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%。

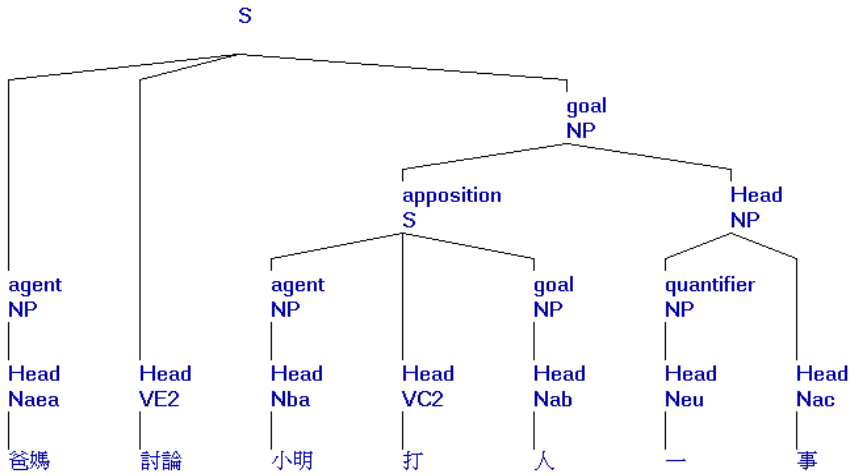
中文名詞組辨識的研究起步較晚，迄今只有零星的研究，還沒有針對同一個語料庫的大規模的測試與比較。例如中國大陸學者 Zhao and Huang (1998)提出以語料庫統計結合規律，利用 minimum description length principle (MDL)得到 quasi-dependency strength 加上規律來得到 base NP。這種採用非監督式機器學習（unsupervised learning）的方法，在封閉測試(close test)和開放測試(open test)中分別有 91.5% 和 88.7%的精確率。

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: