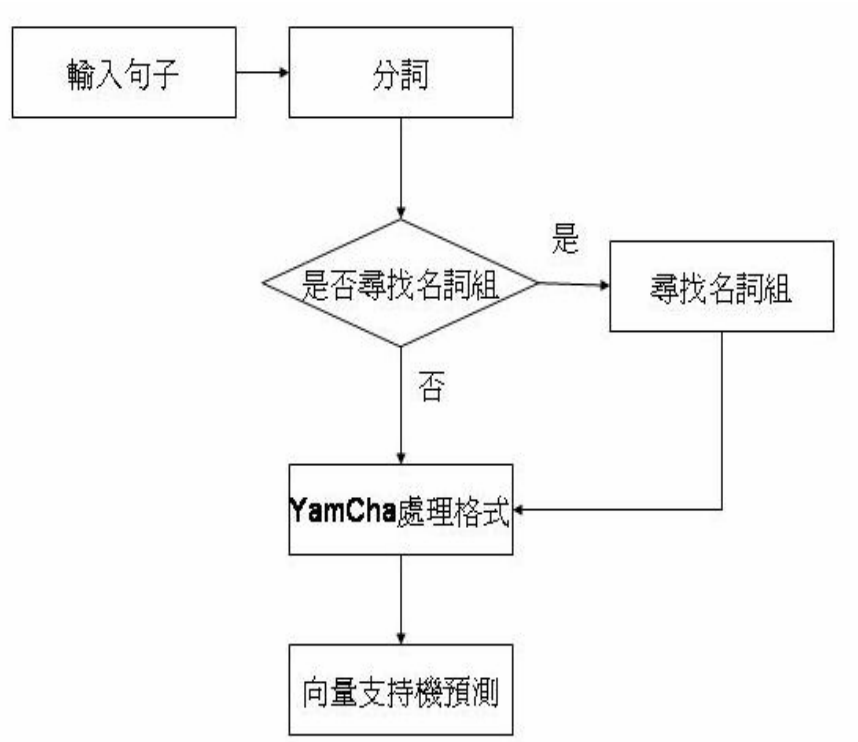
分類的特徵，導致不少基底名詞組被誤判成動詞組。如果拿表（七）最好的結果與第一次的實驗結果表（六）比較, 精確率提高了 10 個百分點, 召回率則提高了 26 個百分點，這顯示 dynamic programming 和使用 IOB 與 Start/End 發揮了功用。雖然與英文的 95% F measure 仍有一大段差距，但是辨識效能已經大幅度的提升。

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