Frame and Utterance Level Likelihood Normalization", Proc. ICASSP, Vol.II, pp.1087-1090, 1997.