語料庫建構技術研究

期末報告

壹、語料庫的定義與發展的歷史

語料庫是指經過抽樣選出具有某一種代表性的口語，書面語，或語音資料庫。這些語料通常以電腦儲存與分析。1947 年 Shannon 以統計的噪音通道模型(Noise Chanel Model)為基礎工具，所發展出的訊息理論(information theory)奠定了語料庫計算語言學的基礎。在實際語料的收集與語料庫的建構方面，1961 年美國布朗大學 Francis 與 Kucera 兩位學者開始建構 Brown Corpus，這個語料庫收集了各類文體的美式英文共一百萬字。Brown Corpus 的重要性在於它是第一個語言學家有計畫建構的大型平衡語料庫，為語料庫語言學的研究開啟了一個新紀元。

由於 Noam Chomsky 對語料庫的批評加上當時機讀語料十分缺乏，電腦價格昂貴且運算速度緩慢等許多因素，使得語料庫語言學的研究在 1960 到 1980 這 20 年僅限於少數學者。一直到 1980 年代中期語音辨認研究人員經過多年的努力證明以語料庫為主使用隱式馬可夫模型(Hidden Markov Model)的統計演算法明顯優於利用語言規則的方法。另一方面 John Sinclair，Geoffrey Leech, Sidney Greenbaum, Jan Svartvik, Randolph Quirk.等學者（其中大部分為英國語言學家）運用電腦與大量機讀語料從事英文詞彙，文法，辭典編纂與計算語言學的的研究，獲得相當好的成果（參看 Sinclair (1987)1, Quirk, Greenbaum, Leech, Svartvik (1985)2, Garside, Leech, Sampson (1987)3）英國伯明罕大學 John Sinclair 教授與 Harper Collins 出版社於 1980 年代合作，建立大型機讀語料庫，並以此語料庫的做為編纂 Collins Cobuild 英文辭典的基礎。Collins Cobuild 的成功促使包括牛津大學與劍橋大學在

[1] Looking Up: An Account of the Cobuild Project.
[2] A Comprehensive Grammar of the English Language.
[3] The Computational Analysis of English : A Corpus-Based Approach.

內的大出版社紛紛建構大型機讀語料庫來編纂英文辭典。以大型機讀語料庫來編纂辭典的好處是可以客觀且方便地檢視詞的頻率，搭配語，及語意，語法，與語用的功能，來判斷詞的用法。以語料庫為主的語言學與計算語言學研究在沈寂了近二十年後才漸漸復甦， 1990 年代從事自然語言處理的研究人員將原先為語音辨認所發展的統計演算法運用到自然語言剖析，詞彙知識自動習得(automatic lexical knowledge acquisition)，機器翻譯等以語料庫為主計算語言學的研究上，獲得豐盛的成果。在加上大型機讀語料因為網際網路的盛行而垂手可得，以及個人電腦功能日益強大，售價卻十分低廉，這些因素使得 1990 年中期以後，以語料庫為主的計算語言學研究成為主流。

貳 、語料庫的資源

料庫種類除了有口語，書面語，與語音資料庫還依是否為平衡語料庫，有否加標記，單語或多語等方式來區分。加標記的語料庫包括加註文章結構標記(如標題，句子，段落等)，詞類標記，語意標記，或語法樹等數種。未加標記的英文語料庫以 Brown Corpus, LOB Lancaster-Oslo-Bergen) Corpus, BNC (British National Corpus), 與 Project Gutenburg 最著名，後者收集了許多英文小說。加詞類標記的英文語料庫包括 Penn Corpus, Sussane Corpus 等，中文加詞類標記的語料庫目前有中研院平衡語料庫(約 500 萬詞)。加註語意標記的語料庫目前尚缺乏。英文語法樹庫則有 Penn Treebank。至於多語語料庫最著名的是加拿大國會以英法文記錄的的 Hansard Corpus。目前國內可以線上取得的中英，中日平行語料庫則有光華雜誌。

目前最常用的兩個句法樹庫資料,分別是中央研究院中文句結構樹資料庫（Sinica Treebank） (http://www.aclclp.org.tw/use_stb_c.php)，以及美國賓州大學

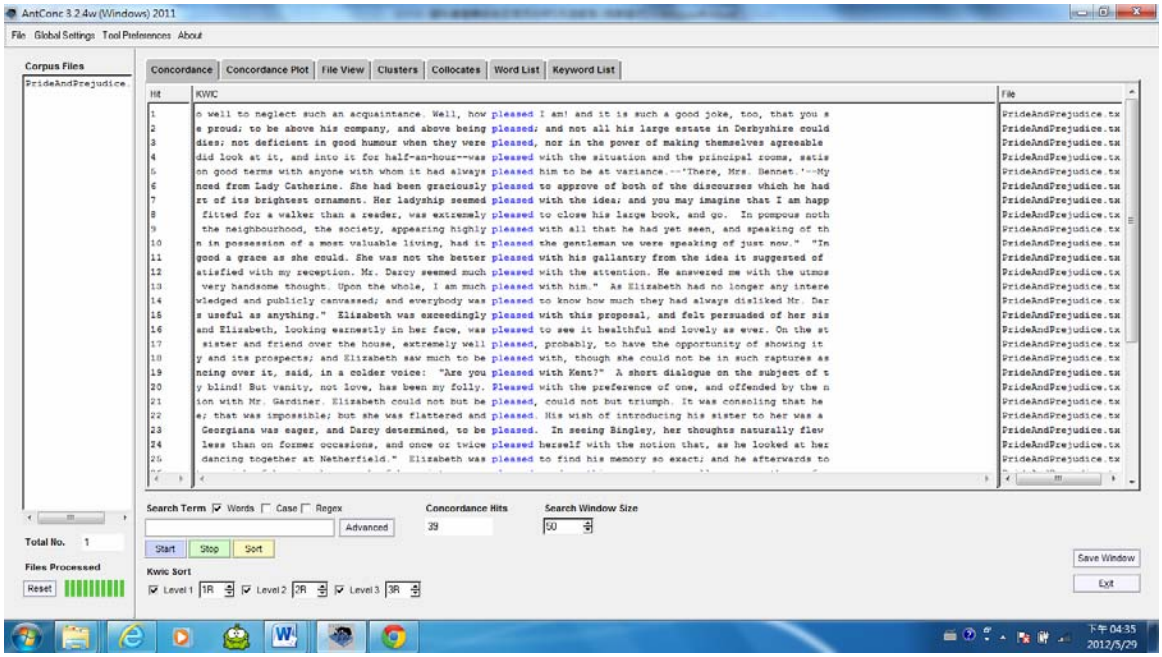
中文句法樹庫（Penn Chinese Treebank ）(http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2004T05)。兩者在語言，語料來源，語料庫大小，標記集，標記單位，標記訊息，及依據的語言學理論都不相同。

Sinica Treebank 與 Penn Chinese Treebank 最大的差別在於結構樹的語法單位不同。前者以標點符號作為分隔不同結構樹的單位，因此一個結構樹很多時候只是一個詞組（如 PP, NP）而不是一個完整的句子。而後者除小部分結構樹是句子的片段（以 FRAG 標示）大部分的結構樹是完整的句子(sentence)(以 IP 標示)。另外 Sinica Treebank 語法結構採取中心語主導原則 (Head-Driven Principle)，註明中心語(Head)和其他成分（如附加語）的語法和語意訊息，表達出句子中詞和詞之間的語法結構和語意角色關係，而 Penn Chinese Treebank 並沒有中心語與語意角色的訊息，而是在詞組上加註如主詞 SBJ 受詞 OBJ 等語法功能的方式來取代。

由於 Sinica Treebank 有未簡化標記，簡化標記及精簡標記三種標記集，相較於 Penn Treebank 只有一種標記集，Sinica Treebank 的三種不同的標記集可以作為不同的特徵。除此之外只有 Sinica Treebank 有標示語意角色的訊息，Penn Chinese Treebank。由 Linguistic Data Consortium (LDC)所發行，其中標示語意角色的 Penn Chinese Treebank 稱為 Chinese Proposition Bank。

參、語料庫語言學的工具

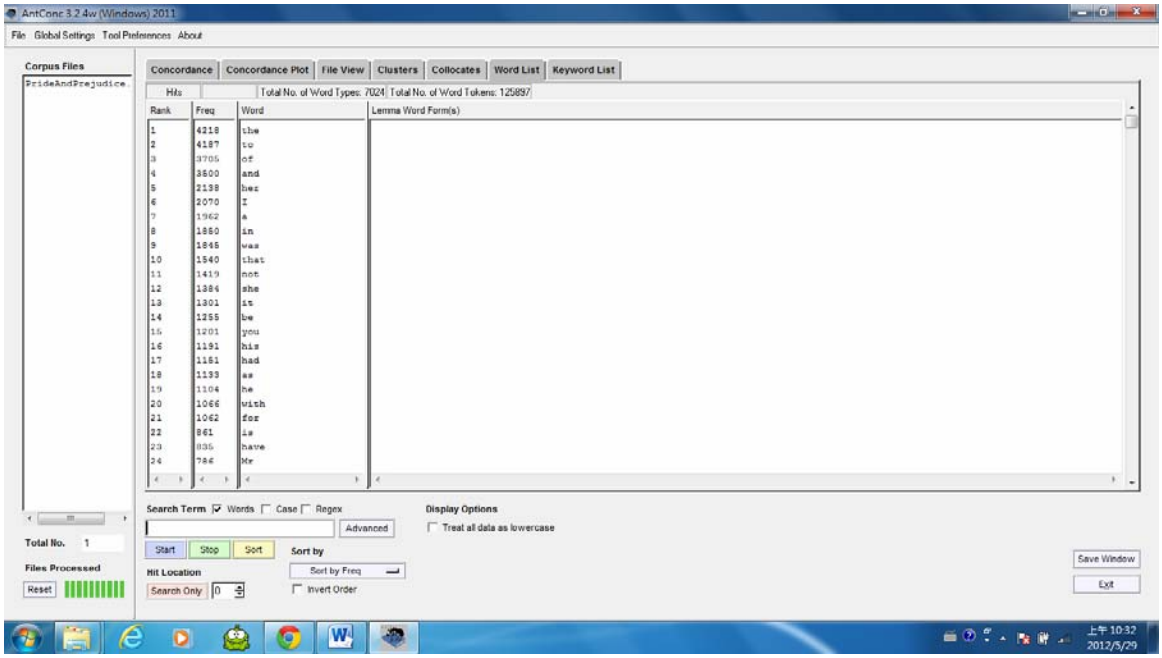
一、關鍵詞前後文程式(concordancer)：輸入一個關鍵詞或字串，程式自動將語料庫中所有包含這個詞或字串例子找出來置中並顯示前後語境。Antconc 是一個免費軟體，可以計算語料關鍵詞的頻率，並檢索關鍵詞以及搭配語。下面的畫面擷取自 Antconc 關鍵詞上下文檢索程式 concordancer 的功能。



圖一 AntConc 關鍵詞前後文排序程式

http://www.antlab.sci.waseda.ac.jp/antconc_index.html

二、詞頻程式：計算某個特定的字串或每個出現在語料庫中的詞的頻率。如上面 Antconc 內建 concordancer 功能，搜尋某一個關鍵詞時，下方 Concordance hits 會顯示這個關鍵詞在這個語料庫出現幾筆。如下圖，點選 Antconc 上方 Wordlist 即可計算每個出現在語料庫中的詞的頻率，且會依照頻率高低排序。



圖二 AntConc 詞頻排序程式

http://www.antlab.sci.waseda.ac.jp/antconc_index.html

三、英文還原詞原型程式(lemmatizer)：輸入一個英文詞，程式自動將句中的每一個詞詞轉為原形。

四、中文分詞程式：輸入一個句子，程式自動找到詞與詞的界線並將詞分開。由於人名，地名，及具有衍生性的詞無法全部列舉在辭典中，在加上分詞程式無法完全解決岐義的問題，中文分詞程式的準確率大約只有 90%到 97%。中文最簡單的分詞演算法是長詞優先，但如下例有時會造成錯誤。

例如輸入：把手舉起來。

輸出：把手　舉　　起來。

最具代表性的正體字分詞程式是中研院詞詞知識庫小組的分詞程式。利用機器學習演算法發展出來且可以自由下載的簡體字中文分詞程式有 LingPipe http://alias-i.com/lingpipe/demos/tutorial/chineseTokens/read-me.html 以及史丹福大學的 Chinese Word Segmenter http://nlp.stanford.edu/software/segmenter.shtml。若要使用簡體字中文分詞程式處理正體字需先轉成簡體字，程式處理完再轉回正體字，在繁簡繁三道轉換過程，有些字可能會轉錯。

五、詞類標記程式(part-of-speech tagger):程式自動將輸入的句子的每一個詞標上詞類。目前英文的詞類標記程式可達到 98%以上的正確率,如 Stanford Parser。繁體中文的詞類標記程式以中研院詞庫小組以最具代表性。中研院詞詞知識庫小組的分詞程式以及史丹福大學的 Chinese Word Segmenter 都可以同時處理分詞和詞性標記，但兩者的分詞標準和詞性標記集(tagset)不同。

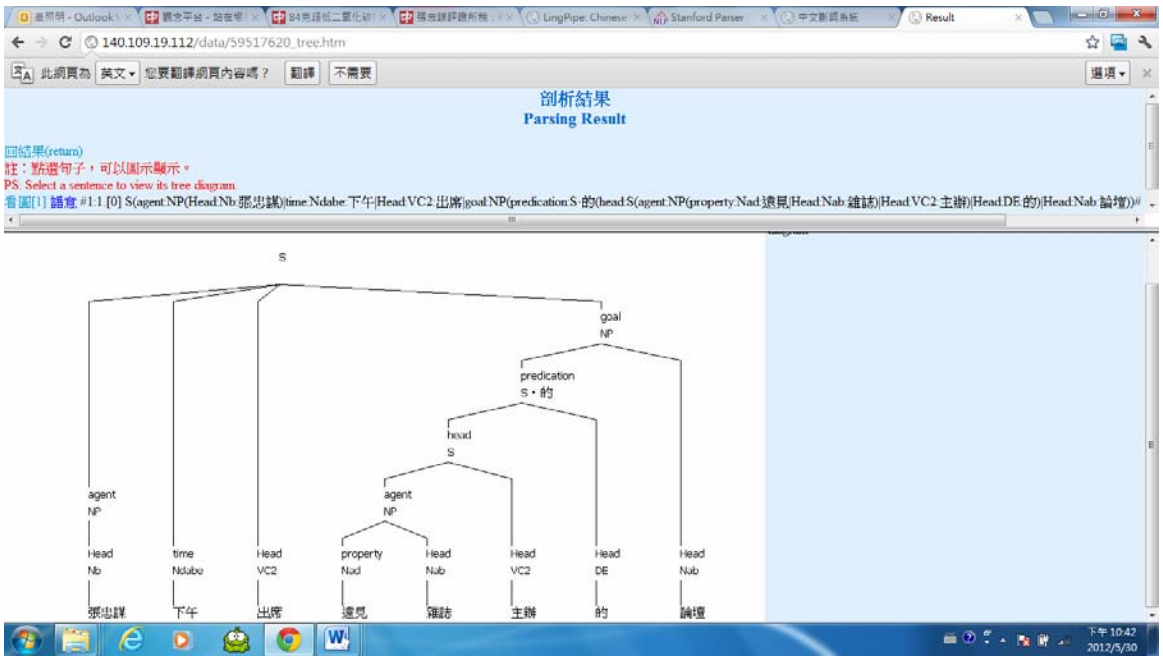
圖三 中研院詞詞知識庫小組的分詞和詞性標記程式
http://ckipsvr.iis.sinica.edu.tw/

六、語法剖析器(parser)：程式自動將輸入的句子的句法層次結構標示出來。語法剖析器可以分成兩種，完全剖析和部分剖析(partial parse)。近年來興起能判斷依存關係的語法剖析器,如英文的 Minipar 及 Stanford Parser。瑞典 Lund 大學以 Mate-tool 為基礎發展簡體中文的語法剖析器提供程式碼供研究人員下載。

Sinica Treebank 與 Penn Chinese Treebank 最大的差別在於結構樹的語法單位不同。前者以標點符號作為分隔不同結構樹的單位，因此一個結構樹很多時候只是一個詞組（如 PP, NP）而不是一個完整的句子。而後者除小部分結構樹是句子的片段（以 FRAG 標示）大部分的結構樹是完整的句子(sentence)(以 IP 標示)。另外 Sinica Treebank 語法結構採取中心語主導原則（Head-Driven Principle），註明中心語(Head)和其他成分（如附加語）的語法和語意訊息，表達出句子中詞和詞之間的語法結構和語意角色關係，而 Penn Chinese Treebank 並沒有中心語與語意角色的訊息，而是在詞組上加註如主詞 SBJ 受詞 OBJ 等語法功能的方式來取代。

如（圖五）所示，中研院的中文句法樹庫的 terminal node 是詞，詞上方有詞性標記和中心語（head）這類的語法訊息，構成詞組的結點(node)有詞組標記和語意角色等語意訊息。

圖四　中研院詞詞知識庫小組中文剖析器的輸入介面

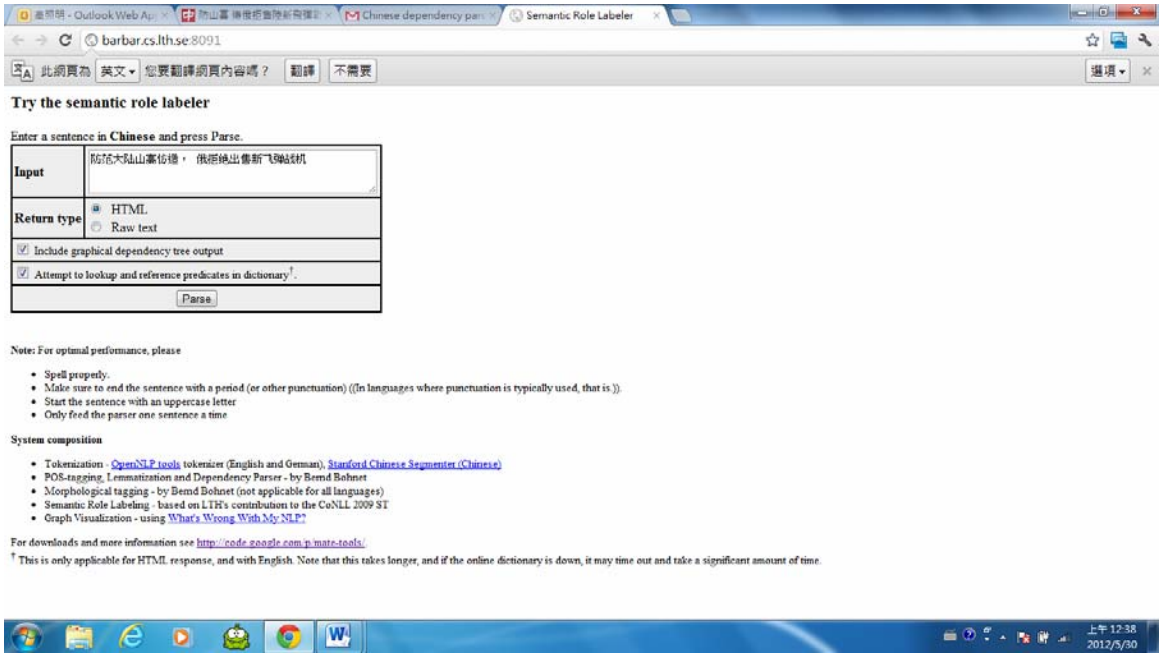


圖五　中研院詞知識庫小組中文剖析器的輸出介面

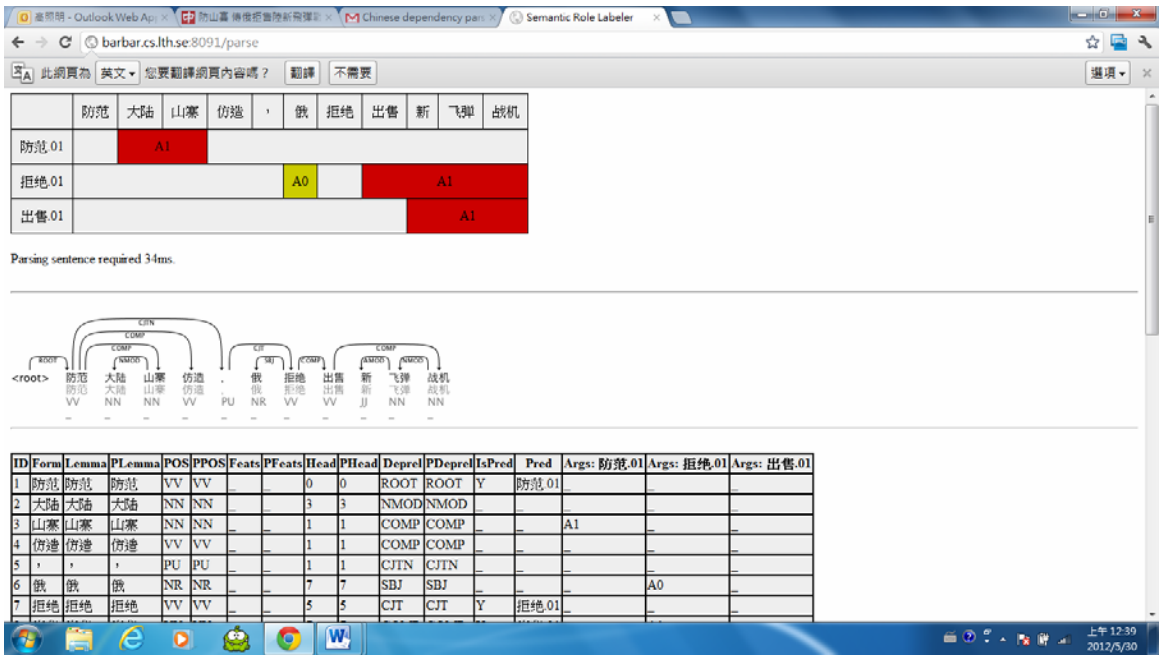
圖六是 Stanford Parser 中文剖析器的輸入介面。圖七 Stanford Parser 中文剖析器的輸出最重要的部分是句子的語法結構樹和語法依存關係。語法結構樹顯示詞組之間的語法關係如 NP 是名詞組，VP 是動詞組，IP 是句子。圖七最下方顯示詞與詞與詞之間的語法依存關係。

圖六 Stanford Parser 中文剖析器的輸入介面
http://nlp.stanford.edu:8080/parser/

圖七 Stanford Parser 中文剖析器的輸出介面
http://nlp.stanford.edu:8080/parser/



圖八 瑞典 Lund 大學以 Mate-tool 為基礎發展簡體中文的語法剖析器的輸入介面
http://barbar.cs.lth.se:8091/



ID	Form	Lemma	PLemma	POS	PPOS	Feats	PFeats	Head	PHead	Deprel	PDeprel	IsPred	Pred	Args: 防范.01	Args: 拒绝.01	Args: 出售.01
1	防范	防范	防范	VV	VV	_	_	0	0	ROOT	ROOT	Y	防范.01	_	_	_
2	大陆	大陆	大陆	NN	NN	_	_	3	3	NMOD	NMOD	_	_	_	_	_
3	山寨	山寨	山寨	NN	NN	_	_	1	1	COMP	COMP	_	_	A1	_	_
4	仿造	仿造	仿造	VV	VV	_	_	1	1	COMP	COMP	_	_	_	_	_
5	，	，	，	PU	PU	_	_	1	1	CJTN	CJTN	_	_	_	_	_
6	俄	俄	俄	NR	NR	_	_	7	7	SBJ	SBJ	_	_	_	A0	_
7	拒绝	拒绝	拒绝	VV	VV	_	_	5	5	CJT	CJT	Y	拒绝.01	_	_	_

圖九 瑞典 Lund 大學以 Mate-tool 為基礎發展簡體中文的語法剖析器的輸出介面
http://barbar.cs.lth.se:8091/

七、線上機讀辭典：教育部國語會出版的國語辭典，除了解釋，並有例句，相似詞，相反詞。

圖十 教育部重編國語辭典修訂本檢索介面
http://dict.revised.moe.edu.tw/

圖十一 教育部重編國語辭典修訂本檢索結果
http://dict.revised.moe.edu.tw/

八、詞彙知識庫：大陸董振東先生獨力發展出來的知網 Hownet 是一個非常重要的詞彙知識庫(參考 Dong and Dong (2006)。知網 Hownet 包含的訊息相當的多，是一個雙語的知識庫，可以表達概念的語意成分，概念之間的語意關係，概

念之間的常識關係。例如醫生在 Hownet 裡面有三個英文翻譯 doctor, surgeon, doctor，它們的義元表示都是{human| 人 :HostOf={Occupation| 職位},domain={medical|醫},{doctor|醫治:agent={~}}}。義元是一種表達語言知識的 meta language,醫生的義元表示醫生是一個人，具有職位，是醫學領域，且是醫治事件裡面扮演主事者的語意角色。 而醫療這個詞只有一個義元{doctor|醫治}。中研院詞庫小組將 HowNet 擴充成為 E-HowNet。

圖十二 詞庫小組的 E-HowNet 檢索結果
http://ehownet.iis.sinica.edu.tw/

「同義詞詞林」是另一個具有中文語義的資料來源。「同義詞詞林」編排的方式是按照語意階層由大類到小類分類。如下列例子所顯示，A 大類的詞都與人有關，Ae142 都與裁判有關而 Ae151 都是醫師。可惜的是，它的分類方式仍然不夠詳細。同義近義語意上下位詞的區分不夠。

Ae142","裁判員" "Ae142","裁判" "Ae142","公正人" "Ae142","國際裁判" "Ae142","國家裁判" "Ae142","巡邊員" "Ae142","記分員" "Ae142","計時員" "Ae151","醫生" "Ae151","醫師" "Ae151","醫" "Ae151","大夫" "Ae151","郎中" "Ae151","醫官" "Ae151","先生" "Ae151","白衣戰士" "Ae151","國醫" "Ae151","中

醫" "Ae151","良醫" "Ae151","賢醫" "Ae151","名醫" "Ae151","神醫" "Ae151","太醫" "Ae151","御醫" "Ae151","法醫" "Ae151","仵作" "Ae151","世醫" "Ae151","儒醫" "Ae151","庸醫" "Ae151","西醫" "Ae151","牙醫" "Ae151","獸醫" "Ae151","軍醫" "Ae151","廠醫" "Ae151","校醫" "Ae151","赤腳醫生" "Ae151","主任醫師" "Ae151","副主任醫師" "Ae151","主治醫師" "Ae151","住院醫師" "Ae151","醫士"

肆、語料庫與計算語言學

利用語料庫與統計是近年來計算語言學研究的主要趨勢。無論對語音辨識，語法剖析，岐義的解決，機器翻譯，與詞彙知識的自動取得在在都需要大型語料庫。語音辨識是將語音訊號轉變成文字，基本上可以分成前處理與後處理。前處理是從聲波的物理性質猜測最有可能的母音與子音組合。而後處理則從最有可能的音中選出最有可能的詞，無論是語音辨識的前處理或後處理或解決詞類岐義目前最常用的統計理論是隱式馬可夫模型(Hidden Markov Model 簡稱 HMM)。而解決結構岐義則常利用語法樹庫計算某一詞出現在某一種結構的機率，例如 John saw the man with a telescope.其中的介詞組 with a telescope 可以修飾名詞組 the man(約翰看見一個帶望遠鏡的人)也可以修飾動詞 saw(約翰用望遠鏡看見一個人)，造成岐義的現象。英文這種所謂 PP attachment 結構岐義的問題跟詞彙的語義與語用有關，過去計算語言學家嘗試用規則來處理效果不好，目前改以語法樹庫計算介詞組內的名詞分別跟受詞與動詞的相關性機率，從而預測介詞組究竟修飾受詞或動詞。統計演算法不需大量的人力來撰寫語言規則或編纂語言知識，可以從大型語料庫中直接抽取諸如同義詞，反義詞，搭配語等語言知識。統計方式也可以自動抽取部分中文詞彙。

伍、語料庫處理的基本的計算工具

UNIX 與 LINUX 作業系統具備相當多的工具可以用來處理語料庫資料。我們建議讀者在 PC 上安裝 LINUX 作業系統，如此可以使用 UNIX 作業系統所具備的工具，並撰寫 PERL 程式抽取語料庫的訊息。[4]

一、字串轉換程式 tr

將英文檔案 datafile 中所有的大寫變成小寫並存到新檔案 output 裡面

tr 'A-Z' 'a-z' < datafile > output

將檔案中所有不是大寫或小寫的符號轉成空白行

tr –sc 'A-Za-z' '\012' < datafile > output

二、排序程式 sort

按照 ASCII 順序排序 sort datafile > output

按照數字從小到大排序 sort –n datafile > output

按照數字從大到小排序 sort –nr datafile > output

按照第三欄位從大到小排序 sort +2 –nr datafile > output

三、處理連續重覆行的程式 uniq

連續重覆的行只保留一行 uniq datafile > output

計算各行資料連續重覆數 uniq –c datafile > output

排序後去掉重覆的行 sort datafile | uniq > output

排序後計算頻率 sort datafile | uniq –c > output

將檔案中所有不是大寫或小寫的符號轉成空白行，排序，再計算頻率

[4] LINUX 作業系統內附 PERL 語言的轉譯器(Interpreter)可以自網路免費取得。此外 PERL 語言的轉譯器也有安裝在 Window 作業系統的版本。

tr –sc ‘A-Za-z’ ‘\012’ < datafile | sort | uniq –c > output

計算行數，字數，字元數的程式 wc

計算行數 wc –l　datafile

計算字數 wc –w　datafile

計算字元數 wc –c　datafile

四、從檔案中將包含某一字串或形式的行列出來 grep

從檔案中將包含 pattern 這個字串的行列出來 grep ‘pattern’ datafile > output

五、awk 一種簡單但功能強大的程式語言

從檔案中將包含 pattern 這個字串的行列出來

awk ‘/pattern/ { print }’ datafile > output

列印 datafile 第二欄與第一欄中間以 tab 間隔

awk ‘{ print $2 “\t” $1}’ datafile > output

如果 datafile 第三欄的值大於 1.65 列印第二欄與第三欄，中間並以 tab 間隔

awk ‘{ if ($3 >= 1.65) { print $2 “\t” $3} }’ datafile > output

六、perl：結合了 C 語言，awk，sed，shell programing，功能比 awk 更強大，非常適合處理語料。

陸、計算語言學中常用的公式

一、互見訊息(mutual information):

互見訊息(mutual information)是訊息理論(information theory)中的基本概念，計算的方式是兩個事件共同出現的機率除以個別事件出現的機率的積除再取以二為底的對數。

$$MI(x,y)=\log_2 \frac{P(x,y)}{P(x)P(y)}$$

如果只考慮緊鄰的兩個詞，則可代入下列公式。其中 N 代表總詞數，f(x,y) 代表 x 與 y 一起出現的次數 f(x)，f(y)分別代表 x 出現的次數與 y 出現的次數。

$$MI(x,y)=\log_2 \frac{\frac{f(x,y)}{(N-1)}}{\frac{f(x)}{N}\times\frac{f(y)}{N}}\cong\log_2 \frac{N\times f(x,y)}{f(x)f(y)}$$

Ken Church (1991)與他的同事率先提出以互見訊息計算詞與詞之間的相連性(word association)[5]。互見訊息值越高表示詞的相連性越高，當語料庫夠大時，而互見訊息值大於零，表示這兩個詞常常一起出現很可能是搭配語(collocations)，成語，或常見的人名，地名。利用互見訊息可以從中文語料庫中自動抽取詞彙。

二、T-值(t-score):

互見訊息可以視為一種相似度測量，T-值則可以視為相異度的測量。T-值

[5] Using Statistics In Lexical Acquisition.In Zernik, U. (1991) (eds.) Lexical Acquisition: Exploiting On-Line Resources to Build a Lexion.

(t-score)是計算語言學中常用的統計顯著性的檢定(statistical significance test)，也是 Ken Church (1991)與他的同事率先運用在計算語言學，通常與互見訊息搭配一起使用。T-值與標準差和信賴區間(confidence interval)密切相關。當語料庫夠大時，而 T-值大於 1.65 時表示有 95%的信心證明差異是存在。

T 值的計算可以採用下列簡化的公式。

$$t \approx \frac{f(x,y) - \frac{f(x)f(y)}{N}}{\sqrt{f(x,y)}}$$

三、熵(entropy):

熵(entropy)也是訊息理論中的基本概念，1960 年代被用於資訊檢索，1990 後被廣泛運用到計算語言學中。熵可視為是測量驚奇，不確定性，或訊息的量。公式如下。

entropy = - k Σ pi log(pi)

四、N 連詞(ngram)語言模型(language model)是是統計式計算語言學中最簡單也最常用的語言模型。N 連詞語言模型假設一個句子第 N 個詞的機率可以由前面的 N-1 個詞決定。雖然這個假設過於簡化語言的複雜性，但在語音辨識與其它應用上不失為一個有效的方法。最常用的是二連詞(bigram)與三連詞(trigram)模型。

柒、中文語料庫的比較與分析

目前正體字的中文語料庫最具有代表性的兩個，分別是中研院詞詞知識庫小組的現代漢語平衡語料庫及 Sketch Engine。前者包含 1 千萬詞，後者則包含 4 億 5 千多萬詞，兩者除了語料的數量之外，最大的差別在於 Sketch Engine 從分詞到詞性標記到語法依存分析完全是程式自動處理，且沒有經過人工檢查，現代漢語平衡語料庫除了程式處理之外，則經過人工多次的檢查，正確率較高。至於檢索的方式，兩者都支援關鍵詞檢索，也可以搭配語詞性檢索，也都可以檢索關鍵詞左邊或右邊若干詞之內的搭配語。現代漢語平衡語料庫並支援各種重疊詞的檢索，以及依據文類來檢索。

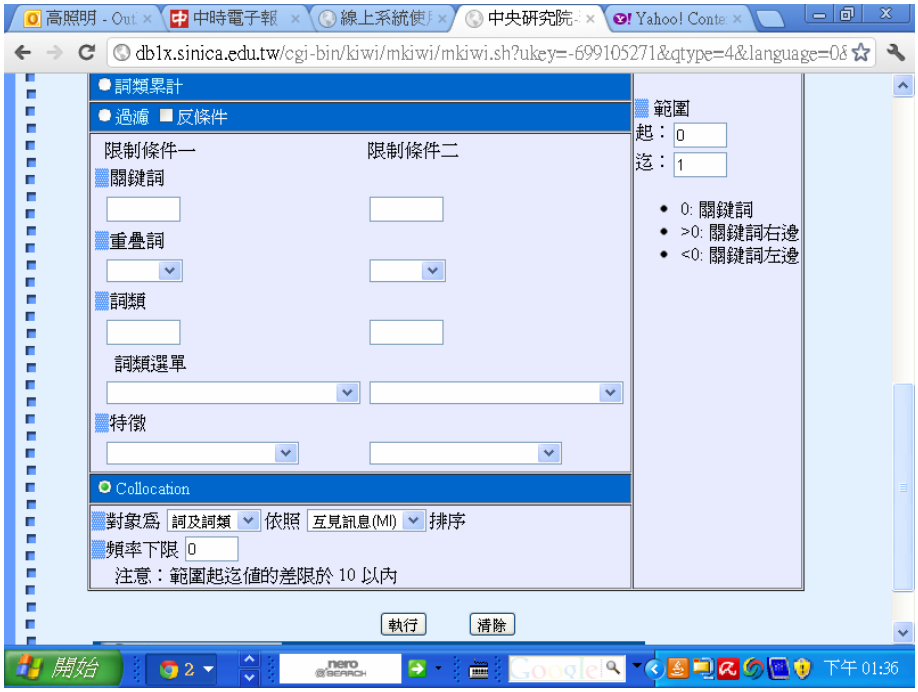
圖十三 現代漢語平衡語料庫的檢索介面
http://db1x.sinica.edu.tw/kiwi/mkiwi/

圖十四 現代漢語平衡語料庫檢索輸出結果

http://db1x.sinica.edu.tw/kiwi/mkiwi/

圖十五 現代漢語平衡語料庫檢索輸出結果(顯示詞性)

http://db1x.sinica.edu.tw/kiwi/mkiwi/



圖十六 現代漢語平衡語料庫進階檢索介面

http://db1x.sinica.edu.tw/kiwi/mkiwi/

共 143 項

MI	freq(y)	freq(x,y)	y:詞/詞類
8.715	2	1	旱船(Na)
8.715	2	1	拱北殿(Nc)
8.715	2	1	山貨(Na)
8.309	3	1	安全第一(VH)
8.309	3	1	亂踢(VC)
7.798	5	1	十多萬(Neu)
7.798	5	1	大隊接力(Na)
7.616	6	1	婦女版(Na)
7.328	16	2	商展(Na)
7.211	9	1	接力賽(Na)
7.211	18	2	急診(VB)
7.010	11	1	歐美所(Nc)
7.010	11	1	三點半(Nd)
6.923	12	1	環(VC)
6.882	50	4	江湖(Nc)
6.635	32	2	馬拉松(Na)

圖十七 現代漢語平衡語料庫進階檢索功能中的搭配語的檢索

http://db1x.sinica.edu.tw/kiwi/mkiwi/

圖十八 Sketch Engine 詞彙速描介面

http://www.sketchengine.co.uk/

圖十九 Sketch Engine 詞彙速描輸出結果

http://www.sketchengine.co.uk/

捌、雙語語料庫建構技術

一、如何建構雙語平行語料庫

由於自動對齊雙語文章的句子是計算語言學界近年來積極研究的議題，且牽涉到相當複雜的計算，我們留到下一節敘述。我們先探討是否有中英雙語資料可以不經過複雜的自動句子對齊程序來建立一個電腦輔助翻譯工具。答案是肯定的。有一些雙語資料由於有特殊的段落或句子標記可以輕易的找出對應的句子或段落。Resnik, Olsen, and Diab (1999)就注意到聖經的每一章節段落與詩篇都有數字標記，透過這些標記即可找到對應的句子或段落。類似這樣有句子或段落標記的雙語語料還可以從開放程式碼(Open Source)軟體的說明文件找到一些。

對於沒有明顯段落標記的雙語資料，如果翻譯者在翻譯原文時相當忠實的保留了原文的段落，沒有增加或刪減，那麼我們可以紀錄每一個詞出現在哪幾篇文章的哪幾個段落並做成索引檔，使用者輸入一個詞後，程式查索引檔得到詞出現的檔案及段落位置，即可顯示出包含關鍵詞的段落及對應的翻譯。為了幫助使用者快速找到正確的翻譯，關鍵詞及包含關鍵詞的段落及可能的翻譯以較顯目的顏色標示出來，從使用者的角度來看，這樣的工具雖然在找對應段落的正確率不是特別高，但因為正確的段落對應通常落在程式判斷的段落附近，所以仍然有相當高的實用性。

如前所述，利用段落對應來找對應句並不是一個很可靠的方法，因為翻譯者在翻譯原文的時候多少會做一些增減。另一個困難是中文對於句子的定義相當模糊，有些時候用逗點，有些時候則用句點，不同的人對同一段文字通常就會有不同的標法。這些都是嘗試以中英平行語料庫自動找翻譯對應句時會遭遇的困難。下面是光華雜誌的例子。

(1) 近年來，校園民主的呼聲日切，大學生自主意識越來越高，中國文化中

特有的「尊師重道」、「一日為師，終身為父」倫理觀念，也在時代的衝擊下逐漸解體。

'In recent years calls for democratization of campuses have grown more insistent. Traditional Chinese concepts of the proper ethical relationship between students and teachers, in which students accorded teachers the same level of respect they accorded their own fathers, are dissolving.'

(2) 在大學校園裡，這樣的故事越來越不是特例；許多教師感覺到，經過了社會泛政治化和民主化的洗禮、新「大學法」的頒布實施，和女性主義在校園中蔚然成風的衝擊，大學校園裡，師生之間似乎隱隱形成了角力戰，關係也愈來愈微妙。

'Stories like these are less and less exceptional on university campuses. Manyprofessors have come to believe that a number of factors have laid hidden bones of contention in teacher-student relations in recent years' politicisation of all aspects of life, democratisation in society, the promulgation of the new

"University Law" three years ago, rising feminism. . . . Relations have become much more subtle and complex.'

從上面的例子我們可以發現由於標點符號使用不嚴謹，中文句子有時以逗點有時以句點表示。在找對應句時，如果以英文句子為單位來找中文的對應句將會相當困難。Gao (1998)提出中文的句點，驚嘆號，問號是比句子大的言談單位(discourse unit)的標記，以這些標點符號為單位來找英文對應單位比較容易。

二、如何從平行語料庫中自動找對應句

隨著語料庫計算語言學的興起，研究人員發現可以用機讀雙語詞典或統計方法從平行語料庫自動抽取翻譯對應句。以下簡述幾種常用的方法及所面臨的問題。

（一）以機讀雙語詞典找對應句

機讀英漢電子辭典可以用來找平行語料庫中英文詞的對應，進而猜測其句子的對應。作法又可以分為精確匹配與部分匹配兩種，前者只能找到很有限的翻譯對應，而後者雖可找到較多的翻譯對應，但其中與上下文不符合造成對應錯誤的情形相當多。我們將在後面提出實際的例子。

（二）統計方法

統計演算法的優點在於只需要大量語料庫不需要機讀辭典或語言知識即可找出句子的對應。統計的方法有兩類。一類是直接利用句子的長度關聯性的假設(Brown et al. (1991), Gale and Church (1993))，也就是如果原文某一句較長，那麼翻譯的句子應該也會較長，再利用動態規劃的技巧(dynamic programming)找出哪一句最有可能對應哪一句， Brown et al. (1991)及 Gale and Church (1993)利用加拿大國會英法雙語資料(Hansard)找出段落的標記後找句子對應，正確率在 93%以上。Gao (1998)在實驗後發現上述方法不適用中英雙語語料。如表（一）顯示英法雙語語料(Hansard)有 89%的句子是一對一對應。一對多或多對多的句子對應關係相當少。而利用光華中英雙語語料所做的實驗表（二）顯示無論以中文句點或逗點做為單位，與英文句子一對一的關係都不高，分別是 53%與 35%且多對多的對應關係相當的普遍。因此以句長的關連性來找中英文對應句相當困難。

表（一）Gale and Church (1993)句子的對應關係的機率

Category	Frequency	Prob(match)
1 - 1	1167	0.89
1 - 0 or 0 -1	13	0.0099
2 - 1 or 1 - 2	117	0.089
2 - 2	15	0.011

表（二）光華雜誌中英句子應關係的機率（以中文句點當作單位）

Bead Types	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
frequency	0.53	0.32	0.06	0.06	0.03

表（三）光華雜誌中英句子應關係的機率（以中文逗點當作單位）

Bead Types	(1, 1)	(1,2)	(1,3)	(1,4)	(1,5)
frequency	0.35	0.38	0.17	0.06	0.04

另一種統計的方法是以詞的頻率與分佈情形來猜測詞的對應，進而找出句子的對應（例如： Kay and Roscheisen (1993), Fung and Church (1994)）。這種方法的缺點是受到頻率，語系,文類,風格等因素的影響很大。再者，根據詞在文章出現位置的分佈情形與出現頻率只能抽取一小部分頻率不高不低的詞彙（頻率太高可能是功能詞很難找到固定的翻譯，頻率太低則無法透過統計得到）。無論是利用統計或機讀電子辭典從中英平行語料庫自動擷取雙語詞彙對應句的困難在於翻譯並非一對一對應，而是隨著上下文語境而變化。如何有效地結合統計與語言知識（例如：辭典、詞類標記、與語法結構）成為研究的重點。

Gao (1998)測試並改良 Fung 與 Church (1994) 的演算法。Fung 與 Church (1994)提出 K-vec 演算法結合互見訊息(mutual information)與 t 值(t-score)等兩個統計方法來計算兩個詞在文件內部區段的共現關聯性。互見訊息(mutual information)是訊息理論(information theory)中的基本概念，計算的方式是兩個事件共同出現的機率除以個別事件出現的機率的積再取以二為底的對數。如果只考慮緊鄰的兩個詞，則可代入下列公式。其中 N 代表語料庫大小（即總詞數），f(x,y) 代表 x 與 y 一起出現的次數 f(x)，f(y)分別代表 x 出現的次數與 y 出現的次數。

互見訊息 $\mathrm{Log}_2 ((P(A \cap B) / (P(A) * P(B)) = \mathrm{Log}_2 (f(x,y)/f(x)*f(y))$

Ken Church (1991)與他的同事率先提出以互見訊息計算詞與詞之間的關聯性(word association)。互見訊息值越高表示詞的關聯性越高，當語料庫夠大時，而互見訊息值大於 1.65，表示這兩個詞常常一起出現。互見訊息可以視為一種相似度測量，T-值則可以視為相異度的測量。T-值(t-score)是計算語言學中常用的統計顯著性的檢定(statistical significance test)，也是 Ken Church (1991)與他的同事率先提出運用在計算語言學，通常與互見訊息搭配一起使用。T-值與標準差和信賴區間

(confidence interval)密切相關。當語料庫夠大時，而 T-值大於 1.65 時表示有 95%的信心證明差異是存在。計算 T 值的公式如下。

$$t = \frac{P(x|y) - P(x|z)}{\sqrt{\sigma^2 P(x|y) + \sigma^2 P(x|z)}}$$

其中 x|y 表示 y 出現時 x 出現的機率。σ 表示標準差。

T 值的計算可以採用下列簡化的公式。其中 N 代表語料庫大小（即總詞數）。f(x,y) 代表 x 與 y 一起出現的次數 f(x)，f(y)分別代表 x 出現的次數與 y 出現的次數。

$$t \approx \frac{f(x,y) - \frac{f(x)f(y)}{N}}{\sqrt{f(x,y)}}$$

Fung 與 Church (1994)的基本的假設是如果有兩篇互相對應的文章，某語言一個詞與另一個語言的一個詞在某些區段一起出現的機率大於個別出現的機率，則它們兩個詞有可能是翻譯。Fung 與 Church 將相對應的翻譯文章均分為 K 個區段（K 為文章長度的平方根），以 K 維向量來紀錄兩個語言中某個詞出現在哪幾個區段，例如在第一區段出現就將對應的向量值設為 1，否則設為 0。詞頻太低與太高的詞都不適合使用此演算法，因為若只出現一兩次的詞即使分佈區段完全相同也很有可能是巧合，而出現很頻繁的詞很可能是功能詞才會在很多區段一起出現，這些都必須先排除掉，否則會影響演算法的精確度。Fung 與 Church 建議使用詞頻在 5 次到 10 間的詞，以 K 維向量（K 為文章長度開平方）來表示其分佈情形之後再利用互見訊息與 t 值來計算頻率相近的中文與英文詞在相同區段一起出現的機率。Fung 與 Church (1994)使用下列聯方表。

表(四) 聯方表

a = k(A B)	b = k(~ A B)
c = k(A ~B)	d = k(~A ~B)

a 表示某個中文詞與英文詞一起出現的區段數， b 表示英文詞出現但中文詞沒有出現的區段數， c 表示中文詞出現但英文詞沒有出現的區段數， d 表示中文

詞與英文詞都沒有出現的區段數。再利用下列稍微修改過的互現訊息與 t 值，其中 P(Vc) 為某一中文詞出現在區段的機率，P(Ve) 為某一英文詞出現在區段的機率。

$$MI(V_c, V_e) = \log_2 \frac{P(V_c, V_e)}{P(V_c)P(V_e)}$$

$$P(V_c) = \frac{a+b}{a+b+c+d}$$

$$P(V_e) = \frac{a+c}{a+b+c+d}$$

$$t(V_c, V_e) = \frac{P(V_c, V_e) - P(V_c)P(V_e)}{\sqrt{\frac{P(V_c, V_e)}{K}}}$$

Gao (1998)使用中研院詞知識庫小組發展的分詞程式處理中文的分詞，並以中英對照之光華雜誌做實驗證明上述方法的精確度受到文類的影響很大，如下表所示，精確度有可能高至 70%也有可能低至 30%以下，此外利用此演算法實際能找到的對應詞相當有限。數千詞長度的對應文章，大多只能找到幾個對應詞。為了提高精確度，Gao (1998)改良 Fung 與 Church (1994)演算法。首先我們計算中文與英文的文章段落數目是否一樣，若一樣我們則將 K 設為段落數，若不一樣則依舊採用原先的定義。我們也合併賓州大學（University of Pennsylvania）與美國暑期語言學院（Summer Institute of Linguistics）所發展的構詞分析程式，將英文所有的名詞，動詞，形容詞換成原型，使計算共現機率時能更準確。此外我們不僅利用文章內部區段一起出現的機率，也收集數十篇對應文章，再以中文詞與英文詞出現在同一篇文章的機率（亦即文件的共現關聯性）來過濾 Fung 與 Church (1994)演算法所得到的結果，將精確度大幅提高至 90%以上。

除了改良 Fung and Church (1994)的 K-vec 演算法，我們也利用機讀英漢電子辭典得到部分翻譯對應，以便推論其它的翻譯對應。一般人認為利用機讀英漢電子辭典即可很容易得到平行語料中的詞彙對應關係，事實上撰寫程式呼叫電子辭

典自動查詢所得到的對應仍然非常有限。主要原因在於（1）一個詞可能有幾個翻譯，以機讀辭典判斷那一個詞對應哪一個詞，必須從上下文找訊息，相當困難，從實驗中我們發現功能詞的意義相當多，利用機讀辭典來得到翻譯對應非常不可靠。（2）利用完全字串匹配（exact string match）所能得到的翻譯對應相當有限。例如字典中 teacher 的翻譯是「教師」，實際上文章可能翻譯成「老師」，若採用部分字串匹配-（partial string match）則可以找到辭典中的翻譯與文章的翻譯有一個字「師」相同。採用部分字串匹配雖可以找到相當多可能的翻譯對應，但錯誤率也相對提高許多。為了解決這個問題，我們先排除最常出現的功能詞「的」。凡是部分字串匹配為「的」的翻譯對應一律排除。接著再找出相鄰兩個英文詞至少各有一個字與辭典翻譯吻合的連續詞。雖然利用上述緊鄰性(proximity)原則找到的詞組翻譯 (translation equivalents)相當有限，但精確度高達 90％以上。透過這一些正確率非常高的對應詞或詞組，我們即可得到某些翻譯句的對應。

利用段落對應來找對應句並不是一個很可靠的方法，因為翻譯者在翻譯的時候多少會做一些增減。另一個困難是中文對於句子的定義相當模糊，有些時候用逗點，有些時候則用句點，不同的人對同一段文字通常就會有不同的標法。這些都是嘗試以中英平行語料庫自動找翻譯對應句時會遭遇的困難。

（三）採用 Champollion Tool Kit (CTK) 找出中英文對應句

CTK 利用雙語辭典，數字，及簡體中文中的英文詞的對應再加上句長對應的關聯性透過統計演算法找出中英文句對應。我們在實驗中發現 CTK 在簡體中文與英文句對應方面得到不錯的效果。我們透過 Perl 中文簡繁體字的對應程式，將繁體中文轉換成簡體中文再使用 CTK 這個工具程式找出文章裡面中英文句對應。圖一是我們利用 CTK 及其它工具程式所發展的中英平行語料句對應程式的介面，使用者貼入兩篇互為翻譯的中文及英文。圖二十一是這個程式輸出的中英文對應句。

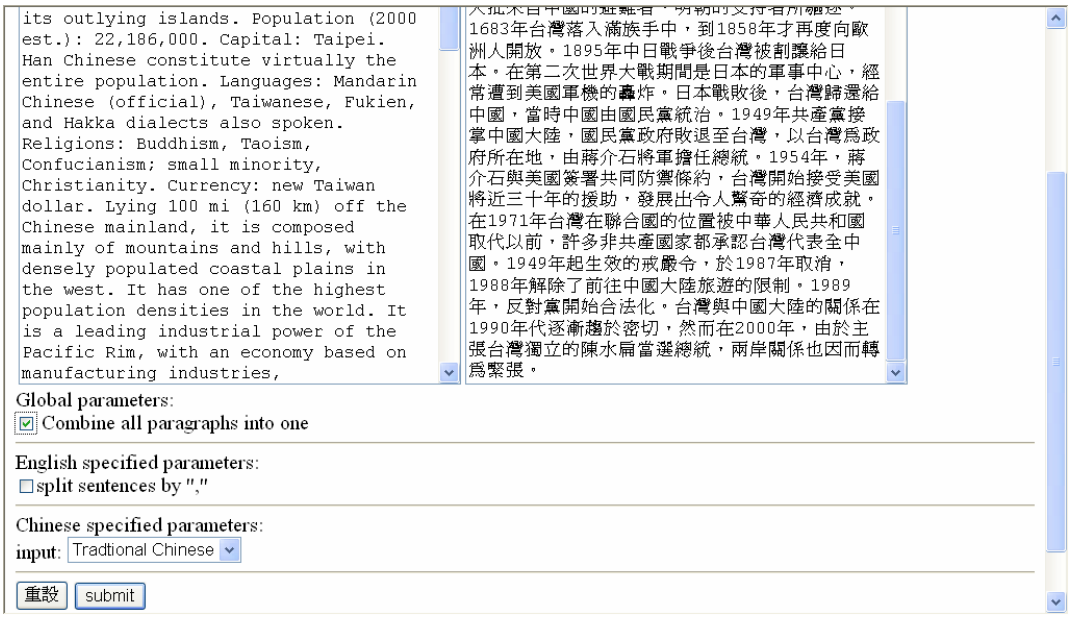
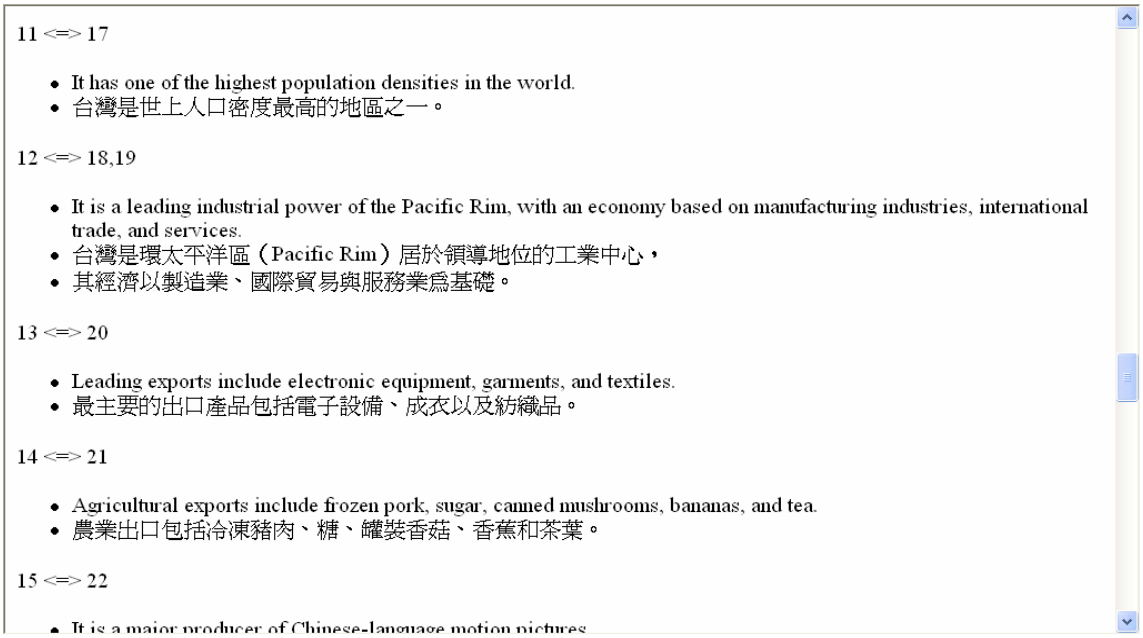


圖　二十 利用 CTK 所發展出來的中英平行語料對應程式介面



圖二十一　中英對應句程式輸出的結果

經過上述程式及分詞程式處理後，我們利用 Lucene 搜尋引擎將語料庫裡面所有中文文章的句子和英文翻譯自動做索引。我們所建構的這個中英雙語語料庫規模相當大，不但包括各種文類，來源語中屬於中文與英文比例接近，因此對-華語學習者或英語學習者而言都是一個很寶貴的資源。

玖、如何實做中文機率式無語境語法剖析器

自然語言處理的研究方法由 80 年代的規則式(rule-based)轉為資料導向(data-driven)的統計式機器學習演算法。標示各種語言訊息的語料因此成為非常重要的研究資源與工具。中研院的中文語法樹庫(Sinica Chinese Treebank)和美國賓州大學中文語法樹庫（Penn Chinese Treebank）就是其中的代表。另一方面自然語言處理學界也體會到開發一些共同的套件(toolkit)對於教學與研究的重要性，Bird, Klein, and Loper (2009)所發展出來的 NLTK (http://nltk.sourceforge.net/index.php/Main_Page)就是在這樣的背景下誕生。

從計算複雜度來看，無語境語法(context-free parser)比有限狀態機(finite-state machine)複雜，而有語境語法(context-sensitive grammar)又比無語境語法(context-free parser)複雜。然而即使有語境語法也無法完全解釋自然語言的現象（學者早已證明自然語言的複雜度等同於圖林機 Turing Machine）從效率和解釋力的觀點來看，無語境語法無疑是折衷的選擇。另一方面，無語境語法(context-free parser)等同於語言學的詞組結構律(phrase structure rules)，不但從語法樹庫擷取較容易，語言學家也比較容易解讀資料提出改善的意見。

Bird, Klein, and Loper (forthcoming)所發展的 NLTK 是一套以 python 語言寫成的程式模組、資料、與教學縱合套件。主要用來支援語言運算，自然語言處理方面的教學與研究之用。NLTK 適合想要學習 NLP（Natural language processing），對 NLP 或相近領域進行研究的學者。NLTK 已被成功的使用在教學、研究、與建立整個研究系統。我們使用 NLTK 中的 PCFG 工具來進行機率式無語境語法(probabilistic context-free parser)的建立。

Stanford parser (Klein and Manning (2002))是一個由史丹佛自然語言處理小組於 2002/12/05 首次釋出的多語 parser，最新的版本為 2006/6/11 釋出之 1.5.1 版。

Stanford parser 是 PCFG 的 JAVA 實作版本，並已做到詞彙處理的高度最佳化。在原始的版本中，這個 parser 主要是由 Dan Klein 與 Christopher Manning 在語言學文法上的基礎下撰寫而成；而之後新增的功能（如：國際化、語言專用模型、彈性 I/O、文法壓縮、使用者支援……等等。）則由 Roger Levy、Christopher Manning、Teg Grenager、Galen Andrew 一齊完成。Stanford parser 本身是一個多國語言剖析器，只要輸入符合英文 Penn Treebank 格式的語法樹庫語料，即可以得到這一個語言的語法剖析器。Penn Chinese Treebank 使用的中文樹庫格式與英文 Penn Treebank 大致相同，但有額外標記和訊息，此種格式由 Xue and Xia (2000)所提出。

Stanford parser 語法分析模型使用 Klein and Manning (2002)提出之 factored parsing 模型。這個模型包括了兩種獨立分析方式：其一為純 maximum likelihood-estimated PCFG 模型；另一個則是 constituent-free dependency parse。而在將原始模型套用在中文分析之上時，每個詞被分開而不標記，語法分析由主要觀察到歧異的幾個分類逐步做過改良。

與 Stanford Parser 同樣架構只要輸入 Penn ChineseTreebank 格式的中文語法樹庫語料即可以處理中文的多國語言的剖析器還包括 Chiang (2003)及 Bikel (2004)。

剖析(parsing)，即在一個句子上，將整個句子的語法樹建立出來的過程。在一般的 context free grammar 之上，語言會產生許多的歧異（ambiguous），舉例：

S → VP

S → NP

VP → V NN

NP → V NN

V → 進口

NN → 汽車

在這個文法之中，〔進口〕〔汽車〕可被建構成兩種不同的結構樹：

[S [VP [V 進口] [NN 汽車]]]

[S [NP [V 進口] [NN 汽車]]]

一個純粹的 CFG parser 便無法判斷此句子的結構，無法良好的應用在語言處理之上。因此，(probabilistic context free grammar)出現了，其中主要是加權文法的概念（weighted grammar）。在每一個 derivation(也就是詞組結構律)之中，我們增加了另一個參數：機率值。PCFG 建構原始 CFG parser 會建構出來的分析結果，但在每個結果之中，PCFG 為其連結了一個機率值，其值即簡單的由所有使用到的 derivation 機率值相乘而得到。

PCFG 的實作方面，我們先對中研院語法樹庫及賓州大學中文語法樹庫做前處理，由於中研院句法樹庫的格式與 Penn Chinese Treebank 不同，因此我們先將中研院語法樹庫改成跟賓州大學中文語法樹庫一致的結構，接下來寫一個剖析這些樹狀結構的剖析器，計算每條詞組結構律的機率，再利用 NLTK 裡面的函式庫。我們採用 Viterbi-style 的分析模型。Viterbi PCFG parser 是一由下而上的剖析器(bottom-up parser)，使用動態規畫(dynamic programming)來找出最有可能的分析結構樹。它藉由反覆的填寫一個最有可能的詞組表格“most likely constituent table”來做句子的分析，這個表格為所有分支與節點記錄了最有可能之樹結構。更詳細的說，它擁有四個欄位，分別記錄：

1. 分支開始索引
2. 分支結束索引
3. 節點標記
4. 結構樹

舉例來說，在分析「我 在 樓梯 上 看到 教授」之後，表格可能如表（五）所示：

表（五）

Most Likely Constituents Table			
分支	節點	結構樹	機率
[0:1]	NP	(NP 我)	0.3
[2:3]	NP	(NP 樓梯)	0.3
[5:6]	NP	(NP 教授)	0.3
[1:4]	PP	(PP 在 (NP 樓梯) 上)	0.05
[5:6]	VP	(VP 看到 (NP 教授))	0.03
[0:4]	NP	(NP (NP 我) (PP 在 (NP 樓梯) 上))	0.01
[0:6]	S	(S (NP (NP 我) (PP 在 (NP 樓梯) 上)) (VP 看到 (NP 教授)))	0.0001

當成功的填寫完這個表格時，parser 簡單的傳回起始節點（S）中，機率值最大的結構樹。

由於 Viterbi parser 使用的文法是 PCFG，任何一個元素的機率值都可以由它的子孫求得，而任何一個分支並不可能含括到比自己還要大的分支，因此任何一個分支便僅與較小的分支有關。

Viterbi parser 利用上述的條件，由較小的單元開始填表，從大小為 1 的元素開始，再來大小為 2、3、4，以此下去直到表內所有的空位都被填滿為止。

以下為上面例子的分析標記過程：

Inserting tokens into the most likely constituents table...

Insert: |=.....| 我

Insert: |.=....| 在

Insert: |..=...| 樓梯

Insert: |...=..| 上

Insert: |....=.| 看到

Insert: |.....=| 教授

Finding the most likely constituents spanning 1 text elements...

Insert: |=.....| NP -> '我' (p=0.15) 0.1500000000

Insert: |.=....| P -> '在' (p=0.61) 0.6100000000

Insert: |.=....| NP -> '樓梯' (p=0.5) 0.500000000

Insert: |..=...| LC -> '上' (p=0.1) 0.5000000000

Insert: |...=..| V -> '看到' (p=0.61) 0.6500000000

Insert: |....=.| VP -> V (p=0.2) 0.1300000000

Insert: |.....=| NP -> '教授' (p=0.5) 0.5000000000

Finding the most likely constituents spanning 2 text elements...

Insert: |....==| VP -> V NP (p=0.7) 0.0455000000

Finding the most likely constituents spanning 3 text elements...

Insert: |.===..| PP -> P NP LC (p=1.0) 0.1525000000

Finding the most likely constituents spanning 4 text elements...

Insert: |====..| NP -> NP PP (p=0.25) 0.0057187500

Finding the most likely constituents spanning 5 text elements...

Insert: |.=====| VP -> PP VP (p=0.1) 0.0069387500

Discard: |.=====| VP -> PP VP (p=0.1) 0.0069387500

Finding the most likely constituents spanning 6 text elements...

Insert: |======| S -> NP VP (p=1.0) 0.0002602031

拾、如何辨識中文名詞組（NP Chunking）

名詞組的辨識與標示（NP Chunking）是自然語言處理（NLP）的一個重要研究議題 (Ramshaw and Marcus (1995), Kudo and Matsumoto (2000, 2001))，無論是句法處理中的剖析(parsing)語意處理中的語意角色的標示（semantic role labeling）及篇章處理中的回指(co-reference)與連貫性（coherence），其它領域如資訊檢索(information retrieval) 資訊擷取（information extraction）文件探勘（text mining）文件分類，與文件自動摘要都需要名詞組的辨識，例如在資訊檢索中最常被檢索的大都是名詞組（特別是人名，地名，組織名等所謂的 name entity），因此在文件或網頁中自動辨識名詞組並建立索引以方便檢索分類及自動摘要是智慧型資訊處理極為重要的一環。

一般名詞組的辨識指的是基底名詞組（base NP），也就是將名詞組下面又包含名詞組的複雜名詞組（如關係子句及名詞組並列結構(NP conjunction)）排除在外。目前英文名詞組的辨識正確率可以達到 94%以上(Kudo and Matsumoto (2000, 2001))，但中文名詞組的辨識至今只有少數零星的研究。

在大規模語法樹庫還沒有建立之前，名詞組辨識常將組成名詞組結構的規律透過有限狀態機（finite state machines）去找出符合名詞組的 pattern (Voutilainen (1993)) 或從標記好詞性的語料庫以統計的方式得到 （Church (1988)）, 或結和語言規律和語料庫統計（Chen and Chen (1994)）。自從賓州大學大規模的英文語法樹庫(Penn Treebank)建構完成後 (Marcus, Santorini and Marcinkiewicz (1993)), 絕大多數的名詞組辨識研究是以機器學習（machine learning）的方法透過語法樹庫裡面的語法結構及前後語境的特徵得到。運用機器學習辨識名詞組的方法大致可分為 HMM (hidden Markov model)，transformation-based（Ramshaw and Marcus （1995））, memory-based（Veenstra（1998）, Tjong Kim Sang and Veenstra（1999）Argamon, Dagan and Krymolowski （1998））, maximum entropy （Skut and Brants

（1998）), 及 SVM (Kudo and Matsumoto, 2000, 2001)等方法。上述幾種的方法都是監督式學習。HMM (hidden Markov model)使用統計的方法在 finite state machine 的 transition function 之上加上語料庫的統計結果。transformation-based learning 由現有的語料庫訓練出 transformational rules, 再利用這些規則對測試資料作 parse。HMM，transformation-based learning, memory-based learning 在自然語言處理中已被廣泛應用。 SVM 則是一種較新的 machine learning 技術,近幾年逐漸被應用到自然語言處理的各項研究議題。

上述這些演算法針對英文 Wall Street Journal Corpus 訓練得到的結果顯示,精確率(precision)與召回率(recall)大都超過 90%, 其中以 SVM（Kudo and Matsumoto（2001））的效果最好，精確率(precision)與召回率(recall)都超過 94%。

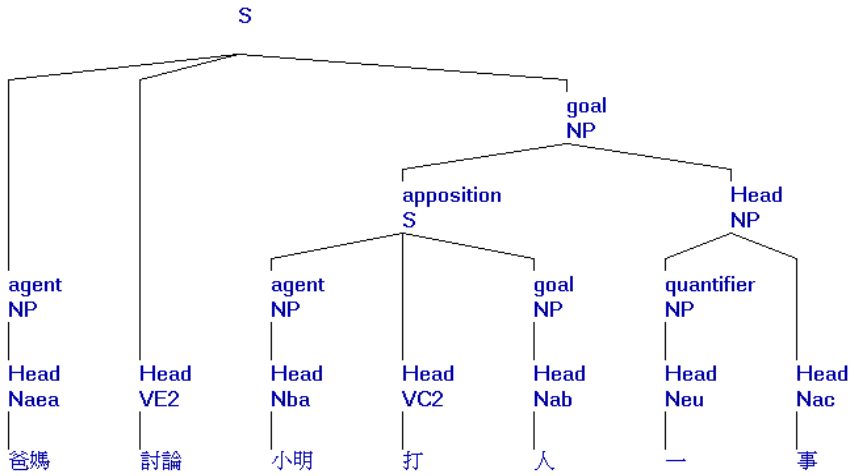
中文名詞組辨識的研究起步較晚，迄今只有零星的研究，還沒有針對同一個語料庫的大規模的測試與比較。例如中國大陸學者 Zhao and Huang (1998)提出以語料庫統計結合規律，利用 minimum description length principle (MDL)得到 quasi-dependency strength 加上規律來得到 base NP。這種採用非監督式機器學習（unsupervised learning）的方法，在封閉測試(close test)和開放測試(open test)中分別有 91.5% 和 88.7%的精確率。

由於 SVM 是監督式學習的演算法，必須擁有中文句法樹庫(treebank)的資料才能訓練出辨識名詞組的程式。

中研院的詞性標記集及每個標記代表的語言學涵請參考附錄。中研院詞知識庫小組所出版的「中文詞類分析」技術報告所提出的中文詞類的分類比簡化詞類更細，但為了顧及實用性中研院的漢語平衡語料庫所用的詞類標記為已經經過合併的簡化詞類。我們可以看出即使是簡化詞類，連接詞，名詞，動詞，副詞每一項都有不少的次分類。以動詞為例除了先分成動作與狀態兩大類之外，另外又根據動詞所帶的論元（argument）數目與種類各自分為若干小類。中研院另外又將簡化詞類做進一步的合併形成所謂的精簡詞類。在簡化詞類裡面的動詞原先有 16

類但在精簡標記裡面只剩及物與不及物動詞 2 類。

NP, VP 等詞組的判斷標準亦可採用中研院句法樹庫的資料做為測試的標準, 圖(二十二）是一個範例樹圖:



圖二十二 中研院句法樹庫範例
http://godel.iis.sinica.edu.tw/CKIP/treebank/apposition.htm

如（圖二十二）所示，中研院的中文句法樹庫的 terminal node 是詞，詞上方有詞性標記和中心語（head）這類的語法訊息，構成詞組的結點(node)有詞組標記和語意角色等語意訊息。我們的焦點是 NP,也就是由 ”爸媽”, ”小明”, ”人”, ”一”, ”一事”組成的詞組。”小明打人一事”這類名詞組因為包含其它的名詞組，不屬於基底名詞（base NP），所以不在我們的討論之列。

訓練語料由於採取中研院的句法樹庫所以句子已經分詞並標注詞性。張,高,劉(2005)以 Kudo and Matsumoto (2000, 2001)的經驗做為名詞組的辨識基礎。第一次實驗以 (I,O,B)三個標記分類:

這個方法以三個 class （I,O,B）表示一個詞在詞組中的位置:

I: 詞在詞組之中

O: 詞在詞組之外

B: 緊接著一個詞組之詞組的開頭

此種方法被 Tjong Kim Sang 稱為 IOB1 表示法。而 Start/End 標記最初被用在日本語的作業上 (Uchimoto et al.（2000)）。S，E，加上 I,O,B,共五個 class:

B: 多詞詞組的開頭

E: 多詞詞組的結尾

I: 詞在多詞詞組中

S: 單詞詞組

O: 詞在詞組之外

以下為兩者之範例標記:

	Inside/Outside	Start/End
這	I	S
是	O	O
詞組	I	B
標記	I	I
範例	I	E
說明	B	S

一開始, 我們簡單的將測試資料排列成 7 維的向量, Wordi 是 i 位置的詞，POSi 是 i 位置詞的標記，加上前後各兩個詞的標記:

Wordi POS(i-2) POS(i-1) POSi POS(i+1) POS(i+2)

這裡根據詞，詞的標記, 和前面後面各兩個詞的標記來做分類。上面的範例向量表示如下:

I	1:這	2:0	3:0	4:N	5:S	6:N
O	1:是	2:0	3:N	4:S	5:N	6:V
I	1:詞組	2:N	3:S	4:N	5:V	6:N
I	1:標記	2:S	3:N	4:V	5:N	6:V
I	1:範例	2:N	3:V	4:N	5:V	6:0
B	1:說明	2:V	3:N	4:V	5:0	6:0

從語言學的角度來分析，中文名詞組的辨識比英文困難原因在於中文的動詞可以修飾名詞，例如投資大眾，建設公司,流浪教師等。這些詞沒有任何構詞上的特徵或證據可以視為名物化（nominalization），因此詞性標記程式很難將這些詞判斷成名詞。由於中文的動詞可以修飾名詞使得自動辨識中文名詞組變得相當困難。不過我們仔細觀察後可以發現並不是所有的中文動詞都可修飾名詞，例如VD(雙賓動詞), VK(狀態句賓動詞), VG(分類動詞)等這些類的動詞很少有修飾名詞的例子。動詞次分類這個重要特徵若沒有考慮進去，辯識結的果將非常不理想。如下面的例子:

可能(D) 代表(VK) 台灣(Nc) 人民(Na) 對(P) 朝野(Na) 政黨(Na) 傳達(VD) 訊息(Na)

程式抽取出來的 NP chunks 為: “台灣人民”, “朝野政黨傳達訊息”; 顯然的“傳達” 並不應該出現在 NP chunk 之中, 而就我們給予 SVM 的資料來看, 這邊並沒有明顯的訊息可以得知其不適用 (我們給予 SVM 的資料為 “傳達(V)”), 而如 VH 等靜態動詞之類的動詞, 卻又常常出現在 NP 之中, 同樣標示為 V。 由於我們採取簡化詞類標記的第一個字母的大類來表示，在缺乏動詞次分類訊息特徵的情形下使得實驗結果非常不理想.因此, 我們保留將簡化標記動詞次分類的特徵，其它詞性則仍然使用大類, 結果如表（六）第二列所顯示，改良的方法在精

確率上提升了 23%以上, 召回率也提升了 6%以上,雖然還不是非常好, 但顯示了詞性標記的選擇（有無動詞次分類的訊息）是影響 SVM 效果的重要的特徵。

表（六）動詞次分類訊息對 SVM 的影響

	Precision	Recall
(1)取簡化標記詞性第一個字母做大部分類	54.99%	53.17%
(2)動詞採用簡化標記細部分類 其餘詞性取第一個字母大部分類	78.18%	59.33%

無論是精確率或召回率,我們實驗的結果與 Kudo and Matsumoto (2000,2001) 發表的結果 (94%) 差了一大段距離; 可以改進的地方如下:

IOB tag, 我們的實驗只採取了 I/O 兩種 tag, 這在當兩個 chunk 緊連的時候會是一個致命的問題 (無法確認 chunk 的終結點)。修改 tag, 使用 IOB 與 Start/End 將可提升辨識率。

由目前的經驗得知, 好的詞性分類有助於準確度的提升。所謂好的詞性分類是指透過細部的詞性分類將能名詞組內部與外部兩種不同的特徵顯示出來，而將無助於此項辨識工作的詞性細部分類精簡成大類。如此透過 SVM 演算法可以提升名詞組的辨識精確率。

kernel function 與其微調的參數是影響 SVM 準確度的一大原因, 預期將會使用 linear, polynomial, radial basis function, sigmoid... 等等函數來做逼近, 並嘗試採用 cross validation 來尋找最佳參數。

目前面對的問題還有一點為: 訓練的時間太久. 一個約 8,000 詞的訓練資料約需要花費 4 分鐘, SVM 之 time complexity 約為 O(n2), 也就是說若有一 300,000 詞之訓練資料, 將需要花費約三天以上的時間訓練, 如此一來, 對於要使用 cross validation 將會是一大挑戰, 因此會嘗試使用 scaling 的方式來減少所需要訓練的時間。

YAMCHA (http://chasen.org/~taku/software/YamCha/)是 Taku Kudo 專門為 NP Chunking 所設計的 SVM 工具，因此比一般性 SVM 工具（SVM Tool: LIBSVM

(Chih-Chung Chang and Chih-Jen Lin, 2004)）方便實做。YAMCHA 與 libsvm 的最大不同點在於:

a) Dynamic programming

b) Kernel Function

由於 libsvm 本身的限制, 我們很難能即時的將 chunking 的結果應用在下面一個未知 chunking 的判斷. 舉例而言, 之前的句子:

	Inside/Outside
這	I
是	O
詞組	I
標記	I
範例	I
說明	(B)

當 SVM 要判斷"說明"這個詞的 tag 時, 它會去參考"標記"與"範例"的詞與詞性;原來的設計並未考慮到它們的 IOB tag, 而由於中文 (其實任何語言應該都一樣) 有前後相依性, 因此把 IOB tag 計算在內, 會是一個適當而重要的特徵。

YAMCHA（Kudo and Matsumoto (2000,2001)）使用 IOB tag 代替 IO tag 方面, 由於 B tag 表示了一個緊鄰之前 NP-chunk 的開頭, 解決了兩個相鄰 NP-chunk 的分類問題。

另外 Kudo and Matsumoto (2000,2001)使用 voting 來提升辨識效果。voting 在很多應用中經常被使用。我們有許多不同的標記集, 和不同方向的 parsing 方式 (backward 即將所有的詞顛倒排列後做訓練與測試), 藉著由不同標記集和不同的 parsing 方向訓練出來的 SVM 模型, 可以採用其 Accuracy 之分數來統計未知詞組

的得分. 這種方法可以避開某些詞性標記或者是 parsing 方向的盲點, 以提升準確度。

另外從我們第一次的實驗結果得知動詞次分類訊息是一個影響 SVM 效果的重要的特徵。忽略動詞次分類的訊息會使辨識效果差很多。我們希望能從實驗數據中比較使用簡化詞類和精簡詞類是否會有很大的差別。

Kudo and Matsumoto (2000)以資訊檢索常用的 F measure 作為評估系統的標準。F = (2* precision * recall) / (precision + recall)。由於 precision 高時則 recall 低，而 recall 高時則 precision 低，F measure 同時考慮 precision 與 recall，成為評估時的綜合指標。

表（七）是我們利用 YAMCHA 實作 Base-NP chunking 所得到的結果。

表（七）不同的標記集和 parsing 方向的辨識率

	Precision	Recall	F measure
簡化詞類 (Forward)	86.48% (10360/11980)	88.41% (10360/11716)	87.43%
簡化詞類 (Backward)	86.29% (9983/11569)	85.21% (9983/11716)	85.74%
精簡詞類 (Forward)	87.34% (8789/10063)	75.02% (8789/11716)	80.71%
精簡詞類 (Backward)	84.88% (8651/10192)	73.84% (8651/11716)	78.98%
Vote using Accuracy Rate	88.71% (10048/11327)	85.76% (10048/11716)	87.21%

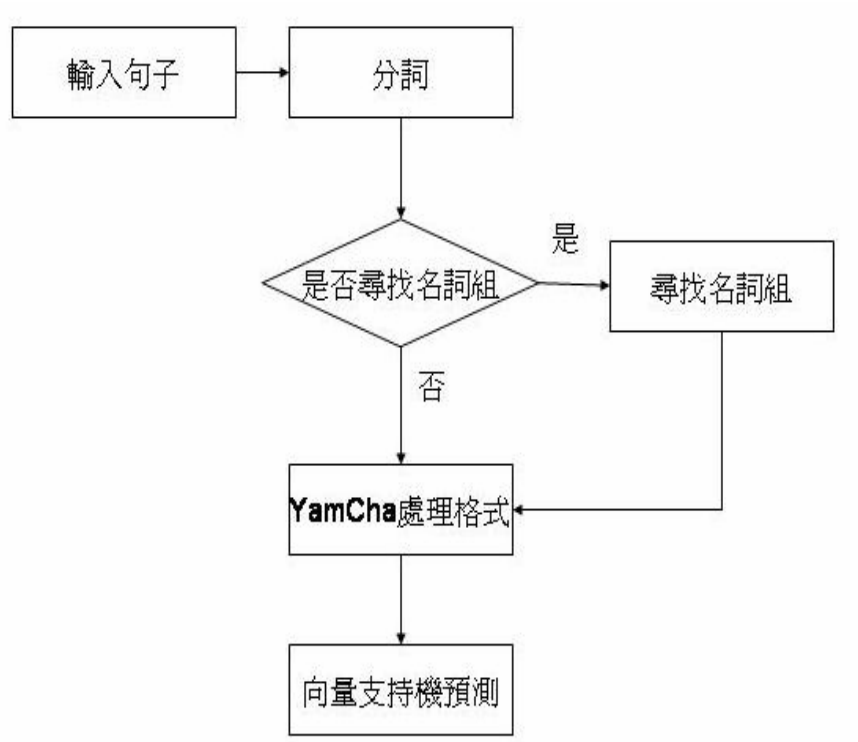
從表（七）可以觀察到 F measure 最高的是簡化詞類 forward parsing，使用 voting 並沒有提升 F measure，這是不是與訓練語料量不夠大有關，或其它因素造成，還是意味著中文只要 forward parsing 就能得到最好的效果不需要 backward parsing 和 voting，這些都有待進一步研究。值得注意的是在召回率（recall）方面簡化標記比精簡標記高 12 個百分點以上，原因是簡化標記具有 16 個動詞次分類而精簡標記動詞只有及物和不及物兩個次分類。由於精簡標記沒有足夠詳細的次

分類的特徵，導致不少基底名詞組被誤判成動詞組。如果拿表（七）最好的結果與第一次的實驗結果表（六）比較, 精確率提高了 10 個百分點, 召回率則提高了 26 個百分點，這顯示 dynamic programming 和使用 IOB 與 Start/End 發揮了功用。雖然與英文的 95% F measure 仍有一大段差距，但是辨識效能已經大幅度的提升。

拾壹、如何利用支持向量機預測中文句子依存關係

以下敘述我們如何利用中研院句法樹庫和目前常用的機器學習演算法支持向量機(Support Vector Machine)來實做一個能偵測中文句子中詞與詞之間的依存關係的剖析器(dependency parser)，並利用此剖析器來判斷名詞組。

因為中文並沒有固定的修飾方向,分詞也較為複雜,所以尋找中文的依存關係可能是一個較為困難的問題。我們的作法目前暫時不討論分詞的部份,而把重新著重在幫已經分詞好的中文句子尋找內部的依存關係。我們目前直接使用中研院的分詞系統來幫我們完成分詞這個步驟。這個方法大致的流程如下:



圖二十三 利用 Yamcha 和中文句法樹庫訓練中文剖析器流程

我們使用簡單的方法,搭配監督式機器學習(Supervised Machine Learning)來預測可能的依存關係。事實上,一般經過分詞的句子都不會太長。假設文章的有 l 個詞(terms),在 l 不夠大的時候,使用複雜的 O(l)的演算法並不一定會比簡單的 O(l2)來得快。我們將一個句子拆成 lx(l-1) 種詞的組合,並將相近的組合也當作特徵(feature),使用 R, L, O 三種類別來表示修飾的關係是左詞修飾右詞、右詞修飾左詞、或是無關係。例如「我 喜歡 唱歌」這個句子,全部就會有 3x(3-1) / 2 = 3 種可能的關係組合:

表（八）

前詞	後詞	關係
我	喜歡	R（中心語是後詞）
我	唱歌	O（無關係）
喜歡	唱歌	L（中心語是前詞）

利用向量支持機(Support Vector Machine),我們可以找出所有的組合中,哪些詞組是可能有關係的:最後再利用重建語意樹的,排除掉多餘的關係。

若 l_{LO} , l_{LR} , l_{OR} 分別表示 L 對 O,L 對 R,及 O 對 R 三種分類器的支持向量數;而 n 表示一組關係資料附帶的特徵數,則尋找一個句子的依存關係的複雜度大約是 $O(12 \times (l_{LO} + l_{LR} + l_{OR}) \times n)$。

這種方法的優點是只需要三個二元分類器(binary classifier)或是一個複類別分類器(multiclass classifier)。因為現在大多數的支持向量機工具都有提供複類別分類的功能,所以可以很簡單地架構出一個效果並不差的依存關係判斷程式。

從修飾對象、被修飾對象、以及其詞性,根據語法規則,我們可以很容易地判斷兩個詞的語法關係。如動詞後面接一個名詞,而且動詞是中心語,在沒有例外的情況下,一般都是動詞和受詞的關係。同樣地,我們也可以用類似的方式找出主詞跟動詞、動詞跟補語、修飾詞與名詞等關係。

有些例外包括介係詞「把」、「將」後面接中心語動詞,在這種情況下,「把」、「將」後面的名詞才會是後面動詞的賓語。

中文有許多名詞組很難利用詞性等特徵輕易的找出。像具備動詞＋名詞詞性組合的 「採購人員」與「採購武器」雖然詞性相同,結構卻完全不同。「採購人員」是一個名詞組,中心詞(HEADER)是「人員」;而「採購武器」是一個動詞句,中心詞是「採購」。林晏僖（2008）的實驗顯示完全不倚賴分類器的非監督式學習法(Unsupervised Machine Learning)，在開放測試中比規則式判別、監督式、以及半監督式學習法效果高許多，能夠解決許多動詞＋名詞的結構岐義的問題。林晏僖(2008)的非監督式學習法利用 Google 搜尋引擎尋找可能造成岐義的字串例句

並從例句的語境統計動詞性的特徵與名詞性的特徵哪一種較多，以較多的那一種作為判斷的依據。本文採用採用林晏僖（2008）的方法和程式作為中文剖析程式的一個模組。

我們使用 TinySVM 及 YamCha (Kudo 與 Matsumoto (2000))作為我們的支持向量機及展開資料特徵的工具並參考(Kudo 與 Matsumoto (2002))的作法。對於每一組詞有十個特徵,依序為兩個詞、兩個詞的詞性、兩個詞置、兩個詞量化前及量化後的距離、兩詞中間是否包含「的」、以及兩詞中間是否包含動詞。除此之外,前後各兩組詞的所有特徵以及前兩組詞的關係(L, R, O)也會被加入特徵中。YamCha 可以幫我們解決這一部份的資料處理。其餘處理資料、輸入輸出的部份我們使用 Perl 來實作。

監督式機器學習的部份我們使用二次多項式核心的一次漏失支持向量機(2-degree polynomial kernel L1-loss support vector machine),並設誤差項的係數為 1(C = 1)。實際訓練時間大約五天(約 125 小時)。詳細訓練語料資訊及訓練結果如下:

表（九）

句子數	43253
訓練詞組數	882708
L 對 O 分類器的支持向量數(l_{LO})	77161
L 對 R 分類器的支持向量數(l_{LR})	34490
O 對 R 分類器的支持向量數(l_{OR})	130692

排除分詞錯誤的句子後, 一共 12492 個句子, 246054 個需要預測的詞組,74850 個詞需要找出依存的對象。再沒有使用名詞組程式的情況下,平均一個句子需要的計算時間大約是 0.4 秒。

因為一個有 l 個詞的句子,每個詞最多只會修飾另一個詞,所以整個句子最多只有 l 個關係。也因為如此, lx(l-l) 個詞組中大部分都是沒有關係的(0), 所以計算正確的預測兩詞關係很容易達到很高的正確率。故這邊不討論預測兩詞關係的正確率,而討論有多少詞預測修飾的對象是正確的。結果參見下表:

表（十）

正確預測修飾對象詞	57109（76.298%）
結構完全正確的句子數	6724（53.826%）

拾貳、多義詞詞義辯識

一個詞可能有好幾個不同的意思,例如 bank 有銀行,河堤,庫等多個意義。詞義辨識的目的就是要讓電腦自動辨識一個岐義詞在某一個語境裡正確的意義。由於現有詞性標記的演算法正確率都相當的高，如果岐義詞的意義具有不同的詞性很容易透過詞性標記程式辨識出不同的意義。而像前面的例子 bank 不同的意義如銀行，河堤，庫都是名詞，辨識的困難度增高許多。我們所使用的訓練語料 Senseval-2 English lexical sample，是在 2001 年所發布，語料中包含了 73 個不同的目標詞，詞性有名詞、動詞、形容詞，但同一個目標詞的不同意義詞性都是相同的，對於詞義辨識的演算法形成很大的挑戰。

早期詞義辨識的演算法大都利用利用辭典的定義、或同義詞辭典(thesaurus)的語義分類訊息。例如 Lesk（1986）判斷目標詞的語境與辭典的哪

一個意義的定義最接近，所採用的相似度計算方式以兩者相同的非功能詞的數目為主。Walker (1987)則利用同義詞辭典(thesaurus)當中的語義類別。這些演算法跟目前常用的機器學習演算法相比正確率低許多。

機器學習方法主要可分為監督式(supervised learning)及非監督式(unsupervised learning)。兩者的差別在於前者的訓練語料有標記答案的而後者沒有，我們所採用的方法是監督式的方法。無論是哪一種機器學習的詞義辨識演算法都需要利用語境的訊息。例如 Purandare and Pedersen (2004) 採用非監督式的方法，從沒有標示詞義純文字語料抽出語境並將機讀辭典 Wordnet 裡面不同詞義的定義去除功能詞後建立共現矩陣(co-occurrence matrix)，利用 Singular Value Decomposition (SVD)將維數降到 100，最後用 Latent Semantic Indexing (LSI)找出某一句中的目標詞最有可能的詞義。Jurafsky and Martin (2000)將常用的語境特徵分成兩類。一類是搭配語特徵(collocational features)，另一類是 bag of words information。兩者的最大差別在於後者只考慮某些詞在目標詞左右一定範圍的詞有沒有出現，不考慮這些詞彼此或跟目標詞前後的關係，而前者則納入與目標詞前後相對位置的訊息，甚至用語法剖析器得到語法依存關係。

詞義辨識方法除了可以利用 Semantic Concordancer 或 Senseval 這些有標示詞義的語料之外，還可以利用 pseudoword 或雙語語料。pseudoword 是 Gale et al. (1992)和 Schutze(1992)為了省去標示詞義所需的大量人力與時間所創造出來的方法。透過人造的岐義詞如 banana-door，將語料中所有出現 banana 或 door 都代換成 banana-door，這樣就可以得到類似人工標記詞義的訓練語料。此外，某一個有岐義的詞在另一個語言通常沒有岐義，例如英文的 duty 有兩個意義，但在中文裡則由海關和責任兩個詞來表達。Brown et al. (1991) 及 Gale et al. (1992)利用這個特性，以英法雙語語料庫作為訓練語料，採取目標詞左右若干詞（例如 50 個詞）構成一個語境向量(context vector),再利用 Bayesian

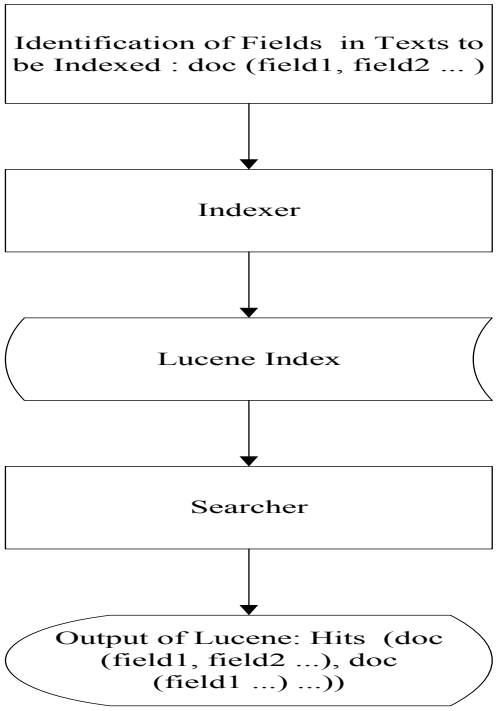
classification 來選擇在某一個語境當中哪一個詞義的機率最大。我們也採用 Bayesian classification 但搭配不同的特徵。Bayesian classification 的概念是目標詞周圍的詞會反映出目標詞的意義，因此將周圍的詞以及目標詞做統計再利用機率選擇詞義，在第三節中會有詳細的介紹。

Yarowsky (1995)注意到在某一篇文章中一個目標詞的詞義通常是固定某一個詞義(One sense per discourse)。且目標詞的搭配語提示了這個目標詞的詞義(One sense per collocation) 。本文所採用搭配語作為機器學習演算的法特徵受到 Yarowsky (1995)的啟發。Lin (1997)有鑑於以機器學習分類器(classifier)來辨識詞義需為不同的詞分別訓練出不同的分類器，頗不方便，因此提出一種使用同一種知識來源(knowledge source)的方法。他利用自己所發展的 MINIPAR 英文剖析器得到的語法依存關係(dependency relations)，如動詞與受詞的關係作為機器學習演算法的特徵。比較特別的地方在於他的方法不需要標示詞義的語料，而是利用相同語意的詞會出現在具有相同的依存關係所組成的局部語境(local context)。Lin (1997) 的正確率達到與其它機器學習演算法相同水準。有關於特徵的選取，Le and Shimazu (2004)針對英文詞義辨識提出數個特徵並以 Forward Sequential Selection Algorithm 來得到最佳的特徵組合。

除了上面介紹的方法，還有許多詞義辨識的方法，例如利用 mutual information 的 Flip-Flop algorithm (Brown et al. (1991)), 使用 decision list (Yarowsky (1994))等，限於篇幅無法一一介紹。近幾年詞義辨識的演算法除了 Naïve Bayes 之外，越來越多人使用 Maximum Entropy，Support Vector Machine，及 Conditional Random Field 等較新的機器學習演算法。

拾參、利用 Lucene 搜尋引擎檢索大量語料

Lucene 是一個以 Java 語言開發而成的全文搜尋引擎的套件，可以為各種檔製作倒置索引檔(Inverted File)，並透過簡單的 API 來檢索。Lucene 是免費開放程式碼(open source)。Lucene 可以將檔的每一個字建立索引，這樣讓搜尋就不需要逐字比對，檢索的效率可以大幅提高，Lucene 提供一組具有彈性且功能強大包括能夠解讀，過濾，分析檔，編排和使用索引的 API，讓使用者可以自訂功能。Lucene 雖然不是關連式資料庫，但可以透過類似關連式資料庫定義欄位元的方式達到關連式資料庫的功能。只要事先建立好索引檔，檢索的速度不會因為語料龐大的顯著降低檢索的速度，對於數十億詞的龐大語料庫而言，Lucene 搜尋引擎是一個不錯的選擇。下圖顯示 Lucene 欄位的建立與檢索的流程。



圖二十四 Lucene 欄位的建立與檢索的流程

拾肆、LDC 所發行的中文語料庫以及 Sketch Engine 語料庫檢索

收集大量的語料曠日廢時，且牽涉版權問題，如果加上標記語料所花的人力和物力更為可觀，美國賓州大學的 Linguistic Data Constorium (LDC)經常發行各式語料，包括 Chinese Gigaword、中文新聞和句法樹庫以及語音的語料庫。這些語料庫有些已經標注詞性、語法結構、和語義角色等訊息的，購買 LDC 的語料庫，可以大幅減少語料庫收集和開發的時間。

另一個減少語料庫開發時間的方法是使用 Sketch Engine 的服務，透過繳交個人年費約 1 千 2 百元，可以檢索 Sketch Engine 裡面 10 幾億繁體和簡體中文語料庫的內容，也可以上傳不超過 50 萬詞的語料，由 Sketch Engine 來自動分詞，建立索引，並產生關鍵詞和搭配語檢索程式。

拾伍、 結論與建議

語言的研究傳統上是純人文的研究，1950 年代後期電腦發明之後，計算語言學及自然語言處理技術這兩門新興學科誕生，語言的研究開始與科技有密切的關係。1980 年代起由於電腦軟硬體技術的突飛猛進與價格快速下降及個人電腦的日漸普及，電腦輔助語言教學系統逐漸普及。1990 年代中期以後網路興起，網頁及機讀資料的普遍使得語料庫語言學這門新興學科快速興起。語料庫，資訊檢索，

計算語言學，數位學習這幾門新興學科與英語教學形成一個橫跨語言，教育，及科技的新興研究專題。

語料庫及語言科技的重要性可以從下列事實可以看出來。英國牛津大學於1990 年代初整合英國數個研究機構發展出一億詞英文的英國國家語料庫(BNC)及檢索的工具，其它包括朗文出版社，牛津大學出版社，劍橋大學出版社，Collin出版社與伯明罕大學合作的Collins Cobuild Project，及倫敦大學為了辭典編纂學及英文文法的研究，也紛紛建置大型語料庫。我國語料庫的建設首推由中研院詞庫小組陳克健教授及黃居仁教授在1990 年代初期開始的漢語平衡語料庫及之後的句法樹庫，這些資源奠定台灣在語料庫語言學厚實的基礎，並培養了相當多年輕的語料庫語言學家。美國雖然在1960 年代即有Brown Corpus，但在英國發展了英國國家語料庫十多年之後，也開始美國國家語料庫的建設。除了英國，台灣，美國，歐洲國家，日本，大陸，幾乎世界各國都努力建置大型語料庫。

語言科技與語料庫息息相關，語言機率模型的建立需要大量語料。語料庫及語言科技相關研究早已經成為世界級學術重鎮與資訊大廠鎖定發展的重點科技。史丹福大學，柏克萊大學，麻省理工學院，卡耐基美崙大學，約翰霍普金斯大學，哥倫比亞大學，馬理蘭大學，劍橋大學，愛丁堡大學，東京大學，京都大學，北京大學無一不設有自然語言處理及語言科技的相關學程及大型研究計畫。除了學界之外，IBM 及AT&T 對於自然語言處理及語言科技的研究已經累積數十年的經驗。微軟在華盛頓州西雅圖總部及北京微軟研究院都有多組研究人員從事與語言科技產品的開發。Google 則於數年前開始網羅自然語言處理的學界菁英。語言科技對人文教育的影響最直接的一個例子就是主辦並設計托福考試的ETS 前幾年已經利用語言科技開發出英文作文自動評分系統，由於實驗顯示這套系統與專家的評分高度一致，ETS 於幾年前已將兩個專家評分減為一人評分，另一個由電腦系統取代，如果兩者差距大於某一級距才由第二位專家評分。語言科技對於國

防科技也息息相關，美國國防部先進研究計畫總署 DARPA 每年均有大筆經費支持語言科技的研究作為戰略及反恐情報收集與分析。此外，美國國家標準局 NIST 幾年前開始舉辦語言科技相關技術的競賽和評比，凡此均足以證明語言科技已經無遠弗屆，甚至無所不在。語料庫是語言科技的基礎，語言科技是語料庫的應用，兩者密切相關，缺一不可。

1988 年中研院中研院資訊所陳克健研究員及語言所黃居仁研究員成立詞知識庫小組，並規劃我國大型語料庫的建立。24 年來在他們的努力下奠定了我國大型語料庫的發展的基礎，目前除了已經完成具有詞性標記的 1 千萬平衡語料庫及數萬句的中文句法樹庫之外，還有 8 萬目詞具有語法訊息的詞庫，並且擴充了 HowNet 的語義，此外在中文分詞、詞性標記、句法剖析、語義分析的技術也打下了堅實的基礎。為了進一步促進語言科技相關產業，發揮最大的綜效，我們建議整合國科會、經濟部、教育部的經費與資源，結合產官學的力量，以語料庫語言科技為核心，研究並開發以下相關技術 1．具有自然處理技術的文本及網頁的資訊及知識擷取系統，能辯識文章裡面所包含的人、事、時、地、物資訊，並能理解文章中某些特定的語意。2. 中英及中日雙向的大型機器翻譯系統 3.能夠對於中英文作文自動偵錯及評分的系統 4.中文及多語辭典的半自動編纂系統,如半自動編纂搭配語。5.能夠自動回答問題的問答系統。這些系統所整合的技術有助於帶領相關產業開發出數百億產值的市場，開拓知識經濟新的科技服務業。

附錄

附錄一　中研院中文詞性標記集對照表

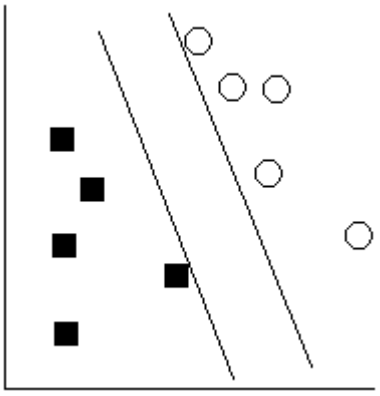
簡化詞類	代表意義	精簡詞類	簡化詞類	代表意義	精簡詞類	簡化詞類	代表意義	精簡詞類
A	非謂形容詞	A	Nb	專有名稱	N	VB	動作類及物動詞	Vi
Caa	對等連接詞	C	Nc	地方詞	N	VC	動作及物動詞	Vt
Cab	連接詞，如：等等	POST	Ncd	位置詞	N	VCL	動作接地方賓語動詞	Vt
Cba	連接詞，如：的話	POST	Nd	時間詞	N	VD	雙賓動詞	Vt
Cbb	關聯連接詞	C	Nep	指代定詞	DET	VE	動作句賓動詞	Vt
D	副詞	ADV	Neqa	數量定詞	DET	VF	動作謂賓動詞	Vt
DE	的, 之, 得, 地	T	Neqb	後置數量定詞	POST	VG	分類動詞	Vt
Da	數量副詞	ADV	Nes	數量副詞	DET	VH	狀態不及物動詞	Vi
Dfa	動詞前程度副詞	ADV	Neu	數詞定詞	DET	VHC	狀態使動動詞	Vt
Dfb	動詞後程度副詞	ADV	Nf	量詞	M	VI	狀態類及物動詞	Vi
Di	時態標	ASP	Ng	後置	POST	VJ	狀態	Vt

	記			詞			及物動詞	
Dk	句副詞	ADV	Nh	代名詞	N	VK	狀態句賓動詞	Vt
FW	外文標記	FW	SHI	外文標記	Vt	VL	狀態謂賓動詞	Vt
I	感嘆詞	T	T	語助詞	T	V_2	有	Vt
NAV	名謂詞	NAV	VA	動作不及物動詞	Vi			
Na	的, 之, 得, 地	N	VAC	動作使動動詞	Vi			

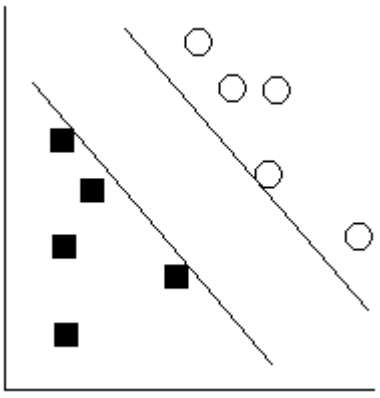
附錄二 支持向量機 Support Vector Machine (SVM) 簡介

SVM 是較新的 machine learning 技術 (Boser, Guyon, and Vapnik (1992), Corts and Vapnik (1995)) 它使用一些策略來最大化具有不同特徵的資料中間的界限, 並針對未知資料的特徵來判斷它屬於哪個類別。SVM 已在文件分類 (Joachims (1998) Taira and Haruno (1999)) 以及名詞組標示 (Kudo and Matsumoto (2000, 20001))取得超越其它作法的準確性, 而近幾年應用在自然語言處理的各個議題的研究更是方興未艾，如未知詞辨識 (unknown word guessing) (Nakagawa, Kudo, and Matsumoto (2001)) 詞性標注(part of speech tagging)Nakagawa, Kudo, and Matsumoto (2002)， Giménez J esús and Márquez Lluís (2004))句法依存關係辨識(dependency analysis)(Kudo and Matsumoto (2000))詞義辨別與標注(word sense disambiguation and sense tagging) (Cabezas, Resnik, and Stevens (2001))語意剖析(semantic parsing) (Pradhan et al. (2004) Sun and Jurafsky (2004))等都取得不錯的成果。

SVM 是一個分類用的 machine。請參照圖 (一，二),



圖一



圖二

SVM 找出兩種資料 (黑色方形與白色圓形) 中間的界限, 圖一，圖二顯示出可能的兩種分割方式, 顯然的, 後者的切割方式是較佳的 (兩種資料的界線為兩平行線之中線), 而 SVM 以滿足下面條件

$$\min\Phi(\omega)= (1/2)|\omega|^2$$

找出最佳平面（即在線性可分的情況下，可視為解二次規畫的問題），而此可由拉格朗日乘子法（Lagrange multiplier）求解。

由於很多的問題常常並不是線性可分的（如我們的詞組切割），這個時候 SVM 在比現有資料更高的向量空間 H 使用線性分類函數 $\Phi: R^d \rightarrow H$ 將 x 對應到高維空間，便可

在此以不破壞資料特徵亦不增加複雜度的方式對其進行分類.在轉換的過程中，我們會使用一 kernel function: $K(x_i,x_j) = \Phi(x_i)^T\Phi(x_j)$ 來實現非線性變換後的線性分類，而使用不同的 kernel function 對不同的資料會有不同的效果.以下為一個簡單的SVM 運作方式

給定一個訓練的資料集合:

(x_i, y_i) { i= 1, 2, ..., l; x_i 屬於 R^n; y_i 屬於 { 1, -1 }}

其中 l 為訓練之資料數，x_i 為一個 n 維向量，y_i 則是其類別（分為正類別 1 與負類別 -1）SVM 找到正類別與負類別中之最大的界限，即解決下面的最佳化問題的解答

$$\min_{w,b,e}(1/2)w^Tw + C\sum_{i=1}^{l}e_i \text{ 使得}$$
$$y_i(w^T\Phi(x_i) + b) >= 1- e_i,\ e_i >= 0$$

x_i 經由 Φ 函數被對應到一個更高維的向量空間 H之後 SVM 於此找到不同類別之間最大的界限; $K(x_i,x_j)$ 為 Kernel function.

附錄三 Bayesian Classification 簡介

以下簡述 Bayesian Classification。假設我們現在要對一個目標詞做詞義辨認，該目標詞的詞義有 k 個，依序是$s_1, s_2, \ldots, s_k$，則目標就是要找出一個s'，使得$P(s'|c)$為最大，c 是目標詞所含有的某種特徵。根據貝式定理，可以得到如下的等式：

$$P(s_k|c) = \frac{P(c|s_k)}{P(c)}P(s_k)$$

因此

$$s' = \arg\max_{s_k} P(s_k|c)$$

$$= \arg\max_{s_k} \frac{P(c|s_k)}{P(c)}P(s_k)$$

$$= \arg\max_{s_k} P(c|s_k)\,P(s_k)$$

$$= \arg\max_{s_k} [\log P(c|s_k) + \log P(s_k)]$$

相關軟體

中研院詞知識庫小組中文斷詞系統(包含未知詞擷取與標記)
http://ckipsvr.iis.sinica.edu.tw/

AntConchttp://www.antlab.sci.waseda.ac.jp/software.html

Hownet http://www.keenage.com/

Mate Parser http://barbar.cs.lth.se:8091/

Penn Chinese Treebank
http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2004T05

Sinica Treebank 3.0. http:// www.aclclp.org.tw/use_stb_c.php

Sketch Engine http://www.sketchengine.co.uk/

Stanford NLP Softwarehttp://nlp.stanford.edu/software/index.shtml

Stanford Parser http://www-nlp.stanford.edu/downloads/lex-parser.shtml

Tools for Natural Language Analysis, Generation and Machine Learning
http://code.google.com/p/mate-tools/

YamCha: Yet Another Multipurpose CHunk Annotator
http://chasen.org/~taku/software/YamCha/

參考文獻

Bikel, Daniel. (2004). On the Parameter Space of Generative Lexicalized Statistical Parsing Models. Ph.D. Dissertation. University of Pennsylvania.

Bird, Steven, Klein, Ewan and Loper Edward (2009) *Natural Language Processing with Python*. O'Reilley.

Chiang, David. (2003) Statistical parsing with an automatically extracted tree adjoining grammar. In *Data Oriented Parsing*, CSLI Publications, pages 299–316.

Levy, Roger and Manning, Christopher D. (2003). Is it harder to parse Chinese, or the Chinese Treebank?. *ACL 2003*.

Manning, Christopher, and Schutze, Hinrich. (1999) Foundations of Statistical Natural Language Processing. MIT Press.

Xue, Nianwen and Xia, Fei. (2000) "The Bracketing Guidelines for the Penn Chinese Treebank (3.0)", *IRCS Report 00-08*, University of Pennsylvania, Oct 2000.

Argamon, Shlomo, Dagan, Ido, and Krymolowski,Yuval (1998). A Memory-Based Approach to Learning Shallow Natural Language Patterns. In Proceedings of the 17th international conference on Computational linguistics, Vol. 1, pp. 67 - 73 , Montreal, Quebec, Canada."

Brill, Eric and Ngai, Grace (1999), Man vs. Machine: A Case Study in Base Noun Phrase Learning. In Proceedings of ACL'99, pp. 65-72, University of Maryland, MD, USA.

Boser, E. Bernhard, Guyon, Isabelle, and Vapnik, Vladimir. (1992). A Training Algorithm for Optimal Margin Classifiers. COLT: pp. 144-152

Cabezas, Clara, Resnik, Philip, and Stevens,Jessica. (2001). Supervised Sense Tagging using Support Vector Machines. Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems (SENSEVAL-2), Toulouse, France, 5-6 July 2001.

Cardie, Claire and Pierce, David (1998). Error-Driven Pruning of Treebank Grammars for Base Noun Phrase Identification. In Proceedings of COLING-ACL'98, pp. 218-224, Montreal, Canada.

Chang, Chih-Chung and Lin, Chih-Jen. (2004) LIBSVM -- A Library for Support Vector Machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Chen, Kuang-hua and Chen, Hsin-Hsi (1994). Extracting Noun Phrases from Large-Scale Texts: A Hybrid Approach and Its Automatic Evaluation, In Proceedings of ACL-94, Las Cruses, NM, USA.

Church, K. (1988) A Stochastic Parts Program and Noun Phrase Parser for Unrestricted

Text. *Second Conference on Applied Natural Language Processing*, Austin , Texas , pp. 136-143.

Corte, Corinna, and Vapnik, Vladimir (1995). Support-Vector Networks. Machine Learning 20(3), pp. 273-297.

Giménez J esús and Márquez Lluís (2004). SVMTool: A general POS tagger generator based on Support Vector Machines Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04). Lisbon, Portugal. 2004 .

Joachims, Thorsten. (1998) *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*. Proceedings of the European Conference on Machine Learning (ECML), Springer, 1998.

Hsu, Chih-Wei, Chang, Chih-Chung, and Lin, Chih-Jen. (2004). A Practical Guide to Support Vector Classification.

Kudo, Taku, and Matsumoto, Yuji. (2000). Use of Support Vector Learning for Chunk Identification. In Proceedings of CoNLL-2000, pp. 142-144.

Kudo, Taku, and Matsumoto, Yuji (2000). Japanese Dependency Analysis Based on Support Vector Machines, EMNLP/VLC 2000

Kudo, Taku, and Matsumoto, Yuji. (2001). Chunking with Support Vector Machine. In Proceedings of NAACL 2001, pp. 192-199.

Marcus, Mitchell P., Santorini, Beatrice and Marcinkiewicz, Mary Ann. (1993) Building a large annotated corpus of English: the Penn Treebank, Computational Linguistics, 19:2. vol. 19, no. 2, pp. 313–330.

Pradhan, Sameer, Ward, Wayne, Hacioglu, Kadri, Martin, James H.and Jurafsky, Daniel. (2004). Shallow Semantic Parsing Using Support Vector Machines. In Proceedings of NAACL-HLT 2004,pp. 233-240..

Nakagawa, Tetsuji, Kudo, Taku, and Matsumoto, Yuji. (2001). Unknown Word Guessing and Part-of-Speech Tagging Using Support Vector Machines. NLPRS, pp. 325-331

Nakagawa, Tetsuji, Kudo, Taku, and Matsumoto, Yuji. (2002). Revision Learning and its Application to Part-of-Speech Tagging. In Proceedings of ACL 2002, pp. 497-504.

Ramshaw, Lance A., and Marcus, Mitchell P.. (1995). Text Chunking Using Transformation-based Learning. In Proceedings of the Third ACL Workshop on Very Large Corpora, pp. 82-94, Cambridge MA, USA.

Skut, Wojciech and Brants, Thorsten. (1998) A Maximum-Entropy Partial Parser for Unrestricted Text. In Proceedings of the Sixth Workshop on Very Large Corpora, pp. 143-151, Montreal, Canada.

Sun, Honglin and Jurafsky, Daniel. 2004. Shallow Semantic Parsing of Chinese. In Proceedings of NAACL-HLT 2004, pp.192-199.

Taira, Hirotoshi, Haruno, Masahiko（1999）: Feature Selection in SVM Text Categorization. AAAI/IAAI 1999, pp. 480-486.

Tjong Kim Sang, Erik F. and Veenstra, Jorn (1999). Representing Text Chunks. In Proceedings of EACL'99, 173-179, Bergen, Norway.

Tjong Kim Sang, Erik F. (2002) Memory-Based Shallow Parsing. Journal of Machine Learning Research, Vol. 2, pp. 559-594.

Uchimoto, Kiyotaka, Ma, Qing, Murata, Masaki, Ozaku, Hiromi, Isahara, Hitoshi. (2000) Named entity extraction based on a maximum entropy model and transformation rules. Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, Hong Kong, pp. 326 – 335.

Veenstra, Jorn. (1998). Fast NP chunking using memory-based learning techniques, In F. Verdenius and W. van den Broek eds.,Proceedings of BENELEARN-98, pp. 71-79, Wageningen, The Netherlands.

Voutilainen, A. (1993) NPtool, a Detector of English Noun Phrase. In Proceedings of the First Annual Workshop on Very Large Corpora, pp. 48-57. Berners-Lee, Tim. (2000) Weaving the Web : the original design and ultimate destiny of the World Wide Web by its inventor. New York : HarperBusiness.

Boguraev, Branimir. and Briscoe, Ted. (1989) Computational Lexicography for Natural Language Processing. Longman: Harlow. Boguraev, Branimir and Pustejovsky, James (eds.) (1996) Corpus Processing for Lexical Acquisition, MIT Press.

Chaffin, Roger and Illerrmann, Douglas. (1988) The Nature of Semantic Relations: a Comparisons of Two Approaches. In Evens (eds) (1988), pp. 289-334.

Church, K. and Hanks, P. (1990) “Word Association Norms, Mutual Information, and Lexicography.” Computational Linguistics, Vol. 16, No. 1, pp. 22-29.

Church, K. et al. (1991) “Parsing, Word Associations, and Typical Predicate-Argument Relations.” In Tomita (ed) *Current Issues in Parsing Technology*, Kluwer.

Church, Kenneth, William Gale, Patrick Hanks, and Donald Hindle. (1994) ‘Lexical Substitutability,’ in Atkins and Zampolli (eds.) Computational Approaches to the Lexicon, pp. 153- 177. Oxford, Oxford University Press.

Cruse, Allan. (1986) Lexical Semantics. Cambridge: Cambridge University Press.

Dong, Zhendong and Dong, Qiang. (2006) Hownet and the Computation of Meaning. World Scientific.

Evens, Martha. (eds.) (1988) Relational Models of the Lexicon: Representing Knowledge in Semantic Networks. Cambridge University Press.

Fillmore, Charles. (1968) The Case for Case. In E. Bach and R. T. Harms, eds., Universals in Linguistic Theory, Holt, Riinehart and Winston, New York, 1-88.

Koenig, Jean-Pierre. (1999) Lexical Relations. CSLI , Stanford University.

Girju, R., Nakov, P., Nastase, V., Szpakowicz, S., Turney, P., and Yuret, D. (2007), SemEval-2007 Task 04: Classification of Semantic Relations between Nominals, *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval 2007)*, Prague, Czech Republic, pp. 13-18.

Grefefenstette, Gregory. (1994) Explorations in Automatic Thesaurus Discovery. Kluwer Academic Publishers.

Hearst, M.A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics*, pages 539–545, Nantes, France.

Levin, Beth. (1985) 'Introduction,' in B. Levin (ed.) Lexical Semantics in Review, Lexicon Project Working Papers 1, Center for Cognitive Science, MIT, pp. 1-62.

Melcuk, Igor. (1988) 'The Explanatory Combinatory Dictionary,' in M. Evens (ed.) (1988), pp. 41 - 74.

Pustejovsky, James, Sabine Bergler, and Peter Annick (1993) 'Lexical Semantic Techniques for Corpus Analysis,' Computational Linguistics, Vol. 19, No. 2, pp. 331 - 358.

Pustevojsky, James. (1995)The Generative Lexicon. The MIT Press.

Pustevojsky, James. (2000) Syntagmatic Processes. in Handbook of Lexicology and Lexicography**,** de Gruyter, 2000.

Jackendoff, Ray. (1983) Semantics and Cognition. Cambridge, Mass.: MIT Press.

Jackendoff, Ray. (1990) Semantic Structures. Cambridge, Mass.: MIT Press.Jones, Stevens. (2002). Antonymy: A Corpus-based Perspective. London ; New York : Routledge, 2002

Pedersen, Patwardhan, and Michelizzi (2004) WordNet::Similarity - Measuring the Relatedness of Concepts - Appears in the *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, pp. 1024-1025, July 25-29, 2004, San Jose, CA (Intelligent Systems Demonstration)

Resnik, Phillip. (1992) 'WordNet and Distributional Analysis: A Class-based Approach to Lexical Discovery,' in Workshop Notes, Statistically-Based NLP Techniques, American Association for Artificial Intelligence, pp. 109 - 113.

Schank, Roger. (1975) Conceptual Information Processing. Amsterdam: North-Holland.

Sinclair, John. (eds). (1987) Looking up. Glasglow: Collins.

Turney, P.D. (2006), Expressing implicit semantic relations without supervision, *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (Coling/ACL-06)*, Sydney, Australia, pp. 313-320.

Wilks, A. Yorick (1968) On-line Semantic Analysis of English Texts. Machine Translation,

Vol. 11, pp. 59-72. Brown, Peter et al. (1991) Word sense disambiguation using statistical methods. In ACL 29, pp. 264-270.

Dong, Zhendong and Dong, Qiang. (2006) Hownet and the Computation of Meaning. World Scientific.

Gale, William, Church, Kenneth, and Yarowsky, David. (1992) A method of disambiguating word senses in a large corpus. Computers and the Humanties 26:415-439.

Jurafsky, Daniel, and James H. Martin. (2000) Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics. Prentice-Hall.

Klein, Dan. and Manning, Christopher. (2003) Accurate Unlexicalized Parsing. Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430.

Le, Cuong Anh and Shimazu, Akira. (2004) High WSD Accuracy Using Naïve Bayesian Classifier with Rich Features. PACLIC 18, Tokyo. http://dspace.wul.waseda.ac.jp/dspace/bitstream/2065/564/1/oral-8.pdf

Lesk, Michael. (1986) Automatic Sense Disambiguation: How to tell a pine cone from an ice cream cone. In Proceedings of the 1986 SIGDOC Conference, pp. 24-26, New York. Association for Computing Machinery.

Lin, Dekang . (1997). Using Syntactic Dependency as Local Context to Resolve Word Sense Ambiguity In *Proceedings of ACL-97,* Madrid, Spain. July, 1997.

Manning, Christopher, and Schutze, Hinrich. (1999) Foundations of Statistical Natural Language Processing. MIT Press.

Patwardhan, Banerjee, and Pedersen (2005) SenseRelate::TargetWord - A Generalized Framework for Word Sense Disambiguation. Appears in the Proceedings of the Twentieth National Conference on Artificial Intelligence, July 12, 2005, Pittsburgh, PA. (Intelligent Systems Demonstration)

Purandare and Pedersen (2004) Improving Word Sense Discrimination with Gloss Augmented Feature Vectors. Appears in the Proceedings of the Workshop on Lexical Resources for the Web and Word Sense Disambiguation, November 22, 2004, Puebla Mexico.

Yarowsky, D. (1994) Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French.'' In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics.* Las Cruces, NM, pp. 88-95.

Zhao,Jun and Huang, Changning. (1998). A Quasi-Dependency Model for Structural Analysis of Chinese BaseNPs. In Proceedings of COLING-ACL 98, pp. 1-7 , Montreal,

Canada.

高照明(2007) *中文詞彙語意資料的整合及擷取：詞彙語意學的觀點*。第十九屆自然語言與語音處理研討會論文集, pp. pp 257-272。台北。

高紹航，高照明(2007) *詞義辨識:機器學習演算法特徵的選取與組合*。第十九屆自然語言與語音處理研討會論文集, pp 131-144。台北。

張席維，高照明，劉昭麟 （2005）*利用向量支撐機辨識中文基底名詞組的初步研究*。第十七屆自然語言與語音處理研討會。pp. 317-332

黃子桓，高照明（2005）*基於統計與佚代的中英雙語語料詞與小句對應演算法*。第十七屆自然語言與語音處理研討會。pp. 385-396.

林語君 高照明（2004）*結合統計與語言訊息的混合式中英雙語句對應演算法*。第十六屆自然語言與語音處理研討會論文集。台北。

魯川 (2001) *漢語語法的意合網路*。北京：商務印書館。

「中文句結構樹資料庫」(Sinica Treebank Version 3.0). 中華民國計算語言學會 http://www.aclclp.org.tw/use_stb_c.php

中文詞類分析 (1988). *中央研究院詞知識庫小組技術報告*,台北。