應用構詞法則與類神經網路於中文新詞萃取

梁婷，葉大榮

國立交通大學資訊科學系

Page 21 ~ 40

Proceedings of Research on Computational Linguistics Conference XIII (ROCLING XIII)

Taipei, Taiwan

2000-08-24/2000-08-25

應用構詞法則與類神經網路於中文新詞萃取

梁婷　　葉大榮

國立交通大學資訊科學系

新竹市 中華民國

email:tliang@cis.nctu.edu.tw

Fax: 886-3-5721490

摘要

中文自然語言的應用近年來越來越受到重視，例如中英翻譯、文件辨識等系統。在這些應用系統中，詞庫扮演著非常重要的角色。然而，新詞不斷的產生，會影響以詞庫為基礎的應用系統效能。因此在本論文裡，我們將建構一個二階段新詞萃取機制。在第一階段利用構詞學的原理建立三音詞萃取法則用以萃取三音詞，再以非詞彙篩檢法則來過濾掉非詞彙字組以減少第二階段的分辨量。第二階段的則以詞組間的特徵統計資訊，利用類神經網路作新詞的進一步的辨認。從實驗的結果可知我們所設計的篩檢與萃取法則將可迅速地萃取新詞。此外我們並探討特徵資訊的選取與多寡對作新詞的辨認成效影響。

1. 緒論

在許多以自然語言處理為導向的文件擷取系統，常是以詞庫來輔助系統，以提升效率。然而藉由人們的使用需求，新詞將不斷地產生、增加，因此新詞的萃取技術發展也就日顯重要。

以中文語料而言，新詞的萃取方法多與斷詞程序相連結。先將語料作切割再合併切割過短的字組形成長字組，經由篩選機制過濾掉可能非為詞彙的字組，最後再對有疑義的字組予以處理。目前所發展的技術可分為統計式與法則式。

統計式的方法多是利用語料中詞彙的組成或特徵資訊，並以統計法則計算作為萃

取原則。例如，相關度(association) [Sproat90]是以相互關連度(mutual information) [Church90]為基礎，並加入字元出現順序性的考量，用來衡量字組中字元與字組間的相關程度。骰子矩陣(dice matrix) [Smadja93, 96]亦是以相互關連度為基礎，改進字組的組成字元出現機率都很低的時候，相互關連度值會過大的問題。Smadja 等人利用骰子矩陣進行連字(collocation)的抽取與兩種語言間連字的翻譯。相對頻率(relative frequency) [Wu93]是將字組出現頻率正規化的統計式特徵，利用可能度比率模組結合相互關連度與相對頻率，作英文的複合詞萃取。熵(entropy) [Tung94]可用來考量字組與語料中相鄰字元間的相關程度，Tung 利用熵作新詞萃取，並應用到文字辨識系統中。這些統計式的方法多以門檻值來判讀詞或非詞彙。雖然，詞的特徵資訊統計值，如相關度、熵等，通常都較非詞彙的統計值高，但是統計值高的卻不一定是詞。因此 Wu[93]建立可能度比率模組，結合不同的特徵統計資訊，以考量字組是為詞彙或非詞彙的機率。Chang[97]除建立可能度比率模組，更進一步結合斷詞程式作遞迴式的中文新詞的萃取。

另一方面，法則式的萃取模組則是利用構詞學與構句學的理論，配合語意資訊或詞性進行萃取，例如詞性標籤(part of speech tag)等。Yeh[91]利用馬可夫模式斷詞，再使用語意與語法的分析選取最適當的斷詞。Lin[93]先作斷詞然後利用構詞學的法則修正斷詞的結果，並以可能度比率模組來萃取新詞，增加斷詞與辨詞的效能。Nie[95]使用 maximum-matching 與經驗法則來作斷詞。將斷詞的結果再利用構詞學的方法萃取三音詞，並且利用構詞學中不具語意功能的字元刪除候選字組。Chen[97]利用詞性標籤建立法則，並利用一部份已經切割好的語料作為訓練資料，用以挑選法則。

有別於上述的萃取技術，本篇論文將利用構詞法則和字組間的特徵統計值，直接從字組庫中而不考量斷詞程序來萃取出新詞。我們主要針對二、三音詞作萃取對象並希望能快速地將其挑選出。在本論文裡，我們將建構一個二階段新詞萃取機制。在第一階段利用構詞學的原理建立三音詞萃取法則用以萃取三音詞，再以非詞彙篩檢法則來過濾掉非詞彙字組以減少第二階段的分辨量。第二階段的則以詞組間的特徵統計資訊，利用類神經網路作新詞的辨認。從實驗的結果可知我們所設計的篩檢與萃取法則

將可迅速地萃取新詞；而不同的特徵資訊與多寡也將影響類神經網路作新詞的辨認成效。

本篇論文除了緒論外第二節將介紹所提的系統概觀。第三節將介紹法則式辨認模組及一些構詞學的基本原理，包括詞的主要構成方式，與詞的一些特性。訂定詞彙萃取法則，與非詞彙辨認法則，進行詞彙的萃取與非詞彙字組的辨認，並作實驗與分析。第四節將描述我們所應用的類神經網路辨認模組及分析統計式特徵，並以實驗結果作驗證。

2. 系統概觀

本系統主要包含兩個模組，一是法則式辨認模組，另一是類神經網路辨認模組，如圖2-1。首先我們為了提供類神經網路辨認模組統計式特徵，先計算字組的統計式特徵，建立特徵資料庫，然後利用系統詞庫將已知詞從字組庫中去除。在法則式的辨認模組中，我們利用構詞學的原理訂定三音詞萃取法則，然後來萃取三音詞，再利用非詞彙辨認法則來過濾屬於非詞彙的雙字組與三字組，經過法則式辨認模組之後，尚有一些無法決定是屬於詞彙或是非詞彙的字組，再交由類神經網路辨認模組結合統計式特徵資訊來作最後的判斷屬於詞彙或非詞彙。

3. 法則式辨認模組

3-1 詞的基本定義與構成

由於有些字組是屬於詞彙或是非詞彙，若我們不加以明確的定義，則難以評估系統的效能。因此我們參考語言學與中央研究院資訊科學研究所中文詞庫小組（以下簡稱詞庫小組）訂定的分詞標準，對詞加以定義。

在構詞學中詞素(morpheme) 是語言系統中具有語意或語法功能的最小的單位[Thompson 92]。有些詞素可以獨立而自由使用，如中文的『我』、『你』、『人』等等。這些稱為『自由詞素(free morpheme)』。 不加任何附著詞素的自由詞素稱為『詞根(root)』。而有些詞素則永遠不可以單獨使用，稱為『附著詞素(bound morpheme)』。附

著詞素也叫做詞綴(affix)，附在詞根前面的叫做『前綴』(或稱『詞頭』prefix)，例如『可微分』,『可』即屬於前綴;附在詞根後面的叫『後綴』(或稱『詞尾』suffix)，例如『正規化』,『化』即屬於後綴;加插在詞根中間的稱為『中綴』(或稱『詞嵌』infix)。

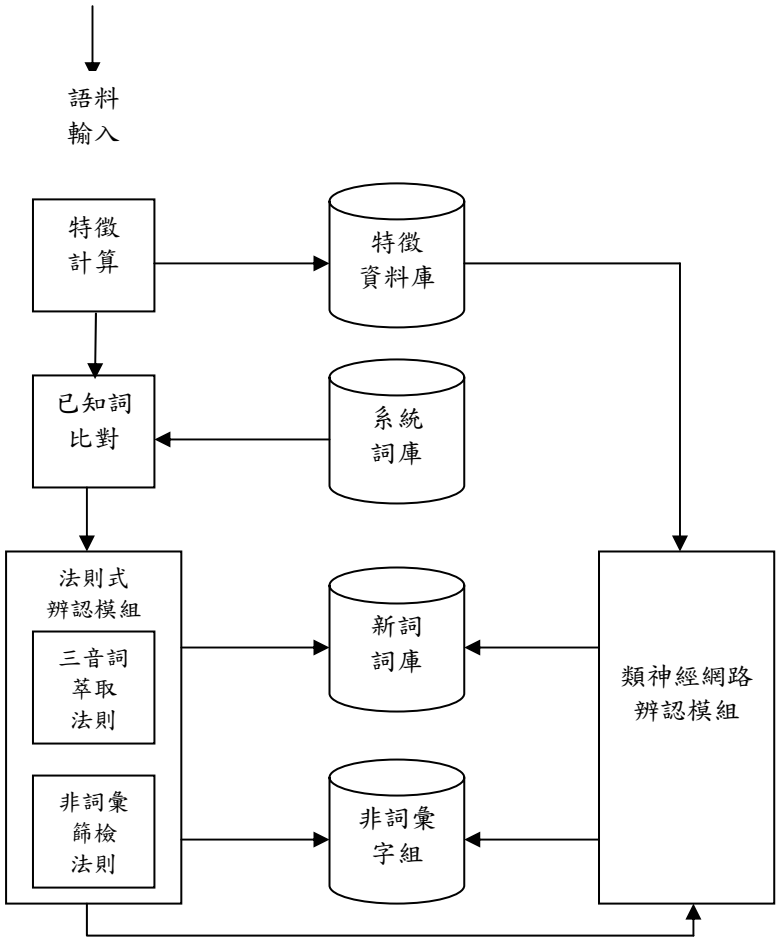


圖 2-1：系統流程圖

根據中文詞界研究與資訊用分詞標準中對詞的定義[詞庫小組]，詞為『一個具有獨立意義，且扮演特定語法功能的字串應視為一個詞』。根據詞性分類，則詞可以大略分為動詞、名詞、形容詞、副詞、定詞、量詞、介詞、方位詞、連接詞、語助詞等類。其中，動詞與名詞因為可具有詞組形式，所以有複合詞的認定問題。並且動詞、名詞是屬於『開放性詞集』[謝國平 86]，可能有新詞產生，因此在認定上困難度較高。『開放式詞集』是指可能會有新詞產生的詞集稱之，而『封閉性詞集』則是幾乎不可能有新詞產生的詞集。

詞的構成方式有許多種，比較重要的有『衍生(Derivation)』與『複合(Compounding)』兩種構詞方式[謝國平 86]。『衍生』是以衍生詞綴與詞根組合而成衍生詞的過程，例如『工業化』、『可微分』皆是衍生詞。『複合』則是指兩個詞併在一起構成另一個詞的過程，例如『遊戲』與『樹』是兩個詞，新詞『遊戲樹』可由此二詞合併得到。除了『衍生』與『複合』之外，詞的構成方式尚有許多，以下列舉幾種『略語(Acronym)』、『溶合(Blending)』、『反向構詞法(Back-formation)』、『借字(Borrowing)』、『簡縮(Abbreviation)』…等[謝國平 86]。

對於難以決定該歸於複合詞或是片語的詞組，我們依據中文詞界研究與資訊用分詞標準中所定的兩條基本原則與六條輔助原則加以分類[詞庫小組]。此外，我們對出自外來語翻譯的新詞認定，一方面考慮其在漢語的語意與語法，另一方面亦考慮原文的語意與語法，例如『能隙』(energy gap)一詞，每次在語料庫中出現都是其他詞的部分字組如『光能隙』(optical energy gap)，由於『能隙』明顯具有獨立的語意，因此認定為新詞。又例如『直方』每次在語料庫中出現都是『直方圖』(histogram)的部分字組，在原文中 'histo' 此字串在原文中並不是一個詞，而是一前接詞綴，有組織的意思，且『直方』在漢語中並無獨立的語意與語法，因此認定為非詞彙。對於化學式構成的新詞認定方面，我們將化學式視為不可切分的單位，因此化學式的部分字組一律視為非詞彙。

3-2 萃取法則

由於一般二音詞可以容易地從統計資訊萃取出來[Sproat90]，因此，本法則模組主要是針對新詞的三音詞部分提出萃取法則。萃取法則主要是依據衍生式構詞與複合式構詞，將我們所切分出來的字組中符合詞與複合詞頭、複合詞尾、衍生前綴、衍生後綴的字組視為詞。由於有一些組成詞素是屬於自由詞素或是附著詞素，各文件上的定義有所出入，但其與詞的組合都可歸為衍生詞或複合詞。

根據[詞庫小組]所附的語法詞綴、衍生詞綴、接頭/接尾詞一覽表，衍生前綴與接頭詞共有五十個，衍生後綴與接尾詞共有四百五十七個。由於我們使用詞彙萃取法則

的目的是想快速的將易於辨認出屬於詞彙的字組，所以我們觀察語料中經常出現的複合詞，挑選適合的衍生前綴與接頭詞有『主』、『副』、『非』、『多』、『超』、『子』、『單』、『雙』等共八個。挑選衍生後綴與接尾詞有『化』、『性』、『度』、『機』、『器』、『法』、『式』、『率』、『值』、『體』、『表』、『型』、『量』、『集』、『圖』、『碼』等共十六個。我們的詞彙萃取法則可以定義為以下兩條法則:

(三音詞萃取法則一):

若三字組($c_1c_2c_3$)，其中 c_1c_2 屬於雙音詞且 c_3 屬於衍生後綴或接尾詞者，並且三字組($c_1c_2c_3$)出現於語料中，其部分字組不得每次與相鄰字元構成雙音詞或三音詞。

(三音詞萃取法則二):

若三字組($c_1c_2c_3$)，其中 c_2c_3 屬於雙音詞且 c_1 屬於衍生前綴或接頭詞者，並且三字組($c_1c_2c_3$)出現於語料中，其部分字組不得每次與相鄰字元構成雙音詞或三音詞。

3-3 篩檢法則

詞常常因為語法功能相同而分為好些詞集，例如『開放性詞集』和『封閉性詞集』。『開放性詞集』包涵名詞、動詞、形容詞、及副詞，新詞往往出於此類 [謝國平 86]。反之『封閉性詞集』包涵介詞、連詞、冠詞等。這些詞類幾乎不會有新詞產生。以我們將字組分類來說，開放性詞集有可能與其他詞素結合成為新詞，封閉性詞集由於具有較固定的語法功能，與其他詞素構成新詞的機率較低。雖然，我們將詞分為『開放性』與『封閉性』，但並不保證封閉性的詞集不會有新詞產生。我們可利用封閉性詞集與其他詞素結合產生新詞機率較低的特性，加以訂定非詞彙字組的篩檢法則。

首先我們定義『封閉字集』以利於我們進行將字組篩檢。對於單字詞素能與其他詞素構成新詞的機率很低者我們稱為『封閉字』。『封閉字』的集合為『封閉字集』。我們所挑選的封閉字不一定來自封閉詞集，而是依據以下原則：

（封閉字挑選原則一）必須與其他詞素構成新詞的機率很低者。

（封閉字挑選原則二）挑選語意與語法功能簡單固定者。

（封閉字挑選原則三）盡量挑選出現頻率高者，因為這樣才能將較多的字組歸類。

原則一是我們挑選封閉字的最主要依據，因為若一中文字與其他詞素構成新詞的機率低，才符合封閉性的原則;原則二是因為語法與語意功能具有多種用法者，由其所產生新詞的機率不一定低;原則三是基於效率的考量，相對而言，通常出現頻率高者，有較多的字組可依照非詞彙篩檢法則將之歸為非詞彙。

根據上述三個原則，我們挑選的封閉字有『和』、『與』、『或』、『且』、『及』、『而』、『此』、『本』、『是』、『其』、『了』、『的』、『之』、『於』、『為』等十五個字。

首先『和』、『與』、『或』、『且』、『及』，這些字的詞性都是屬於連接詞，而且語法功能都相當明確且簡單。由連接詞的語法功能可知這些字會符合原則一與原則三。然而單字詞可能有兩種以上的語意或語法功能，例如『暖和』、『或然率』、『苟且』等，但這些情形大部分是詞庫中已經存在的詞，或是出現的機率很低。

『而』、『此』的語法及語意功能較為固定。『本』則有較多語意上的變化如『樣本數』、『超本文』(超本文會根據衍生詞構詞規則，歸為詞彙)，然而，『本』在我們的語料中單字的出現頻率是屬於出現頻率高的單字，且主要仍以『冠詞』的詞性出現，因此將『本』加入封閉字集中。『是』在我們語料中單字出現頻率是屬於高出現頻率單字，且語意語法固定，所以加入封閉字集。『其』的詞性歸於代名詞，之外用法『其他』、『其餘』。由於『其』的語意語法固定，與其他詞素結合為新詞的機率低，所以亦加入封閉字集。至於『了』我們視為構形詞綴如在『吃了』，並不改變語義。而『的』、『之』可視為修飾語與中心語之間的分隔標記且其使用頻率上高所以亦加入封閉字集。『於』、『為』則視為介詞。我們基於易於處理一律將『動詞＋於』、『動詞＋為』、『動詞＋成』視為非詞彙。

因此我們將字組歸為非詞彙的法則，即

(非詞彙字組篩檢法則一):

字組中包含{和、與、或、且、及、而、此、本、是、其、了、的、之、於、為}者則歸為非詞彙。

除了以封閉字將非詞彙字組篩檢出來以外，我們又使用另一簡單而有效率的非詞彙篩檢法則稱為部分詞彙篩檢法則：

***(非詞彙字組篩檢法則二)**:*

若一字組在語料庫中每次出現的情形，其部分字組皆與相鄰字元形成雙音詞或三音詞者，則歸為非詞彙。

3-4 實驗與分析

我們所使用的語料庫是由交通大學圖書館的中華碩博士論文查詢系統，所下載的資訊相關系所的 3,646 篇碩博士論文，來自資訊工程研究所、資訊及電子工程研究所、資訊科學研究所、資訊教育研究所、資訊管理技術研究所、資訊管理研究所、電子與資訊工程技術研究所、電機與資訊工程研究所等不同系所。在語料庫 3,646 篇碩博士論文中共有 1,163,928 字，包括 2680 個不同的字，本論文針對雙音詞與三音詞作新詞萃取，經字組抽取後共有 1,058,078 個雙字組與 956,046 個三字組，其中不同的雙字組共有 110,258 個，不同的三字組有 344,585 個。字組出現次數超過四次且不存在於系統詞庫中的雙字組與三字組各 22,172 個與 32,119 個。我們使用教育部所發展的詞庫作為系統詞庫來過濾已知詞，其中包含有 48,330 筆雙音詞，11,558 筆三音詞。

我們分別定義了新詞萃取正確率與召回率，非詞彙字組篩檢正確率與召回率，來衡量系統的效能：

$$\text{新詞萃取正確率} = \frac{\text{萃取正確之總數}}{\text{萃取為新詞之總數}} \tag{3.1}$$

$$\text{新詞萃取召回率} = \frac{\text{萃取正確之總數}}{\text{新詞之總數}} \tag{3.2}$$

$$\text{非詞彙字組篩檢正確率} = \frac{\text{篩檢正確之總數}}{\text{篩檢為非詞彙字組之總數}} \tag{3.3}$$

$$\text{非詞彙字組篩檢召回率} = \frac{\text{篩檢正確之總數}}{\text{非詞彙字組之總數}} \tag{3.4}$$

實驗中，三字組庫中所有的三音新詞共有 1246 個，經由詞彙字組萃取法則所取出來的

三字組共有 761 個三字組，其中正確的共有 707 個新詞，例如，壓縮率、門檻值、超媒體、子集合等; 錯誤的的共有 56 個錯誤，例如，利用圖、行程式等，因此新詞萃取正確率是 92.9%，召回率是 56.74%如圖 3-1。

造成詞彙字組萃取法則錯誤的原因，是因為雙音詞與接頭詞、接尾詞、詞綴的結合，在語料中並不一定會構成正確的句法結構。根據萃取錯誤的三字組與前後文的關係，可以將錯誤分為兩類，第一類是萃取錯誤的三字組中其部分字組是屬於另一雙音詞或三音詞。因為萃取錯誤的三字組中的部分字組是屬於一新詞，並非所有的新詞皆是由詞庫中的詞與詞綴、接頭詞或接尾詞的結合而成，例如『四元樹』、『波茲曼』等。

在尚未歸類的字組中有許多可以經由法則式非詞彙篩檢法則正確地篩檢出來，因此我們先利用法則式非詞彙篩檢法則將一些可正確歸類的字組篩檢出來。雙字組庫中原本有 22172 個雙字組，其中 21399 個非詞彙雙字組。根據兩條非詞彙篩檢法則，將 15884 個雙字組篩檢出來，其中 15882 個正確篩檢，正確率 99.99%，召回率 74.22% 見圖 3-2。經過三音詞萃取法則之後，剩下 31358 個三字組，其中包含 30858 個非詞彙三字組，依據兩條非詞彙篩檢法則篩檢出 21231 個三字組，其中有 21213 個正確的篩檢，正確率 99.92，召回率 68.74%見圖 3-3。

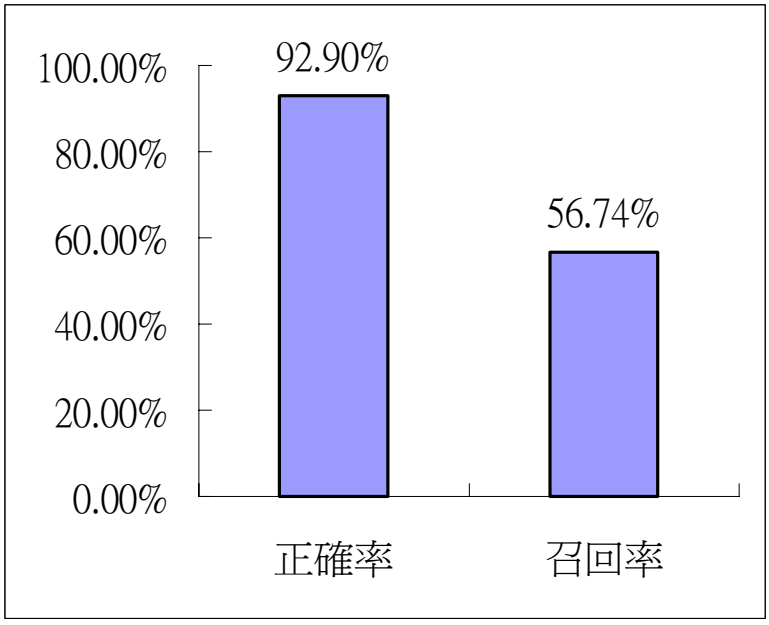


圖 3-1：法則式三音詞詞彙萃取正確率與召回率

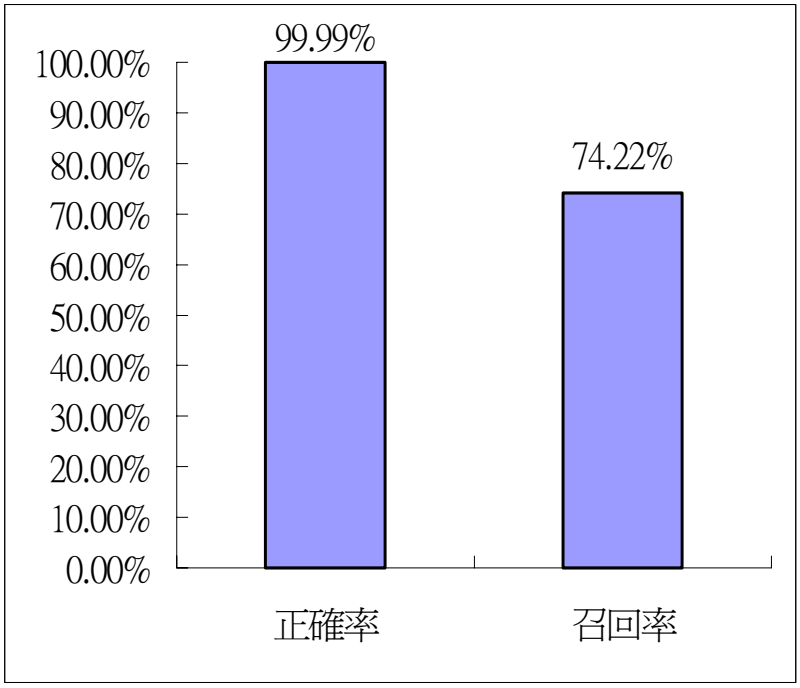


圖 3-2：雙字組非詞彙篩檢正確率與召回率

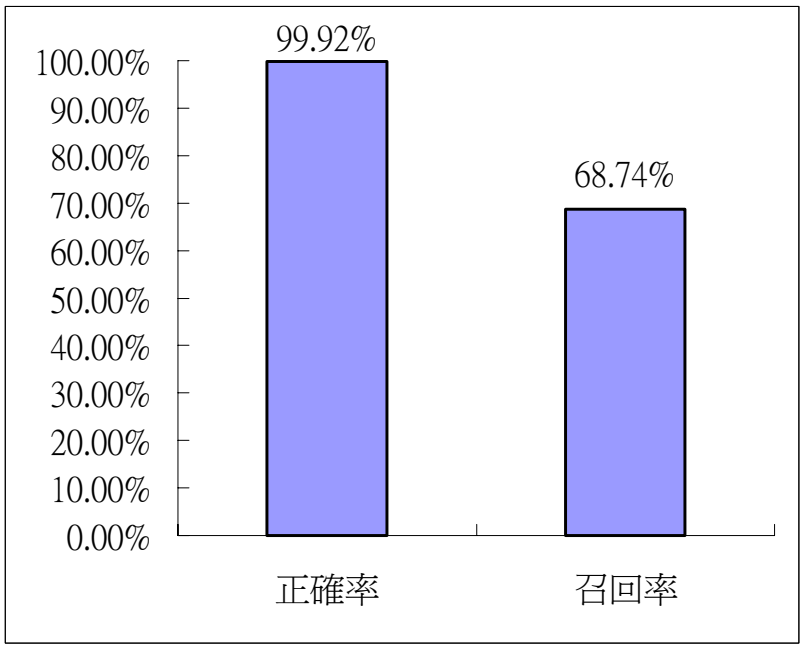


圖 3-3：三字組非詞彙篩檢正確率與召回率.

4. 類神經網路辨認模組

4-1 倒傳遞網路

由於詞的構成方式頗為複雜，且各中文字可能具有特殊的使用情形，因此若要以純法則式的辨認方法來判斷分類，必須要建立起很多的法則。另一缺點是建立的法則需考量不同的語料庫的特性。因此在第二階段的篩選時讀我們乃以字組間特徵值做為辨詞依據。

由於類神經網路中倒傳遞網路(multi-layer feed-forward with back propagation)具有學習正確率高、理論簡明[Zurada 92]，因此我們可將所挑選的特徵統計值做為此網路的輸入，建構成詞彙與非詞彙的分類器。我們使用具有一層隱藏層的倒傳遞網路，並且與輸入層是完全連接(full connect)且與輸出層亦是完全連接。在神經元的架構中，我們使用雙曲線正切函數(hyperbolic-tangent function)作為轉換函數。此函數具有微分容易的優點，可配合差距法則調整神經元間的權重，此函數當自變數趨於正負無限大時，函數值趨近於常數，其函數值域在[-1,1]之間。

4-2 特徵選取

統計式的特徵經常用到字組的出現機率，因此先定義字組的出現機率，再說明各種統計式的特徵。本論文將利用出現頻率來評估字組的出現機率如下

$$P(G_{ij}) = \frac{T(G_{ij})}{\sum_{j} T(G_{ij})} \tag{4.1}$$

其中 T（G_{ij}）表示長度為 i 的第 j 個字組 G_{ij} 的出現次數。字組間的特徵有

(1) 相對頻率(relative frequency count)[Wu 93]是將字組的出現次數除以所有字組的平均出現次數如公式(4.2)。

$$r_{ij} = \frac{f_{ij}}{K_i} \tag{4.2}$$

其中 r_{ij} 是指長度為 i 的字組庫中的第 j 個字組，f_{ij} 是 r_{ij} 的出現次數，K_i 是指長度為 i 的字組庫中所有字組的平均出現次數。一般的情況來說，相對頻率越高的字組，可能是屬於詞類的機率越高。

(2) 相關度(Association)[Sproat 90]定義如下：

$$A(ab) = \log_2 \frac{P(ab)}{P(a) \times P(b)} \tag{4.3}$$

其中 P(a)、P(b) 分別代表中文字 a 與 b 的出現機率。P(ab)代表雙字組 ab 的出現機率。此統計特徵有一缺點，當 P(a)、P(b)都很小的時候，A(ab)容易變得很大。三字組的相關度 A(abc)定義為：

$$A(abc) = \log_2 \frac{P(abc)}{P(a) \times P(b) \times P(c)} \tag{4.4}$$

其中 P(a)、P(b) 、P(c) 分別代表中文字 a、b 與 c 的出現機率，P(abc)則代表三字組 abc 的出現機率。

(3) 骰子矩陣 [Smadja93, 96]的定義如下：

$$D_2(x, y) = \frac{2P(x=1, y=1)}{P(x=1) + P(y=1)} \tag{4.5}$$

其中 P(x=1,y=1) 是中文字 y 緊跟著中文字 x 出現的機率，P(x=1)與 P(y=1)則分別是中文字 x、y 出現的機率。由上式可發現骰子矩陣與相關度很像，當 P(x=1)與 P(y=1)都很小的時候，則骰子矩陣是比相關度好的評量標準。三字組的骰子矩陣定義如下：

$$D_3(x, y, z) = \frac{3P(x=1, y=1, z=1)}{P(x=1) + P(y=1) + P(z=1)} \tag{4.6}$$

其中 P(x=1,y=1,z=1) 是中文字 z 緊跟著 xy 出現的機率，P(x=1)、P(y=1)與 P(z=1)則分別是中文字 x、y、z 出現的機率。

(4) 熵(Entropy) [Tung 94]是用來衡量字組與其相鄰字元的關係。若是有一字組的相鄰字元出現的分佈很亂，則可以想見在此字組很可能是一個詞。熵的定義如下：

$$H\_L(G_i) = -\sum_{C_j \in LN(G_i)} P\_L(C_j) \log_{T(G_i)} P\_L(C_j) \tag{4.7a}$$

$$H\_R(G_i) = -\sum_{C_j \in RN(G_i)} P\_R(C_j) \log_{T(G_i)} P\_R(C_j) \tag{4.7b}$$

$$P\_L(C_j):\frac{T(C_j-G_i)}{T(G_i)} \tag{4.7c}$$

$$P\_R(C_j):\frac{T(G_i-C_j)}{T(G_i)} \tag{4.7d}$$

其中 H_L(G_i)與 H_R(G_i)分別代表字組 G_i 的左熵與右熵，LN(G_i)與 RN(G_i)分別代表字組 G_i 的左相鄰字元集合與右相鄰字元集合，P_L(C_j)與 P_R(C_j)則分別代表字元 C_j 在 G_i 的左相鄰字元集合的出現機率，與右相鄰字元集合的出現機率。

從實驗中我們發現幾乎所有字組的相對頻率與骰子矩陣的特徵值都落在值域的最小百分之五，尤其骰子矩陣幾乎全都落在 0 到 0.05 之間，這樣的分佈幾乎顯不出字組的差異性。而二字組的相關度分佈情形相當接近高斯分佈（Normal Distribution），左熵與右熵除了特徵值為 0 的個數較多之外，其餘的分佈較為平均。因此我們首先以相關度、左熵與右熵作為系統的輸入特徵。若要考慮自動特徵選取的問題可以參考循序向前選取(Sequential Forward Selection)、Generalized "Plus l-Take Away r" Selection[Devijver 82]。若是原來特徵值分佈情形不佳的情形，可以透過一些轉換函數例如高斯函數(Gaussian Distribution)、雙彎曲函數(Sigmoid function)、雙彎曲正切函數(Hyperbolictangent function)來將值域與分佈情形加以轉換。利用自動特徵選取以獲得更好的系統效能是未來需要再加以研究的。

4-3 實驗與分析

在實驗中我們乃是以所提的倒傳遞網路辨識器和可能度(Likelihood)模組做分析比較[Duda 73]。可能度是評估在某一特定的情形下事件發生的機率。我們使用的可能度比率模組主要是修改 Wu [93]所提出用於抽取英文複合詞的模組。Wu 所選取的統計式特徵為相對頻率與相關度，因此利用雙變數的高斯函數的分佈作為可能度比率模組的機率分佈。以下是 Wu 所使用的詞類與非詞類雙變數機率函數：

$$f(A,R\,|\,Word)=$$
$$\frac{1}{2\pi\sigma_a\sigma_r\sqrt{1-r^2}}\exp\left\{-\frac{1}{2(1-r^2)}\left(\frac{(A-\mu_a)^2}{\sigma^2{}_m}-2r\frac{(A-\mu_a)(R-\mu_r)}{\sigma_a\sigma_r}+\frac{(R-\mu_r)^2}{\sigma^2{}_r}\right)\right\} \tag{4.8a}$$

$$f(A,R \mid Non-Word) =$$
$$\frac{1}{2\pi\sigma'_a\sigma'_r\sqrt{1-r'^2}}\exp\left\{-\frac{1}{2(1-r'^2)}\left(\frac{(A-\mu'_a)^2}{\sigma'^2_m}-2r'\frac{(A-\mu'_a)(R-\mu'_r)}{\sigma'_a\sigma'_r}+\frac{(R-\mu'_r)^2}{\sigma'^2_r}\right)\right\} \quad (4.8b)$$

其中 A 和 R 是代表相關度與相對頻率的變數。假設 A 和 R 都是屬於高斯分佈，而 μ_a 是詞類字組的相關度平均數、μ'_a 是詞類字組的相關度平均數，μ_a 是詞類字組的相對頻率平均數、μ'_a 是詞類字組的相對頻率平均數，σ_a 是詞類字組的相關度標準差，σ'_a 是非詞類字組的相關度標準差，σ_r 是詞類字組的相對頻率標準差，σ'_r 是非詞類字組的相對頻率標準差，r 是詞類字組相關度與相對頻率的相關係數，r' 是非詞類字組相關度與相對頻率的相關係數。定義了機率函數後，將機率函數套入對數可能度比率模組，若是 log λ 小於門檻值 T_{lrm} 則是屬於非詞類，若是 log λ 大於 T_{lrm} 則屬於詞類，在本系統 T_{lrm} 的預設值是 0。

我們所選取的特徵是相關度、左熵與右熵，因此我們使用多變數的高斯函數來作為可能機率函數[Duda 73]：

$$f(x \mid word) = \frac{1}{(2\pi)^{d/2}\,|\Sigma|^{1/2}}\exp\left[-\frac{1}{2}(x-\mu)^t\Sigma^{-1}(x-\mu)\right] \quad (4.9a)$$

$$\mu = E[x \mid word] \quad (4.9b)$$

$$\Sigma = E[(x-\mu)(x-\mu)^t] \quad (4.9c)$$

$$f(x \mid Non-word) = \frac{1}{(2\pi)^{d/2}\,|\Sigma'|^{1/2}}\exp\left[-\frac{1}{2}(x-\mu')^t\Sigma'^{-1}(x-\mu')\right] \quad (4.9d)$$

$$\mu' = E[x \mid Non-word] \quad (4.9e)$$

$$\Sigma' = E[(x-\mu')(x-\mu')^t] \quad (4.9f)$$

其中 x 代表一個行向量[A LH RH]^t，A、LH 與 RH 分別代表相關度、左熵與右熵的變數，並假設此三變數是屬於高斯分佈，μ 是代表詞類字組的特徵平均值，μ=[μ_A μ_{LH} μ_{RH}]^t，μ_A μ_{LH} μ_{RH} 分別代表詞類字組的相關度、左熵與右熵的平均值，μ' 是代表非詞類字組的特徵平均值，μ'=[μ'_A μ'_{LH} μ'_{RH}]^t，μ'_A μ'_{LH} μ'_{RH} 分別代表非

詞類字組的相關度、左熵與右熵的平均值，Σ是代表詞類字組特徵的相關係數矩陣，Σ'代表非詞類字組的特徵相關係數矩陣。

我們利用在 3.4 節定義的正確率與召回率來評估系統的效能，另外以加權式正確召回率(weighted precision recall,WPR)做為衡量，

$$\text{加權式正確召回率} = W_1 \times \text{正確率} + W_2 \times \text{召回率}, \tag{4.10}$$

其中 W_1 與 W_2 皆設定為二分之一。

我們以亂數選取三分之二的字組作為訓練資料，分別使用可能度比率模組與類神經網路模組進行新詞萃取。我們使用相關度、左熵與右熵作為統計式特徵，並且利用多變數高斯函數作為可能度比率模組的機率分佈，計算訓練資料的平均值與相關係數矩陣，可得到高斯函數的參數，套入高斯函數後可得到可能度比率模組。

在類神經網路萃取模組方面，由於相關度的值域比左熵與右熵大許多，因此我們先將此三種特徵作一簡單的值域轉換，將特徵的值域轉換到[0.05，0.95]及[-0.95，-0.05]。因為 0 在倒傳遞網路中，是沒有作用的，因此避開 0。若是特徵 X 的值永遠大於零，則使用以下的轉換函數

$$f(x) = \frac{0.95 - 0.05}{\max - \min}(x - \min) + 0.05 \tag{4.11a}$$

否則使用此函數

$$f(x) = \frac{0.95 - 0.05}{\max}(x) + 0.05, \text{ if x } \geqq \ 0 \tag{4.11b}$$

$$f(x) = \frac{0.95 - 0.05}{\max}(x) - 0.05, \text{ if x} < 0 \tag{4.11c}$$

因為我們首先只使用三種特徵，所以輸入層的節點個數是三個，輸出值亦只有一個，我們使用的轉換函數是雙彎曲函數其值域為[-1，1]，若是輸出值大於 T_{mlff} 則視為詞彙，若是輸出值小於 T_{mlff} 則視為非詞彙類別，系統預設的 T_{mlff} 是 0。

圖 4-1 是調整不同門檻值 T_{mlff} 與 T_{lrm} 時，類神經網路模組與可能度比率模組雙音

詞萃取正確率與召回率的變化情形。在雙字組的新詞萃取方面，當高召回率的情形時，類神經網路模組的正確率優於可能度比率模組;而當低召回率的情形時，可能度比率模組的正確率則優於類神經網路模組。

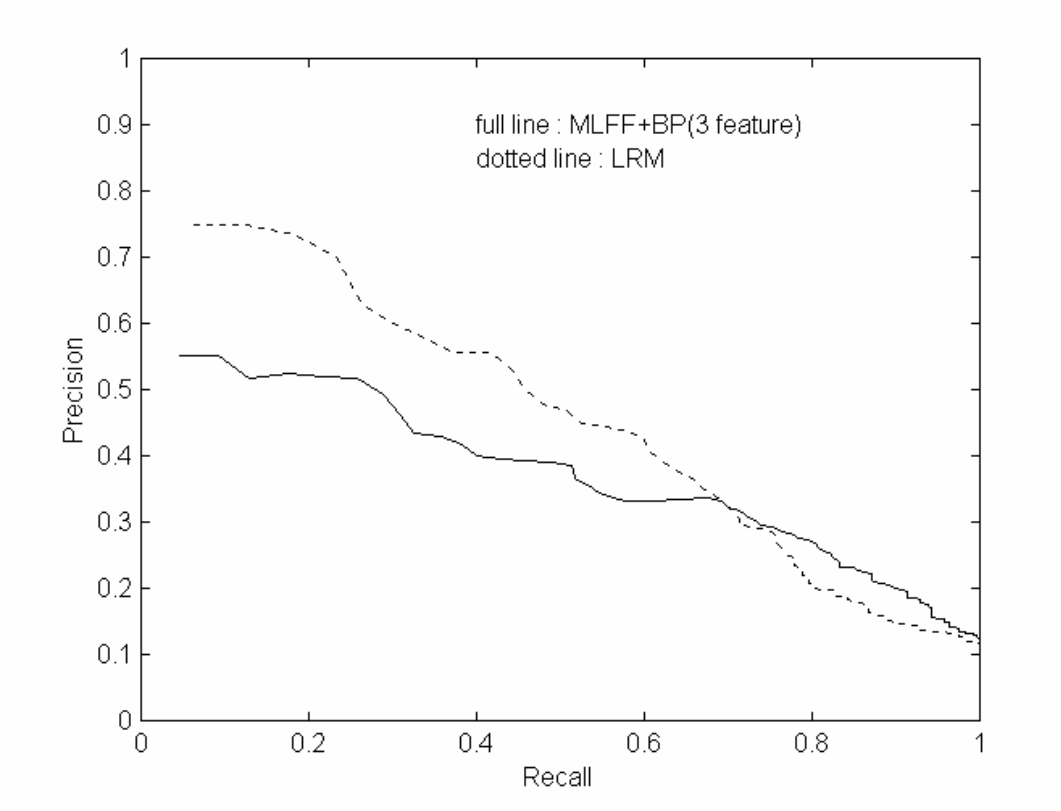


圖 4-1 雙音詞萃取效能比較圖

在三音詞的萃取方面，類神經網路模組在門檻值為預設值時，略優於可能度比率模組。觀察圖 4-2，發現類神經網路模組與可能度比率模組於三音詞的萃取能力並無明顯的優劣分別。

由於三字組中其二字組的資訊是有意義的因此在類神經網路模組三字組 $c_1c_2c_3$ 新詞萃取中除了原本使用的相關度、左熵與右熵的三個特徵外，我們另加入其部分字組 c_1c_2 與 c_2c_3 的相關度、左熵與右熵作為特徵，特徵個數增加為九個。在表 4.1 和圖 4-3 可知以特徵數的增加確實可提高分辨的正確率。

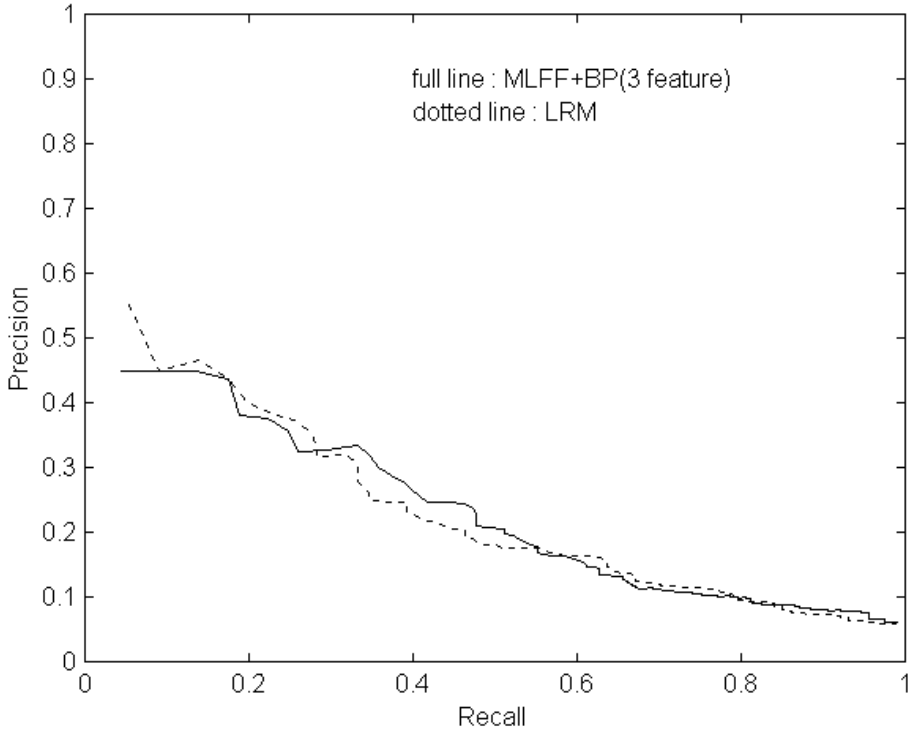


圖 4-2：三音詞萃取效能比較圖

	可能度比率模組	類神經網路模組（三種特徵）	類神經網路模組（九種特徵）
正確率	16.32%	13.68%	18.97%
召回率	59.3%	63.32%	77.89%
加權式正確召回率	37.81%	38.5%	48.83%

表 4-1 三音詞萃取效能比較表

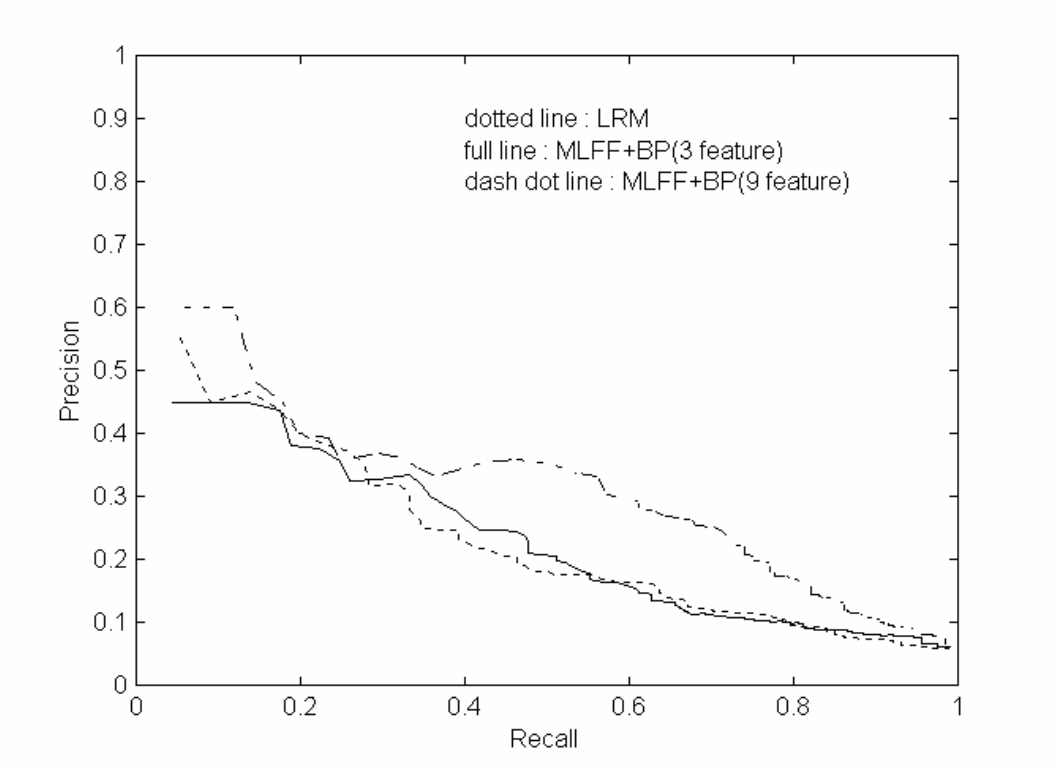


圖 4-3：三音詞萃取效能比較圖

5. 結論

本論文提出一個兩階段的中文新詞萃取技術，可應用於中文文件處理系統，將語料中有意義的新詞萃取出來。實驗數據的分析顯示利用構詞學的方法確實能有效的將三音新詞萃取出來，並且正確地將大部分非詞彙字組過濾掉。另一方面利用類神經網路結合各種統計式資訊來萃取新詞，可彌補構詞法則的侷限性。最後我們亦探討特徵的選取對於萃取的影響，並與可能度模組比較。從實驗的結果我們得知三音詞中二字組特徵的加入確實能提高三音詞新詞的正確率與召回率。

本論文的後續研究方向主要有特徵的自動選取。在使用類神經萃取模組時，選取的特徵的好壞會直接影響到系統的效能，在本論文使用分析其值域分佈情形來作特徵選取。但是當可使用特徵很多，導致難以逐個分析時，則需利用特徵自動選取來解決這個問題。與此問題相關的還有特徵的值域轉換問題，當各種特徵資訊的值域範圍相差太大時，就需要特徵的值域轉換，避免系統被少數幾個特徵所主宰。

6. 參考文獻

Li, Charles N. and Thompson, Sandra A., "Mandarin Chinese," University of California Press, New York, 1992.

Yeh, Ching-Long and Lee, His-Jian, "Rule-Based Word Identification for Mandarin Chinese Sentences – A Unification Approach," Computer Processing of Chinese & Oriental Languages, Vol. 5, No.2, March 1991.

Smadja, Frank, "Retrieving Collocations from Text: Xtract," Computational Linguistics, Vol. 19, No. 1, 1993, pp. 143-177.

Smadja, Frank, McKeown, K.R. and Hatzivasiloglou, V. "Translating Collocations for Bilingual Lexicons," A Statistical Approach," Computational Linguistics, Vol. 22, No. 1, 1996.

Zurada, Jacek M., "Introduction to Artificial Neural Systems", West Publishing Company, USA, 1992.

Nie, Jian Yun, Hannan, Marie-Louise and Hannan, Wanying, "Combining Dictionary, Rules and Statistical Information in Segmentation of Chinese," Computer Processing of Chinese and Oriental Languages, Vol. 9, No. 2, December 1995, pp. 125-143.

Chang, Jing Shin, "Automatic Lexicon Acquisition and Precision-Recall Maximization for Untagged Text Corpora", National Tsing-Hua University, P.h.D. thesis, 1997.

Chang, J. S., Chen, C. D. and Chen, S. D., "Chinese Word Segmentation through Constraint Satisfaction and Statistical Optimization," (in Chinese) Proceedings of ROCLING-IV, R.O.C. Computational Linguistics Conferences, Taiwan ROC, 1991, pp. 147-165.

Church, K. and Hanks, P., "Word Association Norms, Mutual Information and Lexicography," Computational Linguistics, Vol.16, March. 1990, pp. 22-29.

Chen, Keh Jiann, Bai, Ming Hong, "Unknown Word Detection for Chinese by a Corpus-based Learning Method," Proceedings of ROCLING X, Taipei, Taiwan, ROC, 1997, pp. 159-174.

Lin, M.Y., Chang, T. H. and Su, K. Y., "A preliminary study on unknown word problem in Chinese word segmentation," Proceedings of 1993 R.O.C. Computational Linguistics Conference, Taiwan, 1993, pp.119-137.

Wu, M. W. and Su, K. Y. "Corpus-based Automatic Compound Extraction with Mutual Information and Relative Frequency Count," Proceedings of ROCLING VI, Nantou, Taiwan, ROC, Sep. 1993pp. 207-216.

Sproat, Richard and Shin, Chilin "A Statistical Method For Finding Word Boundaries In Chinese Text," Computer Processing of Chinese & Oriental Language, Vol. 4, No. 4, March 1990.

Chen, S. C. and Su, K. Y. "The Processing of English Compound and Complex Words in an English-Chinese Machine Translation System," Proceedings of ROCLING I, Nantou, Taiwan, 1988, pp. 87-98.

劉興寰, “中文語料詞類自動標記,” 國立清華大學, 碩士論文, 1994。
謝國平, “語言學概論,” 三民書局, 1986。
詞庫小組, “新聞與語料詞頻統計表,” 1993。
詞庫小組, “搜文解字：中文詞界研究與資訊用分詞標準,” 1996。

Parsing Chinese by Examples

Oliver Streiter

Academia Sinica, Institute of Information Science

Page 41 ~ 65

Proceedings of Research on Computational Linguistics

Conference XIII (ROCLING XIII)

Taipei, Taiwan

2000-08-24/2000-08-25

Parsing Chinese by Examples

Oliver Streiter
Academia Sinica, Institute of Information Science,
Nankang, Taipei, Taiwan 115
http://rockey.iis.sinica.edu.tw/oliver

Abstract

This paper presents a Chinese parser which has been derived from the Chinese treebank developed at CKIP, Academia Sinica. Contrary to previous approaches which aim at the conversion of a treebank into a parsers, we do not derive phrase structure rules of any type. Instead, the approach chosen relies on a fuzzy pattern matching strategy in order to extract relevant examples from the treebank. Via a set of adaptation mechanism, these examples are merged and modified so as to produce the best parse for the given set of examples. A detailed description of the parser is provided. The different modules of this parser are evaluated. It is shown that the parser is not only efficient and robust but provides a reasonable level of linguistic adequacy which can be improved upon by restricting the application domain or increasing the number of examples. Competitive approaches are presented and compared to the proposed approach.

1 Introduction

1.1 From a Treebank to a Parser

The Academia Sinica (AS) disposes of rich resources for the automatic treatment of Modern Mandarin Chinese, among them the manually tagged AS-corpus of about 5 Million words (Huang and Chen, 1992) and a lexicon containing about 80.000 words described with respect to their main semantic and syntactic properties (Huang et al., 1995). With the help of a rule-based parser (Chen, 1996) a treebank of manually corrected sentences has been create recently, containing about 40.000 trees (Chen et al., 1999).

This paper describes the attempt to reshape these resources into an example-based parser which ascribes as detailed information to a sentence as can be found in the treebank.

The annotation guidelines for the treebank and a sample of 1000 trees can be found at `http://godel.iis.sinica.edu.tw/CKIP/`. One example tree is reproduced in Fig.1. A BNF of the tree structure is added in Fig.2. While the semantic role labels are almost self-explaining, POS tags are more complex: Tags starting with `N` refer to nouns, starting with `V` refer to verbs, starting with `P` refer to prepositions etc. Additional characters develop a finer classification e.g. `VK1` is a subset of `VK` which is a subset of `V`. The current specifications

comprise almost 200 POS tags, 45 phrasal labels and 46 semantic role labels.

```
S(experiencer:Nep:"ta1"|
  Head:VE2:"xia3ng"|
  goal:S(agent:Nap:"ge1ge"|
         epistemics:Dbaa:"yi2di4ng"|
         epistemics:Dbaa:"hui4"|
         Head:VE2:"shuo1"))
```

Figure 1: Example from the treebank: *he think older_brother certainly can speak*

```
<top> ::= <ident> " " <cat> "(" <tree> *( "|" <tree>) ")"
<tree> ::= <role> ":" <cat> ( "(" <tree> *( "|" <tree>) ")" | ":" <Word>)
<ident> ::= {00001, 00002, 00003, ...}
<role> ::= {agent, theme, goal, Head, head, epistemics,  ...}
<cat> ::=  {S, NP, VP, PP, GP, Nep, VE2, Nap, ...}
<Word> ::= {word1, word2, ....}
```

Figure 2: The Backus-Naur Form of the tree structure.

Comparing this treebank to the Chinese Penn-Treebank currently under development (Xia et al., 2000; Xue et al., 2000), we observe the following main differences .

	Penn-Treebank	CKIP Treebank
Chinese character	simplified	traditional (BIG5)
size	3.289 sentences	40.000 sentences
domain	mainly economy	balanced, mixed
average sentence length	27 words	6.3 words (cf. Fig.10)
word to character ratio	1.72	1.87
word to POS ratio	1.10	1.14
POS tags	33	200
syntactic functions	not for every constituent	0
semantic roles	0	46
underlying linguistic theory	GB	idiosyncratic
empty categories	PRO, pro, T(race) ...	not used
branching	deep (almost binary)	flat

Figure 3: The CKIP Treebank compared to the Chinese Penn-Treebank.

1.2 From the "Ultimate Parser" to the Nearest Neighbor (NN)

An example-based parser, in its most simple form, consists of a storing and retrieval function that returns for every learned sentence the associated tree-structure. If for every sentence such a tree-structure can be found this parser would be the "Ultimate Parser" (Sekine and

Grisman, 1995). It would be easy to implement, efficient and easy to maintain. Unfortunately, with large or open subject domains, such an idealized parser cannot perform well, an insight which let Sekine and Grisman return to more conventional parsing approaches. We think, however, that the main idea of the "Ultimate Parser" may be retained if we do not require exact matches of the Input Sentence (ISs) with stored Example Trees (ETs) but content us to retrieve similar ETs and to adapt them according to the kind of mismatch which has occurred.

For this purpose we employ the so-called k-nearest neighbor classifier (k-NN classifier), an approach underlying paradigms as *Example-Based Machine Translation*, *Translation Memories* and *Case-Based Reasoning* (Collins and Cunningham, 1996). According to the latter approach, a new problem is approached by retrieving similar problem formulation from a data-base together with their associated solutions. In a consecutive step the old solutions are *adapted* to the new problem formulation. This approach is considered to result in efficient and qualified problem solutions.

Transferring this approach to the task of parsing, we can identify the problem formulation with an IS to be parsed and the solution with the stored ETs. The adaptation consists in modifying the stored ETs there where the IS does not match the ET.

1.3 From Generalizations to Fuzzy Match

If we accept differences between IS and ET, we may try to find out automatically, or via human reflections what this mismatch can and should be. For example, we could allow pronouns to be matched on proper nouns and vice versa. Predictions of such allowed mismatches are called *generalizations* and are commonly used within memory-based NLP-systems, e.g (Brown, 1999b; Brown, 1999a; Carl, 1999; Streiter, 1999). Due to the pre-definition of such generalizations, they can form part of the indexing system and, as a consequence, matches can be performed efficiently.

In previous experiments (Streiter, 1999) we could show that a still greater degree of flexibility, which cannot or should not be pre-determined may improve the performance. It could be shown that a retrieval requiring strict or generalized matches cannot compete with a fuzzy retrieval that allows for *substitutions* (an incompatible tree slot, e.g. a pronoun matches an adverb) or *deletions* (a tree slot is missing, e.g. no time adverb in the ET) if *adaptations* are applied in order to handle the mismatch. If, for example, an adverb substitutes a subject pronoun, simple frequencies collected during the training of the treebank allow to overwrite the labels associated to the pronoun by the most probable labels associated to this adverb. If a temporal adverb has been deleted, the adaptation inserts this word together with its most likely POS and semantic role.

1.4 From Generalizations to Exact Matches

Generalizations are useful if the set of examples is not large enough to *cover* the input: They help to increase the *coverage*. As could be shown in (Streiter, 2000) however, they may threaten the *reliability*, i.e. the capacity to correctly retrieve ETs once learned in the case the generalizations lead to an ambiguous matching of input and output. In such a case we speak of an *over-generalization*. Unfortunately, *over-generalization* can hardly be avoided as each word behaves differently. Therefore, in many memory-based approaches, generalizations are stored in addition to the original ungeneralized form. If the match with the generalization is ambiguous, the system may resort to the original encoding in order to resolve the ambiguity (Bod and Kaplan, 1998; Daelemans, 1998; Daelemans et al., 1999; Carl, 1999).

2 The parser from a bird eye's view

2.1 Training

The training phase consists of two runs through the treebank. The first run aims at the statistical acquisition of weights which describe how strong a word or its POS is related to a syntactic pattern it occurs in. In a second run, the trees of the treebank including all subtrees are indexed. As indeces we use the words and their POS. Each index is associated with a weight which has been calculated in the first run and points to all trees in which it occurs in. The indexing technique is called an inverted index (Grandy, 1999), a strategy used otherwise for full-text indexing. This technique in not only fast but allows also for the required fuzziness.

2.2 Parsing

Parsing starts with the extraction of k ETs, summing up the weights for all ETs referred to by the words and POSs of the IS. k ETs which accumulate the highest sum of weights are retained. As not every position in an ET has to be matched by an index (word or POS), the match between the IS and the ET may be inexact. A mismatch does not block an already retrieved ET; it only does not increase the total score for the ET.[1]

In the following step the k ETs and the IS are aligned: If, an ET is smaller than the IS (we face a *deletion*), the best mapping of the words of the IS and the positions in the ET has to be found. Words which are not aligned (deleted words) are inserted later on in order to obtain a complete parse.

One way to insert deleted words is the combination (mixing) of the k aligned ETs. For this purpose the aligned trees are segmented into opening phrases, e.g. "$S_{level=1,head=VE2}$(" ,

[1]This and the following steps are illustrated in detail in (Streiter and Hsueh, 2000) for one parsing example.

words, e.g. "Head$_{level=1,head=VE2}$:VE2:xiang3", and closing phrases, e.g. ")S$_{level=1,head=VE2}$". Such fragments of all ETs are transformed into nodes of a common lattice. Those nodes which are neighbors in a sentence are linked via a transition. The best path through the lattice is generated in the backward pass of a forward-backward two-pass search. As for the alignment mentioned above, the use of the Viterbi-algorithm allows for an efficient implementation of this task. This adaptation strategy will be referred to as *combinatorial adaptation*.

The next adaptation step, referred to as *derivational adaptations* identifies awkward subtrees and replaces them with subtrees obtained by the recursive application of the hole parsing procedure to the words of the subtree. As the phrase to be parsed is shorter than the original IS, we are likely to obtain a better match, unless the chunking into phrases is wrong (as illustrated in Streiter and Hsueh (2000)). The purpose of this recursive call is, similar to the previous adaptation step, to correct badly matched trees and to insert deleted words.

The final *structural adaptations* operate on single words. They handle accidental word mismatches, unknown words, phenomena of type shifting and metonymical extentions of words. They compare the encoding of a word in the retrieved ET with what is know about this word in the lexicon and the learned tree structures. In the case of a mismatch, either the mismatch is maintained (type shifting and metonymies) or the mismatch is attempted to be corrected by assigning the words most likely POS and semantic role (given the POS of the head-word).

3 The Parser in Detail

3.1 Training

3.1.1 Deriving Weights

In a first run we calculate the statistical association between a specific index and a specific tree structure, where an index ($\mathcal{I}$) is a) a word-form ($\mathcal{L}$) (which is Chinese is almost identical to the lexeme), b) its POS ($\mathcal{C}$), c) an abbreviated POS for verbs and nouns ($\mathcal{A}$), and d) a semantic feature ($\mathcal{S}$).

$$\mathcal{I}_x \in \{\mathcal{L}_x, \mathcal{C}_x, \mathcal{A}_x, \mathcal{S}_x\} \quad e.g. \quad \mathcal{I}_{\text{"}ge1ge\text{"}} \in \{ge1ge, Nap, N, human\} \tag{1}$$

The weight $W^s_{I_{pos}}$ we assign to an index $\mathcal{I}_x$ is its paradigmatic weight, i.e. the relation between the index and the sentence structure. For $\mathcal{L}_x$ and only for $\mathcal{L}_x$ this weight is completed by the syntagmatic weight.

$$W^s_{L_{pos}} = Wsyntag^s_{L_{pos}} + Wparadig^s_{L_{pos}} \tag{2}$$

Syntagmatic Weights The syntagmatic weight describes the contribution of $\mathcal{L}_x$ to the string of a sentence, comparable to the power of a word to trigger a poem or a song in humans. The syntagmatic weight for $\mathcal{L}_x$ in the position *pos* of a sentence s calculated as follows:

$$Wsyntag^{s}_{L_{pos}} = \frac{log(10 + length(s))}{length(s)} \tag{3}$$

The aim of the syntagmatic weight is to enhance the reliability of the parser, i.e. to correctly retrieve learned examples. Without this syntagmatic weight the parser max prefer similar matches with a high probability over exact but unlikely matches. This may already happen with a training corpus as small as 100 sentences. Using this syntagmatic weight, the reliability can be maintained even with very large training corpora.

Paradigmatic Weights The paradigmatic weight of an index $\mathcal{I}$ for a sentence s is calculated indirectly by braking down sentence s in a set of paradigms $\mathcal{P}^{s}_{p}$.

$$Wparadig^{s}_{I_{pos}} \sim= Wparadig^{p}_{I_{pos}} \tag{4}$$

How to obtain $Wparadig^{s}_{I_{pos}}$ from $Wparadig^{p}_{I_{pos}}$ will be shown below in Section 3.1.2. What a paradigm is and how it is related to a tree is illustrated in Fig.4.

```
S(experiencer,Head,goal)
S(agent,epistemics,epistemics,Head)
```

Figure 4: Paradigms derived from $sentence_{Fig.1}$.

There is more than one way to calculate the association between an index and the pattern it occurs in. Many association measures are reported to be equivalent when their outcome is transformed onto an ordinal scale (Rijsbergen, 1979) (they are said to be monotone with respect to each other). However, within the current setting we use a ratio scale which has to represent the fact that, for example, two bad matches are better than one good match or not. Whether the different association measures can provide valid information of this type cannot be concluded given the definition of the association measure. We therefore conducted a sequence of experiments reported on in Section 4.2 in which we identified association measures which are more adequate for the task at hand than others.

In order to allow for a better understanding of the weights tested, we develop here, as an example, two weights. The first is derived from the Mutual Information of the paradigm ($\mathcal{P}_p$) and the index in that position of the paradigm ($\mathcal{I}_{pos}$) (with *pos* = 1,2,3,...) and the

second is derived from the conditional probability to have the paradigm ($\mathcal{P}_p$) given the index in that position of the paradigm ($\mathcal{I}_{pos}$).

Be fq_t the total number of observations we make in the training corpus during which we observe a) the joint occurrence of position *pos* with an index $\mathcal{I}$ ($fq_{i,pos}$), b) a paradigm $\mathcal{P}_p$ (fq_p) and c) the joint occurrence of $\mathcal{I}_{pos}$ in $\mathcal{P}_p$ ($fq_{p,i,pos}$). We can calculate the $MI_{P_p,I_{pos}}$ and $P(p|I_{pos})$ as:

$$Wparadig1^p_{I_{pos}} \sim= MI_{P_p,I_{pos}} = log\frac{P(\mathcal{P}_p \bigcap \mathcal{I}_{pos})}{P(\mathcal{I}_{pos}) \cdot P(\mathcal{P}_p)} = log\frac{\frac{fq_{p,i,pos}}{fq_t}}{\frac{fq_{i,pos}}{fq_t} \cdot \frac{fq_p}{fq_t}} = log\frac{fq_{p,i,pos} \cdot fq_t}{fq_p \cdot fq_{i,pos}} \quad (5)$$

$$Wparadig2^p_{I_{pos}} \sim= P(p|I_{pos}) = \frac{P(\mathcal{P}_p \bigcap \mathcal{I}_{pos})}{P(\mathcal{I}_{pos})} = \frac{\frac{fq_{p,i,pos}}{fq_t}}{\frac{fq_{i,pos}}{fq_t}} = \frac{fq_{p,i,pos} \cdot fq_t}{fq_{i,pos}} \quad (6)$$

By removing the constant fq_t we simplify the calculus and obtain scores between 0 and 1, so that the logarithmic transformation is no longer necessary.

$$Wparadigm1^p_{I_{pos}} = \frac{fq_{p,i,pos}}{fq_p \cdot fq_{i,pos}} \quad (7)$$

$$Wparadigm2^p_{I_{pos}} = \frac{fq_{p,i,pos}}{fq_{i,pos}} \quad (8)$$

As can be seen, these two values and those we shall test below differ with respect to the normalization of the joint occurrence $fq_{p,i,pos}$. The measure derived from the Mutual Information provides for a maximal normalization, while the measure derived from the conditional probability does not normalize for fq_p and thus reproduces frequent structures more often than infrequent structures.[2] Both normalize for $fq_{i,pos}$, i.e. reduce the score if the index $\mathcal{I}$ occurred also in different patterns $\overline{p}$. This is may be questionable, especially if $\mathcal{I}$ nevertheless occurred in all or most patterns p. After all, none of the scores tested below seems to be perfect, as none of them obtains the best scores for frequent and infrequent structures.

3.1.2 Indexing

During the indexing of the ETs, we transform the weights we have obtained for the patterns into weights for ETs. In general, the weight of an index $\mathcal{I}$ in sentence s is the weight we calculated for $\mathcal{I}$ in $\mathcal{P}^s_p$. For head-words of embedded phrases (patters), which have received two scorings, one as dependent of the upper level and one as the head of the lower level (e.g.

[2]The conditional probability is frequently used in studies which try to model the human language performance (Hoogweg, 1999; Kaplan, 1996; Bod and Kaplan, 1998). In this light, it seems reasonable to assume that studies using the Mutual Information are competence studies - but do the authors agree upon that?

"shuo1" in Fig.1 is scored once as "Head" at level 2 and once as "goal" at level 1) only the better score is retained.

Inverted indices as used here to retrieve ETs are position independent and as such an optimal indexing mechanism for free word order languages like Russian (although the words of different phrasal levels should not be confused). Word order in Chinese is less free and therefore Chinese may not be well suited for a position-less indexing. In addition, position-less matching requires complex adaptation strategies which have not been investigated until now. Therefore we have to assign the index and its score to a position in the ET. As we intend to match ISs onto ETs which are smaller than the IS, which is not possible if we retain absolute position values (e.g. 3th word of a sentence of 12 words), we map the position onto what we call an index-position *ipos* by transforming the position onto a scale of 10 (e.g. 3/12 = index-position 2). The resulting tuple $< \mathcal{I}, ipos >$ serves as index to the tuple $< tree_s, Wparadig^s_{I_{ipos}} >$.

$$Wparadig^s_{I_{ipos}} = max\, Wparadig^s_{I_{integer(10 \cdot \frac{pos(I)}{length(s)})}} \tag{9}$$

In order to parse fast, even with tens of thousand trees learned, we let the system automatically extend the index with key-words. A $key-word$ is a word which occurs more than 100 times at a given index-position. If a sentence to be indexed contains a key-word, all indices of the sentence are extended by the keyword and its index position. More than one keyword are allowed, extending the index to $< \mathcal{I}, ipos * (, keyword, ipos_{keyword}) >$. More sentences are learned, more words are used as key-word. The additional indexing may improve the performance (not only in time), if the search-space is limited correctly (similar to document clustering in Information Retrieval (Rijsbergen, 1979), or the parsing experiments reported in (Kim and Kim, 1995)), however, if the search-space becomes to small, or, in the worst case, no intersection of the key-words can be found, also negative effects may be expected. It goes without saying that most key-words are high frequent function words like *de* and *le* together with their index-position, but also *shì... de*, *bú shì*, *zài... de... xià* or *zài... de... zhōng* constructions are indexed when training data becomes larger.

3.2 Parsing

3.2.1 NN-Retrieval

Parsing starts with a lexicon look-up which transforms the word of an IS into indices ($\mathcal{I}_x$). The positions of the words are transformed into the index-position as described above. With the help of the resulting index $< \mathcal{I}, ipos * (, keyword, ipos_{keyword} >)$ we access the database and retrieve tuples of $< tree_s, Wparadig^s_{I_{ipos}} >$. One matching index is sufficient in order to

```
sub NN-retrieval {
    KEY-WORD=mkKEY-WORD(WORD);
    for each (WORD,POSITION) {
        INDEX=mkINDEX(WORDi);                          # make the index
        IND-POS=mkINDEX-POS(POSITION);                 # make the index-position
        for each(INDEX) {                              # database is looked up
            (TREE,SCORE)=$treeDB{INDEXi,IND-POS,KEY-WORD}
            $NNscore{TREE}+=SCORE }                    # accumulate scores per tree
    return best_NN(%NNscore) }                         # return best NN
```

Figure 5: Fuzzy NN-retrieval algorithm.

retrieve an ET. In order to distinguish this match from a better match we sum up the scores for every tree.

3.2.2 Alignment

After the NN-Retrieval k ETs and IS are aligned. As IS may have more words than ET, we have to determine which words of IS matches best with which word in ET and which words are not matched (i.e. deleted during the fuzzy match). In order to solve this task efficiently, dynamic programming strategies can be used: Imagine the words of IS to be plotted on the x-axis and the slots in ET to be plotted on the y-axis. We first determine the "envelope", i.e. all possible combinations of x and y. Within this envelop every cell is filled with a score.

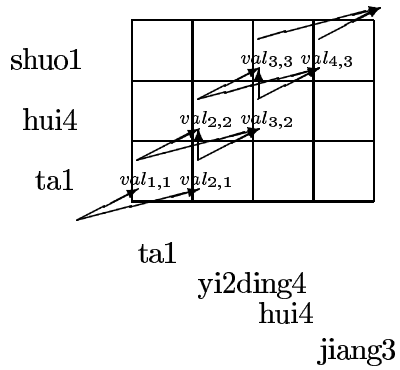


Figure 6: The alignment of IS and ET.

This score consists of the sum of three sub-scores, each being the product of $W_{P_p,I_{pos}}$ defined above and the similarity between $\mathcal{I}_x$ and $\mathcal{I}_y$. For the moment these similarity measures are quite rudimentary.[3]

[3]For $\mathcal{L}$, they yield binary values (0,1), for $\mathcal{C}$, the surface similarity of the features is used (e.g. $\mathcal{C}_x$=Nha and $\mathcal{C}_y$=Nhb yields 0.66). For $\mathcal{S}$, some hand-coded rough estimations are used.

$$val_{x,y} = \sum_{\mathcal{I} \in \{\mathcal{L},\mathcal{C},\mathcal{S}\}} (W_{P_p,I_{pos}} \cdot similarity(\mathcal{I}_x, \mathcal{I}_y)) \quad (10)$$

The alignment of the IS and ET in Fig.6 consists of finding a path through this lattice which accumulates most scores. By storing in a hash table partial best paths (e.g. the best path starting from (3,2) to the end), not all possible paths have to be run through, but can be calculated by summing up these partial results (Viterbi-Algorithm cf Ryan and Nudd (1993)). The last step of the alignment consists of the replacement of the words in the ET by the aligned words of the IS. The POS found in the ET is still retained for a while, as this coupling of word and POS allows to identify mismatches.

3.2.3 Combinatorial Adaptation

An almost identical algorithm is used for the *combinatorial adaptation*. The lattice we see in Fig.7 consists not of words as in Fig.6 but of all segments of the k aligned trees. The scores for each segment are those which have been calculated during the alignment procedure and are not reproduced.

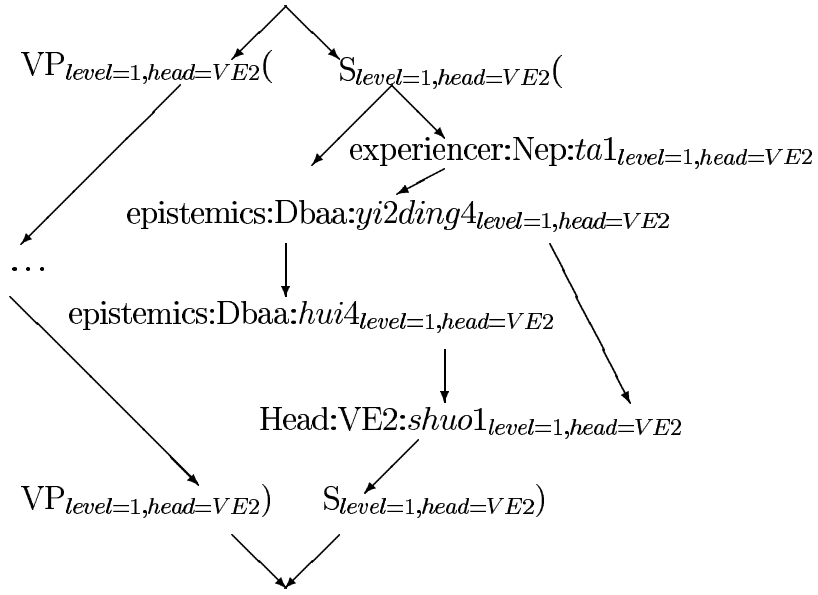


Figure 7: A lattice of tree-segments for the *combinatorial adaptation*, merging two partial matches: *yídìng huì shuō* and *tā yídìng shuō* into *tā yídìng huì shuō*.

Looking for the best path trough this lattice allows for the combination of different trees: the insertion of the analysis of one word or phrase of tree A into tree B. In order to obtain coherent tree structures, the level (= the depth) and the head-POS of each segment are annotated on the nodes of the lattice. In addition, the sequence of words of the resulting tree must not contradict that of the IS.

3.2.4 Derivational Adaptations

The next adaptation step identifies awkward subtrees and replaces such subtrees with the re-parse of the words of the awkward subtrees. A phrase is re-parsed if there is a relation between a word and its POS which is not attested in the learned corpus nor in the lexicon. This adaptation allows, similar to the previous adaptation, to correct badly matched trees and to insert deleted words. The largest possible sub-tree is chosen for the re-parsing in order to have the largest possible context for the unmatched word and, secondly, to correct possible errors in the surrounding which may have been caused by this mismatch.

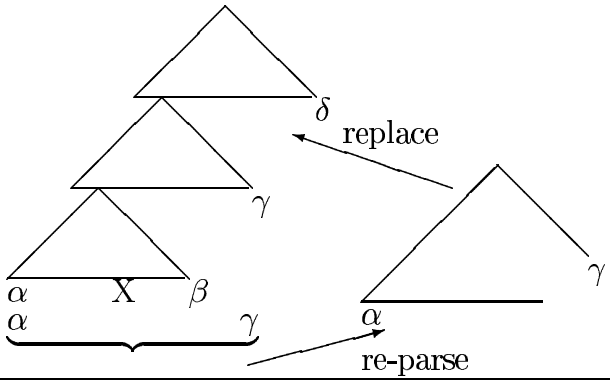


Figure 8: Re-parsing of the subtree form α to γ, triggered by a mismatch in X.

It goes without saying that mismatches at the sentence top level cannot be corrected by this adaptation strategy. The last adaptation, the structural adaptation, is applied to these words.

3.2.5 Structural Adaptations

Structural Adaptations are triggered by the same unattested relations which also trigger the re-parsing, i.e an unknown combination of a word and its POS given the POS of the head-word. If the word is unknown, the assigned POS is maintained. In the future we intend to combine this top-down unknown word guessing with a bottom-up unknown word guessing, so that at this stage two kinds of analysis would have to be conciliated.

If the POS found in the current ET and that found in the lexicon or in the learned trees are very similar, the POS is replaced by the attested known similar POS, however the semantic role of the ET is maintained (assuming that it is compatible with the new POS). If the POS is very different, a new POS and a new semantic role are searched for. POS and semantic role should combine with the word in question and the POS of the head-word. Such information has been collected during the training of the treebank, is however not sensitive to the context (e.g. the role "theme" might be assigned, although already present

in this phrase, cf. Appendix B, sentence 11).
In the future we hope to be able to handle metaphorical extentions of words at this level also. Given the statistical relatedness of a semantic feature and a pattern (sentence) expressed by the association measure, we may retain the semantic feature found in the ET if it is very strongly related to the ET and add a POS of the word of the IS, which is normally not compatible with this semantic feature.

4 Evaluation of Modules

4.1 Main Approach

The Evaluated Unit We have chosen the semantic role relation between a head-word and its dependent words as the entity to be evaluated. The semantic role relation can be though of as turning the phrase structure tree into a dependency tree the arcs of which are labeled with semantic roles. Thus, given the tree in Fig.1, we evaluate the correctness of the triples $< head - word, relation, dependent - word >$, (e.g. $< xia3ng, experiencer, ta1 >$, $< xia3ng, goal, shuo1 >$, $< shuo1, agent, ge1ge >$ etc.

Assuming a hierarchy of increasingly hard evaluation measures (Fig.9), the correct assignment of a semantic role between α and β implies the correct syntactic function, which requires the correct identification of the dependency relation which again requires the bracketing of α and β into one phrase.

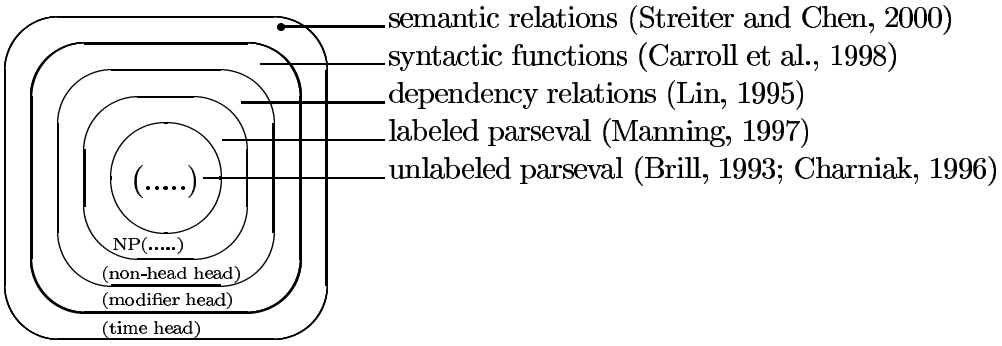


Figure 9: The chosen evaluated unit in a hierarchy of increasing hard evaluation units.

The Basic Measures Dividing the number of correctly identified semantic role relations by the number of semantic role relations in the reference corpus, we obtain the *recall*. Dividing the number of correctly identified semantic role relations by the number of semantic role relations in the parsing output we obtain the *precision*. Both scores are combined into the *f-score* via the following formula.

$$f\text{-}score = 2 \cdot \frac{precision \cdot recall}{recall + precision} \tag{11}$$

Beside specifying the *recall*, *precision* and *f-score* over the whole test corpus we specify the *f-score* for each sentence length. This allows us to estimate not only the contribution of a specific measure to the overall *f-score* but shows whether long or short sentences take more or less advantage of them.

Derived Measures From the basic measures (recall, precision and f-score) we derive the *coverage* and the *reliability*. The *coverage* describes the performance (in terms of the f-score) on untrained ISs. The *reliability* as defined in (Streiter, 2000; Streiter et al., 2000) is measured using the *f-score* obtained with trained + untrained items. Contrary to the *coverage*, the *reliability* quantifies the ability of the parser to correctly retrieve learned items. To correctly retrieve learned items is not evident for approaches which decompose during training and re-compose during learning, including probabilistic phrase structure grammars, hand-written grammars or even Translation Memories (Carl and Hansen, 1999). *Reliability* values which are not 1 or close to 1 may explain bad performances with large training data. In addition, such systems cannot be trained satisfyingly for a closed domain application which requires 100% correctness. In order to determine the *reliability* we train the training corpus together with reference corpus and test with the same test corpus (hide-and-seek).

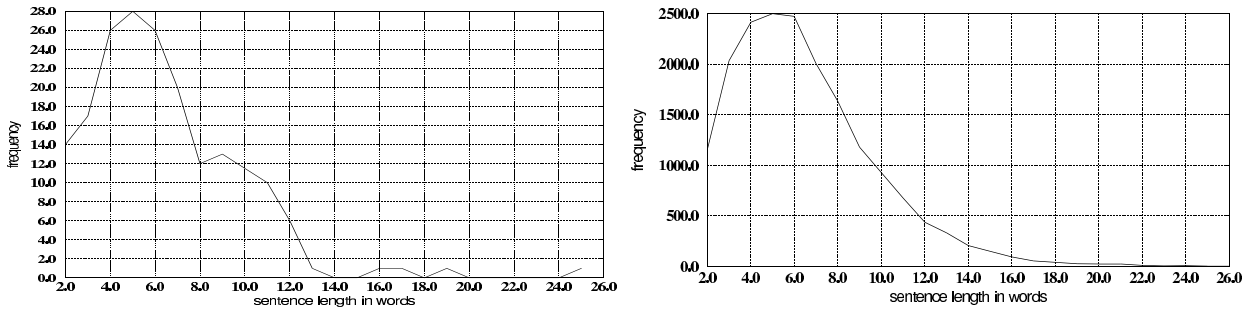


Figure 10: The frequency distribution for the test corpus (left) and the training corpus (right).

Test and Reference Corpus Our point of departure are 20.000 entries of the treebank. As the treebank consists of different articles representing different subject domains and different speech styles, we first have to shuffle the 20.000 sentences into a random order if we do not want to learn the whole set of trees at once, otherwise the system would be trained on text type A and tested on text type A, B and C. After shuffling the treebank using the Fisher-Yates shuffle, we randomly selected 1% of the sentences for testing purposes while the rest is used for training.[4] By this procedure we obtained 197 reference trees with an average

[4]Training and test corpora are kept small due to time constraints in the completion of this contribution.

length of 6.12 words used for automatic evaluation.

In addition we derive two kinds of test corpora (sentences to be parsed) from the reference corpus, one containing the lexical tags and one which does not. The first 50 sentences of the reference corpus are reproduced in the appendix A. The frequency distribution of the test/reference and training corpus are shown in Fig.10.

4.2 Evaluating different Weights

In a first set of experiments we evaluated the NN-retrieval. More precisely, we turned off all adaptation measures (except for the alignment which cannot be dispensed with) and compared the *coverage* with different measures attached to the indices. 3.000 trees have been learned for this experiment. The results in Table 4.2 reveal great differences between the association measures. Neither the competence nor the performance measure yield the best results. Instead those measures which normalize **moderately** for fq_p and $fq_{i,pos}$ perform best. However, no "ultimate" measure can be established as some of them perform better on frequent items and some perform better on infrequent items. The Cosine Coefficient will be used throughout the following experiments.

style	weight	derived from	f-score obtained
coverage	$W^p_{I_{pos}} = \frac{fq_{p,i,pos}}{fq_p \cdot fq_{i,pos}}$	mutual information (MI)	0.240
coverage	$W^p_{I_{pos}} = \frac{fq_{p,i,pos}}{fq_{i,pos}}$	conditional probability (CP)	0.280
coverage	$W^p_{I_{pos}} = \frac{fq_{p,i,pos}}{fq_p + fq_{i,pos}}$	Dice's coefficient	0.295
coverage	$W^p_{I_{pos}} = \frac{fq_{p,i,pos}}{\sqrt{(fq_p)} \cdot \sqrt{(fq_{i,pos})}}$	Cosine Coefficient	0.300

Figure 11: Comparison of Different Weights for NN-retrieval with 3.000 Training Sentences.

4.3 Contribution of Adaptation: 4.000

style	training	additional condition	recall	precision	f-score	time (sec.)
coverage	4.000	alignment only	0.335	0.316	0.325	0.47
coverage	4.000	+ struct. adapt.	0.356	0.336	0.346	0.43
coverage	4.000	+ comb. adapt.	0.340	0.321	0.331	0.51
coverage	4.000	+ recurs. adapt.	0.343	0.324	0.333	0.68
coverage	4.000	all adapt.	0.371	0.349	0.360	0.89
coverage	4.000	tagged input	0.398	0.3376	0.387	0.75
reliability	4.000+197	all adapt.	1	1	1	0.49

Figure 12: The Coverage, reliability and mean parsing time for 4.000 trained sentences with and without adaptation measures.

In order to evaluate the contribution of adaptation strategies we trained 4.000 trees and

established the coverage, reliability and the speed of the processing with and without adaptation steps. Results are presented in Fig.4.3 and Fig.13.

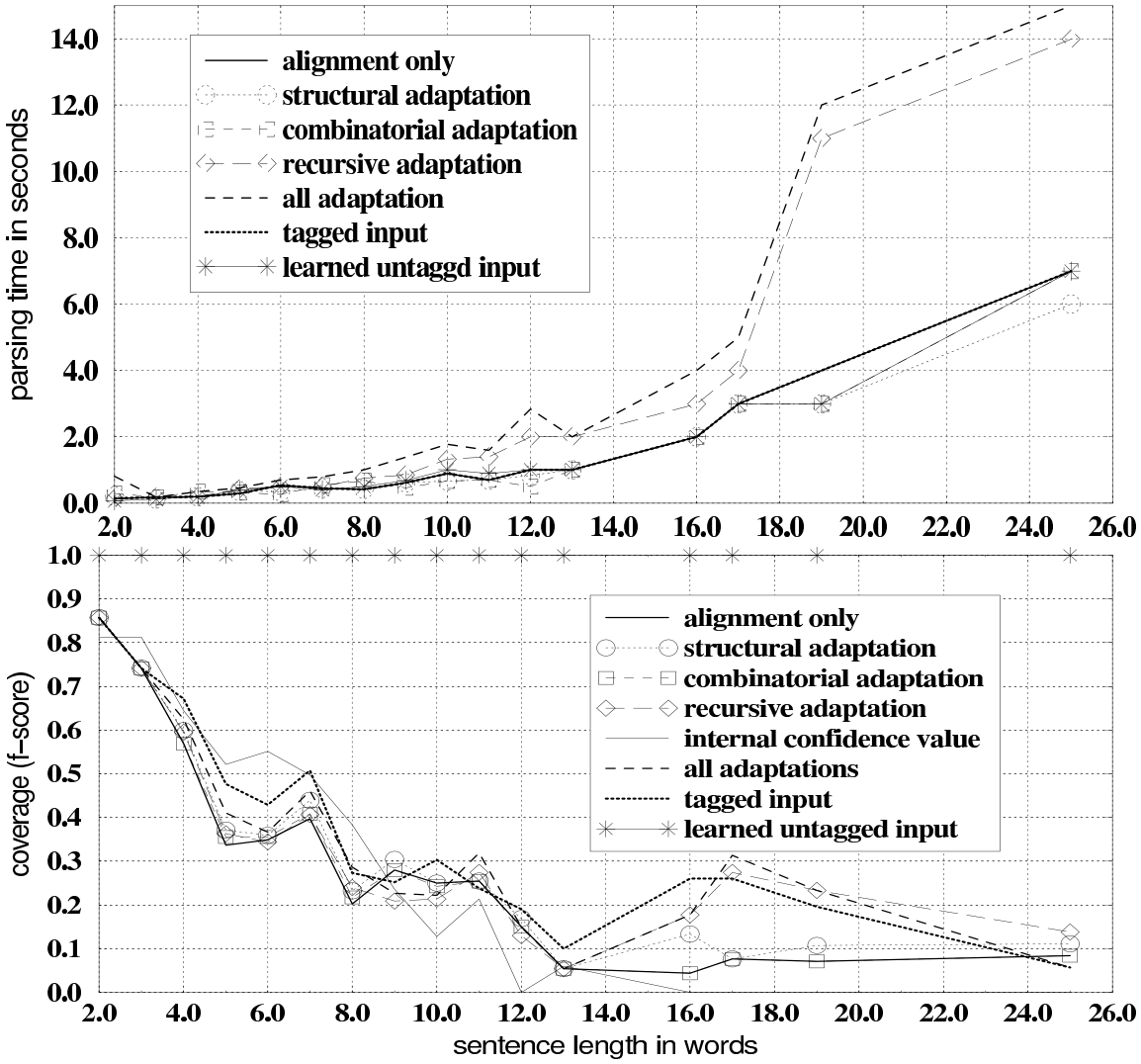


Figure 13: The Impact of Adaptation: Parsing time in seconds and the coverage and reliability measured as f-score for 4.000 training sentences.

As can be seen from the data, the adaptation is time consuming and the gain sometimes very limited. The *structural adaptation* performs best. The run-time behavior of the *recursive adaptation* is almost uncontrollable, as in the worst case, $2+3+4+\ldots+n$ words have to be parsed instead of n words (e.g. 209 words instead of 20), however with long sentences, the parsing results may improve considerably. Only if the quality of the match increases, the re-parsing is no longer that time consuming (or no longer performed), as can be seen from the tagged or learned input. The reliability of the parser is very high, i.e. the system can be perfectly trained for a closed domain application.

4.4 Contribution of Adaptation: 12.000

The above experiments are repeated with a training corpus of 12.000 sentences in order to illustrate the impact of more training data on the behavior of the parser.

style	training	additional condition	recall	precision	f-score	time (sec.)
coverage	12.000	alignment only	0.378	0.360	0.369	0.70
coverage	12.000	+ struct. adapt.	0.402	0.383	0.392	0.70
coverage	12.000	+ comb. adapt.	0.383	0.366	0.374	0.56
coverage	12.000	+ recurs. adapt.	0.401	0.378	0.389	0.71
coverage	12.000	all adapt.	0.423	0.4	0.411	0.92
coverage	12.000	tagged input	0.454	0.433	0.443	0.98
reliability	12.000+197	all adapt.	1	1	1	0.75

Figure 14: The Impact of adaptation for 12.000 training sentences.

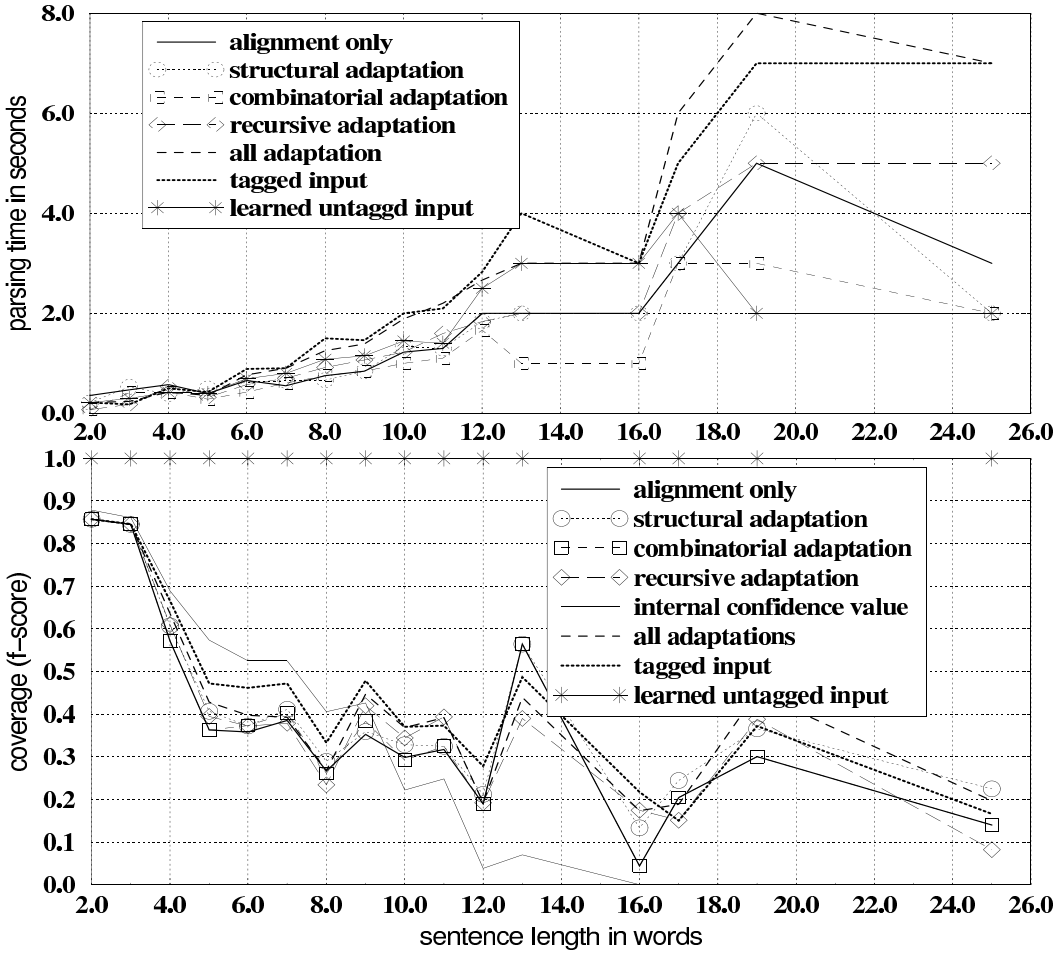


Figure 15: Parsing time in seconds and coverage and reliability for 12.000 training sentences.

Although parsing times increased with more training data, longer sentences require less time as a) the number of keywords grows and b) the recursive adaptation is applied less frequently.

4.5 All Training Data : 19.803

The above experiments are repeated with the complete training corpus. In order to judge the quality of the parse independent from the statistical deviation from the reference corpus refer to Appendix B, where parsing results are shown.

style	training	additional condition	recall	precision	f-score	time (sec.)
coverage	19.803	alignment only	0.399	0.389	0.394	1.10
coverage	19.803	+ struct. adapt.	0.423	0.413	0.418	1.13
coverage	19.803	+ comb. adapt.	0.402	0.392	0.397	1.13
coverage	19.803	+ recurs. adapt.	0.408	0.392	0.401	1.01
coverage	19.803	all adapt.	0.428	0.413	0.420	1.78
coverage	19.803	tagged input	0.424	0.413	0.419	1.57
reliability	12.000+197	all adapt.	1	1	1	1.29

Figure 16: The Coverage, reliability and mean parsing time for 19.803 training sentences.

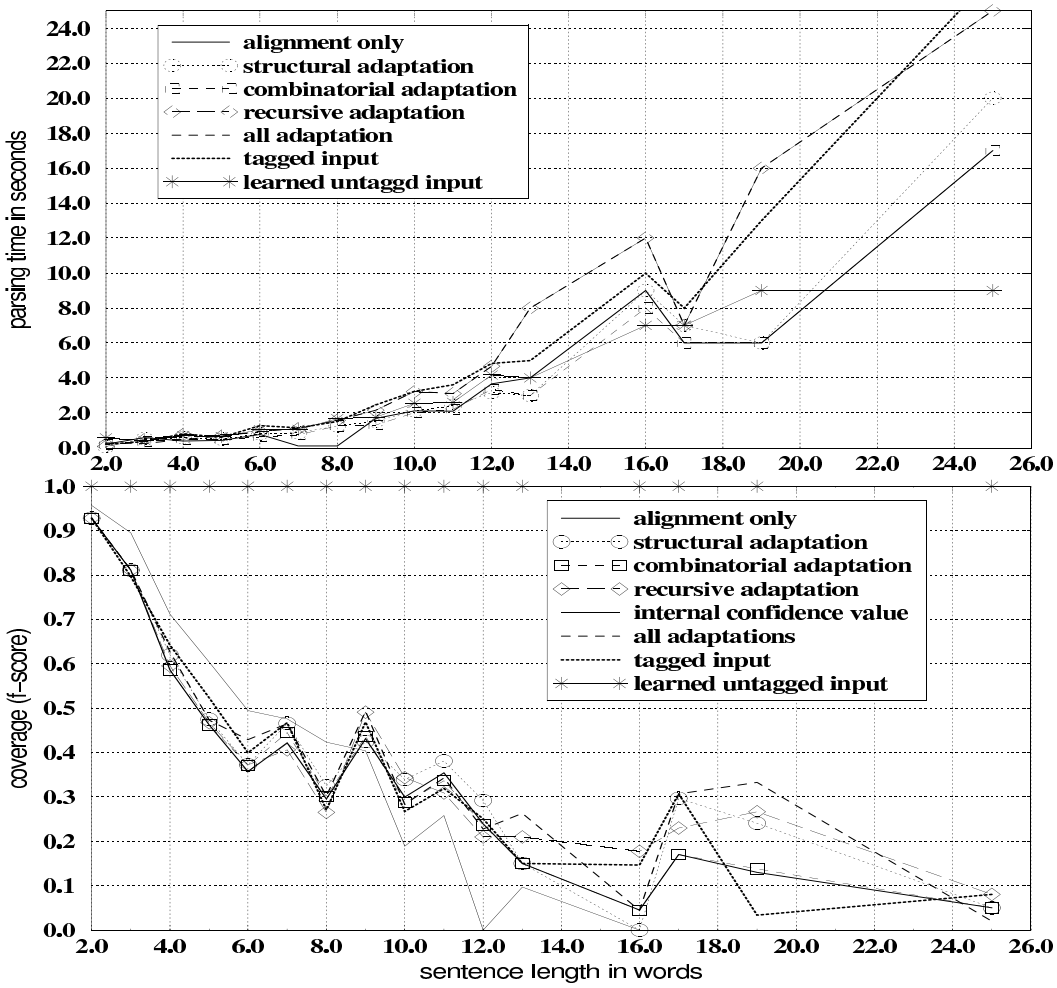


Figure 17: Parsing time in seconds and the coverage and reliability measured as f-score for 19.803 training sentences.

As can be seen in the data plots, the reliability remains high while the coverage increased

slowly with more training data. Some sentences however perform worse with more training data than in the previous experiments, showing that the selection of ETs is not optimal and still requires improvements. It may also be the case that sentences are over-indexed, i.e. no intersection of the keywords can be found.
The yet unmentioned internal confidence value based on the score of the NN-retrieval allows for a quite reasonable a-priori estimation of the outcome of the parsing results. As a consequence, this internal confidence value might be used in the future in order to trigger or block specific adaptation mechanism (for sentences they seem or do not seem to be adequate for). It might be used further for the interaction with other parsers or for the automatic acquisition of parsing trees.

5 Related Research

Since the first appearance of treebanks, there have been attempts to use such resources for parsing. A standard approach is to convert the subtrees represented in the treebank into stochastic phrase structure grammars. Such grammars generally outperform hand-written grammars.
Charniak (1996) derives a probabilistic context-free grammar from a 1.000.000 word hand-annotated corpus. The parsing is performed by a probabilistic chart parser. No lexical material except the lexical POS is integrated into the phrase structure rules. Using this strategy about 16.000 rules are derived from the corpus. 10.000 of them have a frequence of 1 and proved irrelevant for the parsing results. Using unlabeled parseval (cf. Fig.9) to evaluate the recall and precision of the parser scores between 80% and 90 % are achieved, depending on the size of the sentence.

The author mentions two drawbacks related to this approach. The first is the lack of lexicalization. Such grammars express only with difficulty relations between lexemes. In most cases, the lexemes are removed during the extraction of rules. The author hopes to obtain better results if more lexical information is integrated into the phrase structure rules. In fact, this claim is confirmed in Bod (1999). A higher degree of lexicalization equally might solve the second problem, i.e. the problem of over-generations. As the parser assigns a parse to almost every combination of POSs, it has difficulties to determine the best sub-parse from the chart with which the parse has to be continued, often resulting in an incorrect or failing parse. Parsing using all sub-parses stored in the chart seems impossible given the high redundancy of the grammar. Lexicalization thus seems to be the method to overcome the over-generation. Improvements to this standard approach are suggested in (Manning, 1997; Manning and Hinrich, 1999).

Tree Adjoining Grammars (TAGs) are equally extracted from treebanks (Xia, 1999; Chen and Vijay-Shanker, 2000) and used for grammar development and testing (Sarkar, 2000; Xia and Martha, 2000). It seems however, that the grammar extraction does not take full advantage of the treebanks, as the trees are split into elementary trees without retaining, in addition to the elementary trees, the unsplit (sub)tree, e.g "eat hot soup with a spoon". That the knowledge contained in these unsplit trees is critical for high quality parsing has been shown repeatedly (Rayner and Christer, 1994; Srivinas and Joshi, 1995; Bod, 1999; Streiter, 2000).

Data-oriented parsing (DOP) (Bod, 1992; Bod and Kaplan, 1998; Hoogweg, 1999) represents a parsing approach which promises to do away with the deficiencies in lexicalization. This approach consists of breaking the learned trees into all possible sub-trees and using all these sub-trees during the parsing. That parse which is obtained most frequently is chosen as final parse. It goes without saying that such an approach, as interesting as it may be, leads to a crazy computational complexity (Manning and Hinrich, 1999). Whether or not this approach becomes tractable by building up random samples of parses (Monte Carlo Parsing) and what the effect on the performance is, is a topic of current research. Practical systems following this approach cannot be expected in the near future.

Approaches which do not involve standard parsers while converting a treebank into a parser are hard to find. An interesting approach is presented by Lepage (1999). Sentences are analyzed with the help of analogy relations. Triples of ETs are extracted from a treebank the sentences to which they belong stand in a relation of analogy to the IS. The parse of the sentence is supposed to be the analogous tree derived by this triple. Although this method is extremely elegant, the system does not know on the basis of which ETs the analogy has to be made. As a consequence, a sentence may produce, depending on the size of the treebank almost as many parses as there are ETs. The selection of the best parse and the question whether this parse is a correct parse, as well as the efficiency of the algorithm are yet unsolved problems.

5.1 Summary and Conclusions

In this paper we present a approach to the analysis of natural language which does not follow any traditional parsing approach. Differently from standard approaches we do not attempt a local identification of forms, functions and meaning. We try instead to identify large sentence patterns by comparing the input sentence with examples from the treebank and concentrate on individual word meanings after the global structure has been established. We claim that this approach can maintain the highest degree of lexicalization while remaining efficient.

In order to identify the main sentence patterns, the system makes use of a fuzzy matching strategy. The inexact matches are worked over by a set of adaptation strategies. Thus, instead of considering mismatches to be harmful exceptions, they constitute a fundamental part of our approach, resulting in an extremely robust and adaptive system (there is no sentence which cannot be parsed).

Although parsing results are not especially good for long sentences in open domains, perfect parsing results are achieved in closed domains, independent of the size of the domain and independent of the size of the tree. This is not possible for any approach which during parsing re-composes subtrees, even for small domain areas.

The evaluation of different adaptation strategies has shown that the structural and recursive adaptation should be retained, as they improve the parsing results in open domains significantly. The combinatorial adaptation does not seem to be as performing. However, the combinatorial adaptation may provide a good interface for the cooperation with other (still hypothetical) parsers running in parallel. Such parsers running in parallel could fill the lattice with (partial) results until the example-based parser has completed the lattice and starts the evaluation.

Future work is manifold and overwhelming. First of all, the coverage has to be increased by optimizing scores, parameters and thresholds. Secondly, we intend to investigate experimentally the usefulness of this parser for sublanguage applications
Our claim that metonymies and maybe metaphors may be treated in this framework still awaits an experimental confirmation. We further intend to apply this parsing approach to a free word-order language, using the Russian corpus developed at IPPI (Boguslavskij et al., 2000). The integration of a bottom-up unknown word classifier as well as the cooperation with other parsers complete the set of future tasks.

6 Resources

The parser is written in Perl and has been developed under Linux. With minor changes the parser may run also under commercial operating systems. Experiments have been performed with 200 MHz CPU. The parser is a multi-tasking server which can be accessed via the TCP/IP. A demo-system and a download of a parsing-client can be found under http::/rockey.iis.sinica.edu.tw/oliver/parser.

Acknowledgment This work would not have been possible without the group of diligent tree-bank writers of CKIP. Special thanks to Chiu Chih-Ming, Luo Chi-Ching and Tsai Pi-Fang. Chen Laoshi has been a constant source of welcome objections and suggestions. Xiao-An finally supported me mentally. Without her patience and courage this work would have been neither started nor finished.

References

Rens Bod and Ronald M. Kaplan. 1998. A probabilistic corpus-driven model for lexical-functional analysis. In *COLING-ACL'98*. URL http://www/lfg.standord.edu/lfg/lfg-dop.

Rens Bod. 1992. Data oriented parsing (dop). In *COLING'92*.

Rens Bod. 1999. Extracting stochastic grammars from treebanks. In *Journées ATALA sur les Corpus annotés pour la syntaxe*. Talana, Paris VII.

I.M. Boguslavskij, Grigoreva S.A. Grigorev, N.V., L.L. Iomdin, M.V. Kreidlin, V.Z. Sannikov, and N.E. Frid. 2000. Annotierovannyj korpus russkix tekstov: koncepcija, instrumenty razmetki, tipy informacii. In *Proceedings of the Dialogue'2000 International Seminar in Computational Linguistics and Applications, Volume 2*, pages 41–47, Prodvino, Russia.

Eric Brill. 1993. Automatic grammar induction and parsing free text: A transformation-based approach. In *31rd Annual Meeting of the ACL*. URL http://www.cs.jhu.edu/ brill/acadpubs.html.

Ralf D. Brown. 1999a. Adding linguistic knowledge to a lexical example-based translation system. In *Theoretical and Methodological Issues in Machine Translation-99*. URL http://www.cs.cmu.edu/afs/cs.cmu.edu/user/ralf/pub/WWW.

Ralf D. Brown. 1999b. Generalized example-based machine translation. URL http://www.cs.cmu.edu/afs/cs.cmu.edu/user/ralf/pub/WWW/ebmt.html, April.

Michael Carl and Silvia Hansen. 1999. Linking translation memories with example-based machine translation. In *Proceedings of the MT-Summit'99*, Singapore. URL http://www.iai.uni-sb.de/.

Michael Carl. 1999. Inducing translation templates for example-based machine translation. In *Proceedings of the MT-Summit'99*, Singapore. URL http://www.iai.uni-sb.de/.

John Carroll, Ted Briscoe, and Antonio Sanfilippo. 1998. Parser evaluation: a survey and a new proposal. In *First International Conference on Language Resources & Evaluation, Granada, Spain*. URL http://www.cogs.susx.ac.uk/lab/nlp/carroll/carroll.html#cbs98.

Eugene Charniak. 1996. Tree-bank grammars. In *13th National Conference on Artificial Intelligence, AAAI-96*, pages 1031–1036.

John Chen and K. Vijay-Shanker. 2000. Automated extraction of TAGS from the Penn Treebank. In *IWPT 2000, Sixth International Workshop on Parsing Technologies*, Trento, Italy, 23-25 February.

Keh-Jiann Chen, Chi-Ching Luo, Zhao-Ming Gao, Ming-Chung Chang, Feng-Yi Chen, and Chao-Jan Chen. 1999. The CKIP Chinese Treebank. In *Journées ATALA sur les Corpus annotés pour la syntaxe*. Talana, Paris VII.

Keh-Jiann Chen. 1996. A model for robust Chinese parser. *Computational Linguistics and Chinese Language*, 1(1):183–204. URL http://rocling.iis.sinica.edu.tw/ROCLING/publish/clclp/v1/a6.htm.

Bróna Collins and Pádraig Cunningham. 1996. Adaptation guided retrieval in EBMT: A case-based approach to machine translation. In *Advances in Case-Based Reasoning*, LNAI, pages 91–104. Springer.

Walter Daelemans, Antal Van den Bosch, and Jakub Zavrel. 1999. Forgetting exceptions

is harmful in language learning. *Machine Learning*, special issue on natural language learning(34):11–43. URL http://ilk.kub.nl/papers.html.

Walter Daelemans. 1998. Abstraction is Harmful in Language Learning. In *Proceedings of NeMLaP (New Methods in Language Processing)*, pages 1–2, Sydney, January. URL http://ilk.kub.nl/papers.html.

Cheryl Grandy. 1999. The art of indexing. A white paper, Dynamic Systems Corporation, October. URL http://www.disc.com/home/artindex.html.

Lars Hoogweg. 1999. A data-oriented approach to lexicalize tree adjoining grammars. Report, University of Amsterdam, URL http://gene.wins.uva.nl/ lhoogweg/dop.html.

Chu-Ren Huang and Keh-Jiann Chen. 1992. A Chinese corpus for linguistics research. In *COLING'92*, Nantes.

Chu-Ren Huang, Keh-Jiann Chen, Li-ping Chang, and Hui-li Hsu. 1995. Zhōng yāng yán jīu yuàn pīng héng yǔ liào kù jiǎn jiè. In *Proceedings of ROCLING VIII.*

Ronald M. Kaplan. 1996. A probabilistic approach to LFG. keynote lecture held at LFG-workshop. Grenoble. ftp://ftp-lfg.stanford.edu/pub/lfg/.

Sung Dong Kim and Yung Taek Kim. 1995. Sentence analysis using pattern matching in English-Korean Machine Translation. In *International Conference on Computer Processing of Oriental Languages*, pages 199–206, Honolulu.

Yves Lepage. 1999. Open set experiments with direct analysis by analogy. In *Proceedings NLPRS'99 (Natural Language Processing Pacific Rim Symposium)*, pages 363–368, Beijing.

D. Lin. 1995. A dependency-based method for evaluating broad-coverage parsers. In *Proceedings of the IJCAI-95*, pages 1420–1425, Montreal, Canada.

Christoher D. Manning and Schütz Hinrich. 1999. *Foundations of Statistical Natural Language Processing.* The MIT Press, Cambridge, London. URL http://www.sultry.arts.usyd.edu.au/cmanning/.

Christoher D. Manning. 1997. Probabilistic models of language structure. URL http://www.sultry.arts.usyd.edu.au/cmanning/LAE, May.

Manny Rayner and Samuelsson Christer. 1994. Corpus-based grammar specification for fast analysis. In Agnas et al., editor, *Spoken Language Translator: First Year Report*, SRI Technical Report CRC-043, pg. 41-54. URL http://www.cam.sri.com.

C.J. Rijsbergen. 1979. *Information Retrieval.* Butterworth, London. URL http://www.dcs.ac.uk/Keith/Preface.html.

Matthew S. Ryan and Grahan R. Nudd. 1993. The Viterbi Algorithm. Research report cs-rr-238, Department of Computer Science, University of Warwick, Coventry,UK, URL http://www.dcs.warwick.ac.uk/pub/reports/rr/238.html.

Anoop Sarkar. 2000. Practical experiments in parsing using tree adjoining grammars. In *TAG+5, Fifth International Workshop on Tree Adjoining Grammars and Related Formalism*, pages 193–198, Paris. URL http://tagplus.linguist.jussieu.fr/acceptedPapers.html.

Satoshi Sekine and Ralph Grisman. 1995. A corpus-based probabilistic grammar with only two non-terminals. In *Forth International Workshop on Parsing Technology*, Prague, September. URL http://www.cs.nyu.edu/cs/projects/proteus/sekine.

B. Srivinas and Aravind K. Joshi. 1995. Some novel applications of explanation-based learning to parsing lexicalized tree-adjoining grammars. cmp-lg archive 9505023.

Oliver Streiter and Keh-Jiann Chen. 2000. Experiments in example-based parsing. In *Proceedings of the Dialogue 2000 International Seminar in Computational Linguistics and Applications, Volume 2*, pages 410–418, Prodvino, Russia. URL http://rockey.iis.sinica.edu.tw/oliver/publ.html.

Oliver Streiter and Pei-Yun Hsueh. 2000. A case-study in example-based parsing. In *ICCLC2000, 2000 International Conference on Chinese Language Computing*, Chicago. URL http://rockey.iis.sinica.edu.tw/oliver/publ.html.

Oliver Streiter, Michael Carl, and Leonid L. Iomdin. 2000. A virtual translation machine for hybrid machine translation. In *Proceedings of the Dialogue'2000 International Seminar in Computational Linguistics and Applications, Volume 2*, pages 382–394, Prodvino, Russia. URL http://rockey.iis.sinica.edu.tw/oliver/publ.html.

Oliver Streiter. 1999. Parsing Chinese with randomly generalized examples. In *NLPRS'99 Workshop on Multi-lingual Information Processing and Asian Language Processing (Natural Language Processing Pacific Rim Symposium)*, pages 114–119, Beijing, November. URL http://rockey.iis.sinica.edu.tw/oliver/publ.html.

Oliver Streiter. 2000. Reliability in example-based parsing. In *TAG+5, Fifth International Workshop on Tree Adjoining Grammars and Related Formalism*, pages 257–260, Paris. URL http://tagplus.linguist.jussieu.fr/acceptedPapers.html.

Fei Xia and Palmer Martha. 2000. Comparing and integrating tree adjoining grammars. In *TAG+5, Fifth International Workshop on Tree Adjoining Grammars and Related Formalism*, pages 265–268, Paris. URL http://tagplus.linguist.jussieu.fr/acceptedPapers.html.

Fei Xia, Palmer Martha, Xue Nianwen, Okurowski Mary Ellen, Kovarik John, Chiou Fu-Dong, Huang Shizhe, Kroch Tony, and Marcus Mitch. 2000. Developing guidelines and ensuring consistency for Chinese text annotation. In *Second International Conference on Language Resources & Evaluation*, Athenes, Greece.

Fei Xia. 1999. Extracting tree adjoining grammars from bracketed corpora. In *Proceedings NLPRS'99 (Natural Language Processing Pacific Rim Symposium)*, pages 398–403, Beijing.

Nianwen Xue, Huang Shizhe Xia, Fei, and Anthony Kroch. 2000. The bracketing guidelines for the Penn Chinese Treebank (draft II). Technical report, University of Pennsylvania, URL http://www.ldc.upenn-ctb/.

1) S(theme:VH12:漲升|Head:VG2:屬|range:Nad:强勢)
2) VP(result:Cbca:故|goal:NP(property:NP‧的(head:Ndabd:今日|Head:DE:的)|Head:Nad:成交量)|epistemics:Dbaa:應|manner:V‧的(head:VH11:温和|Head:DE:的)|Head:VC2:放大)
3) VP(companion:PP(Head:P63:跟|DUMMY:Nba:亞巴納爾)|Head:VH11:一樣)
4) S(contrast:Cbca:而|theme:NP(DUMMY:Nba:官某|Head:Cab:等)|time:Dd:又|Head:VC1:前往|complement:VA4:取款)
5) NP(property:NP(apposition:NP(DUMMY:NP(property:Nab:援力|Head:Nba:賴正全)|Head:Cab:等)|Head:NP(quantifier:Neu:五|property:Nab:人))|Head:NP(property:A:三等|Head:Nad:一級))
6) VP(manner:VH11:嚴重|Head:VC2:影響|goal:NP(DUMMY1:Nad:市容|Head:Caa:與|DUMMY2:NP(property:Nad:居家|Head:Nv4:安寧)))
7) S(evaluation:Dbb:難怪|theme:NP(Head:Nhaa:我們|quantifier:DM:幾位)|Head:VA11:走|location:PP(Head:P21:在|DUMMY:NP(property:Nab:路|Head:Ncda:上)))
8) NP(property:NP‧的(head:NP(quantifier:DM:這種|property:Nad:政治|Head:Nac:層面)|Head:DE:的)|property:Nv4:去勢|Head:Nac:現象)
9) S(DUMMY:S(evaluation:Dbb:到底|theme:Nv4:誠實|Head:V_11:是|range:Nep:什麼)|Head:Tc:呢)
10) VP(quantity:Daa:幾乎|time:Dd:已|Head:VC1:到|aspect:Di:了|goal:NP(predication:VP‧的(head:VH11:分秒必爭|Head:DE:的)|Head:Nad:地步))
11) S(theme:NP(property:NP(property:Nca:南非|property:A:特有|predication:VP‧的(head:VH11:新鮮|Head:DE:的)|Head:Nab:龍蝦)|Head:Nad:原味)|Head:VJ1:漫於|range:NP(property:Nab:齒|Head:Ncda:間))
12) VP(quantity:Dab:完全|Head:VJ2:漠視|goal:NP(quantifier:Nep:其|Head:Nv4:存在))
13) VP(Head:VL4:使|goal:Nv1:防守|theme:VP(negation:Dc:不|Head:VH14:出現|theme:Nac:漏洞))
14) S(agent:Nba:韓國隊|quantity:Daa:共|Head:VC31:擊出|theme:NP(quantifier:DM:十四支|Head:Nac:安打))
15) NP(predication:VP‧的(head:VP(Head:VH11(Head:VH11:明顯|Head:VH11:無用))|Head:DE:的)|Head:Nad:結構)
16) S(theme:Nhaa:我們|manner:VH11:乾脆|Head:VH11:奢侈|time:Dd:一下)
17) VP(Head:VK2:包括|goal:NP(property:Ncb:高中|Head:Nab:籃球))
18) S(theme:NP(agent:Nbc:尤|Head:Nv1:應對)|goal:PP(Head:P11:以|DUMMY:Nac:變化球)|Head:VI2:為主)
19) ADV(Head:Cbca:因而)
20) S(agent:Nhaa:他|Head:VC31:花|aspect:Di:了|theme:NP(quantifier:DM‧的(head:DM:一年|Head:DE:的)|Head:Nad:時間)|complement:VP(Head:VC2:苦練|goal:NP(Head:Nad(DUMMY1:Nad:拳擊|Head:Caa:和|DUMMY2:Nad:舉重))))
21) VP(time:Nac:終場|manner:PP(Head:P11:以|DUMMY:NP(Head:Neu(DUMMY1:Neu:十|Head:Caa:比|DUMMY2:Neu:六)))|Head:VC2:擊敗|goal:Nba:台電隊)
22) S(theme:NP(quantifier:DM:第一道|Head:Nab:菜)|time:Dd:終於|Head:VA11:來|particle:Ta:了)
23) S(agent:Nhaa:她|Head:VF1:下決心|goal:VP(manner:Dh:死命|Head:VA4:減肥))
24) S(agent:NP(property:Nca:蘇聯|Head:Nab:官員)|evaluation:Dbb:則|Head:VE2:表示)
25) S(theme:Nhaa:你|time:Ndabd:今天|Head:V_2:有|range:VP(Head:VJ3:沒有|range:Nv4:空))
26) NP(property:GP‧的(head:GP(DUMMY:Nab:長明燈|Head:Ng:裡)|Head:DE:的)|Head:Nba:莊光七)
27) VP(contrast:Cbca:否則|epistemics:Dbaa:會|Head:VH11:赤倉|duration:DM:一年)
28) S(theme:Nba:吉田修司|epistemics:Dbaa:是|time:PP(Head:P21:在|DUMMY:Ndaba:去年)|Head:VC1:升上|goal:NP(property:Nba:巨人隊|Head:Nab:一軍))
29) NP(quantifier:DM:這位|predication:VP‧的(head:VH11:勇敢|Head:DE:的)|property:Nab:司機|Head:Nab:老兄)
30) S(theme:Nhab:雙方|topic:PP(Head:P31:對|DUMMY:VP(manner:Dh:如何|Head:VC2:解決|goal:NP(property:Nca:波斯灣|Head:Nac:危機)))|time:Dd:仍|Head:VJ3:存有|range:NP(quantifier:Neqa:若干|Head:Nac:歧見))
31) PP(Head:P21:在|DUMMY:GP(DUMMY:NP(quantifier:Nep:此|Head:Ndabf:亂世)|Head:Ng:中))
32) VP(Head:VK1:希望|goal:NP(property:NP‧的(head:Ncc:世間|Head:DE:的)|Head:Nab:妻子))
33) S(reason:Cbaa:因|theme:NP(quantifier:Nep:此|Head:Ncb:區)|time:Ndabf:早期|Head:VJ3:受到|range:NP(theme:NP(DUMMY1:NP(property:Nab:山|Head:Nab:溪)|Head:Caa:和|DUMMY2:NP(property:Nv1:噴出|Head:Naa:泉水))|nominal:Str:的|Head:Nv4:温潤))
34) VP(Head:VL4:致使|goal:NP(DUMMY1:NP(property:Nac:我國|property:Nad:精密性|property:Nac:科技|property:Nv1:醫療|Head:Naeb:設備)|Head:Caa:和|DUMMY2:Naeb:藥品)|theme:VP(time:Dd:仍|deontics:Dbab:需|quantity:Neqa:大量|Head:VC31:進口))
35) PP(Head:P03:為了|DUMMY:VH11:一鳴驚人)
36) S(theme:NP(property:Naeb:融資|Head:Naeb:餘額)|Head:VP(Head:VP(Head:VH16:增加|quantifier:NP(quantity:Daa:近|Head:DM:九億元))|Head:VP(Head:VG2:為|range:DM:２５２億元)))
37) VP(time:PP(Head:P13:趁|DUMMY:S(theme:NP(property:Nhab:對方|Head:Nab:投手)|negation:Dc:不|Head:VH11:穩))|time:Dd:頻頻|Head:VD2:搶|theme:Nac:分)
38) VP(manner:PP(Head:P11:以|DUMMY:NP(Head:Neu(DUMMY1:Neu:一|Head:Caa:比|DUMMY2:Neu:一)))|Head:VH11:戰平)
39) VP(Head:VJ2:禁得起|goal:VP(time:NP(quantifier:DM:每個|Head:N(Head:Nac:星期|Head:Ndabf:假日))|quantity:Dab:都|Head:VC1:上|goal:Ncb:高爾夫球場)|particle:Ta:的)
40) S(agent:Nad:大盤|Head:VC31:燃出|theme:NP(property:VP‧之(head:VP(manner:Dh:更|Head:VH11:盛大)|Head:DE:之)|Head:Naa:火焰))
41) VP(goal:PP(Head:P07:將|DUMMY:Nab:書頁)|quantity:Neqa:全部|Head:VC2:打開)
42) NP(Head:Ndabc:九月份)
43) NP(predication:VP‧的(head:VP(Head:VC31:拿|aspect:Di:過|theme:NP(possessor:NP(property:Nba:士林電機|Head:Nab:董事長)|Head:Nab:名片))|Head:DE:的)|Head:Nab:人)
44) VP(deontics:Dbab:要|Head:VC2:慎選|goal:Nab:同伴)
45) S(topic:NP(quantifier:DM:兩個|Head:Nab:兒子)|agent:P02:被|Head:VC2:派任|goal:NP(predication:VP‧的(head:VP(evaluation:Dbb:並|negation:Dc:不|Head:VH15:合適)|Head:DE:的)|Head:Nac:位置))
46) VP(Head:VC2:撞|aspect:Di:了|goal:NP(quantifier:DM:個|quantifier:Neqa:滿|Head:Nab:頭)|complement:VH11:金星亂冒)
47) S(agent:NP(quantifier:DM:四十位|property:Nad:民主黨籍|Head:Nab:眾議員)|time:Ndabd:今天|Head:VE2:說)
48) PP(Head:P31[+part]:對|DUMMY:NP(property:Nba:長榮|Head:Nad:航空)|Head:P31[+part]:而言)
49) VP(time:DM:前八局|time:Dd:曾|frequency:NP(quantifier:Nes:有|Head:DM:三次)|Head:VC2:攻到|goal:Nab:三壘)
50) NP(DUMMY:Nca:中研院|Head:Cab:等)

Figure 18: Appendix A: The 50 first trees of the reference corpus.

1) S(time:Dd:漲升|Head:VG2:屬|range:VH11:強勢)
2) S(agent:NP(property:NP‧的(head:NP(contrast:Cbca:故|Head:Ncda:今日)|Head:DE:的)|Head:Nad:成交量)|Head:VC2:應|goal:NP(
predication:VP‧的(head:VH11:溫和|Head:DE:的)|Head:Nv1:放大))
3) S(theme:VP(Head:VC2:跟|goal:Nba:亞巴納爾)|Head:VH11:一樣)
4) S(contrast:Cbca:而|Head:VC2:官某|goal:VP(agent:NP(property:NP(DUMMY:Cab:等|Head:Cab:又)|Head:Nv1:前往)|Head:VA4:取款))
5) VP(Head:VC33:援力|theme:NP(property:Nba:賴正全|reason:Cab:等|quantifier:Neu:五|Head:Nab:人)|manner:A:三等|theme:Nad:一級)
6) VP(Head:VH11:獻重|theme:NP(DUMMY1:NP(property:Nv1:影響|Head:Nad:市容)|Head:Caa:與|DUMMY2:NP(theme:Nad:居家|Head:Nv4:安寧)))
7) S(evaluation:Dbb:難怪|theme:Nhaa:我們|quantifier:DM:幾位|Head:VC1:走|location:PP(Head:P21:在|DUMMY:NP(property:Nab:路|
Head:Ncda:上)))
8) S(theme:NP(quantifier:DM:這種|property:NP‧的(head:NP(property:Nad:政治|Head:Nac:層面)|Head:DE:的)|Head:Nv4:去勢)|
Head:VK2:現象)
9) NP(property:VP(evaluation:Dbb:到底(Head:VH11:誠實)|property:Nv4:誠實|Head:Nab:是)|Head:Tc:什麼)
10) S(quantity:Daa:幾乎|time:Dd:已|Head:VC1:到|aspect:Di:了|goal:NP(predication:VP‧的(head:VH11:分秒必爭|Head:DE:的)|
Head:Nad:地步))
11) S(theme:NP(property:Nca:南非|Head:NP(predication:VP‧的(head:VP(manner:A:特有|Head:VH11:新鮮)|Head:DE:的)|Head:Nab:龍蝦))|
theme:Nad:原味|theme:VJ1:漫於|Head:VA11:齒|goal:Ncda:間)
12) S(quantity:Dab:完全|Head:VJ2:漠視|goal:NP(quantifier:Nep:其|Head:Nv4:存在))
13) S(Head:VL4:使|goal:Nv1:防守|theme:VP(negation:Dc:不|Head:VH14:出現|theme:Nac:漏洞))
14) S(agent:Nba:韓國隊|quantity:Daa:共|Head:VC31:擊出|theme:NP(quantifier:DM:十四支|Head:Nac:安打))
15) NP(property:VP‧的(head:VP(manner:VH11:明顯|Head:VH11:無用)|Head:DE:的)|Head:Nad:結構)
16) S(agent:Nhaa:我們|Head:VH11:乾脆|goal:VP(Head:VH11:著侈|duration:DM:一下))
17) S(Head:VK2:包括|goal:NP(property:Nad:高中|Head:Nab:籃球))
18) S(quantity:Dab:尤|theme:Nv1:應對|goal:PP(Head:P11:以|DUMMY:Nac:變化球)|Head:VI2:為主)
19) conjunction(Head:Cbca:因而)
20) S(goal:Nhaa:他|manner:VP‧的(Head:NP(property:VC31:花|particle:Ta:了)|Head:DE:一年)|agent:PP(Head:P49:的|DUMMY:NP(
property:Nad:時間|Head:Nad:苦練))|agent:Nad:拳擊|Head:VC2:和)
21) S(agent:Nac:終場|manner:PP(Head:P11:以|DUMMY:NP(Head:Neu(DUMMY1:Neu:十|Head:Caa:比|DUMMY2:Neu:六)))|Head:VC2:擊敗|
goal:Nba:台電隊)
22) S(Head:VC2:第一道|goal:NP(apposition:Nab:菜|Head:Nba:終於)|complement:VP(Head:VA11:來|particle:Ta:了))
23) S(experiencer:Nhaa:她|Head:VF1:下決心|goal:S(manner:Dh:死命|Head:VA4:減肥))
24) S(agent:NP(property:Nca:蘇聯|Head:Nab:官員)|evaluation:Dbb:則|Head:VE2:表示)
25) S(theme:NP(apposition:Nhaa:你|Head:Nddc:今天)|Head:V_2:有|range:NP(quantifier:Nes:沒有|Head:Nab:空))
26) NP(property:GP‧的(head:GP(DUMMY:Nab:長明燈|Head:Ng:裡)|Head:DE:的)|Head:Nba:莊光七)
27) S(contrast:Cbca:否則|epistemics:Dbaa:會|Head:VH11(property:VH11:赤魯|Head:DM:一年))
28) S(agent:Nba:吉田修司|epistemics:Dbaa:是|time:PP(Head:P21:在|DUMMY:VP(time:Ndaba:去年|Head:VC1:升上))|
Head:VC31:巨人隊|goal:Nab:一軍)
29) NP(DUMMY:NP(quantifier:DM:這位|predication:VP‧的(head:VH11:勇敢|Head:DE:的)|Head:Nab:司機)|Head:Tc:老兄)
30) S(theme:Nhab:雙方|topic:PP(Head:P31:對|DUMMY:VP(manner:Dh:如何|Head:VC2:解決|goal:NP(property:Nca:波斯灣|
Head:Nac:危機)))|time:Dd:仍|Head:VJ3:存有|range:NP(quantifier:Neqa:若干|Head:Nac:歧見))
31) PP(Head:P21:在|DUMMY:GP(DUMMY:NP(property:Nep:此|Head:Ndabf:亂世)|Head:Ng:中))
32) NP(predication:VK1:希望|predication:VP‧的(theme:Ncc:世間|Head:DE:的)|Head:Nab:妻子)
33) S(agent:NP(reason:Cbaa:因|DUMMY:Nep:此)|Head:VE2:區|goal:NP(DUMMY1:NP(property:Ndabf:早期|Head:Naeb:受到)|Head:Caa:山|
DUMMY2:NP(property:NP‧的(head:NP(DUMMY1:Nab:溪|Head:Caa:和|DUMMY2:NP(property:Nv1:噴出|Head:Naa:泉水))|Head:DE:的)|
Head:Nv1:溫潤)))
34) S(Head:VL4:致使|goal:NP(property:NP(property:Nac:我國|property:Nad:精密性|Head:Nac:科技)|property:Nv1:醫療|
Head:Naeb:設備)|theme:VP(theme:NP(reason:Caa:和|Head:Naeb:藥品)|time:Dd:仍|Head:VK2:需|goal:Neqa:大量|goal:Nv1:進口))
35) PP(Head:P03:為了|DUMMY:VA12:一鳴驚人)
36) S(theme:NP(property:Naeb:融資|Head:Naeb:餘額)|manner:H16:增加|quantity:Daa:近|theme:DM:九億元|Head:VG2:為|
range:DM:２５２億元)
37) S(time:PP(Head:P13:趁|DUMMY:S(theme:NP(property:Nhab:對方|Head:Nab:投手)|negation:Dc:不|Head:VH11:穩))|time:Dd:頻頻|
Head:VD2:搶|theme:Nac:分)
38) VP(manner:PP(Head:P11:以|DUMMY:NP(Head:Neu(DUMMY1:Neu:一|Head:Caa:比|DUMMY2:Neu:一)))|Head:VH11:戰平)
39) VP‧的(head:VP(benefactor:PP(Head:P03:恭得起|DUMMY:NP(quantifier:DM:每個|Head:Nac:星期))|Head:VC31:假日|theme:NP(
quantity:Dab:都|property:Ncda:上|Head:Ncb:高爾夫球場))|Head:DE:的)
40) S(Head:VL1:大盤|goal:VP(Head:VC31:燃出|goal:VP(Head:VG2:更|range:NP(predication:VP‧的(head:VH11:盛大|Head:DE:之)|
Head:Naa:火焰))))
41) S(theme:PP(Head:P07:將|DUMMY:Nab:書頁)|Head:VG1:全部|range:Nv4:打開)
42) NP(Head:Ndabb:九月份)
43) NP(predication:VP‧的(head:VP(Head:VC31:拿|location:GP(DUMMY:NP(possessor:NP(degree:Dfa:過|Head:Nba:士林電機)|Head:Nab:董事長)|
Head:Ng:名片))|Head:DE:的)|Head:Nab:人)
44) S(Head:VE2:要|goal:VP(Head:VC2:慎選|goal:Nad:同伴))
45) S(theme:VP(DUMMY:NP(quantifier:DM:兩個|Head:Nab:兒子)|Head:Ng:被)|theme:VC2:派任|Head:VJ1:並|range:NP(property:VP‧的(
head:VP(negation:Dc:不|Head:VH15:合適)|Head:DE:的)|Head:Nac:位置))
46) S(Head:VC2:撞|aspect:Di:了|goal:NP(quantifier:DM:個|quantifier:Neqa:滿|property:Nab:頭|Head:Nv1:金星亂冒))
47) PP(Head:P43:四十位|DUMMY:S(agent:NP(apposition:NP(property:Nad:民主黨籍|Head:Nab:眾議員)|Head:Ndabd:今天)|Head:VE2:說))
48) S(Head:VC2:對|goal:NP(property:Nba:長榮|property:Nad:航空|Head:Ndabd:而言))
49) S(theme:NP(quantifier:DM:前八局|Head:Nbc:曾)|Head:V_2:有|range:NP(quantifier:DM:三次|predication:VC2:攻到|Head:Nab:三壘))
50) NP(DUMMY:Naa:中研院|Head:Cab:等)

Figure 19: Appendix B: Coverage with 19.803 training sentences.

The Improving Techniques for Disambiguating Non-alphabet Sense Categories

Feng-Long Hwang, Ming-Shing Yu, Min-Jer Wu

Department of Applied Mathematics, National Chung-Hsing University

Page 67 ~ 86

Proceedings of Research on Computational Linguistics Conference XIII (ROCLING XIII)

Taipei, Taiwan

2000-08-24/2000-08-25

The Improving Techniques for Disambiguating Non-alphabet Sense Categories

Feng-Long Hwang[Ψ], **Ming-Shing Yu, Min-Jer Wu**
Department of Applied Mathematics, National Chung-Hsing University,
Taichung 40227, Taiwan,
flhwang@mail.lctc.edu.tw, msyu@dragon.nchu.edu.tw

ABSTRACT

Usually, there are various non-alphabet symbols ("/", "**:**", "**-**", etc.) occurring in Mandarin texts. Such symbols may be pronounced more than one oral expression with respect to its sense category. In our previous works, we proposed the multi-layer decision classifier to disambiguate the sense category of non-alphabet symbols; the elementary feature is the statistical probability of token adopting the Bayesian rule. This paper adopts more features of tokens in sentences. Three techniques are further proposed to improve the performance. Experiments show that the proposed techniques can disambiguate the sense category of target symbols quite well, even with small size of data. The precision rates for inside and outside tests are upgraded to 99.6% and 96.5% by using more features of token and techniques.

Key Words**:** *Multi-layer decision classifier*, *Bayesian rule*, *word* sense *disambiguation, voting scheme, pattern table.*

1. Introduction

Various homographs or non-alphabet symbols in the Mandarin (but not limited to) occur frequently. The patterns containing these symbols may be pronounced with respect to its semantic sense. The **non-alphabet symbols** are defined: the symbols which are not the Mandarin characters (字) and may be pronounced different oral expressions. We call such phenomenon *oral ambiguity*.

The purpose of word sense disambiguation (WSD) is to identify the most possible category among candidate's sense category. It is important to disambiguate the word sense automatically for the natural language processing (NLP). Many works [Brown etc., 1991], [Fujii and Inue,1998] and [Ide and Veronis,1998], addressed WSD problems in the past.

In our previous works [Hwang, etc., 1999a; Hwang, etc., 1999b], we proposed the

[Ψ] Correspondence author.

multi-layer decision classifier (MLDC) to predict the sense category, in which the voting scheme is used to predict the final category. Even though the domains of sense in the paper just focus on three non-alphabet symbols, the proposed approach can be extended into other symbols in Mandarin and related ambiguity problems. The features of token and improving techniques described in this paper will be employed in the 2nd layer classifier. The main domain will focus on the improvements for the 2nd layer decision classifier. The model of our previous works is regarded as the baseline system. Comparing with the *baseline* model, the proposed features of token and techniques in this paper improve the performance of inside test from 97.8 to 99.6% and outside test from 93.0 to 96.6%.

The paper is organized as follows: related information and previous works will be described first. Section 3 elaborates the principal techniques for 2nd layer classifier in MLDC. Section 4 focuses on the evaluation for empirical features. Some improving techniques are proposed in section 5. The conclusions are presented in last Section.

2. Description of Related Works

In this Section, we first describe the applications of word sense disambiguation. The precious literatures on WSD and several methods, which are used to disambiguate the sense categories and classification problems of ambiguity, will be introduced next. Finally we will illustrate our previous approach.

2.1 Applications of Word Sense Disambiguation

The applications of WSD in natural language processing include the following domains:

- **Content and thematic analysis**
  Analyzing the distribution of pre-defined categories of words in text.
- **Information retrieval and extraction**
  When querying information, in a standalone system or Internet environment, the system should identify the real meaning for the query; excluding unnecessary data then correctly return desirable information among heterogeneous data.
- **Machine translation**
  We can first disambiguate the word sense categories, and then translate the word into correct semantic meanings associated with the target word.
- **Speech processing**
  Within the text analysis phase of TTS synthesis, the sense ambiguity of non-alphabet symbols or homographs should be resolved. The patterns containing such symbols can be translated into their oral expressions. The problem dealt with in our paper is very important for the precise speech output of TTS system.

2.2 Related Works

A lot of literatures have been published on word sense disambiguation in the past. They range from dictionary-based to corpus-based approaches. The former is dependent on the definitions of machine readable dictionary (MRD) [Veronis, etc., 1990] while the later usually rely only on the frequency of word extracted from the text corpus to construct the feature database [Schutze, etc.,1995]. Corpus-based approach adopts the co-occurrence of words which are extracted from the large text corpora to construct the feature database [Leacock, 1993] and provides the advantage of being generally applicable to new text, domains and corpus without the costly, error-prone parsing and semantic analysis. However, corpus-based approach also has some weakness: the corpus is always hard to collect and is time-consuming. The situation is so called "knowledge acquisition bottleneck" [Gale etc., 1992].

Based on the type of context in examples, the classifiers for word sense category use two contextual information: *local* and *topical context*. Hearst, etc. [1999] use local context with a narrow syntactic parse, in which the context is segmented into noun phrases, verb groups and other groups. Gale etc.[1992] developed a topical classifier, in which the Bayesian rule is used and the only information adopted is the co-occurrence of unordered word.

With respect to the contextual information, lexical information is formalized form of information involved in each surrounding word. Lee etc. [1997] adopt the discrimination score, based on maximum entropy of surrounding words in a sentence, to discriminate the word sense. Its precision rate is 80 % average.

Yarowsky [1994 and 1997] build a classifier using the local context cues within $\pm$ k windows for target word. A log-likelihood ratio is generated, which stands for the strength of each clue of local context. The decision will be made for matching sorted ratio sequence to decide the sense category of target word. The average performance ranges from 96% to 97% while the domain size of sense is only 2 for all ambiguous questions.

2.3 Our Previous Works

In contrast to *2*-gram, *3*-gram and *n*-gram language models, our previous paper [Hwang, etc., 1999a, 1999b] proposed an approach of multi-layer decision classifiers, which can resolve the category ambiguity of oral expression for non-alphabet symbols. A two-layer classifier has been developed. The first layer decision classifier can be viewed as decision tree based on the linguistic knowledge. Some impossible categories will be excluded while the remaining categories are all the possible categories. The second classifier employs a voting scheme to predict the final category with maximum probability score. The precision rates for inside and outside testing are 97.8% and 93.0% average.

3. The Principal Techniques

At first, the data set and sense categories for three target symbols are described. In 2nd decision classifier, a voting scheme, derived from Bayesian rule, is used to predict the portable sense category with maximum score.

3.1 Elementary Information of Data Set

The original data set is collected through different source, including: Academic Sinica Balance Corpus (ASBC), text files downloaded from Internet. ASBC is composed of 316 text files which contain 5.22M characters in Mandarin, English and other symbols totally [Huang, 1995; CKIP, 1995]. Only the sentence with such non-alphabet symbols will be extracted and appended into the empirical data set. Examples of three non-alphabet symbols *slash* (/), *colon* (:) and *dash* (-) are extracted and appended into our empirical data set. The sentence size of three non-alphabet symbols is 1115,1282 and 1685 respectively. The ratio of training and testing set is 4:1 appropriately. These sentences will be classified into different sense category with respect to target symbols. The sense categories and their oral expressions are listed in Tables 1-3. Less frequent (less than 1%) sense categories will be neglected.

Word segmentation paradigm is based on the Academia Sinica Chinese Electronic Dictionary (ASCED), which contains about 78,000 words. The words in ASCED are composed of one to 10 characters. Our principal rule of segmentation is first subject to maximal length of words and then to least number of words in a segmented pattern sequence. The priority scheme is that the segmented word sequence, which contains a word of maximal length, will be chosen. If two sequences have same maximum length of words, we compare further the total number of words in such sequences; then the sequence that is composed of least number of words will be chosen. The same segmentation's priority will be adopted within the training phase and testing phase.

There are several categories which speech for non-alphabet symbol "/" are silence; the duration for silence in prosodic parameter is still different to other senses. During the synthesis processing in TTS system, the duration with respective to its category will be varied and decided with respect to prosody needed. The numbers of token and sentence for three target symbols in our feature database are listed in Table 4.

Table 1: Seven sense categories and their related oral expressions of the target symbol "/".

	category	lexical patterns with non-alphabet symbol "／"	oral expression in Mandarin	data dis. (%)
1.	date	３／４ (March 4th)	三月四日	15.96
2.	fraction	３／４ (three fourth)	四分之三	8.88
3.	time(music)	３／４ (three four time)	四分之三拍	17.52
4.	path, directory	／ｄｅｖ／ｎｕｌｌ	斜線ｄｅｖ斜線ｎｕｌｌ	25.69
5,	computer words	Ｉ／Ｏ	Silence or 斜線	2.04
6.	production version	ＶＡＸ／ＶＭＳ	Silence (longer pause) or 斜線	5.52
7.	others	中／日／韓文(China/Japan/Korea)	Silence (longer pause)	25.45

Table 2: Five sense categories and its related oral expressions of target symbol ":".

Sense category	lexical patterns with non-alphabet symbol "："	oral expression in Mandarin	data dis. (%)
1. punctuation	優點：經濟省時	優點(silence)經濟省時	32.64
2. time	３：２０PM	下午三點二十分(three twenty PM)	11.63
3. versus	３：２０	三比二十(three versus twenty)	13.39
4. telephone	TEL：４２６４８５６	電話(silence)４２６４８５６	8.50
5. expression	教練表示：照常進行	教練表示(silence)照常進行	33.43

Table 3: Seven sense categories and its related oral expressions of target symbol "-".

Category	lexical patterns with non-alphabet symbol "－"	oral expression in Mandarin	data dis. (%)
1. figure, address	圖２－１(Figure 2-1)	圖２之１	7.64
2. interval	６－９月份營業收入	６至９月份營業收入	21.05
3. production	ｐｃ－ｃｉｌｌｉｏｎ	ｐｃ(silence)ｃｉｌｌｉｏｎ	17.01
4. computer term	Ｅ－Ｍａｉｌ	Ｅ(silence)Ｍａｉｌ	5.91
5. tel. fax	電話：４２６－４８５６	電話：４２６(silence)４８５６	21.91
6. hyphen	登記地點－圖書館前	登記地點(silence)圖書館前	24.22
7. minus	公式：Ｘ－２＝２０	公式：Ｘ減２等於２０	2.23

Table 4: numbers of token and sentence for three target symbols.

S_n	1	2	3	4	5	6	7	token no.	sentences no.
slash(/)	2906	1325	2471	3051	232	821	3772	14578	1115
colon(:)	4198	2564	2801	1464	3963	0	0	15028	1282
dash(-)	1568	5083	3481	1199	6004	4328	445	22103	1685

Table 5 displays several entries of token word "公車" in the feature database. Twelve entries of token "公車" are listed in Table 5, in which each entry is composed of 5 tuples (*w*, *l*, *count*, *s*, *pos*). Tag "*Na*" represents the *common noun*. The number in field *l* represents that the location of token *w* is preceding (negative number) or following (positive number) the target symbol respectively. It is possible that one token maybe occurs in more than two categories. Table 6 represents the tokens occurrence only considering the two location types: CH_L and CH_R. Field *l* represents the token's location preceding (CH_L) or following (CH_R) the non-alphabet symbols neglecting the token order. *p* and *f* in field *l* denote the location preceding and following the non-alphabet symbols. In our experiments, two location schemes will be evaluated in Section 4 and 5.

Table 5: the token word "公車" in feature database occurs in sense category 1,3,6

w	*l*	*count*	*s*	*pos*	*w*	*l*	*count*	*s*	*pos*
公車	-6	1	1	Na	公車	+5	4	3	Na
公車	-5	2	1	Na	公車	-7	1	6	Na
公車	-2	1	1	Na	公車	-5	2	6	Na
公車	-1	3	1	Na	公車	-2	1	6	Na
公車	-4	2	3	Na	公車	-1	2	6	Na
公車	-1	1	3	Na	公車	+5	8	6	Na

.

Table 6: The token "公車" occurs in feature database; without regarding the individual location.

W	*l*	*count*	*s*	*pos*
公車	p	7	1	Na
公車	p	3	3	Na
公車	f	4	3	Na
公車	f	5	6	Na
公車	f	8	6	Na

3.2 The Structure of MLDC

The function of multiple decision classifiers (MLDC) can be described as follow:

Suppose that *E* denotes the example with non-alphabet symbols, Φ_1 and Φ_2 denote the 1st and 2nd classifier respectively. And *possi_set* is the set containing all possible categories induced by 1st classifier. *TScore*(·) will compute the total score for a given category based on the voting criterion and statistical parameters schemes.

$$\Phi_1(E) = possi\_set, \tag{1}$$

$$\Phi_2(possi\_set) = \arg\max_{s_j \in possi\_set} TScore(s_j) \tag{2}$$

where s_j denotes the sense category for target symbols. *possi_set* contains all the possible sense categories. *TScore*(·) denotes the function of computing the total score for sense category.

3.3 The Statistical Decision Classifier with voting schemes

The segmentation task of testing phase adopts same criterions as that in training phase. A sentence will be divided into CH_L and CH_R, which are segmented into one to several basic tokens (Mandarin word or character). For each token in example, the probability of each category can be calculated and summed up based on the evidence (parameters found in feature database) respectively. It is called the ***voting scheme***.

Based on the *voting schemes,* each token in CH_L and CH_R have a statistical probability value, which looks like the voting suffrage, assigned to each category of the non-alphabet symbol. Like the political voting mechanism, the only candidate who gets the tickets in majority (maximum score in our approach) will become to be the predicted one. First the token unit we use is word with the location feature in CH_L or CH_R, in which the count of token occurred in same chunk (CH_L or CH_R) will be summed up with respect to the sense category. The scheme with character token will be analyzed in Section 4.

The prediction processing is based on the occurrence of each token inside training corpus for each category. The example E is composed of word sequence W and contains three parts: chunk-L(CH_L), non-alphabet symbol TS (target symbol) and chunk-R(CH_R). E, CH_L and CH_R can be expressed as:

$$E = CH_L + TS + CH_R \tag{3}$$

$$CH_L = w_{-m} w_{-(m-1)} \cdots w_{-j} \cdots \mathrm{w}_{-1} \tag{4}$$

$$CH_R = w_{+1} w_{+2} \cdots w_{+j} \cdots \mathrm{w}_{+n} \tag{5}$$

where m and n are the total number of tokens in CH_L and CH_R.

Let the category $s_{\max}$ be the sense category with maximum conditional probability of sense category s, given the word sequence W. By the definition of the Bayesian rule, $P(s|W)$ can be written as:

$$P(s|W) = \frac{P(s) \bullet P(W|s)}{P(W)} \tag{6}$$

MLDC needs to find the sense category $s_{\max}$ with maximum conditional probability $P(s|W)$. Thus:

$$s_{\max} = \max_s \frac{P(s) \bullet P(W \mid s)}{P(W)} = \frac{p(s) \bullet p(w_1, w_2, \cdots, w_M \mid s)}{p(w_1, w_2, \cdots, w_M)}, \tag{7}$$

where N and M *denote* the number of sense category of target symbol and token (word) in word sequence W.

Two problems should be considered for the Eq. (7). One is the fact that the probability of $p(w_1,w_2,\ldots,w_n|s)$ needs large memory and computation for the word sequence W. The other is the data sparseness because of the small amount of data set; which usually cause the situation of zero frequency. Each token w in word sequence W, under our voting scheme of preference scoring, can be regarded independent to other token. For the probability of sense category s given a token w, the Eq. (7) can be modified as:

$$Score(s \mid W) = \sum_i P(s \mid w_i) \tag{8}$$

where $P(s|w_i)$ is the probability of sense category s given a token w_i. Such probability can be

considered further as the score for token *w* to vote for sense category *s*. Eq. (8) can be expressed as:

$$P(s \mid w) = Score(s \mid w) = \frac{C(s,w)}{TC(w)} \tag{9}$$

where C*(s,w)* denotes the count of token *w* occurred in feature database for certain sense category *s*. *TC(w)* is the total count of token *w* in feature database for target symbol.

Score(s|w) is the relative frequency, which can be regarded as the score of token *w* voting to sense category *s* in our voting approaches. Eq. (9) satisfies the Bayesian rule and easy to understand intuitively. When computing the probability score of each word *w* for sense category *s*, we just need to use token count *C(s,w)* and total count *TC(w)* with respect to the sense category *s* and target symbol. So, the *Score*(*s*|*w*) can be computed easily for all tokens in the word sequence *W* of sentence. The probability can be regarded further as a score for each token in CH_L and CH_R to vote for each category of non-text symbol.

Referring to the Eq. (10), the score[1] $Score_L$ and $Score_R$ of each token in CH_L and CH_R voting for sense category s_j of non-text symbol can be computed as:

$$Score_L(s_j, w_{-i}) = \frac{C_L(s_j, w_{-i})}{TC_L(w_{-i})} \quad , \quad Score_R(s_j, w_{+i}) = \frac{C_R(s_j, w_{+i})}{TC_R(w_{+i})} \tag{10}$$

where $-1 \le -i \le -m$ and $+1 \le +i \le +n$, w_{-i} and w_{+i} are labeled as the token *w* in CH_L and CH_R . $C_L(s_j, w_{-i})$ and $C_R(s_j, w_{+i})$ are the count of token w_{-i} and w_{+i} occurred in CH_L and CH_R for the category s_j in feature database. $TC_L(w_{-i})$ and $TC_R(w_{+i})$ stand for the total count of w_{-i} and w_{+i} occurred in CH_L and CH_R , which can be computed as:

$$TC_L(w_{-i}) = \sum_{j=1}^{J} C_L(s_j, w_{-i}) \ , \qquad TC_R(w_{+i}) = \sum_{j=1}^{J} C_R(s_j, w_{+i}) \tag{11}$$

$$\sum_{j=1}^{J} Score_L(s_j, w_L) = 1 \ , \qquad \sum_{j=1}^{J} Score_R(s_j, w_R) = 1 \tag{12}$$

where J denotes the number of sense category for target symbol.

By definition of *score( )* above, $Score_L(s_j, w_{-i})$ and $Score_R(s_j, w_{+i})$ can be regarded as the relative frequency which the w_{-i} and w_{+i} will occur in the sense category s_j. As the result, our voting schemes are based on such probability value.

For the 2nd decision classifier in MLDC, the total score $TScore_L(\bullet)$ and $TScore_R(\bullet)$ for all the tokens in substring CH_L and CH_R of example *E* to vote for sense category s_j can be computed as:

---

[1] The resulting score of each token fall between 0 and 1, while it is possible that the accumulated scores of all tokens in sentence for certain sense category will be greater than 1.

$$TScore_L(s_j) = \sum_{-i=-1}^{-m} Score_L(s_j, w_{-i}) \ , \quad TScore_R(s_j) = \sum_{+i=+1}^{+n} Score_R(s_j, w_{+i}) \qquad (13)$$

In 2nd decision classifier, total score $TScore(\bullet)$ of all tokens in example E for each sense category are displayed as:

$$\underset{s_j \in possi\_set}{TScore(s_j)} = TScore_L(s_j) + TScore_R(s_j) \qquad (14)$$

$TScore(\bullet)$ will be used in Eq. (2) by the multi-layer decision classifiers to predict the final sense category s_j.

3.4 The Probability of Unknown token

Several well-known methods for probability of unknown words are described in [Su etc.,1996; Daniel et al.,2000]: *additive discounting*, *Good-Turing* and *Back-Off* . The principle reason is that there are a lot of tokens in natural language, usually more several ten thousands. New lexicons or tokens will be occurred in near future. Within natural language processing, it is so hard to collect all the words.

In our paper, the so-called unknown tokens can be considered that do not occur in our feature database, which have been generated in the training phase. It is so apparent that the distribution and total number of collected data set will affect the statistical parameters seriously, especially on the statistical models. Another situation is the data sparseness. The smoothing techniques can alleviate the problems. In this paper we use additive discounting and assign 0.5 to the count of unknown tokens.

4. Evaluations

The experiments with elementary approach and schemes are evaluated first. Two different scoring scheme adopted by our classifier are tested to decide which is better for WSD problems in this paper. We will compare the 2nd classifier in MLDC with the well-known language model. The location effectiveness with respect to different token unit (Mandarin word or character) is also evaluated in final subsection.

4.1 Evaluation for Two Scoring Schemes

At first, we will describe the voting scheme with winner-take-all scoring then compare such two scoring schemes. In contrast to the so-called preference-scoring scheme described in Section 4.3, the voting scheme with *winner-take-all* scoring adopts a different scoring rule. Ho Lee etc [1997]. Lee employed the *winner-take-all* scoring scheme to word sense disambiguation, without comparison between these two schemes in his paper. Lee's precision rate was 80% average.

For each token in sentence, $Score_L(s_{j^*}, w_{-i})$ and $Score_R(s_{j^*}, w_{+i})$ will be assigned the

score 1 to sense category s_j^* for token w_{-i} and w_{+i} and 0 to all the other sense categories. Eq. (10) should be rewritten as:

$$Score_L(s_{j^*}, w_{-i}) = \begin{cases} 1 & \text{if } s_{j^*} \in possi\_set \text{ and } Score_L(s_{j^*}, w_{-i_1}) = \arg\max_{j=1,2,...J}(Score_L(s_j, w_{-i})) \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

$$Score_R(s_{j^*}, w_{+i}) = \begin{cases} 1 & \text{if } s_{j^*} \in possi\_set \text{ and } Score_R(s_{j^*}, w_{+i}) = \arg\max_{j=1,2,...J}(Score_R(s_j, w_{+i})) \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

where sense category (s_{j^*}) is with respect to the category of which the $Score_L(s_{j^*}, w_{-i})$ and $Score_R(s_{j^*}, w_{+i})$ have the maximum score among all categories for w_{-i} and w_{+i}. Based on the voting scheme with winner-take-all scoring, Eqs. (10) – (14) should not be modified.

In case that several sense categories have the maximum score for token w, Eqs (15) and (16) should be revised. The total probability score 1 for token w will be shared by these sense categories. It means that the total score 1 will be divided by the number of sense categories with same maximum score.

The first parameter to be evaluated is the scoring scheme for each token. Figure 6 displays an example of the accumulated score for 5 categories using two different scoring methods: preference and winner-take-all scoring on the Eqs (15) and (16). The example (E1) contains 15 individual tokens (including symbol ":"). Sense category *time* (s_2) gets the maximum score 6.92 in Figure 1. Similarly, it still gets maximum score 7.0 by using the winner-take-all scoring.

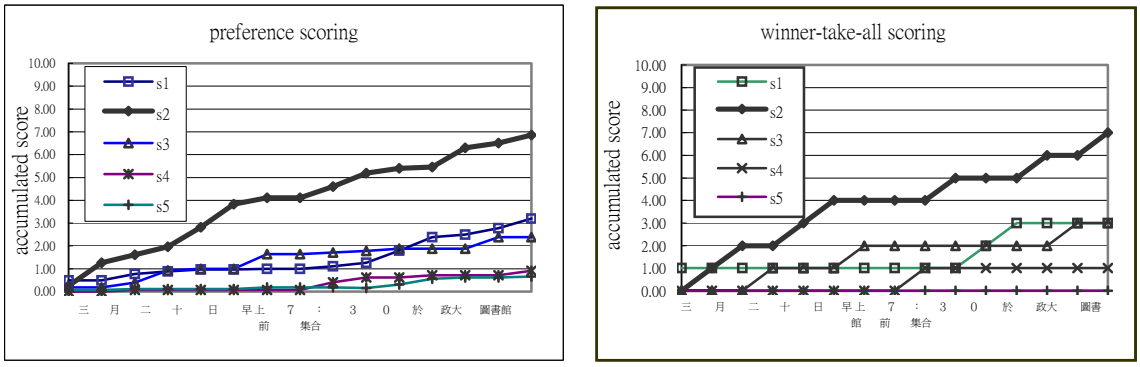


Figure 1: (left) accumulated score of categories for non text symbol ":" based on the preference scoring ; category *time* (s_2) gets maximum score 6.92. (right) based on winner-take-all scoring; the category *time* (s_2)

(E1) 三 月 二 十 日 早 上 7 : 3 0 於 政 大 圖 書 館 前 集 合 。

s_n / scoring scheme	1 punctuation	2 time	3 versus	4 telephone	5 expression	prediction
winner-take-all	3.0	**7.0**	3.0	1.0	0.0	correct
preference	3.1	**6.9**	2.3	0.9	0.8	correct

The sense category *time* (s_2) in (E1) gets maximum score in two scoring schemes, however, some other examples may not hold yet. Especially, while top 2 scores are so close it is possible that the sense category with second maximum score will precede the first category with maximum score by employing different scoring scheme. For instance, as shown in example (E2), the sense category *date* (s_1) got the maximum score and is predicted as the final category by using the winner-take-all scoring scheme. Instead of such scoring scheme, we use the preference scoring to predict the category and the result is correct. In fact, the substring "1/3" means "one third". This is an example that winner-take-all scoring makes a wrong prediction while preference scoring can finds the correct sense category. The scores for each sense category[2] are listed below example (E2).

(E2) 價 格 比 台 灣 便 宜 約 1／3 左 右 ，

S_n / scoring	1 date	2 fraction	3 time	4 directory	5 computer term	6 version	7 others	prediction
winner-take-all	**4.0**	0.0	3.0	0.0	1.0	1.0	0.0	incorrect
preference	3.1	0.4	**3.2**	0.4	0.9	0.7	0.4	correct

Table 7 lists the performances with two voting schemes: preference and winner-take-all scoring. Obviously, the former is superior uniformly to the later on both inside and outside testing for three symbols. So we adopt the voting scheme of preference scoring, excluding winner-take-all scoring, for all following experiments. Note that the 2nd decision classifier in MLDC, based on the voting scheme of preference scoring with Mandarin word's token, is regarded as the *baseline* model in this paper. As shown in Table 7, the net results are enhanced up 5.5% and 5.9% for inside and outside testing respectively.

Table 7: The performance of the 2nd decision classifier (baseline) in MLDC; employing two scoring scheme.

scoring scheme / precision rate(%)	preference		winner-take-all	
	inside test	outside test	inside test	outside test
"／"	99.2	94.6	92.9	84.8
"："	95.7	91.1	91.5	84.1
"＿"	96.8	85.7	90.8	83.5
average (net)	**97.2(+5.5)**	**90.0(+5.9)**	91.7	84.1

4.2 Comparing the 2nd Classifier with *n*-gram Models

In this Section, we will compare baseline defined in previous subsection with the *n*-gram

[2] All the sense categories for three target symbols discussed in our paper are displayed in Tables 1-3.

(*n*=1, 2 in this experiments), widely used in various domains of natural language processing. The base line model displays attractive empirical results.

Table 8 indicates the performance of three models: baseline with voting scheme, *uni*-gram and *2*-gram, on the same testing data set without employing the 1st layer decision classifier or other techniques. Comparing the *2*-gram with *uni*-gram, it is so apparent that the former is superior to the latter. The average net results for inside and outside test are 1.3% and 4.3% respectively.

We observe further the performance between baseline and *n*-gram. The minimum difference between is +1.4% for outside testing of target symbol ":". The baseline is superior to *2*-gram model for all target symbols. The average net results for inside and outside test are 0.5% and 4.7%.

Because of the data sparseness and small size of data set on our WSD problem, there are more unknown tokens for *n*-gram model than that for baseline. The performance for outside testing of *n*-gram is upgraded by baseline model for three target symbols. The ratio of unknown tokens (words) for three target symbols: 11.8%, 15.3% and 19.3%. The more the unknown tokens appear, the lower the performance is. The size of unknown tokens will affect seriously the performance of *n*-gram model. The zero count of token leads to the degradation for *n*-gram.

Table 8: Comparisons between our *base line* and *n*-gram (n=1,2).
The numbers in parenthesis denote the net performance comparing *base line* with *2*-gram.

scheme / symbols	inside test			outside test		
	base line	*uni*-gram	*2*-gram	*baseline*	*uni*-gram	*2*-gram
" ／"	99.2(+0.3)	97.6	98.9	94.6(+2.0)	90.5	92.6
" : "	95.7(+0.5)	92.2	95.2	91.1(+1.4)	79.9	89.7
"—"	96.8(+0.7)	95.9	96.1	85.7(+9.2)	74.3	76.5
average (net)	**97.2(+0.5)**	95.4	96.7	**90.0(+4.7)**	81.0	85.3

4.3 Merging Two Layer Classifiers Together

In addition to our baseline model, we will analyze further the effectiveness of the 1st classifier in MLDC. Two classifiers in MLDC could be merged together to improve the prediction rate.

For instance, example (E3) shows the effectiveness of merging the 1st layer classifier into baseline (the 2nd layer classifier). Exploiting the 1st classifier to exclude some impossible

categories first. As shown in example (E3), the sense category with maximum score (2.4), predicted by using the 2nd layer classifier with voting scheme only, is *date* (s_1) and it is apparent that the prediction is incorrect. The number of w_{+1} token (32) in pattern "3/32" is larger than 31, which is the maximum number of date. Therefore sense category *date*[3] was excluded for target symbol "/" by the 1st layer classifier. However, the category *music time* (s_3) with second maximum score (1.8) was predicted as the final one among all remained categories correctly by the 2nd layer classifier with voting scheme.

(E3) 演 奏 的 曲子 是 3／32 拍 且 爲 D 大 調 。

S_n / merging	1 date	2 fraction	3 time	4 directory	5 computer term	6 version	7 others	prediction
2nd classifier only	**2.4**	1.3	1.8	1.1	0.3	0.7	0.4	incorrect
merging two classifier	2.4*	1.3	**1.8**	1.1	0.3*	0.7	0.4*	correct

ps. * denotes the sense category was excluded by the 1st layer decision classifier.

The performances are attractive and listed in Table 9. As shown, the final results for outside testing is 97.8, 95.6 and 92.1 for three symbols respectively by combining the 1st and 2nd classifier with voting scheme of preference scoring in 2nd classifier. The numbers in parenthesis are the net results. The average net results by merging two classifiers are upgraded 0.5% and 4.5% (referring to Table 8 and Table 10).

Table 9: The effectiveness of merging the 1st and 2nd decision classifiers

	merging 1st classifier ?	inside testing	outside testing
"／"	without merging	99.2	94.6
	merging	**99.5(+0.3)**	**97.9(+3.3)**
"："	without merging	95.7	91.1
	merging	**98.3(+2.6)**	**95.6(+4.5)**
"_"	without merging	96.8	85.7
	merging	**98.4(+1.6)**	**92.1(+5.4)**
average	merging	**97.7**	**94.5**

4.4 Evaluation for the Effect of Word's Location

In previous Section, the location of each token is just labeled two types: preceding (*p*) and following (*f*) the target symbol. While the count for each token was statistically accumulated, we just consider whether the token is located within the chunk-L (CH_L) or chunk-R (CH_R) of

[3] In fact, the decision tree excludes three sense categories: *date*, *computer term* and *version*.

sentence. Will the performance be improved by considering further the individual location of each token in CH_L (w_{-i}) and CH_R (w_{+i})? In this Section, the effect of individual location for each token (word) will be evaluated further.

In this Section Token unit is still Mandarin word. Instead of the two chunk types described previously, each token is labeled with the individual location in CH_L and CH_R, in which the count of each token occurred in same location will be summed up with respect to the sense category. So the technique is the *word-based* scheme with individual location.

The Eqs. (10)-(12) can be changed as follow:

$$Score_L(s_j, w_{-i}) = \frac{C_{-i}(s_j, w_{-i})}{TC_{-i}(w_{-i})} \quad Score_R(s_j, w_{+i}) = \frac{C_{+i}(s_j, w_{+i})}{TC_{+i}(w_{+i})} \tag{17}$$

$$TC_{-i}(w_{-i}) = \sum_{j=1}^{J} C_{-i}(s_j, w_{-i}) \qquad TC_{+i}(w_{+i}) = \sum_{j=1}^{J} C_{+i}(s_j, w_{+i}) \tag{18}$$

$$\sum_{j=1}^{J} Score_L(s_j, w_{-i}) = 1\ , \qquad \sum_{j=1}^{J} Score_R(s_j, w_{+i}) = 1 \tag{19}$$

where i is the location of word with respect to the non-text symbol , $-m <= -i <= -1$ and $1<=+i<=n$. $C_{-i}(s_j, w_{-i})$ and $C_{+i}(s_j, w_{+i})$ are the count of word w_{-i} and w_{+i} with the location $-i$ and $+i$ occurred in feature corpus for sense category s_j respectively.

$TC_{-i}(w_{-i})$ and $TC_{+i}(w_{+i})$ are the total count of word w_{-i} and w_{+i} occurred in the location $-i$ and $+i$ in feature database respectively

Let's take a look at the example (E4), the sense category (*date*) is incorrectly predicted based on the chunk scheme whereas correctly predicted on individual location of each token.

(E4) 曹 錦輝還 有機會 在 11／20 代 表 台灣大聯 盟 與統一 隊比 賽。

s_n token location	1 date	2 fraction	3 time	4 directory	5 computer term	6 version	7 others	prediction
chunk	2.9	1.7	**5.2**	1.2	0.2*	1.8*	4.0	incorrect
individual	**2.4**	0.5	2.2	0..5	0.2*	0.4*	0.2	correct

Comparing two schemes of token (word) with individual and two chunks' location, the net precision rates of outside testing are 0.6%, 1.5% and –0.3% for three target symbols respectively. As Shown the Table 10, the former is average superior to the later, in which the sentence is divided into two chunks (CH_L *or* CH_R). Referring to the accumulated score for correct predicted sense category, although the rate of unknown words token in data set reaches about 45%, the former still make the prediction efficiently. However, it is easier for

the techniques with voting scheme, which identify a half of total tokens in sentence, to make the correct prediction. The net precision rates for inside and outside testing are 0.2 and 0.6.

Table 10: The comparison of two location schemes for each token.

	inside testing		outside testing	
	individual	chunk	individual	chunk
“ ／”	**99.3(-0.2)**	99.5	**98.6(+0.7)**	97.9
“ : ”	**99.2(+0.9)**	98.3	**97.1(+1.5)**	95.6
“—”	**98.2(-0.2)**	98.4	**91.8(-0.3)**	92.1
average	**98.9(+0.2)**	**98.7**	**95.1(+0.6)**	**94.5**

4.5 Evaluation for Effect of Token Unit

Until now, the sentence will be divided into two chunks: chunk-L(CH_L) and chunk-R(CH_R), which are in the left and right side of target symbol *TS* in sentence. Such chunks will be segmented into one to several words based the ASCED and segmentation scheme. In Mandarin Vocabulary, there are about 70000 frequent Mandarin words, which are composed of one to ten characters. For example, the number for 1-character token (Mandarin word) is 7522 and 48315 for 2-character token (Mandarin word) in ASCED while just 13053 for frequent Mandarin characters. It is apparent that segmented sentence will generate more unknown tokens for the same data set. The more unknown tokens are in sentence, the less precision rate will be. The process of word segmentation may generate possible mistake, which will also degrade the performance of prediction. Usually the situation becomes serious if the data set is sparse or volume of sentence is small.

In this section, the sentence will not be segmented so each character in sentence is the voting token. The location of each character will be considered same as described in previous section. The token unit is character with the individual location in CH_L or CH_R, in which the count of each character occurred in same chunk (CH_L or CH_R) will be summed up with respect to the sense category. So the technique is the *character-based* scheme with individual location. Example *E* is still composed of three parts: CH_L, *TS* and CH_R. Each chunk may comprise one to several characters. Note that the foreign words (such as: *IBM*, *DR.*, *Windows*, etc.) within chunk will be regarded as a token.

$$\begin{aligned} E &= CH_L + TS + CH_R \\ CH_L &= c_{-m} c_{-(m-1)} \cdots c_{-j} \cdots c_{-1} \\ CH_R &= c_{+1} c_{+2} \cdots c_{+j} \cdots c_{+n} \end{aligned} \tag{20}$$

where *c* denotes the individual character in CH_L and CH_R and *m*, *n* the number of characters

in CH_L and CH_R respectively. The Eqs. (10)-(12) of probability scoring can be rewritten as:

$$Score_L(s_j, w_{-i}) = \frac{C_{-i}(s_j, c_{-i})}{TC_{-i}(c_{-i})}, \quad Score_R(s_j, w_{+i}) = \frac{C_{+i}(s_j, c_{+i})}{TC_{+i}(c_{+i})} \tag{21}$$

$$TC_{-i}(c_{-i}) = \sum_{j=1}^{J} C_{-i}(s_j, c_{-i}), \qquad TC_{+i}(c_{+i}) = \sum_{j=1}^{J} C_{+i}(s_j, c_{+i}) \tag{22}$$

$$\sum_{j=1}^{J} Score_L(s_j, c_{-i}) = 1\ , \qquad \sum_{j=1}^{J} Score_R(s_j, c_{+i}) = 1 \tag{23}$$

where i is the location of character with respect to the non-text symbol , $-m <= -i <= -1$ and $1 <= +i <= n$. $C_{-i}(s_j, c_{-i})$ and $C_{+1}(s_j, c_{+i})$ are the count of character c_{-i} and c_{+i} occurred in feature corpus with the location $-i$ and $+i$ for sense category s_j respectively.
$TC_{-i}(c_{-i})$ and $TC_{+i}(c_{+i})$ are the total count of character c_{-i} and c_{+i} occurred with the location $-i$ and $+i$ in feature corpus respectively

The total score $TScore_L(\bullet)$ and $TScore_R(\bullet)$ for all individual characters of CH_L and CH_R in example E to vote for sense category can be computed like Eqs. (13) and (14). The method will be regarded as the character-based approach with location scheme.

Until now, the adopted token unit of sentence is Mandarin *word*. There are some possible errors occurred during the segmentation process for generating the token (word). Based on the character[4] token unit with location scheme, there are fewer unknown token. The example (E5) in our data set is divided into two chunks, in which the individual token is the character without needing the word segmentation. The characters in CH_L will be labeled with location $-m$~-1 and the characters in CH_R labeled with +1~+n. (E5) is an example in which the correct sense category can't be predicted by using the scheme with word token, while it can be correctly predicted by using character as token.

(E5) 結 果 １０／１０ 那 天 桃 園 縣 建 築 師 再 來 認 定 時 ，[5]

s_n / location/tioken	1 date	2 fraction	3 time	4 directory	5 computer term	6 version	7 others	prediction
individual/word	2.4	0.5	**3.5**	0.2	0.2*	0.4*	0.2	incorrect
individual/character	**1.3**	0.8	0.7	0.2	0.1*	0.1*	0.3	correct

Intuitively, in natural language processing of Mandarin, the token unit used is usually word, which is the basic unit containing complete and useful semantic information. Instead,

[4] The Mandarin characters we use is 13053, which are collected in the BIG-5 character set.
[5] In contract to our previous example, each Mandarin character here is regarded as a token, without word

why the performance for *character* tokens is superior to that for *word* tokens both with individual location?

Depending on our observations, there are three following reasons with respect to such phenomenon. First, it is not easy for the process of word segmentation to generate the most portable word sequence *W*. The second reason is the data sparseness; the situation exists in our WSD problem and more unknown tokens will happen. The third, related to the unknown token, is the token unit. The number for Mandarin character is approximately 13,000 whereas 70,000 for Mandarin word. It is obvious that adopting word's token will lead to more unknown tokens than that of character's token. Such situation will affect the performance. As described below, suppose that a two-character word "昨天(yesterday)" occurred with specific location in our feature database. Now a token "今天(today)" in a testing example occurs, labeled by same location of token "昨天", and will be still regarded as a unknown token based on the token with scheme of individual location. However, the token "昨天" can further be divided into two characters: "昨" and "天". The second character of word "昨天" and "今天" is both "天". So character "天" is a known token and can provide the statistical information based on the character token with individual location. Referring to Table 10, the average precision rates in Table 11 are upgraded 0.5% and 0.4% for inside and outside testing obtained from the individual location for each token (character).

Table 11: Two token units: *word* and *character*. Each token is labeled by individual location.

	inside testing		outside testing	
	character	word	character	word
" ／"	**99.6(+0.3)**	99.3	**98.3(-0.3)**	98.6
" : "	**99.6(+0.4)**	99.2	**98.1(+1.0)**	97.1
"—"	**99.2(+1.0)**	98.2	**92.4(+0.6)**	91.8
average	**99.4(+0.5)**	98.9	**95.5(+0.4)**	95.1

Currently, the elementary experiments have been implemented and several schemes in our proposed approach were evaluated. The best performance for WSD problem based on such empirical parameters can be achieved. In summary, that are the following empirical features: preference scoring, merging the 1st and 2nd decision classifier together, individual location (-*m*~+*n*) of token, character token. The precision rates, obtained by using the techniques above, of outside testing are 98.3%, 98.1% and 92.4% (95.5% average) for the three target symbols respectively.

5. Further Improvements

In this Section, we will discuss several features of token in example to improve the performance. At first, the weighting of token in different location with respect to the target symbol will be analyzed. We hope to find the effectiveness of weighting value for each individual token. Another technique is subject to the specific patterns contained in example. Such patterns represent a special semantic meaning. In the next subsection, we will discuss the difference of top 2 score for each example. A threshold value will be used to decide when the alternative technique can be used to improve the performance.

5.1 Weights for Individual Token

It is our intuition that the nearer a token is to target symbol, the higher prediction capability to token is. So in this Section we will try to find the effect of the tokens in different locations. And possibly, we can assign different weights to tokens with respect to its location in sentence.

The function *weight(i)* denotes the weighting value for token unit with location i , which can be derived from experiments for three symbols. Therefore, the related Equations, Eqs. (13) and (14), will be revised as:

$$TScore_L(s_j) = \sum_{i=-1}^{-m} (Score_L(s_j, w_i) * weight(i)) \tag{24}$$

$$TScore_R(s_j) = \sum_{i=1}^{-n} (Score_R(s_j, w_i) * weight(i)) \tag{25}$$

5.2 Pattern Table

In this subsection, we will discuss the patterns in text, which belong to the specific sense category and can be assigned directly. For instance, example (E6) contains the pattern “42/7”, which is incorrectly predicted as category *others* (s_7) with maximum score 4.6 generated by MLDC.

In fact, the pattern “42/7” stands for a name of network company. The target symbol “/” in “24/7” will be a silence. Therefore the pattern should be pronounced directly in Mandarin “四十二 (shi si er), a silence and 七 (chi)”. All such specific patterns, which are ambiguous and represent the specific term, such as a company name, specific date “9/21” etc., will be collected into the pattern table. Such table should be searched in front of adopting the MLDC. If the specific patterns of examples are found, its associated sense category will be assigned immediately without the prediction of MLDC. Currently, there are 12 entries collected in our pattern table. The use of pattern table can resolve several special cases and improve the performance by the amounts 0.6% ~ 1.0% for the three target symbols.

(E6) 42／7 可 協 助 網 站 解 決 網 路 廣 告 存 貨 問 題。

method \ s_n	1 date	2 fraction	3 time	4 directory	5 computer term	6 version	7 others	prediction
our approach	1.6*	0.9	1.6*	0.6	0.1*	1.5	**4.6**	incorrect

5.3 Adopting the Alternative

In the previous section, we introduced the token schemes of word and character, which are based on the different token unit in sentence. Finally the best average precision rate of outside test are 97.83%, 98.46 and 92.37% for symbols "/", ":" and "-" respectively using the character token scheme with location. One consideration is that whether the performance can be improved further by merging different token schemes or not? Although the token scheme of characters can obtain highest precision rate currently, what is the condition to adopt the alternative schemes to improve the performance further?

The normalized difference is defined as: $(score_1\text{-}score_2)/NT$. $score_1$ and $score_2$ are the top 2 score computed by proposed approach for target symbols. *TN* denotes the token number of sentence and will be changed with different token schemes. *TN* will normalize the difference of top 2 scores.

Note that the Elementary approach here was described at the end of Section 4.5. The final empirical performances of inside and outside testing are 99.6% and 96.5% average, employing the improving techniques proposed in this Section.

6. Conclusions

We have developed an approach, which contains the multi-layer decision classifiers and can disambiguate the sense ambiguity of non-alphabet symbols in Mandarin effectively. In contract to the *n*-gram language models, the new approach just needs smaller size of corpus and still hold the linguistic knowledge for statistical parameters. The model with voting scheme (baseline) is superior to *n*-gram (n=1,2) model. Several techniques are proposed and evaluated in our elementary experiment. Some examples are displayed to illustrate for each technique. The precision rates are 99.4% and 95.5% for inside and outside testing.

Three techniques are proposed to improve the performance further: weights for token with individual location, pattern table and the alternative. The final precision rates of further improvements are 99.6% and 96.5% for inside and outside test respectively.

In addition to the target symbols " /", ":" and "-" analyzed in the paper, there are some other symbols, such as *, %, [] and so on, in which the oral ambiguity problems will be incurred and should be resolved. Our approaches can be extended into related symbols.

References

P. Brown, S. Della Pietra, V. Della Pietra and R. Mercer. *Word Sense Disambiguation Using Statistical Methods*. In Proceeding of the 29th Annual Meeting of the Association for Computational Linguistics, Berkeley, pp. 264- 270, 1991.

Atsushi Fujii, Kentaro Inui, *Selective Sampling for Example-base Word Sense Disambiguation*, Computational Linguistics, vol. 24, number 4, 1998,pp 573-597.

William Gale, Kenneth W., Church and David Yarowsky, *A Method for Disambiguating Word Sense in a Large corpus*, Computer and the Humanities, 1992, Vol. 26.

A.R.Golding, *A Bayesian hybrid method for Context-Sensitive Spelling Correction*, In Proceedings of the third workshop on Very Large Corpora, pp. 39-53, Boston, USA, 1995.

Chu Ren Huang, Introduction to the Academic Sinica Balance Corpus, Proceeding of ROCLLING VII, pp. 81-99, 1995.

Feng-Long Hwang, Ming-Shing Yu, Min-Jer Wu and Shyh-Yang Hwang**,** *Semantic Classification for Patterns Containing Non-alphabet Symbols in Mandarin Text,* ROCLING XII, NCTU, 1999a, pp. 55-66.

Feng-Long Hwang, Ming-Shing Yu, Min-Jer Wu and Shyh-Yang Hwang**,** *Sense Disambiguation of Non-alphabet Symbols in Mandarin Text Using Multiple Layer Decision Classifiers*, Proceedings of 5th Natural Language Processing Pacific Rim Symposium (NLPRS), Beijing China, 1999b, pp. 334-339.

Nancy Ide and Jean Veronis, *Introduction to the Special issue on Word Sense Disambiguation: The State of the Art*, Computational Linguistics, vol. 24, number 1,1998,pp 1-40.

Daniel Jurafsky, James H. Martin, Speech and Language Processing, Printice Hall, 2000.

Ho Lee, Dae-Ho Baek, Hae-Chang Rim, *Word Sense Disambiguation Based on the Information Theory*, Proceedings of ROCLING X International Conference, Research on Computational Linguistics, Taiwan, pp. 49-58,1997

Claudia Leacock, Geoffery Towell, and Ellen M. Voorhees, *Corpus-based Statistical sense Resolution*, In proceedings of ARPA Workshop on Human Language Technology, San Francisco, CA, Morgan Kaufman, 1993.

Hinrich Schutze, *Ambiguity and Language Learning: Computational and Cognitive Models*, Ph. D Thesis, and Standard University, 1995.

K. Y. Su, T. H. Chiang, J. S. Chang, *A Overview of Corpus-Based Statistical-Oriented (CBSO) Techniques for Natural Language Processing*, Computational Linguistics and Chinese Language Processing, vol. 1, no. 1, pp.101-157, August 1996.

Jean Verious and Nancy Ide, *Word sense Disambiguation with very large neural extracted from Machine Readable Dictionaries*, in proceeding of COLING-90, 1990.

David Yarowsky, *Homograph Disambiguation in Text-to Speech Synthesis*, pp.157-172, 1997.

Chinese Knowledge Information Processing (CKIP) Group, *Technical Report: The Content of Academia Sinica Balanced Corpus(ASBC)* , Sinica Academia, R.O.C., 1995.

Building a Chinese Text Summarizer with Phrasal Chunks and Domain Knowledge

Weiquan Liu & Joe Zhou

Intel China Research Center

Page 87 ~ 96

Proceedings of Research on Computational Linguistics Conference XIII (ROCLING XIII)

Taipei, Taiwan

2000-08-24/2000-08-25

Building A Chinese Text Summarizer with Phrasal Chunks and Domain Knowledge

Weiquan Liu and Joe Zhou

{Lious.Liu; Joe .F.Zhou}@intel.com

Intel China Research Center

601 North Tower, Beijing Kerry Center

#1 Guanghua Road, Beijing 10002, China

Abstract

This paper introduces a Chinese summarizier called *ThemePicker*. Though the system incorporates both statistical and text analysis models, the statistical model plays a major role during the automated process. In addition to word segmentation and proper names identification, phrasal chunk extraction and content density calculation are based on a semantic network pre-constructed for a chosen domain. To improve the readability of the extracted sentences as auto-generated summary, a shallow parsing algorithm is used to eliminate the semantic redundancy.

1 Introduction

Due to the overwhelming amount of textual resources over Internet people find it increasingly difficult to grasp targeted information without any adjunctive tools. One of these tools is automatic summarization and abstraction. When coupled with general search and retrieval systems, text summarization can contribute to alleviating the effort in accessing these abundant information resources. It is capable of condensing the amount of original text, enabling the user to quickly capture the main theme of the text.

Based on the techniques employed (Hovy, 1998), existing summarization systems can be divided into three categories, i.e., word-frequency-based, cohesion-based, or information-extraction-based. Comparing to the other two techniques the first one is statistical oriented, fast and domain independent (Brandow *et al*, 1995). The quality, however, is often questionable. Cohesion-based

techniques (or sometimes called as being linguistic oriented) can generate more fluent abstracts, but the sentence-by-sentence computation against the entire raw text is often quite expensive. Even the most advanced part of speech (POS) tagging or syntactic parsing algorithms are unable to handle all the language phenomena emerged from giga-bytes of naturally running text. Summarization based on information extraction relies on the predefined templates. It is domain dependent. The unpredictable textual content over Internet, however, may let the templates suffer from incompletion or intra-contradiction no matter how well they might be predefined.

In this paper we introduces a Chinese summarization system. Though it is a hybrid system incorporating some natural language techniques, considering the speed and efficiency of text processing we still adapted a statistical oriented algorithm and allowed it to play a major role during the automatic process. After pre-processing, the system first extracts phrasal chunks from the input. The phrasal chunks normally refer to meaningful terms and proper names existing in the text that are difficult to capture using simple methods. Then, we use a domain specific concept network to calculate the content density, i.e. measuring the significance score of each individual sentence. Finally, a Chinese dependency grammar applies as a shallow parser to process the extracted sentences into bracketed frames so as to achieve further binding and embellishment for the final output.

2 System Overview

The system, hereafter referred to as *ThemePicker*, works as a plug-in to web browsers. When surfing among some selected Chinese newspaper web sites, *ThemePicker* monitors the content of the browser’s window. When the number of domain words or terms exceeds a pre-defined threshold, it will kick off the summary generation process and display the output in a separate window. Currently, we chose economic news as our specific domain.

The system consists of four components (see *Fig. 1*). The first component is a pre-processor dealing with the layout of the news web pages and removing unnecessary HTML tags while keeping the

headline, title and paragraph hierarchy. The retained information will provide the location of the extracted sentences for later manipulation.

The second component performs two tasks in parallel, resolving Chinese word segmentation and identifying and extracting phrasal chunks. As it is known to all, Chinese is an ideographical character based language with no spaces or delimiting symbols between adjacent words. After breaking the input sentence into a chain of separate character strings we use a lexical knowledge base to look up each word and parse the sentence appropriately. Person names and other proper names are also recognized during the segmentation process. Phrasal chunks are lexical units larger than words but not idioms. They are content oriented special terms (Zhou, 1999). We examined hundreds of documents and frequently encountered these phrasal chunks in the text that bear important information about the document. Since the meaning of a phrasal chunk is by no means the simple aggregation of the meanings of all the words in it, the word segmentation can not handle it. *ThemePicker* uses a statistical algorithm for phrasal chunk identification, aiming at the larger lexical unit that consists of two or more words always occurring in the same sequence.

The third component in sequence computes the degrees of sentence content density. The computation assigns a significance score to each sentence. The concept net that contains of more than 2000 concept nodes on economic news domain is used to define the semantic similarities between different sentences and adjust the significance scores of sentences across the input text. Sentences with high scores are selected for the inclusion in the candidate summary.

The fourth component analyzes the candidate sentences using a Chinese dependency grammar. The purpose is to improve the readability of the output summary.

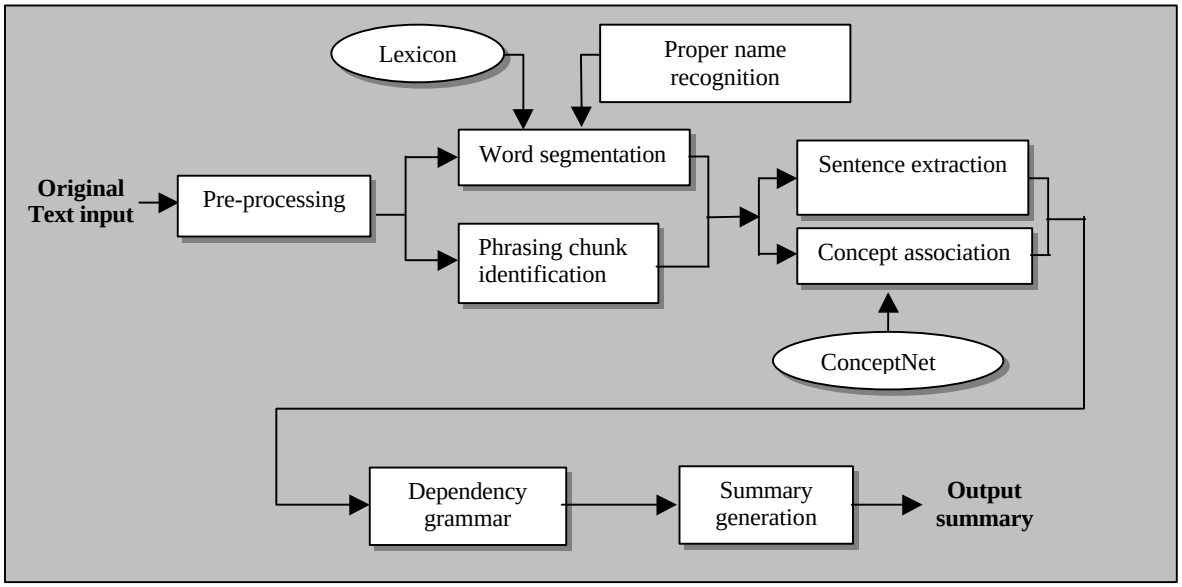


Figure 1: System overview and process flow

In the remaining sections of this paper we will describe in some details the major system components, i.e., word segmentation and proper name identification (Section 3), phrasal chunk extraction (Section 4), domain knowledge for summary generation (Section 5), and the dependency grammar (Section 6). The final section (Section 7) devotes to the system evaluation.

3 Word Segmentation and Proper Name Identification

The segmentation algorithm is a single scan Reverse Maximum Matching (RMM). One major difference from other RMMs is the special lexicon it uses. The lexicon consists of two parts, the indexing pointers and the main body of lexical entries (*see Fig 2*).

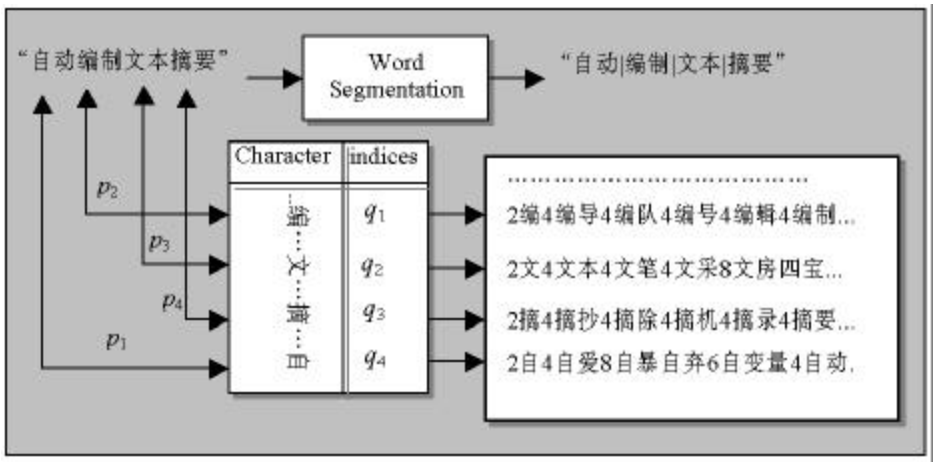


Figure 2: Lexicon structure and segmentation process

The algorithm works efficiently. The average number of comparisons needed to segment each word is only 2.89 (Liu *et al*, 1998). The unregistered single characters that are left behind the word segmentation will become the target of proper name recognition.

Proper names in Chinese carry no signals like capitalization, hyphenation, and interpunction in English, to indicate that they are special and different from other noun phrases. Our algorithm currently can handle two types of proper names, people names and organization names. People names include Chinese person names and names of foreign origin (though treated differently). The majority of organization names are company names due to the nature of the selected domain economic news.

To fulfil the task of recognizing Chinese person names we built a surname and a given name databases. Intuitively, any given Chinese person name is formed by a lead surname and followed by 1 or 2 given names. The surname has only one character and rarely has two, therefore the length of each person name ranges from 2 to 4 characters. In the surname and given name databases, each character is given a possibility value that is obtained by calculating its frequency over a large name bank. Our person name recognition algorithm works as follows.

When an unregistered single character word is encountered during the scan of the segmented text, the algorithm will check a) whether the character is a surname, and b) whether the character is followed by one or two single character words. If both conditions are met, these two to three consecutive character string may likely be a person name, denoted as $n=sc_1c_2$. (four-character names are temporarily omitted since they are rare). Here is the calculation of the possibility of n:

$$p(n) = \log p(s)\,p(c_1), \text{if there is a single given name, or}$$
$$p(n) = \log p(s)\,p(c_1)\,p(c_2), \text{if there are double given names.}$$

Thus, n is recognized as a Chinese person name for two character names if $\eta_1 < p(n) < \eta_2$, or for three character names if $\zeta_1 < p(n) < \zeta_2$. Here, η_1, η_2, ζ_1 and ζ_2 are pre-defined thresholds (Sun, 1998).

When calculating the possibilities, the title words, such as *Mr*., *Mrs*. etc. that immediately before *n* and verbs that follow *n* are also considered heuristically.

The difference between Chinese person name and transliterated foreign name is that the latter uses only a limited set of characters. The number of characters that allow to be used to denote foreign origin names is about 400 to 500 (Sun, 1998). Within this set, a portion of it can only be used as the first character and another subset can only be the tail ones. Using this principle we defined a set of rules to label the margins of foreign names resulting in satisfactory precision and recall.

Company name identification is also statistical and heuristic in nature. Based on the observation and analysis of a large quantity of collected Chinese text, we concluded that most company names can be denoted by the following BNF:

<*Geographical Loc*> + [<*Ordinal Number*>] + {<*Product Name*>|<*Trade Name*>} + <*Appellative Noun*>

Thus, we built a FSM in which heuristic rules are introduced to allow the system capture such text strings as company names.

Our initial evaluation of some sample text databases indicates that approximately 3% of the original text are proper names of various kinds, among whom the above two categories constitute more than 95%. This means that we would lose 2.85% of the segmentation accuracy if no action were taken to handle these two names. The above procedure now achieves more than 96% in accuracy. The improvement to the segmentation is 2.74%.

As mentioned above, proper names denote critical information in the original document. Their incorporation can make the summary more informative. Improved segmentation helps identify domain words more accurately. The identification of proper names also benefits the shallow parsing and improves the coherence and cohesion of summary output. Though phrasal chunk identification is independent to the segmentation, it is character based not word based.

4 Phrasal Chunk Identification

The phrasal chunk identification algorithm is to locate new terms formed by two or more words that frequently occur in the input text. For the words "香港", "金融" and "改革" found in the input text, if their frequencies all exceed a pre-defined threshold, we can say that they are key words in the original text. But, this does not mean the whole phrasal chunk "香港金融改革" is also a key word. To determine such a long term or a phrase chunk is also a key word we have to prove that these three words or 6 characters frequently appear in exactly the same sequence.

Our phrasal chunk identification algorithm uses a data structure used called Association Tree (A-Tree). A unique A-Tree can be constructed for each individual character using itself as the root of the respective tree.

Fig.3 shows an example of A-Trees. Each node consists of a character and an associated integer shows in parentheses. The integer refers to the number of occurrences of the character in the input text. The integers associated with other child nodes denote the number of occurrences that particular character follows its parent node. An A-Tree is constructed in the following way:

- Scan the input and record the position of each individual character $\boldsymbol{C}$. Define $\psi = \{\boldsymbol{C}_i \mid \boldsymbol{C}_i \in \sum\}$ as the set of all possible characters found in the input. $|\boldsymbol{C}_i|$ is the number of occurrence of $\boldsymbol{C}_i$. Delete all $\boldsymbol{C}_i$ when $|\boldsymbol{C}_i| < T$ with T as a predefined threshold

- For each remaining individual character $\boldsymbol{C}_i \in \psi$, create a A-tree and place $\boldsymbol{C}_i$ ***(n)*** at the root of the tree and ***n*** as the associated integer

- Add all the descendants of $\boldsymbol{C}_i$ to the leaf node set $\varphi = \{\boldsymbol{d}_j \mid \boldsymbol{d}_j \in \sum\}$**.** Delete those $\boldsymbol{d}_j$ where $|\boldsymbol{d}_i| < T$ with T as a predefined threshold

- For each node $\boldsymbol{d}_j$ in φ, add its descendant characters as described in step 3 and remove $\boldsymbol{d}_j$ after it gets expanded

- Repeat step 4 until no leaf can be expanded, then the A-Tree of $\boldsymbol{C}_i$ is complete.

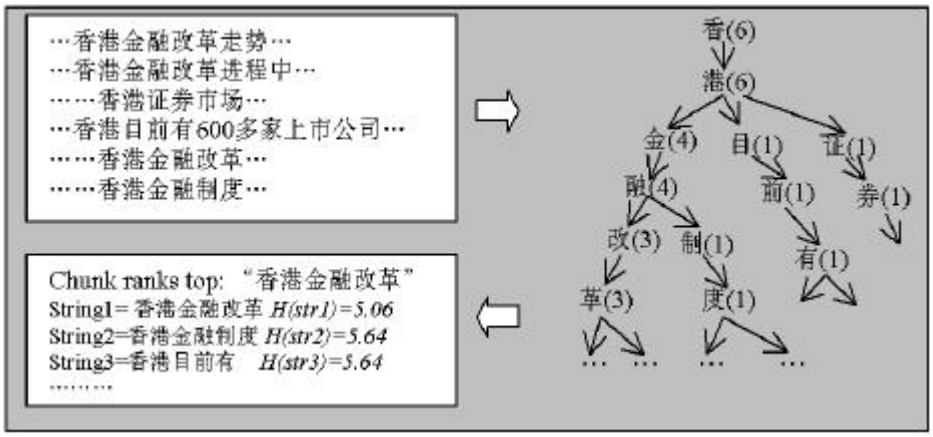


Figure 3: Phrasal chunk identification and an A-Tree

Once all A-Trees are constructed, new phrasal chunks can be extracted using entropy measurement. By tracking from the root node to each leaf node we can get a string of characters. For example, given a string $a_1a_2\cdots a_nb_1b_2\cdots b_m$ that denotes two sub-strings $A=a_1a_2\cdots a_n$ and $B=b_1b_2\cdots b_m$ with a_1 as the root, the entropy in B given A is: $H(B \mid A) = -\log p(B \mid A)$.

For an A-Tree the ratio $|b_m| / |a_n|$ is an estimation of $p(B|A)$. The smaller the H value the closer the relationship between these two sub-strings. A zero value means B always follows A, suggesting that AB is a meaningful phrasal chunk.

For a string $G=C_0C_1C_2\cdots C_n$, the entropy in C_1 given C_0 is H_{C1}= -log $P(C_1|C_0)$. Given C_0C_1, entropy in C_2 is H_{C2}= - log $p(C_2|C_0C_1)$. Thus, the total entropy measurement of G is defined as:

$$H_\Gamma = \sum_{i=0}^{n} H_{Ci} = -\log p(C_0 ... C_n), \quad \text{where } H_{C0} = -\log p(C_0)$$

As shown in Fig. 3 there are three phrasal chunks that have been listed with their respective H values with the first one bearing the lowest. The chunk identification algorithm will collect all the phrasal chunks with H value less than a certain threshold among all the A-Trees built from the input text. These phrasal chunks are larger than a word and likely express the key content of the input.

5 Sentence Extraction Using Domain Knowledge

The significance score of a sentence is determined based on the sum of two measurements, the density of domain concepts and the density of phrasal chunks.

Suppose a sentence denoted as $S=U_1U_2U_3\cdots U_L$, $U_i \in [F \mid W \mid K]$, $1<i<L$ (here F: function words, W: domain concept words and K: phrasal chunks), for those U_i that belong to F, no contribution will be made to the significance score. For other U_i that belong to W, their contribution to the significance score is gained from the domain knowledge contained in a *ConceptNet*. The *ConceptNet* is a graphic network constructed semi-automatically with nodes as various concepts and arcs as relations between concepts. The current version of our *ConceptNet* contains more than 2,000 nodes all collected from a large economic news database (see Fig. 4). The relations between concepts are of several types, such as a-kind-of, a-part-of, abbreviation-of, product-of, member-of, etc. The density of domain concepts α_w is calculated as follows:

$$a_w = \sum_{U_i \in W} g_i (1 - \sum_{U_j \in W} R(U_i, U_j)) / |U|, \; g_i \text{ is a heuristic coefficient.}$$

$R(w_1,w_2)$ is a function that determines the semantic relations between w_1 and w_2.

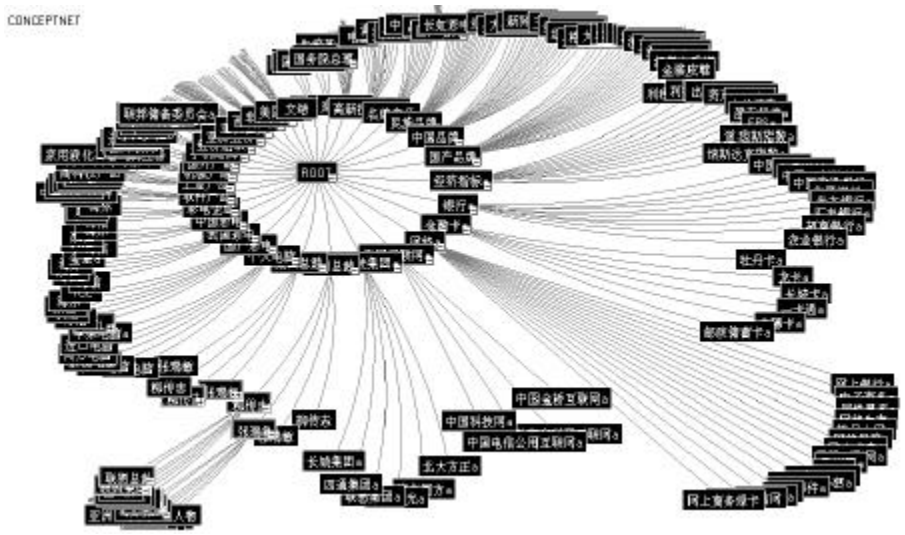


Figure 4: A partial snapshot of *ConceptNet* for economic news domain

For those U_i that belong to K, their contribution to the significance score is calculated as $a_K = \sum_{U_i \in K} g_i H(U_i) / |U|$ (referring to the previous section on the calculation of $H(U_i)$, the entropy of U_i.). Thus, the final significance score for the sentence S is:

$$a_S = l_S(b_1 a_W + b_2(1 - a_T)).$$

Conceptually, we give special treatment to domain concept words and phrasal chunks that appear in the title and headline. Some cue words or phrases are also detected that may bring positive or negative contributions to the significance score depending on their properties. b_1 and b_2 are balance factors for a_W and a_K. ls is determined by the location of S in the paragraph.

After all the input sentences receive the significance scores, those having values greater than a pre-defined threshold are chosen for the possible inclusion in the generated summary. The default length of the output summary is within 10~20% of the original text.

6 Dependency Grammar

Though they receive higher significance scores, the extracted sentences cannot be treated as the abstract of the original text. The readability is low even if they are strung together in the order as they occur in the input. The duplication in meaning and the appearance of improper conjunction words often make readers confused. Anaphora without contextual reference also poses difficulty in comprehension.

To bind and embellish the output summary we employed a Chinese dependency grammar to parse the extracted sentence into Dependency Relation Tree (DRT). Based on the methodology introduced in Liu *et al*, 1998, DRT can further be bracketed into cells. One of the cells is called the core with others being dominated by the core. There exist unique mappings between dependency relations in DRT and the dominating relations among cells. Fig. 5 illustrates such an example.

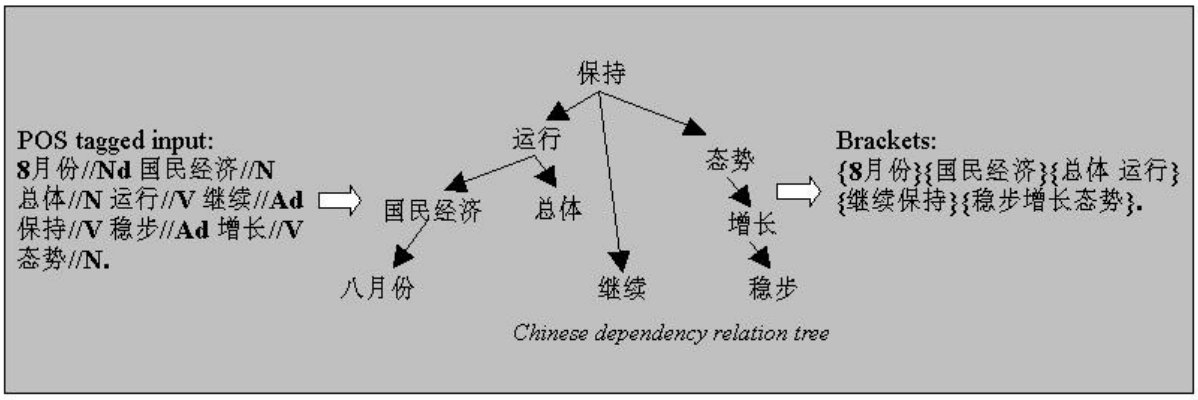


Figure 5: A sample sentence and its DRT

To eliminate the redundancy between two extracted sentences, we defined a semantic distance between them. Suppose that the bracketed cells of sentence S are represented as: $Core(S):=[slot_1(S), slot_2(S),\cdots slot_n(S)]$, then we can define the semantic distance between S_1 and S_2 as $D(S_1, S_2)$:

$$D(S_1, S_2) = \sum_i diff\,(slot_i(S_1), slot_i(S_2))$$

If $Core(S_1)$ and $Core(S_2)$ are different, $D(S_1, S_2)$ is indefinite. If $Core(S_1)$ and $Core(S_2)$ are the same, $diff(\cdot)$ is used to denote the semantic similarities between $slot_i(S_1)$ and $slot_i(S_2)$. The more similar the contents in the two slots, the smaller the value of $diff(\cdot)$, thus the smaller the distance $D(S_1, S_2)$.

A special case of the semantic distance is $D(S_1, S_2)=0$, that means S_1 and S_2 are basically identical in meaning, so one of them can be deleted. In most cases, $D(S_1, S_2)$ is greater than zero. A distance threshold is pre-defined in order to determine which extracted sentence can be eliminated. After the redundancy elimination the remaining portion of extracted sentences is reorganized to assemble the final output summary.

7 Performance Evaluation

In this paper we introduced a Chinese summarizier called *ThemePicker*. It is a hybrid system incorporating both statistical and text analysis models. For the sake of speed and efficiency, the

algorithm was implemented in a way that allows the statistical model to take the major role during the automated process. We built a semantic network (ConceptNet), a knowledge base that contains more than 2000 concept nodes with arcs indicating the conceptual relationships between or across nodes. Our experiments have showed that the content density measured based on ConceptNet can be more valid than an algorithm purely based on key terms. To achieve higher degrees of readability of the auto-generated summary, we adapted a shallow parsing algorithm to eliminate the semantic redundancy between the extracted sentences. While enhancing the summary cohesion and coherence, the computational overhead is restricted.

As pointed out in the literature, due to the lack of the evaluation standards for auto summaries, it remains to be an open research topic regarding how to compare the performance of a text summarizer with any concrete and solid measurement (Paice, 1990). We conducted a preliminary system evaluation against the database that contains 2800 news articles (2.4M words in total) on the economic domain. First, two human analysts manually screened 1200 articles and identifies 80 specific topics like *Euro, Fortune Forum*, *RMB won't be depreciated,* etc. Then, they manually generated summaries for several selected documents from each of the 40 topics. After that, they compared the automatically generated summaries with those they manually composed. The benchmark uses three grading scales, comparing to the manually generated summary the auto counterpart was assigned as either, *good* or *acceptable* or *non-acceptable*. The results indicated that the total documents that received either good or acceptable grades constitute more than two-thirds of the total documents evaluated. Evaluation using more rigid methodology will be performed in the future.

8 References

(Brandow *et al,* 1995) Brandow R. Mitze K. and Rau L F. *Automatic Condensation of Electronic Publication by Sentence Selection.* Information Processing & Management, 31(5): 675-68, 1995

(Cohen, 1995) Cohen J D. *Highlights: Language and Domain Independent Automatic Indexing Terms for Abstracting.* Journal of the American Society for Information Science, 46(3): 162-174, 1995

(Hovy, 1998) Hovy E. and Marcu D. *Automatic Text Summarization*. Tutorial of CONLING/ACL'98. 1998

(Liu *et al*, 1998) Liu W. Wang M. and Zhong Y. *Implementation of a Field Non-specific Hybrid Automatic Abstracting System*, in the Proceedings of 2nd Intl Conf on Information Infrastructure (ICOII'98), Beijing pp275-278

(Paice, 1990) Paice C D. *Constructing Literature Abstracts by Computer: Techniques and Prospects.* Information Processing & Management, 26(1):171-186, 1990

(Sun, 1998) Sun, M S *et al*. *Identifying Chinese names in Unrestricted texts*. Journal of Chinese Information Processing 9(2):16-27, 1998

(Zhou, 1999) Zhou F J. *Phrasal Terms in Real-world IR Applications*. In Strzalkowski T. *eds* Natural Language Information Retrieval, pp215-260. Kluwer Academic Publishers, 1999

反向異文字音譯相似度評量方法與跨語言資訊檢索

林偉豪，陳信希

國立台灣大學資訊工程學系

Page 97 ~ 113

Proceedings of Research on Computational Linguistics

Conference XIII (ROCLING XIII)

Taipei, Taiwan

2000-08-24/2000-08-25

反向異文字音譯相似度評量方法與跨語言資訊檢索

林偉豪　　陳信希

國立台灣大學資訊工程學系

Similarity Measure in Backward Transliteration between Different Character Sets and Its Application to CLIR

Wei-Hao Lin and Hsin-Hsi Chen

Department of Computer Science and Information Engineering

National Taiwan University

Taipei, TAIWAN, R.O.C.

E-mail: b4506060@csie.ntu.edu.tw; hh_chen@csie.ntu.edu.tw

Abstract

This paper classifies the problem of machine transliteration into four types, i.e., forward/backward transliteration between same/different character sets, based on transliteration direction and character sets. A phoneme-based similarity measure is proposed to deal with backward transliteration between different character sets. Chinese-English information retrieval is taken as an example. The experiments show that phoneme-based approach is better than grapheme-based approach. In a mate matching of 1,261 candidates, the average rank is 7.80 and 57.65% of candidates are ranked as number one.

摘要

本文首先根據音譯的方向是否跨不同文字系統，將機器音譯分成「正向同文字」、「正向異文字」、「反向同文字」與「反向異文字」等四種來討論。接著以相似度的比較作為音譯系統的基礎，將語音相似度分為物理聲音、音素、和形素三

個層級，並討論計算語音相似度的方式。最後，提出一個以音素相似度為基礎的方法，以中文和英文的音譯為例，進行反向異文字的音譯。實驗結果顯示在音素上的比較，比在形素上的比較來得有效。在一個 1,261 個人名的候選名單中，執行配偶配對實驗，平均排名是 7.80，其中 57.65%的排名為第一名。

1. 介紹

網際網路隨著電子商務的快速發展，更深入我們的日常生活中，也擴大了世界各國在網際網路上參與的熱度，不同語言所呈現的「內容」(content)在網際網路上傳播。舉凡許多廠商覬覦的中國大陸市場所使用的中文，或是整個歐盟成員國間的各種語言，已經讓網際網路從大部分以英文為主的內容，擴大成為多語言的內容。對網際網路的使用者、或是電腦應用系統(例如搜尋引擎、網路資源蒐集軟體、或是新聞自動摘要系統(Chen and Lin, 2000))來說，因為語言不同所形成的閱讀與處理障礙，也日漸增加。在這種多語的大環境下，機器翻譯(machine translation)與跨語言資訊檢索(cross language information retrieval)等相關自然語言處理系統研究，就極為受到重視。

所謂跨語言資訊檢索(Chen, 1997)就是以一種語言所表達的查詢(query)，去檢索另一種語言所呈現的內容。因為語言上的差異，通常需要將查詢轉換成跟內容一樣的語言。歧義分析(disambiguation)，是查詢翻譯(query translation)一項重要的研究(Bian and Chen, 2000)。根據 1995 年網路使用者，對 Wall Street Journal、Los Angeles Times 和 Washington Post 等新聞語料檢索的統計(Thompson and Dozier, 1997)，分別有 67.8%、83.4%、和 38.8%的檢索詞含專有名詞。我們知道辭典的覆蓋度，一直是查詢翻譯的重要問題，在專有名詞的翻譯更是挑戰。Chen 等人(1998)，Knight 和 Graehl(1998)，Wan 和 Verspoor(1998)都相繼提出機器音譯(machine transliteration)的方法，來處理這個問題。

音譯可以根據處理的方向，區分成正向音譯(forward transliteration)與反向音

譯(backward transliteration)。當一個語言的專有名詞，因為沒有適當或是不容易以意譯來表示時，會採用正向音譯，將其音呈現出來。例如義大利的觀光勝地Firenze，中文就音譯成「翡冷翠」，此為正向音譯。反過來說，當我們看到一個中文的音譯人名「阿諾史瓦辛格」，如果想要找出原文是 Arnold Schwarzenegger，就是反向音譯。一般來說，使用羅馬字母的拼音文字語言，會保持原詞語字母的拼法，然後以原語言的發音規則，或是自己語言的發音規則來發音。但如果在象形文字與拼音文字語言之間作音譯時，則需要將聲音由原語言盡量用另外一種語言相近的音素來表示，而且要符合目的語言(target language)的語音組合規則。很顯然地，拼音文字與象形文字之間的音譯處理相對來說較為困難，反向音譯比正向音譯更難。正向音譯允許某種程度的失真，所能夠接受的錯誤範圍較大；但反向音譯則不是。反向音譯較不允許錯誤，也就是在找出原文的過程中，必須要相當準確，否則反向音譯的結果應用性就較低。

本文第二節由音譯的正向和反向，以及是否跨文字來分析音譯問題，並介紹過去相關的研究。第三節由相似度的觀念，來執行機器反向音譯的程序。第四節提出一種以音素進行相似度比較的方法。第五節介紹實驗規畫，並對實驗結果進行討論，最後是結論。

2. 音譯分類與相關研究

根據音譯的方向，我們將音譯問題區分為「正向音譯」與「反向音譯」兩種。另外，根據音譯的原始語言與目標語言所採用的字母系統，還可以將音譯區分為「同文字系統間音譯」與「異文字系統間音譯」兩種。以下各小節，就針對這四種組合來介紹相關問題。

2-1 正向同文字間音譯

相同文字系統之間由於共用同一種文字，尤其以羅馬字母為基礎的拼音文字，不同語言在形素(grapheme)與音素(phoneme)間的組合規則雖不一樣，但是一個語言的詞語，要表達成另外一個同文字系統的語言，通常沒有問題。這類型的

音譯通常保持原始語言的文字拼法，而目的語言的使用者則以目的語言的發音規則，或是以原始語言的發音規則來發音。例如 Beethoven 雖然是德國名字，但是在英文的文本中，還是直接使用相同的文字拼法。即使在使用相同拼音字母的語言中，還是可能存在音譯。例如義大利觀光勝地 Firenze(義大利文)，英文則音譯為 Florence。

不同語言使用者在發音時，會採用自己語言的發音規則。例如英語使用者可能會依英語的發音規則來發音，這樣就跟原來德文的發音不同。但大體來說在音素上的發音較為接近，而且越來越多的人會選擇以原始語言來發音，以尊重原始語言。另外，日文中的漢字雖然與中文相通，但由於在發音上差距甚大，所以通常日文漢字翻譯成中文時，表面上與羅馬拼音文字一樣，保持原來日文漢字的寫法，但中文使用者通常會以中文的念法來對日文漢字發音。除非這位使用者學習過日文，才有辦法以正確的日文漢字來發音。

2-2 正向異文字間音譯

在正向異文字間音譯時，主要的工作在於將原始語言的音素，以目的語言的音素來呈現，並配合目的語言的組合規則表示。如果應用在書寫系統上，還要進一步將之前音譯後的結果，選擇目的語言適當的書寫文字，來呈現最後音譯的結果。Wan 與 Verspoor(1998)發展出一套自動將英文專有名詞，正向音譯成中文的系統。在將英文形素轉成音素的過程中，這個系統先將英文字母音節化(syllabification)。拆音節的方法主要有以規則為本(rule-based)，以及範例學習(instance learning)兩種。此系統採用規則為本的方式，但並不是利用上千條的規則來拆解音節，而是利用子音群(consonant cluster)與母音來當成音節的分界來拆解。由於中文為單音節的文字，且多為「子音＋母音」的結構，所以系統還要進一步將之前拆解的音節，做進一步的次音節化(sub-syllabification)。將沒有辦法以中文字發音的英文子音群拆開，並加上跟情境相關的母音，以兜成「子音＋母音」的音節。在將音素轉成目標語言(在這裡是中文)的文字過程時，Wan 與

Verspoor 的系統，先將拆解完成的音節查表轉換成漢語拼音，接著再查表將漢語拼音最後的中文音譯結果輸出。

2-3 反向同文字間音譯

如前所述，同文字系統間音譯，通常都是保持原來的詞彙組合與型態，所以並不需要做反向的音譯，來找出原始語言的詞彙到底為何。因此，這方面的處理比較簡單。

2-4 反向異文字間音譯

中文和英文間的轉換，是屬於反向且跨文字系統的音譯，這是本文所要討論的重點。在反向音譯(以後如果沒有特別說明，指的都是異文字間的反向音譯)的研究，有兩種不同的處理方式：一種是直接將音譯後目標語言的詞彙，利用某個模型反推出原始語言的詞彙；另一種是將音譯後目標語言的音譯字，與一串原始語言的候選字相比對，判斷何者可能是原來原始語言所使用的詞彙。

Knight 與 Graehl(1998)利用衍生模型(generative model)，設計一個反向音譯的系統，將音譯後的日文字反向音譯出原來的英文詞彙。當嘗試將英文(原始語言)專有名詞，音譯成日文(目的語言)片假名(katakana)時，衍生模型分成幾個階段處理，包括寫下要音譯的英文詞彙，用英文將該詞彙發音，將英文發音修改成日文可以發的音，將這個日文發音轉成片假名，並寫出片假名。假設我們有一個根據 P(w)機率分佈來產生英文字（word）的產生器，又假設我們有一個英文發音器。給定一個英文字時，發音器會依據 P(p|w)的機率來設定該字的發音（pronunciation）。對一個英文發音 p，如果我們想要找出這個發音可能的英文字時，我們就可以尋找看看哪一個英文字 w 可以讓 P(w|p)這個機率有最大值。根據貝式定理（Bayes' Theorem），這相當於尋找 P(w)・P(p|w)。這個系統用到如下五個機率分佈，其中 w 為英文字、e 為英文發音、j 為日文發音、k 為片假名、o 為光學辨識出來的字元：

(1) P(w)：產生英文詞彙。

(2) P(e|w)：英文詞彙發音。

(3) P(j|e)：將英文發音轉成日文發音。

(4) P(k|j)：將日文發音轉成片假名。

(5) P(o|k)：加入因為光學字元辨識所產生的錯誤。

當 OCR 取得一個片假名字串 o 時，反向音譯使用下面的公式，找出英文字串 w。

$$\arg\max_{w} P(w)\times P(e\mid w)\times P(j\mid e)\times P(k\mid j)\times P(o\mid k)$$

Chen 等人(1998)提出一個將英文音譯成中文(目的語言)的音譯字，反向音譯回英文(原始語言)的模組，並應用於中英跨語言資訊檢索系統。這個系統是將可能的音譯字辨識出來，再進行反向音譯。首先利用漢字羅馬拼音系統(例如 Wade Giles (威翟)，或是漢語拼音(Pinyin))，把可能的音譯字(中文)轉成羅馬字母。接著將這個詞彙與一串可能的專有名詞進行比對，藉此找出可能的原文(英文)。

3. 語音相似度

本篇論文把音譯問題視為相似度的衡量。正向音譯即是在不同語言之間，讓音譯後的結果能夠保持最大的相似度。在反向音譯，如果預先給予一份候選名單，則系統比較音譯字與候選名單上的詞彙，計算兩兩相似度。相似度的比對，可以分成三個層次：形素、音素、和物理聲音。

音譯後的詞彙與原詞彙之間，最直接的比較方式，就是請母語使用者發音，然後以物理上可以測量到的音波來比較。如果從人類可以發出的語音來看，音素集合是固定且有限的，我們可以嘗試在音素的層次來比較。兩個音素的發音位置，或是發音方式越相近，兩個聲音也會越相似。當我們以書寫文字來比較時，就是直接比較形素的相似度。如果書寫文字系統不同，例如中文的方塊字，與英文的羅馬拼音文字，就必須先轉換到相同的字母集合，才能進行比對。

在形素上的比較，Odell 與 Russell 的 Soundex 系統(Knuth, 1973)，是屬於同語言的羅馬拼音字母，利用子音來捕捉詞彙發音的特性。當兩個詞彙的子音位置

與發音相似時，表示這兩個詞彙的發音就可能越相似。而 Chen 等人(1998)的研究，可以視為在形素上比較相似度的反向音譯系統。由於所討論的中文音譯字，與原始語言英文的書寫系統不同，他們先將音譯字轉換成羅馬字母，這個動作稱為「羅馬拼音化」(romanization)。他們所採用的標準拼音系統，有威翟與漢語拼音，並加上一些經驗法則修正，來提高系統效能。

由於羅馬拼音系統，主要並不是考慮語音上的相近來設計，例如漢語拼音就用到了 Zh、Q 與 X 等羅馬字母，來表示與字母發音完全無關的漢語語音，所以英文音譯成中文的音譯字，在利用羅馬拼音系統轉換成羅馬拼音字母後，這些羅馬拼音字母，跟原來詞彙的拼音字母，在發音上並不十分相近。

有鑑於在形素層次上做羅馬拼音化時，非常需要一個以形素相近為出發點而設計的的羅馬拼音系統。例如在中文和英文這兩種書寫系統完全不同的語言，我們可以設計一個「自動建立羅馬拼音對照表」的系統。這個系統分為兩個階段：第一個階段是訓練，我們從已知的英-中音譯字與原文詞彙的配對中，學習英中音譯字所應該轉換的羅馬拼音字母。例如 Elton 與「愛爾頓」這個配對，先將中文代換成注音符號後，然後分別對兩個字做音節拆解的動作，得到「El・ton」與「ㄞ・ㄦ・ㄉㄨㄣ」，這裡忽略英文重音與中文聲調符號，而・為音節間隔符號。接著進一步將英文音節做次音節化後，我們就可以得到英文音節與中文字的音節對應共三組，包括「ㄞ→e」、「ㄦ→l」與「ㄉㄨㄣ→don」。第二個階段實際從事形素相似度衡量，系統根據前一個階段訓練所得到的對照表，將英-中音譯字轉換成英文詞之後，再與候選名單相比較。如前例，「愛爾頓」先轉換成注音符號「ㄞ・ㄦ・ㄉㄨㄣ」，然後查表後得到「e・l・don」。拿掉音節符號後就得到「eldon」，然後再做配偶配對(mate matching)。

表一列出上述例子的訓練結果。跟其他羅馬拼音系統來比較，我們可以發現：由這個系統所產生的對應，在形素上比其他拼音系統來得更接近實際情形。像是英-中音譯字中的ㄦ，例如貝爾（Bell）中的「爾」字，如果採用其他拼音系

統來做形素上的比較時，可以發現其他系統完全配對失敗（er≠l），只有經過訓練階段所產生對應才能正確配對（l=l）。換句話說，這個系統的對照表是比較有效的，所以能夠在形素層次上的相似度比較，有更好的效能。

表一・訓練結果與羅馬拼音系統

注音符號	威翟	耶魯	漢語拼音	注音符號第二式	「自動產生羅馬拼音對照表」系統的結果
ㄞ	ai	ai	Ai	Ai	e
ㄦ	erh	er	Er	Er	l
ㄉㄨㄟ	tun	dwei	Duan	duan	don

4. 音素相似度評量

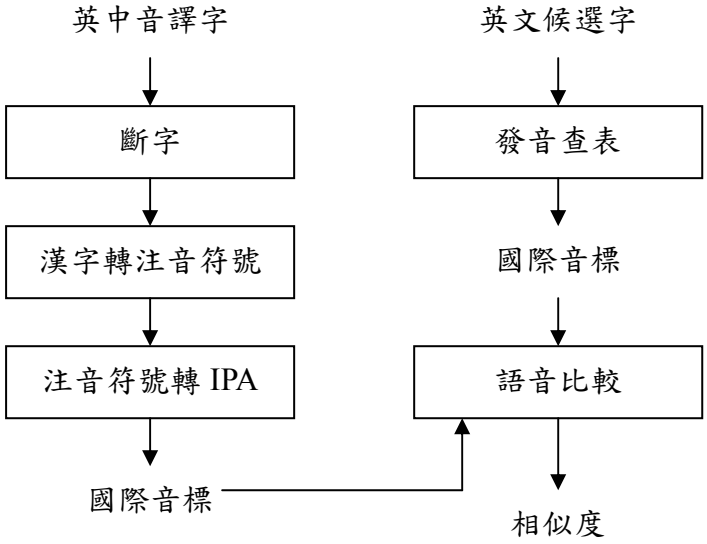
考量物理發音在跨語言資訊檢索的實用性，以及形素層次上比對的事前訓練，因此音素層次上的相似度比較易顯重要。而衡量兩個詞彙的音素相似度，我們提出一個以國際音標(International Phonetic Alphabet，IPA)為基準的比較，先將兩個詞彙的國際音標列出來，然後比較國際音標的相似度，進而達到反向音譯的目的。圖一顯示音素相似度比較的流程。我們先說明流程圖左邊的部分，也就是英中音譯字處理的部分：

(1) 斷字：在收到英中音譯字時，第一個步驟即是取出其中的中文字，也就是斷字。例如「亞瑟」這個音譯字，經過斷字後取出「亞」與「瑟」。

(2) 漢字轉注音符號：將前一個步驟斷出來的漢字，經查表後得到相對應漢字的注音符號。例如「亞」查表後，得到「ㄧㄚ」，在此我們忽略聲調符號。

(3) 注音符號轉 IPA：將前一個步驟中的注音符號，經查表二後，得到相對應注音符號的 IPA。表二列出母音和子音與 IPA 對照，這部份參考謝國平(1998)，並略作修正。例如「ㄧ」查表後，得到「ɪ」。一般 IPA 的表示必須配合特殊字體，才能顯現，CMU pronunciation dictionary 0.6 版(簡稱 CMU dict)

(ftp://ftp.cs.cmu.edu/project/fgdata/dict/)採用 ASCII 來表示，附錄列出 CMU dict 符號和 IPA 符號對照。例如，「ɪ」對應「IY」。



圖一・音素比對

表二・注音符號與 IPA 對照表

(a) 子音部份

注音符號	ㄅ	ㄆ	ㄇ	ㄈ	ㄉ	ㄊ	ㄋ	ㄌ	ㄍ	ㄎ	ㄏ	ㄐ	ㄑ	ㄒ	ㄓ	ㄔ
IPA	π	πH	μ	ϕ	τ	τH	ν	λ	κ	κH	ξ	τ˩	τ˩H	˩	τ♣	τ♣H
注音符號	ㄕ	ㄖ	ㄗ	ㄘ	ㄙ											
IPA	♣	|	τσ	τσH	σ											

(b) 母音部份

注音符號	ㄧ	ㄨ	ㄩ	ㄚ	ㄛ	ㄜ	ㄝ	ㄞ	ㄟ	ㄠ	ㄡ	ㄢ	ㄣ	ㄤ	ㄥ	ㄦ
IPA	ι	υ	ψ	α	ο	Φ	ε	αι	ει	αυ	ου	αν	↔ν	αN	↔N	™

對候選名單中的音譯字，處理的方式也是類似。每一個英文詞彙，我們查表(CMU dict 0.6)，以得到該詞彙的發音。例如「Arthur」的發音，經查表後得到「AA R TH ER」(忽略重音標示)。

「語音比較」是整個流程最重要的部分，當我們拿到兩串 IPA 時，如何比較這兩個 IPA 字串的相似度呢？首先來看以下三個定義，由字母對齊相似度、到字

串對齊相似度，最後到字串相似度。

定義 1：字母對齊相似度

假設 S_1 與 S_2 這兩個字串的字母集合為 Σ，而 Σ'表示 Σ 加上「_」(_表示空白字元)，給予 Σ'中的兩個字元 x 與 y，s(x, y)表示 x 與 y 對齊後，所得到的分數，稱為字母對齊相似度。

定義 2：字串對齊相似度

假設 A 為字串 S_1 與 S_2 的某一種對齊方式(alignment)，S_1'與 S_2'為插入空白後的字串。如果 S_1'與 S_2'的長度為 l，則對齊方式 A 的分數如下：

$$\sum_{i=1}^{l} s(S_1'(i), S_2'(i)) \text{。}$$

我們以一個例子說明上述定義。如前例，「亞瑟」經查表後，得到的發音是「IY AA S r」，而 Arthur 的發音為「AA R TH ER」，所以此時的 Σ={AA, ER, IY, R, r, S, TH}，而音素間彼此的分數，以下面的對稱矩陣表示：

S	AA	ER	IY	R	r	S	TH	_
AA	5	0	0	-10	0	-10	-10	-5
ER	0	5	0	-10	8	-10	-10	-5
IY	0	0	5	-10	0	-10	-10	-5
R	-10	-10	-10	10	-10	-10	-10	-5
r	0	8	0	-10	5	-10	-10	-5
S	-10	-10	-10	-10	-10	10	8	-5
TH	-10	-10	-10	-10	-10	8	10	-5
_	-5	-5	-5	-5	-5	-5	-5	-5

下面這個對齊方式：

亞瑟	IY	AA	_	S	r
Arthur	_	AA	R	TH	ER

依定義 2 所給定的字串對齊相似度分數為：-5 + 5 + -5 + 8 + 8 = 11。

然後我們來定義字串相似度。

定義 3：字串相似度

給定一個字母集合 Σ'，和成對的分數矩陣。字串 S_1 與 S_2 的相似度，定義

成 S_1 與 S_2 的最佳對齊方式 A 的值，也就是最大的字串對齊相似度值。相似度跟相關的最佳對齊方式，可以用 dynamic programming 的方式來找出。Gusfield (1997)曾定義基底條件(base condition)為

$$V(i,0) = \sum_{1 \le k \le i} s(S_1(k), \_)$$
$$V(0,j) = \sum_{1 \le k \le j} s(\_, S_2(k))$$

一般的 recurrence 式可以寫成：

$$V(i,j) = \max[V(i-1,j-1) + s(S_1(i), S_2(j)),$$
$$V(i-1,j) + s(S_1(i), \_),$$
$$V(i,j-1) + s(\_, S_2(j))]$$

$0 \leqq i \leqq \text{length}(S_1)$，$0 \leqq j \leqq \text{length}(S_2)$，$V(0, 0) = 0$。其中 V(i, j)為 $S_1[1..i]$與 $S_2[1..j]$這兩個前字串(prefix)，最佳對齊方式的值。假設 S_1 與 S_2 的長度各為 n 與 m，則最佳對齊方式的值就是 V(n, m)。如果利用 dynamic programming 的方式來求，這個值可以在 O(nm)的時間內算出來。

在我們的反向異文字音譯語音相似度評量中，Σ'為 63 個 IPA 音標符號(含空白)，其中英文有 39 個，中文除了共用的之外，另外還有 24 個中文所獨用的符號，所以整個分數矩陣的大小為 63x63。我們對分數矩陣中的分數指定方式如下：

(1) 原則上，IPA 匹配(match)給 10 分，不匹配(mismatch)扣 10 分。但若匹配的為母音，則只給 5 分，而母音不匹配不扣分。這裡我們希望利用母音來捕捉音節的對齊，所以母音不對齊不扣分。但由於母音在不同語言間的匹配，意義較不顯著，因此相同的母音只給 5 分。

(2) 與空白字元(_)對齊，可以看做 insertion 或是 deletion。由於不匹配可以看成是一個 insertion 加上一個 deletion。例如 abcdfgh 和 abcdigh，其中 f 與 i 未匹配，當要對齊時，可以採用如下的方式：

abcdf_gh

abcd_igh

所以未匹配要扣的分數，跟兩個字元對上空白，亦即做一次 insertion 和一次 deletion 要相同，這樣才沒有偏好。因此，為了公平起見，我們讓 insertion 或是 deletion 的扣分，等於不相同配對的一半，也就是 10/2=5 分。另外，關於空白對空白分數還是設-5(參考分數矩陣範例)，原因是兩個字串 ab 與 ac 在對齊時，如果 a 對 a 匹配給 10 分，不匹配扣 10 分，則 ab 和 ac 字串對齊相似度為：10 + (-10) = 0 分。如果加上空白，再進行對齊，如 a_b 和 a_c，這樣的分數為 10 + (-5) + (-10) = -5 分。也就是在對列時，同時加上空白是沒有用的，只是會把分數拉低，所以空白對空白是-5 分。

(3) 其他根據發音位置與發音方式的相近，中英文在音譯上的習慣、中英文各自的發音特性、將某些音標之間的配對分數設為 8 分，如表三所列。

表三・其他音標之配對

理由	例子
中文不分清濁	P 與 B、D 與 T、F 與 V、G 與 K、S 與 Z
發音方式與位置相近	B 與 Ph、K 與 Kh、D 與 Th、P 與 Ph
發音位置相近	L 與 R、DH 與 Th
發音方式相近	CH 與 Tch、CH 與 TSch、H 與 Th、G 與 Tc、JH 與 Tc，L 與 R、M 與 ANG、N 與 AN、N 與 AHN、N 與 ANG、NG 與 ANG、NG 與 AN、NG 與 AHNG、S 與 Sc、S 與 c、S 與 TH、S 與 TS、Z 與 Sc、Z 與 TS、Z 與 TSc
音譯習慣以及跨語言所造成的音標空缺	K 與 Tc、L 與 e、R 與 e、TH 與 Th、ZH 與 Tch、ER 與 r、ER 與 L、ER 與 e、UW 與 V、JH 與 TSc、G 與 Tch
中文不分長短母音	IH 與 IY、UW 與 W
半母音與母音	IY 與 Y

5. 實驗結果

我們採用配偶配對(mate matching)的方法，來評估語音的相似度。方法如下所述：給予已知的原始語言詞彙 o_i，與音譯後的目的語言詞彙 t_i 的配對清單集合，$\{(o_1, t_1), (o_2, t_2), \ldots, (o_n, t_n)\}$。當讀入音譯後的目的語言詞彙 t_k 時，測量語音相似度的系統，對整個清單中的每個原始語言詞彙作相似性比對，並計算每一對相似度的分數。之後再看看正確的原始語言詞彙 i_k，落在依分數高低排序的配對結果中的名次。名次越高，表示語音相似度比較越準確。

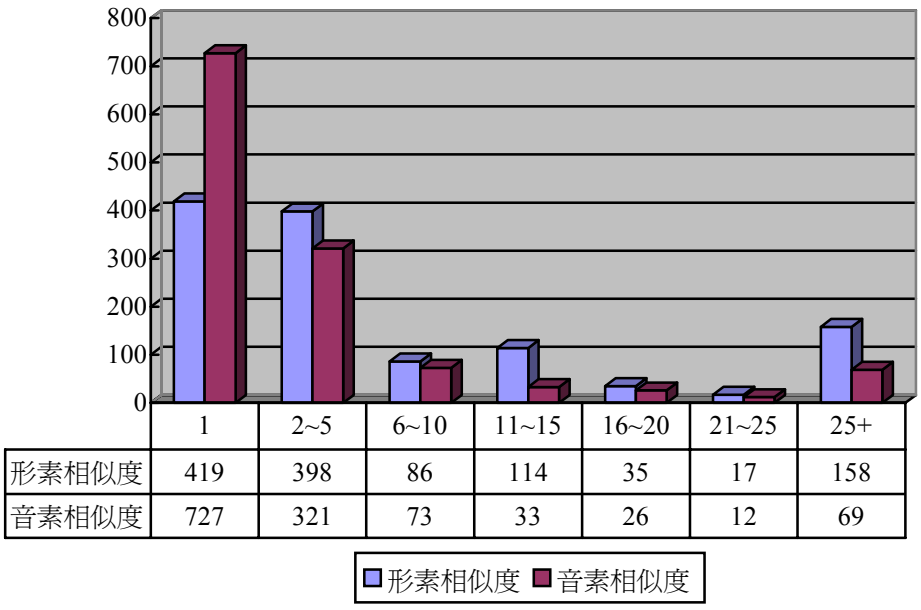
集合中的每一個目的語言詞彙，都設定名次後，我們就可以取這些名次平均值，作為一個語音相似度比較方法的評量標準。另外一個標準化的指標，是將這個平均的名次除以整個配對集合的個數 n。這表示當一個音譯後的詞彙輸入後，系統需要提出多少個候選詞彙，才會包括到正確答案。

根據這項評量語音相似度的方法，我們採用與 Chen 等人(1998)實驗相同的候選名單，共 1574 個人名。扣除無法找到發音的人名 313 個，合格的候選名單共 1,261 個人名。表三列出音素相似度和形素相似度的結果。本文所採用的音素相似度平均排名為 7.80，比 Chen 等人所採用的形素相似度平均排名 9.69，表現還要好。

表三・評估結果

	音素相似度	形素相似度
平均排名	7.80	9.69

圖二進一步列出排名分佈情況。



圖二・排名分佈

從以上結果我們可以清楚發現，在語音相似度的比較上，音素層次比形素層次表現的好。不僅平均排名上，音素相似度比形素相似度的平均排名好。如果進

一步觀察名次的分佈，音素相似度有 57.65%的結果都是最相似的，也就是正確答案。反觀形素相似度，只有 33.28%。

進一步觀察實驗結果中匹配失敗的配對，我們可以將失敗的原因歸類如下：

(1) 約定俗成但聲音並不相近的音譯：由於中文與英文是兩種不論書寫與發音都是相差甚遠的語言系統，因此不論對專業譯者或是一般作家，音譯並不是一件容易的事。但是一些已經約定俗成的翻法，例如 Bach（巴哈）、Caesar（凱薩）、John（約翰）等音譯，在音素上卻不十分相近，所以在音素層次上比對的效果不好並不令人意外。

(2) 英文非重音節的子音被忽略：英文中不在重音節的子音（通常是靠近結尾的部分），由於在中文使用者的語音知覺上並不明顯，所以音譯時經常就直接省略不翻，例如：Briand（白里安）中結尾的 d 在音譯成中文時就沒有被翻出來。

(3) 插入的母音造成混淆：由於中文字為單音節且多為子音加母音（CV）的結構，當英文字要轉換成中文時，勢必要在適當的地方插入母音才能構成 CV 結構。例如 Paul（保羅）與 Young（楊格）中結尾的ㄜ，原來是一個音節的字，到中文變成了兩個音節，也造成在音素層次上比對的困擾。

(4) 不是追求聲音接近的翻法：在一些特別的場合，特別是書寫的文本，翻譯者並不純粹追求聲音上相近的音譯方式，而可能為了與中文命名法相近（像 Gertrude，葛麗露）、或是為了簡潔（像 Gillian，姬兒），或是一味因襲傳統音譯方式卻忽略聲音上是否相近（像 Patricia，珮格麗特），這些在在都造成音素上的比對並不成功。

雖然如此，這些配對失敗的中文音譯字，比較音素相似度方法所找出來的最相似字，仍然反應音素上的相近。例如「保羅」雖然跟正確答案 Paul 並不十分相近，但系統比對得到的 Polo 在音素上其實是比 Paul 來得比「保羅」更接近；

又或「姬兒」雖然無法對到 Gillian，但是這個方法找到的 Jill 也是更接近「姬兒」，讓我們對這個方式在音素上比較相似度的能力深具信心。

6. 結論與未來的研究

機器音譯研究中，最具挑戰性，也最具實用價值的問題，就是在跨文字系統的反向翻譯。這種反向音譯在跨語言資訊檢索，或是機器翻譯時，都是一個不能忽略的問題。利用語音相似度的原理，從事反向音譯時，如果相似度的比較層次分為物理聲音、音素、與形素，而物理聲音無法進行時，我們發現音素層次上的比較，比之前在形素層次上的比較來得準確。

根據 Knight 與 Grahel(1998)對音譯系統的評量標準，這個以音素相似度來進行反向音譯作業的方式，相當接近人類在判斷音譯字是否相近，因為音素比較接近實際的聲音，而形素通常差距較大。而這個方法在應用到其他語言配對時，只要給定不同的配分矩陣就可以。最後，這個方法可以根據分數的高低，來提供一串可能的清單。所以，這個方法不管在理論與實際應用上都是深具價值。

機器音譯並不完全只是在語音上追求相等，有的專有名詞翻譯，因為歷史因素或是語言使用者的習慣，採取意譯而不是音譯。例如國家名稱 the United States，在大部分的中文文件中都是意譯成「美國」，而不採取音譯。同時，並不是所有專有名詞都採取意譯，例如 British Virgin Island 中的 Virgin，在中文音譯成「維京」，而不是採取意譯，Island 則直接翻譯成島。因此，在反向異文字音譯處理之前，先將地名送進雙語字典。如果已有現存的翻譯，就直接採用此翻譯。如果沒有，再檢查有沒有關鍵詞。關鍵詞查雙語辭典，其餘部份才經反向異文字音譯處理。

參考文獻

Bian, Guo-Wei and Chen, Hsin-Hsi (2000) "Cross Language Information Access to Multilingual Collections on the Internet," *Journal of American Society for*

Information Science, **51**(3), 2000, pp. 281-296.

Chen, Hsin-Hsi (1997) "Cross-Language Information Retrieval," *Proceedings of ROCLING Workshop on ED/MT/IR*, Academic Sinica, Taipei, June 2, 1997, pp. 4-1~4-27.

Chen, Hsin-Hsi; Huang, Sheng-Jie; Ding, Yung-Wei and Tsai, Shih-Chung Tsai (1998) "Proper Name Translation in Cross-Language Information Retrieval," *Proceedings of 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics*, Montreal, Quebec, Canada, August 10-14 1998, pp. 232-236.

Chen, Hsin-Hsi and Lin, Chuan-Jie (2000) "A Multilingual News Summarizer," *Proceedings of 18th International Conference on Computational Linguistics*, July 31-August 4 2000, University of Saarlandes.

Gusfield, Dan (1997) *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*, 1997, Cambridge University Press.

Knight, Kevin and Graehl, Jonathan (1998) "Machine Transliteration," *Computational Linguistics*, Vol. 24, No. 4, 1998, pp. 599-612.

Knuth, Donald E. (1973) *The Art of Computer Programming, Volume 3, Sorting and Searching*, Addison-Wesley, Reading, Mass, 1973, pp. 391-392.

Thompson, P. and Dozier, C. (1997) "Name Searching and Information Retrieval," *Proceedings of Second Conference on Empirical Methods in Natural Language Processing*, Providence, Rhode Island, 1997.

Wan, Stephen and Verspoor, Cornelia Maria (1998) "Automatic English-Chinese Name Transliteration for Development of Multilingual Resources," *Proceedings of 17th COLING and 36th ACL*, 1998, pp. 1352-1356.

謝國平(1998)，*語言學概論*，三民書局，台北，1998 年 10 月。

附錄・CMU dict 符號與 IPA 符號對照表

本表根據 CMU dict 0.6 版所訂定， *表示該符號原來不在 CMU dict 中，我們為了中英音譯而增加。

cmu dict 符號	IPA 符號	cmu dict 符號	IPA 符號	cmu dict 符號	IPA 符號
AA	α	M	μ	*Tc	τ⌡
AE	Θ	N	ν	*Tch	τ⌡H
AH	℘ or ↔	NG	N	*c	⌡
AO	□	OW	o	*TSc	τ♣
AW	αυ	OY	oι	*TSch	τ♣H
AY	αι	P	π	*Sc	♣
B	β	R	ρ	*Zc	⏐
CH	τΣ	S	σ	*TS	τσ
D	δ	SH	Σ	*TSh	τσH
DH	Δ	T	τ	*r	Φ
EH	E	TH	T	*AIY	αι
ER	™	UH	Y	*EYIY	ει
EY	ε	UW	υ	*AUW	α
F	ϕ	V	ϖ	*OWUW	o
G	γ	W	ω	*AN	αν
HH	η	Y	φ	*AHN	↔ν
IH	I	Z	ζ	*ANG	αN
IY	ι	ZH	Z	*AHNG	↔N
JH	δZ	*Ph	πH	*e	™
K	κ	*Th	τH	*y	ψ
L	λ	*Kh	κH		

Clustering Similar Query Sessions Toward Interactive Web Search

Chien-Kang Huang, Lee-Feng Chien, and Yen-Jen Oyang

Department of Computer Science, National Taiwan University

Page 115 ~ 134

Proceedings of Research on Computational Linguistics Conference XIII (ROCLING XIII)

Taipei, Taiwan

2000-08-24/2000-08-25

Clustering Similar Query Sessions Toward Interactive Web Search

Chien-Kang Huang, Lee-Feng Chien*, Yen-Jen Oyang
Department of Computer Science, National Taiwan University, Taiwan.
*Institute of Information Science, Academic Sinica, Taiwan.
ckhuang@mars.csie.ntu.edu.tw, *lfchien@iis.sinica.edu.tw, yjoyang@csie.ntu.edu.tw

Abstract

A new effective log-based approach for interactive Web search is presented in this paper. The most important feature of the proposed approach is that the suggested terms corresponding to the user's query are extracted from similar query sessions, rather than from the contents of the retrieved documents. The experiment results demonstrate that this approach has a great potential in developing more effective web search utilities and may inspire more studies on advanced log mining mechanisms.

1. Introduction

Users' queries for Web search are usually short. For example, the average length of TREC topic description for conventional text retrieval is 15 tokens [11,12], while analyses of web search engine logs reveal that the average query length for Web search is about 2.3 tokens [6,9]. Short queries means that the information about the user's intention provided to the search engine is very limited. To deal with the short query problem, interactive search techniques [2,7] which attempt to identify the user's intentions and suggest more precise query terms are therefore commonly incorporated in Web search engine design.

To determine more relevant query terms for each given query, the conventional

interactive search processes often rely on the key terms in the retrieved documents [2,7,10]. The key term set is extracted either statically from the documents during preprocessing or dynamically on-the-fly. Since the precision rates of the retrieved documents are usually not high enough, the extracted key terms are often found not relevant and not very helpful in practical Web search services.

In fact, extraction of relevant terms can be carried out by analyzing users' logs. In recent years, mining search engine logs has been obtaining more attention. Silverstein et al. [9] performed a second-order analysis on a log with a huge number of Web query terms. The results are then used to facilitate phrase recognition and query expansion [3].

In this paper, we propose a new approach based on log analysis for developing more effective interactive Web search engines. The most important feature of the proposed approach is that the suggested terms are extracted from similar query sessions, rather than from the contents of the retrieved documents. A query session is defined as a sequence of search requests issued by a user for a certain information need. The basis of the proposed approach is that two users with the same information need will issue common or related query terms. For example, in search for a subject regarding "search engine technology", a user may submit query terms such as "search engine", "Web search", "Google", "Web search and multimedia", while another user may submit "Web search", "Lycos". Therefore, if similar query sessions could be identified, query terms for the same information need can be extracted and applied to improve the effectiveness of search engines.

The remainder of the paper will be organized as follows. Section 2 is a brief

introduction to the idea of interactive search based on similar query sessions. The method proposed for segmenting query sessions from proxy logs will be described in Section 3. Then, how query sessions are clustered is addressed in Section 4. Section 5 will present some experiment results and a conclusion is given in Section 6.

2. Interactive Search Based on Similar Query Sessions

Fig.1 is an abstract diagram showing our idea for interactive search. Before introducing the basic idea of the proposed approach, the concept of query session is presented and defined below:

Definition of Query Session:

Query session = (ID, R_1,..., R_m) where ID means the identifier of a user submitting a sequence of requests to a search engine in a certain period of time. Each request R_i = (t_i, q_i) means user ID sends a query term q to the search engine at time t

The proposed approach is assumed that the query space of users is formed by clusters of users' query sessions, and a set of query sessions grouped in the same clusters contain similar information needs. For each input query session with a sequence of *i* query terms, the interactive search process is then designed to retrieve the most similar cluster of query sessions from the query space, and then extract relevant terms in the cluster as suggested terms for next search. Once the *i+1*th query term is selected, it forms a new query session with *i+1* terms and the interactive process will perform again.

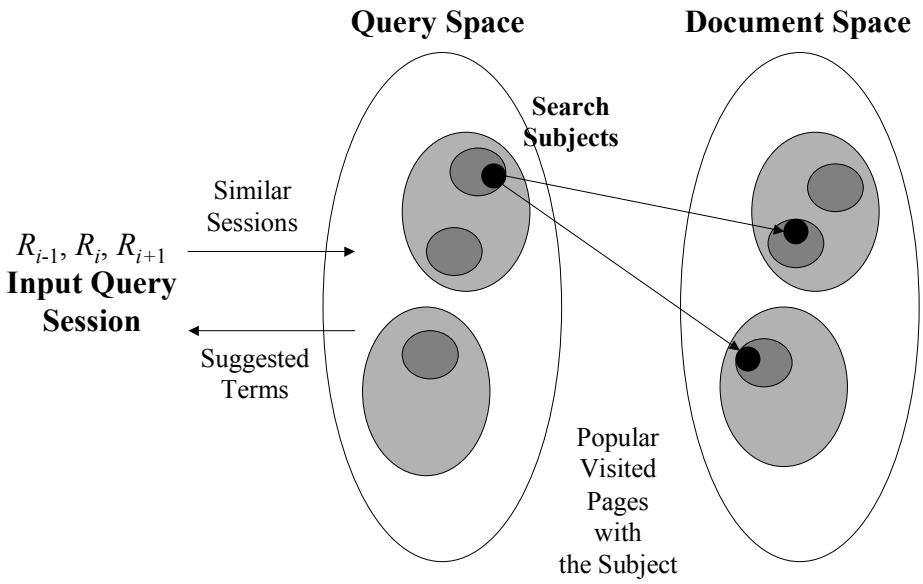


Fig.1 An abstract diagram showing our idea for interactive search.

Based on the above definition and idea, the problem to be dealt with is then formulated.

The Query Session Clustering Problem

For a set of query sessions from a query session log, the considering problem is to cluster these query sessions into different groups based on estimated similarity between query sessions. Each cluster can be defined as $\{S_i | f(S_i, S_j) > threshold\}$*, in which f() is the similarity estimation function between query sessions.*

Overview of the Proposed Approach

The proposed approach, as shown in Fig. 2, is composed of three processing modules: query session segmentation module, query session clustering module and relevant term extraction module. In the stage of query session segmentation, each query

session will be segmented and extracted from a proxy log, according to the time gap between successive search requests. All of the extracted query sessions will form as a query session log. In the session clustering stage, the sessions with similar queries will be clustered and the cluster names extracted from composed high frequency terms. In the relevant term extraction stage, the relevance between the recorded query terms will be calculated and sets of relevant terms will be extracted for term suggestion applications in a search engine.

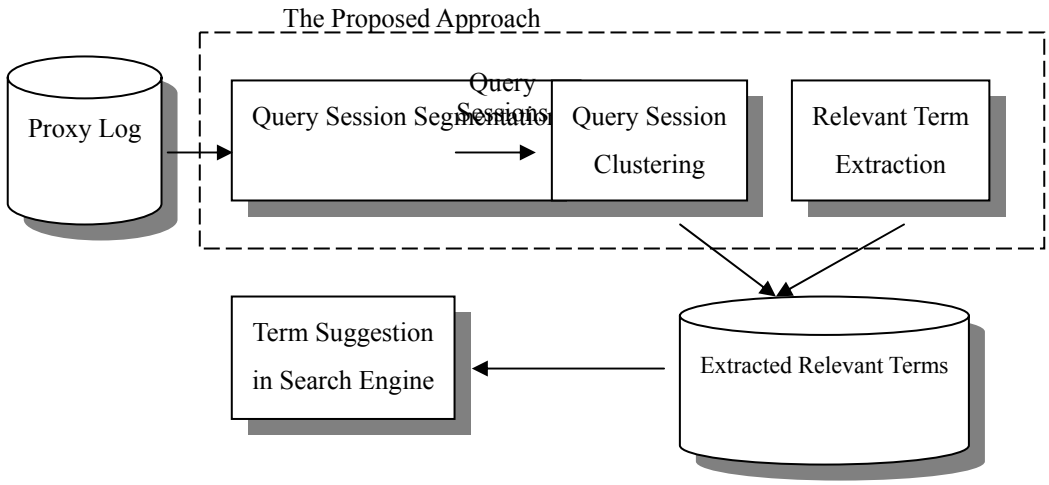


Fig.2 An overview of the proposed approach

3. Query Session Segmentation

A common proxy server might easily have thousands of clients accessing the web through it. Not only the general HTTP requests could pass through the proxy server, all of search HTTP requests are same. Compared with common search engine logs, a proxy server's log records more rigid information for users' information access and, more importantly, the recorded search requests are not limited to certain search engines.

However, a proxy log might record too much information and only some of them are useful in terms of search engine applications [13]. In our application it is sufficient to only use the following fields of logging information:

- A **timestamp** that indicates when a search request was submitted.
- A **client address** that indicates the IP address of the requesting instance.
- A **URL** string that contains the request content.

Since the experiments are just performing, the testing log is from NTU proxy servers and is still small. Some statistics of the testing proxy log are listed in Table 1.

Logging Days	15 days (2000/4/22 00:00~ 2000/5/7 00:00)
No. of Total Clients	12,005
No. of Total Queries	341,443
No. of Distinct Queries	51,125

Table 1. Some statistics of the testing proxy log.

It is noted that the recorded search queries in the log are limited to that for two representative search engine sites in Taiwan: www.kimo.com.tw and www.yam.com.tw.

In addition to identifying unique users, an effective query session segmentation algorithm has to determine which are the starting and ending requests for each user's information need. Most of search requests posse a property of time locality. Client ID with temporal information really provides a strong constraint in determining the query sessions. For this reason, we adopt an assumption similar to Silverstein et al.

that queries for a single information need come clustered in time, and then there is a gap before the user returns to the search engine.

The method for query session segmentation is then proposed as follows:

The Method for Query Session Segmentation:

For a proxy log, it will segment the whole log $L = \{T_i |$ *where* $T_i = (ID_k, t_i, q_i)\}$ *into a set of query sessions* $\{S_i | S_i = (ID_k, R_1, \ldots, R_m)$, *where* $R_i = (t_i, q_i)$, *and* $t_i - t_{i-1} <$ *threshold}, where* t_i *is the timestamp when the query* q_i *issued.*

Analysis of Segmented Query Sessions

To realize the performance of the above method, several experiments have been performed. Fig. 3 shows the relationship between the time thresholds and the numbers of segmented query sessions. The time thresholds determine the maximum time gap between two successive requests from the same client. The values of the time thresholds were tuned from 0 seconds to 360 seconds. In the research of Silverstein et al, 5 minutes as suggested is a proper threshold value. With the same threshold value, the number of segmented query sessions is shown in Table 2. The percentages of the segmented singleton and non-singleton query sessions are found similar to those reported by Silverstein et al.

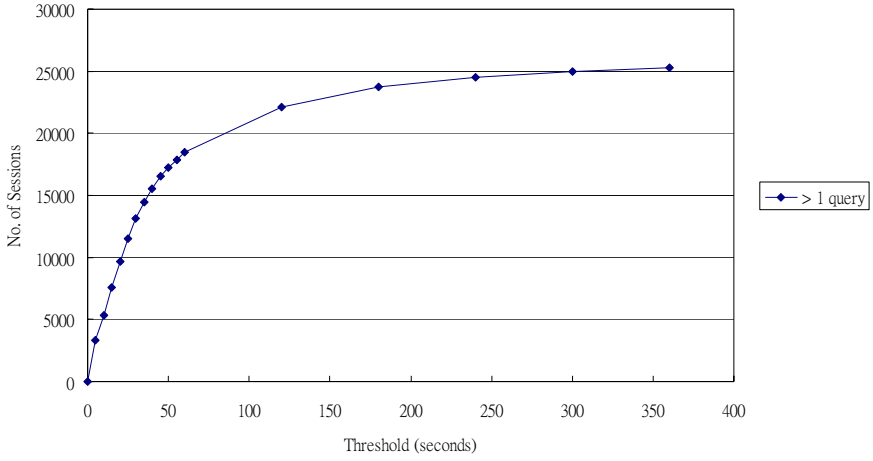


Fig 3. The numbers of segmented query sessions (that with more than 1 queries), regarding to the change of increasing time thresholds.

	No of sessions	Percentage
1 query per session	71,790	74.8%
> 2 queries per session	24,986	25.2%
Total	96,776	100%

Table 2: Percentages of the extracted singleton and non-singleton query sessions, when the time threshold is set as 5 minutes.

4. Query Session Clustering

As the definition of the session clustering problem in Section 2, the similarity estimation function is necessary and formulated below:

Similarity Estimation Between Query Sessions:

Given two sessions, $S_1 = (ID_K, R_{11},\ldots,R_{1m})$ and $S_1 = (ID_L, R_{21},\ldots,R_{2n})$, in which R_{ij} is the j-th query term occurred in session S_i which is issued by a client. The similarity estimation function is defined as:

$$sim(S_1, S_2) = \sum_{1<i<m, 1<j<n} sim(R_{1i}, R_{2j}) / mn$$

The similarity between two composed query terms will be further described below. Development of an effective relevance estimation function is important. Since our research is just in the beginning, only two kinds of relevance estimation functions were developed and tested. In the first method, the relevance between two query terms is simply calculated by the co-occurrence frequency value of the query terms in the segmented query sessions. In the second method, the relevance is calculated by the cosine value of the query terms' feature vectors.

Method I for Similarity Estimation of Relevant Terms

In the first method, we define the relevance estimation function below.

$$f(x, y_i) = co\text{-}occurrence(x, y_i)$$

Before calculating the relevance between query terms, a set of query sessions has been segmented and extracted from the testing proxy log. After preprocessing the query session log, we calculate the co-occurrence frequency between each unique query term and its associated terms occurring together in the same query sessions. We explain the calculation process with a simple example below. After segmenting the proxy log, it is assumed that we got five query sessions S1-5 and each contains several query terms from A to F, e.g.,

S1: {A, B}

S2: {C, D, B}

S3: {A, B, C}

S4: {A, E}

S5: {B, C, E, F}

In this case, f(B, C) will be 3, because B and C occur together in three sessions, i.e., S1, S2 and S3. Although the above method looks straightforward, its obtained performance is really out of our expectation.

Method II for Similarity Estimation of Relevant Terms

In the first method, the relevance of two query terms needs a strong support of their co-existence in a certain number of query sessions. Using a VSM-like technique it can release such a constraint. The second method is based on vector space model, and it can be formalized as below.

$f(x, y_i)$ = cos(FV(x), $FV(y_I)$), FV(x) means feature vector of term x,

$FV(T_i)$ = {S_j:N_{ij}| Sj and Ti are coexist in query sessions, Nij is the count of their co-occurrence}

Assuming there are two terms T_1 and T_2:

T_1 → {S_1:N_{11}, S_2:N_{12}, S_4:N_{14}, S_5:N_{15}}

T_2 → {S_1:N_{21}, S_2:N_{22}, S_3:N_{23}, S_7:N_{27}}

The relevance value of T1 and T2 is the obtained cosine or say the inner product value of these two vectors.

$$f(T_1, T_2) = \cos(FV(T_1) \bullet FV(T_2)) = \frac{\sum_i (N_{1i} \bullet N_{2i})}{\sqrt{\sum_j N_{1j}^2} \bullet \sqrt{\sum_k N_{1k}^2}}$$

The Clustering Process

A issues to be dealt with in the clustering process, that is, what each cluster means and how to name these clusters. In order to find out the representative meaning of each cluster and avoid the difficulty in classifying short sessions, the clustering process is being developed as shown in Fig. 4, which is designed as an incremental adaptive procedure.

This procedure consists of 4 processing steps:

1. For each incoming query session, check whether there are certain common query terms between the session and existing clusters. If the common query terms exist, assign the session to these clusters.

2. If the incoming session doesn't have sufficient common query term with existing clusters, calculate the similarity between the session and existing clusters. If the estimated similarity is higher than a predefined threshold, the session will be assigned to the cluster.

3. If the incoming session isn't assigned to any cluster, it will be sent to the delay queue for further processing. In this step, the incoming session will compare with other sessions in delay queue to check whether there are common query terms in the sessions that could be combined.

4. A standalone module will dynamically merge or split the clusters according to the new requirements or the new incoming sessions. When the similarity of two clusters are higher than another pre-defined threshold, merge will happen; when the cluster grows larger, split will happen. Merging and splitting are strategies for maintaining the similarity of query sessions in a cluster.

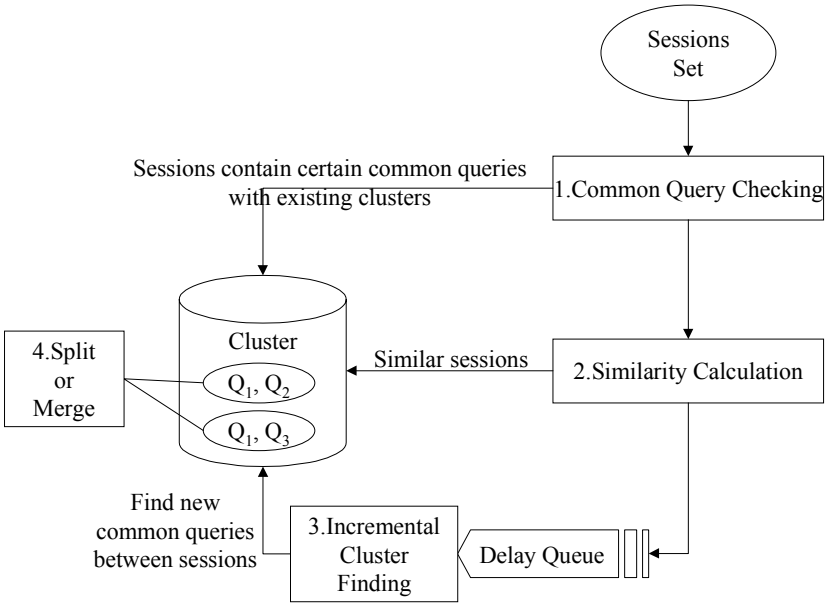


Fig 4. The work flow of the session clustering process.

5. Preliminary Experiments

Performance of Query Session Clustering

The above clustering process was just implemented. The sessions grouped by Step 1 are set that should contain at least two common query terms, and each obtained cluster is then named by the pair of common query terms with the highest frequency.

Currently, there are about 700 initial clusters have been obtained from the query session log shown in Section 2. Table 3 illustrates an example of the clusters. The numbers ahead in each row of Column 2 are frequency values of the corresponding sessions. Based on our initial observations, the relevance of the clustered query sessions is often high. It is obviously higher than that obtained with document-based approach in our experiences.

Cluster Name	Sessions in Cluster
台大__台灣大學	10: 台大 台灣大學 1: +台大 台大 台灣大學 1: 圖書館 台大圖書館 台大 台灣大學 台灣大學圖書館 1: 台大中國文學系 台大 台灣大學 1: 台大 台灣大學 兵學 1: 臺大物理 台大 台灣大學 1: 台灣大學 台大 台灣大學^招標 台灣大學^工程 招標^台大 1: 台大電機 台大 台灣大學 1: 台大 台灣大學 國立台灣大學 1: 社會學 台大圖書館 台大 台灣大學 市立圖書館 1: 台大 台大醫學院 台大醫學系 台灣大學 1: 時報育樂 時報育樂股份有限公司 時報 台大 台灣大學 大學聯招放榜 榜單 1: 台灣大學 台大 成功大學 1: 圖書 台大圖書 台大 台灣大學 1: 椰林 台大 台灣大學 台灣大學計算機中心 1: 台大 台灣學大 台灣大學 比賽

Table 3. An example of the obtained session clusters.

It is worthy to note that clusters with similar names (that with shared query terms as the names of the clusters) usually contain similar information needs. Table 4 is an example which contains a number of clusters with information needs related to 圖片 (picture). In these clusters, 圖片(picture) will relate to several different kinds of search subjects, including characters in cartoon (e.g. kitty and pokemon), downloading, online picture banks, greeting cards for some festivals, and etc. These similar clusters could be further taken as sub-clusters of the information needs. The obtained information would be very useful in performing term suggestion in interactive search process.

Cluster Name	Translation
卡通圖片__kitty	cartoon picture __ kitty
可愛圖片__卡通	lovable picture __ cartoon
母親節圖片__母親	mother's day picture __ mother
母親節圖片__母親節	mother's day picture __ mother's day
母親節圖片__母親節卡片	mother's day picture __ mother's day greeting card
母親節圖片__趴趴熊	mother's day picture __ bear
母親節圖片__康乃馨	mother's day picture __ carnation
圖片__kitty	picture __ kitty
圖片__卡通	picture __ cartoon
圖片__卡通圖片	picture __ cartoon picture
圖片__布丁狗	picture __ pudding dog
圖片__母親節	picture __ mother's day
圖片__母親節卡片	picture __ mother's day greeting card
圖片__母親節圖片	picture __ mother's day picture
圖片__皮卡丘	picture __ picachu
圖片__有趣	picture __ funny
圖片__狗	picture __ dog
圖片__趴趴熊	picture __ bear
圖片__桌面	picture __ theme
圖片__桌面王	picture __ themeking
圖片__神奇寶貝	picture __ pokemon
圖片__動物	picture __ animal
圖片__動畫	picture __ animation
圖片__康乃馨	picture __ carnation
圖片__遊戲	picture __ game
圖片__遊戲下載	picture __ game download
圖片__圖	picture __ graph
圖片__圖片下載	picture __ picture download
圖片__圖庫	picture __ picture bank
圖片__圖檔	picture __ picture file
圖片__漫畫	picture __ comic

Table 4. An example which contains a number of obtained clusters with information needs related to 圖片(picture)

Performance of Relevant Term Extraction

In fact, the proposed approach is also useful in relevant term extraction. We evaluate the proposed estimation methods with a testing set of query terms that were randomly selected from the testing proxy log. For Method I, the relevant terms are whose co-occurrence frequency large than 1, and for Method II the relevant terms are whose cosine value large than 0.25. The obtained preliminary result is shown in Table 5.

The first column "rank" in Table 5 is the order of the testing terms in the extracted term set, which is sorted by their occurrences. The real query terms are listed in the

"term" column, and their English translations are listed in the next column. The data in the "freq" column represents the occurrence of each query term. The "total" column indicates the numbers of all different co-occurred terms, and the "related" column the numbers of relevant terms among co-occurred terms that were checked manually. The next nine columns are the obtained statistics of the proposed methods. Each method consists of three columns, the first is the number of extracted relevant terms, the second is the number of correct relevant terms, and the third is the obtained accuracy. Note that the third method is the result merged with the proposed two methods.

						method 1			method 2			merge		
rank	term	translation	freq	total	related	extract	related	accuracy	extract	related	accuracy	extract	related	accuracy
2	聊天室	chatroom	149	448	157	66	48	0.73	50	27	0.54	116	75	0.65
4	mp3		144	266	97	27	17	0.63	1	0	0	28	17	0.61
12	電影	movies	103	172	104	13	12	0.92	7	7	1	20	19	0.95
14	台灣大學	Taiwan U.	100	152	81	11	11	1	26	21	0.81	37	32	0.86
38	政治大學	[Univ.]	63	88	43	11	11	1	3	3	1	14	14	1
40	中國時報	Chinatimes	62	102	56	10	10	1	13	5	0.38	23	15	0.65
45	sina		60	146	46	15	13	0.87	42	14	0.33	57	27	0.47
55	pchome		52	113	48	7	6	0.86	23	1	0.04	30	7	0.23
56	華視	CTV	52	89	29	8	6	0.75	13	6	0.46	21	12	0.57
63	日本	Japan	48	92	55	6	6	1	0	0		6	6	1
111	雲林科技大學	[Univ]	34	86	54	11	11	1	38	35	0.92	49	46	0.94
116	音樂	music	34	76	46	2	1	0.5	4	4	1	6	5	0.83
204	陽明大學	[Univ]	24	35	28	4	4	1	6	5	0.83	10	9	0.9
233	司法院	Judicial Yuan	22	30	24	2	2	1	0	0		2	2	1
300	故宮	Ching Palace	17	31	19	0	0		8	8	1	8	8	1
345	證期會	[government]	16	24	21	2	2	1	5	4	0.8	7	6	0.86
531	輸入法	input method	12	14	12	2	2	1	5	4	0.8	7	6	0.86
654	九份	[Place]	10	15	11	1	1	1	4	3	0.75	5	4	0.8
760	高雄科技大學	[Univ.]	9	45	33	7	7	1	33	23	0.7	40	30	0.75
789	插圖	pictorial	9	32	10	1	0	0	16	5	0.31	17	5	0.29
818	潮州高中	[school]	9	35	5	7	0	0	22	3	0.14	29	3	0.1
884	幾米	[painter]	8	12	11	1	0	0	4	4	1	5	4	0.8
1032	宏基戲谷	[web site]	7	24	19	0	0		14	9	0.64	14	9	0.64
1092	戲劇	drama	7	20	15	0	0		7	7	1	7	7	1
1124	樂譜	music notation	7	14	13	0	0		7	7	1	7	7	1
1343	達文西特展	[Exhibition]	6	4	4	3	3	1	4	4	1	7	7	1
1629	北海道	Hokkaido	5	14	5	1	1	1	8	0	0	9	1	0.11
2220	玻璃	glass	4	19	7	0	0		14	4	0.29	14	4	0.29
2454	史記	[history book]	4	7	5	0	0		5	3	0.6	5	3	0.6
2491	圖形	graph	4	7	7	1	1	1	3	3	1	4	4	1
2515	大聯全球科技基金	[Mutual Fund]	4	3	3	1	1	1	0	0		1	1	1
2668	全華	[Publisher]	3	5	5	0	0		3	3	1	3	3	1
2900	米勒	Miller	3	6	5	0	0		3	3	1	3	3	1
3119	嶺東商專	[school]	3	8	8	1	1	1	4	4	1	5	5	1
3885	+手機	cell phone	3	5	3	0	0		2	1	0.5	2	1	0.5
4378	證券暨期貨市場發展基金會	[foundation]	2	6	6	0	0		5	5	1	5	5	1

4429	電影介紹	introduction of movie	2	2	2	0	0		0	0		0	0	
4432	格鬥	wrestling	2	7	1	0	0		6	0	0	6	0	0
4858	+智邦	[company]	2	1	1	1	1	1	0	0		1	1	1
5094	電子商務之發展	development of e-commerce	2	7	4	0	0		6	3	0.5	6	3	0.5
5274	保甄	[school addmission]	2	3	2	0	0		0	0		0	0	
5524	血管炎	endangeitis	2	2	1	0	0		0	0		0	0	
5699	藝軒	[bookstore]	2	3	2	0	0		2	1	0.5	2	1	0.5
7083	spinal^cord^compression		2	1	1	1	1	1	0	0		1	1	1
7243	木山層	[geographic term]	2	4	4	0	0		3	3	1	3	3	1
7290	小木屋	wood house	2	1	1	1	1	1	0	0		1	1	1
7563	女	female	2	7	7	0	0		6	6	1	6	6	1
8044	聯合航空公司	United Air Line	2	2	1	0	0		0	0		0	0	

Table 5. The performance obtained with the proposed methods.

Analyzing Table 5., we can find that Method I favors high frequency terms (e.g., term frequency > 50). It is really suited in applications that need not many but accurate relevant terms. However, for the query terms with not high frequency, we might rely on Method II. On the other hand, for those low frequency terms (term frequency < 10), Method II can not maintain a consistent performance. The effectiveness of this method is not reliable. In order to realize the effectiveness of the obtained result, we list an example of the extracted relevant query terms in Table 6.

The query term is 台灣大學 (Taiwan University). The obtained relevant terms can be classified into 4 major categories. The first category is abbreviations including 台大, +台大 ("+" is the query syntax). The second is synonyms with different character forms like 臺灣大學, with additional prefix like 國立台灣大學, or nick name in semantics like 椰林 (Palm trees). The third is the sub divisions of Taiwan University includes 台大醫學院 (medical school), 台灣大學計算機中心 (computing center), 台大圖書館 (library), 台大電機 (department of electrical engineering), 台大中國文學系 (department of Chinese literature), 台大醫學系 (department of medicine). The final category is the events happened in Taiwan

University, like 招標, 工程, 大學聯招放榜 and 榜單.

	Query	Frequency	
Search	台灣大學	100	
	Similar Query	Co-occur > 2	related
method 1 co-occurrence	台大	25	Y
	台大圖書館	5	Y
	台灣大學圖書館	5	Y
	+台灣大學	3	Y
	+台大	3	Y
	圖書館	3	Y
	台大醫學院	3	Y
	臺灣大學	2	Y
	國立台灣大學	2	Y
	台灣大學計算機中心	2	Y
	成功大學	2	Y
	Similar Query	Threshold > 0.25	related
method 2 VSM-like method	台大圖書	0.707	Y
	招標^台大	0.667	Y
	臺大物理	0.600	Y
	台大電機	0.600	Y
	台大中國文學系	0.600	Y
	台灣學大	0.510	Y
	台大醫學系	0.475	Y
	台灣大學^工程	0.458	Y
	台灣大學^招標	0.458	Y
	時報育樂	0.402	
	大學聯招放榜	0.402	Y
	時報育樂股份有限公司	0.402	
	社會學	0.397	Y
	時報	0.328	
	比賽	0.312	
	椰林	0.305	Y
	台大醫學院圖書館	0.278	Y
	榜單	0.272	Y
	市立圖書館	0.262	

Table 6. An example of relevant terms extracted with the proposed methods.

For more references, there are several examples that were not used in the testing are also illustrated below:

- 手機 76
  - 台灣大哥大:6 中華電信:5 易利信:4 T10:2 諾基亞:2 安瑟:2 桌面王:2 行動電話:2 motorola:2 中古手機:2 全虹:2 sagem:2
- 圖片 45
  - 皮卡丘:7 卡通圖片:7 神奇寶貝:4 圖庫:4 圖:3 kitty:3 聊天室:3 漫畫:3 趴趴熊:3 皮卡丘圖片:2 小番薯:2 美麗人生:2 母親節:2 美女:2 動畫:2 康乃馨:2 日本卡通:2 圖畫:2 桌面王:2 布丁狗:2

- 圖庫 41
  - 圖片:4 動畫:3 圖檔:3 網頁製作:2 圖:2 網頁圖庫:2 遊戲下載:2 世界地圖:2 地圖:2
- 替代役 20
  - 國防部:5 社會役:3 兵役:3 南投縣政府:2 內政部:2 國防役:2

6. Conclusion

In this paper, a new approach based on log analysis is proposed for implementing interactive Web search. The most important feature of the proposed approach is that the suggested terms corresponding to a user query are extracted from similar query sessions, rather than from the contents of the retrieved documents. Furthermore, the estimation of term relevance is also based on co-occurrence analysis of the query terms in query sessions. The experiment results presented in this paper are based on analysis of the proxy server logs. The results obtained so far demonstrate that the proposed approach is quite promising in respect to improving the effectiveness of interactive web search engines.

The results presented in this paper is just a beginning of mining log data toward developing more effective web search engines. Since this approach already demonstrates quite promising results, further investigation on mining log data deserves more of our attention. Further study may result in more advanced mining mechanism that can give us more comprehensive information about term relevance and allow us to identify users' information need more effectively. For example, some sort of thesaurus information may be derived from mining log data.

References

[1] AltaVista. http://www.altavista.com.

[2] P.G. Anick and S. Tipirneni, “The Paraphrase Search Assistant: Terminology Feedback for Iterative Information Seeking,” in Proceedings of 22nd International Conference on Research and Development in Information Retrieval (SIGIR-99), pages 153-159, 1999.

[3] E.F. de Lima and J.O. Pedersen, “Phrase recognition and expansion for short precision-biased queries based on a query log,” in Proceedings of 22nd International Conference on Research and Development in Information Retrieval (SIGIR-99), pages 145-152, 1999.

[4] Direct Hit. http://www.directhit.com.

[5] Infoseek. http://www.infoseek.com.

[6] B.J. Jansen, A. Spink, J. Bateman, and T. Saracevic, “Real life information retrieval: A study of user queries on the web,” SIGIR FORUM, 32(1), 1998.

[7] S. Jones and M.S. Staveley, “Phrasier: a System for Interactive Document Retrieval Using Keyphrases,” in Proceedings of 22nd International Conference on Research and Development in Information Retrieval (SIGIR-99), pages 160-167, 1999.

[8] Kimo. http://www.kimo.com.tw.

[9] C. Silverstein, M. Henzinger, H. Marais, and M. Morics., “Analysis of a very large AltaVista query log,” Technical Report 1998-014, Digital Systems Research Center, 1998.

[10] B. Velez, R. Weiss, M.A. Sheldon and D.K. Gifford, “Fast and Effective Query Refinement,” in Proceedings of 20th International Conference on Research and Development in Information Retrieval (SIGIR-97), pages 6-15, 1997.

[11] E. Voorhees and D.K. Harman, “Overviw of the sixth text retrieval conference TREC-5,” in Proceedings of the Fifth Text REtrieval Conference (TREC-5), 1997

[12] E. Voorhees and D.K. Harman, “Overviw of the sixth text retrieval conference TREC-6,” in Proceedings of the Sixth Text REtrieval Conference (TREC-6), 1998

[13] D. Wessels, SQUID Frequently Asked Questions. Section 6. Squid Log Files. http://www.squid-cache.org/Doc/FAQ/FAQ-6.html

[14] Yam. http://www.yam.com.tw.

網際網路 FAQ 檢索中意圖萃取與語意比對之研究

賴育昇，李坤霖，吳宗憲

國立成功大學資訊工程研究所

Page 135 ~ 155

Proceedings of Research on Computational Linguistics

Conference XIII (ROCLING XIII)

Taipei, Taiwan

2000-08-24/2000-08-25

網際網路 FAQ 檢索中意圖萃取與語意比對之研究

賴育昇、李坤霖、吳宗憲

國立成功大學資訊工程研究所
{laiys, leekl, chwu}@csie.ncku.edu.tw
Fax: +886-6-2747076

摘要

本論文之主要目的是希望能利用自然語言查詢來做為 FAQ 檢索的方式。一個完整的 FAQ 樣本必定含有一個問題與該問題的答案。藉由比較使用者的詢問句以及 FAQ 樣本的問句，如果兩者的語意相當接近，則該 FAQ 樣本的答案也就可能包含使用者想要的資訊。此外，一個 FAQ 樣本的答案也可能包含其他額外的資訊。因此，除了兩個疑問句的比對之外，使用者所需的資訊也可以透過比對詢問句與 FAQ 樣本的答案而得到。

透過語意文法以及停用詞的篩選，我們將問句分成兩個部分：「意圖區段」和「關鍵詞區段」。意圖區段傳達使用者主要的意圖，關鍵詞區段包含問句中所有的關鍵詞，問句句意的比對將建立在這兩部分各自的語意比對上。此外，我們採用向量空間模型來比較詢問句中的關鍵詞與 FAQ 樣本的答案。

經實驗驗證，本論文所提出的方法確實比單純使用關鍵詞查詢來得準確，使平均正確答案的排名從第 12.04 名提升到第 2.91 名，且使得前十名的召回率由 78.06%提升到 95.11%。

1. 緒論

1-1. 背景說明

目前資訊檢索(information retrieval)的技術已經廣泛使用在我們日常生活中。舉凡上圖書館借書、網路搜尋資料，我們常會需要一些資訊檢索的工具協助我們找出想要的資料。以目前的技術，資訊檢索的應用大多只提供由關鍵詞進行查詢，藉由關鍵詞的比對以找出相關的文章或資料。但是，只利用關鍵詞查詢有兩個缺點：(1)關鍵詞不能清楚且完整地表達使用者的意圖，以致相關的搜尋結果過多，使用者往往需要經過好幾次的來回修改關鍵詞或查詢方

式才能得到想要的結果。(2)當使用者想要查詢的資料不存在關鍵詞，或者使用者無法找到適當的關鍵詞，則甚至無法找到所需的資料。

相較於關鍵詞查詢，使用自然語言查詢是最能夠清楚表達使用者意圖的方式，也是最自然的方式。隨著網路的蓬勃發展以及自然語言處理技術的提昇，以自然語言為主的資訊檢索是一個正在興起的研究方向。目前已有幾個網站提供自然語言查詢的服務：在國外有 Ask Jeeves 網站[1]以及 FAQ Finder 系統[7]，國內有寶來證券的 E 博士[5]。但是由於目前電腦技術還不能做到完全理解自然語言的意義，以致使用自然語言來做資訊檢索的研究尚未成熟，但是這卻是未來資訊檢索必定要發展的方向。若能使之結合前端的語音辨識，直接利用語音查詢，將是更加便利且人性化的一種方式。

1-2. 研究動機與目的

在以自然語言查詢為主的資訊檢索應用中，FAQ (Frequently Asked Questions)檢索是一個不錯的方向。許多網站通常會針對該領域中常被問到的問題，經由人工整理這些問題及答案，提供給進入該網站的使用者直接閱覽，以節省詢問與回答重複或相關性問題的時間。但是隨著量的增加，使用者也愈來愈難藉由直接閱覽找到所需的答案，因此，現今許多網站也提供 FAQ 檢索的服務，讓使用者搜尋所需的資訊。本論文之主要目的便是希望能利用自然語言查詢來做為 FAQ 檢索的方式。

1-3. 研究方法簡介

一個完整的 FAQ 樣本必定含有一個問題與該問題的答案。藉由比較使用者的詢問句以及 FAQ 樣本的問句，如果兩者的語意相當接近，則該 FAQ 樣本的答案也就可能包含使用者想要的資訊。此外，一個 FAQ 樣本的答案也可能包含其他額外的資訊。因此，除了兩個疑問句的比對之外，使用者所需的資訊也可以透過比對詢問句與 FAQ 樣本的答案而得到。

FAQ Finder 系統利用 Word-Net 來衡量英文詞與詞的語意相似度，為整個系統發展語意相似度的基礎。但是在問句的相似度部分，則是單純地比對兩個問句中所包含的詞組，我們認為僅僅是比較詞組並不足以代表整個句意、有欠周延，而且也有明顯的缺失。例如：「肝癌會不會導致肝硬化？」、「肝硬化會不會導致肝癌？」，此二句有完全相同的詞組，但是在意義上卻是完全不同。

每個問句都有其意圖(intention)，該意圖唯一而且在句子裡扮演相當重要的角色。本研究

所提出的方法便是希望能有效地萃取出詢問句中所包含的意圖，並且藉由意圖來協助我們分辨兩個句子的語意。透過語意文法(semantic grammar)以及停用詞(stopping words)的篩選，我們將問句分成兩個部分：「意圖區段(intention segment, IS)」和「關鍵詞區段(keyword segment, KS)」，問句句意的比對將建立在這兩部分各自的語意比對上。此外，在關鍵詞的比對上，我們依舊保留目前被廣泛使用的關鍵詞查詢為基礎的資訊檢索技術—向量空間模型(vector space model, VSM)，用來比較詢問句中的關鍵詞與 FAQ 樣本的答案。

2. 系統架構

如圖 1 所示，本論文所提出之系統架構主要分為三大部分：「語意分析器」、「問句比對器」及「內文比對器」。以下針對這三個部分做一個簡單的介紹。

2-1. 語意分析器

透過語意分析器，我們可以從問句中萃取出 IS 及 KS，做為後續問句比對以及內文比對之用。語意分析器由下面幾個子部分所組成：(1) AutoTag，中研院 CKIP 小組發展的詞性標記系統，做為本系統的前處理器，將一個句子斷詞並標示詞性。(2) 關鍵詞萃取，由詞性的判斷以及停用詞的篩選，從斷詞後的句子中找出其 KS。(3) 意圖萃取，經由整理歸納的語意文法，從問句中找出其 IS。

2-2. 問句比對器

將使用者詢問句所萃取出來的 IS 及 KS 與 FAQ 的每一個問題的 IS 及 KS 逐一做比對。問句比對器可分為下面幾個子部分：(1) 剖析器(Parser)，將語意分析器萃取出來的 IS 剖析成剖析樹(IS parse tree)。(2) IS 相似度衡量，對於任兩個 IS parse tree，採用遞迴的方式配合一對一函數的最佳化，求取兩者的最大相似度。(3) KS 相似度衡量，透過比對兩個 KS 中所包含的關鍵詞相似度，配合一對一函數的最佳化，求取兩者的最大相似度。

2-3. 內文比對器

本論文採用向量空間模型，透過比對 KS 與 FAQ 答案，找出最適合回答該詢問句的答案。其中，在 Indexing 方面，以 TF×IDF 做為詞的權重，將每一個 FAQ 樣本的答案表示成實數向量。在 Content 相似度比對上，藉由向量相似的觀點，將 KS 所含的關鍵詞組與每一個 FAQ

答案所表示成的向量做比對，找出與 KS 最相關的答案。

除了上述的三大機制外，Ranking Strategy 將問句比對器及內文比對器所得到的結果，在此做一整合，最後將排名後的網頁超連結輸出。

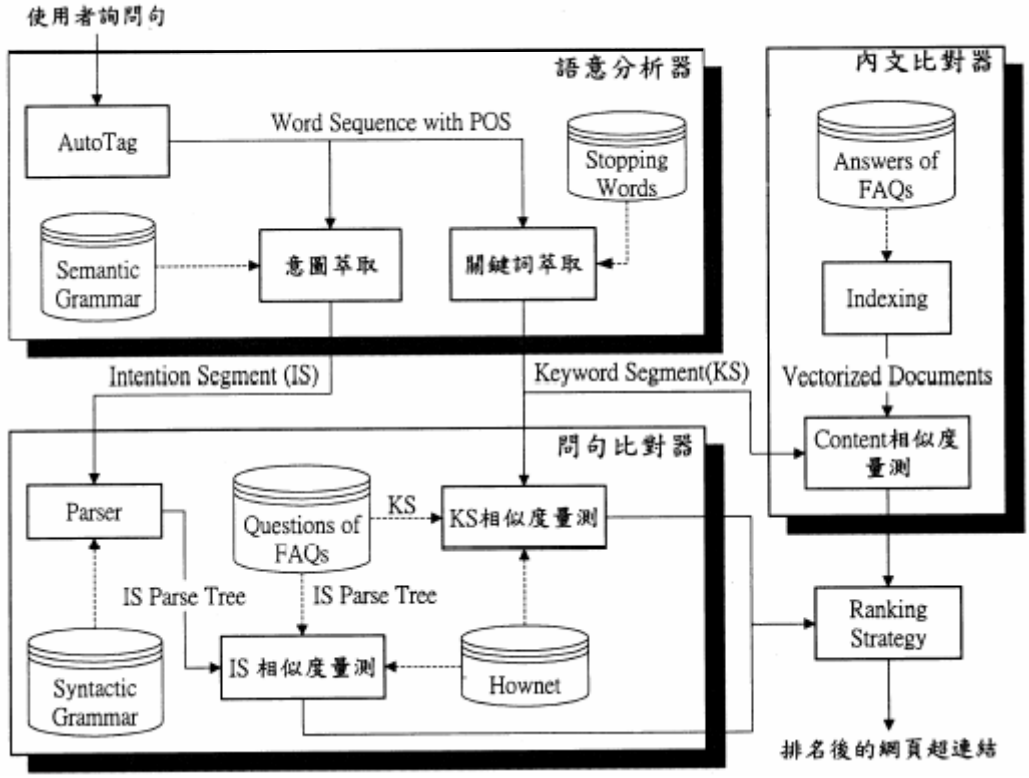


圖 1 系統架構圖

3. 問句的語意分析與處理

在大部分的情況下，關鍵詞有助於檢索出我們想要的答案，但是在符合關鍵詞比對的結果中，往往含有大量不是原來所期望獲得的答案，而其主要原因在於關鍵詞沒有辦法正確地傳達使用者的意圖。因此，我們希望透過對於問句的語意分析，能產生出問句的語意文法，進而萃取出包含在問句中的使用者意圖。

3-1. 疑問句分類

根據張鐘尹[3]的分析，就語法形式而言，疑問句可分成句子和非句子兩大類，再歸成「疑問詞問句」、「選擇問句」、「句尾語助詞問句」、「獨立語助詞問句」、「是非問句」、「附加問句」及「直述問句」等七個類型。就溝通功能而言，疑問句可分為外在訊息問句、言談問句、關

係問句及表意問句四大類。這些功能成一線性分佈，從說話者的肯定度來看，分別表示說話者不確定性高的到不確定性低的；從訊息的角度來看，則表示說話者在尋求訊息的到傳遞訊息的疑問句。同時，疑問句亦顯現出從尋求較客觀、指示性的訊息，至傳遞較主觀、以說話者為出發點的態度和看法的分佈。因此這說明即使在句構層次意義的主觀化或說話者介入程度的表達，那種機制的運作亦明顯可見。

其研究結果顯示，疑問句的語法形式與溝通功能雖是多對多的關係，其中卻仍存有某種特定的對應關係。說話者傾向於使用「疑問詞問句」、「是非問句」及「句尾語助詞問句為嗎的問句」來尋求自己不瞭解答案的外在訊息。在網際網路上的問題也多以這三種形式存在，因此，本論文即針對此三種類型的問句來做分析。

3-2. 意圖區段(Intention Segment)的定義

對一個自然語言問句而言，我們認為除了關鍵詞之外，仍有其他因素可用來分辨問句間的差異。觀察下面三個問句：「怎麼治療感冒？」、「為什麼要治療感冒？」、「治療感冒的方法有哪些？」。如果只考慮關鍵詞，則「治療」和「感冒」都為以上三句的關鍵詞。如此一來，我們就無法從關鍵詞來判斷第一和第三句應該較接近，因為此二句皆旨在詢問治療感冒的方法，而第二句則是在詢問之所以要治療感冒的原因。

因此，一個自然語言問句中的「意圖區段」，我們將其定義為：「問句中所傳達最直接想獲得的答案，不需包含前提；IS 可以是問句之子句或片語，甚至結合其他特定片語而成。」透過對於問句的分析，意義相同卻以不同句型表現的問句，所萃取出來的 IS 應該能夠保持相同。如表 1 所示，透過的 KS 及 IS 的萃取，我們可以輕易地分辨上述例句的異同。

表 1 三個相似問句所對應之關鍵詞區段(KS)及意圖區段(IS)

問句	KS	IS
怎麼治療感冒？	治療、感冒	治療感冒的方法
為什麼要治療感冒？	治療、感冒	治療感冒的原因
治療感冒的方法有哪些？	治療、感冒	治療感冒的方法

3-3. 意圖的萃取

由上一個小節的說明得知，如果能從問句中正確的萃取出 IS，對於問句意圖的辨析有很大的幫助。從語言學的角度來看，問句的語意與問句的句型息息相關。我們針對三種最常被

用在網路上的問句類型進行分析，研究問句在各種句型結構下的意圖。

3-3-1. 疑問詞問句

疑問詞問句相對於英文的 WH 問句有相當接近的地位，疑問詞通常出現在與不帶疑問訊息詞相同文法功能的位置上[3]。中文存在有許多疑問詞，例如：「什麼」、「誰」、「怎麼」、「怎麼樣」、「為什麼」、「多少」、「哪裡」、「幹嘛」、「為何」。通常疑問詞可以協助判斷問句的意圖，例如問句中如果問到「為什麼」，幾乎可以想見的該句就是在問某件事情或現象的原因；但是，有些疑問詞會隨著在句子中的相對語法位置不同，其意義也不盡相同。如表 2 所示，「怎麼」這個疑問詞，若出現在副詞之前可做為詢問某件事情或現象的原因，但若出現在動詞之前卻做為詢問做某件事的方法[16]。

表 2 疑問詞「怎麼」的意圖因語法位置的不同而有所不同

問句	意圖
要怎麼治療口臭？	治療口臭的方法
你怎麼會(能/可以)離開？	離開的原因

3-3-2. 句末語助詞為「嗎」的問句

句末語助詞問句指句子末端帶有一個語助詞像是「嗎」、「吧」、「呢」、「啊」等。當語助詞為「嗎」時，該問句對於答案相當不肯定，而需要較多的外在訊息給予解答。這類型的問句在句子中通常會包含一個「法相(modality)副詞」[16]，如「會」、「可能」、「應該」。「法相」的定義是「說話者的對一個可能事件的看法或態度」，法相副詞的定義由語意規定，其所包含的詞性含有以往語言學分類中的大多數助動詞、部分動詞及動詞，但他們卻有許多共同的語法特色。而法相副詞之後所接的是動詞片語，我們認為此動詞片語即為其意圖所在。表 3 中列舉出部分句末語助詞為「嗎」的問句及其對應的 IS。此外，如果這類型問句不含有任何法相副詞，則以主要動詞片語作為 IS，如表 4 所示。

表 3 句末語助詞為「嗎」的問句及其對應的意圖區段(IS)

問句	IS
把脈能診斷出所有疾病嗎？	能診斷出所有疾病嗎
肝炎病人應戒酒嗎？	應戒酒嗎

表 4 不含法相副詞之句末語助詞為「嗎」的問句及其對應的意圖區段(IS)

問句	IS
急性 C 型肝炎可怕嗎？	可怕嗎
子宮切片的結果正確嗎？	正確嗎

3-3-3. 是非問句

是非問句是指包含具有 A-not-AB 或是 A-not-A 特性之詞組的問句，例如：「是不是」、「可不可以」、「是否」。是非問句和句末語助詞為「嗎」的問句，相對於英文便是由 be 動詞或是助動詞開頭的問句，這兩類問句在結構上是可以互換的。同樣地，表現在 IS 上面，相同語意不同句型的問句也會具有相同的 IS。對是非問句而言，我們認為意圖為接在 A-not-A 詞組之後動詞片語。表 5 列舉出部分是非問句及其對應的 IS。

表 5 部分是非問句及其對應的意圖區段(IS)

問句	IS
感染 B 型肝炎後會不會自動痊癒？	會自動痊癒嗎
哺乳的媽媽感冒可不可以服用藥物？	可以服用藥物嗎

經由語言學上的一些研究結果，以及從收集到的問句中整理歸納，我們定義一套結合語法規則與語意的語意文法，當問句符合語意文法中某一則時，其相對應的 IS 之萃取方式也清楚的被規範著。表 6 列舉部分語意文法及其 IS 萃取方式，並舉例說明之。

表 6 部份語意文法及其例句

問句類型	問句	語意文法	IS
疑問詞問句	為什麼產後必須服用生化湯？	QW_1 NP Dba VP →IS=VP 的原因	服用生化湯的原因
句末語助詞問句	肝炎病人應戒酒嗎？	NP Dba VP →IS=VP	應戒酒嗎
是非問句	哺乳中的媽媽感冒可不可以服用藥物？	P Dba1 not Dba2 VP →IS=Dba2 VP	可以服用藥物嗎

3-4. 關鍵詞的萃取

相對於意圖的萃取，關鍵詞的萃取也是一個不可忽略的部分，藉由關鍵詞萃取我們可從問句找出其 KS。對中文而言，斷詞以及詞性標記的問題一直阻礙國內計算語言學的發展。本研究以 AutoTag 做為斷詞及詞性標記的工具，此軟體為中研院資訊所 CKIP 小組所研發的，

經由 AutoTag 的協助，可以將一個句子依照分析的結果轉換成一個帶有詞性的詞序列。

一般在做關鍵詞查詢時，多半用的是名詞或動詞，所以斷詞後，我們先從句子中找出名詞及動詞的部分。但是 AutoTag 所標記的詞性分類相當細，即使是名詞類仍有許多細分，而部分類別雖屬於名詞卻不做關鍵詞用，如定詞(Ne)、量詞(Nf)、方位詞(Ng)以及代名詞(Nh)，我們把這些詞類的詞視為非關鍵詞。

另外，有些詞雖然符合以上規則，但是出現頻率卻相當高；相對而言，其重要性便降低，視為非關鍵詞。經由統計語料庫可得到一些詞頻，將高頻的詞經過人工篩選建立一個停用詞詞典(stopping word dictionary)，當一個詞出現在停用詞詞典中，便將之從關鍵詞組裡去除。表 7 列舉出部分問句及其對應的 KS。

表 7 問句及其相對應關鍵詞區段(KS)之範例

問句	KS
中醫如何治療糖尿病？	中醫(Na)、治療(VC)、糖尿病(Na)
為什麼嬰兒呼吸有雜音？	嬰兒(Na)、呼吸(VC)、雜音(Na)

4. 詞意比對

本論文中，詞意比對是所有語意比對方法的基礎，傳統語言學認為詞是構成語意的最小單元[19]，而目前計算語言學的趨向是把詞視為許多「語意成分」(semantic features)的組合。基於後者，我們利用知網(How-net)[17]作為詞意比對的知識庫。

4-1. 知網概述

知網是針對電腦設計的雙語常識知識庫，為創建人董振東先生研究十幾年的重要成果，提供了設計人工智慧軟體所需的知識。知網共收錄了 50220 個中文詞語，所涵蓋的概念總量達 62174 個，目前仍在擴充當中。做為一個提供中文計算需求的知識庫，知網詳盡地描述了概念之間的關係，概念所具有的屬性之間的關係，以及概念與所具有的屬性之間的關係。

對一個詞而言，在不同情況下可能代表不同的概念。知網將一個概念的定義表示成特徵及標識符號的組合。表 8 列舉幾個概念在知網中之定義，其中 W_C 為一概念，G_C 表示該概念的詞類，DEF 則為其定義。在定義中，特徵間以逗號區隔，第一個特徵稱為主要特徵，表示概念的類別屬性，具有上下位關係，如圖 2 所示；後面所接的特徵則為次要特徵，用來詳細規範概念的屬性。

表 8 How-net 定義範例

W_C	G_C	DEF
警察	N	human|人, police|警
病人	N	human|人, *SufferFrom|罹患, $cure|醫治, #medical|醫, undesired|莠
鮮花	N	FlowerGrass|花草

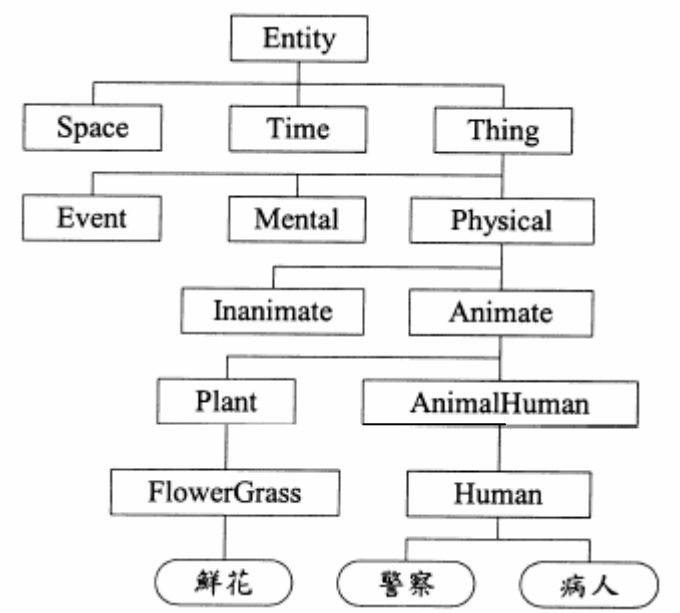


圖 2 How-net 主要特徵階層圖

4-2. 詞意相似度的量測

基於對知網的研究，我們利用知網對於每個詞彙完整的定義，量測兩個詞彙在語意上的相似度。同一個詞彙通常可表示一個以上的概念，所以兩個詞彙的相似度可以由個別的概念相似度求得，而概念相似度則是透過特徵的比對而來。如公式(1)所示，任兩個詞的語意相似度(Sim_{word})被定義成這兩個詞所有可能概念定義之間相似度(Sim_{def})的最大值。

$$Sim_{word}(w_1, w_2) = \max_{d_1 \in def(w_1), d_2 \in def(w_2)} Sim_{def}(d_1, d_2) \quad (1)$$

由於我們採用 AutoTag 做為詞性標記工具，所以可排除部份詞義混淆的情形。例如：當 w_1 同時包含名詞和動詞的概念時，若其詞性標記為動詞，則其他名詞類的概念將不予考慮。由於概念的定義是由主要特徵及次要特徵所共同描述，所以任兩個概念的相似度可定義如下：

$$Sim_{def}(d_1, d_2) = \beta \cdot Sim_{PF}(pf_1, pf_2) + (1-\beta) \cdot Sim_{SF}(sf_1, sf_2) \quad (2)$$

其中 pf_1 與 pf_2 分別是 d_1 與 d_2 的主要特徵，sf_1 與 sf_2 分別是 d_1 與 d_2 的次要特徵，Sim_{PF} 與 Sim_{SF} 分別是 d_1 和 d_2 的主要特徵與次要特徵的相似度，將會在下面的小節中做介紹，特徵結合係數 β 決定主要特徵相似度與次要特徵相似度間的權重。

4-3. 主要特徵相似度

對每一個概念，知網依照其類別屬性的不同，定義在主要特徵上，而類別屬性則構成一個階層式結構。在此階層式結構中，當兩個概念間的差異性越大，所屬的節點距離也越遠，反之則越接近。同樣的情況也發生在英文的 WordNet 上，在國外不少針對 WordNet 的研究也有類似的想法[10][12][13]。參考[8]的作法，我們將這個問題從兩個角度來思考。

一、節點(node)的觀點

從節點的角度來看，每個節點都代表唯一的概念，而且包含了特定程度的資訊量。要衡量兩個概念間的相似度，可以考慮它們所共同分享的資訊量。因此，對階層式架構中任兩個節點而言，其相似度便定義為其最接近的共同祖先節點之 information content (IC)。

根據資訊理論[14]，一個特徵 x 的 IC 可以表示成：

$$IC(x) = -\log p(x) \tag{3}$$

其中 $p(x)$ 表示具有特徵 x 或 x 的祖先特徵的這些概念在語料庫中出現的機率。對於知網的主要特徵而言，其機率值從上層單調遞減到下層的節點；而 IC 值恰與機率值相反，從下層單調遞減到上層，最上層的根節點，由於機率值等於 1，所以其 IC 值等於 0。

二、邊(edge)的觀點

從邊的角度來看，任兩個概念的相似度，可以由節點間的距離來量測；其中，兩個節點間的距離越長，其相似度越小。另外，我們也考慮深度對於任兩個相鄰節點間距離的影響，我們認為深度越大其距離越短；原因在於，對越深層的節點分類時，所描述的差別就越精細。

綜合以上兩個觀點，我們定義任兩個主要特徵的距離如下：

$$Dist(pf_1, pf_2) = \sum_{c_i \in \text{the shortest path}(pf_1, pf_2) - LSuper(pf_1, pf_2)} Cost(c_i, p_i) \tag{4}$$

其中 $LSuper(pf_1, pf_2)$ 表示 pf_1 與 pf_2 的最接近之共同祖先節點，c_i 表示 pf_1 與 pf_2 間最短路徑中除了最接近之共同祖先節點外的所有節點，p_i 則為節點 c_i 的父親節點，而 $Cost$ 代表任兩個相鄰節點間的距離，其定義如公式(5)所示。

$$Cost(c_i, p_i) = \left(\frac{d(p_i)+1}{d(p_i)} \right)^{\alpha} \left[IC_{PF}(c_i) - IC_{PF}(p_i) \right] \tag{5}$$

其中 $d(p_i)$ 表示節點 p_i 的深度(depth)，α 則為控制深度對於 *Cost* 的影響的參數。另外，由於相似度恰與距離的意義相反，因此定義主要特徵相似度如下：

$$Sim_{PF}(pf_1, pf_2) = 1 - \frac{Dist(pf_1, pf_2)}{\max_{i,j} Dist(pf_i, pf_j)} \tag{6}$$

4-4. 次要特徵相似度

知網對於一個概念的定義，除了主要特徵外仍有次要特徵用以輔助標示其屬性，但是次要特徵不具有階層式關係，而且一個定義通常包含不只一個次要特徵。因此可將次要特徵表示為二元向量(binary vector)，如此一來，次要特徵相似度就可藉由量測二元向量的相似度來得到。在向量空間中，對於二元向量相似度的衡量方法有下列幾種：Dice coefficient、Jaccard coefficient、Overlap coefficient、以及 Cosine 等[11]。由於每個次要特徵的重要性不一，如果某個次要特徵經常出現在各個概念定義中，則其辨別詞意的能力就較弱，反之則愈大。因此，我們結合 Dice coefficient 與 IC 為次要特徵相似度的量測方式，定義如下：

$$Sim_{SF}(sf_1, sf_2) = \frac{2 \times \sum_{f_i \in sf_1 \cap sf_2} IC_{SF}(f_i)}{\sum_{f_j \in sf_1} IC_{SF}(f_j) + \sum_{f_k \in sf_2} IC_{SF}(f_k)} \tag{7}$$

5. 語意比對

本研究中，語意比對可分為兩個部份，一個是問句與 FAQ 問題的比對，一個是使用者詢問句中所含的關鍵詞區段跟 FAQ 答案的內文比對。因此，一個問句 q 與一則 FAQ 樣本 p 之語意相似度，可定義成公式(8)。

$$Sim(q, p) = \delta \cdot Sim_{question}(q, q(p)) + (1-\delta) \cdot Sim_{content}(q, a(p)) \tag{8}$$

其中 $Sim_{question}$ 表示該問句與 FAQ 問題之相似度，而 $Sim_{content}$ 表示該問句與 FAQ 答案之相似度，並以一比對結合係數 δ 調整兩者間的權重。

5-1. 問句比對

在問句的比對上，因為每一個問句都由 IS 以及 KS 所組成，因此我們將分別量測 IS 相似

度以及 KS 相似度之後，最後再將這兩個相似度結合成問句相似度，如公式(9)所示。

$$Sim_{question}(query, q(faq)) = \gamma \cdot Sim_{IS}(IS_{query}, IS_{question}) + (1-\gamma) \cdot Sim_{KS}(KS_{query}, KS_{question}) \quad (9)$$

其中意圖-關鍵詞結合係數γ用來調整 IS 相似度(Sim_{IS})和 KS 相似度(Sim_{KS})間的權重。

5-1-1. 意圖區段相似度

經由語意分析器所萃取出來的 IS 通常是一個簡單的名詞片語或動詞片語，但是如何去量測兩個片語間的相似程度呢？考慮下面的例子：

P1：「吃心臟病藥」

P2：「吃治療心臟病的藥」

P3：「治療心臟病的藥」

如果將「吃」、「治療」、「心臟病」、「藥」視為關鍵詞，可以想見的我們將無法分辨 P1 與 P3 何者較為接近 P2，因為 P1 和 P3 同時擁有 P2 四個詞中的三個。但是經由對語言的理解，卻可以清楚的分辨 P1 應該比較接近 P2，甚至相同。

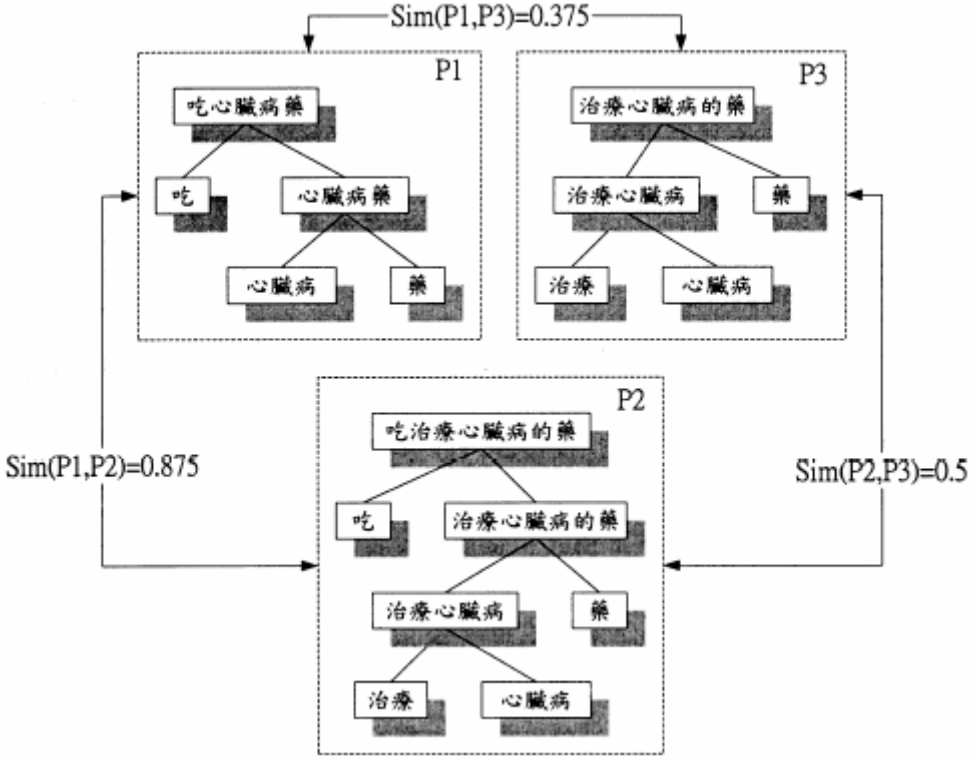


圖 3 IS 剖析樹比對示意圖

因此，我們將片語化成剖析樹(parse tree)，透過剖析樹的比對，來解決上述的問題。如圖

3，在分別建立三個片語的剖析樹之後，可以清楚地發現 P2 的「治療心臟病」這個動詞片語作為形容「藥」之修飾語，其地位相同於 P1 中的「心臟病」，亦為「藥」的修飾語。整體來看，P1 與 P2 都為動詞片語，其動詞都是「吃」，吃的對象都是「藥」，唯有在「藥」的修飾語上略有不同。另一方面，P3 恰為 P2 之子樹，相對而言，兩者之相似度應該小於前述之相似度。

我們參考 CKIP Tree Bank[2]整理部分的語法規則，再根據 Earley algorithm[6]建立一個語法剖析器。其中，若剖析樹之外部節點詞性為介詞或連接詞，則省略該分支以節省比對時間。對於任兩個 IS 剖析樹 T_1 與 T_2 ，我們定義比對公式如下：

$$
\begin{aligned}
&Sim_{IS}(IS_1, IS_2) \\
&= Sim_{tree}(T_1, T_2) \\
&= \begin{cases}
Sim_{word}(T_1, T_2), \text{若 } T_1 \text{ 和 } T_2 \text{ 都是單節點樹} \\
\dfrac{1}{|T_1|}\max\limits_i Sim_{tree}(T_{1,i}, T_2), \text{若 } T_2 \text{ 是單節點樹，而 } T_1 \text{ 不是} \\
\dfrac{1}{|T_2|}\max\limits_j Sim_{tree}(T_1, T_{2,j}), \text{若 } T_1 \text{ 是單節點樹，而 } T_2 \text{ 不是} \\
\max\{\dfrac{1}{|T_1|}\max\limits_i Sim_{subtree}(T_{1,i}, T_2), \dfrac{1}{|T_2|}\max\limits_j Sim_{subtree}(T_1, T_{2,j}), Sim_{subtree}(T_1, T_2)\}, \text{其他}
\end{cases}
\end{aligned}
\tag{10}
$$

其中 $Sim_{word}(T_1, T_2)$ 表示兩個單節點 IS 剖析樹間的相似度，$T_{1,i}$ 和 $T_{2,j}$ 分別表示 T_1 和 T_2 的子樹，$|T_1|$ 和 $|T_2|$ 分別表示 T_1 和 T_2 子樹的個數，$Sim_{subtree}$ 表示兩個非單節點 IS 剖析樹間的相似度，其定義如下：

$$
Sim_{subtree}(T_1, T_2) = \max_g \frac{\sum_{k=1}^{|T_A|} Sim_{tree}(T_{A,k}, g(T_{A,k}))}{|T_A|} \tag{11}
$$

其中 g 是一個從 T_A 到 T_B 的一對一函數，$T_{A,k}$ 表示 T_A 的一個子樹，$|T_A|$ 表示 T_A 子樹的個數。由於 g 為一對一函數，所以 $|T_A| \le |T_B|$，因此需要特別注意：若 $|T_1| \le |T_2|$，則設定 $T_A = T_1$ 且 $T_B = T_2$，否則設定 $T_A = T_2$ 且 $T_B = T_1$。

當 T_1 和 T_2 都是外部節點的時候，表示此二者皆為詞，對於兩個詞的相似度，就利用公式(1)所描述的詞意相似度來量測。當 T_1 或 T_2 其中之一為外部節點時，表示其中一個為詞另一個則為一個片語，此時則遞迴向下找出該片語中與該詞最相似的詞。當 T_1 和 T_2 都不為外部節點

時，就表示T_1和T_2都含有各自的子樹。此時，可以從三個方向來思考：最基本的想法，若兩顆樹的所有子樹都非常相似，則這兩顆樹可能是非常相似的，因此考慮$Sim_{subtree}(T_1,T_2)$作為T_1和T_2的相似度；另外，如果T_1相似於T_2的一個子樹，或是T_2相似於T_1的一個子樹，則根據分支的多寡來決定該相似度之權重。

5-1-2. 關鍵詞區段相似度

在量測兩個 KS 的相似度上，我們做了一個假設：對任一個關鍵詞而言，不會有兩個或兩個以上的關鍵詞與它對應。而這種對應關係恰可以一對一對應函數表示之，所以我們提出公式(12)來量測兩個 KSs $K_1=\{w_1,w_2,\cdots,w_m\}$和$K_2=\{t_1,t_2,\cdots,t_n\}$的相似度。

$$Sim_{KS}(K_1,K_2)=\max_f \frac{\sum_{i=1}^{|A|} Sim_{word}(a_i,f(a_i))}{|A|} \tag{12}$$

其中f是一個從A到B的一對一函數，a_i是A中的一個元素，$Sim_{word}(a_i,f(a_i))$表示關鍵詞a_i與其對應的關鍵詞的詞意相似度。如同前一小節，需特別注意：若$m\le n$，則設定$A=K_1$且$B=K_2$；反之，則設定$A=K_2$且$B=K_1$。

5-2. 內文比對

除了問句比對外，我們也利用問句與 FAQ 答案的比對來協助找出所需的答案，使用的方法則是目前被廣泛使用在資訊檢索應用的 vector space model (VSM)。VSM 主要分成兩個步驟：(1) 萃取特徵並以向量來描述之，(2)比較兩個特徵向量在向量空間中的夾角。本研究中，特徵向量是由每個關鍵詞的 TF×IDF 權重所構成。針對問句及 FAQ 答案求取個別的特徵向量$\vec{u}=\{a_1,a_2,\cdots,a_N\}$和$\vec{v}=\{b_1,b_2,\cdots,b_N\}$；然後利用餘弦公式計算其夾角，夾角愈小表示兩向量愈接近，以此做為該問句與 FAQ 答案的相關程度，如公式(13)所示。

$$Sim_{content}(\vec{u},\vec{v})=\rho_{\cos}(\vec{u},\vec{v})=\frac{\sum_{i=1}^{N} a_i b_i}{\sqrt{\sum_{i=1}^{N} a_i^2}\sqrt{\sum_{i=1}^{N} b_i^2}} \tag{13}$$

其中N表示特徵向量的維度，也就是詞彙量。

6. 實驗結果與討論

本研究中，我們實驗使用的機器為 Pentium III 450 個人電腦，128 MB RAM，開發用的程式語言是 Microsoft Visual C++ 6.0。除了實驗測試之外，也透過 IIS 4.0 架設了一個網站，開放給網路上的使用者查詢，網址在 http://chinese.csie.ncku.edu.tw/faq/。在語料庫的收集方面，我們以人工在網路上收集了 1,022 則 FAQ，內容主要包括醫藥以及投資理財相關之 FAQ。

在系統評估方面，我們請 10 位非系統開發人員，並告知本網站所提供資訊的內容範圍，以人工的方式建立 185 則問句並標記與其相關之 FAQ。有別於關鍵詞資訊檢索，自然語言問句之意圖較明確，因此每則問句所對應的答案相當少，平均只有 1.36 則。因此，我們不使用精確率(precision rate)來衡量系統的效能，因為即使第一名就是正確答案，精確率仍會隨著名次增加而遞減。我們提出一個較恰當的評估方式—平均正確答案排名，其定義如下：

$$\text{平均正確答案排名}(AvgRank)=\frac{\sum \text{正確答案所在之名次}}{\text{正確答案個數}} \tag{14}$$

6-1. 意圖區段萃取實驗

根據語料庫中問句的語法型態，訂定了 85 條語意文法。為了測試根據該語意文法所萃取出來的 IS 的正確性，以人工建立 185 則問句來做測試，並以人工檢驗是否符合原本預期的結果。檢驗時，若其誤差不影響意圖的辨別，則視為正確萃取，經統計可達到 91.89%的正確萃取率，其中無法正確萃取的情況可分為以下幾種：

一、 屬於疑問詞問句、是非語句、句末語助詞為「嗎」之外的問句，由於並未在語意文法中定義其萃取方式，所以屬於「超越文法範圍 (out-of-grammar) 」而無法萃取。

二、 問句結構過於複雜甚至帶有兩個疑問子句，對於這類型問句目前仍無法處理。

三、 在 AutoTag 斷詞及標示詞性時已經出錯，導致後面意圖萃取無法正確判斷。

6-2. 基準系統

本實驗以關鍵詞查詢為基準(baseline)，與自然語言查詢做比較。因此我們令公式(8)中的係數 $\delta=0$，使得僅由內容比對來決定整體之相似度。經由統計每一條測試句之答案排名，結果獲得平均正確答案排名為 12.04 名，並得到前 N 名的召回率(recall rate)表列如下：

表 9 基線系統之前 N 名召回率

Top N	1	2	3	4	5	6	7	8	9	10
召回率 (%)	36.06	48.56	56.00	60.22	63.89	66.56	73.56	73.56	76.72	78.06

6-3. 詞意相似度之實驗

6-3-1. 主要特徵相似度之深度影響係數實驗

從公式(5)中得知，係數α決定深度對於任兩相鄰節點間距離(*Cost*)的影響程度，為了找出α之最佳值，我們固定公式(2)中的係數$\beta=1$，也就是完全以主要特徵相似度做為詞意相似度；公式(9)中的係數$\gamma=0$，表示不考慮 IS 對問句相似度之影響；公式(8)中的係數$\delta=1$，表示完全以問句相似度作為檢索的依據，然後根據平均正確答案排名來決定α之最佳值。

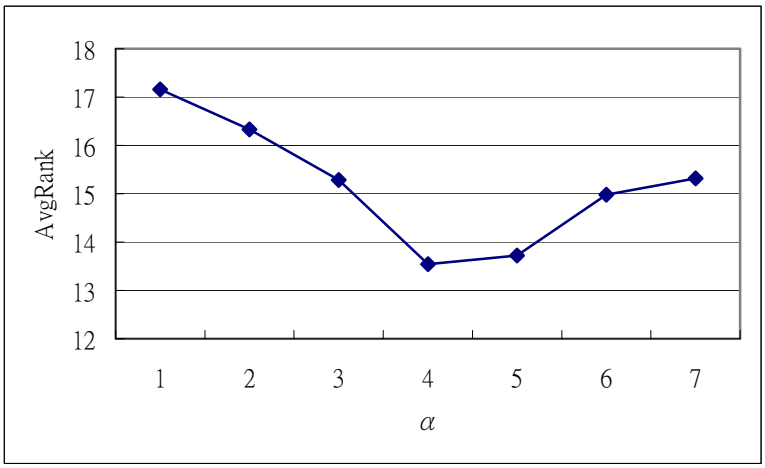


圖 4 係數α相對於平均正確答案排名之比較圖

如圖 4 所示，當$\alpha=4.0$時，其平均正確答案排名 13.54 為最佳結果，此結果與「深度越深則節點間距離越短」的觀點相符。

6-3-2. 次要特徵相似度計算方式實驗

本實驗比較四種二元向量相似度量測方式對系統效能的影響。我們固定$\beta=0$，也就是完全以次要特徵相似度為主，$\gamma=0$即不考慮 IS，$\delta=1$完全以問句比對來評估，結果表列如下：

表 10 比較各種二元向量相似度量測係數對系統平均正確答案排名之影響

	Dice coefficient	Jaccard coefficient	Overlap coefficient	Cosine
平均正確答案排名	6.28	6.29	7.61	6.51

表 10 顯示使用 Dice coefficient 之結果為最佳，所以在接下來的實驗都採用 Dice coefficient 來作為次要特徵相似度之量測方法。

6-3-3. 主要特徵與次要特徵之結合係數實驗

概念定義的相似度由主要特徵相似度及次要特徵相似度結合而來，因此本實驗的希望得到特徵結合係數 β 對系統效能的影響，同樣地，我們固定係數 $\gamma=0$ 與 $\delta=1$。如圖 5 所示，該實驗結果顯示，$\beta=0.3$ 使得平均正確答案排名達到 5.89 為最小。

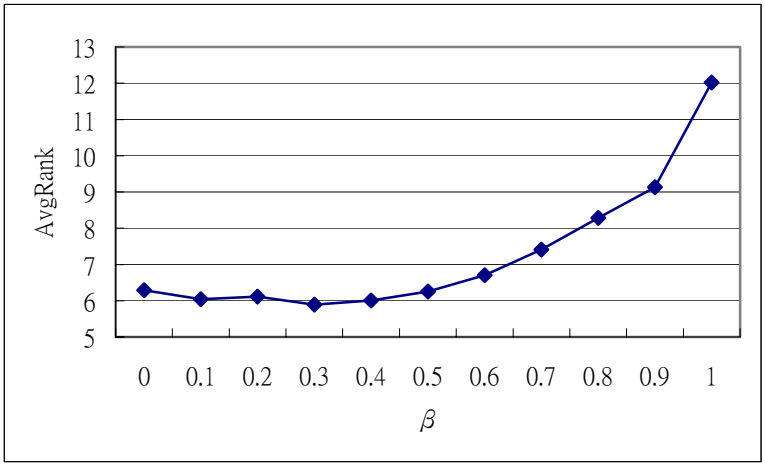


圖 5 特徵結合係數正確答案排名比較圖

6-4. 句意相似度實驗

由公式(9)，本實驗想了解意圖-關鍵詞結合係數 γ 對系統效能的影響，因此固定實驗值 $\alpha=4$ 與 $\beta=0.3$ 以及尚未實驗的 $\delta=1$。由圖 6 得知，當 $\gamma=0.3$ 時，其平均正確答案排名 3.59 為最佳結果。此外，當 γ 較大時，曲線迅速上揚，表示當 IS 相似度的比重過大時，其結果並不理想。這是因為 IS 僅包含問題的意圖，並未將前提包含進來；因此，IS 並不能完全取代 KS，而是相輔相成。

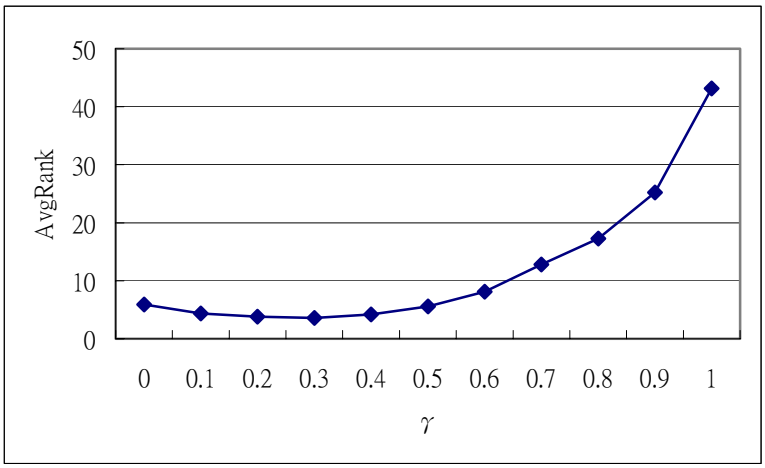


圖 6 意圖-關鍵詞結合係數γ之於平均正確答案排名比較圖

6-5. 問句相似度與內文相似度之結合係數實驗

由公式(8)，問句與 FAQ 樣本的比對由問句的相似度與內文相似度共同決定，因此本小節實驗其係數δ。實驗結果顯示，$\delta = 0.5$ 時，其平均正確答案排名落在 2.91 為最佳結果。觀察圖 7，δ在範圍[0.2, 1.0]中時，對系統效能的影響並不大；可得知，相較於內文相似度，問句相似度對系統效能的影響較大。

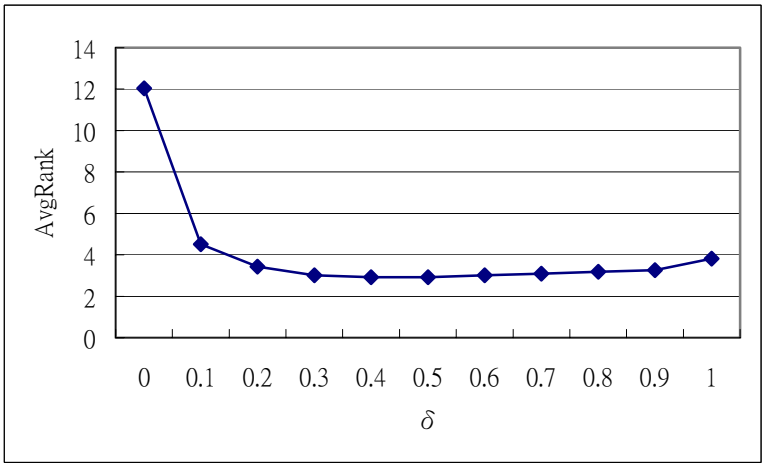


圖 7 比對結合參數δ之於平均正確答案排名比較圖

6-6. 實驗總結

最後，藉由控制參數，將各個方法以平均正確答案排名與召回率做一個比較。由圖(8)和圖(9)可以發現，無論從平均正確答案排名或是前 N 名的召回率來看，本論文所提出的方法明顯地改善了效能。相較於基準系統，平均正確答案排名約進步了 9 個名次。 第一名的召回率

從 36.06%提升到 64.67%，約提昇了 80%；而前十名的召回率也從 78.06%提升到 95.11%。

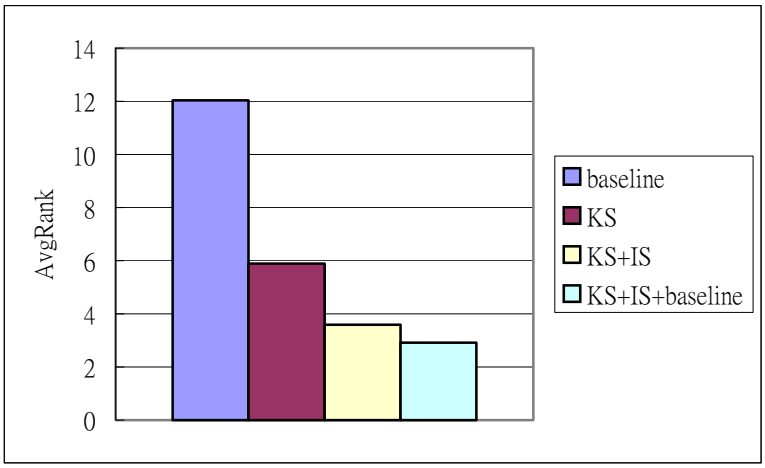


圖 8 系統平均正確答案排名比較圖，其中 baseline 表示只比較內文的關鍵詞，KS 表示只比較問句的關鍵詞，IS 表示只比較問句的意圖

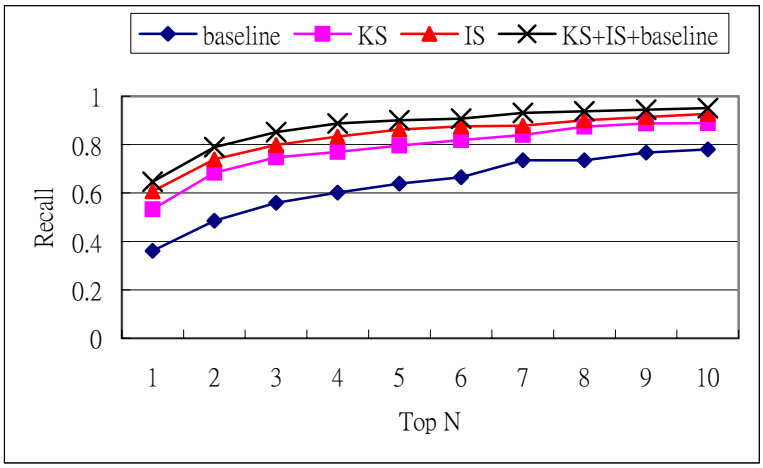


圖 9 系統召回率比較圖，其中 baseline 表示只比較內文的關鍵詞，KS 表示只比較問句的關鍵詞，IS 表示只比較問句的意圖

7. 結論與未來展望

本論文提出以問句意圖萃取以及語意比對的方法，應用到自然語言 FAQ 檢索上。經實驗驗證，該方法確實比單純使用關鍵詞查詢來得準確，使平均正確答案的排名從第 12.04 名提升到第 2.91 名，且使得前十名的召回率由 78.06%提升到 95.11%，但是其中仍存在一些待改進之處：

一、意圖萃取方面，雖然我們能處理 92%的語料，仍有許多問句的型態不在收集的範圍內，以及對於較複雜語法問句的誤判，可以藉由改善語意文法上來解決。

二、在語意相似度方面，我們採用知網做為詞意相似度量測的知識庫，但是知網中沒有定義的詞，則無法藉由它來量測詞意相似度。解決的方法有二：一是增加未定義詞到知識庫中，另一個是找出自動建立知識庫的方法。

三、在建立意圖區段剖析樹方面，對於剖析時普遍遭遇到詞性不明確的問題(ambiguity)，仍有困難無法克服。考慮現有資源，可以先建立機率剖析器[4][15]，進而建立包含語意之剖析器。

四、在自然語言理解方面，目前的系統並未具備推理能力，在許多情況下，詞語的組合可能引申另外的意義。這些會遭遇到但仍無法解決的問題，有待未來持續地研究。

參考文獻

[1] Ask Jeeves, http://www.ask.com.

[2] CKIP Tree Bank, http://godel.iis.sinica.edu.tw/CKIP/trees1000.htm.

[3] Chang, Chung-Yin, "A Discourse Analysis of Questions in Mandarin Conversion," M.A. Thesis, National Taiwan University Graduate Institute of Linguistics, June 1997, pp. 16-81.

[4] Collins, M. J., "Head-driven Statistical Models for Natural Language Parsing," Ph.D. Thesis, University of Pennsylvania, Philadelphia, 1999.

[5] Dr. E, http://drdai.polaris.com.tw.

[6] Earley, J., "An Efficient Context-free Parsing Algorithm," Communications of the ACM, vol. 6, no. 8, 1970, pp. 451-455.

[7] FAQ Finder, http://faqfinder.ics.uci.edu:8001.

[8] Jiang, Jay J. and David W. Conrath, "Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy," Proceedings of the ROCLING X, 1997, pp. 19-33.

[9] Li, Charles and Sandra A. Thompson, "Mandarin Chinese: A functional reference grammar," Berkeley and Los Angeles: University of California Press, 1981.

[10] Lin, D., "An Information-Theoretic Definition of Similarity," Proceedings of the International Conference on Machine Learning, July 1998.

[11] Manning, Christopher D. and Hinrich Schütze, "Foundations of Statistical Natural Language Processing," The MIT Press, 1999, pp. 296-303.

[12] Markman, A. B. and D. Gentner, "Structural Alignment During Similarity Comparisons," Cognitive Psychology, vol. 25, 1993, pp. 431-467.

[13] Resnik, P., “Using Information Content to Evaluate Semantic Similarity in a Taxonomy,” Proceedings of the 14th International Joint Conference on Artificial Intelligence, vol. 1, August 1995, pp. 448-453.

[14] Ross, S., “A First Course in Probability,” Macmillan, 1994.

[15] Stolcke, A., “An Efficient Probabilistic Context-Free Parsing Algorithm that Computes Prefix Probabilities,” Computational Linguistics, vol. 21, no. 2, 1995, pp. 165-202.

[16] 張麗麗, “現代漢語中的法相詞”, CKIP Technical Report, no.93-06, June 1993, pp. 1-16.

[17] 董振東, 董強, “知網”, http://how-net.com.

[18] 蔡維天, “The Hows of Why and the Whys of How”, 台灣語言學的創造力學術研討會, 2000, pp.1-27.

[19] 謝國平, “語言學概論”, 三民書局, 1996, pp.189-197.

從語料庫看漢語助動詞的語法特點

鄭縈

靜宜大學中文學系

Page 157 ~ 170

Proceedings of Research on Computational Linguistics

Conference XIII (ROCLING XIII)

Taipei, Taiwan

2000-08-24/2000-08-25

從語料庫看漢語助動詞的語法特點

鄭縈
靜宜大學中文系
E-mail: ycheng@pu.edu.tw

摘要

本文以中研院的平衡語料庫為基礎，重新檢討助動詞的語法特點。Li & Thompson主張助動詞不能被程度副詞(很/更)修飾，湯&湯卻以之區分情態動詞和形容詞，結果語料顯示多數助動詞可與「很」或「更」搭配；再則就助動詞的位置而言，助動詞典型的位置是在句中，後接動詞。雖然學者都肯定助動詞可出現於正反問句中，但檢驗語料的結果卻非如此。語言學理論或假設試圖闡述說話者的語言本能，以往學者研究語言現象時多以個人語感為準，對語法特點的描述流於主觀，經由語料的驗證可以反映語言的實際使用情形，並檢驗理論的可行性。

0. 前言

以往學者討論助動詞的語法特點時，多根據個人語感或所蒐集的有限語料加以歸納而成，結果出現不同學者間提出的語法特點不僅有所出入，甚至互相矛盾。以助動詞為例，Li & Thompson(1981)認為「應該、能夠、會、可以、能、可以」等助動詞不接受程度副詞(如「很」)的修飾；湯&湯(1998)則將助動詞歸於動詞或形容詞，甚至將程度副詞的修飾視為區分情態動詞與形容詞的主要條件，因此他們的情態形容詞包括「應該/可能/可以/能(夠)/願意/肯/敢/會(說話)」，自然可接受程度副詞(如「很」)的修飾。本文主旨即針對這些學者所提的語法特點以實際語料加以檢驗[1]。此處語料是指中研院詞庫小組發展的平衡語料庫，有關其詳細內容及說明請參詞庫小組(1998)。

底下內容包括三個部份：第一節簡介 Li & Thompson(1981)和湯&湯(1998)為助動詞（或稱情態動詞/形容詞)所列的語法特點，第二節即根據實際語料來檢驗這些助動詞與程度副詞的關係及其他相關的語法特點，包括助動詞與程度副詞、助動詞的否定式、助動詞的位置及正反問句等。第三節做一總結。

1. 助動詞的語法特點

Li & Thompson (1981)、朱 (1982)、湯(1984)、湯&湯(1998)討論助動詞時所描述的的語法特點不盡相同，下面以 Li & Thompson (1981)和湯&湯(1998)為例加以說明。

Li & Thompson 所列的語法特點：

1) 可以出現於正反問句

2)可加以否定

3) 不能被「很」或「更」等程度副詞修飾

[1] 本文主要根據筆者國科會計畫成果報告(”台灣國語、閩南語和客家話各類情態詞搭配關係的

4)不能在主語之前

5）助動詞必需和動詞一起出現，除非因上下文而省略動詞

6）不能帶時貌詞

7)不能名詞化

8)不能帶名詞組賓語[2]

Li & Thompson 認為就一、二點而言，助動詞和動詞無別，但二者在後六點上有所不同，故主張助動詞應單獨成一類。他根據這八個特點列出的助動詞有：「應該、應當、該、能夠、會、可以、能、敢、肯、得、必須、必要、必得」等 13 個。Li & Thompson 在註解 2 指出有些「會」的用法似乎違反第三條：「他很會說話」。作者認為這句並非反例，因「會說話」是成語，一般而言「會」仍不能用「很」或「更」來修飾：

(1)*他很/更會游泳

湯&湯(1998)認為情態動詞/形容詞的句法特徵如下所示：

1）可以出現於正反問句

2）可出現於否定副詞之前後

3）情態形容詞可被程度副詞(如「很」)修飾

4）可出現於句中甚或句首的位置，如「可能/應該」

5）可以單獨回答

6）可充當分裂句的信息焦點(如「他是可能中獎的」)

7）允許同類或不同類的情態動詞或形容詞連用

由上述比較可以看出，Li & Thompson (1981)所列的八點和湯&湯(1998)的七點中，有重疊的部份只有第 1-4 點，其中只有第 1 點兩文是一致的，第 2 和 4 點稍有不同，而兩者對第三點的看法互相矛盾。事實上第三點牽涉到助動詞的歸類問題，目前就助動詞在整個漢語詞類系統中所占的位置，各家說法可分為三大類：一、動詞說，二、副詞說，三、獨立一類。Li & Thompson 採第三種說法，且認為「應該、能夠、會、可以、能、可以」等助動詞不接受程度副詞(如「很」)的修飾；湯&湯則將助動詞歸於動詞或形容詞，甚至將程度副詞的修飾視為區分情態動詞與形容詞的主要條件，如情態形容詞包括「應該/可能/可以/能(夠)/願意/肯/敢/會(說話)」；換言之，「應該、能夠、會、可以、能、可以」等助動詞是形容詞的一種，自然可接受程度副詞(如「很」)的修飾。下面將針對 Li & Thompson(1981)及湯&湯(1997)所列的助動詞語法特點中重疊的部份(即前四項)，以實際語料來檢驗這兩篇論文所列的 17 個助動詞。

2. 助動詞語法特點的檢驗

綜合上述兩家討論的助動詞語法特點，本節分為助動詞與程度副詞、助動詞的否

比較”，編號 NSC 88-2411-H-126-005)的部份內容修改而成。

[2] 在收到的電子信件中發現有「如果星期六不能讀書會」(能+名詞組)的例外，這或許是省略(不能舉行讀書會)或漏字的結果。

定式、助動詞的位置和正反問句等四小節，以實際語料檢驗 17 個詞的分布。結果顯示多數助動詞並未完全符合學者所提的語法特點，第五小節進一步指出學者的理論與語料不完全一致：如有少數情態詞絕大多數用為名詞或名詞的修飾語；也有所謂的助動詞，其非情態用法高於情態用法，這些都應排除於助動詞之外。

2.1 助動詞與程度副詞

「應該、能夠、會、可以、能、可以」等助動詞是否可被程度副詞(如「很」或「更」等)修飾，Li & Thompson (1981)和湯&湯(1998)兩文的看法相反，此處將根據實際語料來檢驗這些助動詞與程度副詞「很」和「更」的搭配情形。檢驗結果如下(X/Y(Z%)分別表示出現次數/總次數(頻率)，以下依照「很+AUX」出現的頻率排列)：

	很+AUX	更+AUX
願意	9/293(3.07%)	2/293(0.68%)
可能	58/2000(2.90%)	1/2000(0.05%)
會	8/2000(0.40%)	9/2000(0.45%)
能	5/2000(0.25%)	24/2000(1.20%)
能夠	1/463(0.22%)	1/463(0.22%)
應當	0	1/31(3.23%)
應該	0	10/873(1.15%)
應	0	12/1381(0.9%)
可以	0	10/2000(0.5%)
敢	0	1/330(0.3%)
得 dei[3]	0	1/222(0.45%)
必須	0	1/1113(0.09%)
得 de	0	0
該	0	0
肯	0	0
必得	0	0
必要	0	0

就上表十七個可能的助動詞來看，能搭配程度副詞的助動詞佔多數，但搭配「很」的助動詞低於「更」。檢驗語料的過程中，我們發現上述結果除了說明助動詞實際上可和程度副詞搭配，更重要的是，助動詞搭配程度副詞時，可能有限制，如「會」可表能力或預斷，語料中前者主要搭配「很」而後者只與「更」共現；「應該」則兼有義務和預斷用法，但只見義務用法和程度副詞一起出現。下面針對這兩點分別舉例說明之。

2.1.1「很+會」/「更+會」

[3] 「得」有兩讀 dei3 和 de2，前者表是必須義，後者允許義，皆屬義務類，但語料顯示兩者語法特點不盡相同，故加以分別。

語料中，「很+會」8 次都是能力義，但「更+會」有 9 次，表預斷者多於能力義

a.「很+會」

(1)巨人族裡的一個英雄。他有兩條長腿，很會跑。

(2)好喜歡這個眉毛都開始變白的男人，你很會誇獎別人。

(3)國父因為從小受到祖母的影響，所以很會說故事，口才也很好。

b.「更+會」(此處「會」往往表預斷義)

(4)張學友表示，吳導演不但會教戲，更會演戲，他得獎，張學友無話可說。

(5)特別是在事業有了基礎，年過壯年，更會去追問：「人生究竟是什麼？」

(6)經常斷章取義傳述別人的話，更會以大帽壓頂的方式假傳聖旨。

(7)沒辦法時，那工商界不只要走上街頭，更會出走海外。

語料中「更+會」出現 9 次，除了例(4)是能力義外，其他 8 例為推測義，如下表所示：

	很+會	更+會
能力義	8	1
預斷義	0	8

從此表看來，「很」和「更」雖然都是程度副詞，卻搭配「會」的不同用法。

2.1.2「應該」、「應」和「該」的比較

助動詞多半有兼義現象，即一個詞兼有兩種以上的情態用法，以「應該」為例。「應該」有兩種用法(呂 1984)：

a. 表示情理上必須如此。

例：學習應該認真

b. 估計情況必然如此

例：他昨天動身的，今天應該到了

「應該 1」是義務類，「應該 2」是認知類。但是整理語料的過程中，發現只有前者可被程度副詞修飾。舉例如下：

a. 更+應該

(1) 並勇於創造就業的新園地，更應該體會自身創造事業所需付出的代價。

(2) 因為我們是過來人，更應該感同身受，不要用中年的眼光來看

(3) 「台灣」不但是一個研究客體，更應該是研究的主體

b. 最+應該

(4) 最應該出走並氣憤台灣的林義雄，即將回來

(5) 最應該成為台灣最值得瞻仰的林家石碑

就語料顯示，這些兼義詞被程度副詞修飾時，出現的位置有其限制：「更/最/*很應該+VP」，否定式也可以被程度副詞修飾，但出現在主要謂語的位置：「子句/名詞組+更/很/真/太不應該」，請參 2.2 節。

「應/該」一般也認為是「應該」的近義詞或同義詞，語料所見卻是以義務用法為主，所以也只見義務用法的「應/該」被程度副詞修飾：

(6)映演業者除了要加強公共安全設施外，更應加強執行電影的三級制管理

(7)同時，在污水處理方面，更應有長遠的規劃，以免帶來水污染的問題。

(8)在家庭中出現的機率愈來愈多，民眾更應具備急救常識，以免突臨意外事故時

(9)研究的奉獻態度 dedication，是研究生最該具備的特質，在此提供給大家做參考。這

「該」只出現(9)一例。

綜合上述的例子，顯示「應該/應/該」只有義務用法可與程度副詞搭配，如下表所示：

	認知	義務
很+應該/應	—	—
更+應該/應	—	＋
最+應該/應	—	＋

2.2 助動詞的否定式

17 個可能的助動詞中，可直接以「不」否定者如下所列：

	不+AUX
願意	＋
可能	＋
會	＋
能	＋
應當	＋
應該	＋
應	＋
可以	＋
敢	＋
該	＋
肯	＋
得 de	＋
必要	＋
必須	?
能夠	—
得 dei	—
必得	—

上表中有三點需要補充。

(一)除了否定詞「不」之外，還有少數可被「沒(有)」否定，就語料所見有「可能/能/必要[4]」三詞：

(1)就沒有機會上電視。上報，就<u>沒有</u>可能受到重視，也就沒有資格存活

[4] 其中「可能」和「必要」被「沒(有)」否定時，詞類上究應分析爲名詞或助動詞有待進一步研究。

在這社會

(2)是自己的負面，自己拋棄了才跟氣，沒能修成正果；而在人我之間用才氣來壓垮

(3)民生物資充裕，民眾不必要恐慌，也沒必要囤積。蕭萬長表示，國內油價調整的

(4)依照衛生署的處事心態，他們似乎沒有必要主動去幫助民間發明人，取得合法證明

(二)助動詞否定式與程度副詞

值得注意的是助動詞的否定式如「不應該」可以被程度副詞修飾，這一點和心理動詞（如「很不喜歡」)或形容詞(如「很不高興」)相同。以「不應該」為例，就語意而言，「不應該」被程度副詞修飾時都是表義務的用法；就句法而言，此時「很/更/太不應該」做為主要謂語，以子句為主語。如以下例子所示：

(5) 他一想，這個事情很不應該。所以他就從這個好漢組退出來

(6) 主秘到縣長一層決行，影印公開，更不應該

(7) 總統蔣公聽了，認為這個日本教官太不應該，怎麼可以不尊重別人的國家。

(8) 既得利益就轉移到省籍問題實在太不應該。陳水扁則以逼退不是本省人排斥

(9) 們真是太可惡了！我覺得小朋友太不應該了，回到操場上，又看到小朋友不

(三)「必須」的否定

「必須」雖不能直接加以否定(「不必須」)，但「不必」在語意上相當於「必須」的反義詞，或可視為助動詞「必須」的否定式。雖然張(1961)和呂(1984)都將「不必」視為副詞，然而根據下面「不必」的四個句法特點，可歸入助動詞一類。

a. 「不必」可出現於句首、句中及句尾

(7)「...今天的情形我完全知道，不必你跟我講。」

(8)他四下看了一看，說：「你總不必侍候那些桌子椅子吧？」

(9) 是小時候救人才會打破水缸，長大後就不必，雙方的辯解語句，各有其巧妙之處。

b. 以下句子「不必」是主要謂語或是單獨回答

(10) 站起來，語氣很溫和地說：幸而，你不必。然後伸手去拿尤蝴用過而摔在洗臉台

(11) 我要提他的善行嗎？謝謝，審判長說：不必。被告要不要提出答辯？本庭現在要

(12) 東尼餘氣未消，憤憤地說：「不必！我打過電話了，旅運公司答應設法，

c. 表比較的程度副詞「更」修飾「不必」

雖然語料中沒有出現「不必」被一般的程度副詞「很」或「非常」的修飾，但有

表比較的程度副詞「更」的修飾。

(13) 鳥籠就在鐵皮附近，我們都熱，小鳥更不必說了，簡直快變成烤小鳥了，我們只好

(14) 不論什麼流派，都不必分家，更不必拘束自己，才能使花藝達到最高境界。

(15) 不僅可使用母國語言，更不必當場付費，日後由受話人按一般費率

d. 「不必」在助動詞之前或後

湯&湯判別助動詞的一個條件是「允許同類或不同類的情態動詞或形容詞連用」，語料顯示「不必」可在助動詞之前或後：

(16) 如果我們能夠謙卑一點，我們就不必一定要說我們都成了上帝

(17) 這三個女的是不是要介紹一下？應該不必了。

(18) 尼奧也認為討論事項可以不必參加，她便又去睡了。

(19) 市民沒有辦法在生活周圍隨時找到可以不必花錢，或不必花太多錢就可以享受的

(20) 對已婚的人來說，單身貴族似乎不必擔心柴米油鹽，不必注意孩子教養

從以上的討論看來，「不必」具有多項助動詞的特徵，雖非典型的助動詞，大致可視為「必須」的否定形式。

2.3 助動詞出現的位置

Li & Thompson (1981) 認為助動詞不能在主語之前，湯&湯(1998)則主張助動詞可出現於句中甚或句首的位置，舉出的例子為「可能/應該」，本節全面檢討助動詞可出現的位置(句首、句中或句尾)，底下分三小節加以討論。

2.3.1 出現於句首的位置

經檢驗語料，「應該」並無出現於句首的例子。而「可能」在主語之前者有 5 例：

(1)的杉原輝雄，對於他的怪異揮桿姿勢，可能一般人不敢苟同

(2)若換成一般的、正常的行文，可能正確率會高些。數字會說話。

(3)若生在太平治世，可能有些人活在鄉下小地方，一輩子也沒想

(4)我只看到那位賣冰老太太的表面，可能她的家庭是父慈子孝、兄友弟恭

(5)我也不知道怎麼回答，這什麼原因，可能前一輩我是中國人，我也不知道。

底下的無主語句不列入：

(6)那時你由胃腸的咕叫聲響動頻率推測，可能已經傍晚了。

其他助動詞沒有出現句首的情形，而「必須」只出現下列一句：

(7)就好像是完成一件工作時的喜悅，必須你親自去工作、去完成，然後自心中

2.3.2 助動詞出現於句中

助動詞出現於句中且後接動詞是最常見的情形，換言之，句中是助動詞最典型的位置，可做為助動詞的一個主要特徵。以「必須」為例，出現 1113 筆的「必須」之後不接動詞者只有下列 6 例：

(1)區域圖書館或社區學校，則 NREN 是必須的，但假如只是建立一個很大的網路

(2)影響可能只是其中的一種，但預防總是必須的，他建議電力公司架設高壓電線

(3)本所提供此項臨床試驗計畫必須的軟、硬體支援，及臨床藥理學研究、

(4)本院現行方法據了解有三種必須的處理方式：第一，先記錄登錄號以便

(5)共分為八期，視選手狀況羅列各種必須的訓練內容。

(6)考試代替教學，未提供學生進行思考所必須的時空條件，致使學子不是逃避排斥

2.3.3 助動詞出現於句尾

Li & Thompson 討論名詞化時所舉的句型:「*他是能的」和分裂句類似，兩者「的」的功能是否相同有待進一步研究，因此本文不列入討論。助動詞出現於句尾有三種情形：(a) 助動詞後的動詞組省略，(b) 助動詞以子句為主語或(c)充當分裂句的信息焦點，本文的討論暫時排除(a)類。一般助動詞極少出現於句尾，即使有也是一、二例而已，如「敢」、「(不)能」只見一例，而「必須」也只有二例：

(1)許多人感到錯愕：幹。連調查都不敢，這羞死人的國民黨！

(2)常見的男性不孕症有：性交不能。尿道下裂。隱睪症。腦下垂體機能低下

(3)區域圖書館或社區學校，則 NREN 是必須的，但假如只是建立一個很大的網路，

(4)影響可能只是其中的一種，但預防總是必須的，他建議電力公司架設高壓電線，

下面以「可以、應該、會」為例來說明助動詞出現於句尾的限制或功能。

若把「可」視為「可以」的省略或變體而列入討論，則「可」當子句主語的謂語的情形恐怕是所有助動詞中最多者，約有八十例：

(5)自開普敦前往一天即可往返，所以住在開普敦即可。

(6)找師長做徵信人時，必須取得他的同意才可，

(7)依舊非得到幾個公立的博物館去不可

(8)或者一條龍，或是雕花的窗櫺等等均可。

(9)因此只要按說明書安裝便可

就「可」而言，須有「即/不/均/才/便」修飾「可」才能成為子句主語的謂語；

而「可以」(出現 11 例)則不一定：

(10) 理論上可以，而實際上不行。(對照句)

(11)明年到日本或是歐洲打球都可以，她只在乎能否增進自己的球技

(12)我只是去看看她，少說幾句也可以，

(13) 我很想告訴施工單位：要挖可以，但是不要隨興亂挖

(14)只要我喜歡，有什麼不可以

(15)光靠聰明是不行的，還要努力用功才可以。

以上是「可以」當句主語的謂語的情形，下面四句則是助動詞後動詞組省略的例子：

(16)孩子們看在眼裡，心想：你們可以，我為什麼不可以？

(17)體驗當中，我們會急切的吶喊：「如果可以，請把這要命的苦杯移掉吧！」

(18)：「你在胡說些什麼啊？不可以就是不可以！法律就是法律。」

再看「應該」的例子，「應該」出現於句尾充當主要述語時(共 7 例)，都是義務用法：

(19)做人不要太過分，趕他們出去已不應該，讓他們在走廊坐坐又有何妨？

(20)同居又太貿然，未做避孕措施真不應該，墮胎當然不得已

另外，「應該」有 12 例[5]出現於分裂句，也都是義務用法：

(21)我總覺得，隨便對人發脾氣，是不應該的。

(22)也是主張民主的社會，多元化是應該的但不應該被分歧、被混淆

(23)父母親照顧你二十年，把醫學念完也是應該的。

2000 筆「會」做為動詞出現於句尾者，只有 3 例，3 例皆是表示懂得之意的一般動詞：

(24) 是做人。如果，連最起碼的做人規矩都不會，圖畫得再好，也沒有用。

(25)並在適當的機會表達，如：「這個我會，我可以試試看。」

(26) 最近一代皮猴了，現在沒人學，將來沒人會，就是去博物館看

2.3.4 小結

就語料所見，各助動詞可出現的位置列表如下：

[5] 平衡語料庫的取樣偶有重複的情形，如例(22)即重複出現，實際上是 11 例。

	句首	句中	句尾
可能	＋	＋	＋
必須	＋	＋	＋
能	—	＋	＋
應該	—	＋	＋
可以	—	＋	＋
必要	—	＋	＋
敢	—	＋	＋
會	—	＋	—
應當	—	＋	—
應	—	＋	—
該	—	＋	—
能夠	—	＋	—
得 de	—	＋	—
得 dei	—	＋	—
必得	—	＋	—
肯	—	＋	—
願意	—	＋	—

就以上的討論看來，助動詞最常見的位置是句中，這一點和情態副詞並無太大區分。下面以「好像、大概、一定、必定、絕對」五詞為例：

	句首	句中	句尾
好像[6]	＋	＋	—
大概	＋[7]	＋	—
一定	—	＋	＋[8]
必定	—	＋	—
絕對	—	＋	—

這五個詞一般歸為情態副詞，出現的位置也以句中為典型。副詞和助動詞若要有區別，恐怕是副詞極少在句尾，不過這一點有待對副詞做較全面的研究才能下定論。

2.4 正反問句

雖然兩文都肯定助動詞可出現於正反問句中，但孫(1996：295)就指出「前人所認定的助動詞的語法特點多半都不具備對內的普遍性」。他以正反問句(V-不-V)為例，趙(1968)所列的助動詞中，有 8 個不能用。此處以平衡語料庫中各助動詞的用例來看，助動詞使用正反問句的頻率不僅很低，而且出現這種句型的助動詞只有下列前 7 個：

---

[6] 曹(1990、1996)將「好像」分析爲認知情態動詞(即本文之助動詞)。

[7] 159 筆資料中只出現一次(「仔細觀察了一會，就令我倒足了味口。大概她認定了我是個冤大頭，便拚命的」)。

[8] 773 筆資料中只有一次出現於句尾(「做更多更好的安排，但是感情世界卻不一定。離婚婦女在婚姻中生活圈本就較窄」)。

	出現次數	頻率%
肯	1/151	0.66
該	2/383	0.52
願意	1/293	0.34
應該	2/873	0.23
可以	3/2000	0.15
會	3/2000[9]	0.15
應	1/1381	0.07
可能	0[10]	0
能	0[11]	0
能夠	0	0
應當	0	0
敢	0	0
得 dei	0	0
得 de	0	0
必須	0	0
必得	0	0
必要	0	0

因例子不多，全部列出如下：

(1)用手去拉她，因為男女授受不親。到底該不該拉呢？

(2)父親：你拿一個吧。孔融：(不知道該不該拿，回頭看看母親)。

(3)不知道大家願不願意也用同樣的練習，去對著鏡子，好好的

(4)我真能為別人著想嗎？那麼，我應不應該挽留凱洛琳呢？

(5)不，正確的說法，應該是凱洛琳應不應該留在這裡？

(6)編織重逢的故事；可不可以用愛把過去都結束，真心攜手肩並肩，

(7)最後才說了：這些貝殼可不可以送給我？不行，我只剩下這些了。

(8)。喂，你不要老看股票那一版，可不可以聽聽我講話？

(9)我會立刻加藥，千萬不可大聲嚷嚷。開刀會不會痛？雅麗問我。

(10)上官赴任，結婚嫁娶真不好意思，不曉得會不會影響你辦桌？。

(11)但她又猜想會不會是因為她知道男友和舊情人碰面，

(12)在進行建交作業時，我們竟會去爭論應不應承認他們？

直覺上，助動詞都應該可以出現於正反問句中，然而檢驗語料的結果卻非如此。可能是正反問句是較口語的句型，但語料庫因口語資料來源較少，口語部份(包

[9] 此處「會不會」只有 3 例是以隨機取樣得到的 2000 筆爲限；若以「會不會」爲關鍵詞搜尋時則有 55 筆資料。

[10] 「可能不可能」只出現一次，卻是名詞用法(見下例 i)，因此不計入：

(i)在狄斯耐的卡通世界，沒有什麼可能不可能，音樂當然也是。

[11] 在平衡語料庫選中的 2000 個「能」字中，雖無正反問句，但未選中者出現了一次，如下列句子的第二個「能」字:「對自己有了了解之後，當然希望別人能了解我，但別人能不能了解呢？」。

括演講稿、劇場台詞、會話及會議記錄)僅佔全部的十分之一[12]，待將來語料庫達到口語與書面語的質和量相當後，或可瞭解正反問句是否為助動詞的一個基本特色。若就目前語料看來，我們無法把正反問句做為判斷助動詞的一個重要特徵。

2.5 理論與語料的不一致

學者的理論或假設是試圖闡述說話者的語言的本能(competence)，而語料則反映出語言的運用(performance)，同時可驗證學者的理論是否成立。從以上的討論看來，程度副詞修飾比正反問句更應視為助動詞的語法特點之一，而 Li & Tompson 主觀的將程度副詞修飾排除在外，恐怕不能反映出真正的語言本能。再者就前四小節討論的助動詞的語法特點中，若不考慮助動詞出現的位置，以其餘三個條件來檢驗 17 個助動詞在語料中的使用情形，列出簡表如下：

	程度副詞修飾	否定式	正反問句
會	＋	＋	＋
應該	＋	＋	＋
應	＋	＋	＋
可以	＋	＋	＋
願意	＋	＋	＋
可能	＋	＋	—
能	＋	＋	—
應當	＋	＋	—
敢	＋	＋	—
必須	＋	＋	—
該	—	＋	＋
肯	—	＋	＋
能夠	＋	—	—
得 dei	＋	—	—
得 de	—	＋	—
必要	—	＋	—
必得	—	—	—

就上表來看，17 個助動詞中只有「會、應該、應、可以、願意」五個詞完全滿足助動詞的這三個特點，其他則或多或少具備其中部份的特點，而「必得」則完全不符合任何一點，所以「必得」應排除於助動詞之外。另外，雖然「必要」可以被「不」否定，在檢驗語料的過程中卻發現，「必要」用為主賓語(名詞)或定語(修飾名詞)者達 88.41%[13]，這在其他助動詞中十分少見，因此把「必要」分析為情態名詞比助動詞合適。

此外，我們在整理語料的過程中發現，有些詞雖被列為助動詞，然而在語言使

[12] 根據詞庫小組(1998：12)，口語部份包括演講稿 1.38%、劇場台詞 0.82%、會話 7.29%及會議記錄 0.36%，共 9.85%。

[13] 詳細討論請參鄭(2000)。

用的頻率上，非情態用法高於情態用法者。如「應該」、「應當」、「應」或「該」都可表示情理上必須如此，或估計必然如此，甚至有些詞還有其他非情態用法。然而在實際語言使用中，這些詞出現的次數並不相同，各種用法的頻率不同，平衡語料庫同樣隨機取樣時，各詞做為情態詞用法的比例差距頗大。

註解:

「應該」、「應當」、「應」和「該」各種詞類的比例分別如下所示(“X(Y)”中X表示出現的次數，Y表示所佔的百分比)[14]：

	應當	應該	應	該
助動詞	31(100)	842(96.45)	1381(94.52)	383(19.15)
動詞		31(3.55)	36(2.47)	8(0.40)
形容詞[15]				1609(80.45)
介詞[16]			44(3.01)	

「應該」、「應」、「該」和「應當」四者在語料庫中做為助動詞的頻率差別頗大。以「該」為例，語料中有八成是做為指示詞來修飾名詞，可考慮把此詞排除於助動詞之外。由此觀之，以語料中所見助動詞的非情態用法為例可用來說明，語言的使用(此處以語料為代表)和主觀印象往往有一段落差。

孫德金(1997:295)指出「通用語料中的助動詞必定是非同質性的，有口語的，有書面語的，有文言的」，就我們的語料來看，雖然「應該」、「應當」、「應」和「該」做為助動詞時，往往被視為同義詞，就我們的語感而言，「應」似乎是較為文言的詞彙，而「應該」或「該」應屬口語的詞彙。理論上在日常生活中，後者出現的頻率應高於前者，然而語料卻告訴我們，「應」的使用次數反而比「應該」或「該」高得多，這可能是目前平衡語料庫的語料來源以書面語為主(達百分之九十以上)所致。目前受限於語料庫，無法得知真相；待將來語料庫能改進到口語與書面語質和量相當後，可進一步探討語式和助動詞使用的關係。

3.結語

理論或假設闡述的是說話者的語言本能，而語料反映的是語言的運用，綜合以上討論可知，以往學者研究語言現象時多以個人語感為準，造成兩者之間往往有落差。以能否被程度副詞(很/更)修飾這個特點為例，對 Li & Thompson 而言是決定助動詞的一個條件，湯&湯則以之區分情態動詞和形容詞，結果語料顯示多數助動詞是可與「很」或「更」搭配，然而助動詞搭配程度副詞時，可能有限制，如「會」可表能力或預斷，語料中前者主要搭配「很」而後者只與「更」共現；「應該」則兼有義務和預斷用法，但只見義務用法和程度副詞一起出現。助動詞的否定式如「不應該」可以被程度副詞修飾，這一點和心理動詞（如「很不喜歡」）或形容詞(如「很不高興」)相同。雖然學者都指出助動詞可出現於正反問句中，

[14] 此處詞類的判定請參考中文詞知識庫小組 (1993)，他們將情態助動詞歸於副詞，此處爲了與其他副詞區隔，仍稱爲「助動詞」；統計結果則是平衡語料庫所提供的。
[15] 此處形容詞指的是做爲名詞修飾語，而非句子的謂語。
[16] 包括連接詞。

但以平衡語料庫中各助動詞的用例來看，助動詞使用正反問句的頻率不僅很低，而且出現這種句型的助動詞只有「肯、該、願意、應該、可以、會、應」7個。再就助動詞的位置而言，經全面檢討助動詞可出現的位置(句首、句中甚或句尾)，發現(a)「可能」出現在句首者有五例，「必須」只有一例，其他助動詞則無；(b)其次，助動詞出現於句尾時有其限制或功能：如「可」必須伴隨「均/才」等副詞才能出現於句尾，「可以」則無此限制；句尾可解決助動詞的歧義，如「應該」只有表情理上必須如此時(即義務用法)，才出現於句尾，推測義的「應該」則不可。同樣的，表能力的動詞「會」可出現於句尾，表測義者不可。以上的結果顯示，語言學家若忽略語料呈現的語言事實，對語法特點的描述流於主觀，且不符合說話者的語言本能。

參考書目：

朱德熙 1982 語法講義 商務印書館。

呂叔湘 1984 漢語八百詞 商務印書館。

孫德金 1997 漢語助動詞的範圍，詞類問題考察 p.286-307。

詞庫小組 1993 中文詞類分析(三版)，中央研究院資訊科學所，詞庫小組，台北南港。

詞庫小組 1998 中央研究院平衡語料庫的內容與說明(修訂版) ，中央研究院資訊科學所，詞庫小組，台北南港。

張靜 1961 論漢語副詞的範圍 中國語文 8 月號 1-14。

湯廷池 1984 國語的助動詞 中國語文 22-28。

湯廷池&湯志真 1998 華語情態詞序論 第五屆華語文教學研討會論文集 177-197 頁。

鄭　縈 1999 台灣國語、閩南語和客家話各類情態詞搭配關係的比較，國科會計畫成果報告(NSC 88-2411-H-126-005)。

鄭縈 2000 漢語情態動詞的詞序, 第九屆國際漢語語言學會議(IACL-9), 新加坡。

曹逢甫 1996 漢語的提升動詞，中國語文，172-182 頁。

Charles N. Li and Sandra A. Thompson 1981 Mandarin Chinese :a functional reference grammar, Berkeley :University of California Press。

Tsao, F-F(曹逢甫) 1990 Sentence and clause structure in Chinese, Taipei: Student Book Co.。

Part-of-speech Sequences and Distribution in a Learner Corpus of English

Rebecca H. Shih[*], John Y. Chiang[+] and F. Tien[+]

[*]Department of Foreign Languages and Literature

[+]Department of Computer Science and Engineering,

National Sun Yat-sen University

Page 171 ~ 177

Proceedings of Research on Computational Linguistics Conference XIII (ROCLING XIII)

Taipei, Taiwan

2000-08-24/2000-08-25

Part-of-speech Sequences and Distribution in a Learner Corpus of English

Rebecca H. Shih[*], John Y. Chiang[+] and F. Tien[+]

[*]Department of Foreign Languages and Literature

[+]Department of Computer Science and Engineering

National Sun Yat-sen University, Kaohsiung,Taiwan, R.O.C.

E-mail: hsuehueh@mail.nsysu.edu.tw

Abstract

Computer learner corpora have been widely used by SLA/EFL specialists since mid 1990s to gain better insights into authentic learner language. The work presented in this paper examines the inter-language of Taiwanese learners of English from a part-of-speech sequence perspective. Two pre-tagged corpora (one learner corpus and one native corpus) are involved in this work. The experimental results indicate that there are more than one third of eligible POS trigrams that are never practiced by the Taiwanese learners in their writing and the learners have stronger preference than native speakers in using pronouns, especially right after punctuations, verbs and conjunctions.

1. Introduction

With the recognition of its theoretical and practical potential, computer learner corpora (CLC) have been subsequently built up around the world since early 1990s.[1] CLC research aims to gain a better insight into learners' inter-language from the authentic data. The research often involves comparisons between inter-language that learners possess and native language on various linguistic features. For instance, the frequency distributions of most commonly-used words in a native and seven eastern European learner corpora are compared on various parts-of-speech categories[2]; the use of complement clauses in terms of their frequencies in four learner corpora as contrasted with their native counterparts [3] is studied; the use of adverbial connectors by Swedish learners in comparison with the natives' is examined [4]. The quantitative information as such often guides the researchers to carry out insightful qualitative analysis. And this kind of cross-language approach helps SLA and EFL specialists find out what linguistic features the language learners are apt to overuse/underuse,

what particular areas of language behavior that are shared by learners with different backgrounds, and to what extent these phenomena appear in learner English.

The aim of the work in this paper is to discover distinctive inter-language features of Taiwanese learners of English in terms of part-of-speech sequences and distribution. It is based on two corpora: Taiwanese Learner corpus of English (TLCE) and British National Corpus (BNC). Both corpora are tagged by TOSCA tagger, using the TOSCA-ICLE tagset. The details of the corpora and the tagger will be stated subsequently in Section 2, which is followed by a series of experiments in Section 3. Conclusions are drawn in Section 4 with future work.

2. Methodology

2.1 Corpora: TLCE and BNC

As stated in the introduction, CLC-research often compares non-native data with native data in order to reveal the overuse and/or underuse phenomena in a learner corpus. In this work, the Taiwanese Learner Corpus of English (TLCE) is under investigation and the British National Corpus (BNC) is used for comparison. TLCE of 455,000 words is a growing corpus of English compositions and weekly journals written mainly by college English majors(freshmen, sophomores and juniors) from Sun Yat-sen and Chi-nan universities in Taiwan. The BNC contains modern British English and is a unique collaboration between three major U.K. dictionary publishers, two universities, and the British Library [5]. The work here utilizes mainly its subset of 1 million words (from BNC Sampler written text).

2.2 Tagger: TOSCA

The corpora are lemmatized and part-of-speech tagged with the TOSCA tagger [6]. TOSCA is a stochastic tagger, supplemented with a rule-based component which tries to correct observed systematic errors of the statistical components. TOSCA also gives each word form its lemma (basic form). For instance, word forms such as *takes*, *took*, *taken*, and *taking* have the same lemma *take*. This function facilitates the collocation analysis under the same lemma. TOSCA operates with a lexicon, which currently contains about 160,000 lemma-tag pairs, covering about 90,000 lemmas. The TOSCA-ICLE tagset contains 270 different tags within 18 major word classes. For simplicity, only the major word classes are considered in the current study (see Appendix A)

3. Experiments and Results

3.1 Corpus Perplexity in Bigram and Trigram models

Perplexity, in speech recognition community, is often referred to as the number of equi-probable choices at each step of word prediction in a language model such as a bigram/trigram model under the assumption that a word depends merely on the previous one/two words. In this work, given a corpus *L*, the perplexity of the corpus, *S(L)*, can be viewed as a measure of diversity for the next POS in a language model, and it is defined as:

$$S(L) = 2^{H(L)}$$

$$H(L) = \frac{1}{N}\sum_{c} H_c(k)$$

$$H_c(k) = -\sum_{k} P(k \mid c)\log_2 P(k \mid c)$$

where *H(L)* is the entropy of the corpus *L*, *N* is the size of part-of-speech set, and *P(k|c)* is the probability that *k* will be the next POS when the current POS is *c*.

.

In this experiment, the perplexities of BNC and TLCE corpora are calculated using both bigram and trigram models, and the results are shown in Table 1:

	S(BNC)	S(TLCE)
Bigram	4.91	4.36
Trigram	3.01	2.15

Table 1: Corpus perplexity

As can be seen in Table 1, the perplexities of BNC corpus in the two language models are both greater than those of TLCE, especially in the trigram model where the degree of POS diversity in the learner corpus is only 2/3 of BNC's. The above phenomena can be explained by the limiting sentence structure varieties the learners possess.

3.2 Structure Variety

In order to further understand the limit of structure variety in learners' writing, the numbers of POS trigrams, i.e. sequences of three POSs, used in the two corpora are compared and shown in Table 2. As seen in the table, there are 2531 trigram patterns in BNC, 1649 in TLCE, and 1574 in both. If those appearing in BNC can be viewed as the only eligible patterns for English, then the learners merely use 62% of correct trigram structures in their writing, and leave 38% in tact.

BNC	TLCE	overlap
2531	1649	1574

Table 2: the number of POS trigrams in the corpora

Under the same assumption, Figure 1 depicts the divergence of learners' use of trigrams from BNC, the optimum indicated by the square curve, on the scale of top-ranking trigrams in use. The diamond curve denotes the number of the learners' trigrams that overlap with BNC at the same rank. As illustrated, the learners' curve moves away from the optimum when the scope of the rank enlarges, especially after the rank of 1000.

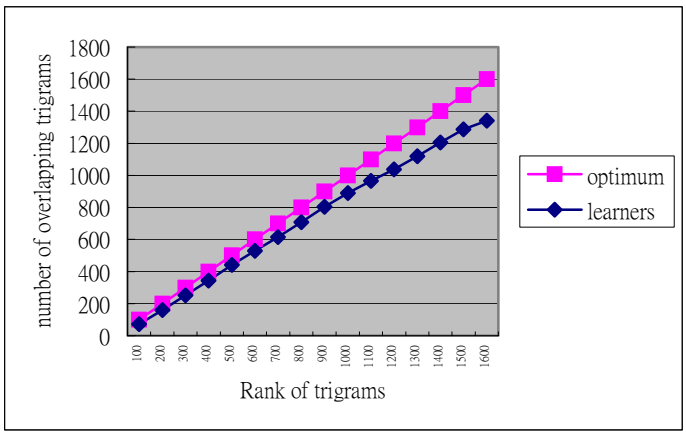


Figure 1: The divergence of the use of POS trigrams

3.3 POS Distribution

As the learners have preference in using certain POS trigrams it is then desirable to understand the learners' preference in using POSs themselves as well. Figure 2 shows the POS distribution in each corpus, and only those taking up at east 5% of the corpus are indicated. Two significant phenomena are observed from the figure. Firstly, although N(Noun) and VB(Verb) are the first two leading POSs in both corpora, there exists a distinct discrepancy of the percentage difference between the two. The difference in distribution percentage between N and VB in BNC reaches 9%, whereas merely 1% difference in TLCE. Secondly, PRON(pronoun), the 3rd highest distribution in the learner corpus but the 7th in BNC, apparently is overused the learners.

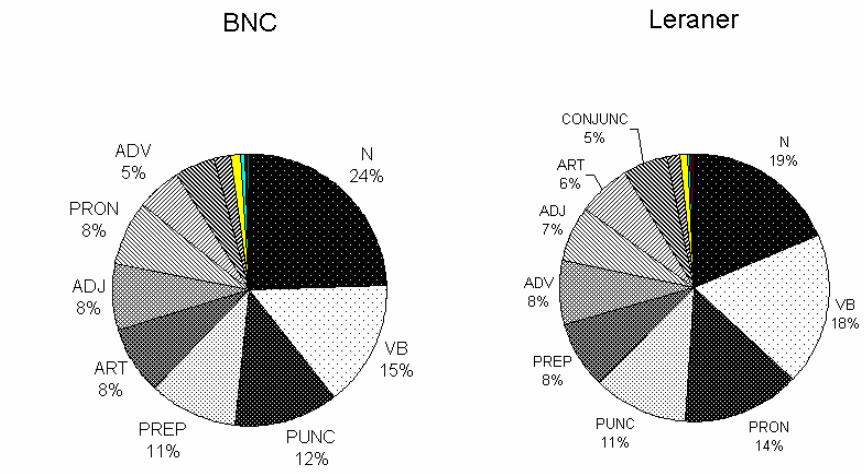


Figure 2: POS distribution

3.4 Distribution of Preceding POSs in PRON bigrams

As the previous figure indicates the excessive use of PRON in the learner corpus, the phenomenon is further analyzed by examining the likelihood of each POS preceding PRON in the bigrams. Figure 3 shows the distribution of preceding POSs of PRON in each corpus. As seen, PUNC(punctuation), VB and CONJ(conjuction) are the three most likely POSs in TLCE to be followed by PRON, and the learners also have stronger preference in using these bigrams than the native speakers. By contrast, the bigrams, PREP(preposition)+PRON and N+PRON, are used more frequently by the native speakers than the learners.

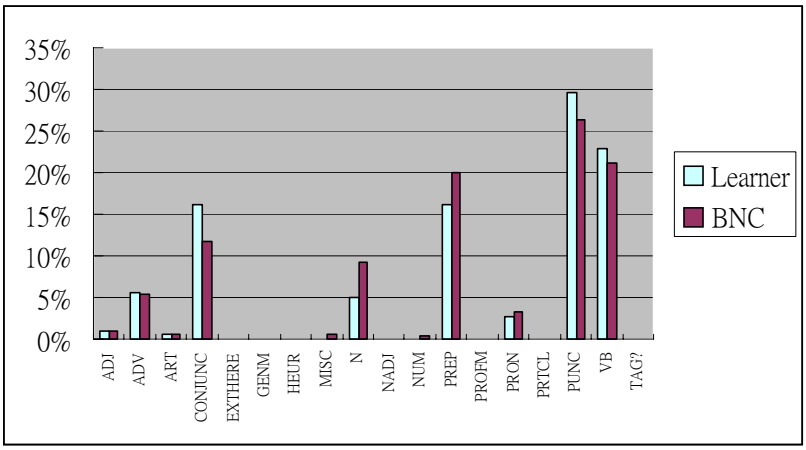


Figure 3: Distribution of Preceding POSs in PRON bigrams

4. Discussions and future work

The results of the preliminary experiments above show that there are more than one third of BNC trigrams that the learners never practice in their writing, whereas there are 4.5% of TLCE trigrams which do not appear in the BNC's. It is intended to believe that this small proportion of TLCE trigrams is contributed from the learner's writing errors. However, increasing the size of the native speaker corpus to observe any changes in the distribution of the trigrams will clarify the findings. It is also worth looking into those BNC trigrams that the learners do not know or are not aware of, and then isolating those with high frequency for the pedagogical purpose.

The experimental results also suggest that the learners use pronouns excessively in their writing and that they have stronger preference than native speakers in using pronouns right after punctuations, verbs and conjunctions but less preference after prepositions and nouns. Pronouns often appear in the informal register, and as the corpus is composed of college students' compositions as well as their weekly journals, the informality of the journals may contribute partly to their excessive use of pronouns. So, it is desirable in the next stage of the work to divide the learner corpus in terms of its different registers and compare their POS distributions with the native speaker corpus.

Acknowledgements

The authors would like to thank the National Science Council, Taiwan, for supporting this project, and Prof. Ching-Yuan Tsai for his insightful comment.

Appendix A

Label	Major word class
ADJ	Adjective
ADV	Adverb
ART	Article
CONJUNC	Conjection
EXTHERE	Existential there
GENM	Genitive marker
HEUR	(unknown)
MISC	Miscellaneous
N	Noun
NADJ	Nominal adjective
NUM	Numeral
PREP	Preposition
PROFM	Proform
PRON	Pronoun
PRTCL	Particle
PUNC	Punctuation
TAG?	Word unable to tag
VB	verb

References

1. Granger, S., *The International Corpus of Learner English*, in *English Language Corpora: Design, Analysis and Exploitation*, J. Aarts, P.d. Haan, and N. Oostdijk, Editors. 1993, Rodopi: Amsterdam. p. 57-69.
2. Lorenz, G., *Overstatement in advanced learners' writing: stylistic aspects of adjective intensification*, in *Learner English on Computer*, S. Granger, Editor. 1998, Addison Wesley Longman Limited. p. 53-66.
3. Biber, D. and R. Reppen, *Comparing native and learner perspectives on English grammar: a study of complement clauses*, in *Learner English on Computer*, S. Granger, Editor. 1998, Addison Wesley Longman Limited. p. 145-158.
4. Tapper, M., *The use of adverbial connectors in advanced Swedish learners' written English*, in *Learner English on Computer*, S. Granger, Editor. 1998, Addison Wesley Longman Limited. p. 80-93.
5. Aston, G. and L. Burnard, *The BNC Handbook*. 1998: Edinburgh University Press.
6. Aarts, J., H. Barkema, and N. Oostdijk, *The TOSCA-ICLE Tagset Software and Tagging Manual*, . 1997, The Department of Language and Speech, University of Nijmegen, The Netherlands.

具有累進學習能力之貝氏預測法則在汽車語音辨識之應用

簡仁宗，廖國鴻

國立成功大學資訊工程學系

Page 179 ~ 197

Proceedings of Research on Computational Linguistics Conference XIII (ROCLING XIII)

Taipei, Taiwan

2000-08-24/2000-08-25

具有累進學習能力之貝氏預測法則在汽車語音辨識之應用

Bayesian Predictive Classification with Incremental Learning Capability for Car Speech Recognition

簡仁宗　廖國鴻

國立成功大學資訊工程學系

Email : jtchien@mail.ncku.edu.tw

摘要

在許多的語音辨識應用中，由於週遭環境噪音的影響，使得測試語句與原始語音隱藏式馬可夫模組產生不匹配，導致辨識率明顯的下降。因此，本論文提出以轉換為主之貝氏預測分類器來提昇雜訊語音的辨識率。我們將隱藏式馬可夫模組的平均值向量作轉換並結合貝氏預測分類法則理論，把轉換參數的不確定性導入辨識決策中，其中轉換參數的不確定性用一常態之事前機率密度來表示，因此，我們可以發展出一套以轉換為主的貝氏預測分類器做語音辨識之強健性決策法則。在本論文中另一個重要的特徵，即事前機率累進學習的能力，藉由累進觀測到之測試語料，我們可以連續的更新事前機率密度的統計量，此更新過的統計量可以追蹤到最新環境的統計特性，事前機率之最新統計量將使用在以轉換為主的貝氏預測分類器法則，一個具有強健性決策且能夠累進學習特性的辨識系統即可建立。我們收集了九十公里、五十公里及零公里(引擎開)三種汽車路況的語料，經由實驗的結果，我們的方法在各種路況下辨識率均有明顯的改善。

1. 簡介

在過去這十幾年來，自動語音辨識技術日新月異，進步的相當迅速，而這樣的進步，觸發了語音辨識技術實際應用的動機。例如：語音辨識應用在汽車行動電話之免持撥號系統，將使得行動電話的使用更具人性化，且讓駕駛者多了一份安全保障。可是對於語音辨識的實際應用，常常會遇到不匹配問題，因為語音辨識主要是以樣本比對的技術為基礎，若是語音辨識之應用環境與原始樣本之訓練環境不匹配，將會使得辨識率大幅地降低。而這樣的不匹配可能是來自於週遭的環境噪音、傳輸語音的通道不同、語者不同或訓練樣本模組時模組的不正確等等，而在實際的應用中，影響語音辨識的因素往往是上述多個失真來源的組合。因

此，為了克服語音辨識時不匹配之問題，提出一強健且有效率的補償技術是必要的。

許多學者也提出了許多的方法來解決訓練與測試環境不匹配的問題，一般來說，不匹配的問題可分別在訊號、特徵或模組空間來解決[17]，在此論文中，探討方向著重在模組空間的補償，一般模組補償的方法可分成兩大類：

一、**非參數形式之補償：**此類方法對模組做補償時並不做任何的補償函數形式的假設，例如，Minimax 分類法[14]之主要觀念，假設測試語料的最佳決策參數，會落在所給定模組參數之限定的鄰近範圍內，然後對其決策規則和所對應的參數做調整，以解決不匹配問題。此方法之優點可以克服最差不匹配的情況，但是當鄰近的範圍重疊時，字與字之間的混淆度增加，此方法便無法發揮效用，而且在連續語音辨識方面的應用，此方法將有困難。

二、**參數形式之補償：**此方法先假設一補償函數之形式，然後估測此補償函數中的參數，例如 MLLR（Maximum Likelihood Linear Regression）方法[13]，其假設訓練與測試環境的不匹配可用一線性轉換函數來補償，把原本訓練好之模組參數的平均值部分，做一線性轉換，使得轉換過的參數能較匹配於測試環境，而此轉換參數的估測，可用最佳相似度（Maximum Likelihood, ML）之演算法[7]來估測，然後將估測到的轉換參數嵌入（plug in）辨識的決策中。這樣的方式是假設所估測到的轉換參數是一個能夠代表訓練與測試環境不匹配的正確值，此方法之缺點，在辨識時存在著轉換參數估測錯誤的風險，不能夠很可靠地提昇辨識率。另外一種常見的參數形式之補償是 MAP（Maximum *A Posteriori*）調整方法[6]，其主要是應用最佳事後機率的法則來做調整，利用 EM 演算法[7]來估測參數，此方法也是存在著“參數估測錯誤”的風險，且由於在實際的語音辨識應用中，我們所能獲得的只是訓練好的模組參數與測試語料，而 ML 演算法需要大量的語料才能估測出可靠的參數值，所以此方法亦不適用於線上調整策略

由於在實際的語音辨識應用中，訓練與測試語料之間的不匹配是未知的，所以我們利用 Minimax 分類演算法的優點，把參數的不確定性引入辨識的決策中，在本研究中，我們採用貝氏預測分類器（Bayesian Predictive Classification, BPC）[8][16]並結合轉換為主的調整技術[13]，建立一以轉換為主之貝氏預測分類（Transformation-Based BPC, TBPC）強健決策法則，這裡我們是把轉換參數的不確定性引入辨識的決策中，如此可避免掉 ML 演算法點估測的估測錯誤風險。此外，由於在辨識的應用中，測試環境的統計特性是非固定的（Nonstationary），會隨著時間而改變，因此，我們把 TBPC 和事前機率線上累進學習（Online Prior Evolution, OPE）的策略結合在一起，建立一強健且具學習能力之辨識系統 TBPC-OPE。為了要讓辨識

系統具有累進學習的能力，我們採用近似最佳事後機率演算法[3][10]，此演算法，可提供在 TBPC 決策中轉換參數事前統計量之累進學習機制，從累進觀察到的測試語料中進行累進式學習，使辨識器不斷地追蹤到最新環境的統計特性。所以，我們所提出的辨識系統 TBPC-OPE 不僅具有強健性的決策 TBPC 而且具有線上累進學習的能力 OPE。

2. 以轉換為主之貝氏預測分類法則

為了要加強語音辨識的效能，發展一套以隱藏式馬可夫模組（Hidden Markov Model, HMM）為主的強健性決策法則是必要的。在本論文中我們要把強健統計決策的貝氏預測分類法則和 HMM 參數的線上調整技術結合在一起。因此，我們提出轉換為主的貝氏預測分類法則，以處理對於 HMM 平均值向量之轉換參數的不確定性，而在轉換參數之事前機率的部分，採用線上累進的策略來追蹤最新環境的統計量，如此，具有強健性決策且具學習能力之辨識系統即可建立。

2.1 近似最佳事後機率估測（Approximate MAP Estimation）

在一個人機互動系統中，測試語者的語音資料通常是充裕且以累進方式觀察到，而我們要以這些累進收集到的測試或調整語料來調整已訓練好的非特定語者 HMM 參數以適應測試環境的特性。根據線上轉換演算法[3]，我們可以利用累進收集到的語料來估測轉換參數，然後用此轉換參數來補償測試環境所產生的聲學變化。使用近似最佳事後機率估測[3][10]，我們可以得到

$$(W^{(n)},\eta^{(n)}) = \arg\max_{(W,\eta)} p(W,\eta|\chi^n) = \arg\max_{(W,\eta)} p(\mathbf{X}_n|W,\eta)\cdot p(W,\eta|\chi^{n-1})$$

$$\cong \arg\max_{(W,\eta)} p(\mathbf{X}_n|W,\eta)\cdot p(W,\eta|\varphi^{(n-1)}), \tag{1}$$

在（1）式中，先前累積之測試語料的事後機率密度函數 $p(W,\eta|\chi^{n-1})$ 用一最相近的事前機率密度函數 $p(W,\eta|\varphi^{(n-1)})$ 來取代，其中 $\varphi^{(n-1)}$ 會根據累積觀察到的測試語料 χ^{n-1} 不斷的更新。在本質上，W 和 η 是互相獨立的，因此，（1）式的估測可分解下列兩個步驟

$$W^{(n)} = \arg\max_{W} p(\mathbf{X}_n|W,\eta)\cdot p(W), \tag{2}$$

$$\eta^{(n)} = \arg\max_{\eta} p(\mathbf{X}_n|W^{(n)},\eta)\cdot p(\eta|\varphi^{(n-1)}), \tag{3}$$

其中 $\chi^n = \{\mathbf{X}_1,\mathbf{X}_2,\cdots,\mathbf{X}_n\}$ 是累進觀測到之測試語料，這裡假設每個語音音框互相統計獨立且

機率分佈相同，n 是測試語料的索引，$W^{(n)}$ 是第 n 句未知語料 $\mathbf{X_n}$ 的內容或字串列，$\eta^{(n)}$ 是第 n 句語料對應的轉換參數，$\varphi^{(n-1)}$ 是由前 n-1 句語料 χ^{n-1} 累進得到的事前機率參數（代表環境之統計量），$p(W)$ 是字串列的事前機率，也就是語言模型，$p(\eta|\varphi^{(n-1)})$ 是轉換參數 η 的事前機率密度函數。近似最佳事後機率估測法中的第一個步驟即執行（2）式，估測出對於現在語料 $\mathbf{X}_n$ 之最可能的語料內容 $W^{(n)}$，在這個步驟中，轉換參數 η 假設已被萃取出且被嵌入最佳事後機率解碼器（MAP decoder）的辨識系統中。在得到最可能的語料內容 $W^{(n)}$ 之後，第二個步驟，利用（3）式估測出最佳的轉換參數 $\eta^{(n)}$。基於此最佳事後機率估測演算法，當給定事前機率初始的統計量 $\varphi^{(0)}$，我們可以藉由測試語料 $\mathbf{X}_1$ 和其所對應最可能之語料內容 $W^{(1)}$ 代入式子（3），估測出轉換參數 $\eta^{(1)}$，同時 $\varphi^{(0)}$ 會更新變為 $\varphi^{(1)}$，此更新過的事前機率統計量 $\varphi^{(1)}$ 可用在下一次參數 $(W^{(2)},\eta^{(2)})$ 的估測。以此類推，我們可累進地估測出參數 $(W^{(1)},W^{(2)},\cdots,W^{(n)})$ 和 $(\eta^{(1)},\eta^{(2)},\cdots,\eta^{(n)})$，而此遞迴的演算法可使我們能夠累進地去追蹤最新環境的統計特性。因此可知，在本論文中辨識系統所採用的學習策略是*累進*而且是*非監督式（unsupervised）*的。

2.2 貝氏預測分類的應用

在傳統的辨識調整方法中，轉換參數 η 的估測可用最佳相似度法[13][17]、最佳事後機率法[1][6]或近似貝氏[3][10]估測法得到，而此估測到的轉換參數再嵌入最佳事後機率的辨識器中，然後辨識出最可能的語料內容。這樣的調整方法是在辨識的過程中，把點估測（Point Estimate）之轉換參數假裝為一個正確的值，而嵌入其聲學模組相似度的計算中，因此，隱含了點估測錯誤的風險，造成辨識的錯誤。所以，在本論文中提出轉換為主之貝氏預測分類法則，其主要的觀念是把轉換參數視為隨機變數並將其機率分佈考慮在聲學模組相似度的計算中，將轉換參數的不確定性平均起來取代點估測的方式，如此，可避免點估測錯誤的風險。在本研究中，我們用 $\tilde{p}_\eta(\mathbf{X}_n|W)$ 來取代（2）式中的相似度 $p(\mathbf{X}_n|W,\eta)$，（2）式可改寫成

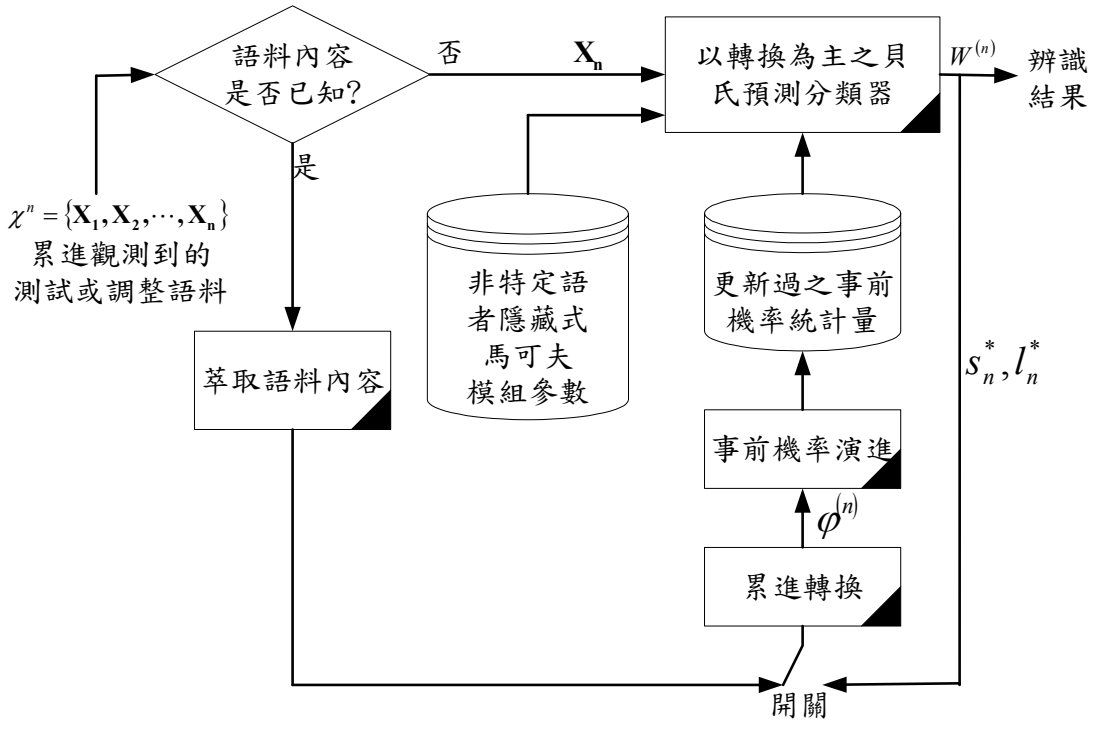
$$W^{(n)} = \arg\max_W \tilde{p}_\eta(\mathbf{X}_n|W)\cdot p(W), \tag{4}$$

其中相似度 $\tilde{p}_\eta(\mathbf{X}_n|W)$ 是由下列的積分得到

$$\tilde{p}_\eta(\mathbf{X}_n|W) = \int p(\mathbf{X}_n|W,\eta)p(\eta|\varphi^{(n-1)},W)d\eta.$$

$$= E\left\{p(\mathbf{X}_n|W,\eta)|\varphi^{(n-1)}\right\} \tag{5}$$

觀察第（5）式，我們可知以轉換為主的貝氏預測分類法是把轉換參數的不確定性引入辨識時相似度的計算，此預測機率分佈（Predictive Distribution）函式 $\tilde{p}_{\eta}(\mathbf{X}_n|W)$ 為轉換參數分佈中相似度函式 $p(\mathbf{X}_n|W,\eta)$ 的期望值，因此 TBPC 可視為是把轉換參數的不確定性平均化，這是和 ML 點估測不一樣的地方。所以，很明顯地，我們所提出的演算法其好處在於將轉換參數 η 的事前機率同時應用在第（4）和（5）式 TBPC 的辨識決策中而且也應用第（3）式累進轉換的技術裡，使事前機率具有累進學習的能力。因此，基於近似最佳事後機率估測法（3）（4），我們可以建立具有強健決策與累進學習能力的辨識系統，整個系統的策略如圖一所示。輸入端是累進觀測到的測試或調整語料 $\chi^n=\{\mathbf{X_1},\mathbf{X_2},\cdots,\mathbf{X_n}\}$ ，若輸入語料 $\mathbf{X_n}$ 的內容未知則為非監督式的學習，走“否”路線，此時亦需要 SI HMM 參數和上一次更新過之事前機率統計量，然後輸入以轉換為主之貝氏預測分類器，辨識出結果 $W^{(n)}$ 並得到最佳的狀態和混合數序列 s_n^*,l_n^* ，利用 s_n^*,l_n^* 的資訊，經過累進轉換處理可得到一組更新過的轉換參數事前機率統計量 $\varphi^{(n)}$ ，此更新過的統計量將用在下一句語料的 TBPC 決策，如此循環以達到累進式的學習，同理，對於監督式的學習（“是”路線），因為已經有相對應的語料內容，所以和非監督式的差別在於能夠提供較正確的資訊供事前機率統計量的累進，不過在一般較實用的語音辨識應用中，環境的學習均為非監督式。



圖一、TBPC-OPE 之辨識系統架構圖

3. 貝氏預測相似度量測

在上個章節中討論到以轉換為主之貝氏預測分類法則的應用，在這個章節我們將詳細地解說此強健性的決策法則如何推導出來。

3.1 以轉換為主之調整（Transformation-based Adaptation）

考慮一具有 L 個狀態 K 個混合數的隱藏式馬可夫模組 $\lambda=\{\lambda_i\}=\{\omega_{ik},\mu_{ik},r_{ik}\}$，$i=1,\cdots,L$，$k=1,\cdots,K$，我們定義對於語料 $\mathbf{X}_n$ 中觀測音框 $\mathbf{x}_t^{(n)}$ 之狀態觀測機率為一多變數高斯（Gaussian）機率密度函式

$$p(\mathrm{x}_t^{(n)}|\lambda_i)=\sum_{k=1}^{K}\omega_{ik}N(\mathrm{x}_t^{(n)}|\mu_{ik},r_{ik})$$

$$=\sum_{k=1}^{K}\omega_{ik}(2\pi)^{-d/2}|r_{ik}|^{1/2}\exp\left[-\frac{1}{2}(\mathbf{x}_t^{(n)}-\mu_{ik})^T r_{ik}(\mathbf{x}_t^{(n)}-\mu_{ik})\right] \tag{6}$$

其中 ω_{ik} 為混合數之權重且 $\sum_{k=1}^{K}\omega_{ik}=1$，而 $N(\mathbf{x}_t^{(n)}|\mu_{ik},r_{ik})$ 是具有 d 維度之平均值向量 μ_{ik} 和 $d\times d$ 精確矩陣（變異數矩陣之反矩陣）的高斯機率密度函式。當以轉換為主的調整被應用來克服環境的不匹配時，我們將會使用一些特定環境的語料 χ^n 估測出參數為 $\eta^{(n)}=\{\eta_c^{(n)}\}$ 的轉換函式 $G_{\eta^{(n)}}(\cdot)$，將分群過之隱藏式馬可夫模組參數做調整，使得調整過的模組參數能適用在新的辨識環境。在本論文中，我們考慮隱藏式馬可夫模組之平均值向量，加上一偏差向量 $\mu_c^{(n)}$ 以達到調整的目的，因此，轉換函式將被定義為

$$\lambda^{(n)}=G_{\eta^{(n)}}(\lambda)=\{\omega_{ik},\mu_{ik}+\mu_c^{(n)},r_{ik}\} \tag{7}$$

其中 c 代表轉換群組的索引，而具有索引 i 和 k 的隱藏式馬可夫參數假設是屬於第 c 轉換群組，亦即 $\lambda_{ik}\in\Omega_c$。因此，當給予隱藏式馬可夫模組參數 λ_{ik} 和轉換參數 $\eta_c^{(n)}=\mu_c^{(n)}$，則 $\mathbf{x}_t^{(n)}$ 的相似度函式表示為

$$p(\mathbf{x}_t^{(n)}|\lambda_{ik},\eta_c^{(n)})\propto|r_{ik}|^{1/2}\exp\left[-\frac{1}{2}(\mathbf{x}_t^{(n)}-\mu_{ik}-\mu_c^{(n)})^T r_{ik}(\mathbf{x}_t^{(n)}-\mu_{ik}-\mu_c^{(n)})\right]. \tag{8}$$

此外，在我們提出的架構中另一個重要的課題是事前機率密度函式的選擇。基於統計學上機率密度分佈之共軛特性（Conjugate Prior）[4]和式子（5）的計算考量上，我們選擇一多變數高斯機率密度函式來當轉換參數的事前機率密度函式，定義如下

$$p(\eta_c^{(n)}|\varphi_c^{(n-1)})=p(\mu_c^{(n)}|\nu_c^{(n-1)},\kappa_c^{(n-1)})\propto$$

$$|\kappa_c^{(n-1)}|^{1/2}\exp[(\mu_c^{(n)}-\nu_c^{(n-1)})^T\kappa_c^{(n-1)}(\mu_c^{(n)}-\nu_c^{(n-1)})] \tag{9}$$

其中 $\varphi_c^{(n-1)}=\left(\nu_c^{(n-1)},\kappa_c^{(n-1)}\right)$ ，代表累積（*n-1*）個語句對應之轉換參數事前機率統計量， $\nu_c^{(n-1)}$ 和 $\kappa_c^{(n-1)}$ 分別為此高斯函式之平均值向量和精確矩陣。

3.2 轉換為主貝氏預測分類法則之實現

在以轉換為主之貝氏預測分類法則中，對於一個未知之測試語料的決策是根據一預測機率分佈來決定，其計算是藉由導入轉換參數的不確定性而得到，然而式子（5）中的預測機率密度函式的計算是相當困難的。Huo *et al.*[8]對積分使用 Laplace 的方法去近似得到預測機率密度函式的值。他們也考慮隱藏式馬可夫模組 Missing Data 問題的特性來計算預測機率密度函式。因此，將觀測語料 $\mathbf{X}_n$ 之狀態和混合數序列 $(\mathbf{s}_n,\mathbf{l}_n)$ 整合進預測機率密度函式的計算中，產生下列式子[12]

$$\tilde{p}_\eta(\mathbf{X}_n|W)=\sum_{\mathbf{s}_n,\mathbf{l}_n}\int p(\mathbf{X}_n,\mathbf{s}_n,\mathbf{l}_n|W,\eta)p(\eta|\varphi^{(n-1)},W)d\eta$$

$$\cong\max_{\mathbf{s}_n,\mathbf{l}_n}\int p(\mathbf{X}_n,\mathbf{s}_n,\mathbf{l}_n|W,\eta)p(\eta|\varphi^{(n-1)},W)d\eta \tag{10}$$

其中所有可能狀態和混合數序列 $(\mathbf{s}_n,\mathbf{l}_n)$ 之機率總和，藉由代入最相似的狀態和混合數序列來趨近，此即所謂的 Viterbi 貝氏預測分類（Viterbi BPC）。在[12]中，一近似 Viterbi 貝氏演算法被提出並用來搜尋最佳未知的狀態和混合數序列 $(\mathbf{s}_n^*,\mathbf{l}_n^*)$ ，並且找出近似的預測機率密度函式。事實上，此 Viterbi 貝氏搜尋演算法並不能精確地完成 Viterbi 貝氏預測分類法則[12]。因此，另一種近似預測機率密度函式的方法，是直接地計算每一個觀測音框的預測機率密度函式，取代傳統使用的高斯機率密度函式。此預測機率密度函式可視為一補償過的觀測機率密度函式，並且將其應用到式子（2）之 MAP 決策中，如此便能近似貝氏預測分類法則。在[12]，此方法稱為 BP-MC （Bayesian Predictive Density Based Model Compensation）。在他們的實驗中， BP-MC 和 Viterbi BPC 在效能上互相媲美。所以，在本論文中採用 BP-MC 的方法實現式子（4）（5）以轉換為主之貝氏預測分類法則。使用這個方法，式子（6）中的狀態觀測機率密度函式被修改為

$$\tilde{p}(\mathbf{x}_t^{(n)}|\lambda_i)=\sum_{k=1}^{K}\omega_{ik}f(\mathbf{x}_t^{(n)}|\lambda_{ik}) \tag{11}$$

其中對於單一音框之補償過的觀測機率密度函式 $f(\mathbf{x}_t^{(n)}|\lambda_{ik})$ 被定義為

$$f(\mathbf{x}_t^{(n)}|\lambda_{ik})=\int p(\mathbf{x}_t^{(n)}|\lambda_{ik},\eta_c)p(\eta_c|\varphi_c^{(n-1)})d\eta_c \tag{12}$$

因此，我們將藉由把（11）式之預測機率密度函式嵌入 MAP 辨識器中進而建立以轉換為主之貝氏預測分類法則。此法則的重點即在推導（12）式中貝氏預測相似度量測（Bayesian Predictive Likelihood Measure, BPLM）。

3.3 貝氏預測相似度量測（BPLM）之推導

在式子（12）貝氏預測相似度量測中，我們把式子（8）和（9）代入式子（12）化簡即可得到

$$f(\mathbf{x}_t^{(n)}|\lambda_{ik}) \propto |r_{ik}|^{1/2} |\kappa_c^{(n-1)}|^{1/2} \int \exp\left\{ -\frac{1}{2}\left[(\mathbf{x}_t^{(n)} - \mu_{ik} - \mu_c)^T r_{ik} (\mathbf{x}_t^{(n)} - \mu_{ik} - \mu_c) \right.\right.$$
$$\left.\left. + (\mu_c - \nu_c^{(n-1)})^T \kappa_c^{(n-1)} (\mu_c - \nu_c^{(n-1)}) \right] \right\} d\mu_c \tag{13}$$

由於式子（13）中指數的部分可應用統計書[4]上的定理得到下列的等式

$$-\frac{1}{2}\left[(\mathbf{x}_t^{(n)} - \mu_{ik} - \mu_c)^T r_{ik} (\mathbf{x}_t^{(n)} - \mu_{ik} - \mu_c) + (\mu_c - \nu_c^{(n-1)})^T \kappa_c^{(n-1)} (\mu_c - \nu_c^{(n-1)}) \right]$$
$$= -\frac{1}{2}\left\{ (\mu_c - \overline{\mathbf{m}})^T (\kappa_c^{(n-1)} + r_{ik}) (\mu_c - \overline{\mathbf{m}}) \right.$$
$$\left. + (\kappa_c^{(n-1)} + r_{ik})^{-1} \kappa_c^{(n-1)} r_{ik} (\mathbf{x}_t^{(n)} - \mu_{ik} - \nu_c^{(n-1)}) (\mathbf{x}_t^{(n)} - \mu_{ik} - \nu_c^{(n-1)})^T \right\} \tag{14}$$

其中

$$\overline{\mathbf{m}} = (\kappa_c^{(n-1)} + r_{ik})^{-1} \left[\kappa_c^{(n-1)} \nu_c^{(n-1)} + r_{ik} (\mathbf{x}_t^{(n)} - \mu_{ik}) \right] \tag{15}$$

因此（13）式可以整理成

$$|r_{ik}|^{1/2} |\kappa_c^{(n-1)}|^{1/2} \exp\left[-\frac{1}{2} (\mathbf{x}_t^{(n)} - \mu_{ik} - \nu_c^{(n-1)})^T (\kappa_c^{(n-1)} + r_{ik})^{-1} \kappa_c^{(n-1)} r_{ik} (\mathbf{x}_t^{(n)} - \mu_{ik} - \nu_c^{(n-1)}) \right]$$
$$\times |\kappa_c^{(n-1)} + r_{ik}|^{-1/2} \int |\kappa_c^{(n-1)} + r_{ik}|^{1/2} \exp\left[-\frac{1}{2} (\mu_c - \overline{\mathbf{m}})^T (\kappa_c^{(n-1)} + r_{ik}) (\mu_c - \overline{\mathbf{m}}) \right] d\mu_c \tag{16}$$

在（16）式中包含了一個高斯機率密度函數的積分項，其積分結果為 1 ，因此我們得到

$$f(\mathbf{x}_t^{(n)}|\lambda_{ik}) \propto |r_{ik}|^{1/2} |\kappa_c^{(n-1)}|^{1/2} |\kappa_c^{(n-1)} + r_{ik}|^{-1/2}$$
$$\exp\left[-\frac{1}{2} (\mathbf{x}_t^{(n)} - \mu_{ik} - \nu_c^{(n-1)})^T (\kappa_c^{(n-1)} + r_{ik})^{-1} \kappa_c^{(n-1)} r_{ik} (\mathbf{x}_t^{(n)} - \mu_{ik} - \nu_c^{(n-1)}) \right] \tag{17}$$

本實驗我們假設 r_{ik} 和 $\kappa_c^{(n-1)}$ 均為對角化矩陣，我們可推導出

$$f(\mathbf{x}_t^{(n)}|\lambda_{ik}) \propto |\kappa_c^{(n-1)} + r_{ik}|^{1/2}$$

$$\exp\left[-\frac{1}{2}(\mathbf{x}_t^{(n)}-\mu_{ik}-\nu_c^{(n-1)})^T(\kappa_c^{(n-1)}+r_{ik})(\mathbf{x}_t^{(n)}-\mu_{ik}-\nu_c^{(n-1)})\right] \tag{18}$$

此結果將應用在辨識的決策中，我們可以觀察到此貝氏預測相似度量測為高斯分佈，與原本未補償的高斯分佈之差別在於（18）式把事前機率密度函式的平均值向量 $\nu_c^{(n-1)}$ 和精確矩陣 $\kappa_c^{(n-1)}$ 分別加到隱藏式馬可夫模組中的平均值向量 μ_{ik} 與精確矩陣 r_{ik} ，因此在此決策中引入轉換參數的事前機率統計量，可以使系統的決策更具強健性。

4. 事前機率之累進學習

在我們提出的方法中，除了強健的 TBPC 決策之外，另一個重要的特點為事前機率累進學習的能力。在[9]中，闡述以 BPC 為基礎之語音辨識系統，事前機率密度函式佔了相當重要的地位，他們評估出事前機率統計量的變化對於 BPC 的效能具有關鍵性的影響。因此，我們把累進學習之技術應用到本論文所提出的 TBPC 決策之事前機率統計量，使得辨識系統能不斷地追蹤到最新環境的統計特性，讓辨識系統更具強健性。

4.1 Viterbi 近似法

在式子（3）近似最佳事後機率估測法中，第 n 個累進轉換參數 $\eta^{(n)}$ 的估測是藉由最大化 $p(\mathbf{X}_n|W^{(n)},\eta)$ 和 $p(\eta|\varphi^{(n-1)})$ 的乘積而得到，其中 $p(\mathbf{X}_n|W^{(n)},\eta)$ 為目前語料 $\mathbf{X}_n$ 之觀測機率密度函式， $p(\eta|\varphi^{(n-1)})$ 為 η 之事前機率密度函式且其參數 $\varphi^{(n-1)}$ 由觀測語料 χ^{n-1} 累進學習得到。因為 TBPC 決策，在辨識出最佳 $W^{(n)}$ 的過程中， $\mathbf{X}_n$ 所對應之最相似（most likely）狀態和混合數序列 $(\mathbf{s}_n^*,\mathbf{l}_n^*)$ 亦可同時產生，所以在（3）式參數估測的演算法中，其 missing data 的問題可用下列的 Viterbi 近似法來克服

$$\eta^{(n)} \cong \arg\max_{\eta} R(\eta) = \arg\max_{\eta} p(\mathbf{X}_n,\mathbf{s}_n^*,\mathbf{l}_n^*|W^{(n)},\eta)p(\eta|\varphi^{(n-1)}) \tag{19}$$

第 c 類的 $R(\eta_c)$ 可以推導出以下的高斯機率函式

$$R(\eta_c) \propto \left|\kappa_c^{(n)}\right|^{1/2}\exp\left[-\frac{1}{2}\left(\mu_c-\nu_c^{(n)}\right)^T\kappa_c^{(n)}\left(\mu_c-\nu_c^{(n)}\right)\right] \propto p(\mu_c|\varphi_c^{(n)}) \tag{20}$$

其中參數統計量 $\varphi_c^{(n)}=\left(\nu_c^{(n)},\kappa_c^{(n)}\right)$ 表示為

$$\nu_c^{(n)} = \left(\kappa_c^{(n-1)}+\sum_{i,k\in\Omega_c}c_{ik}r_{ik}\right)^{-1}\cdot\left(\kappa_c^{(n-1)}\nu_c^{(n-1)}+\sum_{i,k\in\Omega_c}c_{ik}r_{ik}\overline{\mathbf{b}_c}\right) \tag{21}$$

$$\kappa_c^{(n)} = \kappa_c^{(n-1)} + \sum_{i,k\in\Omega_c} c_{ik} r_{ik} \tag{22}$$

其中

$$c_{ik} = \sum_t \delta(s_{n,t}^* - i)\delta(l_{n,t}^* - k) \tag{23}$$

$$\overline{\mathbf{b}_c} = \frac{\sum_t \sum_{i,k\in\Omega_c} (\mathbf{x}_t^{(n)} - \mu_{ik})\delta(s_{n,t}^* - i)\delta(l_{n,t}^* - k)}{\sum_t \sum_{i,k\in\Omega_c} \delta(s_{n,t}^* - i)\delta(l_{n,t}^* - k)} \tag{24}$$

從（21）（22）式可知事前機率密度函式之參數 $\varphi_c^{(n-1)}$ 經觀測語料 $\mathbf{X_n}$ 辨識後更新為 $\varphi_c^{(n)} = \left(\nu_c^{(n)}, \kappa_c^{(n)}\right)$，因此我們的重點在於建立起可重複產生的高斯事前/事後機率對，這符合統計學上共軛分佈之特性，也就是當隨機變數其事前機率分佈為高斯分佈，則對於任何樣本量和樣本值，此隨機變數之事後機率分佈也是高斯分佈[4]，而在（19）式所得到的轉換參數 μ_c 之事後機率可表示為 $p\left(\mu_c \middle| \varphi_c^{(n)}\right)$，所以新的參數 $\left(\nu_c^{(n)}, \kappa_c^{(n)}\right)$ 可視為更新過之轉換參數事前機率的統計量，並將應用在下一句測試語句 $\mathbf{X}_{n+1}$ 的 TBPC 辨識。因此，線上累進學習能夠累進地提供強健性 TBPC 決策所需要轉換參數不確定性的最新知識。

4.2 初始參數之估測

由於我們所提出的 TBPC 決策法則導入轉換參數事前機率的統計量 $\left(\nu_c^{(n-1)}, \kappa_c^{(n-1)}\right)$，而對於此統計量我們所採取的策略為累進式的學習，所以，當第一測試語料區塊 $\mathbf{X}_1$ 進入 TBPC 決策時，此時我們需要一初始值 $\left(\nu_c^{(0)}, \kappa_c^{(0)}\right)$ 提供 TBPC 決策。基本上，此初始值對於未知的轉換參數要能提供足夠的知識，如此才能夠累進地產生可靠的參數值 $\left(\nu_c^{(n)}, \kappa_c^{(n)}\right)$。為了要讓此初始值能夠充分地反應出轉換的物理意義，我們從大量的訓練語料中估測出此初始值，估測的方式如下所示

$$\nu_c^{(0)} = \frac{\sum_t \sum_{i,k\in\Omega_c} \delta(s_{n,t}^* - i)\delta(l_{n,t}^* - k)(\mathbf{x}_t - \mu_{ik})}{\sum_t \sum_{i,k\in\Omega_c} \delta(s_{n,t}^* - i)\delta(l_{n,t}^* - k)} \tag{25}$$

$$\kappa_c^{(0)} = \frac{\sum_t \sum_{i,k\in\Omega_c} \delta(s_{n,t}^* - i)\delta(l_{n,t}^* - k) r_{ik}}{\sum_t \sum_{i,k\in\Omega_c} \delta(s_{n,t}^* - i)\delta(l_{n,t}^* - k)} \tag{26}$$

其中 $\mathbf{x}_t$ 代表訓練語料中音框 t 的特徵向量，也就是我們在 Segmental K-Means 訓練的最後

一次遞迴中，先找出訓練語料所對應的 HMM 參數。 $\nu_c^{(0)}$ 的產生為所有訓練語料與其所對應屬於轉換類別 c 之 HMM 參數 μ_{ik} 之偏差（bias）的平均值，$\kappa_c^{(0)}$ 為屬於 c 類別中精確矩陣的平均值，因此 $\left(\nu_c^{(0)},\kappa_c^{(0)}\right)$ 可充分地代表轉換參數的初始統計量。

5. 實驗

5.1 語料庫

我們準備了兩組語料庫[1]，一組是以近距離麥克風之方式錄下的*乾淨語料*，這組語料總共有 1400 句中文連續數字，包含了 70 位男生 70 位女生，每個人發音 10 句，其中 1000 句 50 男 50 女用來訓練乾淨之非特定語者 HMM 參數，另外 400 句 20 男 20 女用來做測試，另一組是*噪音語料庫*，此組語料庫為實際汽車環境下以遠距離麥克風方式錄得，此組語料包含 5 位男生 5 位女生發音的中文連續數字語料，每人分別在車速 0 公里（怠速路況）錄製 10 句，50 公里（市區路況）錄製 20 句，90 公里（高速公路路況）錄製 30 句，其中每位語者在不同路況下取出 5 句做為調整語料，其餘的做為測試語料。所使用的汽車為 TOYOTA COROLLA 1.8（錄製 2 男 2 女）和 YULON SENTRA 1.6（錄製 3 男 3 女），使用的錄音設備為 MD 隨身聽，型號為 MZ-R55，錄音採用遠距離錄音麥克風，型號為 SONY ECM-717 ，麥克風與語者間的距離約 50 公分，錄音時冷氣開啟，窗戶緊閉，所錄下的語音經由音效卡轉成數位音檔。此兩組語料庫取樣頻率均為 8 kHz ，以 16 bit 的方式儲存，連續數字長度為 3 至 11 個數字長。

為了瞭解汽車噪音語料庫中噪音程度，我們從錄得的噪音語料計算出兩種汽車在不同路況下的信號雜訊比（SNR），由於乾淨語音訊號並無法精確地從噪音語料中還原得到，因此，我們假設乾淨語音信號與背景噪音為互相獨立，故 SNR 的計算方式定義為

$$SNR(dB)=10\log_{10}\left(\frac{\sigma_{noisy\ speech}^2-\sigma_{noise}^2}{\sigma_{noise}^2}\right) \tag{27}$$

其中 $\sigma_{noisy\ speech}^2$ 為噪音語料的變異數， σ_{noise}^2 為背景噪音的變異數，SNR 以分貝（dB）為單位。由於使用兩種不同等級的汽車錄音，因此我們分別算出其所錄製語料的 SNR ，結果列於表一，我們可看到，車速越快所產生的噪音程度越高，SNR 值越低，另一方面，汽車等級越高，所產生的噪音程度越低，SNR 值越高。

信號雜訊比 SNR（dB）			
	YULON	TOYOTA	平均
0 公里	5.63	10.3	7.96
50 公里	-6.53	0.34	-3.1
90 公里	-10.14	-3.77	-6.96
乾淨語料	25.1		

表一、不同環境語料之信號雜訊比

5.2 辨識架構與基本辨識結果

在本論文的實驗中，語音特徵參數包含 12 階 LPC derived cepstrum、12 階 delta cepstrum、1 階 delta log energy 和 1 階 delta delta log energy，共 26 個維度。語音模組 HMM 的規格部分，每個數字用 7 個狀態來代表，每一語句前端插入一個背景雜訊狀態，後端亦插入一個背景雜訊狀態，而語句中數字與數字之間插入一個選擇性背景雜訊狀態，此三個背景雜訊狀態均不相同，所以我們的 HMM 共有 73 個狀態，每個狀態的混合數為 4 個。

在實驗部分，我們首先實現出一組在乾淨環境、0 公里車速、50 公里車速和 90 公里車速的基本實驗結果，實驗數據以數字錯誤率 （digit error rate, DER）來表示，乾淨環境下的測試語料為乾淨語料庫中的 400 句測試語料，而 0 公里、 50 公里和 90 公里車速下的測試語料分別為噪音語料庫中的 50 句、150 句和 250 句（每人在各公里車速分別有 5、15 和 25，共有 10 人）測試語料，在基本系統實驗中，語音辨識器是使用乾淨非特定語者 HMM 參數，不做任何的補償，實驗結果列於表二

測試語料	**數字錯誤率 DER（%）**
乾淨語料	10.6
0 公里噪音語料	25.61
50 公里噪音語料	54.97
90 公里噪音語料	62.33

表二、乾淨語料、0、50、90 公里語料之基本實驗結果

5.3 轉換類別個數與數字錯誤率之關係

在以轉換為主的貝氏預測分類法則中，辨識率會受轉換參數類別的個數影響，如果能夠

適當地增加類別的個數，將可提昇辨識率。這裡我們做了個簡單的實驗，利用人工把非特定語者 HMM 參數分為兩類，第一類為背景雜訊狀態類，第二類為非背景雜訊狀態類，因此我們需要估測出兩類的事前機率統計量做 TBPC 的決策。其實驗結果為表三，在表中列出使用 1 個和 2 個轉換類別個數的數字錯誤率，轉換參數之事前機率統計量是從測試語料中採非監督式的學習。由於本論文所提出的策略為累進式的學習，學習的效能會受測試語料的測試順序影響，因此我們將各個公里語料隨機產生 10 組測試順序，然後再將 10 組測試結果取平均，列出實驗結果。我們可看到使用 2 個轉換類別個數的數字錯誤率明顯降低。因此，在以下的實驗中，轉換類別個數均採用兩個。

測試語料 ＼ 類別個數	1	2
0 公里	14.32	12.53
50 公里	39.94	36.24
90 公里	51.65	46.32

表三、TBPC-OPE 中轉換類別個數與數字錯誤率之關係

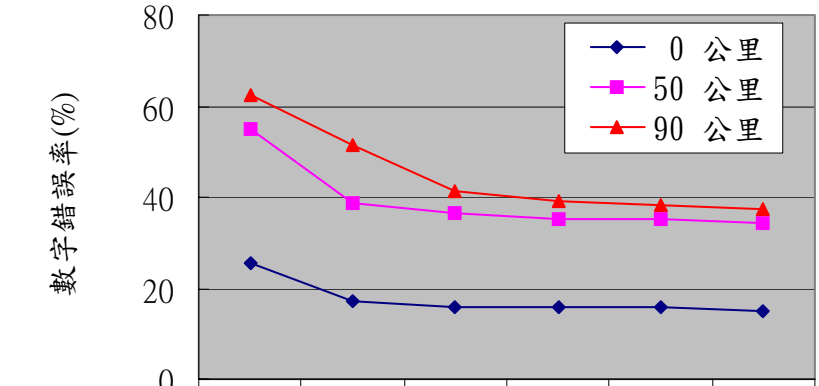
5.4 調整語料量與數字錯誤率之關係

為了瞭解調整語料與數字錯誤率之關係，我們利用監督式學習的方式，首先利用調整語料，以累進學習的方式估測轉換參數之事前機率統計量，然後把估測出的統計量應用在測試階段的 TBPC 決策，在測試階段並不做事前機率統計量的累進。所使用的調整語料為噪音語料庫中的調整語料，在各個車速下每人均有 5 句調整語料，在實驗中，我們考慮語者不匹配的問題，因此把每個語者的調整語料分開做監督式學習，然後把由調整語料累進學習到的事前機率統計量，應用在測試階段的 TBPC 決策中，在測試階段並不做事前機率統計量的更新。實驗結果列於圖三，由圖中可看出 TBPC 只需少量的調整語料，即使僅僅一句，就可明顯地降低數字錯誤率，例如 50 公里之語料，數字錯誤率降低了 30% 。當調整語料累進時，數字錯誤率呈現遞減之趨勢，我們可看到當使用完 5 句調整語料語料後，在各個公里車速下之語料均有 40 % 數字錯誤率的降低，由此可知，以轉換為主的貝氏預測分類法則加上事前機率累進式的學習，能夠累進地提昇辨識率。

5.5 不同貝氏預測分類器之比較

在這個部份的實驗，我們實現 Surendran 與 Lee 所提出的貝氏預測分類器[19]和 Jiang

與 Huo 所提出的貝氏預測分類器[11]，實驗結果比較於表四，



	0	1	2	3	4	5
0 公里	25.61	17.28	15.74	16.04	15.74	15.12
50 公里	54.97	38.48	36.51	35.26	34.95	34.12
90 公里	62.33	51.47	41.3	39.04	38.22	37.47

調整語料之句數

圖三、TBPC-OPE 中調整語料量與數字錯誤率之關係

測試語料＼方法	Baseline	Jiang and Huo	Surendran and Lee	TBPC-OPE
乾淨語料	10.6	8.51	8.4	7.47
0 公里	25.6	18.61	15.43	12.53
50 公里	55	49.83	38.38	36.24
90 公里	62.3	60.25	50.91	46.32

表四、不同 BPC 方法之數字錯誤率之比較

由實驗數據可看出，我們所提出的方法 TBPC-OPE 對於每種測試語料，數字錯誤率均比其他方法低，雖然 Jiang 與 Huo 所提出的貝氏預測分類器之辨識結果比基本系統好，但由於其所考慮的是非特定語者 HMM 參數中平均值向量的不確定性，所以當訓練與測試語料的不匹配程度增大時，利用 BPC 決策找出的最佳狀態和混合數序列 $(\mathbf{s}_n^*,\mathbf{l}_n^*)$ 的資訊常常不是很可靠，而其方法對於每一個 HMM 狀態和混合數的平均值向量均需一組事前機率統計量，當要利用少量觀測語句且不可靠的資訊 $(\mathbf{s}_n^*,\mathbf{l}_n^*)$ 來更新很多組的事前機率統計量，此時將有可能造成事前機率統計量的更新錯誤，導致辨識率的提昇有限，從乾淨、0 公里、50 公里、90 公里測試語料之辨識結果，即可得到驗證。對於 Surendran 與 Lee 所提出的貝氏預測分類器，

其考慮的是轉換參數的不確定性。轉換類別之個數在實驗中我們設定為 2 個，因此將能利用 Viterbi 演算法找出較佳的 $(\mathbf{s}_n^*,\mathbf{l}_n^*)$ 來估測出較可靠之轉換參數事前機率統計量，應用在第二次辨識時的 TBPC 決策，但由於其 TBPC 使用到的轉換參數事前機率統計量是利用當時輸入的測試語料估測出來的，並沒有累進式學習的能力，因此其數字錯誤率高於我們提出的 TBPC-OPE。在這個部分的實驗中，另一個值得注意的是乾淨測試語料的辨識，由於 BPC 把參數的不確定性引入辨識的決策中，因此 BPC 也能克服訓練模組時模組的不正確及語者的差異，由實驗結果得知， BPC 決策不僅能應用在噪音環境下做語音辨識，在乾淨的環境中也能提昇辨識率。

在語音辨識的應用中，辨識時間亦是重要之考量之一，因此我們也列出辨識時間的比較。實驗中我們所使用的設備為 Pentium III 450 及 256 Mega RAM，實驗結果列於表五，這裡我們列出每句語料所花的平均辨識秒數，數據顯示我們所提出的 TBPC-OPE 所花的辨識時間比其他兩種 BPC 方法少。由於 Surendran 和 Lee 所提出的方法需要兩次的辨認且須估測轉換參數事前機率的統計量，因此所花的辨識時間為最久。然而， Jiang 和 Huo 所提出的方法，其只須一次的辨識並對每個狀態每個混合數的事前機率統計量做估測與更新，因此所花的辨識時間比 Surendran 和 Lee 的方法少，而我們所提出的 TBPC-OPE 之辨識時間比 Jiang 和 Huo 的方法少，這是因為 TBPC-OPE 需要更新轉換參數事前機率統計量只有兩類，所以 TBPC-OPE 的辨識時間比 Jiang 和 Huo 的方法少，只比基本系統的辨識時間多一點點。

方法	Baseline	Jiang and Huo	Surendran and Lee	TBPC-OPE
辨識時間（秒／句）	0.138	0.245	0.382	0.192

表五、不同 BPC 法則之辨識時間比較

經由實驗結果發現我們所提出的 TBPC-OPE 降低的數字錯誤率最多，所花的辨識時間最短，因此 TBPC-OPE 是相當地具有潛力的方法。

6. 即時展示系統

為了實際評估我們所提出的演算法效能，我們實作了一套即時展示系統，直接在線上做語音辨識，此展示系統並不是直接開車在車上做展示；而是先錄製一段汽車背景雜訊，用喇叭放出來，以模擬實際的汽車噪音環境，而麥克風的位置是以免持式遠距離麥克風為主，大

約和語者相距 25 到 35 公分之間，線上錄下一段語音後做即時語音辨識。在展示系統中列出了三種方法的辨識結果：第一是基本系統的結果，在這個部分並不使用任何的補償技術，第二為 Jiang 與 Huo 所提出的貝氏預測分類器之辨識結果，第三為我們所提出的 TBPC-OPE 之辨識結果。

6.1 背景模組

由於在噪音環境下的中文連續語音辨識並不能達到百分之百的正確，而為了提昇噪音環境下的切音結果，我們在系統中提供可線上訓練背景模組的選項，當此選項勾選時，在辨識時的背景雜訊狀態均用此線上訓練出來的背景模組來取代。背景模組的訓練方式，會依實際錄得的聲音長短訓練出不同混合數個數的背景模組，如 32 或 16 個混合數。所使用的訓練演算法是用向量量化（Vector Quantization, VQ）演算法[15]。

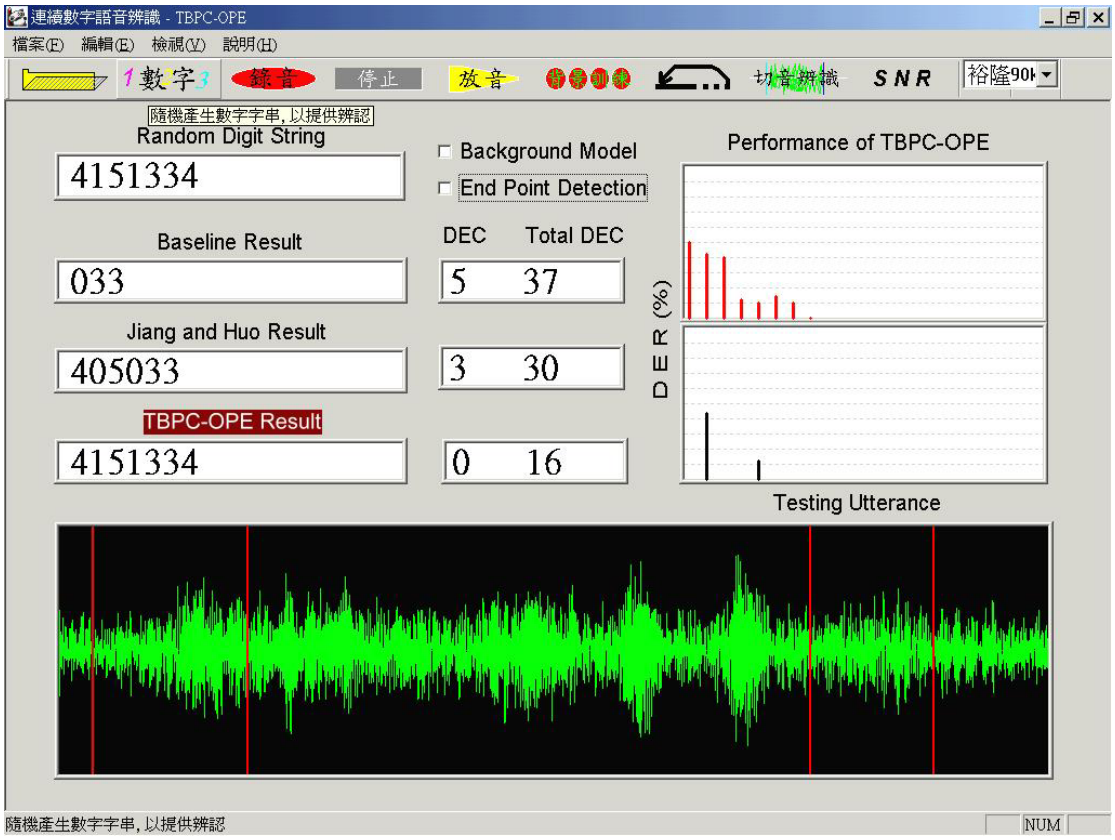
6.2 即時展示系統介面

如圖四所示，此即為我們所發展出的展示系統介面。所使用的工具為 Visual C++ 6.0 ，工作平台為 Microsoft Windows 2000。所使用的音效卡為創巨公司生產的，可支援全雙工，能在撥放噪音的同時進行錄音，因此我們把噪音種類的撥放選項加入展示系統的介面，讓使用者可以更方便選擇所要模擬的車速路況。因為在展示時，噪音撥放的音量大小會影響辨識的效能，因此在展示系統的介面中，我們提供可線上計算 SNR 的功能選項，以瞭解系統測試時的噪音強度。由於本系統做連續數字的辨識，因此我們可利用電腦隨機產生一組長度為 7 至 10 個數字字串，使用者就唸此字串，然後把每個方法得到的辨識結果和隨機產生的字串做比較，即可算出每個方法辨識的數字錯誤個數 DEC（digit error count）。為了要能客觀地顯示出 TBPC-OPE 的效能，我們也列出累積至目前的總數字錯誤個數 Total DEC，另一方面，為了顯示出 TBPC-OPE 累進學習演算法的效能，我們畫出 TBPC-OPE 每一句和每三句的數字錯誤率 DER（%）。

7. 結論

在本論文中，我們為噪音環境下的語音辨識提出一具有線上累進學習能力的 TBPC 強健性決策，由於 TBPC 決策把轉換參數的不確定性導入辨識的決策中，因此在噪音環境下的辨識效能會比傳統用 ML 或 MAP 做點估測（point estimate）的方法更具強健性。由於在 TBPC

的決策中使用轉換參數的事前機率統計量，而累進學習的策略能讓此統計量不斷地追蹤最新環境的統計特性，因此使得 TBPC 的決策更具可靠性。從監督式學習的實驗中，我們可觀察到 TBPC 強健性的決策，運用少量的調整語料，即使僅僅一句，就能降低在噪音環境下的數字錯誤率，而且數字錯誤率隨著調整語料的增加而遞減，更驗證了我們所提出的線上累進學習策略。在不同 BPC 應用的比較實驗中，由於我們所提出的是以轉換為主的 TBPC ，且採用累進學習的策略，因此，不論在辨識率或辨識時間方面均比 Surendran 與 Lee 所提出的貝氏預測分類器和 Jiang 與 Huo 所提出的貝氏預測分類器，具有較佳的辨識效能。另外一個值得注意的是， TBPC-OPE 在乾淨的測試語料中也能降低數字錯誤率，也就是克服在訓練模組階段時模組的不正確及語者差異所產生的不匹配問題。



圖四、展示系統之介面

參考文獻

[1] 簡仁宗 ，林敏順 (1999), “音框同步之雜訊補償方法在汽車語音辨識之應用”, 第十二屆計算語言學研討會, pp. 239-251.

[2] C. Chesta, O. Siohan and C.-H. Lee, “Maximum *a posteriori* linear regression for hidden Markov model adaptation”, *Proceedings of European Conference on Speech Communication and Technology (EUROSPEECH)*, vol. 1, pp. 211-214, 1999.

[3] J.-T. Chien, “Online hierarchical transformation of hidden Markov models for speech recognition”, *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 6, pp. 656-667, November 1999.

[4] M. H. DeGroot, *Optimal Statistical Decisions*, McGraw-Hill, 1970.

[5] A. P. Dempster, N. M. Laird and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm”, *J. Royal Statist. Society (B)*, vol. 39, pp. 1-38, 1977.

[6] J. L. Gauvain and C.-H. Lee, “Maximum *a posteriori* estimation for multivariate Gaussian mixture observations of Markov chains”, *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 291-298, 1994.

[7] X. D. Huang, Y. Ariki and M. A. Jack, *Hidden Markov Models for Speech Recognition*, Edinburgh University Press, 1990.

[8] Q. Huo, H. Jiang and C.-H. Lee, “A Bayesian predictive classification approach to robust speech recognition”, *IEEE Proceedings of International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pp. 1547-1550, 1997.

[9] Q. Huo and C.-H. Lee, “A study of prior sensitivity for Bayesian predictive classification based robust speech recognition”, *IEEE Proceedings of International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, vol 2, pp. 741-744, 1998.

[10] Q. Huo and C.-H. Lee, “On-line adaptive learning of the continuous density hidden Markov model based on approximate recursive Bayes estimate”, *IEEE Transactions Speech and Audio Processing*, vol. 5, no. 2, pp. 161-172, March 1997.

[11] H. Jiang, K. Hirose and Q. Huo, “Improving Viterbi Bayesian predictive classification via sequential Bayesian learning in robust speech recognition”, *Speech Communication*, vol. 28, no. 4, 1999.

[12] H. Jiang, K. Hirose and Q. Huo, “Robust speech recognition based on a Bayesian prediction approach”, *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 4, 1999.

[13] C. J. Leggetter and P. C. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models”, *Computer Speech and Language*, vol. 9, pp. 171-185, 1995.

[14] N. Merhav and C.-H. Lee, “A minimax classification approach with application to robust speech recognition”, *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 1, pp. 90-100, 1993.

[15] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[16] B. D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, UK, 1996.

[17] A. Sankar and C.-H. Lee, "A maximum-likelihood approach to stochastic matching for robust speech recognition", *IEEE Transactions on Speech and Audio Processing*, vol. 4, pp. 190-202, 1996.

[18] K. Shinoda and T. Watanabe, "Speaker adaptation with autonomous control using tree structure", *Proceedings of European Conference on Speech Communication and Technology (EUROSPEECH)*, pp. 1143-1146, 1995.

[19] A. C. Surendran and C.-H. Lee, "Predictive adaptation and compensation for robust speech recognition", *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, vol. 2, pp. 463-466, 1998.

結合麥克風陣列及模型調整技術之遠距離語音辨識系統

賴建瑞，簡仁宗

國立成功大學資訊工程學系

Page 199 ~ 213

Proceedings of Research on Computational Linguistics

Conference XIII (ROCLING XIII)

Taipei, Taiwan

2000-08-24/2000-08-25

結合麥克風陣列及模型調整技術之遠距離語音辨識系統

Far-Distant Speech Recognition System Using Combined Techniques of Microphone Array and Model Adaptation

賴建瑞　簡仁宗

國立成功大學資訊工程學系

Email：jtchien@mail.ncku.edu.tw

摘要

本篇論文提出一種可應用於噪音環境下麥克風陣列(Microphone Array)的語音辨識演算法，其主要的目的在於克服傳統電腦語音辨識系統需要使用者頭戴或手持麥克風的不方便。為了消除遠距離麥克風的噪音干擾，我們的方法是先將每個麥克風收集到的語音，利用語音到達每個麥克風角度的不同，使用 Time Domain Cross Correlation (TDCC)演算法找出語者發音的方向及語音到達每個麥可風的時間延遲，再應用 Delay-and-Sum Beamformer 陣列訊號處理技術將語音訊號加強，最後我們再將加強過的語音訊號和語音模型參數間的不匹配用最佳相似度線性回歸(MLLR)的模型調整演算法來克服。在噪音環境下使用麥克風陣列之連續數字辨識實驗中，我們提出來的方法對於提升辨識率有良好的效果。

1. 導論

現實生活環境中，充滿了各式各樣的噪音和回音，這些干擾會嚴重的降低語音辨認系統的效能，其中之一的解決方式是使用頭戴式麥克風(Head-Mounted Microphone)，使得聲音源和麥克風盡可能的靠近，來降低環境噪音和回音的影響。然而使用頭戴式麥克風設備會造成使用者的不便，因此如何發展以免持式麥克風(Hands-Free Microphone)為主的語音辨認系統已成為一個相當重要的研究課題。

基本上，使用麥克風陣列可以進行遠距離錄音，因此可以解決頭戴式麥克風造成使用者不便的問題，而我們常用的麥克風陣列訊號處理技術是採用 Delay-and-Sum Beamformer，它可以克服環境噪音和回音對語音訊號的影響，還原出乾淨的語音。而且此一技術並非針對特定噪音環境，它可適用於任何噪音環境下，得到令人滿意的效果。在本論文中，我們將麥克風陣列應用於降低汽車環境噪音的干擾，以達到提高語音辨認率之目的。

一般的語音辨認皆使用單一麥克風做為語音訊號的輸入，在安靜的環境下已有不錯的辨識成果，然而，當應用在噪音很大的汽車裡，語音辨識的效果將大打折扣，因此，如何抑制噪音並加強語音訊號已成為汽車語音辨識的關鍵性技術。因此本論文中我們使用一組遠距離麥克風陣列做語音訊號輸入，然後使用 TDCC 將每個麥克風之間的時間延遲計算出來，再利用 Delay-and-Sum Beamformer 的方式，得到一組具抗噪性且加強過後的語音訊號。為了使加強過的語音訊號在進行隱藏式馬可夫模型(Hidden Markov Model, HMM)為主語音辨識時有更佳的辨識效果，我們使用最佳相似度線性回歸理論(maximum likelihood linear regression, MLLR) (Leggetter and Woodland, 1995)將原始語音訓練出來的隱藏式馬可夫模型參數做調整，以補償測試語音與模型參數之間的不匹配。

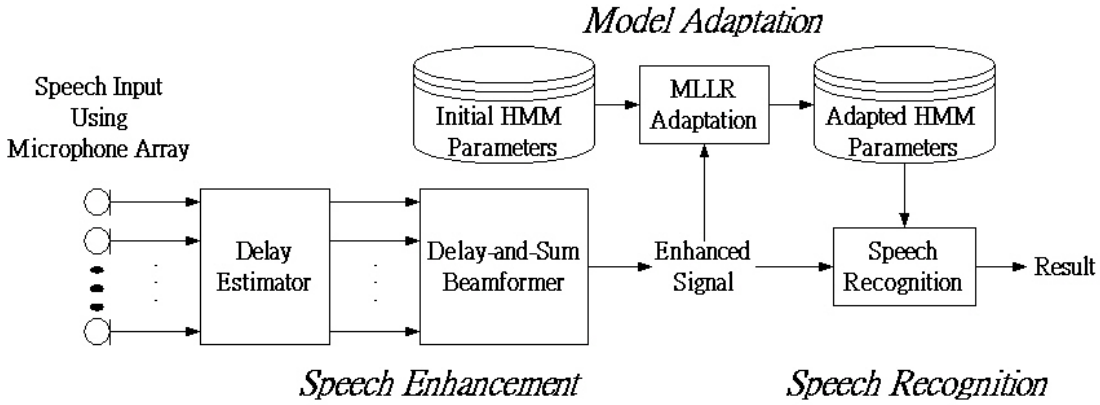
目前台灣的學術研究機構對於架構於麥克風陣列上的語音處理技術尚屬起步階段，在中文語音辨識上的應用發表在相關學術會議及期刊論文尚不多見，然而，國外的研究機構則早已投入此一領域，並且獲得不錯的成果，比較有名的包括三大類的方法，第一類是著重在不同麥克風間時間延遲的計算，它主要是利用語者定位演算法來計算不同麥克風間的時間延遲(Yamada et al., 1996; Inoue et al., 1997)，以及利用不同麥克風之間的頻譜能量來找出不同麥克風的時間延遲(Omologo & Svaizer, 1994, 1996; Giuliani et al., 1996)。第二類的方法是將麥克風間時間延遲併入隱藏式馬可夫模型的參數，它的觀念是擴充傳統語音隱藏式馬可夫模型的參數，加入各種不同的語者角度，並使用一種三維的維特比演算法(Three-Dimensional Viterbi Search)作語音辨識(Yamada et al., 1996, 1998a, 1998b, 1999)。第三類方法是將常用的語音增強技術和麥克風間時間延遲的估測作結合，主要是使用多頻(Multiband)的技術，將語音訊號分成數個不同的頻帶，在各個頻帶上作 Delay-and-Sum Beamformer 然後再將各個頻帶的訊號作合成，產生出強健性的加強語音訊號(Mahmoudi, 1998)。

2. 麥克風陣列語音辨識系統

麥克風陣列語音辨認系統架構圖如圖一所示，可分為語音加強、語音辨識及模型參數調整三部分。

在語音加強部分，輸入語音是由麥克風陣列錄得每個麥克風收集到的語句，再利用語音到達每個麥克風角度的不同，找出語者發音的方向及語音到達每個麥克風的時間延遲，應用 Delay-and-Sum Beamformer 的陣列訊號處理技術將原始語音做訊號加強。本論文中我們提出

TDCC 的演算法來計算時間延遲並於實驗中和其它演算法作比較。另外在語音辨識部分，我們是使用傳統的隱藏式馬可夫模型和一階段(One-Pass)演算法，利用最佳相似度(Maximum Likelihood, ML)法則來進行連續語音辨識。



圖一、麥克風陣列語音辨識系統架構圖

第三部分是模型參數的調整，一般較流行的調整方式有兩種，分別為最佳事後機率(Maximum *A Posteriori*, MAP)調整演算法(Gauvain and Lee, 1994)和最佳相似度線性回歸(Maximum Likelihood Linear Regression, MLLR)演算法(Leggetter and Woodland, 1995)。MAP 和 MLLR 都可以依據目前的測試語料來動態的對語音模型進行調整，主要的分別為 MAP 利用最佳事後機率法則來對語音模型參數做調整，調整的部分為測試語音所對應的狀態及混合數的 HMM 參數，MLLR 則是利用線性回歸的方式根據測試語料來對語音模型進行調整，它是藉由估測出的線性回歸函數，調整所有狀態及混合數的 HMM 參數。本論文中我們使用的調整技術為 MLLR。

2.1 Delay-and-Sum Beamformer

假設有一包含 M 個麥克風的麥克風陣列，每一組相鄰的麥克風的距離為 d，今有一語音訊號(假設為平面波)從我們偵測出的最佳方向 θ_s 傳播過來，麥克風的輸出為 $\mathbf{x}_t^i$，$1 \le i \le M$，則在時間 t 的時候，當第 i 支麥克風收到平面波的訊號，第 i+1 支麥克風則需等到聲波再前進距離 R ($R = d\cos\theta_s$)方可收到訊號，如圖二所示。

若聲波的速度為 C，則第 i+1 個麥克風延遲的時間

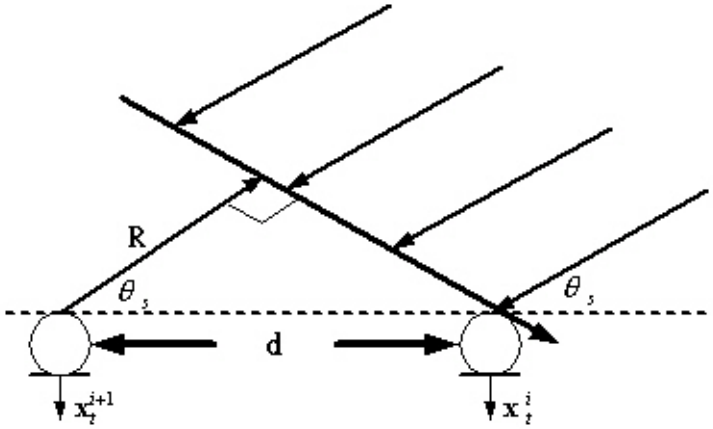
$$\tau = \frac{R}{C} = \frac{d\cos\theta_s}{C} \tag{1}$$

亦即 $\mathbf{x}_t^i = \mathbf{x}_{t+\tau}^{i+1}$，因此我們可以估算第 i 個麥克風與第 1 個麥克風的關係如下：

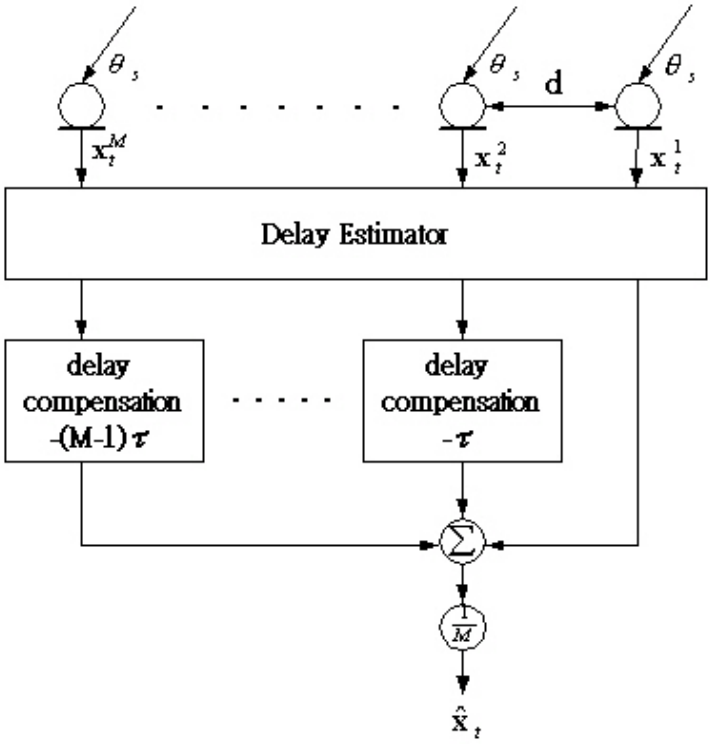
$$\mathbf{x}_t^i = \mathbf{x}_{t+(i-1)\tau}^1 \tag{2}$$

而整個 Delay-and-Sum Beamformer 的輸出 $\hat{\mathbf{x}}_t$，如圖三所示，就是將每個麥克風間的時間延遲作補償後合成再取平均而得

$$\hat{\mathbf{x}}_t = \frac{1}{M}\sum_{i=1}^{M} \mathbf{x}_{t+(i-1)\tau}^i \tag{3}$$



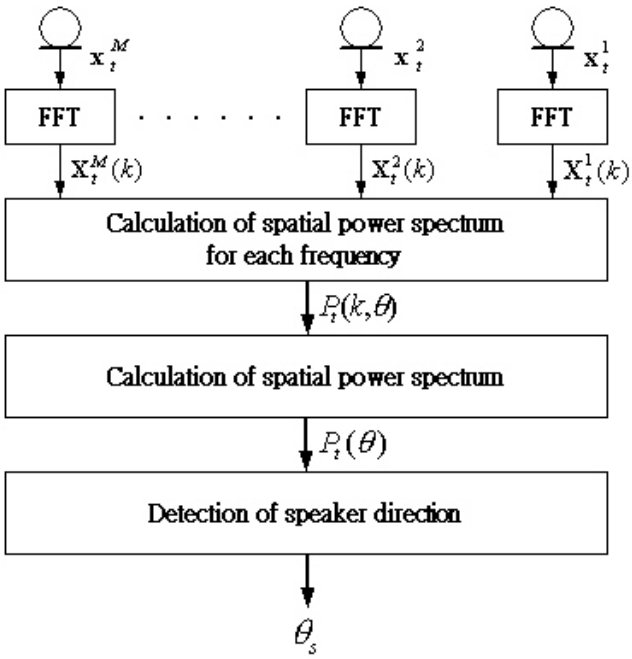
圖二、相鄰麥克風的時間延遲



圖三、Delay-and-Sum Beamformer 流程圖

2.2語者定位演算法(Speaker Localization Algorithm, SLA)

語者定位演算法的主要目的在於估測出語者發話的方向，其系統流程圖如圖四所示，我們將分成下列三部份做說明。



圖四、語者定位演算法(SLA)流程圖

首先，我們將 M 個麥克風在時域上的語音訊號 $\{\mathbf{x}_t^i, i=1,...,M\}$ 經過快速傅利葉轉換後得到的麥克風在頻率上的訊號 $\{\mathbf{X}_t^i(k), i=1,...,M, k=0,...,K-1\}$，其中 i 表示麥克風的引數，k 表示頻率的引數，t 表示音框的引數。

第二部分，我們計算不同聲音方向角度 $\theta=1,...,180$ 的空間功率頻譜

$$P_t(\theta)=\sum_{k=0}^{K-1}P_t(k,\theta)\ ,\ \theta=1,\cdots,180 \tag{4}$$

其中

$$P_t(k,\theta)=\left|\sum_{i=1}^{M}\mathbf{X}_t^i(k)\exp\{j2\pi f_k(i-1)\frac{d\cos\theta}{c}\}\right|^2 \tag{5}$$

f_k 表示 k 所對應的頻率，d 表示相鄰麥克風的間距，c 表示聲波速度。

第三部分我們做語者方向的偵測，基本上，語者方向 θ_s 的偵測是去尋找空間功率頻譜中最大的空間功率頻譜所對應的角度，也就是進行以下的運算

$$\theta_s = \arg\max_{\theta} P_t(\theta) \tag{6}$$

其中，θ 為聲音方向的隨機變數。

求出θ_s後再利用 2.1 節中的 Delay-and-Sum Beamformer 即可求出相鄰麥克風間的時間延遲 τ，對每一個麥克風做時間延遲補償後即可得到加強過後的語音訊號。

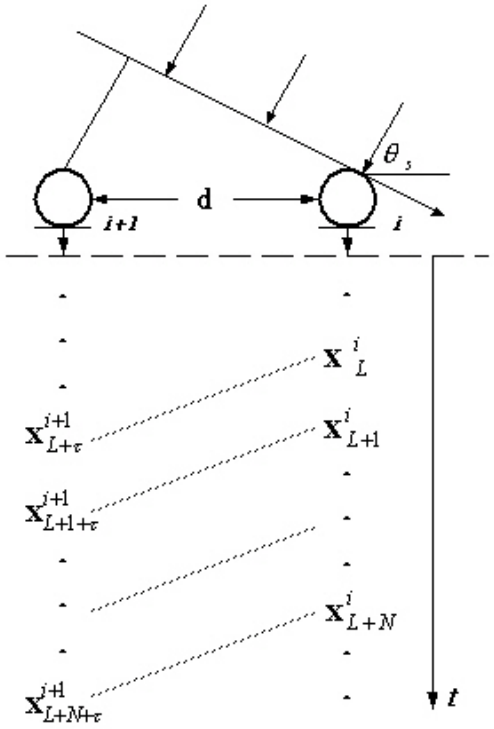
2.3 Time Domain Cross Correlation (TDCC)

不同於語者定位演算法，TDCC 是直接於時域上利用不同麥克風語音間的相關性來求取時間延遲。其基本想法是假設第 i 個麥克風和其所相鄰的第 i+1 個麥克風在第 L 個數位點後的語音訊號分別表示如下：

$$\mathbf{x}_L^i, \mathbf{x}_{L+1}^i, ..., \mathbf{x}_{L+N}^i \quad 和 \quad \mathbf{x}_{L+\tau}^{i+1}, \mathbf{x}_{L+1+\tau}^{i+1}, ..., \mathbf{x}_{L+N+\tau}^{i+1}$$

其中我們取出 N 個數位點，如圖五所示。

在不考慮噪音以及訊號衰減的情形下，若 τ 為麥克風 i 和麥克風 $i+1$ 間的時間延遲，則 $\mathbf{x}_t^i$ 和 $\mathbf{x}_{t+\tau}^{i+1}$ 之間具有最大的相關性且 $\sum_{t=L}^{L+N} \mathbf{x}_t^i \cdot \mathbf{x}_{t+\tau}^{i+1}$ 點積和為最大，此一乘積和可稱之為 Time Domain Cross Correlation。



圖五、TDCC示意圖

經由以上的想法我們發展出 TDCC 的演算法：若現有一麥克風陣列包含有 M 個麥克風，麥克風 i 於時間 t 所收到的訊號稱為 $\mathbf{x}_t^i$，則對語音訊號中任一音框 m 的 TDCC 定義如下

$$C(m)=\sum_{i=2}^{M}\sum_{j=1}^{N}\mathbf{x}_{(m-1)\cdot p+j}^{1}\cdot\mathbf{x}_{(m-1)\cdot p+j}^{i} \tag{7}$$

其中 N 為音框內所包含的點數，P 為音框間的位移點數。我們以第一個麥克風為基準麥克風，所以式子(6)中麥克風的引數 i 從 2 開始累加。若 τ 為時間延遲的隨機變數，則麥克風陣列中相鄰麥克風間的最佳時間延遲 $\hat{\tau}_H$ 為

$$\hat{\tau}_H=\arg\max_{\tau}C(m,\tau) \tag{8}$$

其中

$$C(m,\tau)=\sum_{i=2}^{M}\sum_{j=1}^{N}\mathbf{x}_{(m-1)\cdot p+j}^{1}\cdot\mathbf{x}_{(m-1)\cdot p+j+(i-1)\tau}^{i} \tag{9}$$

這裡我們是以語句中能量最高的音框為基準來計算時間延遲，另外若將語句內全部音框的 TDCC 累加起來，根據此累加值則相鄰麥克風間的時間延遲 $\hat{\tau}_A$ 為

$$\hat{\tau}_A=\arg\max_{\tau}\sum_{m}C(m,\tau) \tag{10}$$

我們在後面的實驗部分會針對此二種方法分別做實驗並分析其結果。計算出相鄰麥克風間的時間延遲 $\hat{\tau}$ 後，再利用 2.1 節中的 Delay-and-Sum Beamformer 即可求出加強過後的語音訊號。

2.4 最佳相似度線性回歸演算法

MLLR 是一種常見使用於語音模型參數調整的技術，它是從測試語料計算出一個轉移矩陣，然後利用此轉移矩陣來調整語音模型中每一個狀態及混合數的平均值向量。

一般我們常使用高斯機率密度函數來表示隱藏式馬可夫模型的觀測機率

$$P(o_t\mid\mu_s,\Sigma_s)=\frac{1}{(2\pi)^{n/2}\left|\Sigma_s\right|^{1/2}}e^{-1/2(o_t-\mu_s)'\Sigma_s^{-1}(o_t-\mu_s)} \tag{11}$$

其中 μ_s 表示平均值向量，Σ_s 表示變異數矩陣，o_t 為觀測到的特徵向量，n 為向量的維度。

定義一個大小為 $n\times(n+1)$ 的轉移矩陣 W_s，它可將擴展後的平均值向量 ξ_s 調整而得到新的平均值向量

$$\hat{\mu}_s=W_s\xi_s \tag{12}$$

其中 $\xi_s = [\omega, \mu_1, \mu_2, ..., \mu_n]'$，$\omega$ 是在進行回歸計算時考慮是否使用偏差量(使用則 ω 為 1，不使用則為 0)。因此調整過後的高斯機率分佈如下所示

$$P(o_t \mid W_s, \mu_s, \Sigma_s) = \frac{1}{(2\pi)^{n/2}\left|\Sigma_s\right|^{1/2}} e^{-1/2(o_t - W_s\xi_s)'\Sigma_s^{-1}(o_t - W_s\xi_s)} \tag{13}$$

根據最佳相似度法則，最佳轉移矩陣為

$$\hat{W}_s = \arg\max_{W_s} P(o_t \mid W_s, u_s, \Sigma_s) \tag{14}$$

這裡我們簡化轉移矩陣內的參數為

$$W_s = \begin{pmatrix} w_{1,1} & w_{1,2} & 0 & .. & 0 \\ w_{2,1} & 0 & w_{2,3} & .. & 0 \\ : & & & & : \\ w_{n,1} & .. & .. & 0 & w_{n,n+1} \end{pmatrix} \tag{15}$$

也就是將轉移矩陣改寫為以下的轉移參數向量

$$w_s = [w_{1,1}, ..., w_{n,1}, w_{1,2}, w_{2,2}, ..., w_{n,n+1}]' \tag{16}$$

那麼最佳轉移參數向量可以由 EM 演算法(Dempster et al.,1977)推導而得

$$\hat{\mathbf{w}}_s = \left[\sum_{r=1}^{R}\sum_{t=1}^{T}\gamma_{s_r}(t) D'_{s_r} C^{-1}_{s_r} D_{s_r}\right]^{-1}\left[\sum_{r=1}^{R}\sum_{t=1}^{T}\gamma_{s_r}(t) D'_{s_r} C^{-1}_{s_r} o_t\right] \tag{17}$$

其中 r 為狀態的索引，t 為時間的索引，γ_{s_r} 為一事後機率，若 o_t 經由 Viterbi Decoding 後對應到狀態 s_r 則其值為 1 否則為 0。D_s 定義為

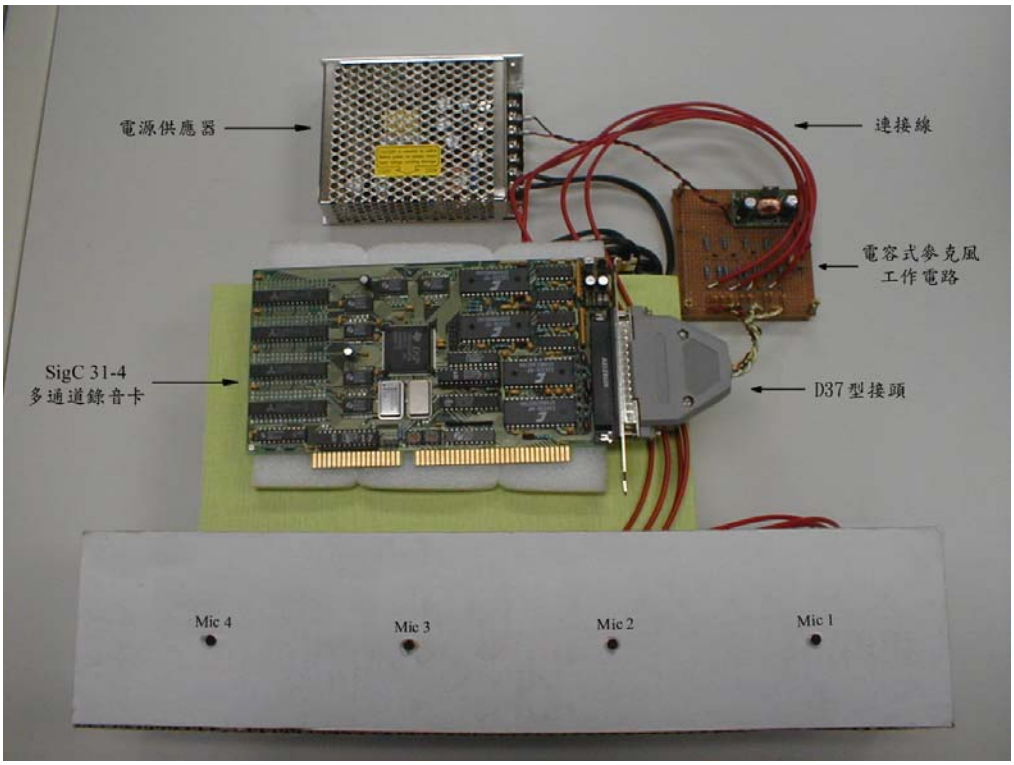
$$D_s = \begin{pmatrix} \omega & 0 & .. & .. & 0 & \mu_1 & 0 & .. & .. & 0 \\ 0 & \omega & 0 & .. & .. & 0 & \mu_2 & 0 & .. & 0 \\ : & & & & & & & & & : \\ 0 & .. & 0 & \omega & 0 & .. & .. & 0 & \mu_{n-1} & 0 \\ 0 & .. & .. & 0 & \omega & 0 & .. & .. & 0 & \mu_n \end{pmatrix} \tag{18}$$

3. 實驗結果

3.1 麥克風陣列錄音設備

麥克風陣列錄音設備如圖六所示，主要的部分包含多通道錄音卡 SigC31-4、四個全方向電容式麥克風、供給麥克風運作的電源供應器和工作電路以及必要的連接線。

多通道錄音卡 SigC31-4 是由美國 Signalogic 公司所生產的，為一 4 個通道的錄音卡，使用的數位訊號處理晶片(DSP processor)為德州儀器公司(TI)所生產的 TM8320C31，可同時提供 4 個通道進行錄音的動作，此錄音卡的介面為 ISA 介面可裝於個人電腦上，並有提供 D37 型接頭經由工作電路和麥克風相連接，透過所附的軟體即可利用 4 個麥克風同時錄音，電容式麥克風我們使用國內音賜公司所生產的全方向電容式麥克風(Omni-directional Condenser Microphone)，型號為 ECM9D，所對應的頻寬為 20~10000Hz，靈敏度為-38±3dB，訊噪比(signal-to-noise ratio, SNR)大於 60dB，工作電壓則介於 DC 3V 至 DC 10V 之間。工作電路主要的作用是將電源供應器所提供的電力進行穩壓，之後再送至麥克風提供錄音時所需的電力，並保持麥克風錄音時訊號的穩定，並將訊號送至多通道錄音卡 SigC31-4。此工作電路是依據音賜公司針對電容式麥克風的建議電路稍加修改後由我們自行焊接製作的。



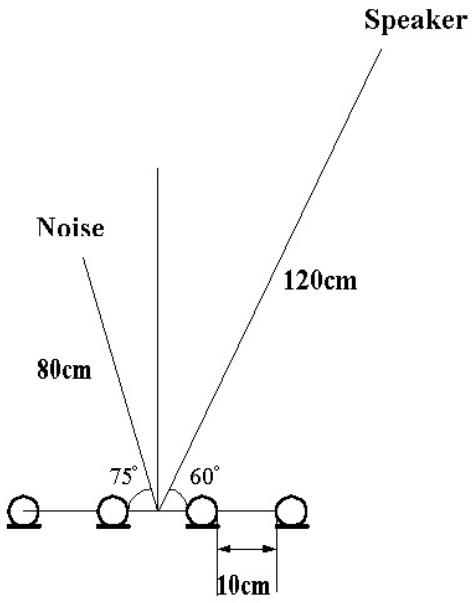
圖六、麥克風陣列錄音設備連接圖

3.2 語料庫

我們使用的語音特徵參數為 12 階 MFCC 和 12 階 delta MFCC 和 1 階 delta log energy 和 1 階 delta delta log Energy 共 26 階。訓練語料是在一般辦公室環境下使用近距離麥克風所錄製的，共有 1400 句中文連續數字，其中包含 70 位男生和 70 位女生。語音模型是使用隱藏式馬可夫模型來表示，每一個中文數字使用 7 個狀態，背景噪音則使用 3 個狀態，分別表示音

檔前後的噪音和數字間的噪音，所以總共的狀態數目為 73 個。每一個狀態包含 4 個混合數，因此共有 292 個混合數。

測試語料是在實驗室中使用遠距離麥克風陣列所錄製，我們模擬了三種不同車速的路況，分別為 0 km/h、50 km/h 和 90 km/h。在 0 km/h 路況下不加任何噪音，而 50 km/h 和 90 km/h 路況則利用喇叭放出汽車於時速 50 km/h 和 90 km/h 時所錄下的噪音來模擬。錄音時語者距離麥克風中心約 120 公分和麥克風陣列的夾角約為 60 度，噪音源則擺放於距離麥克風中心約 80 公分處和麥克風的的夾角約 75 度，麥克風陣列是線性配置的，相鄰麥克風的間距為 10 公分。語者和麥克風陣列以及噪音的相對位置如圖七所示。總共有 15 人參與錄音，包含 12 位男生和 3 位女生，每一種路況有 30 句不同的中文連續數字。每種路況總共錄得 450 句音檔。連續語音辨認所使用的演算法為一階段演算法。實驗結果我們以數字錯誤率(Digit Error Rate)來表示。



圖七、錄音時麥克風陣列和語者以及噪音間的相對位置圖

3.3 Delay-and-Sum Beamformer實驗結果

對於每一個麥克風(Mic1, Mic2, Mic3, Mic4)所收集的語音訊號其個別的字元錯誤率以及錯誤率的平均值如表一所示，此結果可視為基本系統(Baseline)的錯誤率。在此我們使用所有麥克風辨認錯誤率的平均值作為麥克風陣列的整體錯誤率。

我們所進行的第一組實驗是事先預設一些聲音源角度值來進行實驗以找出最佳效果的角度，這裡我們將聲音源的方向固定從 30° 到 150° 每間隔 30° 做一次實驗，再對經由

Delay-and-Sum Beamformer 處理後的語音訊號分別計算其辨認結果，辨認結果如表二所示。觀察實驗結果我們可以發現在不同路況下最佳辨認率都出現在60°，此一結果和我們實際上錄製語音時的方向是十分吻合的。

麥克風／Digit Error Rate (%)／路況	0 km/h	50 km/h	90 km/h
Mic 1	47.0	52.1	55.1
Mic 2	42.0	48.3	53.4
Mic 3	51.0	54.8	58.7
Mic 4	46.5	53.0	57.2
Mic 平均	**46.6**	**52.1**	**56.1**

表一、基本系統的辨認結果

路況 / Digit Error Rate (%) / 角度	30°	60°	90°	120°	150°
0 km/h	35.2	**31.9**	60.6	58.6	55.4
50 km/h	40.9	**38.1**	66.9	68.3	62.6
90 km/h	42.8	**40.4**	70.0	69.6	65.7

表二、不同聲音源角度下Delay-and-Sum Beamformer的辨認結果

演算法／Digit Error Rate (%)／路況	0 km/h	50 km/h	90 km/h
Mic 平均	46.7	52.1	56.1
固定角度 60°	**31.9**	**38.1**	**40.4**
SLA	43.8	48.6	52.1
TDCC H	37.5	43.6	47.0
TDCC A	**31.7**	**38.4**	**40.9**

表三、SLA和TDCC辨認結果比較圖

第二組實驗是將所錄得的測試語料分別經由語者定位演算法和 TDCC 求取不同麥克風間的時間延遲，經過補償後產生加強過後的語音訊號，再分別進行辨認，實驗結果如表三所示。其中 TDCC 有兩種不同的計算方法，TDCC H 表示每一個語句僅使用最高能量的音框來計算時間延遲，而 TDCC A 則表示每一個語句的所有音框皆被考慮來計算時間延遲。此外表三亦列出麥克風陣列的平均辨識率和固定角度60°時的辨認錯誤率以方便比較。

觀察表三的辨認結果比較，我們可以發現使用 Delay-and-Sum Beamformer 的 SLA 和 TDCC 確實能夠有效降低錯誤率，而 TDCC 和傳統的語者定位演算法 SLA 相比，TDCC 更能有效降低辨認錯誤率，且 TDCC 使用所有的音框的效果不但比使用一個音框效果還好，而且

幾乎和固定角度60°的錯誤率相同，顯示 TDCC 對於計算時間延遲是十分有效的。

3.4 取樣頻率對SLA和TDCC的影響

經由以上的實驗結果，我們發現語者定位演算法的效能較差。仔細研究其原因後發現，語者定位演算法是先求出語者的方向θ_s，再利用公式

$$b = (\text{Sampling Rate}) \cdot \tau = \frac{(\text{Sampling Rate}) \cdot d \cdot \cos\theta_s}{C} \tag{19}$$

來求出取樣點上的位移b。因為聲音的速度C和麥克風的間距d都是固定的，因此我們設計了一些實驗來瞭解取樣頻率對 SLA 和 TDCC 兩種演算法的影響。

實驗時我們先對測試語料利用內差法提高取樣頻率，經由 Delay-and-Sum Beamformer 求出增強過的語音訊號後，再將取樣頻率降為 8KHz。然後再進行辨識，辨認結果如表四(8KHz)、表五(16KHz)和表六(24KHz)所示。

我們可以發現 SLA 的辨認錯誤率有明顯的改變，取樣頻率由 8KHz 提高至 16KHz 和 24KHz 時錯誤率在 3 種不同路況都有顯著的下降。而提高至 16KHz 和提高至 24KHz 相比時則在 3 種不同路況上錯誤率僅有些許的改變。至於 TDCC 則因為其運算的對象就是時域上的取樣點，因此辨認錯誤率並無明顯改變。此一結果顯示由於 TDCC 不需要計算語者方向，因此可以適用於各種取樣頻率，能維持一定的辨認效果。

演算法／Digit Error Rate (%)／路況	0 km/h	50 km/h	90 km/h
SLA H	43.79	48.64	52.12
TDCC H	37.50	43.58	47.03

表四、取樣頻率 8KHz 時 SLA 和 TDCC 的辨認結果

演算法／Digit Error Rate (%)／路況	0 km/h	50 km/h	90 km/h
SLA H	34.94	39.97	42.00
TDCC H	36.90	42.69	47.09

表五、取樣頻率 16KHz 時 SLA 和 TDCC 的辨認結果

演算法／Digit Error Rate (%)／路況	0 km/h	50 km/h	90 km/h
SLA H	34.17	39.52	42.79
TDCC H	38.27	44.08	49.07

表六、取樣頻率 24KHz 時 SLA 和 TDCC 的辨認結果

3.5 加入語音模型調整的實驗結果

接下來的實驗著重於瞭解語音模型調整對麥克風陣列語音辨認系統的影響。基本系統經由 MLLR 調整後的實驗結果如表七所示。此外，SLA 和 TDCC 分別加上 MLLR 的實驗結果如表八所示。

麥克風／Digit Error Rate (%)／路況	0 km/h	50 km/h	90 km/h
Mic 1 + MLLR	29.0	31.7	33.4
Mic 2 + MLLR	25.5	29.9	33.1
Mic 3 + MLLR	31.6	33.4	36.2
Mic 4 + MLLR	28.3	31.4	34.5
(Mic + MLLR)平均	**28.6**	**31.6**	**34.4**
Mic 平均	46.6	52.1	56.1

表七、基本系統經由MLLR調整後的實驗結果

演算法／Digit Error Rate (%)／路況	0 km/h	50 km/h	90 km/h
(Mic + MLLR)平均	28.6	31.6	34.4
SLA + MLLR	29.9	31.5	35.1
TDCC H + MLLR	25.2	28.8	31.4
TDCC A + MLLR	**21.1**	**24.9**	**28.2**

表八、SLA和TDCC經由MLLR調整後的辨認結果比較圖

經由表七的實驗結果我們可以發現，基本系統使用 MLLR 的語音模型調整技術後，不管在哪一種路況都可有效的降低辨認錯誤率(0 km/h 由 46.64%降至 28.61%，50 km/h 由 52.07%降至 31.60%，90 km/h 由 56.12%降至 34.42%)，其原因為使用傳統麥克風於乾淨環境下所錄製的訓練語料和利用麥克風陣列於噪音環境下所錄製的測試語料間的不匹配現象相當嚴重。

分析表八的結果，我們發現加入語音模型調整的 SLA 和 TDCC 在降低辨認錯誤率上亦有顯著的效果，在三種不同路況上 TDCC 的效能仍然優於 SLA。最低的辨認錯誤率(21.10%)為路況 0 km/h 下使用全部音框來計算的 TDCC 演算法。

3.6 辨認時間的比較

辨認時間的計算是統計所有測試語料(共 1350 句，平均一句包含 6 個中文連續數字)經由時間延遲的計算、Delay-and-Sum Beamformer 的處理、語音特徵參數的求取和語音辨認的所有時間再做平均而得，實驗結果如表九所示。基本系統則僅計算特徵參數和語音辨認的時間

再平均。執行測試的電腦配備為 Pentium II 350 處理器和 128MB 記憶體的個人電腦，作業系統則為 Windows 98。觀察實驗結果，我們發現不管是僅使用最大能量的音框或是全部音框的 TDCC 演算法在執行速度上皆優於傳統的 SLA 演算法。

	Baseline	SLA	TDCC H	TDCC A
Without MLLR	0.28	0.58	0.41	0.56
With MLLR	0.50	0.79	0.63	0.77

表九、SLA 與 TDCC 執行速度之比較 (速度計算單位為秒／句)

4. 結論

本論文中我們建立一個應用麥克風陣列的語音辨認系統，此一系統利用 Delay-and-Sum Beamformer 來降低環境噪音對於語音訊號的影響。同時我們也提出了一個應用於麥克風陣列上計算時間延遲的演算法 TDCC。實驗的部分我們進行了基本系統的實驗、給定各種不同角度的實驗、取樣頻率改變的實驗、使用 SLA 演算法、使用最大能量音框的 TDCC 演算法和使用全部音框的 TDCC 演算法以及執行速度比較。經由實驗結果我們可以證明 TDCC 的有效性(在不同路況下平均約可降低 15%的辨認錯誤率)。和傳統的語者定位演算法 SLA 相比較，TDCC 不論是在辨認錯誤率降低的幅度上或執行速度上皆優於 SLA。

本論文中亦結合了語音模型調整的技術。經由實驗我們可以發現，單純只使用 MLLR 來調整語音模型即可獲得不錯的效果。然而若將麥克風陣列和語音模型調整的技術相結合，對於降低辨認錯誤率(在不同路況下平均約可降低 25%的辨認錯誤率)會產生更顯著的效果。從我們研究的結果，可以發現仍然還有許多值得研究的課題，如更精確語者方向的定位、麥克風位置的考量…等。未來我們將主要致力於研究麥克風陣列中麥克風的擺放位置和辨認率間的關係，以及實際將麥克風陣列的演算法應用在汽車環境或有回音、噪音的語音辨識系統上。

5. 參考文獻

[1] A. P. Dempster and N. M. Laird, D. B. Rubin. “Maximum Likelihood from Incomplete Data via the EM Algorithm”, *J. Roy. Stat. Soc.*, 39(1) : 1-38, 1977.

[2] J.-L. Gauvain and C.-H. Lee, “Maximum a Posterior Estimation for Multivariate Gaussian Mixture Observations of Markov Chains”, *IEEE Trans. Speech, Audio Processing*, Volume 2,

pages 291-298, April 1994.

[3] D. Giuliani, M. Omologo and P. Svaizer, “Experiments of Speech Recognition In a Noisy and Reverberant Environment Using a Microphone Array and HMM Adaptation”, In Proc. of ICSLP ’96, pages 1329-1332, October 1996.

[4] M. Inoue, S. NAKAMURA, T. YAMADA and K. SHIKANO, “Microphone Array Design Measures for Hands-Free Speech Recognition”, In Proc. of Eurospeech ’97, Volume 1, pages 331-334, September 1997.

[5] C. J. Leggetter and P. C. Woodland, “Maximun Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models”, Computer Speech and Language, Volume 9, pages 171-185, September 1995.

[6] D. Mahmoudi, “Combined Wiener and Coherence Filtering in Wavelet Domain For Microphone Array Speech Enhancement”, In Proc. of ICASSP ’98, pages 385-388, May 1998.

[7] M. Omologo and P. Svaizer, “Acoustic Event Localization Using a Crosspower-Spectrum Phase Based Technique”, In Proc. of ICASSP ’94, Volume 2, pages 273-276, 1994.

[8] M. Omologo and P. Svaizer, “Acoustic Source Location in Noisy and Reverberant Environment Using CSP Analysis”, In Proc. of ICASSP ’96, pages 921-924, 1996.

[9] T. YAMADA, S. Nakamura and K. Shikano, “Robust Speech Recognition with Speaker Localization by a Microphone Array”, In Proc. of ICSLP ’96, pages 1317-1320, October 1996.

[10] T. YAMADA, S. Nakamura and K. Shikano, “Hands-Free Speech Recognition Based on a 3-D Viterbi Search Using a Microphone Array”, In Proc. of ICASSP ’98, pages 245-248, May 1998a.

[11] T. YAMADA, S. Nakamura and K. Shikano, “An Effect of Adaptive Beamforming on 3-D Viterbi Search”, In Proc. of ICSLP ’98, pages 381-384, December 1998b.

[12] T. YAMADA, S. Nakamura and K. Shikano, “Simultaneous Recognition of Multiple Sound Sources Based on 3-D N-Best Search Using Microphone Array”. In Proc. of Eurospeech ’99, Volume 1, Page 69-72, September 1999.

PC-Based 臺灣手語轉語音溝通輔助系統

邱毓賢、吳宗憲、郭啓祥、*鍾高基

國立成功大學資訊工程研究所、*醫學工程研究所

Page 223 ~ 242

Proceedings of Research on Computational Linguistics Conference XIII (ROCLING XIII)

Taipei, Taiwan

2000-08-24/2000-08-25

PC-Based 台灣手語轉語音溝通輔助系統

邱毓賢、吳宗憲、郭啟祥、*鍾高基

國立成功大學資訊工程研究所、*醫學工程研究所

Email：p7888107@ccmail.ncku.edu.tw, chwu@csie.ncku.edu.tw

Fax：+886-6-274-7076

摘　　要

聲音或語言機能喪失的聽語障礙者，常常發生難以與一般人正常溝通或溝通時發生明顯的障礙。本研究乃考量本土聽語障礙族群實際溝通輔助的需求，研發符合本土化 PC-based 台灣手語轉語音溝通輔助系統，包括 1).手語鍵盤，依據 Row-Column Scanning 及考量認知、注意集中及學習反應之階層式安置（Hierarchical Arrangement）的策略，以作為操作輸入媒介；2).運用詞頻預測、詞性篩選、句型預測及注音縮寫模組等機制來輔助關鍵詞彙輸入與手語符號搜尋；3).結合句型樣版及概念從屬之語格文法來建立關鍵詞彙預測完整文句之轉譯系統。在系統功能性評估部分，由特教老師選取日常生活 1000 句對話語料(平均長度為 4.9 字/句)。免除虛詞輸入可節省 26.25%按鍵數；加入詞彙、句型預測及注音縮寫等輔助構句方式，與未加任何預測功能之檢索速度改善率分別為 67.71%、79.50%、96.87%。在適用性評估部分，經由教學、調適及評估時期的訓練，構句成功率分別為 47.37%、65.0%、68.38%；構句速度與主觀滿意度評量亦有顯著改善。因此，未來可提供符合本土所需之輔助手語訓練與教學系統。

關鍵字：聽語障礙、溝通輔助系統、手語鍵盤、關鍵詞彙預測、句型樣板

1.　緒論

聽語障礙指聲音或語言功能性的損傷，因而造成難以與一般人正常溝通，或溝通時發生明顯的障礙，且由於溝通障礙者在溝通問題上呈現很大的個別差異，有時又難以鑑別，在實際生活中，難以使用聽語功能表達基本生理需求；在求學階段中，也造成許多的學習障礙。

歐美先進國家在 1970 年代開始研發可提供語言學習及溝通替代殘障輔助復健科技與輔具(Augmentative and Alternative Communication, AAC)，主要發展與改良簡易型溝通板、電腦操作輸入介面及輔助性周邊裝置。1980 年代，由於電腦、語音訊號處理及殘障輔助科技的發展，全力整合工程、復健、醫療及教育訓練來改善聽語障礙者日常生活的功能性。1990 年代，則著重應用先前輔助科技與輔具提供之經驗於教育訓練與臨床運用[Reichle, 1991; Webster, et.al., 1985; David and Mirendan, 1992]。

反觀台灣，現階段資訊相關技術已相當成熟，卻未能妥善應用於特殊教育與殘障溝通輔具的研發，主要原因乃是西方的語言/語音特性與台灣本土日常生活所慣用的中文全然不同，使得歐美國家所發展的先進語音科技無法直接移轉為國人所使用[古鴻炎、許文龍，民 85 年; 吳宗憲、陳昭宏、林超群，民 85 年]。大多數聽語障礙者最為常用的溝通方式以手語為主，但是由於聽人不懂手語語言，所以不瞭解聽語障礙者所表達的內容，且大部分聽語障礙者聽不到一般自然的口語，而無法進行有效的溝通。

台灣的手語分為中文文法式手語及自然手語[史文漢，民 89 年]。符合中文文法結構的手語，主要應用於口語溝通、中文教學及會議；自然手語則以其特殊的文法結構獨樹一幟，主要為聾人之間的手語溝通方式，且無一定的規範。其中手語語言結構為影響手語轉譯的主要問題，包括：1).**詞序**：手語的用法習慣、表示集、詞序沒有一定的規律，往往隨著環境、地區性、年齡、教育程度而異。一般常見的結構有 SVO（主體－動詞－客體）、SOV（主體－客體－動詞）及 OSV（客體－主體－動詞）；2).**同時性**：指主客體需依據動詞特性同時存在手勢符號序列中，否則難以表示手語的意圖，因此會產生類似詞序的問題；3).**單位詞**：中文的單位詞形形色色、變化萬千，手語則因受限於僅能表達簡單的單位詞或因語言慣用規則不同，而將單位詞省略。

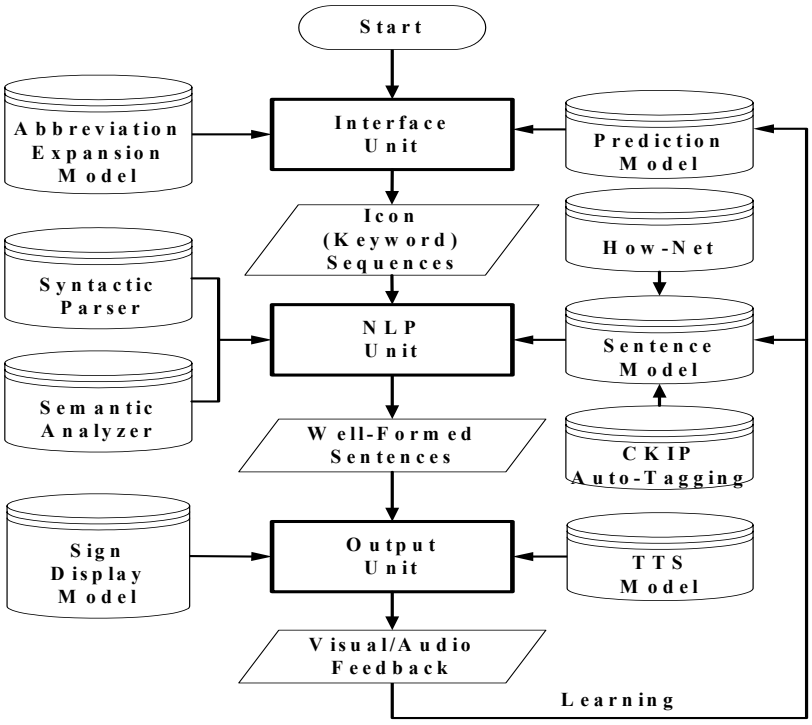
目前歐美國家相關溝通輔助科技的研究，主要朝向 1.)針對嚴重肢體功能性障礙族群發展高操作效率之虛擬鍵盤（Virtual Keyboard）[Reichle, 1991; Simpson and Koester, 1999; Webster, et. al., 1985; David and Mirendan, 1992]，透過前詞預測後詞(Word Prediction)[Koester and Levine, 1994; Hunnicut, 1990]、英文詞彙簡稱（Abbreviation Expansion）[Vanderheiden, 1984]、語意編碼（Semantic Coding, e.g. Consumer Product- Minspeak System）[Chang, 1992; Baker, 1982]，其主要目的乃輔助大量詞彙庫中選取特定詞彙，然而其設計之根本以詞彙語意為基礎，使用者必須記憶大量的詞彙及其對應符號，造成認知上的負擔；2.）由北美復健工程協會(RESNA)所提出之 Sentence Compansion(***comp***ress/ex***pansion***)的觀念[Demasco, et. al., 1989; Demasco and McCoy, 1992]，使用者輸入少量詞彙資訊(uninflected content words)，乃根據英文文法結構與特性的分析，擴展出符合語法/語意的簡短語句(well-formed sentence)。但是由於西方語言、語音、手語的特性(如美國手語 American Sign Language/ASL)與複雜的中文及台灣手語全然不同，所研發的輔助性工具無法直接移轉為國人使用，因此，發展符合本土化溝通輔助系統實乃刻不容緩。

2. 研究目的與重要性

本研究目的為應用自然語言處理、中文語音訊號處理科技及本土聽語障礙族群實際溝通輔助需求之考量，系統化研發 PC-based 擴大及替代式本土化溝通輔助科技系統 – 台灣手語轉語音溝通輔具，透過簡易及人性化的操作介面，以提供聽語障礙者在就醫、就養、就學、就業等不同需求的溝通輔具，並改善其日常生活中的溝通表達。其**特定目標**為：1).***針對聽障學童日常生活/學校溝通輔助的實際功能需求***，發展及建立本土化口語/語音資料庫、手語符號階層式認知資料庫及相關教材之彙整；2).***考量聽障學童視聽覺回饋之感覺認知發展***，設計合適之參數可調式人機溝通界面，以符合個別化學習之反應；3).***完全根據本土化的中文語言/台灣手語背景來研發，系統化的建立 PC-based 台灣手語轉語音溝通輔助系統(AAC)***，解決了手語轉譯之斷詞、詞序、補綴等中文文句生成的問題，且可提供符合中文文法式之手語輔助教學與訓練之用。**本研究之重要性**為鑑於台灣目前缺乏適當的本土化聽語障礙者溝通輔助及訓練系統，希望藉由本研究蒐集電腦輔助相關教材、特殊教育語言教材與國語母語教材，作一整理分析，以建立一貼切且有效之訓練教材，並且利用現今之電腦科技，研發一語音多媒體電腦輔助訓練系統，協助進行溝通訓練與語言學習的活動。另外在聽語障礙者輔具上，我們將針對國人之語音特性，收集並建立一資料庫，並對其作分析，以便建立一方便實用之本土化語音合成溝通輔具，帶給聽語障礙者一個方便有尊嚴之生活環境。

3. 系統設計與發展

本研究所發展的「關鍵詞彙預測完整語句之台灣手語溝通輔具轉譯系統」，如圖一所示:



圖一 系統架構圖

主要包括 1).**手語符號鍵盤模組**，採用分頁及分列檢索的模式，結合由上而下、由左而右的檢視策略，並結合詞彙序列動態配置手語符號鍵盤上的手語符號；2).**手語詞彙預測機制**，透過詞頻預測、詞性篩選、句型預測、注音縮寫查詢等機制，提供快速手語符號檢索；3).**文句生成模組**：關鍵詞彙預測完整文句核心，依據 Sentence Compansion 及 Bottom-Up Parsing with Top-Down Filtering 的觀念，以中研院自動斷詞程式及知網（How-Net）為基礎，句型樣版及概念從屬之語格文法為構句基本架構，透過語法剖析、語意分析以及虛詞補綴（介詞、副詞、語助詞、連接詞）等自然語言理解理論/技術的處理，建立此手語符號/詞彙預測完整文句之轉譯系統，並輸出符合中文文法的手語符號序列及合成的中文語音，以達成溝通表達的目標。

3-1. 手語鍵盤模組之建立

手語鍵盤為使用者與輔具溝通的橋樑，基於運用視覺化語言的呈現模式以及聽語障族群實際溝通的訴求，本研究以聽語障礙者所熟悉的手語作為輔具溝通媒介，透過智慧型動態配置之手語符號鍵盤的設計，讓使用者點選安置於手語鍵盤上的手語符號，作為關鍵詞彙輸入，同時配合詞彙分類、詞頻預測、詞性篩選以及注音縮寫查詢等方式輔助使用者於大量手語資料庫中選取所需之手語關鍵詞彙。

3-1-1. 手語鍵盤設計

本研究以聽語障礙者所熟悉的手語為基礎，設計一套智慧型之手語鍵盤，讓使用者透過點選動態安置於手語鍵盤上的手語圖像，作為關鍵詞彙輸入。手語鍵盤之設計採用分頁的方式，配合由左而右、由上而下的檢視策略，並依據詞彙序列動態配置手語圖案於鍵盤之上，以供使用者操作輸入手語關鍵詞彙。

手語詞彙分類，本研究透過 1.)**屬性分類**：即使用者聯想的詞彙所對應之屬性類別，以教育部手語畫冊為例，其根據手語屬性分為：人物、動物、植物、地名、時令、…等類別；2.）**詞性分類**：聯想的詞彙所對應之詞性類別，偏向語言學的分類方式，依使用者認知程度而定，例如單純的以動詞、名詞、形容詞為分類標準等。

高頻詞彙優先，乃應用統計式語言預測模式將最常出現的手語詞彙置前，讓使用者以最少的檢視次數選取詞彙。首先統計分析對話語料，依照詞頻將詞彙序列對應之手語圖案依序安置並呈現於手語鍵盤上，以加快使用者詞彙選取的速度。

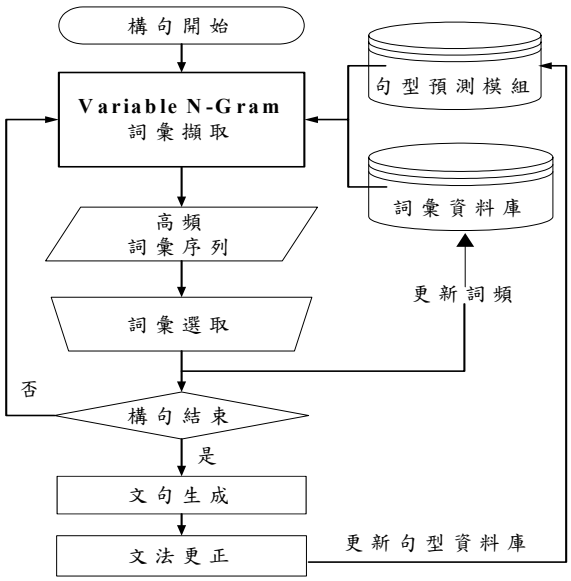
詞性篩選策略，經由實際觀察對話語料得知構句模式中同詞性的出現機率不高，因此，已使用過之同詞性的高頻詞彙，將不安置於手語鍵盤。

3-1-2. 句型預測模組

基於中文文法修正的考量，本模組係透過文句生成模組來輔助修正及預測文法，將符合之句型記錄於句型預測樹，其預測模式乃利用 Variable N-Gram 詞性模組預測下一個最有可能的詞性，同時結合高頻詞彙策略，將符合詞性的高頻詞彙置前。本研究所採用的 Variable N-Gram 詞性模組乃以**詞性**（POS）為基本單位（依中研院詞性分類定義），每一個內部節點皆代表一個詞性樣本，第 n 個節點出現的機率可用下列式子來描述：

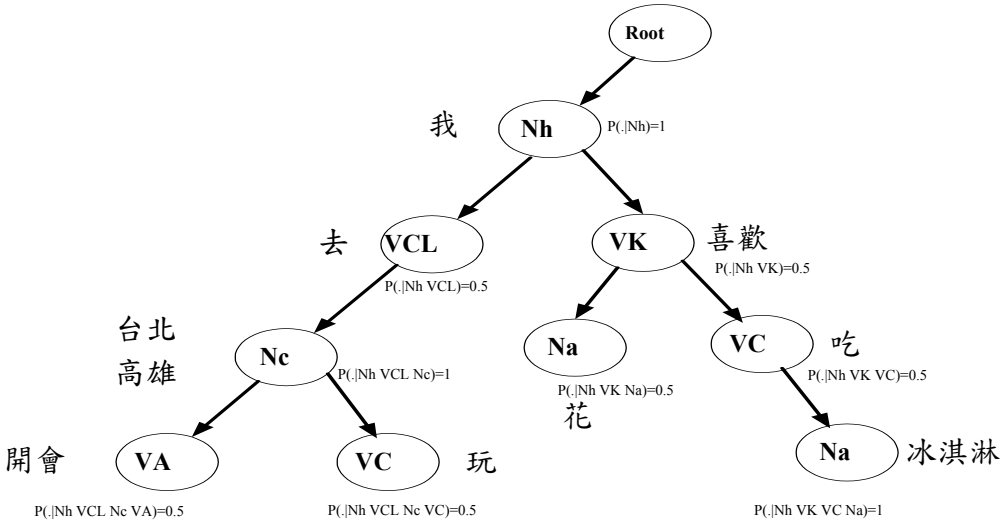
$$P(POS_n \mid POS_1 \cdots POS_{n-1}) = \frac{P(POS_1 \cdots POS_n)}{P(POS_1 \cdots POS_{n-1})} \quad (1)$$

其中，C（$POS_1 \cdots POS_n$）代表 $POS_1 \cdots POS_n$ 詞性串列出現的次數。其處理流程圖三所示



圖二 句型預測模組處理流程圖

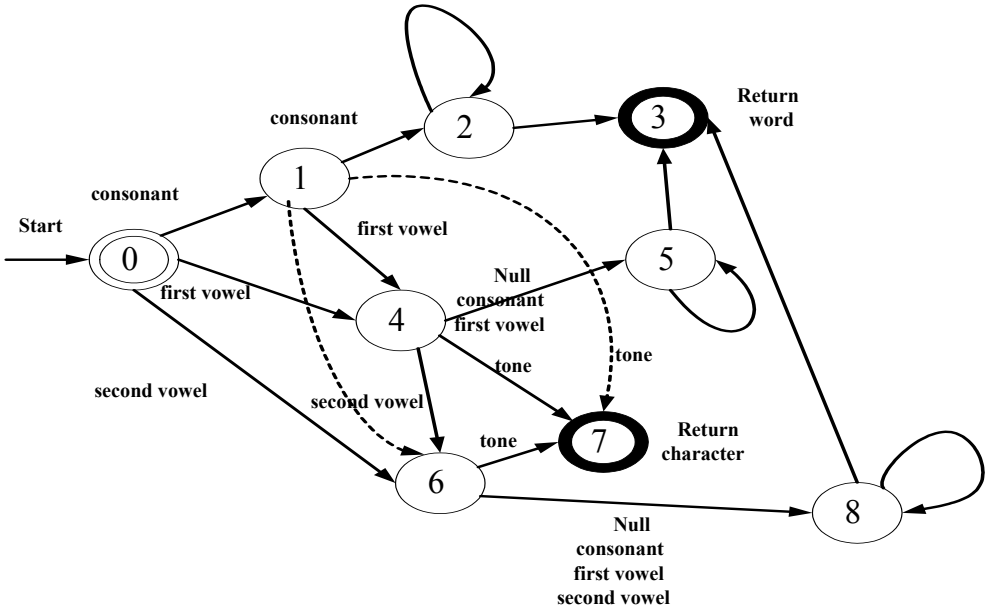
舉例來說，系統依據使用者之前的構句－『我去台北開會』、『我去高雄玩』、『我喜歡花』、『我喜歡吃冰淇淋』，建立了一個 Variable N-grams 詞性模組，如圖三所示：



圖三 Variable N-grams 詞性模組

3-1-3. 注音縮寫輔助查詢

採類似英文縮寫方式，透過完整注音或連續啟始音輸入，幫助具注音認知能力的使用者從龐大的手語資料庫中擷取所需的詞字彙。本研究乃根據中文文句與音韻特性，運用 **Augmented Transition Network** 設計架構(圖四)，以擷取對應的字/詞。建構規則為：1).連續兩個同類音相接，為詞；2).First Vowel 接 Consonant，為詞；3).Second Vowel 接 Consonant 或 First Vowel，為詞；4).無聲調（1 聲），可能為詞；5).包含 2~5 聲調，必為字。



圖四 注音縮寫模組之 ATN 架構圖

3-2. 文句生成模組之建立

針對台灣手語轉譯所遭遇之文法結構問題，提出以關鍵詞彙之句型樣版匹配方式來預測完整語句，其設計理念乃依據 Sentence Compansion 及 Bottom-Up Parsing/Top-Down filtering 的觀念；以中研院自動斷詞及知網（HowNet）知識為基礎，句型樣版及概念從屬之語格文法為構句基本架構，透過語法剖析、語意分析以及 Variable N-grams 補詞等自然語言處理技術，將使用者經由手語鍵盤模組輸入的關鍵詞彙轉為自然完整的語句。

3-2-1. 詞性分析及知網知識之應用

中研院自動斷詞程式，乃結合其斷詞系統與語法剖析器，針對語句做斷詞及詞性標記的工作。本研究所定義的關鍵詞為動詞（V 開頭的詞性）及名詞（N 開頭的詞性），其餘詞類則定義為虛詞。中研院自動斷詞已可達到 96%以上的準確度，但因本研究的詞彙單位為手語詞彙，基於斷詞標準一致性的考量，本研究採用長詞優先的斷詞原則，作為人工修正斷詞及詞性標記的依據。知網是設計智慧軟體的雙語常識知識庫，詳盡地描述概念定義以及屬性之間

的關係，知網的特徵分為兩層：主要特徵及次要特徵，前者為物件最重要特性，後者則描述物件之屬性。舉例來說：『弟弟』在知網裡的定義為【DEF=human|人,family|家,male|男】，其中『human|人』為弟弟的主要特徵，其餘則為次要特徵。利用物件的主要特徵，引用語格文法之概念從屬的觀念，不僅描述動詞與其他語格（keyword slot）的關係，且也用以尋找屬性相似語格及匹配節點，避免單純以詞性為考量所造成的缺失。

3-2-2. 句型樣版樹之建立

句型樣版樹以實際收集彙整之 1000 句日常生活對話語料庫為基礎（平均 4.9 字/句），利用中研院自動斷詞程式取得語句之斷詞結果、詞性（**POS**）及語法資訊，透過知網（**How-Net**）取得詞彙之概念從屬屬性後，接著經由觀察對話使用習慣及評估語法延伸之合理性，歸納句型延伸規則，遞迴建立結合語句、語法及語意資訊的句型樣版樹，以提供日後句型比對以及虛詞補綴之依據，以 Nh + Nd → Nd（省略 **Nh**）為例。透過句型延伸及結構化處理對話語料的方式，不僅可以彌補對話句型不足之缺失，更可縮減對話語料資料庫的儲存空間、動態更新/延伸對話句型及加快句型搜尋速度。

■ **句型樣版樹之節點資料結構**

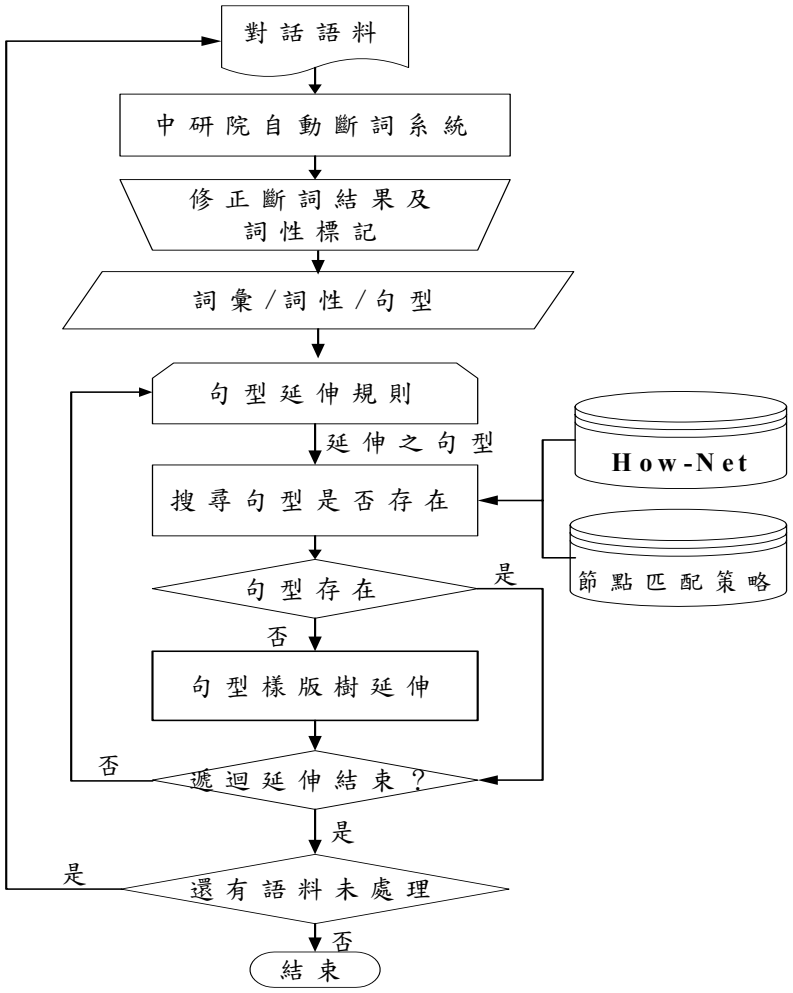
I. 起始節點（root），為句型起始位置，所有句型樣版皆由此開始延伸，有多組子節點。

II. 內部節點（internal node），包含 1.)屬性，同屬性的詞彙可同時存在於同一節點；2.）詞性，同詞性的詞彙可同時存在於同一節點；3.）參考次數，記錄節點對應的參考次數；4.）詞彙組，節點內可同時存在多組詞彙；5.）詞彙參考次數，節點內所有詞彙皆有其對應的參考次數；6.）父節點，只有唯一的父節點；7.）子節點，可有多組子節點。

III. 外部節點（external node、terminal node），為句型結束位置，包含 1.）參考次數，記錄句型樣版出現次數；2.）節點個數，句型的節點個數；3.）名詞個數，句型的名詞總數；4.）動詞個數，句型的動詞總數；5.）虛詞個數，句型的虛詞總數；6.）唯一的父節點。

■ **句型樣版樹之建構流程**，如圖五所示，其建構原則為：

1. 節點匹配嘗試：1).若屬性符合，則匹配之；2).若詞性符合，匹配之；3).節點匹配者，若詞彙已存在於節點之中，則詞彙參考次數加 1；否則，將新詞彙加入節點之中，新詞彙參考次數設為 1。
2. 廣度優先搜尋句型是否存在：1).若句型存在，句型樣版出現次數加 1；2).若句型不存在，則繼續延伸路徑直到句型生成為止；3).句型生成後，產生外部節點，句型樣版出現次數設為 1。目前本研究所建立之句型樣版樹包含 553 條句型路徑，平均路徑長度 3.5 個節點。



圖五 句型樣版樹建構流程圖

3-2-3. 文句生成機制之建立

以句型樣版樹為依據，經由以下步驟：1.）針對使用者輸入之關鍵詞串，根據剖析規則做片語合併及單位詞嵌入之處理，找出關鍵詞串對應的片語，作為節點匹配之基本單位；2.）依據語意分析及節點匹配策略將關鍵詞所組成的片語填入句型樣版的節點之中；接著依照節點屬性匹配程度以及句型樣版參考次數之統計資料，篩選出合適的句型樣版；3.）參照句型樣版，透過時間、地方片語之樣版嵌入以及 Variable N-Gram 虛詞補綴等處理，生成合乎語法及語意的自然完整語句。

- **節點匹配原則**　基於降低語法複雜度及縮減搜尋空間之考量，本研究透過片語合併及單位詞嵌入規則，先找出關鍵詞串對應的片語，節點匹配單位。處理原則為：1).片語合併：分別處理人、時、地、物等類別，人、物類別在語言學裡稱為 **argument**，時、地等類別稱為 **adjunct**；簡言之，時間及地方片語並非構句初期所需物件。因此，在節點匹配時可暫時忽略時間或地方片語，等到語句生成時，再利用片語嵌入的方式，將未作節點匹配的時間及地方片語併入句型樣版中，並同時繼承被修飾者的屬性及詞性。

如：甜/蘋果⇒甜的蘋果；2).單位詞嵌入：台灣手語並無單位詞，因此，採用單位詞對應表，並搭配語法規則，將單位詞嵌入名詞片語之中，如：二/鞋子⇒兩雙鞋子。

■ **句型樣版匹配**　　由於台灣手語文法輸入詞序的問題，本文以關鍵詞合併後的片語組為節點匹配單位，配合節點匹配策略將關鍵片語組填入句型樣版之中；接著參照節點屬性匹配程度以及句型樣版參考次數之統計資料，篩選出合適的句型樣版，以提供自然語句生成之依據。其處理策略為：1).節點匹配策略：透過屬性或詞性比對，將所有未匹配的關鍵片語與句型樣版的節點作一匹配嘗試。本研究以中研院簡化的 46 類詞類作為節點之詞性標記，其動詞詞類分為 12 類（VA～VL），其詳盡的動詞分類固然有助於降低語法的不明確性，卻也造成句型侷限以及句型匹配不易之缺失。例如，若單純以詞性為匹配依據，則『愛（VL）』與『喜歡（VK）』詞性不同；若放寬動詞詞性限制，其兩者皆屬於狀態及物類動詞。因此，本研究將節點匹配條件適度放寬，若屬性相符，則匹配之；否則，若名詞詞性相符，匹配之；若動詞詞性相似，匹配之。2).句型樣版匹配：以句型樣版樹為依據，配合節點匹配策略及比對程序，將關鍵片語組填入合適的句型樣版。基於虛詞補綴之需求及縮減搜尋空間之考量，其條件限制為：樣版動詞個數等於關鍵片語動詞個數、樣版名詞個數小於關鍵片語名詞個數、樣版虛詞個數小於 2。其中，關鍵片語個數係指使用者輸入之關鍵詞彙合併後的片語個數。依據以上限制，從句型樣版樹中挑選出條件符合的句型樣版之後，開始作句型樣版比對，其比對程序如下：

i. 樣版匹配開始，將所有關鍵片語設為未匹配。
ii. 由 root 節點開始，配合節點匹配策略，檢查所有未匹配的關鍵片語中，是否有符合其子節點匹配條件。
iii. 若有符合之片語，則將片語填入其子節點之中，回到步驟 ii.繼續下一節點之匹配。
iv. 若無符合之片語，則檢查其子節點是否為關鍵節點（詞性為 V 或 N 的節點），如果不是關鍵節點，則跳過此子節點，回到步驟 ii.繼續下一節點之匹配；否則，中斷此句型樣版之比對。
v. 完成比對之條件為：所有關鍵片語（時間、地方片語除外），皆有匹配之節點；最後節點的子節點為外部節點（terminal node）。

■ **句型樣版計分處理與篩選**　　由於節點匹配策略將動詞詞性限制放寬，雖增加句型成功匹配的機率；但放寬限制亦增加錯誤匹配的可能性。因此，本研究提出初步的統計計分機制，透過節點屬性相似度及句型樣版參考次數之統計資料，作為候選句型樣版

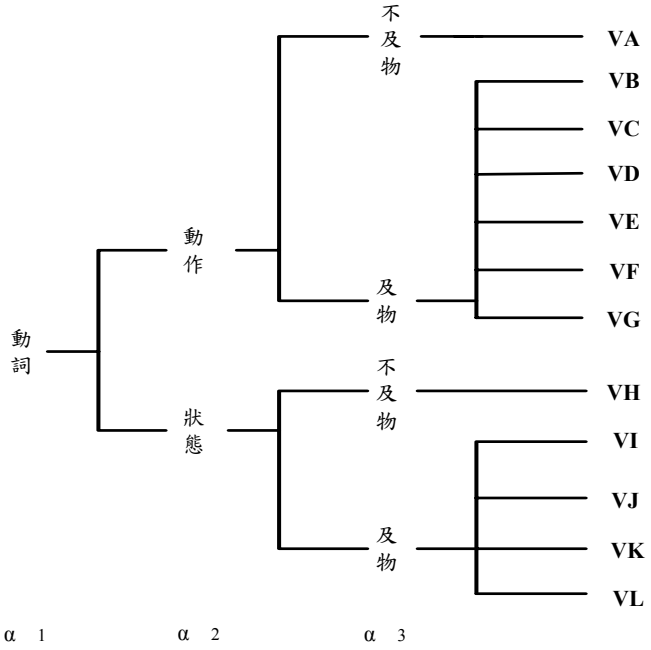
排名及篩選的評估，計分公式為:

$$SentencePatternScore = \frac{\left(\sum_{i=1}^{n} SN(a_i, b_j)\right) * f}{n} \quad (2)$$

其中 a_i 為第 i 個節點；b_j 為與節點 a_i 匹配之關鍵詞彙；n 為句型樣版節點個數；f：句型樣版的出現機率（即句型樣版終端節點之參考次數）；$SN(a_i, b_j)$為節點相似度，其配分值為：

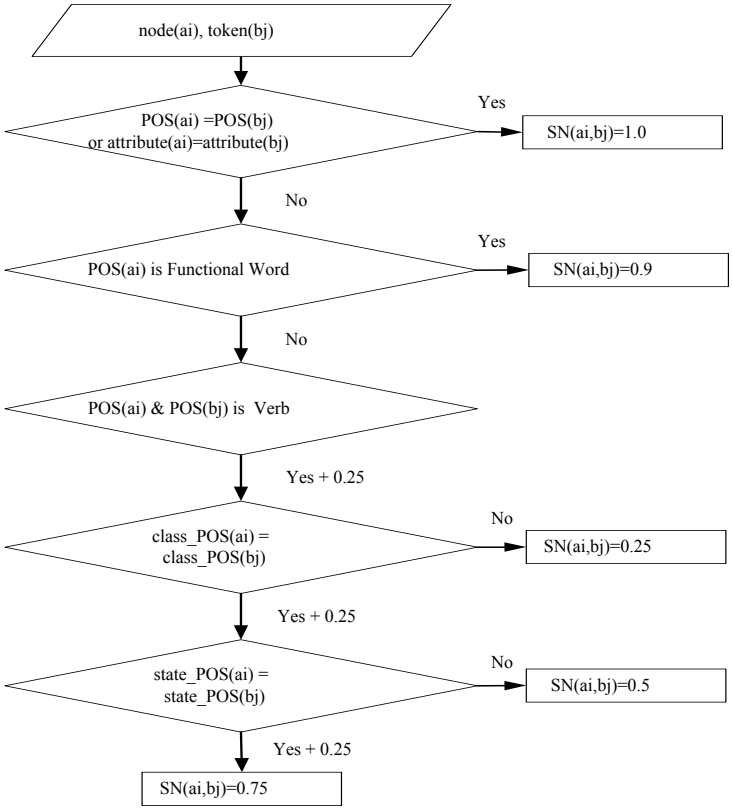
$$SN(a_i, b_j) = \begin{cases} 1 & if\ pos(a_i) = pos(b_j)\ or\ attribute\ (a_i) = attribute\ (b_j) \\ 0.9 & if\ pos(a_i)\ is\ functional\ \ word \\ 0.25 & if\ class\ \_\ pos(a_i) \neq clsss\ \_\ pos(b_j) \\ 0.5 & if\ state\ \_\ pos(a_i) \neq state\ \_\ pos(b_j) \\ 0.75 & otherwise \end{cases} \quad (3)$$

節點相似度係根據 1.）中研院動詞結構（如圖六所示，其參數值乃參照詞性階層式屬性分類的限制程度而定），動詞詞性相似度依關鍵詞與節點屬性關聯程度而異，達到α_1層級為 0.25 分，α_2層級為 0.5 分，α_3層級為 0.75 分；



圖六 中研院動詞分類樹狀結構圖

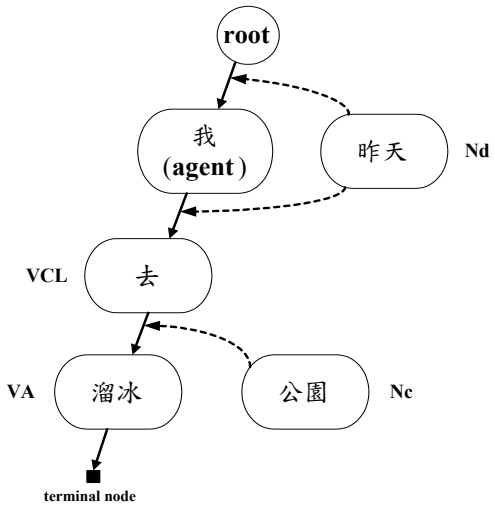
2.）結合虛詞（functional word）省略策略，將無關鍵詞匹配的虛詞節點訂為 0.9 分；3.)與關鍵詞屬性相符或詞性相同的節點訂為 1.0 分，如圖七所示，其中 node(a)代表句型樣版的節點，token(b)代表與該節點匹配之關鍵詞，依據節點匹配策略：1.）首先，檢查關鍵詞 b 與節點 a 之屬性或詞性是否相符；2.）否則，檢查節點 a 是否為虛詞；3.）否則，進行關鍵詞 b 與節點 a 之動詞相似度比對。



圖七 節點相似度比對流程圖

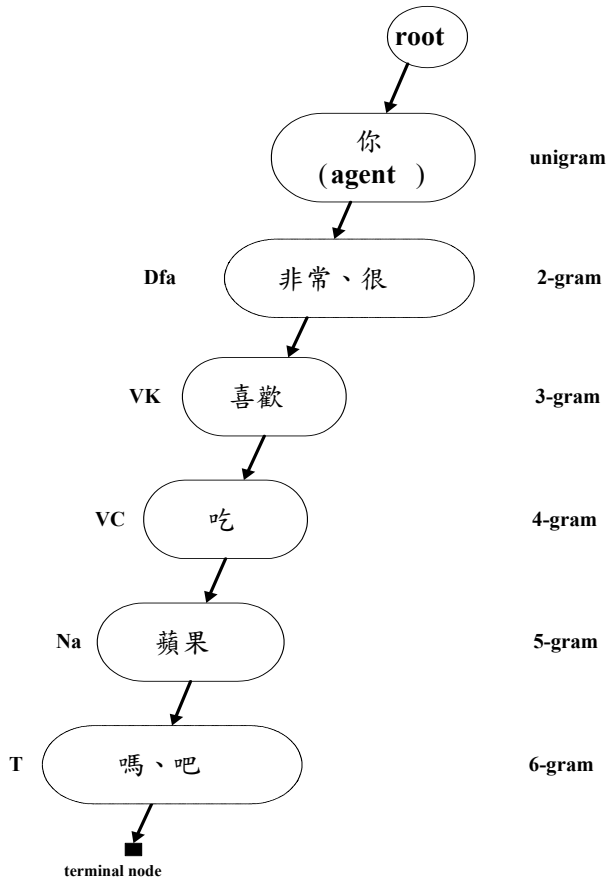
■ **自然語句生成**　以篩選出的候選句型樣版為依據，利用片語嵌入規則，將之前未做節點匹配的時間及地方片語嵌入句型樣本，生成合乎語法及語意的語句，並透過 Variable N-Gram 語言模型虛詞補綴的技術，讓生成之語句更加自然完整。

Step1.）片語嵌入：根據所歸納之時間及地方片語的嵌入規則為：時間片語乃置於行為者（agent）之前或之後，否則，置於句首；地方片語乃置於動作類動詞之前，否則，置於句尾，如圖八所示。



圖八　時間及地方片語之嵌入

Step2.) 補虛詞（**Functional Word Addition**）：依據『關鍵詞彙預測完整語句』之設計理念及生成合乎語法及語意之語句的基本訴求，本研究利用句型樣板之 Variable N-Gram 資訊，針對虛詞節點做適當之補綴，讓生成之語句更加自然完整，同時亦可達到免除虛詞輸入，加快構句速度，如圖九所示。



圖九　Variable N-grams 補詞範例

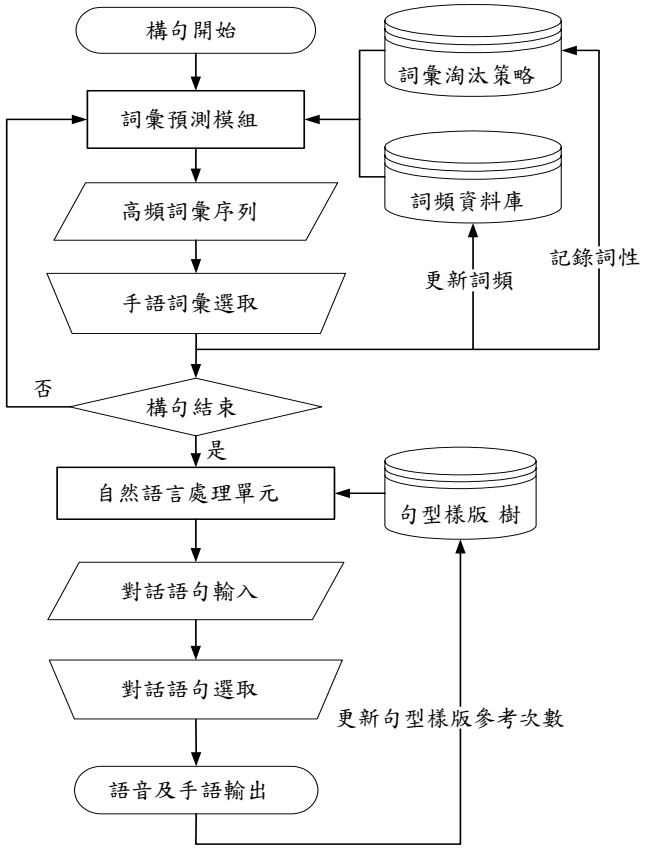
3-2-4. 自動學習機制

基於為使用者量身訂製輔具的訴求，本研究自動記錄使用者構句時常用的手語詞彙及句型樣版，引用統計學習機制的觀念來動態更新詞頻資料庫及句型樣版樹相關的節點資訊，將常用的詞彙及句型置前，以降低手語詞彙檢索次數，加快構句速度；同時透過詞彙淘汰策略，將較少使用的詞彙從手語鍵盤之詞彙序列中移除，以降低搜尋空間，加快手語詞彙選取，系統自動學習流程如圖十所示。

- 動態更新　　本研究依據對話語料庫統計手語詞彙出現頻率後，建立預設之詞頻資料庫，提供詞彙預測之用。基於個別化設計理念之考量，輔具必須隨著個人的使用習慣而調適，因此，本研究根據使用者實際構句所使用的手語詞彙及句型樣版，動態更新詞頻資料庫及句型樣版樹資訊，將常用的手語詞彙及句型置前，加快使用者操作選

擇。其功能為：1).詞頻更新模式，其原則為所選取的手語詞彙詞頻加 1、與所選取之手語詞彙具相同詞性之手語詞彙詞頻減 1；2).句型樣版次數更新，使用的句型樣版參考次數加 1，配合句型樣版計分方式，將常用的句型提前。

- 詞彙淘汰策略　　較少使用的手語詞彙，應適時地從高詞彙序列中移除，以減少手語詞彙的搜尋空間。本研究配合詞頻更新方式，將出現頻率低於 1 次的詞彙從手語鍵盤之詞彙序列中移除，縮減詞彙檢索空間，加快詞彙搜尋速度。



圖十 自動學習機制架構圖

3-3. 視聽覺回饋輸出模組

將文句生成模組所產生的中文文法語句，透過視聽覺回饋之手語(**visual**)及語音(**auditory**)互動式顯示呈現給使用者及其溝通對象，以達到生活溝通及手語學習之目的。基於多樣化口語語音溝通的需求，本研究透過語音合成參數的調整，讓聽人『聽到』不同聽語障礙者的聲音；透過靜動態手語圖案呈現方式，讓聽語障礙者『看到』聽人的言語內容。

3-3-1. 手語教學模組與靜動態顯示策略

基於手語視覺回饋之考量，本研究透過文字轉手語模組，將文字形式的對話語句轉換成圖形介面的中文文法手語，回饋給使用者，其特點為：1).文法修正處理策略：本研究透過句

型樣版將關鍵詞串轉為合乎語法及語意的完整語句，不僅修正自然手語文法結構的問題，其對應的中文文法結構更可提供使用者學習中文文法之用；2).手語學習理念：對話語句經由文字轉手語模組轉換後，所對應之合乎中文文法的手語序列，即代表對話語句的手語表達方式，因此，對於不懂手語的聽人而言，亦是一種手語學習的途徑。

3-3-2. 文字轉手語模組

文字轉手語模組以手語詞庫為斷詞依據，透過文字斷詞後，找出詞彙對應的手語圖案，並依序將其安置於手語鍵盤；若無對應的手語圖案，則以中文字代替顯示。本研究採用的斷詞原則為 1.)長詞優先；2.）左詞優先。其功能為：1).靜動態顯示策略：本研究基於教學觀點之考量，除了靜態顯示方式之外，亦配合特教老師意見，採用動態顯示手語圖像的方式，並經由顯示速度的調整觀察學同學反應的行為，其特點為吸引使用者的注意力、生動地模擬連續手語動作、增加使用者學習興趣。

3-3-3. 中文語音合成及變聲模組之設計與嵌入

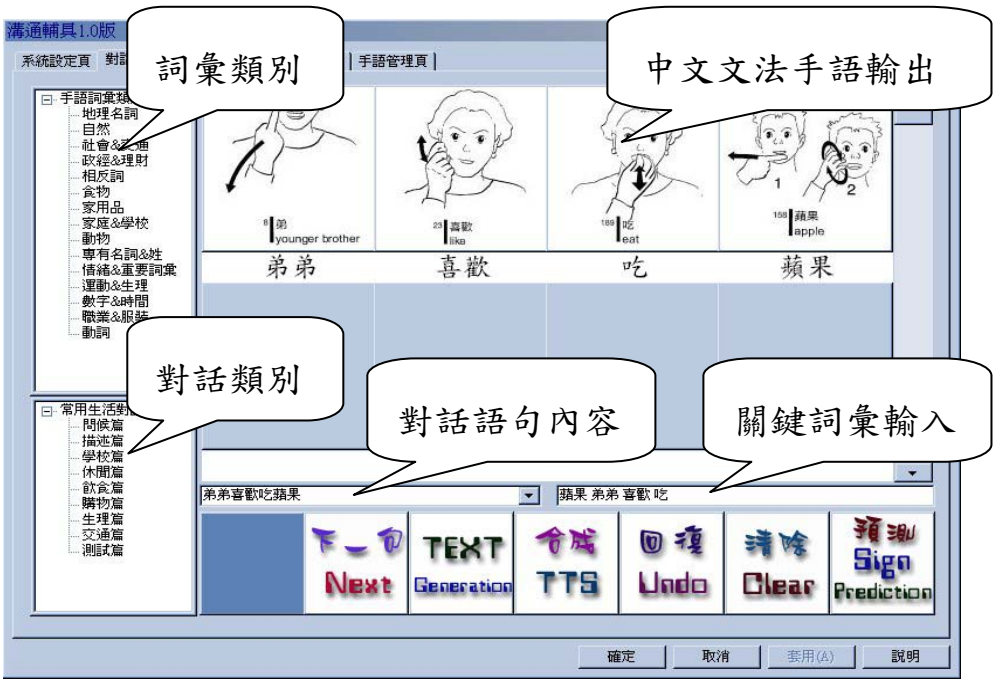
語音是聽語障礙者與正常人溝通的重要替代工具，一套適用於群體的溝通輔具系統，必須讓使用者擁有個別化的聲音，才能達到個體區別及群體溝通的目的，針對此點，本研究整合了成功大學資訊工程研究所研發之中文語音合成系統，透過共振峰（formant）、音高（pitch）、說話速度等合成參數的調整，將合成聲音多樣化，讓聽人可以聽到不同人的聲音。

4. 實驗結果與討論

本研究之智慧型台灣手語轉語音溝通輔助系統（如圖十一所示），乃以教育部之手語畫冊及現代經典出版之手語大師為手語教材，共有 1185 個手語符號，研發平台為 Pentium-III 450 筆記型/個人電腦、64MB RAM，作業系統為 Win98/WinNT，開發工具為 Microsoft Visual C++ 6.0。系統效能評估主要包括: 1). 構句操作輸入速度(Rate Enhancement)之評估: 主要探討系統所提供之預測機制的效能; 2). 構句正確性(Accuracy Enhancement)之評估: 經由實際個案評估之規劃，探討本系統實際使用之可行性。

4-1. 系統功能評估

主要著重於構句速度改善之探討，本實驗透過虛詞省略與否以及不同模式之預測策略評估系統效能：1).虛詞省略（**Functional Word Deletion**）：本研究採用之語料庫（**corpus**）乃為台南師範學院特教系及台南啟聰學校特教老師所提供之 1000 句對話語句（平均長 4.9 字/句），包含上課教材以及學生日常生活對話。本研究採用關鍵詞（N、V）構句，免除虛詞輸入，估計約可節省 26.25%的 **Keystroke**，如表一所示。



圖十一 智慧型 PC-based 台灣手語轉語音溝通輔助系統

（手語符號取自現代經典文化出版之手語大師叢書）

	Keystroke Numbers	Keystroke Saving Rate
關鍵詞+虛詞	2777	Baseline
關鍵詞	2048	26.25%

表一 虛詞省略之 Keystroke Saving Rate

2).詞彙選取（**Keyword Selection Steps**）：依據 Row-column Scanning 策略，採用（2＊4）分頁的方式，並結合由上而下、由左而右的檢視策略，將最常用的手語詞彙置前。檢視次數為搜尋手語詞彙所需檢視步驟（頁→列→行），相當於手語詞彙所在的位置，計算方式為：

$$KSS_{av} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{t(s_i)}\sum_{j=1}^{t(s_i)}\mu * P_j + R_j + C_j \qquad (4)$$

其中，N 表示測試語句個數；$t(S_i)$表示關鍵詞個數；μ 表示分頁的單位（本系統為 8 個手語詞彙）；P_j 、R_j 、C_j 分別表示頁次、列位、行位；如採用本研究所提供之詞彙預測、句型預測及注音縮寫查詢等輔助構句的方式，檢視次數最多可縮減達 96.87%左右，如表二所示。

	Average Scanning Step	Improvement Rate
Without Prediction	39.27	Baseline
Word Prediction	12.68	67.71%
Syntax Prediction	8.05	79.50%
Abbreviation	1.23	96.87%

表二 各詞彙預測模式之 Scanning Improvement Rate

4-2. 個案適用性評估

影響溝通輔具效能的因素很多－包括使用者對語言的認知能力、中文文法概念、構句複雜度，以及系統核心技術、介面設計的問題等，因此，本研究透過實際個案適用的方式，將溝通輔具實際應用於聽語障礙者的日常生活之中，經由系統化的教學與訓練過程中，發掘個案適用所產生的問題，進而提供系統核心技術的改進及輔具介面的設計，讓輔具符合適用個案的需求，達成生活溝通的目的。同時透過適用個案的構句結果，進一步評估系統構句速度及文句生成之正確性。

4-2-1. 適用性評估之規劃

本溝通輔具以聽語障礙者與聽人之相互溝通為研究重點。在此階段性研發的過程，本研究以台南啟聰學校的學生為適用對象，為期 3 個月時間，透過日常生活對話的方式，讓學生實際使用本輔具與聽人相互溝通，以實際評估系統效能。同時，透過三期階段（教學期、調適期、評估期）的實驗，發掘個案適用所產生的問題，分階段性地修改系統核心技術及介面安置設計，以符合使用族群個別化的需求。

4-2-2. 個案篩選程序

由於本輔具適用之個案除了必須熟悉手語而且沒有語言認知的問題之外，對於中文文法概念仍須有一定程度的認知，因此，本研究透過讓學生看圖手寫造句的方式，作為個案選取之考量，其篩選程序為：1).情境語料選取：由南聰特教老師選取 14 張教學及生活圖片；2).測試對象選取：由南聰特教老師評量學生活動反應及語文能力，挑選 4 位候選聽障學生；3).測驗方式：讓學生看圖手寫造句（可造兩句以上）；4).篩選標準：由特教老師解譯及分析學生手稿，選取構句錯誤率最低者。所謂『構句錯誤』係指詞序重組之後，正常人仍然無法理解的語句，如『貓生氣是兔子可怕』；構句錯誤發生之原因，主要是由於個案對於中文文法認知上的問題及多意圖之表達所造成，基於個案適用性之考量，本實驗以構句錯誤率最低者為優先選擇。經過以上的個案篩選程序後，本實驗選取 1 位四年級學童作為個案適用對象。

4-2-3. 試用評估程序

本研究之實驗採用的對話題材為一般日常生活常用對話，主要分為：個人基本資料、家庭類、學校類、興趣類等等。聽語障礙者與不懂手語的聽人透過本溝通輔具，針對不同題材相互對話，其溝通方式為 1.）採用互動問答模式；2.）聽語障礙者以關鍵詞構句後，透過語音合成器將語句傳達給不懂手語的聽人『聽』；3.）聽人輸入語句後，透過手語轉換器將語句轉成對應的手語傳達給聽語障礙者『看』。以此模式讓輔具實際應用於聽語障礙者的生活中，拉近聽語障礙者與聽人之間的距離之外，更是評估系統效能的最佳方式，因為唯有透過實際

的應用，才能確實評估實際的系統效能。

4-2-4. 輔具效能評估方式

評估方式分為兩部分：輸入速度之改善及構句正確性之評估。本研究依照上述實驗程序，分三個研究時期收集聽語障礙者之對話語句評估系統效能，並且經由系統化的教學與訓練過程中，發掘個案適用所產生的問題，修正系統核心技術及介面設計方向：

i. 教學時期（**Training Phase**）：教學時期著重在 1.）輔具功能及使用方式之說明，2.）系統功能之補強及輔具介面之改良。本時期需配合手語老師進行，因為，手語老師最瞭解學生學習的問題所在，而且其教育背景與教學經驗，更是我們輔具設計寶貴意見的來源。

ii. 調適時期（**Adaptation Phase**）：調適時期著重在 1.）熟悉輔具的功能及使用方式，2.）系統自動學習，3.）系統除錯。本時期讓學生自己摸索並熟悉輔具操作方式，透過學生實際操作溝通輔具，系統自動學習個案的使用習慣；而且在實際操作過程之中，更可發現之前未注意到的系統錯誤，在此一併修正之。

iii. 評估時期（**Evaluation Phase**）：評估時期著重在 1.）輔具操作效能之探討，2.）輔具效能瓶頸之分析。本時期為系統效能發揮之時期，因此，除了探討系統操作是否趨向於平穩狀態之外，分析造成系統效能瓶頸的原因，更是提供未來輔具效能改善的重要參考指標。

本實驗依據不同時期所收集的對話語句，評估構句速度的改善，並透過構句成功率及構句滿意度分析，評估系統構句之正確性：1).構句成功率：所謂『構句成功』是指系統成功地利用句型樣版將關鍵詞串轉為完整語句。當系統無法順利產生語句時，系統仍然保留一組由關鍵詞串組合而成的預設語句，供正常人猜想聽語障礙者可能要表達的意思；2).構句滿意度：所謂『構句滿意度』是指正常人對於系統所產生語句的滿意程度。本研究將構句滿意度分為 5 個等級：5 表示優良(excellent)、4 表示良好(good)、3 表示尚可(fair)、2 表示差 (poor)、1 表示極差(unsatisfactory)。構句滿意度由與聽語障礙者溝通的教師或照顧者來評分，當系統『構句成功』時，正常人針對系統構句結果做一評比。

4-3. 適用結果與分析

本研究經過三個時期的對話語料收集之後，彙整出構句速度、構句成功率以及構句滿意度評量表，分別探討比較：1).**教學時期**：由於學生對於輔具功能及操作方式完全陌生，本時期需配合特教老師說明並引導學生使用本輔具，因此，本時期的對話語料以自我介紹主題。學生一開始會以類似平常手寫造句的方式來構句，因為學生對於系統關鍵詞構句的方式一時

還無法適應，所以本時期的平均關鍵詞個數偏高，同時也是造成構句成功率偏低的部分原因；另外，由於學生對於系統操作方式的不熟悉以及系統功能的不足，導致本時期的平均構句時間偏長；至於構句滿意度以及構句成功率，因系統核心尚未成熟，本時期的構句成功率偏低，而且構句滿意度也稍差；2).**調適時期**：根據教學時期特教老師的建議，系統在核心部分以及輔具介面設計皆有修正及補強。本時期之對話語句著重在個人基本資料的相互詢問。本時期學生對於系統操作方式已漸趨熟悉，加上關鍵詞輸入方式及輔具介面的改善，因此，平均構句時間已縮短許多；學生構句方式漸漸以關鍵詞的方式來構句，因此本時期的平均關鍵詞個數較教學時期為少，同時，也是構句成功率提升的部分原因；至於構句滿意度以及構句成功率，因系統核心之修正及補強，本時期之構句滿意度及構句成功率皆有上升的趨勢；3).**評估時期**：經過調適時期的適用，系統已自動學習使用者的使用習慣到達近似穩定的狀態，此時期的系統正處於使用者最為熟悉的狀態，因此，本時期的平均構句時間最短，而對話語料則著重在個人興趣以及生活趣事的討論。因為學生已能掌握關鍵詞構句的訣竅，本時期幾乎皆以關鍵詞的方式來構句，因此，本時期的平均關鍵詞個數較前兩個時期為少，同時，也是構句成功率趨向穩定的部份原因；至於構句滿意度以及構句成功率，因系統核心在適應時期已大致底定，本時期之構句滿意度及構句成功率皆已趨向穩定的狀態。探討系統構句失敗的原因，多來自於複雜構句，因此，系統若朝解決複雜句型的方向研究，相信，對於構句成功率將有顯著的提升。

5. 結論

本 PC-based 擴大及替代式本土化溝通輔助(AAC)科技系統 – 台灣手語轉語音溝通輔具之研發，應用現階段自然語言理解/技術、語意解析、語音訊號處理及多媒體處理的科技，且實際考量聽語障礙者不同臨床殘障功能性需求、台灣手語轉譯模式、中文文法/手語斷詞處理、語言生成處理、人因工程及人機介面設計之最佳化匹配，提供本系統初步的雛形系統發展，且針對台灣手語及中文文法相互轉譯之特性，初步解決了詞序、綴語、Agent/Theme 等問題。初步個案適用性評估結果顯示：雖然本雛形系統之手語認知符號預測選取、文句生成構句處理等基本功能性已能符合實際聽障學童之需求，但距離輔具研發之終極目標-互動式語音雙向溝通，此系統仍尚待改良。然而在進行系統適用性評估的過程，由於聽語障相關的教材相當缺乏，使得目前文句生成仍侷限較簡單的對話語句。

有鑑於語音溝通輔助系統設計及發展為需整合不同學門，其研發以符合個人需求、高功能性、高擴充性、低複雜度及具教育臨床療效為考量重點，因此，仍具相當大的發展空間與

研究方向，以提供更符合國內聽語障礙者所需之溝通輔助工具。而針對目前國內相當缺乏適當的本土化聽語障礙者所需之相關教材、特殊教育語言教材與國語母語教材，本研究提供一多媒體電腦輔助訓練系統，以協助進行溝通訓練與語言學習的活動，更可擴展於輔助手語教學與訓練之用，以期能改善國內聽語障礙者之日常生活溝通表達。

誌　謝

承蒙中華民國聾人協會顧玉山老師、台南師範學院邢敏華教授及台南啟聰學校陳衫吉老師在手語教材、教育訓練方面的協助及提供研究所需之重要參考資料，本文得以完成。感謝國科會 NSC89-2614-H-006-003-F20 經費補助，特此致謝。

參考文獻

Reichle, Joe, "Implementing Augmentative and Alternative Communication: strategies for learners with severe disabilities.", Paul H.Bookes Publishing Co.,1992.

Simpson, R. C. and Koester, H. H., " Adaptive One-Switch Row-Column Scanning", IEEE Transaction on Rehabilitation Engineering, Vol. 7, No. 4., 1999.

Webster, J.G., et. al., " Electronic Devices for Rehabilitation", John Wiley & Sons, Inc., 1985.

David, R. B. and Mirendan, Pat, "Augmentative and Aternative Communication", Paul H. Bookes Publishing Co., 1992.

Koester, H. H. and Levine, S. P., "Modeling the Speed of Text Entry with a Word Prediction Interface", IEEE Transaction on Rehabilitation Engineering, Vol. 2, No. 3., 1994.

Hunnicut, S., "Word Prediction: Exploring the Use of Semantic information", Augmentative and Alternative Communication, Vol. 6, No. 2., 1990.

Vanderheiden, G. C., "A High-Efficiency Flexible Keyboard Input Acceleration Technigues: SPEEDKEY", Proceedings of the Second International Conference on Rehabilitation Engineering, RESNA, pp. 353-354., 1984.

Chang, S. K., et. al., " A Methodology for Iconic Language Design with Application to Augmentative Communication", Preceedings of the 1992 IEEE Workshop on Visual Language, pp110-116., 1992.

史文漢，台灣手語語言學概論，台北市政府勞工局手語翻譯員培訓班教材，民國 89 年。

Baker, B., "Minspeak", Byte., 1982.

Demasco, P., et. al., "Towards More Intelligent AAC Interfaces: The Use of Natural Language Processing", The12th Annual Conference of RESNA, 1989.

Demasco, P. and McCoy, K. F., "Generating Text from Compressed Input: An Intelligent Interface for Prople with Severe Motor Impairements", Communication of the ACM, Vol. 35, No. 5., 1992.

古鴻炎、許文龍，”時間比例基週波形內差-一個國語音節信號合成之新方法”，第九屆計算語言學研討會，pp.61-83，民國 85 年。
吳宗憲、陳昭宏、林超群，”中文文句翻語音系統中連音處理之研究”，第九屆計算語言學研討會，pp.85-104，民國 85 年。