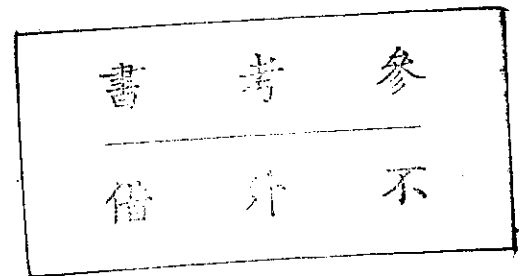


TR-87-007

一個沒有混淆現象的中文注音輸入法之探討

A Study of Disambiguous Chinese  
Phonetic Input

鄭國揚 · 余方國



中研院資訊所圖書室



3 0330 03 00067 8



0067

目 錄

---

誌 謝	.....	i
摘 要	.....	ii
第 一 章	簡介 ( INTRODUCTION ) .....	1 . 1
第 二 章	基本的解決方法 ( A BASIC APPROACH OF DISAMBIGUITION ) .....	2 . 1
第 三 章	簡單詞和辭典 ( SIMPLE PHRASE PATTERNS AND DICTIONARY ) .....	3 . 1
第 一 節	簡單詞和語意屬性 ( SIMPLE PHRASE PATTERNS AND SEMANTIC ATTRIBUTE ) ..	3 . 1
第 二 節	辭典的結構 ( THE ARCHITECTURE OF DICTIONARY ) .....	3 . 10
第 四 章	語法詞和語法規則 ( SYNTACTIC PHRASE PATTERNS AND SYNTAX RULES ) ..	4 . 1
第 五 章	語意匹配 ( SEMANTIC MATCHING ) .....	5 . 1
第 六 章	使用者意願的混淆現象和特殊鍵 ( THE AMBIGUITY OF USER - WANTED AND SPECIAL KEY ) .....	6 . 1
第 七 章	辨認機和轉換過程 ( A DISAMBIGUITION MACHINE AND THE PROCESS OF TRANSLATION ) .....	7 . 1
第 八 章	應用範圍和學習能力 ( APPLICATION DOMAIN AND LEARNING ABILITY ) .....	8 . 1
第 九 章	結論 ( CONCLUSIONS ) .....	9 . 1
參考文獻	( REFERENCES ) .....	R . 1

## 摘 要

---

中文輸入法的研究是發展“中文電腦化”的一個基本的努力方向。在現存的中文輸入法之中，注音符號輸入法是使用最普遍的輸入法之一：其原因在於一般人都曾經在國民教育的階段受過注音符號的訓練，使得注音符號輸入法比較容易接受與學習；但是，注音符號輸入法有一個很嚴重的缺點，就是同音字造成了輸入時的混淆現象（Ambiguity），因此，當使用者在使用注音符號輸入法時，必須目視螢幕以選擇正確的用字，這種情形不但增加了使用者的負擔，同時降低了輸入的效率。

本篇論文的目的即在提出一種新的注音符號輸入方式，以解決同音字所造成的問題。在這種輸入方式之中，是以一個注音符號的句子為輸入，這個句子將藉由系統內部的轉換過程將它直接轉換成相對應的正確中文句子為輸出；我們利用到在人工智慧上處理自然語言的一些方法來處理整個轉換的過程：這些方法包括語法辨認（Syntactic disambiguation）、語意匹配（Semantic matching）以及往覆式的剖析（Backtract parsing）。此外，我們將從理論上說明這種沒有混淆現象的注音符號輸入裝置（Disambiguous Chinese phonetic input device）是一個“零儲存的線性限制自動機”（Empty store linear \_ bounded automaton）。

在論文之中我們亦將提出一種人、機交談式的學習方式供使用者  
教導系統認識更多的生字、生詞或用法，以期能逐漸符合使用者在特  
殊範圍的應用需要。

# 第一章 簡介 ( INTRODUCTION )

在現存的中文輸入法之中，注音符號輸入法是使用最普遍的輸入法之一；〈表 - 1〉 [ 1 ] 中列出目前一些十六位元中文微電腦系統所最常採用的輸入法，我們可以發現注音符號輸入法是其中唯一共同擁有的中文輸入法，足見注音符號輸入法的普遍程度。

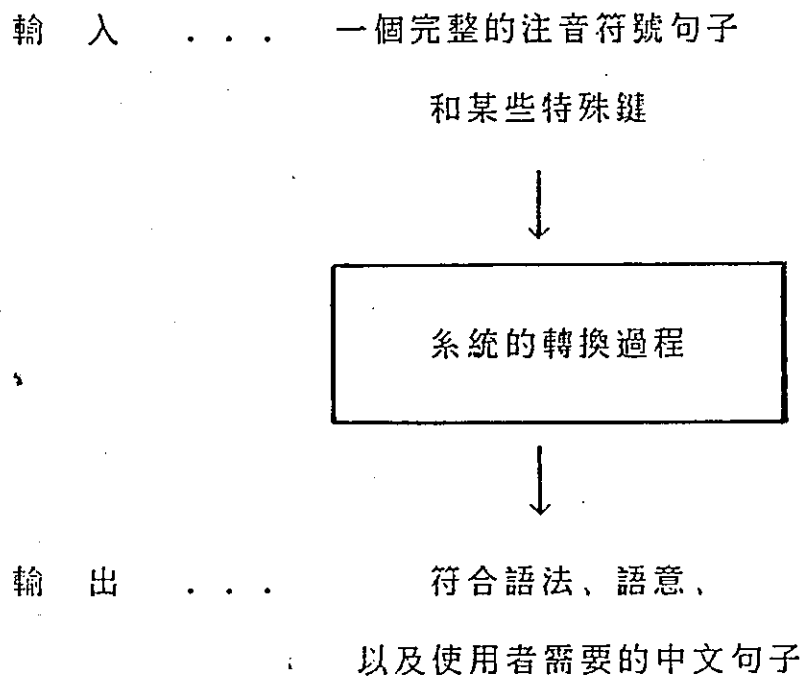
種類 產品類別	倉頡輸入法	注音符號法	簡易輸入法	內碼輸入法	電報明碼法	天龍速成法	三角號碼法	天龍字母法	對角碼法	片語輸入法	四角號碼法	綜合法	漢通輸入法	備註
神通中文		✓		✓						✓			✓	
倚天中文	✓	✓	✓	✓	✓	✓	✓			✓				ET2416
國喬中文	✓	✓	✓	✓					✓					
大千中文	✓	✓		✓										
精業中文	✓	✓			✓					✓	✓	✓		
零壹中文	✓	✓	✓											CJ2025E
5550 中文	✓	✓		✓	✓									
天龍中文		✓		✓	✓	✓		✓						
群智中文	✓	✓	✓							✓				

〈表 - 1〉

但是，由於中文字中有同音字的問題存在，使得注音輸入法本身就具有輸入上的混淆現象 ( Ambiguity ) ；根據統計資料 [ 2 ] 顯

示，中文字僅有 1400 多個讀音(包括聲調變化)，而常用字就有 13000 多個，因此，混淆現象的比例相當高。為了解決這個問題，目前的作法是完全依賴使用者目視螢幕來選取正確的用字，這種方式在實用上非常不方便(Not user - friendly)，同時也會造成輸入時間上的浪費。

本篇論文的目的即在提出一種新的注音符號輸入法，以解決因同音字所造成的混淆現象。在這種輸入法之中，使用者只要以一個完整的注音符號句子和某些特殊鍵為輸入，系統就會自動將它轉換成符合語法、語意、以及使用者需要的中文句子；這種輸入法的過程可以圖示如下：



以“他是學生”這個句子為例，使用者只要輸入“去Y < M > 尸、 T 口 世 / 尸 厶 ”（註：< M >代表一個特殊鍵的符號），就能夠得到正確的中文輸出。

在整個轉換過程之中，基本上我們將處理兩種不同型態的同音字混淆現象：一種是屬於語法上的（Syntactic），另一種是屬於語意上的（Semantic）。

所謂辨認一個語法上的混淆現象，是指系統如何來決定一組注音符號應該選擇哪些中文字。例如：系統如何正確的來判斷“ㄉㄛ ㄉㄛ、ㄛ ㄛ”只能夠對應“電腦”，而不能對應“佃惱”、“甸腦”或“店腦”。

而辨認一個語意上的混淆現象，則是指當有幾個同音詞存在的時候，系統如何來決定哪一個同音詞才是這個句子中正確的用詞。例如：“< ㄉㄛ ㄉㄛ / ㄉㄛ ㄉㄛ、 ”這組注音符號能夠對應三個同音詞“全力”、“權力”和“權利”，系統如何來判斷只有“全力”才能用在像“全力做好自己份內的工作”這樣的一個句子之中？

在整個轉換過程之中，我們用來解決語法和語意上混淆現象的方法基本上非常類似於對中文句子作自然語言的了解（Natural language understanding）[ 3 ]；但是，實際上自然語言的了解是針對整個句子作語法和語意上的分析，並進一步來掌握整個句子的意義，而我們所處理的注音符號句子的轉換過程僅僅是從一個句子中作

局部上( Local ) 的語法辨認和語意匹配，所以，兩者在本質上是不同的。

在實際應用這種輸入法時，由於系統所能適用的文字範圍，會隨著使用者的需要、或者外在環境的改變而改變，因此，系統的學習能力將是一個重要的考慮因素；為了能夠增加系統適用範圍的彈性，本篇論文之中亦將提出一種交談的方式，讓使用者能夠刪改、或是增加系統所能處理的文字範圍，以符合使用上的需要。

我們把本篇論文的綱要列在下面：

第二章中將簡單地說明要解決同音字混淆現象的問題何在？以及它們基本的解決方法。

第三、四章中分別說明如何對不同型態的詞作個別的辨認。

第五章中說明如何從語意上的觀點來作同音詞的辨認。

第六章中說明一種特殊的同音字混淆現象，和它們的解決方式。

第七章中說明實際的轉換過程以及辨認機的理论架構。

第八章提出一個人、機交談式的學習方式。

第九章作一個結論。



## 第二章 基本的解決方法

### ( A BASIC APPROACH OF DISAMBIGUITION )

這一章將說明在轉換一個注音符號句子成為相對應中文句子的過程之中所要解決的問題是什麼？這些問題包括：什麼是一個“詞”？如何對一個“詞”作辨認？如何從幾個同音詞中找出正確的用詞？

要將一個注音符號的句子轉換成相對應的中文句子，一個直覺的想法是：一般的注音符號輸入法是以一個字一個字來考慮它對應的情形，因此同音字造成了嚴重的混淆現象；如果我們將詞的因素加進來，是不是能夠產生一個注音符號詞和中文詞之間一一對應的關係？例如使用者輸入下面這個注音符號的句子：

” ㄉㄨㄛˊ ㄇㄞˋ ㄓㄨㄛˊ ㄓㄨㄥˊ ㄍㄨㄛˊ ㄕㄨㄛˊ ㄊㄨㄛˊ  
ㄒㄨㄥˊ ㄩㄥˊ ”

需要轉換成相對應的中文句子，如果在我們的資料庫( Data base )  
中存在下面這一組一一對應的關係：

” ㄉㄨㄛˊ ㄇㄞˋ ㄓㄨㄛˊ ” → ” 電 腦 ”  
” ㄓㄨㄥˊ ㄍㄨㄛˊ ” → ” 能 夠 ”

” 出ノ 丁一ノ ” → ” 執 行 ” 。

” 彳ノ 尸、 ” → ” 程 式 ” 。

那麼我們只要找到這一組對應關係，接下來很容易就可以將上面的句子從左到右依次轉換成為 ” 電 腦 能 夠 執 行 程 式 ” 這個中文句子。

但是，一般來說，這種以詞為單元的轉換方式所能夠處理的文字範圍非常有限，其原因有以下幾點：

第一、注音符號詞和中文詞之間一一對應的關係在實際的應用範圍之中是不存在的；例如：”全力”、”權力”和”權利”就是對應於” <ㄎㄨㄢˊ ㄉㄨㄛˋ ㄉㄨㄛˋ ” 的三個同音詞。

第二、一般的中文句子不能完全分割成以詞為單元的句子，例如：”他買了一枝筆”這個句子本身就是以字為單元所組成的，因此，在這種情形之下，以詞為單元的轉換方式不能夠使用。

第三、有一些詞不適合存進資料庫之中；例如：一些由數目字所構成的詞，像：”一千兩百”，或是一些描述動作的語態詞，像：”做過”、”做了”、”做著”，或是一些描述時間的詞，像：”上個月”、”下個月”、”上一個月”、

“下一個月”等等，這些詞因為數量太多而無法逐一存進資料庫中。

雖然這種以詞為單元的轉換方式有一些明顯的缺點，可是它提供了解決同音字混淆現象的一個基本、而且重要的方向，下面我們將考慮要如何來解決上面所提到的三個問題？

要解決在上面所提到的三個問題，首先必須對“詞”作進一步的考慮，我們把一個“詞”看作是單字、一般性的詞或是語法詞三者之一（詳細的情形將在下面說明），這種一般化的定義將有助於對“詞”作個別的辨認——我們稱之為“語法辨認”（Syntactic disambiguation），其次，我們再來考慮如何辨認“同音詞”的情形——我們稱之為“語意匹配”（Semantic matching）。下面將先說明什麼是一個“詞”和它們基本的辨認方式，細節將在以下幾章中討論。

我們考慮所有的中文句子都是由“簡單詞”（Simple phrase patterns）和“語法詞”（Syntactic phrase patterns）兩種詞所組成的。

定義 2.1 一個“簡單詞”包括下面兩種情形：

- ( a ) 長度 = 1，而且能夠單獨使用的字：如：“他”、“買”、“了”、“一”、“枝”、“筆”、... 等等。

( b ) 長度  $> 1$  的一般性的詞：如：“電腦”、“能夠”、“執行”、“程式”、“中華民國”... 等等。 □

所有“注音符號詞”和“簡單詞”之間的對應關係都將被直接存在一種適當的資料結構( Data structure )中以備轉換過程使用，這種資料結構我們稱之為“辭典”( Dictionary )。“辭典”基本上可以看做是一組注音符號詞和簡單詞之間一對多的對應關係；一個比較正式的定義如下：

定義 2.2 D 代表一個“辭典”，

$$D = \{ m \mid m : ( P s )^n \rightarrow ( C )^n ,$$

其中 P s 代表注音符號 ,

C 代表中文字 ,

n 代表詞的長度 ,

n 為正整數 } □

因為在“辭典”之中除了這種對應關係之外，還必須儲存一些分辨同音詞的資料，同時牽涉到儲存空間( Memory space )和搜尋速度( Search speed )上的問題，所以它實際的結構將在下一章中說明。下面舉出“辭典”的一個例子：

例：  $D = \{ \text{ㄍㄨㄢˇ ㄑㄩㄥˋ} \rightarrow \{ \text{全力, 權力,} \}$

權利 } ,

” 去大 / ” → { 塘 , 糖 , 醴 } ,

” 尸 ㄅ 、 ” → { 再 , 在 , 載 } ,

” ㄉ 一 ㄇ 、 ㄋ ㄨ ㄨ / ” → { 電腦 } ,

” ㄋ ㄨ ㄥ / ㄍ ㄨ ㄨ 、 ” → { 能夠 } ,

” ㄕ / ㄒ 一 ㄥ / ” → { 執行 } ,

” ㄒ ㄥ / 尸 、 ” → { 程式 } }

□

如果我們只利用“辭典”中這種一對多的對應關係，而不對同音詞作進一步的辨認，基本上我們可以考慮一種完全不涉及語意的轉換方式。假設在一組同音詞之中存在一個“優先次序”(Priority)如下：兩個同音詞  $x$  和  $y$ ，如果  $x$  的優先次序大於  $y$ ，則代表  $x$  的使用比例大於  $y$ 。這種轉換方式是：如果有同音詞的情形發生時，我們將選取“優先次序”最大的同音詞；這種完全不考慮語意是否正確的轉換方式，事實上在第一次選取之下已經可以達到 80 % 的正確性，這可以根據 Knuth [ 4 ] 的 80 / 20 Law 得到理論上的證明，同時在 [ 5 ] 中也得到實驗上類似的結果。

因此，為了達到正確的中文輸出，我們將進一步考慮如何利用適當的語意關係來作同音詞的辨認？我們將從兩方面來做：首先我們

對一個詞附加它在語意上的意義，我們稱之為“語意屬性”（Semantic attribute），使得每一個詞都能夠在語意的層次上和它的同音詞分開；其次，我們將從一個句子中局部上的語意來決定出應該選擇哪一個同音詞，詳細的情形將在以下幾章中說明。

定義 2.3 一個“語法詞”是指一個詞具有某種特殊的語法結構，例如：“一千兩百”、“做過”、“做了”、“做著”、“上個月”、“下個月”、.. 等等。 □

我們將採用一種間接的方法來解決“語法詞”，而不把它們直接存在“辭典”之中；這種方法不但能夠辨認出一個“語法詞”，同時可以決定出它的“語意屬性”。第四章中將討論“語法詞”和它的處理方式。

### 第三章 簡單詞和辭典

#### ( SIMPLE PHRASE PATTERNS AND DICTIONARY )

在這一章中我們將說明一個“簡單詞”和它“語意屬性”之間的關係，同時說明如何在兼顧儲存空間和搜尋速度的條件之下，將它們適當的安排在辭典中。

#### 第一節 簡單詞和語意屬性

##### ( SIMPLE PHRASE PATTERNS AND SEMANTIC ATTRIBUTE )

我們先正式定義什麼是一個詞的“語意屬性”？

定義 3.1 對所有的詞我們都可以根據它的意義作適當的分類，一個意義的類別我們稱之為“語意單元”( Semantic primitive )。一個詞的“語意屬性”( Semantic attribute ) 包括下面兩項：

1. “語意單元”( Semantic primitive , 縮寫成 Sp )：代表這個詞是屬於哪一個語意單元？
2. “語意子類”( The subclass of semantic primitive , 縮寫成 Sc )：當有另外一個同音詞和它屬於相同的一個語意單元時，這一項代表這個詞在這個語意單元中的子類。□

從這個定義中，我們可以注意到如何來選取一個適當的“語意單

元集” ( The set of semantic primitives ) 將對 “同音詞” 的辨認產生很重要的影響，以下我們將討論一個針對注音符號混淆現象的 “語意單元集”。

基本上 “語意單元” 可以分成兩大類：

第一類用來說明句子中所可能出現的角色，或是用來描述角色的狀態。這一類的語意單元包括下面九個：

- ( a ) AGENT : 代表有關作用者的詞。例如： “張三”、 “李四”、 “人”、 “動物”、 ... 等等。
- ( b ) EVENT : 代表有關事情的詞。例如： “活動”、 “研究”、 ... 等等。
- ( c ) LOCATION : 代表有關地點的詞。例如： “學校”、 “禮堂”、 “教室”、 ... 等等。
- ( d ) MATERIAL : 代表有關物品的詞。例如： “電腦”、 “糖”、 “糖”、 ... 等等。
- ( e ) TIME : 代表有關時間的詞。例如： “上午”、 “下午”、 “晚上”、 ... 等等。
- ( f ) CONCEPT : 代表有關觀念的詞。例如： “主義”、 “民主”、 “自由”、 ... 等等。
- ( g ) STATE : 代表有關狀態的詞。例如： “好”、 “壞”、 “大”、 “小”、 “漂亮”、 ... 等等。



( h ) QUANTITY : 代表有關數目的詞。例如: “一”、“二”、“三”、“許多”、...等等。

( i ) UNIT : 代表有關單位的詞。例如: “篇”、“張”、“章”、“支”、“隻”、...等等。

第二類用來說明句子中所可能出現的動作，這一類的語意單元我們根據 Schank 的“觀念依存理論”( Conceptual dependency theory ) [ 6 ] 和趙元任的“中國話的文法” [ 7 ] 歸納成以下五類:

( a ) MOVE : 代表有關物品移動、出現或消失的詞。例如: “來”、“去”、“走”、“跑”、“跳”、“活動”、...等等。

( b ) MTRANS : 代表有關思想或感情的詞。例如: “愛”、“恨”、“想”、“教導”、“學習”、...等等。

( c ) PROPEL : 代表有關使用或施力於一個物體的詞。例如: “用”、“拿”、“捉”、“推”、“拔”、...等等。

( d ) SPEAK : 代表有關發出聲響的詞。例如: “叫”、“喊”、“說”、“唱”、“吼”、...等等。

( e ) INGEST : 代表有關吃、喝或排泄的詞。例如: “吞”、“服用”、“哭”、“分泌”、...等等。

對於第二類的“語意單元”我們通常不再考慮它的“語意子類”，原因是：這種分類已經足夠來辨認同音、而且同樣是描述動作的“簡單詞”。

上述包含十四個“語意單元”的“語意單元集”，基本上能夠適用於任何一個應用範圍；但是，為了配合實際的需要，就必須進一步考慮如何來選取一個適當的“語意子類集”(The set of subclasses of semantic primitives)，使得它能夠完全分開無法用前面這些“語意單元”來辨認的同音詞。針對這個問題，我們可以讓使用者自己定義他個人的“語意子類集”來適應實際的需要。例如下面這些“語意子類”：“HUMAN”、“ANIMAL”、“PLANT”、“FOOD”、“COMPOUND”、“MONEY”、“POWER”、... 對於某些同音詞會有相當好的辨認作用，下面我們會舉出一些例子來說明。

因此，對應於一組注音符號的每一個同音詞都會擁有它唯一的“語意屬性”，而且和其他的同音詞有所不同。我們用幾個例子來說明“同音詞”和“語意屬性”。

例 1 :考慮 "ㄉㄨˊ" 這個注音符號所常用的五個同音字 "堂"、"塘"、"糖"、"醴" 和 "唐"。在〈表 - 2〉中列出了 "塘"、"糖"、"醴" 三個同音詞和它們的語意屬性，其中 "糖" 字和 "醴" 字因為屬於同一個 "語意單元" ( MATERIAL )，所以要用 "語意子類" 來辨別它們。

"ㄉㄨˊ" 的同音詞	語 意 屬 性	
	語 意 單 元	語 意 子 類
糖	MATERIAL	FOOD
醴	MATERIAL	COMPOUND
塘	LOCATION	ϕ

〈 表 - 2 〉

在〈表 - 2〉中，"ϕ" 代表 "塘" 字沒有語意子類這一項。另外在〈表 - 2〉中並未包括 "堂" 字和 "唐" 字的語意屬性，原因是：一般來說，"堂" 字通常用在 "禮堂"、"教堂" 或其他長度大於 1 的詞中，而 "唐" 字通常用在 "唐人"、"唐山"、"荒唐" 或其他長度大於 1 的詞中，都不被單獨使用 ( i. e. "堂" 字和 "唐" 字都不是 "簡單詞" )，所以，我們不對單獨的 "堂" 字和 "唐" 字附加它們的語意屬性。如果使用者有單獨使用這些字的需要時，當然可以把它加進去，同時附加適當的語意屬性使它能夠和其他的同音字分開來。

□

例 2 :考慮 “ㄎㄨㄚˇ ㄉㄨㄛˋ” 所對應的三個同音詞 “全力”、“權力”和 “權利”，〈表 - 3〉中列出了它們的語意屬性。

”ㄎㄨㄚˇ ㄉㄨㄛˋ” 的同音詞	語 意 屬 性	
	語 意 單 元	語 意 子 類
全 力	STATE	φ
權 力	CONCEPT	POWER
權 利	CONCEPT	MONEY

〈 表 - 3 〉

□

例 3 :如果注音符號詞和中文詞之間是一一對應的情形，如：“ㄉㄨㄛˋ ㄉㄨㄛˋ” 只能對應 “電腦”，基本上我們還是要考慮 “電腦” 這個詞的語意屬性，但是它只有 “語意單元” ( MATERIAL )，而沒有 “語意子類”；必須保留 “語意單元” 的原因是：它的語意單元將有助於對句子中其他文字的辨認，這一點將在第五章中進一步討論。

□

關於 “同音詞” 和 “語意屬性” 有兩點需要特別說明：

第一、兩個非同音詞可能會擁有相同的語意屬性；例如：“張三” 和 “李四” 的語意屬性可能都是 ( “AGENT ” , “HUMAN ” ) (

註：“AGENT”是語意單元，“HUMAN”是語意子類)。但是，這對於我們辨認的工作是沒有影響的，原因非常明顯：它們不是同音詞，所以根本不會造成辨認上的困難。

第二、有時候同樣的一個詞可能具有兩種不同的用法；例如：“研究”這個詞在“他正在研究一個困難的問題”這個句子中是作“動詞”用，而在“他的研究有了重要的突破”這個句子中是作“名詞”用。對於這個問題的解決方法是：我們將這兩個“研究”當成是兩個擁有不同語意屬性的同音詞，只是它們的用字是相同的罷了，參考〈表 - 4〉：

” 一乃 / ㄩ - ㄨ、 ” 的同音詞	語 意 屬 性	
	語 意 單 元	語 意 子 類
研 究	PROPEL	φ
研 究	EVENT	φ

〈 表 - 4 〉

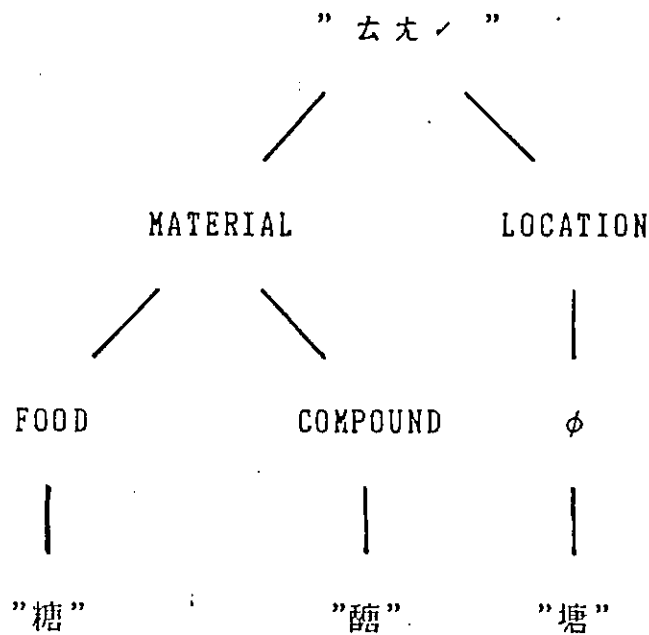
因為這種情形不會很多，所以對於“辭典”而言將不會造成太大的困難。

為了更進一步說明在辨認同音詞的過程之中“語意屬性”所扮演的角色，我們以下利用一種“樹狀結構”(Tree)來說明辨認同音詞的過程。

定義 3.2 一個同音詞的“辨認樹”(Recognize tree) 是一個階數 (Level) 為 3 的樹; 它的樹根 (Root) 代表一個注音符號詞, 以 Ps (Phonetic symboles) 表示; 第一層的子孫 (Descendants of level one) 代表 Ps 同音詞的語意類別; 第二層的子孫 (Descendants of level two) 代表 Ps 同音詞的語意子類; 第三層的子孫 (Descendants of level three) 代表 Ps 的同音詞。 □

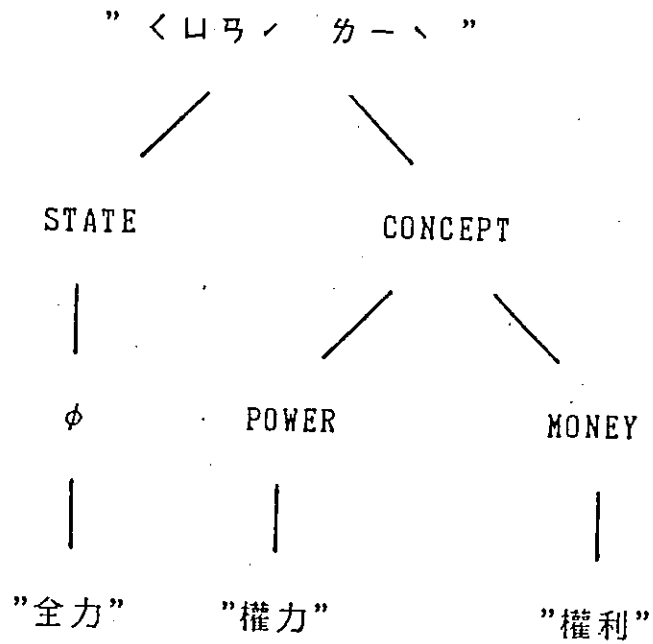
利用“辨認樹”的觀念, 我們可以將“從一組同音詞中決定唯一用詞的辨認過程”看作是“從注音符號詞相對應的辨認樹的樹根經過唯一的一條路徑 (Path) 到達它的一個樹葉 (Leaf) ”。

對例 1 而言, 從定義 3.2 我們可以得到“ㄉㄤˊ”的“辨認樹”如〈圖 - 1〉所示:



〈圖 - 1〉

對例 2 而言，我們可以得到 “ $\langle \text{口} \text{ 弓} \text{ / } \text{ 力} \text{ - } \text{ \setminus } \text{ \rangle}$ ” 的 “辨認樹” 如  $\langle \text{圖} - 2 \rangle$  所示：



$\langle \text{圖} - 2 \rangle$

在第五章中我們將詳細說明如何用 “語意匹配” 的方式來完成這種辨認的過程。

## 第二節 辭典的結構

### ( THE ARCHITECTURE OF DICTIONARY )

在注音符號的句子轉換成為相對應的中文句子的過程中，辭典必須提供所有要用到的簡單詞和簡單詞的語意屬性；同時，我們必需考慮一些相關的因素，包括節省儲存空間( Memory space )和提高搜尋速度( Search speed )。在這一節中將要介紹兼顧到這些限制的一個辭典結構。

一個辭典基本上必須收集所有注音符號詞的辨認樹以備轉換時的需要，因此，一個最簡單的辭典結構是將每一個辨認樹直接記下來，我們可以將一個辨認樹看成如下的一個“串列結構”( List )[ 8 ]：

$$( P s ( S p ( S c , W ) ) ) .$$

利用這種簡單的表示法我們可以將〈圖 - 1〉中“去ㄉㄛ”的辨認樹表示成如下的一個串列：

$$\begin{aligned} & ( " 去ㄉㄛ " ( " MATERIAL " ( " FOOD " , " 糖 " ) \\ & \qquad \qquad \qquad ( " COMPOUND " , " 醣 " ) ) \\ & \qquad \qquad \qquad ( " LOCATION " ( " \phi " , " 塘 " ) ) ) ) . \end{aligned}$$

對於〈圖 - 2〉中“ㄍㄨㄛˊ ㄉㄛˊ”的辨認樹則可以表示



成如下的一個串列：

( " < ㄩ ㄇ ㄛ ㄉ ㄉ ㄉ " )  
    ( " STATE " ( " ϕ " , " 全力 " ) )  
    ( " CONCEPT " ( " POWER " , " 權力 " )  
                    ( " MONEY " , " 權利 " ) ) ) .

這種串列表示法可以用一般的人工智慧語言 ( A. I. Language ) , 如 : LISP 或 PROLOG , 所寫出來。

這種表示法的優點在於很容易表示與了解, 但是對於儲存空間和搜尋速度而言都有負面的影響。

對儲存空間來說, 這種表示法會重覆儲存相同的注音符號和中文字: 原因是可能有很多詞都要用到同一個字, 而這個字的注音符號和中文字都必須在每一個詞的辨認樹中出現一次, 因此造成了不必要的浪費。我們看下面的例子: 如果在辭典中要存 " 塘 "、" 糖 "、" 糖 "、" 糖果 "、" 方糖 "、" 禮堂 "、" 唐人 " 這些詞, 就必須保留下面這些串列

1. ( " ㄉ ㄨ ㄤ ㄉ ㄉ " )  
    ( " MATERIAL " ( " FOOD " , " 糖 " )  
                    ( " COMPOUND " , " 糖 " ) ) )  
    ( " LOCATION " ( " ϕ " , " 塘 " ) ) ) .
2. ( " ㄉ ㄨ ㄤ ㄉ ㄉ ㄉ ㄉ ㄉ " )  
    ( " MATERIAL " ( " ϕ " , " 糖果 " ) ) ) .

3. ( " ㄇㄨˊ ㄉㄨˊ / " )  
( " MATERIAL " ( " ϕ " , " 方糖 " ) ) ) .

4. ( " ㄉㄨˊ / ㄉㄨˊ / " )  
( " LOCATION " ( " ϕ " , " 禮堂 " ) ) ) .

5. ( " ㄉㄨˊ / ㄉㄨˊ / " )  
( " AGENT " ( " ϕ " , " 唐人 " ) ) ) .

而其中 " ㄉㄨˊ / " 這個注音符號和 " 糖 " 這個中文字都被重覆儲存了許多次，因此造成儲存空間上的浪費。

對搜尋速度來說，因為我們要執行往覆式的剖析，所以重覆的選取最長的詞是必要的步驟之一，因此我們希望能夠儘快找到所需要的詞，而不必從頭找一次。我們看下面的例子：如果在辭典中要存 " 台北市 " 和 " 台北 " 兩個詞，在串列表示法中必須表示成兩個串列如下

1. ( " ㄉㄨˊ / ㄉㄨˊ / ㄉㄨˊ / " )  
( " LOCATION " ( " ϕ " , " 台北市 " ) ) ) .

2. ( " ㄉㄨˊ / ㄉㄨˊ / " )  
( " LOCATION " ( " ϕ " , " 台北 " ) ) ) .

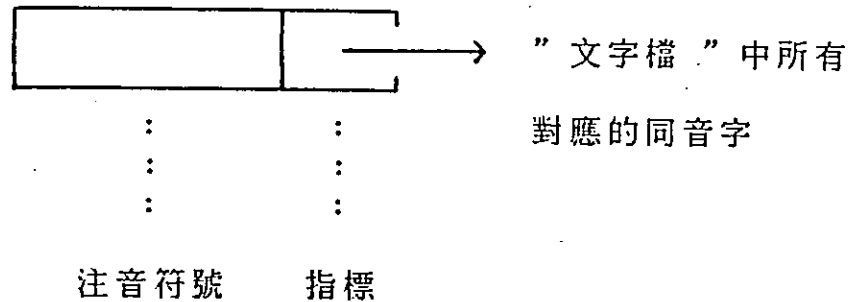
因此對於 " ㄉㄨˊ / ㄉㄨˊ / ㄉㄨˊ / ㄉㄨˊ / ㄉㄨˊ / ㄉㄨˊ / " 這樣的一個句子，第一次會找到第一個串列得到 " 台北市 " 這個詞，如果我們發現 " 台北市 " 不是正確的詞，而需要找另外一個最長的詞，就必須從第一個注音 " ㄉㄨˊ / " 開始找起，才能找到第二個串列

得到“台北”這個詞，因此我們重覆做了搜尋的工作；我們的要求是希望在找最長的詞的過程之中，就將可能的詞一併找出來，省略掉重覆搜尋的工作，以上例來說，也就是一起找出“台北市”和“台北”這兩個詞。對應於“ㄊㄞˇ ㄅㄞˇ ㄊㄞˇ ㄅㄞˇ ㄊㄞˇ ㄅㄞˇ ㄊㄞˇ ㄅㄞˇ”這個注音符號句子的中文句子事實上是“台北是一個都市”。

下面我們提出一個兼顧到儲存空間和搜尋速度的“辭典”安排方式。

我們將“辭典”分成兩部分：第一部分稱為“索引檔”(Index file of phonetic symbols)；第二部分稱為“文字檔”(Master file of Chinese words)。

“索引檔”的目的是將一個注音符號指到“文字檔”中所有相對應的同音字。“索引檔”的每一個索引(Index)可以表示成〈圖 - 3〉：

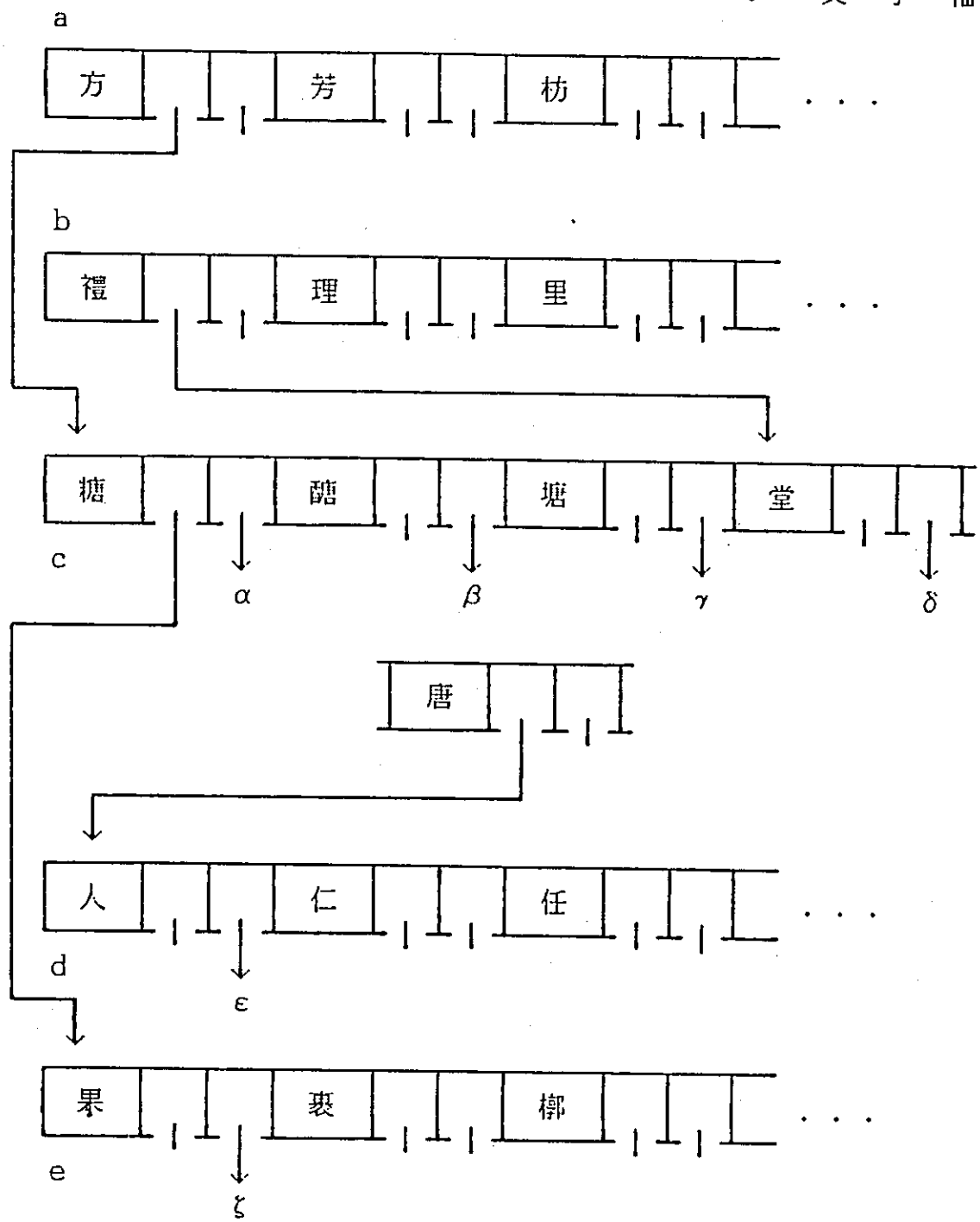


〈圖 - 3〉

“文字檔”的目的是將所有的文字連成簡單詞，並保存它們的語意屬性。“文字檔”中的每一組同音字將表示成〈圖 - 4〉：



▽ 文字 檔



$\alpha = \{ (L1, ("MATERIAL", "FOOD")),$   
 $(L2, ("MATERIAL", "\phi")) \};$   
 $\beta = \{ ("MATERIAL", "COMPOUND") \};$   
 $\gamma = \{ ("LOCATION", "\phi") \};$   
 $\delta = \{ ("LOCATION", "\phi") \};$   
 $\epsilon = \{ ("AGENT", "\phi") \};$   
 $\zeta = \{ ("MATERIAL", "\phi") \};$

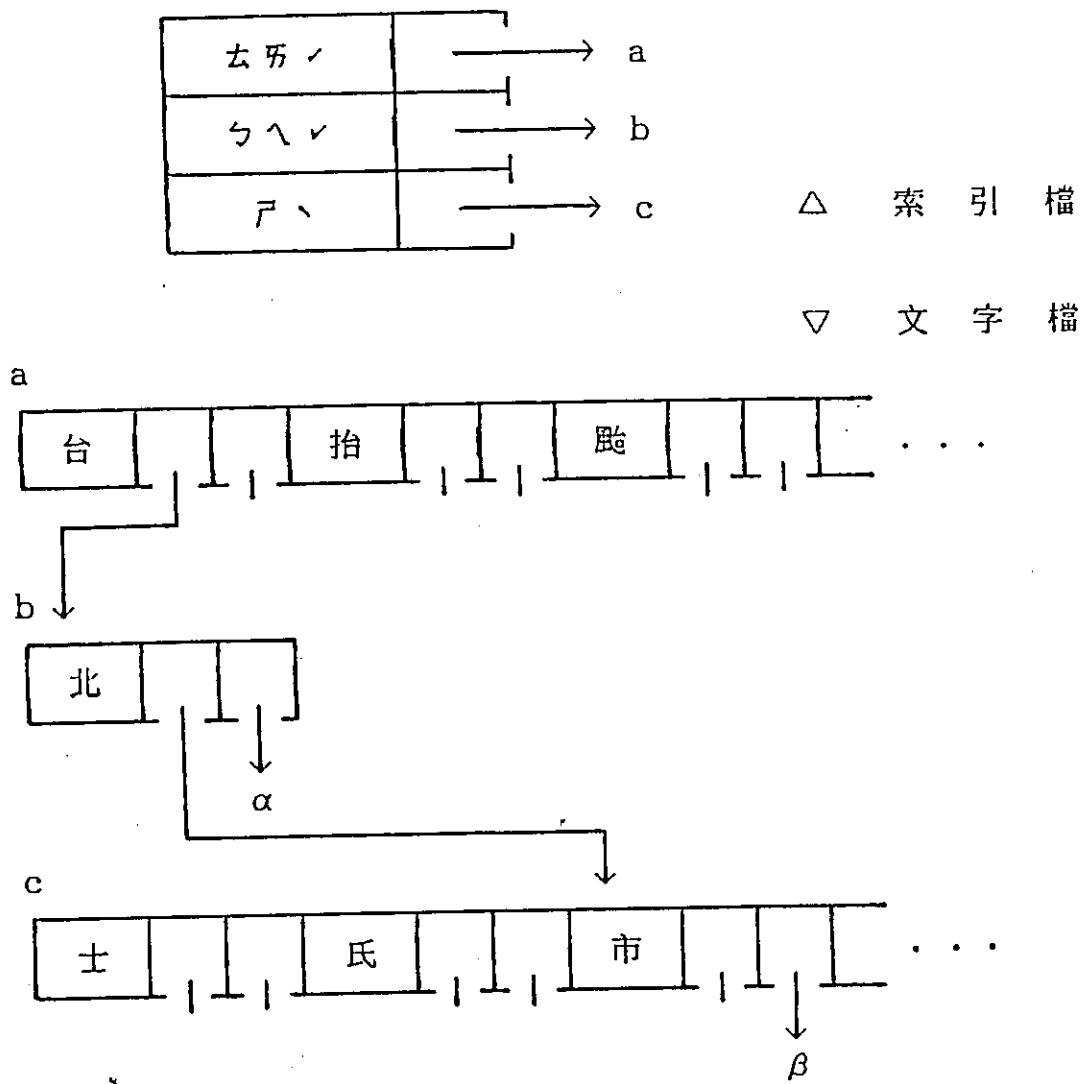
此處 a、b、c、d、e、 $\alpha$ 、 $\beta$ 、 $\gamma$ 、 $\delta$ 、 $\epsilon$ 、 $\zeta$  分別代表實際記憶體中的一個位址 (Address)。在  $\alpha$  中的 L1 和 L2 是兩個特殊的標記 (Label)，用來標明後繼的語意屬性是屬於哪一個詞所擁有的，L1 代表“糖”，L2 代表“方糖”。利用這種“辭典”結構很明顯地可以節省儲存注音符號和中文字的空間 (因為“ㄊㄨˊ”和“糖”都只用了一個位置)。下面我們用“ㄊㄨˊ”和“ㄉㄨˊ ㄊㄨˊ”這兩個注音詞來分別說明如何在這種“辭典”結構中將它們對應的中文詞找出來？

當一個句子中出現“ㄊㄨˊ”這個注音詞時，系統先從索引檔中找到“ㄊㄨˊ”這個注音符號，再找到相對應的文字檔，然後將所有對應於“ㄊㄨˊ”的“簡單詞”(從“語意屬性”的欄位中可以知道)找出來，就能夠得到“糖”、“糖”和“塘”這三個“簡單詞”和它們的“語意屬性”(分別存在  $\alpha(L1)$ 、 $\beta$  和  $\gamma$  中)。

當一個句子中出現 ”ㄉㄤ ㄉㄤˊ ” 這個注音詞時，系統先從索引檔中找到 ”ㄉㄤ ” 這個注音符號，再找到相對應的文字檔，然後從每一個對應於 ”ㄉㄤ ” 這個注音符號的中文字中看它是否會連接到對應於 ”ㄉㄤˊ ” 這個注音符號的某一個中文字，同時它們會構成一個 ”簡單詞”？而符合這種條件的 ”簡單詞” 就是一個對應於 ”ㄉㄤ ㄉㄤˊ ” 的中文詞，因此，我們可以找到 ”方糖” 和它的 ”語意屬性” ( 存在  $\alpha(L2)$  中 )。 □



例 2: 如果在“辭典”中要存“台北市”、“台北”這兩個詞和它們的語意屬性，它們的“辭典”結構安排如下：



圖中  $\alpha = \{ ( " LOCATION " , " \phi " ) \} ;$

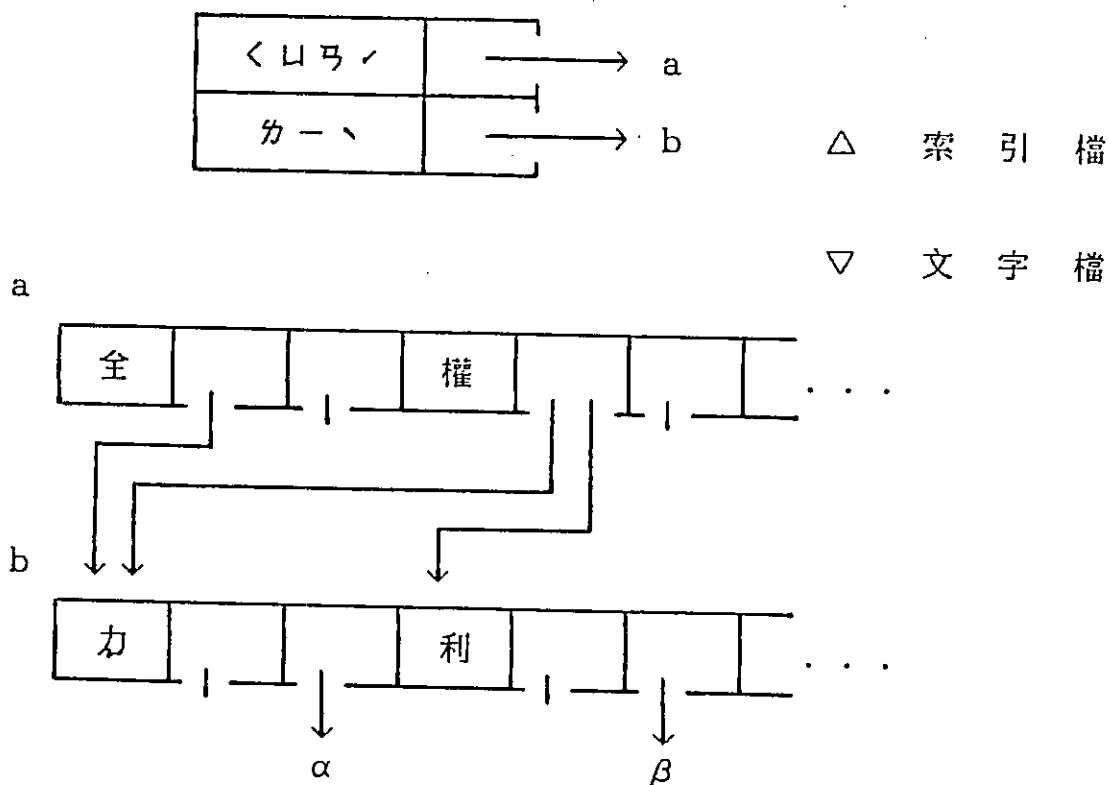
$\beta = \{ ( " LOCATION " , " \phi " ) \} ;$

當一個句子中出現“ㄊㄞˇ ㄊㄞˋ ㄕㄞˋ”這個注音詞時，

系統就能夠在對應的文字檔中找出最長的“簡單詞”（“台北市”）的過程之中，就將其它可能的（長度較短的）“簡單詞”（“台北”）一併找出來，因此，這種“辭典”結構能夠減少搜尋“簡單詞”的速度。

□

例 3：如果在“辭典”中要存“全力”、“權力”和“權利”這三個同音詞和它們的語意屬性，它們的“辭典”結構安排如下：



圖中  $\alpha = \{ (L1, ("STATE", "\phi")), (L2, ("CONCEPT", "POWER")) \}$  ;  
 $\beta = \{ ("CONCEPT", "MONEY") \}$  ;

L1 和 L2 是兩個特殊的標記，L1 代表“全力”，L2 代表“權力”。在“權”字的指標欄位中有兩個指標分別指到“力”字和“利”字，這種兩個指標的情形在實際執行的時候可以用“鏈結串列”（Linked list）[ 8 ] 來完成。 □

在這一章中，我們分別討論了“簡單詞”和“語意屬性”（包括“語意單元”、“語意子類”），以及兩種不同的“辭典”安排方式；下一章將討論“語法詞”的處理。

## 第四章 語法詞和語法規則

### ( SYNTACTIC PHRASE PATTERNS AND SYNTAX RULES )

在這一章中將說明如何來解決“語法詞”無法一一儲存在“辭典”中的問題；我們將採用一種間接的方式來辨認一個“語法詞”，同時利用這種方式能夠自動產生它的語意屬性。

我們對“語法詞”的處理包括兩部份：第一步、將“語法詞”中的每一個單元都當做是一個“簡單詞”儲存在“辭典”中，同時分別記錄它們的語意屬性；第二步、利用一些簡單的規則來表示哪些“簡單詞”可以連接成“語法詞”，並結合出這個“語法詞”的語意屬性，這種結構上的規則我們稱之為“語法規則”(Syntax rule)。利用“語法規則”來辨認一個“語法詞”的優點是：我們僅利用很少的語法規則就可以辨認全部的“語法詞”，而不必將“語法詞”一一存放在“辭典”中。下面將討論“語法規則”的型式和一些例子。

定義 4.1 因為一個“語法規則”必須儘可能在句子中找出最長的一個“語法詞”，因此，它擁有如下的幾種遞迴(Recursive)型式：

$$1. \quad Y^* \rightarrow Y \quad \text{或}$$

$$2. \quad ( Y^* + X^* ) \rightarrow X \quad \text{或}$$

$$3. (X + Y^{*})^{*} \rightarrow X ;$$

此處 X , Y 代表一個詞的語意屬性。 □

對第一類“語法規則”而言，它所代表的意義是說：幾個擁有語意屬性“Y”的“簡單詞”，可以將它們合併成一個“語法詞”，而這個“語法詞”的語意屬性也是“Y”。下面是這一類“語法規則”所能應用的一些例子。

例 1：如果我們將“Y”代換成“QUANTITY”，第一條“語法規則”就可以表示成

$$\text{QUANTITY}^{*} \rightarrow \text{QUANTITY}$$

因此，利用這條“語法規則”就可以將“一千兩百”這種“語法詞”辨認出來（因為“一”、“千”、“兩”、“百”這些簡單詞的“語意屬性”都是“QUANTITY”），同時，能夠得到“一千兩百”的“語意屬性”為“QUANTITY”。 □

例 2：如果我們將“Y”代換成“PROPEL”，第一條“語法規則”就可以表示成

$$\text{PROPEL}^{*} \rightarrow \text{PROPEL}$$

因此，利用這條“語法規則”就可以將“做做看”、“試試看”這種“語法詞”辨認出來（因為“做”、“試”、“看”這些簡單詞的“語意屬性”都是“PROPEL”），同時，能夠得到“做做看”、“試試看”的“語意屬性”都為“PROPEL”。 □

我們可以將例 1 和例 2 中的“語法規則”合併起來，表示如下：

1.  $Y^* \rightarrow Y$  ;
2.  $Y \rightarrow \text{"QUANTITY"} \mid \text{"PROPEL"}$  ;

第二個式子說明 Y 可以用“QUANTITY”或“PROPEL”來取代；“|”這個符號代表“或者”（OR）。

對第二類“語法規則”而言，它所代表的意義是說：幾個擁有語意屬性“Y”的“簡單詞”和一個擁有語意屬性“X”的“簡單詞”，可以將它們合併成一個“語法詞”，而這個“語法詞”的語意屬性將是“X”。第三類“語法規則”和第二類“語法規則”是一種對稱的型式，下面我們對這兩類“語法規則”舉一些例子。

例 3：如果我們將“Y”代換成“QUANTITY”，“X”代換成“TIME” 第二條“語法規則”就可以表示成

” PROPEL ”，而 ”了”、”過”、”著”、”不”的 ”語意屬性” 都是 ” STATE ” )，同時，我們可以進一步得到 ”做了”、”做過”、”做著”、”做不做”、”做過了”、... 這些 ”語法詞” 的 ”語意屬性” 為 ” PROPEL ”。

我們可以將這條 ”語法規則” 表示如下：

- 1 . ( X + Y<sup>\*</sup> )<sup>\*</sup> → X ;
- 2 . X → ” PROPEL ” ;
- 3 . Y → ” STATE ” 。 □

## 第五章 語意匹配

### ( SEMANTIC MATCHING )

在前兩章中我們介紹如何利用“辭典”和“語法規則”來對“簡單詞”和“語法詞”作個別的辨認。這一章中我們將從句子的語意上來考慮如何解決同音詞的混淆現象？

要達到這個目的，一個最好的辦法是對中文句子作自然語言的了解( Natural language understanding )，也就是說，我們可以從所有由同音詞組合而成的句子之中，找出一個最符合語意要求的組合，如果我們能作到這一步，同音詞的混淆現象就能夠被解決了；但是，這種解決方式所要付出的代價非常高，原因是自然語言的了解對於任何一種語言來說都是一個十分困難，而且尚未解決的問題，其次，這種辨認的方式會隨著句子的長度產生組合性的爆炸( Combinatorial explosion )，因此，這種辨認方式實際上是不可行的。

因此，我們退而求其次，希望能夠藉由句子中局部上的語意來處理同音詞的辨認。基本上的做法是：當一個句子中的某一個地方有同音詞的混淆現象時，我們考慮是否有一個同音詞的語意能夠符合它前後文字在語意上的要求？如果存在這樣的一個同音詞，我們就認為它是正確的用詞，這種選取同音詞的方法我們稱之為“語意匹配”( Semantic matching )。“語意匹配”可以應用的原因在於：利用“辭



典”和“語法規則”通常已經能夠決定一個句子中絕大多數沒有混淆現象的詞，因此，在這些詞附近擁有混淆現象的同音詞就可以根據它的語意來作辨認。以下我們將先說明幾個和“語意匹配”有關的概念：“關鍵字”(Key word)和“語格限制”(Case constraints)。

一個常用在人工智慧上處理自然語言的方法是：在一個句子中尋找某一類型的“關鍵字”，並藉此發展出適合的語法或語意上的結構來處理各種不同的工作。

例如 Schank 的“觀念依存理論”(Conceptual dependency theory)和 Fillmore 的“語格文法”(Case grammar) [ 9 ] 是以“動詞”作為關鍵字所發展出來的文法結構，它們的主要目的是用來分析句子的結構，並進一步作句子的了解。

而 Schank 的“腳本”(Scripts) [ 6 ] 則是以“名詞”作為關鍵字所發展出來的文法結構，腳本的主要目的是用來提供某些“名詞”的基本資料。例如在一個描述“餐廳”(Restaurant)的“腳本”中會包括一些可能在餐廳中依次出現的事件，如：找座位、點菜、用餐、付帳、離開、... 等等，所以，當一篇文章中提到“餐廳”這個詞時，系統會找出描述“餐廳”的“腳本”作為背景知識，提供“了解”(Understanding)或“問答”(Question - answer)時的需要。

為了解決同音詞所產生的混淆現象，我們對於“關鍵字”給予如

下的定義。

定義 5.1 在一個句子  $S$  之中，對某一組同音詞  $H = \{ \text{Homonym}(i), i = 1, \dots, k \}$  來說，在這組同音詞  $H$  的“附近”會存在一個詞  $K$  稱為“這組同音詞的關鍵字”（Key word of these homonyms）能夠唯一確定  $\text{Homonym}(j)$ （屬於  $H$ ）是正確的用詞。 □

例 1：在“張三 {再, 在, 載} 學校”這個句子中，{再, 在, 載} 是一組同音詞，它們所對應的注音符號是“ㄖㄨㄞˋ、”，而“學校”是這組同音詞的“關鍵字”；也就是說，我們可以從“學校”的語意來確定該用 {再, 在, 載} 中的哪一個字？在這個例子中應該選取“在”字，實際的匹配過程我們將稍後再作說明。 □

例 2：在“張三吃了一顆 {塘, 糖, 醴}”這個句子中，{塘, 糖, 醴} 是一組同音詞，它們所對應的注音符號是“ㄊㄨㄥˊ、”，而“吃”是這組同音詞的“關鍵字”；也就是說，我們可以從“吃”的語意來確定該用 {塘, 糖, 醴} 中的哪一個字？在這個例子中該選取“糖”字，實際的匹配過程將稍後再作解釋。 □

在此所謂的“關鍵字”是指能夠對一組同音詞作辨認的特殊文字，而不是一般自然語言了解中所考慮的整個句子的關鍵字；所以，如果在一個句子中有兩個地方都擁有同音詞所產生的混淆現象時，這兩組同音詞可能會分別擁有不同的“關鍵字”。考慮下面這個例子：

例 3：“張三 {再，在，載} 學校吃了一顆 {塘，糖，醴}”這個句子中，“學校”還是 {再，在，載} 的“關鍵字”，“吃”還是 {塘，糖，醴} 的“關鍵字”。 □

在上述兩個例子中：例 1 中“關鍵字”的語意單元是屬於第一類中的“LOCATION”，而例 2 的“關鍵字”的語意單元是屬於第二類中的“INGEST”；在說明這兩種情形之前，我們先說明什麼是一個詞的“語格限制”？

定義 5.2 在自然語言中，一種特殊型態的“動詞”或“名詞”通常會需要和某些特定型態的“名詞”結合成固定的語意關係，這種情形一般稱為這個“動詞”或“名詞”的“語格限制”(Case constraints) [3]。 □

我們可以利用“語格限制”來解決一個同音詞的混淆現象。考慮下面這兩種情形：對於第一類的“語意單元”而言，它的“語格限制”用來表示什麼可能會描述這一類的“語意單元”？對於第二類的

“語意單元”而言，它的“語格限制”用來表示什麼會扮演這個“語意單元”的角色？

我們用前面的兩個例子來說明什麼是一個詞的“語格限制”，同時指出“關鍵字”、“語格限制”和“語意匹配”的關係。

例 1：對“LOCATION”這個語意單元來說，通常需要一個詞來表示“任何一個可能位於這個地點的東西”，例如：“AGENT”或“MATERIAL”，另外，“LOCATION”通常還需要一個詞來表示“位於”這個觀念，例如：它的語意單元可以表示為“STATE”；因此，“LOCATION”通常需要兩個“語格”（Case）；一是“AGENT”或“MATERIAL”，一是“STATE”。我們可以將這種“語格限制”表示成下面的型式（也就是一個“語意規則”）：

$$\begin{aligned} \text{"LOCATION"} &\rightarrow \{ x, \text{"STATE"} \} ; \\ x &\rightarrow \text{"AGENT"} \mid \text{"MATERIAL"} ; \end{aligned}$$

因此，對“張三 { 再，在，載 } 學校”這個句子而言，我們可以利用“學校”的語意屬性“LOCATION”的“語格限制”來作同音詞 { 再，在，載 } 的辨認。

如果在“辭典”中關於“ㄉㄞˇ、”這個注音詞的辨認樹如下（為了說明方便，我們用串列表示法來表示一個辨認樹）：

( " 在 " ( " STATE " ( "  $\phi$  " , " 在 " ) )  
 ( " TIME " ( "  $\phi$  " , " 再 " ) )  
 ( " PROPEL " ( "  $\phi$  " , " 載 " ) ) ) .

我們就能夠確定 "在" 字是正確的用詞，因為它的語意屬性是 ( " STATE " , "  $\phi$  " ) 能夠符合 " LOCATION " 的 "語意規則"，而判斷 "再" 字和 "載" 字都不是正確的用詞，因為它們的語意屬性分別是 ( " TIME " , "  $\phi$  " ) 和 ( " PROPEL " , "  $\phi$  " ) ，都不能符合 " LOCATION " 的 "語意規則"。 □

例 2：一般來說 " INGEST " 這個動作需要一個詞來代表 "能吃东西的作用者 ( AGENT )"，例如：" HUMAN " 或是 " ANIMAL "，另外，" INGEST " 這個動作還需要一個詞來代表 "可以吃的東西 ( MATERIAL )"，例如：" FOOD "。所以，從上面的討論中可看出來 " INGEST " 通常需要兩個 "語格" ( Case )：一是 " HUMAN " 或是 " ANIMAL "，一是 " FOOD "。我們可以將這種 "語格限制" 表示成下面的型式：

$$\begin{aligned} \text{" INGEST " } &\rightarrow \{ \text{" AGENT ( x ) " ,} \\ &\quad \text{" MATERIAL ( FOOD ) " } \} ; \\ x &\rightarrow \text{" HUMAN " } | \text{" ANIMAL " } ; \end{aligned}$$

因此，對“張三吃了一顆{糖，糖，糖}”這個句子而言，我們可以利用“吃”字的語意屬性“INGEST”的“語格限制”來作同音詞{糖，糖，糖}的辨認；所以我們能夠確定“糖”字是正確的用詞，因為它的語意屬性是（“MATERIAL”，“FOOD”）能夠符合“INGEST”的“語意規則”，而判斷“糖”字和“糖”字都不是正確的用詞，因為它們的語意屬性分別是（“LOCATION”，“ $\phi$ ”）和（“MATERIAL”，“COMPOUND”），都不能符合“INGEST”的“語意規則”。 □

下面我們說明在整個注音符號句子的轉換過程中，使用“語意規則”會遇到的一些問題。

基本上一個“語意單元”的“語意規則”所考慮的僅是這個“語意單元”在一個句子中“局部上的某些特殊要求”（“語格”），但是，它並沒有限制這些“語格”在句子中出現的先後次序，所以，“語意規則”所能解決的混淆現象和它在句子中出現的先後次序無關。我們看下面這個例子：

例 3：在“{糖，糖，糖}被張三吃掉了”這個句子中，“AGENT（x）”、“MATERIAL（FOOD）”和“INGEST”之間的相對次序都和例 2 有所不同，但是{糖，糖，糖}的辨認方式還是利用到“INGEST”的“語意規則”。 □

由於“語意規則”所著重的是“局部上的語格限制”，所以在一個句子中“語意匹配”時是以最近的一個“語意規則”為準，而不會匹配到較遠的“語意規則”。下面的例子說明了這種情形：

例 4：“張三 { 再，在，載 } 學校吃了一顆 { 塘，糖，醴 }”這個句子中，{ 再，在，載 } 這組同音詞不會選擇“INGEST”（“吃”）的“語意規則”來匹配，而 { 塘，糖，醴 } 這組同音詞不會選擇“LOCATION”（“學校”）的“語意規則”來匹配。 □

## 第六章 使用者意願的混淆現象 和特殊鍵

( THE AMBIGUITY OF USER - WANTED  
AND SPECIAL KEY )

在這一章中將介紹一種能通過語法和語意辨認的同音詞混淆現象，我們稱它為“使用者意願的混淆現象”，同時我們將說明如何來解決這種特殊的混淆現象。

我們先舉幾個例子來說明什麼是“使用者意願的混淆現象”？

例 1：描述數目字的中文字通常有大、小寫之分，如：“一”、“二”、“三”、“四”、“五”、“六”、“七”、“八”、“九”、“十”、“百”、“千”是小寫；“壹”、“貳”、“參”、“肆”、“伍”、“陸”、“柒”、“捌”、“玖”、“拾”、“佰”、“仟”是大寫。一個數目字大小寫的使用完全視使用者的需要而定，無法從前面所說的語法或語意的觀點來辨認，這就是我們所謂“使用者意願的混淆現象”的一個例子。 □

例 2：代名詞的使用也有“使用者意願的混淆現象”。像下面這一個注音符號的句子

” ㄅㄨˋ ㄩˊ ㄆㄨˋ ㄊㄨㄣˋ ㄕㄨˋ ㄨˋ ㄆㄨˋ ”

擁有兩個同音的句子



1 . “ 他 是 學 生 ” 和

2 . “ 她 是 學 生 ”

( “ 他 ” 和 “ 她 ” 的 注 音 符 號 都 是 “ 去 Y ” ) , 而 這 兩 個 同 音 的 句 子 從 前 面 所 說 的 語 法 或 語 意 的 觀 點 來 看 都 是 對 的 。 除 了 這 兩 個 字 之 外 , “ 它 ” 也 和 “ 他 ” 、 “ 她 ” 同 音 。 此 外 “ 你 ” 字 和 “ 妳 ” 字 也 有 類 似 的 情 形 。 □

要 解 決 “ 使 用 者 意 願 的 混 淆 現 象 ” , 我 們 必 須 讓 使 用 者 明 確 的 說 明 他 究 竟 需 要 哪 一 個 字 ? 我 們 的 方 法 是 : 當 使 用 者 在 輸 入 注 音 符 號 的 句 子 的 時 候 , 必 須 打 一 些 特 殊 鍵 用 來 指 出 他 所 想 要 的 字 。 下 面 列 出 幾 個 有 用 的 特 殊 鍵 :

1 . “ C ” ( Captial ) : 代 表 “ 大 寫 鍵 ” , 用 來 指 出 數 目 字 中 的 大 寫 ;

2 . “ S ” ( Small ) : 代 表 “ 小 寫 鍵 ” , 用 來 指 出 數 目 字 中 的 小 寫 ;

3 . “ M ” ( Man ) : 代 表 “ 男 性 鍵 ” , 用 來 指 出 代 名 詞 中 的 “ 他 ” 字 和 “ 你 ” 字 ;

4 . “ W ” ( Woman ) : 代 表 “ 女 性 鍵 ” , 用 來 指 出 代 名 詞 中 的 “ 她 ” 字 和 “ 妳 ” 字 ;

5. "L" (Sexless) : 代表 "中性鍵", 用來指出代名詞中的 "它" 字。

下面舉一些例子說明特殊鍵的用法, 我們用 < C > 代表打 "大寫" 這個特殊鍵, 其他的情形類似。

例 1 : " < C > 《 X ㄨ、 ㄩ 一、 一、 < 一 ㄩ ㄩ ㄩ、  
ㄨ ㄨ、 ” 將得到 "共計壹仟圓整"; 如果將 < C > 改成 < S >  
則會得到 "共計一千元整"。 □

例 2 : " 去 Y < M > 尸、 T ㄩ ㄨ、 尸 ㄨ ” 將得到 "他是  
學生"; 如果將 < M > 改成 < W > 則會得到 "她是學生"。 □

除了數目字和代名詞之外, 還有一種比較複雜的 "使用者意願的  
混淆現象", 我們看下面兩個例子。

例 3 : " 一、 ㄨ 一 ㄩ T 一 ㄩ X ㄩ、 ㄨ ㄨ 去 一 ㄨ 尸 ㄨ、  
ㄩ X、 《 ㄨ、 ㄨ ㄩ、 尸 ㄨ、 ” 這個注音符號的句子, 有兩  
個同音的中文句子

1. " 一 篇 新 聞 張 貼 在 布 告 欄 上 "

和 2. " 一 篇 新 文 章 ( 貼 ) 在 布 告 欄 上 "



為了解決這種“使用者意願的混淆現象”，我們將用到下面這個特殊鍵：

6. ”、” ( Comma ) : 代表前面的注音符號將構成“簡單詞或“語法詞”。

現在我們說明如何利用”、”解決〈例 3〉和〈例 4〉？

例 3： 如果我們要得到

” 一 篇 新 聞 張 貼 在 布 告 欄 上 ”

這個中文句子，就必須輸入 ” 一、 ㄉㄨㄢ ㄊㄨㄢ ㄒㄨㄢ /  
〈、〉 ㄨㄤ ㄍㄨㄢ ㄆㄨ、 ㄅㄨ、 《ㄨ、 ㄉㄨ、 ㄆㄨ、  
” ；

如果我們要得到

” 一 篇 新 文 章 貼 在 布 告 欄 上 ”

這個中文句子，則必須輸入 ” 一、 ㄉㄨㄢ ㄊㄨㄢ ㄒㄨㄢ /  
出 尤 〃 〃 去 一 世 ㄆㄨ、 ㄅㄨ、 《ㄨ、 ㄉㄨ、 ㄆㄨ、  
” 。

例 4： 如果我們要得到

”政府當局不干涉特殊人士的  
言論自由”

這個中文句子，就必須輸入 ” ㄅㄨㄣˇ ㄉㄨㄥˋ ㄉㄨㄥˋ ㄉㄨㄥˋ ㄉㄨㄥˋ  
ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ  
ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ ” ；

如果我們要得到

”政府當局部干涉特殊人士的  
言論自由”

這個中文句子，則必須輸入 ” ㄅㄨㄣˇ ㄉㄨㄥˋ ㄉㄨㄥˋ ㄉㄨㄥˋ ㄉㄨㄥˋ  
ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ  
ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ ㄎㄨㄥˋ ” 。

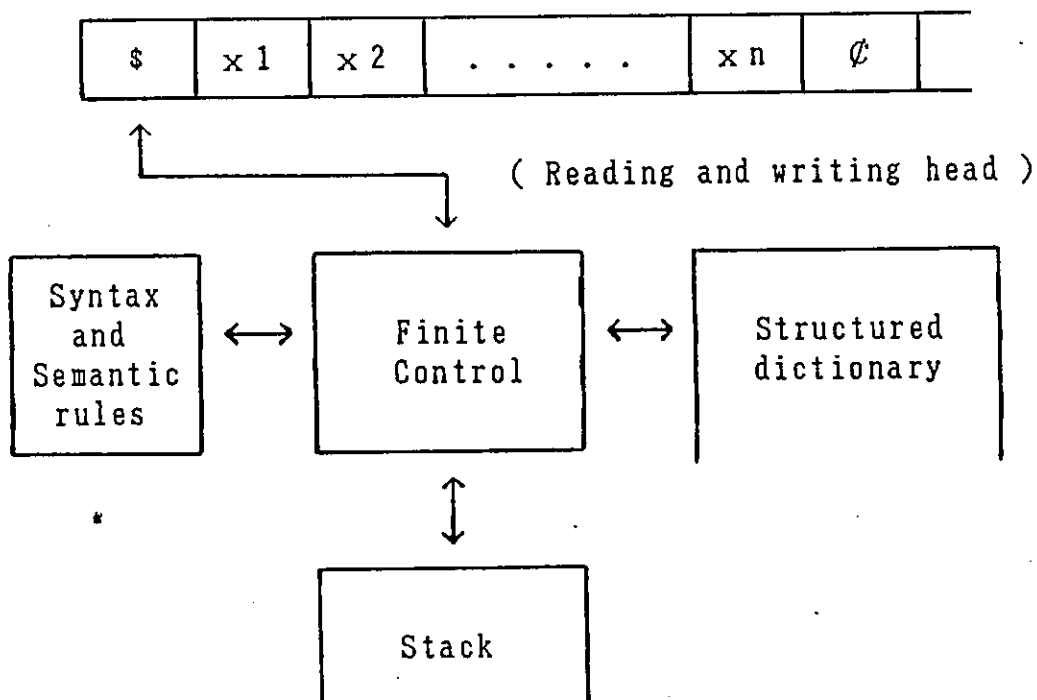
□

# 第七章 辨認機和轉換過程

## ( A DISAMBIGUATION MACHINE AND THE PROCESS OF TRANSLATION )

這一章中將要說明一個能夠轉換注音輸入成為中文輸出的轉換裝置，以及它的轉換過程。

我們用〈圖 - 5〉來說明一個辨認注音輸入的轉換裝置( Disambiguous Chinese phonetic input device ) :



〈圖 :- 5〉

<圖 - 5> 中最上面是一個表示輸出 / 入的帶子 ( Tape )，其中  $S = x_1 x_2 \dots x_n$  代表使用者所輸入的注音符號和特殊鍵的句子， $\$$  和  $\phi$  分別代表左、右邊界的特殊符號， $x_i$  代表一個注音符號或是特殊鍵的符號；中間是一個控制單位 ( Finite Control )，它利用 "結構化的辭典" ( Structured dictionary )、"語法及語意規則" ( Syntax and Semantic rules ) 和 "堆疊" ( Stack ) 來完成輸出 / 入 ( 轉換 ) 的工作。

這個辨認注音輸入的轉換裝置實際的轉換過程如下：

STEP 1 : 系統在 "起始狀態" ( Initial State )，"讀寫頭" ( Reading / writing head ) 指在帶子最左邊的特殊符號  $\$$ ；這時所有的 "堆疊" 都是 "空的" ( Empty ) ( i.e.  $St(i) = \lambda$ ， $i = 1, 2, \dots, n$  )。

STEP 2 : 系統將 "讀寫頭" 逐漸往右移動，同時利用 "辭典"、"語法規則" 和 "往覆剖析" ( Backtract parsing ) 的方式將  $S$  轉換成為  $S$  的同音詞結構  $S'$ ， $S' = x'^1 x'^2 \dots x'^k$ ，其中  $x'^i$  代表一個 "簡單詞" 或是 "語法詞" ( 包括它們的語意屬性 )， $i = 1, \dots, k$ ；如果在  $x'^i$  中有一組同音詞  $\{ y_{i,1}, y_{i,2}, \dots$

$y_{i,m}$  } , 就將所有的同音詞  $\{y_{i,j}, j = 1, \dots, m\}$  和它們的語意屬性放進相對應的“堆棧”  $St(i)$  中, 直到“讀寫頭”指到帶子最右邊的特殊符號  $\phi$  為止。

STEP 3 : 這時系統檢查是否存在一個  $i$ , s.t.  $St(i) \neq \lambda$  (也就是表示它有同音詞的混淆現象)? 如果  $St(i) = \lambda, i = 1, 2, \dots, k$ , 系統就將  $x^i, i = 1, 2, \dots, k$ , 中的中文詞輸出到帶子上, 同時結束轉換的過程。否則繼續執行 STEP 4。

STEP 4 : 系統先將“讀寫頭”重新指到帶子最左邊的特殊符號  $\$$ , 然後將“讀寫頭”右移到  $St(i) \neq \lambda$  的位置, 從它左右最近詞的“語意單元”的“語格限制”來作“語意匹配”, 將匹配成功的詞  $y_{i,r}$ ,  $y_{i,r}$  屬於  $\{y_{i,j}, j = 1, \dots, m\}$ , 放進  $x^i$  中, 同時令  $St(i) = \lambda$ ; 執行這個步驟直到“讀寫頭”再次指到帶子最右邊的特殊符號  $\phi$  為止。

STEP 5 : 將  $x^i, i = 1, 2, \dots, k$ , 中對應的中文詞輸出到帶子上, 結束整個轉換的過程。

□



從上面執行的步驟中，我們可以認定這個辨認注音輸入的轉換裝置實際上是一個“零儲存的線性限制自動機”(Empty store linear bounded automaton) [ 10 ]。

定義 7.1 一個辨認機( The disambiguation machine )  $D_m$  是一個“零儲存的線性限制自動機”( Empty store linear bounded automaton )，它可以表示成如下的六項( Tuple )：

$$D_m = ( K , \Sigma , \Gamma , \delta , q_0 , F )$$

$K$  代表所有的狀態 ( States )，

$$K = \{ q_0 , q_1 , q_2 , q_3 , q_4 , q_5 \}$$

$q_0$  = 起始狀態 ( Initial state )；

$q_1$  = 在“辭典”中搜尋一個“簡單詞”的狀態( The state for searching a simple phrase patterns )；

$q_2$  = 利用“語法規則”合併出一個“語法詞”的狀態( The state for merging a syntactic phrase patterns )；

\*  $q_3$  = 處理 Stack 運算的狀態( The state for manipulating stack operations )；

$q_4$  = 利用“語意規則”處理同音詞混淆現象的狀態( The state for performing the semantic

patterns matching );

$q_5$  = 結束狀態 ( Final state );

$\Sigma$  代表輸入的符號 ( Input symbols ), 包括: 注音符號和代表特殊鍵的符號。

$\Gamma$  代表帶子上的符號 ( Tape symbols ), 包括: 中文字、語意屬性和  $\Sigma$  中的符號。

$\delta$  代表下一次移動的函數 ( Next move function )

$\delta : K \times \Gamma \longrightarrow$

the set of  $K \times \Gamma \times \{ L, R \}$  。

$q_0$  代表起始狀態 ( Initial state )。

$F = \{ \lambda \}$ ,  $\lambda$  代表零儲存 ( Empty store )。

而這個辨認機所能辨認的語言 ( Language )  $L$ ,

$$L = \{ w \mid w \text{ is in } (\Sigma - \{ \$, \emptyset \})^* \text{ and} \\ (q_0, \$ w \emptyset) \vdash^* (q, F) \text{ for} \\ \text{any } q \text{ belongs to } K \} \quad \square$$

下面我們舉一個例子來說明轉換的過程。

假設在“辭典”中存有下面這些“注音符號詞”的辨認樹(為了說明方便,我們用串列表示法來表示一個辨認樹):

1. ( " 出 尢 厶 弓 " ( " AGENT " ( "  $\phi$  " , " 張 三 " ) ) ) .

2. ( " 彳 " ( " INGEST " ( "  $\phi$  " , " 吃 " ) ) ) .

3. ( " 攵 ㄗ " ( " STATE " ( "  $\phi$  " , " 了 " ) ) ) .

4. ( " 一 丶 " ( " STATE " ( "  $\phi$  " , " 亦 " ) ) ( " CONCEPT " ( "  $\phi$  " , " 義 " ) ) ( " QUANTITY " ( " S " , " 一 " ) ( " C " , " 壹 " ) ) ) .

5. ( " ㄗ ㄗ " ( " UNIT " ( " CONCEPT " , " 科 " ) ( " MATERIAL " , " 類 " ) ( " PLANT " , " 棵 " ) ) ) .

6. ( " 去 尢 丿 " ( " MATERIAL " ( " FOOD " , " 糖 " ) ( " COMPOUND " , " 醴 " ) ) ( " LOCATION " ( "  $\phi$  " , " 塘 " ) ) ) .

其中的“語法規則”如下：

1. ( X + Y<sup>\*</sup> )<sup>\*</sup> → X ;

X → " INGEST " ;

Y → " STATE " ;

2. ( Y<sup>\*</sup> + X )<sup>\*</sup> → X ;

X → " UNIT " ;

Y → " QUANTITY " ;

其中的“語意規則”如下：

1. " INGEST " → { " AGENT ( x ) " ,  
" MATERIAL ( FOOD ) " } ;

2. x → " HUMAN " | " ANIMAL " ;

假設使用者輸入的句子是“< S > 出た ムヲ 介 カセ

一、 カセ 去たノ”，它的整個轉換的過程如下：

STEP 1 : ( Initial state = q 0 )

S = x 1 x 2 x 3 x 4 x 5 x 6 x 7 x 8 ;

x 1 = < S > , St( 1 ) = λ ;

x 2 = 出 尢 , St( 2 ) = λ ;

x 3 = 厶 弓 , St( 3 ) = λ ;

x 4 = 彳 , St( 4 ) = λ ;

x 5 = 尗 丩 , St( 5 ) = λ ;

x 6 = 一 丩 , St( 6 ) = λ ;

x 7 = 彳 丩 , St( 7 ) = λ ;

x 8 = 去 尢 丩 , St( 8 ) = λ ;

STEP 2 : ( q 1 , q 2 and q 3 )

S ^ = x ^ 1 x ^ 2 x ^ 3 x ^ 4 ;

x ^ 1 = ( " AGENT " ( " φ " , " 張三 " ) ) ;

( ∴ " 辭典 " 中的第一個串列 )

St( 1 ) = λ ;

\* x ^ 2 = ( " INGEST " ( " φ " , " 吃了 " ) ) ;

( ∴ " 辭典 " 中的第二、三個串列和第一條 " 語法規則 "

)

St( 2 ) = λ ;

$x^3 = ?$

$St(3) = \{ ( " UNIT "$   
                  ( " CONCEPT " , " 一科 " )  
                  ( " MATERIAL " , " 一顆 " )  
                  ( " PLANT " , " 一棵 " ) ) \}

(  $\because$  "辭典" 中的第四、五個串列和第二條 "語法規則"  
                  )

$x^4 = ?$

$St(4) = \{ ( " MATERIAL "$   
                  ( " FOOD " , " 糖 " )  
                  ( " COMPOUND " , " 糖 " ) )  
                  ( " LOCATION "  
                  ( "  $\phi$  " , " 塘 " ) ) \}

(  $\because$  "辭典" 中的第六個串列 )

STEP 3 : ( q3 )

\*  $St(3) \neq \lambda ; St(4) \neq \lambda ;$

STEP 4 : ( q4 )

$S = x^1 x^2 x^3 x^4 ;$

$x^1 = ( " AGENT " ( " \phi " , " 張三 " ) ) ;$

$St(1) = \lambda ;$

$x^2 = ( \text{" INGEST " } ( \text{" } \phi \text{" } , \text{" 吃了 " } ) ) ;$

$St(2) = \lambda ;$

$x^3 = y_{3,2}$

$= ( \text{" UNIT "}$

$( \text{" MATERIAL " } , \text{" 一顆 " } ) ) ;$

$( \because \text{" INGEST " 的 "語意規則" } )$

$St(3) = \lambda ;$

$x^4 = y_{4,1}$

$= ( \text{" MATERIAL "}$

$( \text{" FOOD " } , \text{" 糖 " } ) ) ;$

$( \because \text{" INGEST " 的 "語意規則" } )$

$St(4) = \lambda ;$

STEP 5: ( Final state =  $q_5$  )

$S^{\wedge} = \text{" 張三吃了一顆糖 "}$ 。

□

## 第八章 應用範圍和學習能力

### ( APPLICATION DOMAIN AND LEARNING ABILITY )

一個電腦程式是否能夠具有“學習”( Learning )的能力，是近年來在人工智慧上最重要的研究領域之一。各個研究者對於“學習”的定義通常都有他們個人的看法，其中 Simon [ 11 ] 認為“一個系統改進它本身執行能力的任何一個過程”( Any process by which a system improves its performance ) 都可以稱之為“學習”，一般認為是對“學習”最淺顯的一個定義。

對於前面所提到的中文輸入系統而言，我們認為“學習”就是：如何來增加系統所能處理的文字應用範圍？為了解決這個問題，我們將在這一章中提出一種“交談式的學習方式”( Learning by interactive )，也就是說，由使用者教系統認識新的字、詞或字、詞的語意屬性，使系統能夠符合使用者個人應用上的需要。下面我們將依次討論下面這些問題：一、什麼時候系統需要“學習”？二、人、機如何交談？三、“學習”會如何影響系統中原有的“辭典”安排？

首先，我們回答第一個問題：什麼時候系統需要“學習”？一般需要“學習”的情況可能有以下幾種：

- 一、系統在轉換的過程中找不到使用者所輸入的注音符號、或



是在語意匹配的時候找不到能夠匹配的語意屬性：例如在“辭典”中根本沒有“去犬ノ”這個注音符號和它的辨認樹，那麼我們的系統當然沒有辦法做轉換的工作。因此，系統就必須要求使用者告訴它這些訊息。

二、輸出的中文句子有混淆現象：這種情形通常發生在某一些詞有了新的用法，而是以前所不能處理的。因此，遇到這種情形也必須對“辭典”作適當的修改。

三、系統應用範圍的改變：對任何一個知識庫系統（Knowledge base system）而言，系統應用範圍的“穩定性”（Stability）都是一個相當重要的考慮因素 [ 11 ]，原因是“過期”的資料通常都不能夠使用，而且它會造成儲存空間上的浪費，進而影響到系統搜尋正確資料的速度，在這種情形之下，系統是否具有“學習”的能力來更新（Update）它的資料以符合環境的變遷就顯得相當重要了。以我們的系統為例：如果系統原來所處理的範圍主要是關於電腦資訊方面的文書工作，那麼一些關於電腦資訊的專門字彙，如：“輸入”、“輸出”、“記憶體”、“終端機”、“磁帶”、...，都必須建在“辭典”中，但是，如果系統要應用到關於化學領域的文書工作，那麼一些關於化學的字彙，如：“氧化”、“還原”、“甲苯”、“分子”、“原子”、...，就必須新加進“辭典”中，這就有賴於系統是否具有“學習”的能力來處理這種問題。另外一個會隨之產生的問題是：當

系統遷移到關於化學領域的文書工作之後，可能原來關於電腦資訊的字彙都用不到了（或是使用的比例相當小），因此，我們希望把關於電腦資訊的字彙“忘掉”來增加系統的效率（Performance），針對這個問題的一個可行的解決辦法是：把這些特殊範圍的文字放進輔助記憶體（如“磁帶”或“磁碟”）中，直到有使用上的需要時才把它們再叫進來。

其次的問題是使用者如何和系統做“交談”？下面我們將舉一些情況來說明交談的步驟，和對系統原來結構的影響。

一般在實際使用這種注音符號輸入法時，當一個注音符號句子轉換完成之後，系統會將整個中文句子列印出來供使用者“校對”，如果使用者發現有輸入錯誤，或是輸出的中文句子中還是有同音字，的情形發生時，他可以拒絕接受這個句子，同時要求重新輸入，或是更改某些字；如果情形屬於後者，系統就先檢查在“辭典”中是否有這些字所構成的詞？如果“辭典”的回答是“NO！”，系統就會要求使用者教它這個詞和所擁有的語意屬性是什麼？例如系統並不知道“李四”是一個人名，就有可能造成辨認上的錯誤，當使用者更改這兩個字之後，系統和使用者之間就會產生如下的問答：

S（代表“System”）：“李四”是一個詞嗎？

U（代表“User”）：Yes .

S：“李四”屬於下面那一類？

AGENT	EVENT	LOCATION	MATERIAL	TIME
CONCEPT	STATE	QUANTITY	UNIT	MOVE
MTRANS	PROPEL	SPEAK	INGEST	

U : AGENT .

如果在“辭典”中沒有其他“ㄉㄨㄣˇ ㄉㄨㄣˇ”的同音詞，系統就結束問答，然後在“辭典”中將“李”字和“四”字連起來，並給予它的“語意屬性”——（“AGENT”，“ $\phi$ ”）；如果在“辭典”中存在其他的同音詞，系統就會將“ㄉㄨㄣˇ ㄉㄨㄣˇ”的辨認樹列印出來供使用者參考，是否需要進一步用“語意子類”來分辨？如果系統本身就有這個詞，而是辨認上發生錯誤，這時就必須考慮這個詞是不是有了新的“用法”？或是原來的“語意屬性”不正確？如果是前者，就依照前面的交談過程學習新的“用法”，也就是其他的“語意屬性”，再存在“辭典”中適當的位置；如果情況屬於後者，我們就經由交談的過程來修改它的“語意屬性”。

在這一章中我們所謂的“交談”，基本上只是讓使用者能夠明確的將新的字、詞及語意屬性告訴系統，而系統可以應使用者的要求將“辭典”中部分的資料內容列印出來而已；我們並沒有牽涉到太複雜的交談方式及交談內容。

## 第九章 結 論 ( CONCLUSIONS )

在這篇論文中，我們對於中文注音符號輸入法的同音字混淆現象提出了一個完整解決的方法。我們一方面從一些自然語言上的觀點，例如：“語法辨認”和“語意匹配”，來討論如何作同音詞的辨認，並藉此發展出一個辨認注音輸入的自動轉換裝置；另一方面，我們也從實際上的需要，例如：“辭典”的儲存空間、搜尋速度上的考慮，討論了“辭典”的安排。

在此我們並沒有提出一個實驗上的數據，以顯示這種輸入方式究竟能夠達到多高的正確轉換比例；但是，從 Knuth 的 80 / 20 Law 和進一步語法、語意上的考慮來觀察，我們相信這種輸入方式一定會有相當良好的效果，同時能夠減少輸入所花費的時間。

這種注音符號的輸入方式真正在實際應用上，還有一些值得我們深入研究的問題。

例如：一個注音符號的句子會需要相當長的時間來完成它的轉換過程，因為它必須檢查“辭典”、利用“語法規則”來作“語法辨認”，還要利用“語意規則”來作“語意匹配”，所以，可能不適合在“線上 ( On \_ line )”即時作業。除此之外，因為“語意匹配”局限於一個句子中局部上的語意，而缺乏對整個句子意義的掌握，因此可能會喪失某些有用的訊息，進而產生少數的同音字混淆現象，所以

，在一個句子轉換完畢之後，人工校對的工作還是需要的；我們認為：如何在一個句子中“局部上的語意”和“整體的意義”兩者之間作一個適當的考慮，會是解決這個問題的一個研究方向。

參 考 文 獻 ( REFERENCES )

- [ 1 ] 戴建耘 , " 淺介中文系統的 / 字碼 / 輸入法 / 字形製作 ,  
" 微電腦時代 , 1987 , 三月號 , pp. 34 - 40 .
- [ 2 ] 劉芬 , " 突破輸入瓶頸 — 改進注音符號輸入法鍵盤配置 ,  
" 資訊與電腦 , 1987 , 三月號 , pp. 99 - 101 .
- [ 3 ] R. M. Weischedel , " Knowledge Representation and  
Natural Language Processing , " Proceedings of the IEEE  
, Vol. 74 , NO. 7 , pp. 905 - 920 , July 1986 .
- [ 4 ] D. E. Knuth , " The Art of Computer Programming Vol. 3  
, " Addison - Wesley , Reading , Mass. , pp. 397 - 398  
, 1973 .
- [ 5 ] J. J. Kuo , J. H. Jou , M. S. Hsieh and F. Maehara , "  
The Development of New Chinese Input Method - - -  
Chinese Word - String Input System , " Proceedings of  
International Computer Symposium 1986 , Tainan , Taiwan  
, R. O. C. , pp. 1470 - 1479 .
- [ 6 ] R. C. Schank and R. P. Abelson , " Scripts , Plans ,  
Goals and Understanding , " Lawrence Erlbaum  
Associates , Hillsdale , New Jersey , 1977 .

- [ 7 ] Y. R. Chao , " A Grammar of Spoken Chinese , " translated into Chinese by P. S. Ding , Chinese University Publishing Co. , Hong Kong , 1982 .
- [ 8 ] E. Horowitz and S. Schni , " Fundamentals of Data Structures , " Computer Science Press , Potomoc , Maryland , 1976 .
- [ 9 ] C. Fillmore , " The Case for Case , " in Universals in Linguistic Theory , E. Bach and R. T. Harms ( Eds. ) , Holt , New York , 1968 .
- [ 10 ] J. E. Hopcroft and J. D. Ullman , " Formal Languages and Their Relation to Automata , " Addison - Wesley , Reading , Mass. , 1969 .
- [ 11 ] A. Barr and E. A. Feigenbaum , " The Handbook of Artificial Intelligence , Vol. 1 , " William Kaufmann , Inc. , Los Altos , Ca. , 1982 .