電話查詢口語對話系統中語音辨識不確定性之處理

Dealing With The Uncertainty Of Speech Recognition For Spoken Telephone-Number Inquiry System

+*王駿發　*王獻章　*劉倚男

{wangjf, wangsj, liuyn }@server2.iie.ncku.edu.tw

*國立成功大學資訊工程學系

+國立成功大學電機工程學系

摘要

在口語對話系統中，語音辨識的正確性是很重要的環節。一般語音辨識系統產生不確定的因素有聲調、代換、少字、多字的問題或者建立不恰當的 Bigram 語言模型產生的錯誤。在本論文中，我們提出了一個語音辨識後的精鍊處理系統來解決上述的問題。我們在語音辨識之後，運用原有電話查詢領域的知識庫來做後處理，使得使用者的輸入在語音辨識處理之後，能夠進一步地加以精鍊成正確率更高的文句。

1. 簡介

最近這些年來，學者們對口語對話系統進行了廣泛的研究[1]，其目的就是要完成一台可以適當的與使用者對話進而提供服務的機器。目前有許多的應用系統，像是觀光導覽系統[2]，鐵路資訊查詢/定票系統[4]，汽車買賣資訊系統[5]，或者是餐廳的點菜[3,6]等等，都一一被開發出來並且都展示其服務人類的功能。

而對話系統中佔核心地位的語音辨識系統的正確率目前仍然不能達到百分之百，語音辨識的錯誤通常會導致對話系統無法得到正確的結果。因此我們對語音辨識後的結果做一個分析，然後針對語音辨識可能的結果或錯誤如聲調(Tone)、代換(Substitution) 、或者多字(Insertion)、少字(Deletion)、Bigram 語言模型的錯誤問題加以處理，並將它應用在辦公室語音轉接系統[7]上。實驗結果可將句子的關鍵詞的正確率由原來的 72.5%提升到 82.5%，句子的正確率由原來的 65%提升到 78%，而對話的完成率由 78.10%提升到 88.58%。

我們也歸納出一個快速的演算法可以縮減關鍵詞比對的運算量，此演算法將代換、

多字和少字的處理程序合併運算，使得原先比較的運算量，在一字錯誤(1-error)的情形下，時間的複雜度由 $O(n^2)$縮減為 $O(n)$。

本論文的章節架構如下：第二節介紹我們的系統架構；在第三節，我們描述系統知識庫的建立；第四節則介紹我們精鍊模型的建立與處理；第五節說明如何建立快速的演算法以考慮一字錯誤的比對情形；實驗的結果則在第六節描述；最後，第七節則是結論與討論。

2. 系統架構

我們針對語音辨識端的辨識模組所產生的候選音節(syllable lattice)和構句之後的文字串(word sequence)再加以精練處理(refinement)以得到更高的辨識效果。精鍊模組的處理流程如圖 1。

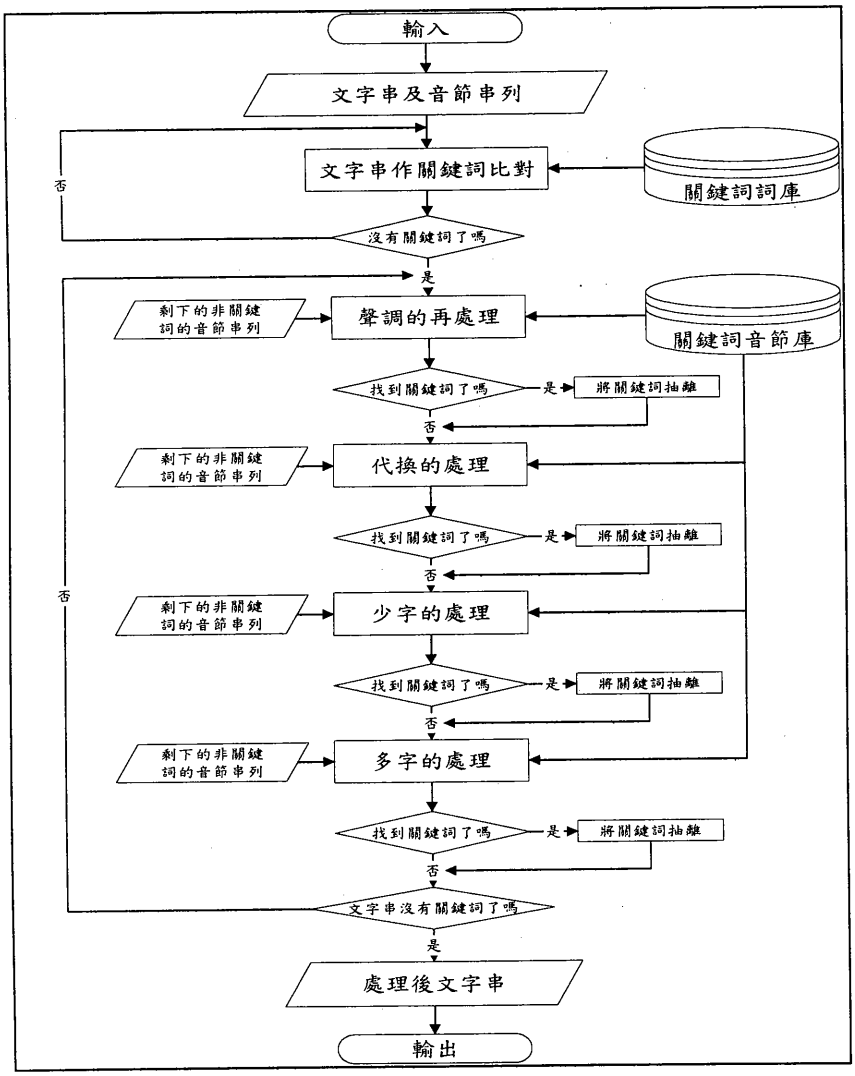

圖 1. 精鍊處理的流程圖

首先，文字串會與關鍵詞庫作比對，將關鍵詞抽取之後，剩下的非關鍵字開始作後處理的動作，我們使用非關鍵詞的候選音節來處理。第一步是作聲調的再處理，我們使用事先轉換好的關鍵詞音節庫作比對，忽略聲調的資訊可以修正大部分的錯誤；接著我們作代換字的處理，除了處理一般音節被代換嘗試修正外，我們並且以候選音節相似度的分數來做考量；再來就是多字少字的處理，這個步驟的混淆度比較高，所以本論文只處理一個字的多字少字的問題。在比對之前我們會先選擇適當的關鍵詞類別來做比對，選擇的依據是以詞類間相連關係來做判定。在處理完之後，我們便將處理後的文字串輸出到對話管理模組跟使用者進行對話。

3. 知識庫的建立

關鍵詞可以作為系統可認知的知識庫，在這裡我們使用查詢系統所定義的關鍵詞來當作精鍊模組的知識庫，我們所訂定的詞庫類別有兩類，每一個類別是以相同的類型作分類，第一類是重要關鍵詞(primary keyword)，是系統認知人員資料的重要資訊，包括姓、全名、性別稱謂、單位、工作/研究領域，第二類是次要關鍵詞(secondary keyword) [9]，定義語者的意圖或語末詞，包括前置詞、後置詞，如下表所示：

詞庫類別	類別名稱	意義	類別資料範例
Primary Keywords	姓(surname)	所有的姓，以百家姓裡的姓為主。	趙，錢，孫，李
	全名(full-name)	人名的全名。	王獻章，劉倚男
	性別(sexual)	男性與女性的性別稱謂。	先生，小姐
	職稱(title)	單位中可能出現的職稱。	教授，所長
	單位(department)	各個單位的名稱。	電算中心，資訊所
	工作/研究領域 (working/research area)	工作及研究領域的名稱，並且將各個名稱可能的簡稱也一併列入。	語音辨識，資訊安全
Secondary Keywords	前置詞 (pre-keyword)	接在重要關鍵詞之前的關鍵詞。	麻煩幫我轉，請幫我接
	後置詞 (post-keyword)	接在重要關鍵詞之後的關鍵詞。	在不在，可以嗎

表 1. 關鍵詞詞庫類別與範例

4. 精鍊模組的建立與處理

4.1 辨識結果錯誤的分析

我們對口述語言對話系統中的語音辨識模組所產生的錯誤作了整理並且將它歸類。這些錯誤有的是與者在輸入語音時發音不良所導致的；有的則是辨識模組的核心程式強健性不足所產生的。這些錯誤分別舉例如下：

1.聲調(Tone)的錯誤：

由於語者說話的口音或者習慣而會有聲調上的辨識錯誤。下面是辨識後聲調錯誤產生錯誤結果的例子如下：

- 使用者：麻煩【找】吳宗憲老師
- 辨識後：麻煩【招】吳宗憲老師

2.代換(Substitution)的錯誤：

語者將某個音唸錯或發音不標準，造成辨識的錯誤，這樣的錯誤會導致配詞結果成為較奇怪的文字串，常常是不合文法或語法或是變成一些贅語，這樣會使得對話管理模組處理困難，降低對話系統處理的正確性，下面是一個代換錯誤的例子：

- 使用者：幫我轉【資訊所】
- 辨識後：幫我轉【制憲所】

3.少字(Deletion)的錯誤：

語者唸的時候，可能是唸太快或者是太小聲，而產生漏字的情形，例如：

- 使用者：可不可以請問一下從台北到【嘉義】的自強號票價多少
- 辨識後：可不可以請問一下從台北到【家　】的自強號票價多少

4.多字(Insertion)的錯誤：

語者唸的時候，可能是唸太慢或者是含混不清，而產生多字的情形，例如：

- 使用者：你好，請幫我接【王獻　章】
- 辨識後：你好，請幫我接【王獻煙章】

5. Bigram 語言模型的錯誤：

由於在通用語料和特定領域語料所合成的 bigram information 中，我們將特定領域的 bigram 的機率分數[10]權重調得比較高，而導致的錯誤，我們可以見下面的例子：

- 使用者：請幫我找【黃】教授好嗎

- 辨識後：請幫我找【煌】教授好嗎

上面發生錯誤的原因是因為語料庫中"郭耀煌教授"出現頻率很高，所以 bigram 的分數比較高，因而造成這種錯誤。

4.2 精鍊的處理

4.2.1 聲調(Tone)的再處理

聲調再處理的流程圖如圖 2.所示：

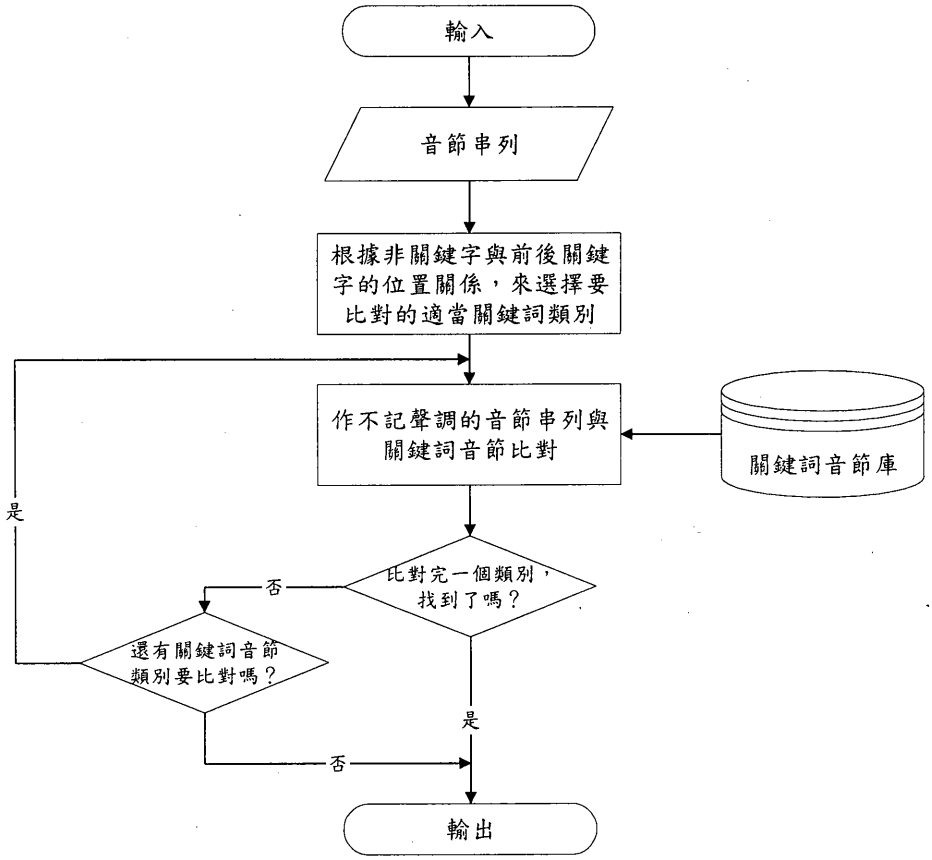

圖 2. 聲調再處理的流程圖

首先我們將關鍵詞庫的音節資訊建立成音節庫。接下來做比對的時候要挑選適當的關鍵詞類別來跟音節串列作比對。我們藉由候選詞與前後可能接的重要關鍵詞的關係來選擇比對的類別，然後我們開始作關鍵詞的比對。比對的方式是將關鍵詞音節與音節串列先做去聲調的處理，然後再比對出相吻合的，如果比對成功就跳出聲調處理的副程式；否則會繼續作下一個類別的比對，直到沒有任何關鍵詞音節。關鍵詞前後相連的規則，根據我們觀察的結果，定義如表 2 所示。

前後已經存在關鍵詞的數目	候選詞與關鍵詞的位置	組合情形	可能的候選詞類別
無	不用考慮	不用考慮	Fullname，Department，Sex，Title，Research，Prekeyword，Postkeyword
一個	候選詞在關鍵詞之前	候選詞＋F 候選詞＋D 候選詞＋S 候選詞＋T 候選詞＋R 候選詞＋Post	D,T,Pre D,Pre F,Pre F,D,R,Pre F,D,Pre F,D,S,T,R,Pre
	候選詞在關鍵詞之後	F+候選詞 D+候選詞 R+候選詞 Pre+候選詞	S,T,R,Post F,D,T,R,Post S,T,Post F,D,S,T,R,Post
兩個	候選詞在前後關鍵詞之間	Pre+候選詞+F Pre+候選詞+D Pre+候選詞+S Pre+候選詞+T Pre+候選詞+Post F+候選詞+D F+候選詞+R F+候選詞+Post D+候選詞+F D+候選詞+S D+候選詞+T D+候選詞+R D+候選詞+Post S+候選詞+D S+候選詞+Post T+候選詞+D T+候選詞+Post R+候選詞+Post	D,S,T,R F,D,T,R F,D,T,R F,D,S,R F,D,S,T,R D,S,T S,T S,T,R D,S,T F,D,T F,D,S F,D F,D,S,T,R D F,R D F,R S,T

表 2. 挑選適當關鍵詞類別方法表

其中 F 表示全名(Full-name)；D 表示單位(Department)；S 表示性別(Sex)；T 表示職稱(Title)；R 表示研究領域/工作性質(Researching Area / Working area)；Pre 表示前置詞(Pre-Keyword)；Post 表示後置詞(Post-Keyword)。

4.2.2 代換(Substitution)的處理

從這個步驟開始，一直到多字情形的處理，聲調的資訊就不再考慮。代換處理的流程圖如圖 3. 所示。

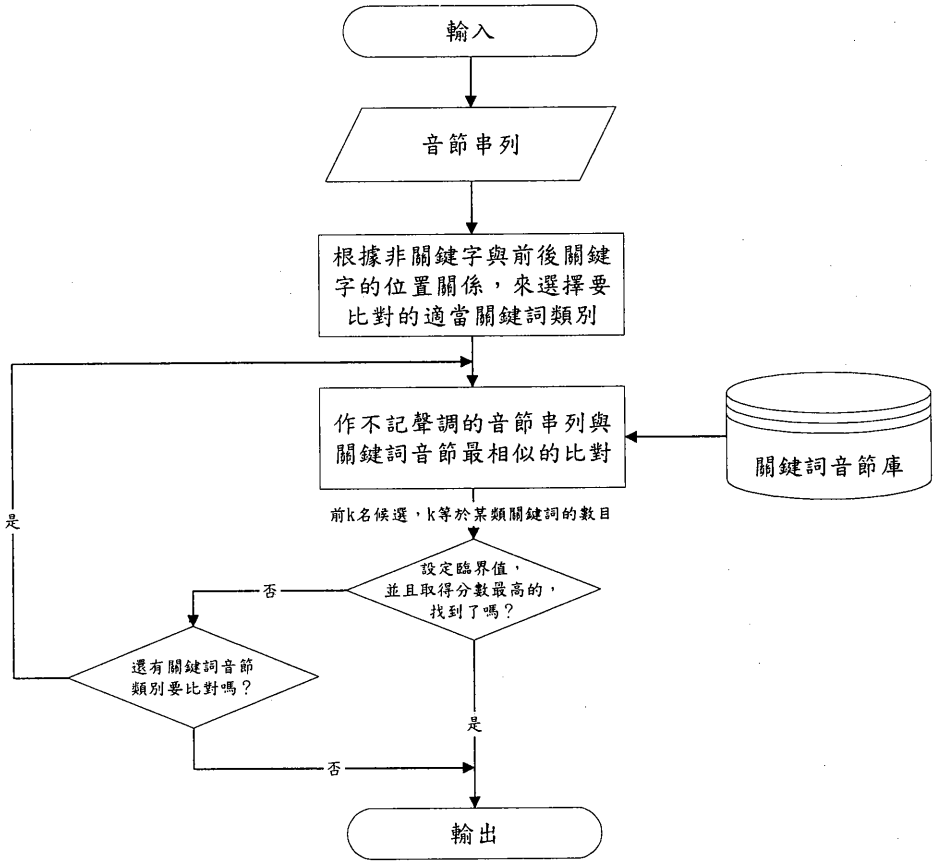

圖 3. 代換的處理流程圖

關鍵詞音節類別的選擇方法如上一節表 2.所示。接下來要作最大相似度的比對，我們定義了分數給定的原則如下表：

相似度	分數
完全正確(子音＋母音)	10
只有母音正確	7
沒有正確的	4

表 3. 音節相似度比對的分數定義表

我們紀錄每個關鍵詞音節的分數，然後選擇一個最高的分數：

$$S_{max} = Max(S_0, S_1, \ldots, S_n), \qquad S_i >= threshold \tag{1}$$

其中 S_i 表示第 i 個關鍵詞的分數。臨界值的計算方式如下，n 為候選音節長度：

$$\text{threshold} = \begin{cases} (n-1)*10+7, if & n=2 \\ (n-1)*10+4, if & n>2 \end{cases} \tag{2}$$

音節分數大於等於臨界值的關鍵詞才有可能是我們要的。在兩字時(n=2)可以處理一個音

節完全正確及另一個音節的母音資訊正確的情形；在三字以上時(n>=3)可以處理一個音節的代換(沒有正確的)或者 n-2 個音節完全正確而且兩個母音資訊正確的情形。圖 4.是比對過程示意圖。

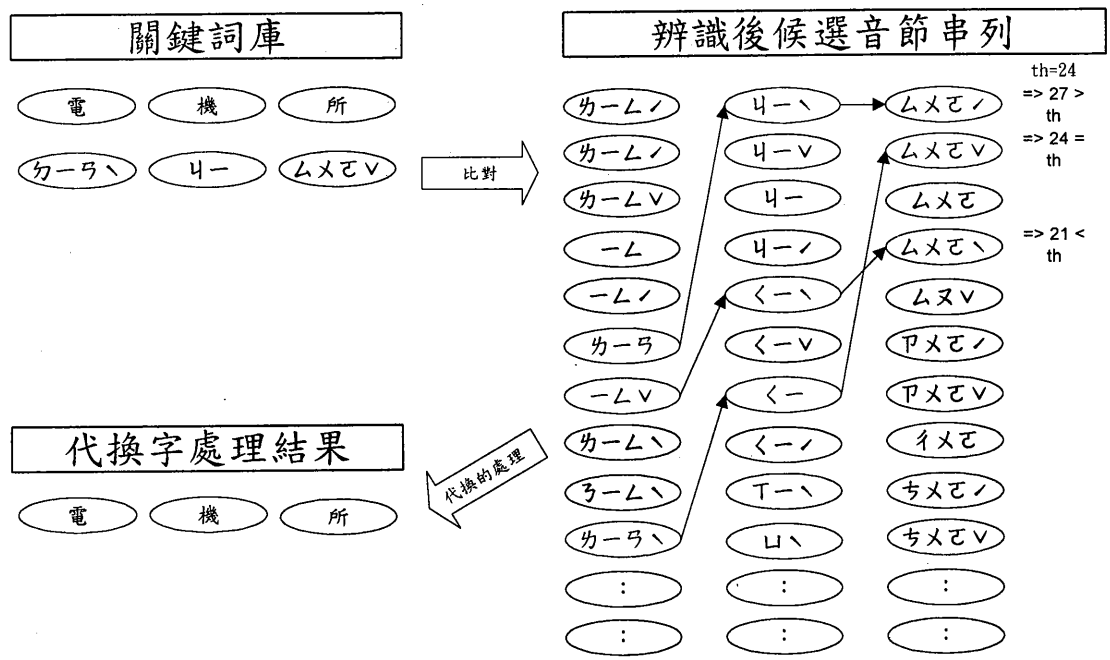

圖 4. 代換字的處理範例圖

4.2.3 少字(Deletion)的處理

少字處理的方式與代換相似，不同的是我們將關鍵詞音節從第一個到最後一個音，輪流刪掉一個之後再跟音節串列作比對，然後選擇一個最高的分數：

$$S_{max} = Max(S_0, S_1, \ldots, S_n), \qquad S_i >= threshold$$

臨界值公式如下，n 為候選音節長度：

$$threshold = n*10, n \geq 2 \tag{3}$$

表示要全部比對正確，加上剛才刪掉的一個音節，就成為少字的情形，找到了就會傳回關鍵詞的字串了。

4.2.4 多字(Insertion)的處理

在多字的處理方式同樣與代換相似。我們將關鍵詞音節從第一個到最後一個音，輪流插入一個音節之後再跟音節串列作比對，插入的音節我們設定為空白，因此分數設為 0，然後選擇一個最高的分數：

然後選擇一個最高的分數：

$$S_{max} = Max(S_0, S_1, \ldots, S_n), \qquad S_i >= threshold$$

臨界值的設定如下，n 為候選音節長度：

$$threshold = (n-1)*10, n \geq 3 \tag{4}$$

表示若比對 n-1 個正確，就可能是多字的情形，找到了就會傳回關鍵詞的字串了。

4.2.5 Bigram 語言模型錯誤的處理

Bigram 語言模型的錯誤，要能夠檢查的出來，必須使用文法的規則來檢查這個錯誤，我們若知道各個關鍵詞可能的連接規則，例如「姓+職稱」或者「姓+性別」等等那我們就可以修正這樣的錯誤。

我們在前面 4.2.1 到 4.2.4 小節的關鍵詞比對之前，會先依據我們所訂定的關鍵詞連接規則來作關鍵詞類別的選擇，這個動作就可以修正這類型的 Bigram 錯誤了。

5. 一字錯誤(1-error)快速演算法的建立

上一章提到的處理方法，需將音節串列與很大的資料庫作比對，雖然先對要比對的資料庫作一個篩選，但是其運算量還是很大，會增加系統處理的時間，我們希望能夠將資料庫比對的計算時間縮減。我們分析整個比對的過程，並找出能夠減少運算的方法。在只有一字錯誤的情形下，若以直覺式(straight forward)的方法，其計算量的分析如圖 5.所示。在此，我們計算比較的次數，因為這個數量佔了大多數的時間。圖 5.中，每個圓圈表示關鍵詞音節庫與音節串列比較一次，其中 X 表示可以為任意字。

1. 多字的情形：

四字時(n=4)時，例如「中華民國」，會有 n-1 個插入字的位置，而每行字則比較(n+1)次，所以總共$(n-1)*(n+1)=n^2-1$ 次。

2. 少字的情形：

在 n=4 時，n 從 1 到 n 會各少一次，而每一行則剩下(n-1)次，所以總共比較了$n*(n-1)=n^2-n$ 次。

3. 代換的情形：

與少字的情形相似，在 n=4 時，n 從 1 到 n 會各被其他字所代換一次，而每一行維持 n 次，所以總共比較了$n*n=n^2$ 次。

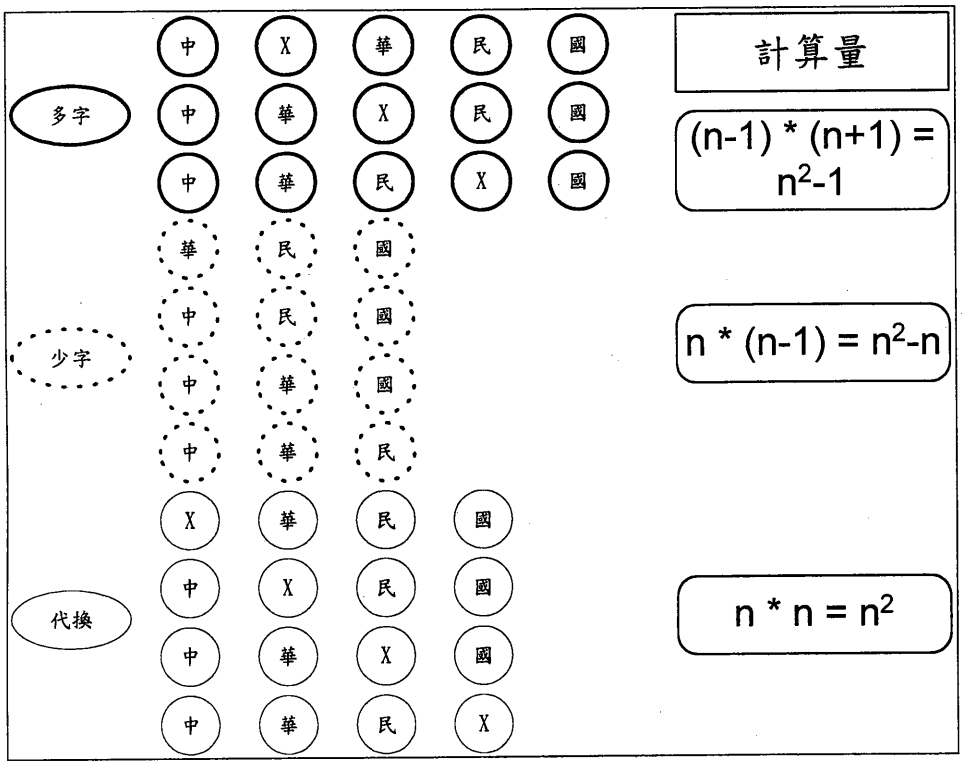

圖 5. 一字錯誤直覺式演算法計算量說明圖，以「中華民國」一詞為例。

接下來，我們將建立一個樹狀的資料結構，並且將上面的資料放入樹的節點，建成樹的方式如下：

1.依照橫列的方式，每一個橫列代表一個樹的分支。

2.將橫列依照從左到右的順序建立成樹的節點。

3.重複(2.)的步驟直到將所有橫列建立完畢。

假設樹高最大是 k 層，則第 k 層為多字，第(k-1)層為代換，而第(k-2)層為少字，範例如下圖：

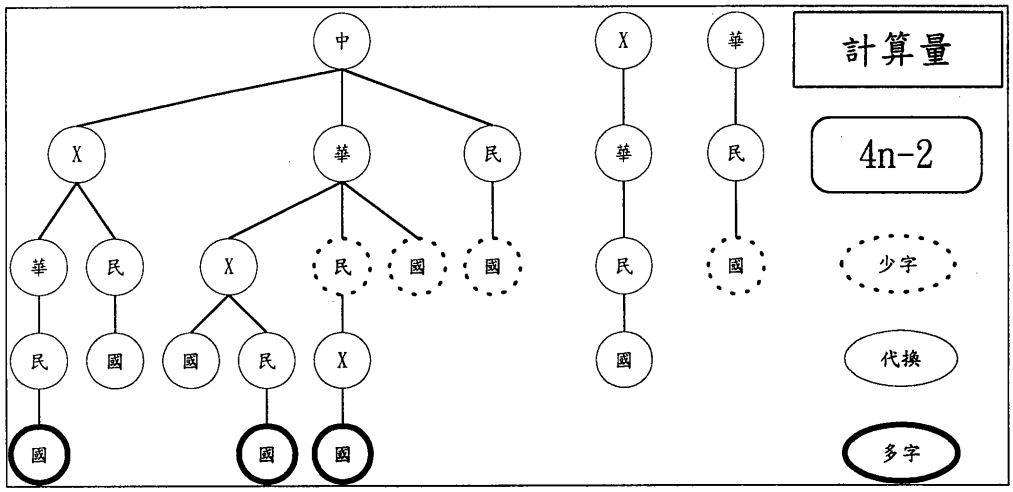

圖 6. 一字錯誤樹狀結構演算法計算量說明圖

舉例說明：從左到右邊，第二子樹，「中華 X 國」為代換，「中華 X 民國」為多字，「中華民」為少字，「中華民 X」為代換，「中華民 X 國」為多字依此類推。

接下來計算它的比較的運算量，設 Tn 代表樹高為 n 的那一層，我們計算每一層出現的不同位置的字的個數，它相當於在直覺式演算法中，每一個直行出現的不同字數，說明如下：

	四字詞「中華民國」	五字詞「資訊工程所」
第 Ti 層	Ti 層需比對的字元	
T0	中, X, 華	資, X, 訊
T1	X, 華, 民	X, 訊, 工
T2	X, 華, 民, 國	X, 訊, 工, 程
T3	X, 民, 國	X, 工, 程, 所
T4	國	X, 程, 所
T5		所

表 4. 一字錯誤情形中各層需要比對的字元範例表

四字時一共是 3+3+4+3+1=14，五字時一共是 3+3+4+4+3+1=18，每增加一個字時，增加 4 個比對時間，因此總計算量為 4n-2，時間複雜度為 O(n)。

6. 實驗

6.1 實驗環境

我們的實驗是以 Pentium 200 個人電腦加上 16 位元聲霸卡、麥克風做測試環境。我們的開發工具是 Microsoft Visual C++ 5.0。語音輸入端是採用連續音辨識程式的 API。文字翻語音的輸出端採用的是成大開發的語音合成 API [8]。

6.2 實驗方法與結果

測試的方式第一種是個別關鍵詞的正確率，我們所使用的資料庫是前面所建立的關鍵詞詞庫，一共有 20 個人名(full name) 、4 個性別稱謂(sex)、10 個職稱(title)、11 個單位(department)、33 個工作/研究領域(working/research area)。表 6 是測試結果，包括語音辨識端的正確率(未經精鍊處理，Base Line)和精鍊模組處理後的正確率(Refined Model)，表中的最後一列是所有關鍵詞正確率的平均，如下所示：

Type	Base Line	Refined Model
Full name	85.0%	92.5 %
Title	82.5 %	90.0 %
Sex	90.0 %	95.0%
Department	92.5 %	95.0 %
Researching area	80.0 %	87.5 %
Average	86.0 %	92.0%

表 6. 個別關鍵詞辨識結果比較表

測試的方式第二種是測試真實語料關鍵詞的正確率，我們測試了 100 句的真實語料。每一句語料都統計所有關鍵詞的個數以及辨認正確的個數；另外是測試整句的正確率，測試方式為，當整句的句子都正確時才算正確。測試結果如表 7 所示，包括語音辨識端的正確率和精鍊模組處理後的正確率。

Type	Base Line	Refined Model
Semantic slot	72.5%	82.5 %
Sentence	65%	78 %

表 7. 語意框的關鍵詞和整句的正確率比較表

測試方式的第三種是測試對話的完成率，我們當自己是使用者然後向系統進行查詢，我們一共測試了 105 組的對話過程。測試結果如表 8 所示：

Type	Base Line	Refined Model
# of dialog	105	105
# of success dialog	82	93
Success rate	78.10%	88.58%

表 8. 對話的完成率

7. 結論與討論

在本論文中，我們提出了解決語音辨識系統常見問題(聲調、代換、少字、多字與 Bigram 語言模型錯誤)的辦法。它可以有效地提高關鍵詞、句子、以至於對話系統的正確率。此外，我們也提出一個增加關鍵詞比對速度的演算法它可以將一字錯誤(1-error)的關鍵詞處理時間由 $O(n^2)$降低為 O(n)。

一套對話系統中，會產生問題的部分，除了本文所提到的之外，尚有很多種，例如 Out-of-Word、Out-of-Grammar、Out-of-Task 等等問題。另外，由於使用者輸入的遲疑、重複等等，會造成二字錯誤(2-error)以上的問題，如何提出有效的方法來解決這些問題，將是我們下一步研究的目標。

參考文獻

[1] Furui and M. M. Sondhi, Advances in Speech Signal Processing, Marcel Dekker, Inc., pp.652-699, 1992.

[2] Victor Zue, James Glass, etc., "Spoken Language System for Human/Machine Interfaces", DARPA N00014-89-J-1332, 1991.

[3] Hsien-Chang Wang, Jhing-Fa Wang, and Yi-Nan Liu, "A Conversational Agent for Food_ordering Dialog Based on VenusDictate", Proceedings of ROCLING X International Conference 1997, pp.325-334.

[4] Bennacef and L. Lamel et al., Dialog in the RAILTEL Telephone-Based System, ICSLP'96 Vol. 1.

[5] Helen M., Senis B, and Victor Zue, et al. WHEELS: A Conversational System in the Automobile Classification Domain, ICSLP'96 Vol. 1. pp. 542-545.

[6] Tsuboi and Y. Takebayashi, "A real-time task-oriented speech understanding system using keyword spotting," Proc. ICASSP, pp.197-200,1992.

[7] Hsien-Chang Wang and Jhing-Fa Wang, "A Telephone Number Inquiry System With Dialog Structure," Proceedings of 1997 Multimedia Technology and Applications Symposium. pp.263-270.

[8] Chung-Hsien Wu, J. F. Wang, etc, "Chinese Text-to-Speech System", National Science Community Project Report, NSC-84-2622-E00-006, 1996.

[9] J. Yang, L. F. Chien and L. S. Lee, Speaker Intention Modeling for Large Vocabulary Mandarin Spoken Dialogues. ICSLP'96, Vol. 2. pp. 713-716.

[10] Jyh-Shing Shyuu, Jhing-Fa Wang, "A Speech Input Interface for Web Page Query Based on A Dynamic Language Model Architecture", ICCE'98