連接詞的語法表達模式 -
以中文訊息格位語法(ICG)爲本的表達形式

魏文眞　　陳克健
中央研究院資訊科學研究所

摘　要

本論文提出一個有關連接詞的語法表達模式，這個語法表達模式是根據訊息爲本的格位語法(ICG)所架構，依連接詞的特性分兩種方式處理，一爲以連接詞爲中心語，組成的成分由所連接的語法結構取代；二爲以連接詞爲修飾語，由連繫特徵照應配對關係。這一個語法表達模式使中文自然語言處理由簡單句向複雜句跨前一步。

一、前言

語言是人類用以表達思想的工具，簡單的意思用簡單句表達即可，但是若要完盡的表達人類思想中繁複的語意，必須藉諸較複雜的語言機制。自然語言發展的過程與人類語言習得歷程相近：由簡單而複雜，由理解而創造。所以在作自然語言處理時，得先考慮語言的基本構件--個別詞彙，再由個別詞彙的訊息聯結與語法組合，完成一個簡單句的分析。若是想對人類語言做較完整全面的處理，必定會面臨複句的問題。

連接詞能連接兩個或兩個以上的語言單位，組成較大的語言單位。本文的目的在探討由連接詞造成的複合形式(包括複句和詞組)，嚐試提出一個適用於自然語言處理的表達形式。我們之所以把討論範圍限定在連接詞，是想簡化自然語言的處理問題，因爲複句的形式關係極爲複雜，必須配合人的理解能力及常識才能詮釋清楚，而目前中文自然語言處理還在起步階段，比較可行的方向是先針對有標記的情形加以研究，再考慮零標記的情形。而連接詞本身是組成複句的明顯標記，是由簡單句過渡到篇章的重要橋樑，辨識出連接詞及其背後代表的意義，可簡化複句結構，對句與句的關係大略可釐清，所以實有討論的必要。

因爲連接詞是組成複句的重要成分，本文首先將就複句的形式作簡單的討論，兼及其他可能造成複句的關聯成分作說明，其次專門討論連接詞的類別，接著將根據連接詞的語法語意特性,以陳與黃[6]的訊息爲本的格位語法表達模式(ICG)爲基礎,提出一套適用於中文自然語言處理的連接詞表達形式，最後將舉實例說明實際運用於剖析時的策略。

二、複句和複句的分類

在語言的構件中，由小至大包括詞素、詞、詞組、句子、句段五種單位，各樣繁雜的語言形式都是由以上五個單位組合而成。基本上，人類在進行訊息溝通時，是以句子爲使用單位，而句子是由詞或詞組的訊息累積組合成，可以簡單到只包含一個詞或詞組(註1)，也可能長至整個段落只是一個句子。

一般而言，句子分爲簡單句和複句，簡單句的構成形式較簡單，只有一個謂語中心，而一般對複句所下的定義，是指由兩個或兩個以上的單句所組成，能夠表達一個比較複雜意思的語言單位，其中被組合在複句裡的單句叫做分句[3]。由上面的定義可知，複句是由單句擴展而來，而且組合在複句裡的單句已改稱分句。複句也可以再擴展，分句本身就包含複句，形成多重複句。

根據分句之間地位關係的不同，複句可分爲聯合複句和偏正複句。聯合分句中各分句地位平等，沒有主次之分，在「現代漢語複句」[5]裡，依照語義把聯合複句分爲並列、承接、選擇、遞進、總分、對比等六類。

(1) 天這麼黑，風這麼大。（並列）
(2) 他喝了盅熱茶，戴上軟帽，站了起身，緩緩走出門去。（承接）
(3) 要麼你就認命，要麼你就奮力一搏。（選擇）
(4) 小妮子不但不緊張，反而忙著安慰人。（遞進）
(5) 成熟的麥子各有歷程：有的變成盤中飧，有的掉進泥土重新復活，有的沉入了溝底。（總分）
(6) 他不是笨，而是懶惰。（對比）

偏正複句則是指構成的兩分句有偏正、主次之分，偏句爲主句服務，對主句有說明、限制的作用。根據分句所表現的關係，吳[3]把偏正分句分爲轉折、因果、假設、條件(註2)、取捨、目的六種關係，而在劉[5]中未列取捨關係，卻另列連鎖關係(註3)。

(7) 雖然我心裡明白，嘴裡卻不願說破。（轉折）
(8) 他因爲身材高大，被選作鼓手。（因果）
(9) 天若有情天亦老。（假設）
(10) 只要他沒意見，我們就照計行事。（條件）
(11) 他寧可叫人指指點點，也不願剪去辮子。（取捨）
(12) 早點動手，省得措手不及。（目的）
(13) 根埋得越深，樹長得越好。（連鎖）

單句形成複句的方式有兩種：一種是直接組合法，另一種是關聯組合法。直接組合法是指兩個以上的單句在意義上有一定的聯繫，就可以把它們直接組合在一起，構成複句[3]，例如(1)的並列複句，表示幾個相關的事物或情況，(2)的連續複句則是依照分句次序順著往下說；至於關聯組合法，是使用關聯語詞把兩個或兩個以上的句子組合起來，構成複句的方法，如例(7)、(8)、(9)、(10)表現轉折、因果、假設、條件等關係，是根據分句關係借助關聯語詞加在複句中。關聯組合法的句子有時沒有關聯語詞，仍能表現出其中的聯繫關係，如(7)句的「雖然」、(8)的「因爲」是可以省略的。雖然用不用關聯詞沒有一定的標準，但站在自然語言處理的觀點，有關聯詞的複句，電腦較易掌握分句間的關係。而且有些情況最好能選用關聯詞語：

(一) 不用關聯語詞的話，可能有幾種不同關係。例如 "資訊不透明，勞工糾紛不斷" 這個句子，就可有下列幾種情況：

(14a) 資訊不透明，而且勞工糾紛不斷。
(14b) 因爲資訊不透明，所以勞工糾紛不斷。
(14c) 如果資訊不透明，就會勞工糾紛不斷。
(14d) 只要資訊不透明，就會勞工糾紛不斷。

(14a)表現的是並列關係，(14b)是因果關係，(14c)是假設關係，(14d)是條件關係。原句內容隱含不明，有了關聯詞，就能明確的表現出兩分句的意含。

(二) 多重複句關係較爲複雜，選用適當的關聯詞語，能明確清楚表達各層關係。例如：

(15) 從1960年起，蘇俄與中共一直處於激烈的競爭狀態，雙方都想稱霸共產世界，因此，毛澤東時代的結束，對克里姆林宮的執政者而言，自然如釋重負。但是，當一九七八年中共開始將市場機能引進中共集權經濟制度後，蘇俄則猛烈的批評中共欲將共產主義帶回「資本主義的道路上」。 (天下79期頁189)

在(15)中用了「因此」、「但是」等關聯詞表達分句間的關係，(15)本身是個多重複句，在複句最外層表現的是轉折的關係，由關聯詞「但是」標識出，前一個分句本身也是個複句，運用關聯詞「因此」表達了因果關係。因爲有關聯詞的使用，使得分句間的連繫密切，結構的層次也較爲清楚。如若去掉了關聯詞，則結構鬆散，無法交代出前後分句的關係。

三、關聯詞語

既然關聯詞語對行文的結構頗具影響力，下面我們將對關聯詞語的定義及類別加以討論。"關聯"是指事物之間互相發生牽聯和影響，故關聯詞語是指能使各級語言單位起互相關聯的語詞[5]。 關聯語詞適用的範圍不僅止於句與句的聯繫，只要是能把兩個或兩個以上較小的語言單位連接起來，組成一個較大的語言單位，這樣的詞就可以說是關聯詞語，所聯繫的可以是詞和詞，也可以是詞組和詞組。

一般而言，能起關聯作用的詞多半是連接詞，前面所舉的例子大部分屬之，常見的連接詞，如：

雖然　　但是　　如果　　因爲　　所以　　不但　　而且

除了連接詞外，有的副詞、詞組也能作關聯詞。能作關聯成分的副詞，常見的有：就、便、才、接著、卻、都。例如：

(16) 他看看屋子悶熱，就把所有的窗子敞開。
(17) 弟弟聰明活潑，哥哥卻沉默寡言。

一般對連接詞及具關聯作用副詞的區辨辦法，是以出現在主語前後位置做判斷，只能出現在主語後邊的是副詞，出現在主語前，也可出現在主語後的是連接詞[1,10]。例：

(18) 他雖然没來，禮卻到了。
(19) 雖然他没來，禮卻到了。
* (20) 我今天去找他，卻他不在。

「雖然」是連接詞，可以在主語前後出現；「卻」是副詞，(20)句出現在主語前，造成了錯誤的句子。

另外，能作關聯語詞的詞組如：

一方面　　另一方面　　總而言之　　換言之　　據說

因爲缺少「關聯短語」「關聯狀語」之類的名稱，所以籠統的把「一方面、另一方面」等叫做連接詞。不過「總而言之、換言之」等詞，僅是語義上的承前，在句法上並無承接功能，所以在我們的系統中以句副詞視之[4]。

劉[5]列有第四種表關聯的成分，認爲判斷詞"是"和包含判斷詞"是" 的詞組合起來有時也具關聯作用。例：

是....還是　　是....而不是　　不是....就是

下一節我們將針對連接詞的類型作一番討論。

四、連接詞的分類

連接詞是用以表示並列關係或標明兩分句關係的詞，根據連接詞的功能與定義，我們把連接詞分爲兩大類，一是並列連接詞，一是關聯連接詞。

(一)並列連接詞(Ca) -- 連接兩個詞性相似的成分形成向心式結構，其中每一個成分的功能都跟整個結構相同。這一類的連接詞帶有的語義特徵可分爲：

1).複數性(+plural)，如：和、跟、與、同、及；
2).選擇性(+disjunction)，如：或、或者、還是；
3).範圍性(+range)，只有「至」和「到」。

「又」除了常見的副詞用法，如：「他又胖了」表示重複的意思外，另可連接兩個狀態動詞(即一般所稱的形容詞)，如「介壽路長又寬」，意思相當於英文的'and'，雖然不能出現在主語前，也可視同表複數意義的連接詞，「且」有時也具有類似的功用。

標點符號「、」也具有並列連接的功能，比較有意思的是它兼有複數及選擇兩種可能，得視情況而定。

(21)　他喜歡的運動有籃球、滑草、溜冰及游泳。
(22)　你想喝咖啡、果汁還是紅茶?
(23)　你想搭車、走路?
(24)　不管晴天、雨天都可看見他掃大街的身影。

「、」的功能常常可由出現在其後的並列連接詞決定，出現在(21)的連接詞是具複數特徵的「及」，所以「、」也帶 "+plural" 特徵，而(22)句中的「、」則與「還是」一樣，帶 "+disjunction" 特徵。有時「、」後面並沒有任何並列連接詞以資辨別，通常作複數性大致不會錯，不過仍有可能是選擇性，可藉帶疑問的標點符號「?」判定，如(23)。出現在連接詞「不論、不管、無論」後的「、」也具選擇性，因爲這些連接詞後要求帶疑問語意的成分，如(24)。「，」除了作分句標記外，有時也與「、」有相同的語法現象，可替代「、」的功能。

(25) 計算中心分行政組，技術組，操作組，及服務組。
(26) 他是闖紅燈，超速還是違規停車?

帶範圍特徵的連接詞用以連接數量定詞或定量式複合詞，表現數量的範圍，例如:

(27) 平均溫度上升三到四度。
(28) 四月四日至七日放假。

並列連接詞的組合成分在句中可以扮演任何語法角色，可能是單句的謂語，也可能是主語、賓語、狀語或是定語。其角色是由所連接的成分及連接成分出現的位置而定。

(29) 今晚的月亮［大又圓］。 (謂語)
(30) ［陽光和水分］都有助植物的生長。 (主語)
(31) 他早餐吃［燒餅油條和豆漿］。 (賓語)
(32) 請你［今天或明天］給我一個答覆。 (時間狀語)
(33) ［正確而且詳實］的資料 (定語)

(二)關聯連接詞(Cb)--能夠把幾個分句連成複句形式的連接詞。句子的連繫方法有前繫(forward linking)、後繫(backward linking)兩種 [13]，前繫是一個分句依賴後面的句子使語意完整，後繫則是位於後面的子句靠前一子句使語意完整。關聯組合式的複句在分句中至少有一個連繫成分，連接詞以其所在分句位置可分爲前繫連接詞、後繫連接詞。如果組合成的是偏正複句，通常有前繫連接詞的分句可具移動性，能移位至後面。不過也有少數例外的情形，如「雖、固然、既」。

(34) 因爲生命有限，要好好把握時間。

(35) 要好好把握時間，因爲生命有限。

(36) 現在雖是正午，陽光並沒有出現。

* (37) 陽光並沒有出現，現在雖是正午。

由連接詞接繫方式及移位情形,可把關聯連接詞分爲三類(參見表一、二):

1.移動性前繫連接詞(Cba) -- 語意上具起頭作用，後面常須接一個分句，可移位於後面。這類詞常見的有:

1). 偏正句移動性連接詞(Cbaa)

轉折	雖然		'+concession'
	儘管		
因果	因為		'+reason'
	由於		
	既然		
假設	即使		'+uncondition'
	縱然		
	哪怕		
	任憑		
	如果	若	'+hypothesis'
	要是	假若	
	一旦	萬一	
條件	只有		'+conditin'
	除非		
	只要		
	不論		'+whatever'
	無論		
	不管		

2). 偏正句句尾連接詞(Cbab)

這一小類只有「的話」一個詞，帶 "+hypothesis"特徵。

2.非移動性前繫連接詞 -- 語意上具起頭作用，後面常須接一個分句，位置固定在前一分句。聯合複句的前繫成分均屬此類，前面提及部分偏正前繫連接詞不可移動也屬這類，故此類連接詞可分爲兩類:

1)．偏正句非移動性前繫連接詞(Cbba)

轉折	雖	'+concession'
因果	既	'+reason'
	之所以	'+reasult'
假設	就是	'+uncondition'

2)．聯合句前繫連接詞(Cbbb)

選擇	要麼	
	與其	'+rejection'
遞進	不但	'+restriction'
	不僅	
	不只	
並列	一來	'+listing'
	首先	

3.後繫連接詞(Cbc)-- 是將一個分句聯繫於前一個句子的連接詞，依其組合方式是偏正、聯合分爲二類：

1)．偏正句後繫連接詞(Cbca)

轉折	可是	'+contrast'
	不過	
因果	所以	'+result'
	以致	
假設	那麼	'+conclusion'
	則	
條件	否則	'+inversion'
目的	省得	'+avoidance'
	以免	
	以便	'+purpose'
取捨	不如	'+selection'

(表一)偏正複句的連接詞

	前繫		後繫
	移動性	非移動性	
轉折	雖然 儘管 +concession	雖 固然	可是、然而、不過 但是、但 +contrast
因果	因、因為 由於、既然 +reason	既	所以、是以、以致 +result
		之所以 +result	
假設	縱使、縱然 縱使、就算 +uncondition	就是	
	如果、要是 假如、的話 +hypothesis		那麼、那、則 +conclusion
條件	只有、唯有 除非 +condition		否則 +conversion
	不論、無論 不管 +whatever		
目的			省得、免得 +avoidance
			以、以便、好 +purpose
取捨		與其 +rejection	不如、倒不如 +selection

(表二)聯合複句的連接詞

選擇	要麼、要不 +alternative	要麼、要不 +alternative
遞進	非但、不獨、不但、 不僅 +restriction	而且、並且、且、 反而 +addition
並列	首先、一來、一方面 +listing	其次、二來、二方面 +listing

2)．聯合句後繫連接詞(Cbcb)

選擇	不若	'+selection'
	寧可	
遞進	而且	'+addition'
	並且	
	並	
	且	
	反而	
並列	二來	'+listing'
	其次	
	另一方面	
	二方面	
	此外	

五、連接詞的語法表達模式

陳與黃[6]中所提的訊息爲本的格位語法(ICG, Information Based Case Grammar)是採用詞彙爲中心的表達方式，將每個詞的語法及語意訊息以特徵結構表示。詞彙結合爲片語，再由片語組成句子，層層累加堆積而成。在建構片語及句子時，是以中心語驅動的方式，所有成分的結合都必須符合詞彙所規定的語法限制。根據中心語主導原則(head-driven principle),中心語的語意訊息欄內規定由此中心語組成的成分，包括中心語本身所帶有的論元角色及附加成分。中心語若爲動詞,則形成動詞片語(VP)；若爲名詞,則形成名詞片語(NP)；若爲介詞，則形成介詞片語(PP)；若爲方位詞，則形成方位詞片語(GP)。不過，也有一些詞類只作爲修飾語，並不形成任何片語結構的；例如：副詞、語助詞、感歎詞等。連接詞比較特殊，並列連接詞做中心語比較方便，但是關聯連接詞不好做爲中心語，因爲：其一．前繫連接詞可出現在句主語前後，難以表達連接詞片語的論元、中心語間的詞序關係。其二．關聯連接詞連繫兩個以標點符號分開的句子而一般語法皆以標點符號之間的句子或片語爲表達的段落單位，因此，我們先討論各別句子中連接詞的表達。我們以兩種不同的類型Ca、Cb分別處理，下一節再討論複句間連接關係建立的方法。

1.連接詞爲中心語(Ca)

並列連接詞爲中心語所表達的連接詞片語在句中的作用和其連接成分的作用完全相同，例如 (29-33)。因此，即使並列連接詞爲中心語，但所形成的連接詞片語在剖析後，將以連接成分的語法結構取代。

例(28)中 " 四日至七日" 爲 DM(Determinative Measure Compound)
例(29)中 " 大又圓" 爲 VP
例(30)中 " 陽光和水分" 爲 NP

連接成分間有語法結構及語義一致性(agreement)的特性。我們採用一個特殊的特徵 AGREE 來表示兩個成分之間的一致性關係(註4)。

(38) 並列連接詞的詞彙訊息

```
/ * 例如: 和, 跟, 與, 或,'、' 等  * /

語意訊息 [ 語義 : and/or
           特徵 : DUMMY1 features, DUMMY2 features
           論元 : [ DUMMY1
                    DUMMY2 ]

語法訊息 [ 分類 : Caa   /* 組成成分語法結構與 DUMMY 的結構相同 */
           形式 : [ DUMMY1 [{NP, VP, PP, GP, S}]
                    DUMMY2 [{NP, VP, PP, GP, S}] ]
           線性律 : { DUMMY1 [AGREE[sem, Cat]] << * << DUMMY2 [AGREE
                      [sem, Cat]]
                    { DUMMY1 [AGREE[sem, Cat]] << * << DUMMY2 [AGREE
                      [sem, Cat]] << {等, 等等, 之類}
```

視詞的個別性略有變動，例如表範圍的連接詞形式爲:DUMMY[NP]，表示其形式必須是名詞成分，包括數量定詞及定量式複合詞。

2.連接詞爲修飾語(Cb)

通常在句中出現並連繫兩個句子成分的連接詞其處理的方式和句子中的附加成分的處理方式相同，不過附帶有連繫特徵，如 +reason, +hypothesis, +condition, +result, ...等。連繫特徵具有配對關係，如表一、表二中，偏正複句及聯合複句中所示以成對出現，不同的連接詞會賦予不同的連繫特徵，有時連接成分並不一定帶有連接詞，這時其配對關係是由其中一個連接成分中的連繫特徵或由常識推論得到；有時某些副詞也帶有連繫特徵，可以和帶有連接成分的從屬句形成複句，不過還是要符合連繫特徵間的配對原則。

(39) 連接詞爲句中附加成分的表示法

以「 所以 」爲例 :

語意訊息
- 語義 : so
- 特徵 : +result

語法訊息
- 分類 : Cbca
- 線性律 : # << Conjunction [Cbca] (註5)

前繫連接詞的移動性或非移動性可以表示在線性律上，移動性前繫連接詞「雖然」的表達結果如下:

(40) 語意訊息
- 語義 : although
- 特徵 : +concession

語法訊息
- 分類 : Cbaa
- 線性律 : Conjunction[Cbaa] < degree < *

六、複句的配對原則

語法的表達單元一般都只及於句子層次。由於複句結構通常由兩個句子所合成，其關聯成分可能帶有連繫兩個句子關係的連接詞或副詞，剖析時如何完成一個複句中兩個分句的結合關係，必需要用一種超越句子層次的處理方法。我們採用連繫特徵配對的方式處理。連繫特徵爲一種自動傳承特徵(foot features)[6, 12]連繫特徵在剖析時由連接詞或副詞傳遞至句子層次，完成第一階段的剖析。第二階段的剖析將連繫兩個相鄰且帶有連繫特徵的句子形成一個複句。

1.前繫特徵及後繫特徵皆存在時，其配對限制爲：

(41) (+concession, +contrast)
(+reason, +result)
(+hypothesis, +conclusion)
(+condition, +conversion)
(+rejection, +selection)
(+restriction, +addition)

2.僅後繫特徵出現時，和前句配對形成複句，其可能關係如(41)。
3.僅前繫特徵出現時，可能和前句或後句形成複句，由常識決定或句號'。'、分號';'決定。
4.連續特徵出現時： +alternative 成對形成複句。
+listing 可以無限制連續出現，形成複句。

七、剖析時的角色辨別及語法策略

綜合前面所述，我們將連接詞分爲不同類別，不同類別的連接詞有不同的詞彙訊息，在句子中可能有不同的角色考量--可能做中心語，也可能是動詞修飾語。以下我們將就一些簡單的實例說明如何利用語法表達模式來組合所要的成分，完成句子正確的分析。

(42) 政府根據國內學者和專家的意見制訂完整的辦法。

句中介詞「根據」所帶的成分應是名詞，「國內學者和專家的意見」可能的組合有幾種情形：

1.[國內學者　和　專家的意見]
2.國內　[學者　和　專家的意見]
3.[國內學者　和　專家]　的意見
4.國內　[學者　和　專家]　的意見

「和」屬於並列連接詞，如(38)所示要求兩個對等的成分，所以雖然可以是第 1 種和第 2 種情形，但是由於「學者」和「意見」分屬不同的名詞觀念範疇[11]，所以並非優先考慮的情形。第 3 中雖然正確連接「學者」和「專家」兩個同屬人的名詞，但是「學者」之前有修飾成分「國內」，而「專家」是光桿名詞(bare NP)，呈現不平衡的連接，所以不若最後一種情形的分析來得最爲妥當：結合「學者」和「專家」爲同心結構的名詞組，做「意見」的定語。

下面我們來看一些偏正複句的例子：

(43)　他因爲樂於助人，所以得到大家的讚賞。
(44)　如果人没有信用做不了事。
(45)　你必須來，否則會議開不成。

這些例句都是由兩個子句形成的複句，剖析時個別句子依據句中主要動詞所規範的語法規律完成句子的剖析。在(43)句中，「因爲」描述事件的理由，帶有連繫特徵 +reason，後面一個動詞組含有 +result特徵。依據連繫特徵配對原則，+reason 和 +result 可以配對形成一個偏正結構的複句。(44)爲前繫連接詞「如果」搭配一個不帶任何連接記號的動詞片語，形成一個(+hypothesis, +conclusion)的配對。(45)有一個後繫連接詞「否則」帶有 +conversion 特徵，由配對原則知道前句爲一條件(+condition)，形成一個條件複句。

有時候偏正複句的偏句會有疊用的現象，這時只要把兩個偏句均視爲與帶有配對特徵的主句相應即可，例：

(46)　雖然今天天氣不太好，雖然時間訂得不太對，但是參加的人卻十分踴躍。

至於聯合複句，只要能掌握複句是聯合結構的關係，並且確認前繫、後繫連接詞前後的對應關係即可。前面曾提及複句若無標記可能有不同的解釋，如：

(47) 他費心安排課程，鼓勵學生參加活動。

我們可以理解爲表目的的連動句，安排課程是爲了鼓勵學生參加，也可能解釋爲他又安排活動同時也鼓勵學生參加的並列句，如果有「不但」、「而且」的連接詞，則使得結構的關係清楚：

(48) 他不但費心安排課程，而且鼓勵學生多參加活動。

在(48)中，前後兩個連接詞直接作動詞修飾語即可，由配對原則可找到相應特徵的連接詞；「不但」可與帶 +addition特徵的後繫連接詞相應，即「而且、並且、且、反而」等連用，因此可確認二者所在分句組成聯合結構。

常常相對應的分句並不是緊鄰出現，中間插有其他句子成分，必須藉諸前後相應的特徵限制，才容易掌握，在偏正句及聯合句均有此現象。例：

(49) 他雖然年紀輕，經驗不豐富，但是處理事情井井有條。
(50) 她不但端莊秀麗，善體人意，並且會説會唱，多才多藝。

對於分句本身是含有連接成分的複句，也可依照上述的方式處理，例：

(51) 他雖然早年的家境不太好，但是由於他個人的努力，成爲一位偉大的音樂家。

句中「雖然早年的家境不太好」作主句的修飾語，因爲主句本身也是個偏正句，由「由於」帶出另一個修飾成分「由於他個人的努力」，整個複句陳述的中心成分是「他成爲一位偉大的音樂家」。

八、結論

本文針對連接詞的不同特性，提出一套適用的表達模式以及剖析策略，由於許多複句是由連接詞連接組合而成，我們根據複句不同的組合形式及連接詞相對應的語意特徵，掌握其配合的情形。

人類理解語言時，運用豐富的背景知識及複雜的認知方式，所以即使是一個複雜句、一個段落、一長篇文章，均不會有太多的困難。而我們進行自然語言的初步，僅能先就有標記、比較簡單的現象做處理，自然無法與人類的認知相比擬，也存在許多困難是電腦不容易解決的，如對應的連接成分中的連接詞完全省略，其連接的分句範圍不容易掌握。所以，未來仍有很長的路要走，僅以簡單的研究心得供作進一步研究的基礎。

註釋：

1.根據「現代漢語複句」[5] 所說，只要一個詞或詞組在交際中能夠完成一定的交際任務，就有資格成爲句子。例如：謝謝！　怎麼了？

2.假設關係表現「若p則q」的邏輯意義，條件關係表現出「唯若p則q」的意義，參見[9]。

3.本文主要是討論關聯組合的情況，沒有連接詞的複句類型不在討論之列，所以文中將聯合複句分爲選擇、遞進、並列三種，偏正複句分轉折、因果、假設、條件、取捨、目的(詳見第四章)。

4.大部分情況並列連接詞所連接的成分形式相同，但有時可連接不同形式的成分，例：

　　傳染方式爲空氣或直接接觸傳染。　　(NP或VP)

　　衛生所派員到家中或工廠爲民衆量血壓。　(GP或NP)

　故這裡要求的一致性並非絕對。

5.此線性律表示「所以」必須出現在句首。

參考書目:

[1] 呂叔湘, 漢語語法分析問題, 1979, 北京:商務印書館。

[2] 宋玉柱, 現代漢語十講, 1986, 南開大學出版社。

[3] 吳啓主, 李裕德, 現代漢語"構件"語法, 1986, 湖北:教育出版社。

[4] 張麗麗與中文詞知識庫小組, 國語的詞類分析(修訂版), 1989, 中央研究院計算中心。

[5] 劉振鋒, 現代漢語複句, 1985, 天津:人民出版社。

[6] 陳克健, 黃居仁, "訊息爲本的格位語法 -- 一個適用於表達中文的語法模式", 中華民國第二屆計算語言學研討會論文集(ROCLING Ⅱ), 頁95-119, 南港:中央研究院。

[7] 戴金木, 黃江海編著, 關聯詞語詞典, 1986, 四川:辭書出版社。

[8] 簡立峰, 陳克健, "詞彙訊息的層次表達與管理", 1990, 中華民國第三屆計算語言學論文集(ROCLING Ⅲ), 頁295 - 310, 新竹:清華大學。

[9] 魏文眞, 黃居仁, "中文的條件句", 1989,華文世界52期, 頁16- 23。

[10] Chao, Yuen Ren, 1968, A Grammar of Spoken Chinese, Berkley and Los Angeles : University of California Press.

[11] Chen, Keh-jiann, C. S. Cha, 1988, The Design of a Conceptual Structure and Its Relation of Chinese Sentences, Proceedings of 1988 International Conference on Computer Processing of Chinese and Oriental Languages (ICCPCOL),pp.428-431. Toronto, Canada.

[12] Gazdar, G., E. Klein, G. K. Pullum, and I. A. Sag 1985, Generalized Phrase Structrure Grammar. Cambridge: Blackwell, and Cambridge, Mass.: Harvard University Press.

[13] Li, Charles N. & Sandra A. Thompson, 黃宣範譯, 漢語語法, 1983, 台北:文鶴出版社。

* 本論文的研究得國科會中文語句剖析的語法模式研究計畫(NSC79-0408-E001-02)及中央研究院與工研院電通所合作「中文語句剖析系統第三期合作研究與開發計畫」(X2-79007)之部分經費贊助, 特此申謝。論文的寫作感謝詞庫小組其它成員的協助, 尤其是美玲熱心的幫忙完成繁瑣的打字工作, 更要謝謝黃居仁教授不厭其煩的閱讀本文並提供寶貴意見。

AMBIGUITY RESOLUTION OF SERIAL NOUN CONSTRUCTIONS IN CHINESE SENTENCES

Ching-Long Yeh
Department of Information Engineering

Tatung Institute of Technology
Taipei, Taiwan 10451
Phone: (02)5925252 EXT. 3484
E-mail: CLYEH@TWNTTIT.BITNET

Hsi-Jian Lee*
Department of Computer Science
and Information Engineering
National Chiao Tung University
Hsinchu, Taiwan 30050
Phone: (035)712121 EXT. 3735
E-mail: HJLEE@TWNCTU01.BITNET

ABSTRACT – We present a rule-based approach to resolve ambiguities of a series of noun constructions in Chinese sentences. According to our statistics, serial noun constructions occur 12.6% in our testing articles. The relationship between two adjacent nouns in a Chinese sentence can be a modification, possession, apposition, conjunction, or two separate noun phrases. We employ both syntactic and semantic features to resolve the possible ambiguities via rules, which take into account the situations that (1) the genitive marker, *de5,* in NP schema is omitted and (2) there is zero pause in coordinated constructions and appositions. The syntactic structure of a series of nouns with length exceeding two depends on the association of different types of combinations. We find that the conjunctions have the strongest association, then modification, possession and finally apposition. This scheme of ambiguity resolution is integrated into our unificationbased chart parser. Experimental results show its applicability.

I. INTRODUCTION

A substantive in Chinese is a word which normally functions as the subject or the object of the sentence[1, 2]. According to the conventions of syntactic categories in GPSG[3], substantives are denoted by the feature specifications,[N +] and [V –], which correspond to nominals in English. Hereafter, we use N to denote substantives in Chinese. To further distinguish words, each substantive is featured with a syntactic type. For example, the partial feature specification of *ban4gong1shi4* (office) is [[N +], [V –], [type place]], while the common noun *ren2* (person) is featured with [[N +],[V –], [type common]].

* To whom correspondence should be addressed.

Li and Thompson [4] formulated Chinese noun phrases (NPs) as follows:

associative phrase + {classifier/measure phrase, relative clause} + adjective + N,

where all the elements except the head noun *N* are optional. An associative phrase (AP) is an NP followed by a genitive marker, *de5*, such as *wo3 de5* (my). A classifier/measure phrase, termed DM hereafter, is composed of a demonstrative followed by a measure, such as *zhe4 ben3 (shu1)*(this (book)) and *yi1 bei1 (ka1fei1)*(a cup (of coffee)). A relative clause (RC) is simply a nominalized clause placed before a head noun, such as *zhang1san1 mai3 de5 (qi4che1)* ((the car) that Zhangsan bought). In this situation, *de5* can not be omitted. However, under certain situations, the genitive marker can be omitted, such as *wo3 (de5) mei4mei5*(my sister) in (1). A noun can also be used as a modifier of another noun, such as *you4ji4yuan2 lao3shi1* (a teacher in a kindergarten) in (1), where *lao3shi1* (teacher) is the head noun modified by *you4ji4yuan2* (kindergarten). Both *wo3 mei4mei5* and *you4ji4yuan2 lao3shi1* are NPs composed of two adjacent nouns without any marker between them.

(1) *wo3 mei4mei5 shi4 yi1 wei4 you4zhi4yuan2 lao3shi1.*

(My sister is a teacher in a kindergarten.)

Hereafter, adjacent nouns or pronouns are called NNs or serial noun constructions (SNC).

Though the above two NNs are NPs, the first one is a possessive phrase and the second is a modifer–head structure. In addition to the above two cases, there are other types of NNs. The NN in (2), *peng2you3 tong2xue2* (friends and classmates) is a conjunction; *mi4shu1 li3xiao3zhu* (the secretary, Li Xiaozhu) in (3) is an apposition. Topicalization also results in a kind of NN, such as *shu1 wo3* (book I) in (4). The NN *wo3 zhong1guo2ren2* (I (am) a Chinese) in (5) is a subject–predicate construction with copula, *shi4*, omitted. A special kind of NP, such as *ka1fei1 wu3bei1* (five cup of coffee) in (6), is of the structure *N + DM*. The subject in double subject sentences is another kind of NNs, such as *wo3 yi1fu2* (my clothes, or I, clothes) in (7).

(2) *ta1 ji4 he4ka3 gei3 peng2you3 tong2xue2.*

(He sent congratulatory cards to his friends and classmates.)

(3) *mi4shu1 li3xiao3zhu1 chu1qu4 le5.*

(The secretary, Li Xiaozhu, went out.)

(4) *zhe4 ben3 shu1 wo3 xi3huan1.*

(This book I like.)

(5) *wo3 (shi4) zhong1guo2ren2.*

(I am a Chinese.)

(6) *ta1men5 jiao4 ka1fei1 wu3bei3.*

(They order five cups of coffee.)

(7) *wo3 yi1fu2 xi3 de5 hen3 gan1jing4.*

(My clothes is washed cleanly, or I wash my clothes cleanly.)

There are no such problems in English[5], since possessive, conjunctive, punctuation, and structural evidence can help distinguish the mutual relations between different nouns. The NNs in (5)–(7) can be processed as appositions, ordinary NPs, and possession, respectively, which will be clear later.

From our experiments, serial noun constructions occur very frequently in Chinese sentences[6]. For 1545 sentences, 9.25 characters per sentence in average, there are 327 NNs appearing in 195 sentences. Thus the frequency of serial noun constructions in the testing samples is 12.6% (195/1545).

The researchers on Chinese syntactic analysis and semantic interpretation paid less attention on the problem of NN combinations [7, 8], or even did not touch the problems [9–13]. In our Chinese-to-English machine translation system (CEMAT) [10], we propose a rule-based approach for processing NN combinations, which can be integrated into the existing parser [9,10] and semantic interpreter [8].

In general, it is very difficult to distinguish the types of NN combinations through syntactic analysis since they are of the same structure: a noun followed by another noun. In our approach, we employ various syntactic and semantic feature to determine the combinations of NNs. The association of nouns in serial noun construction with length exceeding two are not trivially from left to right. For example, in the long NN, *wo3 tong3xue2 zhang1san3* (my classmate, ZhangSan), the association is ((*wo3 tong3xue2*) *zhang1san1*). The association of *wo3 ba4ba5 ma1ma5* (my father and mother) is (*wo3* (*ba4ba5 ma1ma5*)). The different kinds of association rely on the different combination types. In this paper we establish a rule base to determine the hierarchical structure of long NNs.

In Section II, we will show the combination rules for a pair of adjacent nouns. Then in Section III, we will discuss the association of nouns for serial noun constructions with length exceeding two. In Section IV, the implementation is briefly introduced. Concluding remarks are made finally.

II. NN COMBINATION RULES

Let N_1 and N_2 be two adjacent nouns in a sentence. The general form of NN combination rules is

LHS: S_1, S_2

RHS: Combinationtype

where S_1 and S_2 denote syntactic and semantic information encoded in the form of frame-type feature structures. For example, if N_1 is a personal pronoun and N_2 is a noun in the domain hierarchy *role*, then N_1N_2 is a possessive type combination. Encoded in the feature structure form, the rule becomes as follows.

```
LHS: [phon α,
      syn  [head [n +,
                  v ,
                  type  pronoun]],
      sem  [var  [hier  person]]].
     [phon β,
      sem  [var  [hier  role]]].
RHS: [phon αβ,
      syn  [nn_type  possession].
```

This rule states that if two input nouns N_1 and N_2 can unify successfully with the two components in the LHS, respectively, then an additional syntactic feature, *nn_type*, is augmented in the resulting feature structure of N_1N_2. For example, *wo3* (I) and *mei4mei5* (sister) in sentence, *wo3 mei4mei5 shi4 ge5 xiao3xue2 lao3shi1* (My sister is a teacher of a primary school), unify successfully with the two components of the LHS of the above rule; therefore, it is referred as a possessive combination.

In the following, we will discuss the combination rules of possession, conjunction, apposition, separate constituents, and modification, respectively. The rules together with the corresponding examples will be shown in the tabular form. For convenience, we only show the terminal values of the components of the LHS.

A. Possession type

The possession type of NNs happens between two human relatives with *de5* omitted. The first noun is a personal pronoun, and the second noun is either in the domain hierarchy *role*, *component*, or *a_corporate_person*, which means the social individual sentient, components of human body, and the social collective sentient, respectively [14]. The possessive personal pronoun in English corresponds to a personal pronoun followed by an optional genitive marker *de5* in Chinese.

An NP of the structure $DM+N$ commonly represents a definite object, such as *na4 ben3 shu1* (that book) and *zhe4 duei4 jia1ju4* (this pair of furniture). Such kind of NP preceeded by a personal pronoun or a personal proper noun strongly imply that the succeeding NP is owned by the preceeding noun. We summarize the rules for possessive NNs in the following table.

Rule #	S_1	S_2	Examples
1	personal_ pronoun	role	*wo3 ba4ba5* (my father), *ni3 lao3shi* (your teacher)
2	personal_ pronoun	component	*wo3 wei4* (*tong4*) (my stomach (pain)), *ta1 to2* (*tong4*) (his head (pain))
3	personal_ pronoun	a_corporate_person	*wo3 xue2xiao4* (my school), *wo3men5 gong1si1* (our company)
4	personal_ proper_noun	DM + N	*zhang1san1 na4ben3 shu1* (that book of Zhangsan's)
5	personal_ pronoun	DM + N	*wo3 zhe4dui1 jia1ju4* (these furniture of mine)

Ambiguity occurs in a DM followed by two nouns because they can be the structure $(DM + N_1) + N_2$ or $DM + (N_1 + N_2)$. The former structure is a possessive relation, N_2 belonging to $(DM + N_1)$, such as *zhe4jia1 can1ting1 (de5) cai4* (the food of this resturant) in (8). The latter one is that a DM modifies a modification type of NN, such as *zhe4ge5 you4zhi4yuan2 lao3shi1* (this teacher in a kindergarten) in (9). The cooccurrence relation between a DM and its following noun provides an effective clue to resolve this ambiguity [15].

(8) *zhe4jia1 can1ting1 cai4 hen3 hao3 chi1.*

(The dish of the resturant tastes good.)

(9) *zhe4ge5 you4zhi4yuan2 lao3shi1 jiao1 de5 hen3 hao3.*

(This teacher in a kindergarten teaches students well.)

The subject of a double-subject sentence can also be interpreted as a possessive relation, such as the first meaning of *wo3 yi1fu2* (my clothes) in (10). However, (10) can be interpreted alternatively as *I washed clothes cleanly*. The resolution of the meaning ambiguity depends on the context of the sentence.

(10) *wo3 yi1fu2 xi3 de5 hen3 gan1jing4.*

(My cloth is washed cleanly, orI wash my cloth cleanly.)

B. Conjunction type

A conjunctive NN is a coordinative construction with zero marker between two nouns, where each noun of the NN has approximately the same function as the whole construc-

tion[1]. Fragments *ba4 ma1* (father and mother) and *zhuo1zi5 yi3zi5* (tables and chairs) are instances of conjunctive NNs. In English, it is illegal that there is no conjunctive in a conjunction. In the following table we use variables, as those used in Prolog, to catch this notion.

Rule #	S_1	S_2	Examples
1	X	X	*ba4 ma1* (father and mother) *zhuo1zi5 yi3zi5* (tables and chairs)

C. Apposition type

When two expressions in succession refer to the same thing, the relation is one of apposition. They are further classified into close apposition, loose apposition, and interpolated apposition[1]. The examples of close apposition are *wang2 jia1* (Wang family, the Wangs), *li3 dai4fu1* (Doctor Li), and *ke1xue2 zhă2zhi4* (The magazine Science). As a rule, close appositions are subordinate phrases or compounds such that the first part modifies the second. In this paper, they are classified as modification type of NNs. In loose apposition, the expressions are in coordination without pause, as in *wo3 peng3you3 zhang1san1* (my friend, Zhangsan) and *zhong3tong3 li3deng1hui1* (the President, Li Denghui). An interpolated apposition is an inserted phrase with a pause marker ",". The omission of the interpolated apposition does not affect the completeness of sentence, such as *zhang1san1, wo3 de5 yi1 ge5 peng2you3, ming2 tian1 yao4 lai2* (Zhangsan, one of my friend, will be here tomorrow). Since the expressions are identified by commas, they are not considered as NNs here.

Human–related appositions occur frequently because we generallly need to point out the very person we talk about. Thus we obtain a rule that a *role* noun followed by a personal proper noun or personal pronoun is an appositive NN. For the cases of nonhuman related appositions, the first noun is a proper noun and the second one is used to describe the property of the first one.

The Chinese reflexive morpheme, *zi4ji3* (self), may optionally be preceded by a pronoun that is coreferential with the subject of the sentence [4], as in (11). A personal pronoun followed by a DM is also a kind of appositive NN, where the DM mentions the members of the personal pronoun. For example, *ta1men4 san1wei4* (they, three) in (12) is an appositive NN, where the DM *san1wei4* (three) indicates that there are three members in their group.

(11) *zhi3 you3 di4di5 ta1 yao4 shang4xue2.*

(Only my brother, he, needs to go to school.)

(12) *wo3 zhi4ji3 yao4 qu4 mei3guo2.*

(I myself want to go to the U.S.A.)

A noun followed by a DM commonly serves as the object of a verb, which is equal to an ordinary NP, *DM+N.* For example, *zhan4dou4ji1 wu3bai3 jia4* (fighter plane, five hundred) is equal to *wu3bai3jia4 zhan4dou4ji1* (five hundred fighter planes). When parsing, it is transformed to be the form *DM+N* in order to process them as ordinary NPs.

(13) *ta1men4 san1wei4 dian3 le5 wu5bei1 ka1fei1.*

(They three ordered five cups of coffee.)

In summary, we have the following rules of apposition.

	S_1	S_2	Examples
1	hier=role	personal_ proper_noun	*wo3 peng2you3 zhang1san1* (my friend, Zhangsan)
2	hier=role	personal_ pronoun	*di4di5 ta1* (my brother, he)
3	hier=X, proper_name	hier = X	*san1guo2yan3yi4 zhe4 ben3 shu3* (the book "The Romance of the Three Kingdoms")
4	personal_ pronoun	reflexive	*wo3 zi4ji3* (I myself)
5	personal_ pronoun	DM	*ta1men4 san1 wei4* (they three)

The subject–predicate constructions NP_1 NP_2 and NP_1 NP_2 NP_3 are analyzed as the omission of the copuls, *shi4,* to take advantage of the general sentence pattern: NP + VP. For example, see sentences (14) and (15).

(14) *ta1 (shi4) zhong1guo2ren2.*

(He is a Chinese.)

(15) *ka1fei1 yi1bei1 (shi4) wu3shi2yuan2.*

(The price of a cup of coffee is fifty dollars.)

The above consecutive NPs are rather similar as appositive NNs, except that appositive NNs are subjects or objects of verbs.

D. Separate constituent type

Two neighboring nouns may play two different syntactic roles. In order to determine this type of NNs, we first consider which kinds of nouns can not be modified by other nouns. From [1, 4], it is obvious that proper nouns and pronouns can not be modified by other nouns. When such a situation occurs, the two nouns are taken to be two phrases. This class of NNs is mainly from topicalization, such as *shu1 wo3* in (16) and *bao4zhi3 ni3* in (17), and so on.

(16) *zhe1 ben3 shu1 wo3 xi3huan1*

(This book I like it.)

(17) *jin1tian1 de5 bao4zhi3 ni3 kan4guo4 le5 ma5?*

(Have you read today's paper yet?)

In general, topics must occur in sentence–initial position[4]. Time phrases and locative phrases occuring in sentence–initial positions are considered as topics as well [4], as shown below.

(18) *zuo2tian1 yu3 xia4 de5 hen3 da4.*

(It rained heavily yesterday.)

(19) *tai2bei3 yu3 xia4 de5 hen3 da4.*

(It rained heavily in Taipei.)

A time word may be preceded by another noun. If this noun is a time word, then they form a modification type of NN. For example, both *zho2tian1* (yesterday) and *xia4wu3* (afternoon) in (20) are time word; they form a modification type of NN. However, in *wo3 ming2tian1* of (21), *ming2tian1* (tomorrow) is a time word but *wo3* (I) is not; thus they form an NN of separate constituents.

(20) *wo3 zho2tian1 xia4wu3 qu4 tu2shu1guan3.*

(I went to the library yesterday afternoon.)

(21) *wo3 ming2tian1 qu4 tai2bei3.*

(I will go to Taipei tomorrow.)

Rule #	S_1	S_2	Examples
1	X	pronoun	*shu1 wo3(book, I)*
2	X	proper_noun	*bao4zhi3 ni3* (paper, you)
3	time	X	*zuo2tian1 yu3* (yesterday, rain)
4	place	X	*tai2bei3 yu3* (Taipei, rain)
5	time	X {time}	*wo3 ming2tian1* (I, tomorrow)

The NNs identified by Rule 1 and 2 of the above table are mostly resulted from movement of object to the sentence–initial position. In our parser, when such NNs are identified, two NPs are established for each noun. The verb is then established as a VP with a missing object, i.e., *VP/NP* in GPSG.

The direct and indirect object of a ditransitive verb is also an NN of this type. However, we do not include the case here because the verb needs two NPs to make the VP saturated.

E. Modification type

In a modification type NN, the second noun is the head noun and the first one is an adjective. For example, in *xiao3xue2 lao2shi1* (a teacher of a primary school), *xiao3xue2* (primary school) modifies the head noun *lao2shi1*. It is quite difficult to obtain rules for determining the modification type of NNs because there are less grammatical evidence [16]. We use a catch–all rule to solve this problem. That is, if an NN can not trigger any of possession, conjunction, apposition, and modification types of rules, it is taken as a modification type of NNs.

F. Conflict resolution

The conditions of previous rules are not mutually independent. We adopt the specificity ordering strategy to resolve the conflicts when more than one rule are triggered. This strategy states that the rule with the most specific conditions is fired first. When the parser fails to produce a result by the fired rule, the rule with the less specific conditions is fired. This procedure proceeds until the parser completes the analysis. For example, in sentence (2), since the NN *peng2you3 zhang1san1* is a *role* noun followed by a personal pronoun, it can trigger both Rule *1* of appositive type and *Rule* 1 of separate constituent type. According to the specificity, the former one is fired and a correct result is obtained.

III. ASSOCIATION OF NOUNS IN SERIAL NOUN CONSTRUCTIONS

The association of nouns is nontrivial for a serial noun construction with length exceeding two. For example, let $N_1N_2N_3$ be three successive nouns in a sentence. There are two possible structures:

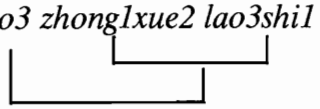
$$(N_1N_2)N_3 \text{ and } N1(N_2N_3)$$

As mentioned in Section I, the phrase *wo3 peng2you3 zhang1san1* should be of the former structure, while the phrase *wo3 ba4ba5 ma1ma1* the structure of the latter one.

Thus to design a parser, we must consider the association of nouns in serial noun constructions. The above examples illustrate the association of N_2 with N_1 and N_3, respectively. It can be reformulated as the problem of comparing the precedence between the combination types of N_1N_2 and N_2N_3. For the phrase *wo3 peng2you3 zhang1san1, wo3 peng2you3* and

peng2you3 zhang1san1 are possession and apposition, respectively. Since each of the nouns in an appositive NN refers to the same thing, omission one of the phrases does not change the meaning of the phrase. Thus, an apposition has the least precedence. In other words, the possessive relation *P* precedes the appositive relation *A*, represented as $P > A$. Accordingly, *wo3* first associates with *peng2you3* and then *wo3 peng2you3* with *zhang1san1*.

For a possessive NN, the preceding noun is an associative phrase with omitted *de5* such as *wo3*(I) in the fragment *wo3 (de5) ge1ge5* (my brother). From observations, a NN combination of the modification type *M* has higher precedence than that of the possession type *P*, denoted as $M > P$. It results in the following structure:



A conjunctive NN is a unit acting, as a whole, like the subject or the object of a verb; they are covered under the scope of an adjective or a modifying noun. In sentence (22), the noun *xue2xiao4* (school) modifies the succeeding nouns *lao3shi1* (teachers) and *xue2sheng1* (students).

(22) *xue2xiao4 lao3shi1 xue2sheng1 dou1 can1jia1 lu3xing2*

(All of the teachers and students in the school participated in the tour)

Thus we obtain that the conjunctive combination *C* precedes the possessive combination *P*, denoted as $C > P$.

When a possessive combination and a modification combination appear together with an appositive combination, the possession *P* and modification *M* precedes apposition *A*. That is,

$$M > P > A,$$

For example, in sentence (23), *wo3 zhong1xue2 lao3shi1* and *wang2xiao3zhu1* refer to the same person. After these two NPs are combined together, they form an appositive NP.

(23) *wo3 zhong1xue2 lao3shi1 wang2xiao3zhu1 jie2hun1 le5*

(My high school teacher, Ms. Wang XiaoZhu, got married)

From the above discussion, we conclude

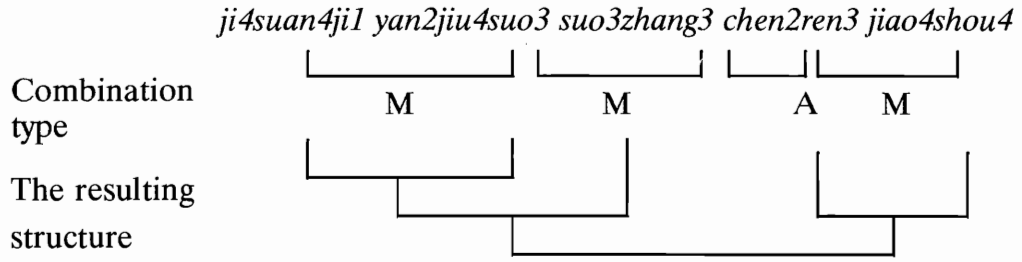
$$C > M > P > A.$$

The above discussions consider the precedence relations for serial noun constructions containing different types of combination. For a sequence of the same type of combinations, the precedence is from left to right. For sentence (24),

(24) *ji4suan4ji1 yan2jiu4suo3 suo3zhang3 chen2zen3 jiao4shou4 chu1guo2 le5*

(The head of the Institute of Computer, Prof. Chen Zen, went abraod.)

the combination types and the resulting structure of the serial noun construction is shown below.



IV. IMPLEMENTATIONS

We present in this section how to employ NN combination rules and NN association in our parser to disambiguate serial noun constructions. HPSG (Head-driven Phrase Structure Grammar) [17], a lexicon-driven unification-based grammar formalism, constructs constituent structures of sentences in accordance with the Head Feature Principle (HFP), Subcategorization Principle (SP), and Adjunct Principle (AP). The HFP declares that a phrase shares the same features with its head daughter. The SP states that in any phrase, each complement daughter must be unifiable with a member of the head daughter's subcat-list, a list of subcategorization specifications, that remains to be satisfied. The AP states that any adjunct daughters must be unifiable with some member of the head daughter's adjunct specifications.

Our original parser employed the above principles to parse Chinese declarative sentences [9]. However, in our further studies, we find that the principles are not enough to replace all the ID (immediate dominant) rules in GPSG[3], especially in handling the nominalizations of Chinese sentences. The new version of the parser thus adds a rule-based mechanism for handling nominalizations.

The association of nouns in serial noun constructions can not be reflected from the adjunct feature. We use rules to determine the combination of nouns. The resulting feature structures of NN combination are head-adjunct structures, except the conjunctive combination, which is represented as a coordinate structure. Considering sentence *ta1 mei4mei5 zhang1xiao3zhu1 shi4 ge5 xiao3xu2 lao3shi* (His sister, Ms. Zhang XiaoZhu, is a teacher in a primary school), the resulting feature structure of *ta1 mei4mei5 zhang1xiao3zhu1* is shown partially as follows.

```
[phon  [ta1, mei4mei5, zhang1xiao3zhu1]],
head_dtr                          /* head-daughter */
```

```
        [phon [zhang1xiao3zhu1],
adjuncts_dtr                        /* adjunct-daughter */
        [phon [ta1, mei4mei4],
        head_dtr [phon [[mei4,mei4]],
        adjuncts_dtr
                [phon [[ta1]],
                nn_type possession],
nn_type apposition].
```

where the additional feature *nn_type* indicates the type of NN combination.

When two consecutive nouns are detected during the course of parsing, the parser looks forwards the following constituents to determine the maximal coverage of the serial noun construction. Then the parser consults the NN combination rules to determine the type of the combination and to form the resulting feature structure. NN association rules are consulted to determine the structure if there are more than two nouns in the serial noun construction.

V. CONCLUDING REMARKS

We have presented a rule-based approach to resolve ambiguities of serial noun constructions. The combination types of neighboring nouns have been examined and determined by a set of rules. They can be applied in a syntactic parser to determine the correct role of each noun. The association of NN combinations have also been analyzed to construct the correct structures.

In our experiments, 23 out of 327 cases, 7.1% approximately, are misidentified by consulting the NN combination rules and the association rules. The types of misidentification are summarized as follows.

1. The genitive marker, *de5*, are occasionally omitted. There are 7 cases of this type of misidentification, which are identified as modification type of NNs.
2. The conjunctive, *he2*, or the conjunctive punctuation mark are omitted. There are two cases of this type of misidentification.
3. One case of the conjunction structure is misidentified as $(N_1\ N_2)\ he2\ N_3$, while it should be $N_1\ (N_2\ he2\ N_3)$.
4. The remaining misidentifications are related to the apposition type of NN due to the incomplete of the apposition rules.

To solve the first two types of misidentification, a commonsense knowledge base is required. Actually, it is very difficult to built such a knowledge base. Acquiring new rules from

the generalization of misidentified patterns can reduce these types of error. This is left for further researches. The last two types of misidentification needs more studies in conjunctive and appositive constructions.

The goal attempting to resolve syntactic ambiguities of serial noun constructions has partially been reached. Our next step in the processing of noun phrases is to work toward lexical selection in English and to order the selected constructions in an appropriate order.

REFERENCES

1. Y. R. Chao, *A Grammar of Spoken Chinese*, CA:University of California Press, 1968.
2. Group of Chinese Word Knowledge Base, *The Analysis of Syntactic Categories for Mandarin*, Revised Edition, Technical Report #T0002, Academic Scinica, Taipei, Taiwan, R.O.C., 1989. (In Chinese)
3. P. Sells, *Lectures on Contemporary Syntactic Theories*, CSLI Lecture Notes, No.13, 1985.
4. C. N. Li and S. A. Thompson, *Mandarin Chinese: a Functional Reference Grammar*, Ber kerley, CA:University of California Press, 1981.
5. J. Allen, *Natural Language Understanding*, Menlo Park, CA:Benjamin/Cumminngs, 1987
6. C. L. Yeh and H. J. Lee, "Rule based word identification for Mandarin Chinese sentences: a unification approach," *Proc. of CPCOL*, Changsha, Hunan, China, pp. 3337, 1990.
7. Y. Yang, *Studies on an Analysis System for Chinese Sentences*, Ph.D. Thesis, Kyoto University, japan, 1985.
8. H. J. Lee and C. H. Lee, "Computer interpretation of Chinese declarative sentences based on situation sematics," *Journal of Information Science and Engineering*, Vol. 5, No. 4, Oct. 1989, pp. 379–394.
9. H. J. Lee and P. R. Hsu, "Parsing Chinese sentences in head driven phrase structure grammar," to appear in *Journal of Computer Processing for Chinese and Oriental Languages*, 1990.
10. Y. S. Chang and H. J. Lee, "Parsing Chinese nominalization based on HPSG," *Proc. of ROCLING III*, Taipei, Taiwan, R.O.C., pp. 311–338, 1990.
11. L. J. Lin, J. J. Huang, K. J. Chen, and L. S. Lee, "A Chinese national language processing system based upon the empty categories," *Proc. of AAAI'86*, pp.1059–1062, 1986.
12. C. G. Chen, K. J. Chen and L. S. Lee, "A model for lexical analysis and parsing of Chinese sentences, *Proc. of Intern. Conference on Chinese Computing*, Singpore, pp. 3340, 1986.

13. C. N. Huang and M. S. Sun, "A deterministic algorithm with multiple scanning for Chinese sentences, *Proc. of CPCOL,* Changsha, Hunan, China, pp. 33–37, 1990.

14. D. Dahlgren, "Using commonsense knowledge to disambiguate word senses," in *Natural Language Understanding and Logic Programming II*, V. Dahl and P. SaintDizier (eds.), Elsevier Science Pub., pp. 255–276, 1981.

15. L. L. Chang, et al., "Classification and Cooccurrence restriction in Chinese simple noun phrases," *Proc. of CPCOL,* Chicago, pp, 107–110, 1987.

16. T. Finin, "The semantic interpretation of nominal compounds," *Proc. of AAAI,* pp. 310312, 1980.

17. C. Pollard and I. A. Sag. "Information–based Syntax and Semantics: Volume I. Fundamentals," CSLI Lecture Notes, No. 13, 1987.

Determinative-Measure Compounds in Mandarin Chinese

Formation Rules and Parser Implementation

*Ruo-ping Jean Mo**, *Yao-Jung Yang**, *Keh-Jiann Chen**, *Chu-Ren Huang***
**The Institute of Information Science, Academia Sinica*
***The Institute of History and Philology, Academia Sinica*
Nankang, Taipei, Taiwan,
Republic of China

Abstract

We deal with the identification of the determinative-measure compounds (DMs) in parsing Mandarin Chinese in this paper. The number of possible DMs is infinite, and cannot be listed exhaustively in a lexicon. However, the set of DMs can be described by regular expressions, and can be recognized by a finite automaton. We propose to identify DMs by regular expression before parsing.

After investigating large linguistic data, we find that DMs are formed compositionally and hierarchically from the simpler constituents. Based upon this fact, some grammar rules are constructed to combine determinatives and measures. Moreover, a parser is also formed to implement these rules. By doing so, almost all of the unlisted DMs are recognized. However, if only the DM recognition procedure is fired, many ambiguous results appear, too. Yet with our word segmentation process, these amiguities are greatly reduced.

I. Introduction

A determinative-measure compound (DM) in Mandarin Chinese is composed of one or more determinatives, together with an optional measure.(1) It is used to determine the reference or the quantity of the noun phrase that co-occurs with it. It may sometimes function as a noun phrase by itself.(2) However, despite the fact that the categories of determinatives and measures are both closed, the combinations of them are not.

(1) 這 三 本
D D M
this three CL
"these three books"

(2) 他 喜 歡 這 三 個
he like this three CL
"He likes these three."

(3) 三 百 二 十 一
three hundred two ten one
"three hundred and twenty one"

五 萬 四 千 三 百 二 十 一
five ten-thousand four thousand three hundred two ten one
"fifty four thousand three hundred and twenty one"

九 億 零 五 萬 四 千 三 百 二 十 一
nine hundred-million zero five ten-thousand four thousand three hundred two ten one
"nine hundred million fifty four thousand three hundred and twenty one"

(4) 九 點 三十 分
nine o'clock thirty minute
"half past nine"

三月 八 日 星期五 上午 九 點 三十 分
March eight day Friday morning nine o'clock thirty minute
"nine-thirty a.m., Friday, March eighth"

民國八十年三月 八 日 星期五 上午 九 點 三十 分
1991 March eight day Friday morning nine o'clock thirty minute
"nine-thirty a.m., Friday, March eighth, 1991"

(5) 二 十 五 件
two ten five CL
"twenty five items"

這 二 十 五 件
this two ten five CL
"these twenty five items"

其餘 二 十 五 件
other two ten five CL
"the other twenty five items"

Take example (3) as an illustration: it is obvious that the total numbers of possible combinations of numerals are enumerable but infinite. Because of the productivity of these DMs, listing them directly in the lexicon becomes almost impossible. Consequently, the process of finding proper word breaks for Chinese sentences is incomplete without DMs in the lexicon. Therefore our design of a word segmentation system utilizes both the words listed in the lexicon and those generated by DM rules. We have the following reasons to support our strategy. First, from the processing point of view, it is better to recognize compound words as early as possible and DMs can be considered as compounds. Since the structure of DMs seems to be exocentric, they are not similar to other endocentric phrase structures and can not be analyzed by head driven parsing strategies. Second, the set of DM forms is a regular language which can be expressed by regular expressions and recognized by finite automata. It is well known that the grammar of Mandarin contains central embedding and must be expressed by context-free grammars. [7] This also suggests that the processing of DMs should be separated from the processing of other phrases. Third, the set of determinatives and measures usually serve only a single grammatical function which are comparatively simpler than other categories which play multiple grammatical functions due to the lack of inflections in Chinese. We believe that DMs can be identified at the level of lexical analysis and this fact has been proven by our experiments. We design a regular grammar interpreter with a chart parser to identify the DMs for input sentences. The flexible design of this interpreter allow us to modify the grammar rules generating DMs without changing the interpreter.

In the next section, the structures of DMs and their representations are given. The third section states the design of the interpreter and its application to improve the DM rules. The fourth section shows the experimental results and discussions. The last section concludes with remarks on other applications of the DM identification system.

II. The Structures and Representations of DMs

Earlier studies of DMs concentrate mainly on 1. listing members of determinative and measure sets, 2. proposing classifications, and 3. describing agreements between measures and their nominal heads. Chao[8], for instance, divides determinatives into four subclasses:

(6) (i) demonstrative determinatives: 這, 那, 哪
(ii) specifying determinatives: 每, 各, 別, 旁, 本, 某, 上, 下, 前, 後, 今, 昨, 明, 去
(iii) numeral determinatives: 二, 百分之三, 四百五 十 etc.
(iv) quantitative determinatives: 一, 滿, 全, 整, 半, 幾, 多, 多少,許多, 好些, 好多, 好幾, 很多

Measures, on the other hand, are divided into nine classes by Chao[8] 1. classifiers, e.g. 本 "a (book)", 2. classifiers associated with V-O constructions, e.g. 手 "hand", 3. group measures, e.g. 對 "pair", 4. partitive measures, e.g. 些 "some", 5. container measures, e.g. 盒 "box", 6. temporary measures, e.g. 身 "body", 7. standard measures, e.g. 公 尺 "meter", 8. quasi-measures, e.g. 國 "country", and 9. measures with verbs, e.g. 次 "number of times". However, earlier studies do not analyze the internal structure of DMs, which is crucial to their recognition and formation.

In what follows, we will first adjust the various determinative sets based on their productivity and co-occurrence restrictions, and then discuss the internal structure of DMs,

as well as the rules to construct them. As for the measures, although they also play a role in forming DMs, the choice of them largely depends on the nature of the entity denoted by the nominal heads. Since this paper focuses on the DM itself, the problem of agreement between measures and nominal heads will not be pursued here.

2.1. The Determinative Sets

In general, determinatives are classified in terms of their meanings. However, if we take typical grammatical properties such as productivity and co-occurrence restrictions into account, we find that some of the classifications based upon meanings are questionable.

Instead we propose three criteria to classify various determinatives. They are 1. productivity, 2. syntactic similarities, and 3. semantic meanings. The determinatives are quite different in terms of productivity. For instance, 今 "today", 明 "tomorrow", 去 "last", 昨 "yesterday" precede no other determinatives. In fact, they can only co-occur with a few measures such as 日 "day", 天 "day", and 年 "year". Since their usage is fixed, we will put all the possible combinations of those determinatives and the measures, such as 今天，明天，今年，明年，今日，明日，昨天，昨日，去年， in the lexicon. On the other hand, the determinatives with high productivity will be classified according to their syntactic and semantic similarity.

Although Mandarin Chinese allows two or more determinatives to be juxtaposed, not every determinative can co-occur with the others. 别 "other", and 旁 "side", for example, are incompatible with other determinatives. But 這 "this" is relatively free: it can be adjoined to either a numeral or a quantitative determinative :

(7) *别 三 家
other three home

*旁 半 天
side half day

這 三 名
this three CL
"these three persons"

這 半 年
this half year
"this half year"

Therefore, co-occurrence relations will be the major syntactic criteria employed to subclassify determinatives.

The primary function of a determinative is to restrict or quantify the references of the following noun phrases. From the data collected, a variety of other words also have much the same function and distribution as those well-discussed determinatives. 近 "near" and 將 近 "near" are two such words. Like 這 "this" in (8), 近 "near" in (9) also modifies the following noun phrase and determines which period the event expressed by the verb phrase occurs[1] Actually, these two words can be substituted with each other in this context. Based upon this principle, those with similar function and distribution as determinatives will also be included in the determinative set.

---

[1] Another reason for treating 近 "near" and 將 近 "near" as determinatives comes from the grammatical theory we adopt. According to one assumption of the Lexical Mapping Theory, every verb must have a subject.[3][7] However, this condition will not be held if we analyze 近 "near" and 將 近 "near" as verbs whenever they appear:

(i) 我 家 離 台 北 很 近
I home from Taipei very near
"My home is near Taipei."

端 午 節 將 近
Dragon-Boat-Festival near
"It's almost Dragon Boat Festival."

(ii) 近 十 時 三 十 分 我 回 到 家
near ten o'clock thirty minute I back home
"I got home at about ten-thirty"

In sentence (ii), 近 "near" cannot take 張 三 "Jangsan" as its subject. In fact, no subject may occur before it. In order not to violate the more accepted condition, we will classify 近 "near" and 將 近 "near" as determinatives besides verbs.

(8) 這 二 十 年 來 他 每 天 晨 泳
this twenty year come he every day morning swim
"He's gone for swim every morning for the past twenty years."

(9) 近 二 十 年 來 他 每 天 晨 泳
near twenty year come he every day morning swim
"He's gone for swim every morning for about twenty years."

Due to space limitations, we will simply list our revised determinative sets in Appendix I and related DM rules in Appendix II without further discussion.

2.2. The Internal Structures and Formation Rules of DMs

As was mentioned at the very beginning of this paper, a DM can contain one or more determinatives together with an optional measure. Closer investigation shows that the composition of the determinative can be complicated: it may consist of only one kind of iterating determinative, like numerals, or have several determinatives belonging to different subsets. For example, in 其他數百名 "hundreds of the other," three different kinds of determinatives are concatenated. In addition, these adjoining determinatives are not freely ordered. They have to conform with some linear precedence restrictions.[2]

(10) a. 這 二 十 人
this twenty person
"these twenty persons"

a' *二 十 這 人
twenty this person

b. 其 餘 近 一 百 五 十 名 團 員
other near one hundred fifty CL member
"the other almost one hundred and fifty members"

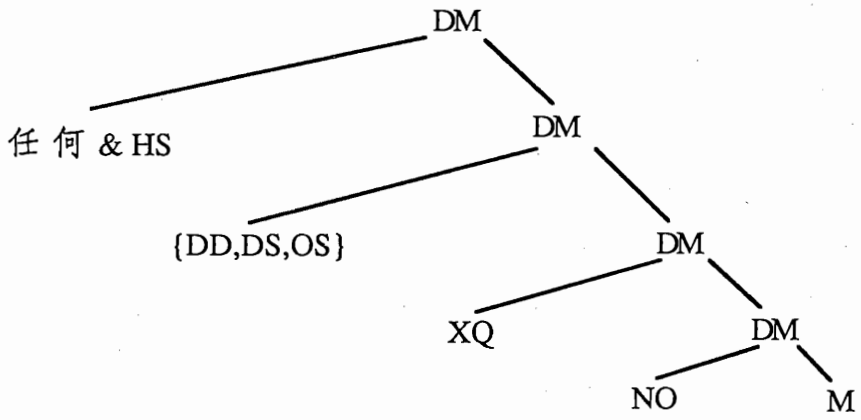
b' *一 百 五 十 其 餘 近 名 團 員
one hundred fifty other near CL member

[2]Similar restrictions also appear among numeral compounds. However, such restrictions depend on mathematic knowledge, not linguistic knowledge. In this paper, we do not handle these restrictions.

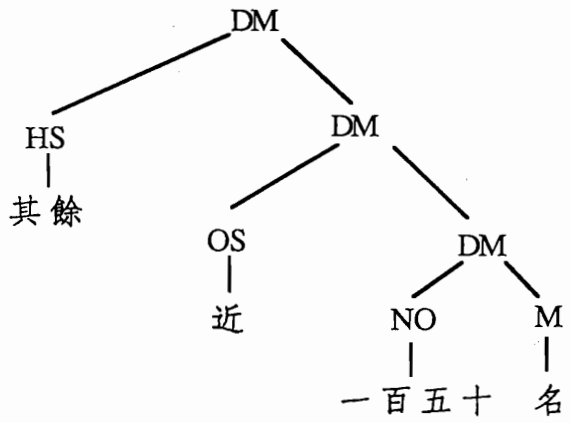
In unmarked cases, a numeral compound occurs at the leftmost position of a compound made up of determinatives. Similar precedence relations also exist among other determinative sets.

In order to account for the precedence order, we propose a tree structure for the general construction of DMs. This tree structure represents two facts: first, that a DM compound is formed compositionally and hierarchically from the simpler constituents such as numerals and measures. Second, that two determinatives belonging to the same level generally do not co-occur.

(11) The Tree Structure of DMs[3]



(12) 其餘近一百五十名



[3]Here M=measure, NO=numerals, XQ are various quantitative determinatives such as interrogative quantitative determinatives. As for DD, DS, and OS, they are demonstrative determinatives, definite specific determinatives, and ordinal specific determinatives respectively. Finally, HS refers to those specific determinatives which have the meaning of "the other".

Based on this tree structure, our DM formation rules begin with the combinations of recurring numerals, numeral compounds, post-nominal modifiers (PNMs), and measures.[4] New DMs can be formed by attaching other determinatives to these basic numeral compounds. Co-occurrence restrictions developed by other linguists as well as by ourselves will be taken into consideration at this stage. For instance, demonstrative determinatives can not co-occur with interrogative determinatives, or with those listed in the DS set. Some example rules can be seen in (13). Please refer to Appendix II for a complete list of rules.

(13)
IN1--> NO* ;
IN2--> NO* {多 ,餘 ,來 } ({萬 ,億 ,兆 }) ;
DN--> (IN1) {點 } IN1 ;
FN--> IN1 {分 之 } IN1 ;
FN--> IN1 {又 } FN ;
NOP1-->IN1 (DESC) ({半 }) (LM) ;
NOP2-->DESC ({半 }) LM ;
NOP3-->IN1 {平 方 ,立 方 } Nfga ({的 }) ;
NOP4-->IN1 (M) PNM ({的 }) ;
NOP5-->M (PNM) ({的 }) ;
NOP6-->{IN2,DN,FN} (LM) ;

Three remarks can be made about the above rules. First, as observed in Lu [12], some adjectives such as大 "big",小 "small",整 "whole" and 長 "long" may be inserted into a DM.[5] However, the measures that can follow 長 "long" are more restricted;

[4] In general, a determinative precedes a measure. But those listed in the PNM set, such as 半 "half", 整 "whole", the situation is quite to the contrary in that most determinatives have to occur after PNM measures.

[5] Lu [12] lists seven such adjectives: 大 "big", 小 "small",長 "long", 寬 "thick", 薄 "thin",滿 "full" and整 "whole". However, owing to dialect variations, 厚 "thick" and薄 "thin" never appear between determinatives and measures in Taiwan Mandarin. As for 滿 "full", we follow Chao [4] as well as CKIP [10] and regard it as a determinative of the WQ subcategory which denotes the concept of wholeness.

actually, only six measures can co-occur with the word 長 "long". Since its productivity is rather low, we will list all the combinations of 長 "long" and the compatible measures directly in the measure set. Second, in Mandarin Chinese a DM may be followed by a clitic 的 "DE" to indicate that it serves as a modifier (Huang [6]), like 三磅的肉 "three pounds of meat" or 兩桌的客人 "two tables of guests." But not every measure can co-occur with the 的 "DE": most classifiers, for example, are incompatible with 的 "DE". For processing efficiency, we list the various combinations of 的 "DE" and the immediately preceding measures in the measure set, too. [6] Third, for the convenience of language analysis, we also consider complex time expressions (14) and reduplicated DMs (15) as a single unit and express them by our DM rules.

(14) 中華民國 八十 年 九月 十日 二時 十分
R.O.C. eighty year September ten day two o'clock ten minute
"ten after two, September tenth, 1991"

(15) 一 個 個
one CL CL
"one by one"

一 瓶 瓶 的
one CL CL DE
"bottle by bottle"

From the above discussion, it is shown that the structures of DMs are quite complicated.

---

[6]However, this does not imply that when 的 "DE" follows a DM, it will be always correct to combine them together. In certain cases, this 的 "DE" should be attached to larger phrase of which the DM is only one of its constituents. For example, in the following two sentences, the 的 "DE" adjacent to the DMs is actually a relativizer relativizing the whole verb phrases.

(i) 坐 前 兩 排 的 學 生
sit front two row DE student
"the students who sit in the first two rows"

(ii) 已 經 喝 完 一 杯 咖 啡 的 人 請 站 起 來
already drink finish one cup coffee DE person please stand up
"Those who finished the first cup of coffee please stand up."

In order to test and modify our determinative sets and formation rules, we construct a rule interpreter and a chart parser.

III. An Interpreter for Regular Grammar and Its Application to Improve DM Rules

We have already shown that DMs in Mandarin Chinese can be expressed by a set of "Regular Expressions". We construct a regular expression interpreter and a chart parser in order to recognize DMs in input sentences. By testing the real input data from corpus, we can iteratively improve our classification and rule sets.

Our system is not the first DM parser. Chuang [11], based on the classifications developed in CKIP [10], creates a set of grammar rules and a program to implement the rules. However, the coverage of his grammar rules is incomplete and his program is procedure oriented which means it has to be modified once the grammar rules have changed.

Our system is divided into two parts: transformation-interpretation, and parsing.[7] At the transformation-interpretation stage the system transfers the grammar rules into a simpler format. The rules are originally in the form of regular expressions.[8] They are transformed into the format known as Chomsky Normal Form (Aho & Ullman [1]). The reason why we did not write it in Chomsky Normal Form originally is because it is easier to write the

---

[7]This original interpreter of the grammar and DM parser is designed and developed by Charles Lee of Stanford University and Yao-Jung Yang cooperatively. All other programs mentioned in this paper are written by Yao-Jung Yang.

[8]In this paper, all the Determinatives and Measure words are defined in symbol sets placed together with the grammar rules. By doing so, it is very convenient to modify the rules as well as the sets when we are running tests. However, this strategy will not be adopted in actual implementation because it will cause data redundancy. The lexical information of Determinatives and measure words must be attached to the words after dictionary lookup.

rules first in the form of regular expressions. On the other hand, it is much easier for computer to interpret the Chomsky Normal Form. The benefit of the interpreting approach is that we can modify the rules over and over again without changing the program. The following is a fragment of our grammar rules before and after the transformation-interpretation stage:

(16)

```
NO      = { ○,一,二,兩,三,四,五,六,七,八,九,十,
            廿,卅,百,千,萬,億,兆,零,幾};

IN1     ->      NO*;

IN2     ->      NO*   { 多,餘,來 }  ( { 萬,億,兆 } );
```

```
NO = {○,一,二,兩,三,四,五,六,七,八,九,十,廿,卅,
      百,千,萬,億,兆,零,幾};
IN1 -> NO IN1;
IN1 -> NO;
_0 = {多,餘,來};
_1 = {萬,億,兆};
_2 -> NO _2;
_2 -> NO;
_3 -> _0 _1;
_3 -> _0;
IN2 -> _2 _3;
_4 = {點};
_5 -> _4 IN1;
```

The parsing part of the program is built according to the concept of "Chart Parsing". The reason why we choose chart parsing as our basic strategy is because the chart data structure can hold all information about words which can then be used in the latter stage of the Information-Based Case Grammar (ICG) (Chen & Huang [5]) parsing process.

However, for actual testing, the program still has to be equipped with a preprocessor and a postprocessor: the former breaks the input article into sentences. Based upon the fact that in general no DMs can cross a punctuation marker, this article can be broken into substrings with punctuations as delimiters before being fed to the parser.[9] The latter reads

---

[9]But, in certain cases, a comma or punctuation mark " " is inserted into a numeral phrase. For example, we may have 五,六 月 間 "during May and June", 第 三、四、五 層 "the third,fourth, and fifth floors", 三、四 百 人

the output chart files produced by the core parser and does a filtering process to eliminate redundant or intermediate results. These two processors can be executed separately from the core parser so that the core parser is kept more adaptive to the other usages.

All the programs mentioned above are developed in the C language on the Borland Turbo C System. Some fragments of the input data and their output forms will be presented and analyzed in the following section.

IV. Discussion of Results

During test runs, postprocessed output is evaluated based upon two factors: first, are all of the DMs in the input article recognized, and second, how many are overgenerated? The former is concerned with the completeness of the rules; the latter is concerned with their accuracy. In the following we define some statistical values for the purpose of analysis.

N_{act} = the number of DMs in the testing article.
N_{ove} = the number of substrings which are recognized but are not DMs.
N_{mis} = the number of DMs in the testing article which are not recognized.
N_{rec} = the number of DMs which are recognized by our system.

"three or four hundred people", 十五，六歲 "fifteen or sixteen years old", etc. These marks either indicate a list (cf. the first two examples), or present an omission resulting from repetition (cf. the last two examples):

三、四百人 = 三百 or 四百人
十五，六歲 = 十五 or 十六歲

At this moment in time, these phrases cannot be correctly recognized for our rules do not take punctuation marks into consideration. The problem should be solved with an appropriate preprocessor.

After testing over 16 articles picked from a corpus,[10] we have:

The recognition rate = (N_{act} - N_{mis}) / N_{act} = 100%
The missing rate = N_{mis} / N_{act} = 0%
The overgeneration rate = N_{ove} / N_{rec} = 39.57%

Article#	N_{act}	N_{rec}	N_{ove}	N_{mis}
1	16	22	6	0
2	71	86	15	0
3	22	40	18	0
4	12	25	13	0
5	13	29	16	0
6	4	22	18	0
7	22	42	20	0
8	20	28	8	0
9	20	38	18	0
10	9	14	5	0
11	22	28	6	0
12	22	33	11	0
13	28	50	22	0
14	26	46	20	0
15	36	59	23	0
16	19	37	18	0
Total	**362**	**599**	**237**	**0**

From the missing rate, it shows that the completeness of the system is perfect. As for the soundness, the overgeneration rate seems to be quite high. However, after carefully studying the test result, we find that the overgenerations are mainly caused by ambiguous word segmentation. Thus these ambiguities can be avoided if we incorperate the DM recognition and word segmentation processes in parallel.

The ambiguities can be further classified into the following different cases:

[10]The corpus is supported regularly by two daily news associations: the Liberty Times and the United Daily News. The amount of data supported per month is about 4 M bytes.

1. Ambiguities resulting from lexical ambiguity. (i.e. polysemy of lexical items)

(17) a. 張 三 一 天 只 要 上 一 節 課
Jangsan one day only want up one CL class
"Jangsan has only one class a day."

b. 草皮 上 三 棵 老 橡 樹
lawn up three CL old oak
"There are three oaks on the front lawn."

c. 上 一 節 課 張 三 沒 來
up one CL class Jangsan not come
"Jangsan was not here for the first part of the class."

This kind of overgeneration is caused by the multi-categorization of individual lexical items. For example, 上 "up" may function as a verb, a localizer, or a determinative. In (17a), it is a verb; in (17b), it is a localizer; only in (17c) does it function as a determinative. We devise the following resolution principle to solve this kind of overgeneration.

> Resolution Principle 1: If the first character of the longest matched DMs is a lexical entry with multi-categories, such as 上 , 下 , 大 , 前 and 後, then both the longest matched DM and the DM without the first character are kept and the ambiguity will be resolved in the parsing stage.

2. Ambiguities resulting from improper word-breaks involving lexical items.

(18) 應 該 把 欠 我 的 錢 還 給 我 了 吧
should BA owe I DE money return I ASP
"(You) shoud return the money you owe me."

要 統 一 分 發 這些 信
want unify distribute these letter
"distribute these letters at the same time"

訂 了 好些 週 刊
order ASP many weekly magazine
"ordered many weekly magazines"

73% of the ambiguities in our test results belong to this type. We also found out that if an ambiguous word break occurs between a lexical word and a DM, the lexical word has the priority, as exemplified in (18). Therefore, we have the following resolution principle:

> Resolution Principle 2: If ambiguous word breaks occur between the words in the lexicon and the DMs, the words in the lexicon should have higher priority to get the shared characters.

3. Ambiguities resulting from improper word breaks involving proper names.

(19) a. 同 時 促成 了 宮本 對 死亡 的 認識
same-time cause ASP miyahon to death DE know
"At the same time Miyahon was forced to realize the meaning of death."

b. 警一分局 忙得不可開交
the-first-precint extremely busy
"The first precint was extremely busy."

The number of proper names is unlimited and therefore can not be exhaustively listed in the lexicon. Thus we are not able to apply resolution principle 2 if the ambiguous word breaks appear between proper names and DMs. So far, we do not have any good solution principles to solve this problem. Fortunately, only 6.33% of ambiguities are of this type.

As was mentioned in the previous paragraph, most of the ambiguities can be disambiguated by word segmentation. This does in fact happen after word segmentation is tried. For instance, in example (18) 應該 "should", 統一 "unify", 分發 "distribute", 週刊 "magazine" are words in lexicon, and thus get the priority in becoming units. Since the characters 該 "should", 一 "one", 分 "distribute", 週 "week" are part of words, no overgenerated DMs in these sentences exist any longer. However, to those overgenerations resulting from lexical ambiguity, the ambiguous word segmentations will still be kept. The following is our new test result derived from combining DM parsing and word break procedure. The recognition rate is $(N_{act}-N_{mis})/N_{act}$=99.17%, and the

ambiguity rate is N_{amb}/N_{act}=12.71%. By ambiguities (N_{amb}) we mean those caused by ambiguous word segmentations and those resulting from proper names.

Article#	N_{act}	N_{amb}	N_{mis}
1	16	1	0
2	71	0	2
3	22	3	0
4	12	2	0
5	13	1	0
6	4	2	0
7	22	3	0
8	20	1	0
9	20	1	0
10	9	1	0
11	22	3	0
12	22	2	1
13	28	14	0
14	26	5	0
15	36	6	0
16	19	1	0
Total	**362**	**46**	**3**

Another type of ambiguity which we do not consider as overgeneration is that some DMs are intrinsically ambiguous with multiple structures, as in (20), or multiple functions as in (21).

(20) 這 下 三 天 也 做 不 完 了
this down three day also do not finish ASP
"Even in three days, we can not finish it."

1) 這 下 三 天
2) 這 下 三 天

(21) a. 簡 直 像 作 夢 一 樣
almost like dream the same
"just like a dream"

b. 她 用 手 一 指
she use hand one point
"She pointed with her finger."

c. 不 知 該 帶 孩子 去 那 兒 玩
not know should bring child go where play
"(I) don't know where to take the children to play."

For such cases, the ambiguity remains to be resolved by parsing.

V. Applications and Concluding Remarks

We pointed out at the beginning of this paper that the combinations of DMs are infinite, and thus can not be exhaustively listed in the dictionary. Moreover, they occur quite frequently in the text. In order to solve this unavoidable problem in parsing, we build a DM parser to be a supplement of the lexicon.

The motivation for us to build this DM parser is to support the word segmentation module of the project developed in the Institute of Information Science, Academia Sinica, whose final goal is to establish a knowledge representation model of Mandarin Chinese. However, the word segmentation module depends heavily on a dictionary, which does not hold a complete list of DMs. With this parser, all those previously unrecognized DMs can be recognized.

Another application of our DM parser involves improving the efficiency of the phonetic input of the Mandarin Chinese. The most common idea to improve the efficiency of the phonetic input method is to utilize a lexicon with a phonetic code of every Chinese word as a key index because the more syllables a word has, the fewer homophones it possesses. With this parser, we can recognize the DMs by their phonetic spelling and greatly reduce the homophonic ambiguity.

In this paper, a DM parser together with some test results are presented. After scrutinizing a large amount of linguistic data, we form some grammar rules to combine determinatives and measures whenever they appear, and a parser to implement these rules. By doing so, all unlisted DMs are recognized. As for the test result, the recognition rate is quite satisfactory, although many pseudo DMs are overgenerated. Nonetheless, these

overgenerated "DMs" are disambiguated by incorperating word segmentation and the DM recognition processes in parallel.

However, at this moment, no semantic features have been taken into consideration. They are not only important to the interpretation of DMs, but also useful for the reduction of ambiguous readings. This is because if co-occurrence restrictions between determinatives and measures can be found, many pseudo DMs will no longer appear. But these restrictions largely depend on the semantic compatibility existing between determinatives and measures. We hope in the near future that these semantic features may be added to our rules to reduce overgenerations, and thus reduce ambiguous readings.

After undergoing large amounts of testing, the rules and sets are proved to be quite complete. The next step is to revise the DM parser program to a finite-automata version instead of an interpreter version in order to improve the performance and reduce the program size. By doing so, the DM parser can be more easily embedded into the whole parsing project.

References

[1] Aho, Alfred V. and Jefferey D. Ullman. 1972. *The Theory of Parsing, Translation, and Compiling*. London: Prentice Hall International.

[2] Allen, James. 1987. *Natural Language Understanding*. Menlo Park, California: The Benjamin/Cummings Publishing Company.

[3] Bresnan, Joan and Jonni M. Kanerva. 1989. Locative Inversion in Chichewa: A Case Study of Factorization in Grammar. *Linguistic Inquiry* 20: 1-50.

[4] Chao, Yuen-Ren. 1968. *A Grammar of Spoken Chinese*. Berkeley: University of California Press.

[5] Chen, Keh-Jiann and Chu-Ren Huang. 1990. Information-based Case Grammar, Proceedings of Coling 90: 54-59.

[6] Huang, Chu-Ren. 1987. Mandarin Chinese NP de: A Comparative Study of Current Grammatical Theories, Ph.D. dissertation Cornell University.

[7] .....1991. Mandarin Chinese and The Lexical Mapping Theory--A Study of the Interaction of Morphology and Argument Changing. *Bulletin of the Institute of History and Philology* 62.2.

[8] Li, Charles N. and Sandra A. Thompson. 1981. *Mandarin Chinese: A Functional Reference Grammar*. Berkeley: University of California Press.

[9] Lyons, John. 1977. *Semantics II*. New York: Cambridge University Press.

[10] 詞庫小組 (CKIP), 1989, 國語的詞分類修訂版, 南港: 中央研究院計算中心

[11] 莊德明 (Chuang, De-Ming),1986,一套中文定--量詞處理系統的研究與設計, 國立清華大學碩士論文

[12] 陸儉明 (Lu, Jianming),1987,"數量詞中間插入形容詞情況考察",語言教學與研究 1987年第四期,pp 53-72

[13] 黃居仁 (Huang, Chu-Ren), 1989,試論漢語的數學規範性質, 中央研究院歷史語言研究所集刊 60.1:47-71.

Appendix I

NO = {〇,一,二,兩,三,四,五,六,七,八,九,十,廿,卅,百,千,萬,
億,兆,零,幾};

ON = {甲,乙,丙,丁,戊};

DESC = {大,小};

PNM = {多,餘,半,出頭,好幾,開外,整,正,許,足,之多};

Ndabe = {清晨,凌晨,早晨,早上,晚上,上午,中午,下午,晨間,午間,
晚間,半夜,午夜,晨,午,晚,傍晚,深夜,曷午,子時,丑時,
寅時,卯時,辰時,巳時,午時,未時,申時,酉時,戌時,亥時};

Ndaac = {光緒,乾隆,廣德,昭和, etc.};

Ndaad = {民國,中華民國,西元,公元, etc.};

Ndabb = {春天,春季,夏天,夏季,秋天,秋季,冬天,冬季};

Ndabd1 = {星期一,星期二,星期三,星期四,星期五,星期六,星期日,
星期天,禮拜一,禮拜二,禮拜三,禮拜四,禮拜五,禮拜六,
禮拜日,禮拜天,週一,週二,週三,週四,週五,週六,週日};

Ndabf = {上旬,中旬,下旬,暑假,寒假,春假, etc.};

TPNM = {半,多,許,整,正};

WQ = {一,全,滿,整,成,一切,一整};

QQ = {多少,若干,幾多};

DQ = {多,許多,很多,好多,好些,少許,多數,少數,大多數};

PQ = {半,若干,有的};

DD = {這,那,哪};

OS = {上,下,前,後,頭,末,次,另,某,近,將近};

DS = {本,貴,敝,什麼,啥,何,別,旁,他};

Appendix II

```
IN1     ->      NO*;
IN2     ->      NO* {多,餘,來} ({萬,億,兆});
DN      ->      (IN1) {點} IN1;
FN1     ->      (IN1 {又}) IN1 {分之} {IN1, DN} ({強,弱}) ({的});
FN2     ->      (IN1 {又}) IN1 {分之} {IN1, DN};
ONP     ->      ON (LM);
NOP1    ->      IN1 (DESC) ({半}) (LM);
NOP2    ->      DESC ({半})  LM ;
NOP4    ->      IN1 (M) PNM ({的});
NOP3    ->      IN1 {平方, 立方} Nfga ({的});
NOP5    ->      M (PNM) ({的});
NOP6    ->      {IN2,DN,FN2} ( LM );
NOP     ->      {FN1,NOP1,NOP3,NOP4,NOP5,NOP6} ;
WQP     ->      WQ (LM);
WQP     ->      WQ (Nff ({的}));
QQP     ->      QQ (NOP5);
QQP     ->      QQ {的} ;
DQP1    ->      {好幾} ({NOP1,NOP2,NOP3,NOP5}) ;
DQP2    ->      {DQ1, DQ2} (LM);
DQP3    ->      {最多, 最少} (DQ3) {NOP1, NOP3, NOP4, NOP6} ;
DQP4    ->      DQ3 {NOP1, NOP3, NOP4, NOP6} ;
DQP5    ->      {DQ1, DQ2} {的} ;
DQP     ->      {DQP1,DQP2,DQP3,DQP4,DQP5} ;
```

```
PQP1    ->      {數} ({NOP1,NOP2,NOP3,NOP5}) ;
PQP2    ->      PQ (NOP5);
PQP     ->      {PQP1, PQP2} ;
XQP     ->      {WQP,QQP,DQP,PQP} ;
CNP     ->      IN1 {年} {IN1,ON} {班} ;
DSP1    ->      DS (LM) ;
DSP2    ->      {該} ({NOP, PQP}) ;
DSP     ->      {DSP1, DSP2} ;
OSP1    ->      {第} NOP1;
OSP2    ->      {每,各} {XQP,NOP,DSP2} ;
OSP3    ->      OS {PQP,NOP1,NOP3,NOP6} ;
OSP3    ->      {前, 後} DESC {半} LM ;
DDP1    ->      DD ({ WQP, DQP, PQP, NOP, NOP2 });
DDP2    ->      {此} ({OSP1, NOP});
OHSP    ->      ({其它, 其他, 其餘} ({的})) {任何} ({NOP1, DSP});
OHSP    ->      ({其它, 其他, 其餘} ({的})) {任何} ({的});
OSP     ->      {OSP1,OSP2,OSP3} ;
HOSP    ->      ({任何}){其它, 其他, 其餘} ({XQP,DDP1,OSP,NOP,ONP});
HOSP    ->      ({任何}){其它, 其他, 其餘} ({的});
STDM    ->      IN1 {分} (IN1 {秒} (IN1));
STDM    ->      IN1 {秒} (IN1);
TDM1    ->      IN1 {時,點,小時} (STDM) (TPNM);
TDM1    ->      IN1 {時,點,小時}  IN1 {刻} (TPNM);
TDM2    ->      ({Ndaac,Ndaad)}{元}{年}({元}{月}(IN1 ({日,號})));
TDM2    ->      ({Ndaac,Ndaad)}IN1 {年}({元}{月}(IN1 ({日,號})));
```

```
TDM2    ->      ({Ndaac,Ndaad}){元}{年}(IN1 {月}(IN1 ({日,號})));
TDM2    ->      ({Ndaac,Ndaad})IN1 {年}(IN1 {月}(IN1 ({日,號})));
TDM3    ->      ({Ndaac,Ndaad}){元}{年}{元}{月份};
TDM3    ->      ({Ndaac,Ndaad})IN1 {年}{元}{月份};
TDM3    ->      ({Ndaac,Ndaad })){元}{年}IN1{月份};
TDM3    ->      ({Ndaac,Ndaad})IN1 {年}IN1{月份};
TDM4    ->      IN1 {月}(IN1 ({日,號}));
TDM4    ->      {元,正,上,下,每,本}{月}(IN1 ({日,號}));
TDM5    ->      IN1 {日,號};
TDM6    ->      {TDM2,TDM4,TDM5}({Ndabb,Ndabd1,Ndabf})(Ndabe)(TDM1);
TDM7    ->      Ndabd1  (Ndabe)  TDM1 ;
TDM8    ->      Ndabd1  (Ndabe)  (TDM1) ;
TDM9    ->      Ndabe  TDM1 ;
TDM10   ->      {每, 上, 下, 本} ({個}) TDM8 ;
LLP     ->      IN1 {度} (IN1 {分} (IN1 {秒}));
ADP     ->      (IN1 {段})(IN1 {巷})(IN1 {弄})IN1({之} IN1)
                {號}(IN1 {樓}) ;
TMP     ->      {攝氏,華氏} ({零下}) {IN1,DN} {度} ;
DM      ->      {FN1,ONP,NOP1,NOP2,NOP3,NOP4,NOP6,XQP,CNP,DSP,OSP,
                OHSP,DDP1,DDP2,HOSP,STDM,TDM1,TDM2,TDM3,TDM4,TDM5,
                TDM6,TDM7,TDM9,TDM10,LLP,ADP,TMP} ;
```

限制式滿足及機率最佳化的中文斷詞方法

張俊盛　陳志達　陳舜德

國立清華大學資訊所

摘 要

本文提出一個使用機率模式的系統，解決斷詞上的問題。首先將斷詞轉換成限制條件滿足問題，再以詞獨立出現機率做動態規劃的最佳化決定，運算時間與詞的長度成幾近線性的關係，詞的長度也不受限制。

一. 介紹

中文句子的結構中，單獨的漢字，並非句法上及語意上的最小元素。詞才是中文中能夠獨立出現，並自由運用的最小單位[5]。因此研究句法語意分析，必須以詞作爲基本單位。中文斷詞的目的，主要是爲了簡化自然語言處理系統的運算步驟，避免考慮太多不可能的斷詞情形。

西方語言並沒有斷詞問題，因爲西方語言的字彙單獨就具有獨立的語意。但要用中文表達一個最基本有意義的概念時，卻可能要用數個漢字來組成。例如相對於英語的grape，在中文要使用兩個漢字「葡萄」。再考慮三個字長的字串：好學生。在這個字串中，隱藏著五個詞彙：好、學、生、

好學、學生。總共可以找出三種組合，卻只有一種組合是合理的：好|學生。

將輸入句子的字串順序，轉換成詞語順序的過程就叫做斷詞。

目前以文法爲主要架構的中文自然語言處理系統中，斷詞是系統辨識中文輸入句子不可缺少的步驟。而愈來愈多的中文電腦應用領域，如光學字體辨識、語音辨識、文書校對工具、資料檢索、簡繁體互換以及中文輸入法等等，也可望利用斷詞做爲新的輔助方法。

雖然詞彙現象是屬於整個文法中的一部份，而且因爲中文的詞彙可以靈活運用，常常參雜了句法成份。但是大部份的情形，仍然可以不牽涉句法分析就決定詞彙在句子中的位置。

以往電腦斷詞的研究，可略分爲由構詞規則爲出發點的法則式斷詞法[1,2]，以及利用統計數據資料歸納爲判斷憑據的統計式斷詞法[3,9]。

法則式斷詞法強調的是語言現象。[1]就利用有些漢字大多只出現在詞彙的首個位置或末個位置的現象，來簡化斷詞步驟。[2]則提出看法，認爲斷詞只是自然語言的一部份，斷詞程式應該不能錯失掉任何可能的正確結果。雖然[1,2]都嘗試利用構詞現象來輔助斷詞，然而語言學界的詞彙研究並未提供簡易的方法，可用來做爲斷詞的依據。因此[1]使用的是『長詞優先』的規則，雖然這個規則在某些情況會失敗。[2]則嘗試利用『字詞的結合性』來解釋斷詞的現象，然而並未提出實際的作法。

相對的，統計式斷詞則著眼於大量資料的處理。認爲語言的性質，可以從大量的語料庫，經由數學模式獲得。此類系統目前以蔡文祥與范長康的鬆弛法[3]，及Sproat-Shih的統計式斷詞法[9]爲代表。[3]首先引進機率模式，利用影像處理常用的鬆弛法，成功地解決斷詞問題上相當複雜的情形。[9]使用的方法較爲特別，他將詞看做是一串相依出現機率特別高的漢字，從大量語料中得到二個漢字相依出現的機率，並利用一階馬可夫機率模式做爲斷詞依據。藉此方法Sproat-Shih無須使用人工預建的詞典，在處

理詞的長度不超二個漢字的情形就能有很好的效果。因爲忽略了構詞規則，所以統計式斷詞法，在正確率方面會有一定的瓶頸。另一方面，因爲方法的不同，前者須要反覆的運算，速度較慢。而後者則須建一個龐大的統計表，並且目前也侷限在只考慮詞長度不超過二個漢字的情形。

近來的做法也有將兩種方法合併的。[12]利用中文中自由詞素(free morpheme)或附著詞素(bound morpheme)的性質簡化斷詞步驟。並且利用字詞詞性的一階馬可夫機率依機率值大小排列所有的可能結果。然後再使用HPSG的剖析器(Head-driven Phrase Structure Grammar parser)，首次在斷詞中利用句子的文法及語意做爲斷詞依據。

雖然[3,9,12]都能達到95%的以上斷詞正確率，然而因爲速度或是記憶體容量的限制，這些系統在實用性上都有一些限制。[3,9]目前都只能處理詞長不超過二個漢字的情形。[3]的方法雖然並未限制詞的長度，然而這將讓原本已偏慢的速度更加惡化。[9]的方法則很難擴張到更高階的機率模式，因爲受限於統計表的大小必須隨著機率的階數呈次方關係增加。雖然[12]展示了一個翻譯系統下的斷詞子系統，但卻使用文法做爲判斷依據。因爲中文文法在自然語言處理上的諸多現象比斷詞還要複雜，應用在斷詞上可能會有實用性的問題。

本文所提出的斷詞系統則完全以機率值爲判斷依據。利用獨立的機率模式，實驗的結果顯示，系統斷詞的正確率並不亞於複雜的鬆弛法。我們將將斷詞轉換成限制式滿足問題(Constraint Satisfaction Problem)，再以詞獨立出現機率做動態規劃(Dynamic Programming)的最佳化決定(Statistical Optimization)。減少了運算時間，詞的長度也不受限制。未來，我們將進一步利用構詞法則，補充詞庫的不足。

以下在第二節中，我們將描述斷詞如何轉換成CSP的問題形式，並說明統計式的最佳化模式。第三節將說明系統架構，並提出實驗數據及結果。第四、五節分別爲本文的討論與結論。

二. 機率式斷詞

(i)斷詞的問題描述

CSP (Constraint Satisfaction Problem)是一種用來解決限制式滿足的問題。著色問題，圖形辨認問題以及排程問題(scheduling problem)都可以利用CSP的方法來解決[7]。在給定一組限制式後，CSP的目的就在找出能滿足所有限制式的一組變數解。

一個二元的CSP問題，其限制式只涉及兩個變數：

給定n個變數$X_1,X_2,...,X_n$，以及一組二元關係的限制式

$K_{i,j}$：$(X_i,X_j)\in K_{i,j}$

找出n個變數的一組值$(x_1,x_2,...,x_n)$滿足所有的限制式$K_{i,j}$。

斷詞問題可以用CSP來解決。

假設斷詞的輸入是由n個漢字所組成的句子，

$S=(C_1,C_2,..,C_n)$

令C_i與C_{i+1}相鄰漢字的間隔爲X_i。斷詞的目的就是將任一個X_i，標上「斷」或「不斷」的註記。而且，被任二個最接近的斷點間隔所分開的漢字，必須是一組詞。

爲了說明方便，我們各用X_0及X_n代表最前與最末的間隔，並用符號"^"與"="表示間隔「斷」或「不斷」。

| C_1 | C_2 | C_3 | .... | C_n |

X_0 X_1 X_2 X_3 X_{n-1} X_n

以下，我們對限制式，作更進一步說明：

對S中的任一串相連漢字$W_{i,j}=(C_i, C_{i+1}, ..., C_j)$，如果$W_{i,j}$被認可爲中文詞，則依下列狀況處理之：

(i). i=j

意味著C_i是一個單字詞。X_{i-1}及X_i的值可以都是"^"。設定限制式(^，^)$\in K_{i-1,i}$。

(ii). i<j

此種情形表示(C_i, C_{i+1}, ...,C_j)是一組多字詞。因此X_{i-1}, X_i, ..., X_j 的一組解可能是(^, =, =, ..., =, ^)

設定限制式：

(^,=) $\in K_{i-1,i}$

(=,=) $\in K_{i,i+1}$

(=,=) $\in K_{i+1,i+2}$

.....

(=,=) $\in K_{j-2,j-1}$

(=,=) $\in K_{j-1,j}$

因此當輸入一個句子之後，我們就可以逐一檢查句中任一個漢字爲首的右向鄰接漢字是否爲一組詞彙。如果是，就設定限制式的內容。

下面我們舉實例說明：

C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
把	他	的	確	實	行	動	作	了	分	析
X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}

句子中隱藏的所有詞彙，列出如下：(下面的說明是基於假設：漢字"析"不在詞彙庫中)

C_1 = 把　　(^,^)$\in K_{0,1}$

C_2 = 他　　(^,^)$\in K_{1,2}$

C_3 = 的 C_3C_4=的確　　(^,^)$\in K_{2,3}$ (^,=)$\in K_{2,3}$ (=,^)$\in K_{3,4}$

C_4 = 確 C_4C_5=確實　　(^,^)$\in K_{3,4}$ (^,=)$\in K_{3,4}$ (=,^)$\in K_{4,5}$

C_5 = 實 C_5C_6=實行　　(^,^)$\in K_{4,5}$ (^,=)$\in K_{4,5}$ (=,^)$\in K_{5,6}$

C_6 = 行 C_6C_7=行動　　(^,^)$\in K_{5,6}$ (^,=)$\in K_{5,6}$ (=,^)$\in K_{6,7}$

C_7 = 動 C_7C_8=動作　　$(\wedge,\wedge)\in K_{6,7}$ $(\wedge,=)\in K_{6,7}$ $(=,\wedge)\in K_{7,8}$

C_8 = 作 $(\wedge,\wedge)\in K_{7,8}$

C_9 = 了 $(\wedge,\wedge)\in K_{8,9}$

C_{10}= 分 $C_{10}C_{11}$=分析　$(\wedge,\wedge)\in K_{9,10}$ $(\wedge,=)\in K_{9,10}$ $(=,\wedge)\in K_{10,11}$

爲了避免不同詞的互相干擾，以上每一個詞所設定限制式中的"="，應與其他詞所設定者不同。不過爲了表達清晰起見，我們把所有的"="都寫成一樣。整理之後可列出限制式如下：

$K_{0,1}$ = { (^,^) }
$K_{1,2}$ = { (^,^) }
$K_{2,3}$ = { (^,^) (^,=) }
$K_{3,4}$ = { (^,^) (=,^) (^,=) }
$K_{4,5}$ = { (^,^) (=,^) (^,=) }
$K_{5,6}$ = { (^,^) (=,^) (^,=) }
$K_{6,7}$ = { (^,^) (=,^) (^,=) }
$K_{7,8}$ = { (^,^) (=,^) }
$K_{8,9}$ = { (^,^) }
$K_{9,10}$ = { (^,^) (^,=) }
$K_{10,11}$ = { (=,^) }

根據文獻中有關研究，CSP中的限制式，可以預先消去相互抵觸的限制條件，而且不會影響最後的結果。這種方法叫做Arc Consistency algorithm，簡稱AC[7,8]。

以上面的例子來說，經過AC簡化限制條件後，可以拿掉：

$K_{9,10}$={(^,^)}

這是因爲由$K_{10,11}$，我們知道X_{10}="="，因此$K_{9,10}$中的(^,^)不會爲最後的結果所滿足，將它消除並不會影響最後的結果。

文獻中二元CSP的研究也證明，若將變數視爲結點，限制式視爲連接結點的邊，若形成一樹狀結構，我們就可以直接從限制式中，找出解答，而不須要回溯(backtrack)。斷詞問題轉換成二元CSP後，形成線性串列的形態，爲樹狀結構的特例，因此也可以不經回溯，就直接找出答案。但是因爲合乎限制式的答案個數依然太多，所以必須另外使用評估方法，幫助找出最可能的結果。

(ii)以機率值決定斷詞

令輸入句子$S=(C_1, C_2, \ldots, C_n)$，通過CSP測試的一個斷詞結果是$(W_1, W_2, \ldots, W_k)$。其中任一組$W_i$都是詞典中記載的詞。查出每一個詞彙的出現機率，使用獨立機率，做爲評估斷詞可能性的依據。因此句子S，最可能的斷詞結果是：

$$\text{argmax}P(W_1,W_2,\ldots,W_k|C_1,C_2,\ldots,C_n)$$

$$\doteqdot \text{argmax}P(W_1)*P(W_2)*\ldots*P(W_k)$$

$$\doteqdot \text{argmax}\prod_{i=1,n}P(W_i)$$

其中$P(W_i)$是經大量語料統計的詞彙獨立出現機率

以獨立機率模式應用於斷詞問題，方法雖不複雜，卻隱含著以往研究在斷詞上的諸多看法，以下將獨立機率與其他方法比較：

(a)長詞優先性質

獨立機率模式隱含長詞優先性質，因爲詞彙的出現機率大多遠小於1，所以項數較少有助於機率乘積的增加，使得斷詞結果傾向於長詞。

(b)詞素的自由性與束縛性

詞素(morpheme)是語言上最小有意義的單位。可以單獨使用的詞素叫做自由詞素，必須和其他詞素連用的詞素叫做附著詞素。

附著詞素不會單獨出現，其獨立出現機率必然趨近於0，也因此降低在斷詞中出現的可能。

(c)與Sproat-Shih統計式斷詞的比較

Sproat-Shih以相鄰漢字的出現機率做爲斷詞依據，其作用相似於本系統使用的詞彙出現機率。但是Sproat在選擇斷詞時使用的是貪婪法策略(greedy method)，相對於此，獨立機率模式則著眼於整體的最佳化，有助於複雜文句的處理。

(d)與鬆弛法之比較

鬆弛法和獨立機率模式都利用詞彙出現機率做爲判斷依據，雖然鬆弛法和獨立機率模式在作法上有顯著差異，然而由於斷詞問題的性質，使用獨立機率模式簡單的計算而無須繁複的收斂過程，就能得到和鬆弛法相近的結果。

(iii)以動態規劃法增進速度

系統將斷詞問題轉化成爲CSP，字的間隔"斷"或"不斷"是用一組變數表示。對應輸入資料的順序，可以排定變數求值的先後順序。因爲每一個變數的值都只跟前面的變數值有關，而且只考慮機率乘積最大的解，此種性質正好符合使用動態規劃法的兩大要素：多階段的決策過程(multistage decision process)及最佳化原則(principle of optimality)。

由於使用動態規劃法，系統可以避免執行時間與句子長度呈指數函數的關係。因此相近於DeRose標註英文字彙詞性的效果[6]，系統可以在幾近線性的時間內，完成中文斷詞。

三.實驗說明及結果分析

(1)實驗說明

這個系統目前是在IBM/AT相容個人電腦上使用TURBO PROLOG發展的。利用TURBO PROLOG的external data base功能，系統預先建立一個六萬個詞的詞庫。詞彙的出現頻率取自劉英茂等編著的[常用中文詞的出現次數][4]。詞彙的來源，則由[常用中文詞的出現次數]及[電子辭典][10]合併而成。合併後，扣除部份重覆的詞彙後，系統共收藏有詞彙60274個。合併後的詞庫，依照詞的長度區分，列表如表一。

因爲劉英茂編著的[常用中文詞的出現次數]，是根據大量語料，統計出一百萬個詞彙中，出現頻率最高的四萬詞，並且分別記載其出現頻率。此一資料正好做爲斷詞判斷的依據。但是電子辭典上還有22018個詞彙，是不在常用的四萬個詞彙以內。我們假設這些詞的出現頻率都小於1，並且給予初始頻率值0.99。合併後的詞庫，依照詞彙的出現頻率，列表如表二。

系統的測試資料，隨機取自一個包括科學、評論、小說、散文等約30萬詞的語料庫。目前系統在斷詞方面，已經測試了擷取自不同類別的11篇文章，共43069個漢字的語料。測試語料分由機器及人工斷詞，再交由機器比對，核算出結果。

由於評估系統目地及方法上的不同，此處使用資料檢索文獻上常用的召回率(Recall)及精確率(Precision)做爲計算斷詞正確率的依據。如果令系統產生的斷詞結果爲M，人工斷詞的結果爲P，機器斷詞與人工斷詞一致的部份爲$P \cap M$，則召回率$= P \cap M / M$，精確率$= P \cap M / P$。

其中召回率的著眼點在得知，正確結果有多少比率可以由系統產生出來。系統平均可達到95.97%的斷詞召回率(正確的詞數/測試的總詞數)。精確率則可以衡量系統產生的結果，有多少比率是正確的。系統可以達到91.83%的精確率(正確的詞數/測試的總詞數)。字正確率則以漢字爲計算單位，系統平均可以達到93.89%的正確率。詳細測試結果列表如三。表四並根據召回率對錯誤情形做進一步分析。

由於Sproat-Shih的系統只考慮簡化的二字詞斷詞問題，爲了在比較上有所依據，我們另外統計二字詞的情形。在人工斷詞的資料中，找出二字詞有11312個，系統辨認到其中的10542個，召回率爲93.19%。而在機器斷詞的輸出中，共產生二字詞10755個，與人工斷詞相符的部份有10542個，精確率爲98.02%。詳細測試資料如表五。

系統斷詞的速度，目前每秒大約可以分別完成6.7組詞的測試。因爲受限於TRUBO PROLOG資料庫的速度限制，因此如果將它移轉到更佳的環境下，預料還可大幅改善其速度。

(2)結果分析

以下我們從測試結果中，摘選一些斷詞錯誤的例子。例句中畫有底線之處，爲錯誤的斷詞。句尾括弧中的編號，爲該例句的來源檔案編號。

(1) 缺乏¦ 讓人¦ 重¦ 讀¦ 的¦ 吸引力 (l-2)
(2) 攻擊¦ 人類¦ 並¦ 以¦ 人爲¦ 食 (j-14)
(3) 出現¦ 不¦ 同¦ 意識¦ 狀態 (j-10)
(4) 患¦ 精神¦ 官¦ 能¦ 異常¦ 的¦ 人 (j-10)
(5) 出¦ 現在¦ 遠離¦ 文明¦ 的¦ 沼澤 (l-4)
(6) 相¦ 對於¦ 國內¦ 食品¦ 廠家 (a-1)
(7) 從¦ 非洲¦ 的¦ 雨林¦ 草¦ 原 (j-14)
(8) 到處¦ 有¦ 鱷¦魚¦ 游¦ 來¦ 游¦ 去 (l-4)
(9) 昊¦ 昊¦ 的¦ 青天¦ 燦¦ 燦¦ 的¦ 白日 (g-1)
(10)經理¦ 小¦ 室¦ 德¦ 太¦ 郎¦ 離開¦ 大廳¦ 後 (l-2)

表一　詞彙長度分佈表

詞彙長度	詞彙個數
1	8097
2	39054
3	8046
4	4723
5	263
6	71
7	12
8	8
9	1
合計	60275

表二　詞彙頻率分佈表

詞彙出現頻率	詞彙個數
0.99	22018
1	16791
2	5929
3	3078
4	1949
5	1365
6	970
7	770
8	642
9	497
10	390
11 ~ 50	3909
50 ~ 54438	1967

表三　測試結果

檔案編號	總字數 A	人工斷詞總詞數 B	機器斷詞總詞數 C	機器與人工斷詞相符部份 D	總錯誤字數 E	召回率	精確率	字正確率
a-1	4314	2741	2867	2643	243	96.42%	92.19%	94.37%
b-1	5122	3291	3464	3137	346	95.32%	90.56%	93.24%
e-1	3178	2332	2463	2243	206	96.18%	91.07%	93.52%
g-1	4457	3201	3305	3104	205	96.97%	93.92%	95.40%
j-10	3361	2237	2372	2140	236	95.66%	90.22%	92.98%
j-14	3932	2734	2800	2666	146	97.51%	95.21%	96.29%
l-2	3588	2432	2554	2295	284	94.37%	89.86%	92.08%
l-3	3950	2720	2937	2517	412	92.54%	85.70%	89.57%
l-4	4709	3333	3466	3209	260	96.28%	92.59%	94.48%
l-8	3582	2546	2628	2469	169	96.98%	93.95%	95.28%
p-1	2876	2104	2155	2053	126	97.58%	95.27%	95.62%
合計	43069	29671	31011	28476	2633	95.97%	91.83%	93.89%

召回率 = D / B
精確率 = D / C
字正確率 = 1- E / A

表四　對召回率錯誤發生原因進一步分析

錯誤原因	因詞庫未收藏詞導致的錯誤				因機率值值而誤斷	錯誤詞數合計	正確詞數合計	人工斷詞總詞數	召回率
檔案編號	一般複合詞	純複合詞	重疊構詞	地名,人名或譯名					
a-1	21	51	0	18	8	98	2643	2741	96.42%
b-1	17	54	0	59	24	154	3137	3291	95.32%
e-1	47	26	0	0	16	89	2243	2332	96.18%
g-1	22	22	14	25	14	97	3104	3201	96.97%
j-10	29	14	0	21	33	97	2140	2237	95.66%
j-14	25	17	1	10	15	68	2666	2734	97.51%
l-2	5	8	1	98	25	137	2295	2432	94.37%
l-3	8	18	7	151	19	203	2517	2720	92.54%
l-4	8	26	3	64	23	124	3209	3333	96.28%
l-8	18	17	0	23	19	77	2469	2546	96.98%
p-1	12	20	3	3	13	51	2053	2104	97.58%
合計	212	273	29	472	209	1195	28476	29671	95.97%
百分比	0.71%	0.92%	0.10%	1.59%	0.70%	4.03%	95.97%	100.00%	95.97%

表五　二字詞統計資料

(i) 召回率

檔案編號	人工斷詞個數	與機器斷詞不相符的個數	召回率
a-1	1284	48	96.26%
b-1	1509	102	93.24%
e-1	696	41	94.11%
g-1	1054	64	93.93%
j-10	993	63	93.66%
j-14	1023	45	95.60%
l-2	989	93	90.60%
l-3	1012	169	83.30%
l-4	1163	80	93.12%
l-8	915	35	96.17%
p-1	674	30	95.55%
合計	11312	770	93.19%

(ii) 精確率

檔案編號	機器斷詞個數	與人工斷詞不相符的個數	精確率
a-1	1268	32	97.48%
b-1	1432	25	98.25%
e-1	678	23	96.61%
g-1	1004	14	98.61%
j-10	945	15	98.41%
j-14	988	10	98.99%
l-2	921	25	97.29%
l-3	865	22	97.46%
l-4	1106	23	97.92%
l-8	890	10	98.88%
p-1	653	14	97.87%
合計	10755	213	98.02%

斷詞錯誤的原因可略分爲兩類，第一類是因爲詞庫未收藏正確的詞彙，導致斷詞失敗。第二類則由於機率模式的限制，最佳機率值的句子並不是正確的結果。進一步分析錯誤發生的原因歸納如下：

1.因機率值導致的錯誤

此類錯誤在測試時共發生209次，約佔全部錯誤詞數的17%。

(1) 假設m個漢字長度的詞 $W = (C_1,C_2,...,C_m)$，其中包含著兩個詞：$W_1=(C_1,...,C_k), W2=(C_{k+1},...,C_m)$。

(i)如果 $P(W)>P(W_1)*P(W_2)$，系統會優先選擇W，而不會選擇 $W_1|W_2$。

雖然大部份的情形長詞優先短詞的規則會成立，但在少數的情形下，也有例外。 例如：例句(1,2)。

此種情形表示著長詞之內包含著短詞，這些短詞不但可以單獨使用，而且在長詞中的排列順序又正好符合句子的文法結構。主述、動賓等複合詞較易發生這種情形。

爲了要保留所有可能的情形，可以在詞典建構時，在這類詞彙上特別記載此種情形。因此，除了選出出現頻率較高的長詞外，還能在剖析回溯時優先考慮此種可能。

(ii)反之如果 $P(W) < P(W_1)*P(W_2)$，系統優先選擇短詞$W_1|W_2$，而非長詞W。在這種情形下，詞彙即使已被詞庫收藏，也不會被選到。例如：例句(3,4)。

產生這種現象的詞彙，詞彙中的漢字單獨出現的頻率一定非常高。這類的詞彙爲數並不多，也常常是非必要的。

以例句(3)的"不同"爲例，實際上的意義是"不+同"。類似的詞彙還有"這次"="這+次"，"一個"="一+個"。

但是例句(4)的"官能"，就無法輕易從漢字原來的組合看出意思來。爲了解決這類問題，系統可以提供次佳選擇來配合句法剖析器，讓錯誤可以彌補得回來。所以在系統優先提供機率值最高的結果後，仍然須要保留長詞，以便剖析程式回溯時可做爲考慮。

(2)假設m個漢字的一串漢字$W=(C_1,C_2,...,C_m)$，W_1及W_2是W中包含的一組詞彙，且$W_1=(C_1,...,C_j)$，$W_2=(C_{j+1},...,C_m)$。如果W中也包含另一組詞彙，$W_3=(C_1,..., C_k)$，$W_4=(C_{k+1},...,C_m)$，而且$P(W_1)*P(W_2) > P(W_3)*P(W_4)$，系統會優先選擇$W_1|W_2$，而非$W_3|W_4$。假如正確的斷詞結果是$W_3|W_4$就會發生錯誤。

系統因機率值而產生的錯誤，大部份屬於這一類。例如例句(5,6)是屬於這種情形。這是因爲出現的頻率雖然較高，局部卻不合句法。單獨地使用詞彙出現頻率做爲斷詞依據，是無法考慮到這種現象的。

以例句(5)爲例，"|出|現在|"的頻率雖然高過"|出現|在|"，但是前者卻可能較不合語法。對於此種情形，一個值得考慮的解決方式，可以加入詞性的相連出現機率來輔助斷詞。再以同一組詞爲例："出"是Va:【動作不及物動詞】，"現在"是Nd：【時間詞】，"出現"是Vh：【狀態

不及物動詞】，"在"是Pe：【時間或地方標誌,後接動作所發生的時間或場所】。因爲系統的詞性相連出現機率顯示，P(VaINd) < P(PeIVh)，表示後者的斷法較易出現在句子中，因此可能會是較正確的斷法。

2.因系統未收藏的詞彙導致錯誤

此類錯誤在測試中共發生986次，約佔全部錯誤83%，顯見此類問題的普偏。分析發生錯誤的詞彙，可歸納爲三大類：複合詞、構詞、地名人名及譯名，以下分別討論。

(i)複合詞

詞素依照本身的意義，還可區分成實詞素及虛詞素兩種。實詞素有實在的意義，虛詞素則沒有[12]。根據[11]的定義，複合詞是由兩個或以上的實詞素組合而成的單位。完全由自由詞素結合而成的複合詞就稱爲純複合詞(pure compound)。若是複合詞中包含有附著詞素，而且全部是實詞素，那麼就叫做一般複合詞(general compound)，否則如果包含了任一個虛附著詞素，就稱爲純詞(pure word)。詞素是虛或實，差別在語意，對斷詞系統而言並無差異。因此此處將純詞也歸類於一般複合詞中，和純複合詞相互比較。

附著詞素因爲須與其他詞素連用，不能獨用，可用來幫助減少一些組合情形的考慮，並可藉以找出包含有附著詞素的一般複合詞。例句(7)的"草原"就是純詞，因爲"草"和"原"都具有自由詞素的用法。然而例句(8)的"鱷魚"卻是一般複合詞，因爲"鱷"只具備附著詞素的用法。

由表四得知，實驗中986個因爲詞庫未收藏的詞彙導致的錯誤，有212個是屬於一般複合詞，273個是屬於純複合詞。(實驗並未實際針對每一個詞彙由語意分析其詞素成份。任一個詞素，只要詞庫中登錄有自由詞素的用法，就不再做附著詞素的考慮。)

這意味著實驗中的212個錯誤，佔全部測試資料的0.71%，可望透過此一簡單性質，而無須使用複雜的規則來幫助辨認。

不過目前這種作法仍隱含著兩個問題：

1. 附著詞素應與那些鄰接的詞素合併目前仍無法得知。
2. 盲目地強迫附著語素與其它語素合併成一個較長詞彙的作法，是否會造成更多的錯誤。

雖然附著詞素從語意上來看是不完整的，然而卻不知道附著詞素和句子中的那些部份形成一個詞，並且也不知道連成的詞會多長。

另一方面，因爲具有附著詞素用法的漢字數量很多，這些漢字多數又同時具有自由詞素的用法，如何不使原來當作自由詞素用法的詞彙被誤用爲附著詞素，仍是一個問題。

(ii) 重疊構詞

重疊構詞具有明顯的特徵，與其他構詞現象相比較，辨識時不須使用太多規則，即使盲目匹配也能得到很好的改進效果。例句(9)爲此種錯誤。因重疊構詞產生的錯誤有29個。

(iii) 地名、人名或譯名

與複合詞或構詞相比較，此類詞彙不具共同特徵，目前沒有有效的方法可用來輔助辨認。例句(10)爲此種錯誤。但注意到當系統未正確辨識詞彙時，斷詞結果往往產生許多的單字詞，而且單字的排列也不按語法，往往使斷詞或詞性標示的機率數值異常偏低。

此類錯誤非常普遍，共有472個，佔全部測試資料的1.59%。

四. 討論

1. 以詞庫做爲詞彙的查詢依據，將會因爲詞庫的容量限制，而影響斷詞的正確率。事實上除了構詞法則所衍生的詞彙以外，有一部份的詞彙是不可能完全收納於詞庫中。尤其在分析報紙社論時，發現根本無法一一將這些可能稍縱即逝的詞彙收集在詞庫內。這反應出一個現象，那就是中文詞彙的組成現象非常活躍。因此對於這個問題，如果不從詞素的層次來分析，還是會不斷遭遇新詞的困擾。[12]曾提出以語法律及語意關係，來判斷複合詞的存在並且預測詞性。
2. 即使存在無限大的詞庫，能夠包容所有詞彙，還是會有許多情形是目前的斷詞系統所無法解決的。例如漢語語法中的"併入現象"，若是發生在複合詞中，就會使詞彙產生變形，而無法從字典中查到。比方說述賓式複合詞的"生氣"，可以在動詞詞素及名詞詞素中併入名詞組，而說成"生你的氣"。這種現象無疑的表示，中文詞彙靈活運用的程度可以超越詞彙的層次。因此，與其在斷詞程式盲目使用簡陋的規則，還不如將這種問題提升到斷詞之後的構詞步驟再解決。
3. 中文的許多詞彙，例如人名或地名等專有名詞，若不在詞庫的收藏範圍之列，根本缺乏可資認定的規則，也很難單純從句法結構就認出此類詞彙。這類情形除了對句法的了解外，往往還需要前後文的語意以及常識輔助，才會有較正確的判斷。
4. 斷詞系統對複合詞的認定，向來沒有一定的標準。例如"棒球手套"或"棒球|手套"，"木棒"或"木|棒"，不同的斷詞方法以及不同的應用範圍，都可能會有不同的看法。這種情況在複合詞的組成份子都是自由詞素時，尤其紛歧。因爲當詞素組成一個較大單位的詞彙時，語意會跟著改變。因此，如果詞庫中並沒有收藏這個複合詞，那麼在由詞素合成詞彙時，必然要有某種方法來指示，合成過程所造成語意上的改變。

五.結論

獨立機率模式在中文斷詞上有很好的表現，可用來輕易架構在其他系統上，提供一種高效率的前置詞彙辨識處理方法。

機率方法可用來幫助語言處理系統大量減少繁複的規則，對於句子的表面結構，往往有出人意表的效果。但是由於機率的方法過於直覺，少量的規則可能無法避免，因此我們將進一步使用構詞法則，來幫助斷詞。

未來我們還將藉助機率模式，幫助句子中專有名詞的認定，並且利用詞性標示的結果輔助斷詞做更精確的判斷。

謝詞

本文研究得到國科會補助，計畫編號NSC80-0408-E011-07，謹此致謝。實驗中所採用的部份詞彙資料，來自工研院電通所技術移轉的國語日報電子辭典。感謝原始發展的中研院詞庫小組，以及電通所王明松先生的技術支援。清大語言所鄭縈小姐給予我們有關辭彙上的許多建議，助理林東游小姐幫忙語料庫的建立，以及區思萍、汪莉娟、陳瑋芬同學協助斷詞結果的人工校正，在此一併申謝。

參考書目

[1] 何文雄，中文斷詞的研究，碩士論文，國立台灣工業研究技術學院，1983.

[2] 陳克健、陳正佳、林隆基，中文語句分析的研究-斷詞與構詞，技術報告TR-86-004，中央研究院，1986.

[3] C.K.Fan and W.H.Tsai, 1987, "Automatic word identification in Chinese sentences by the relaxation technique," Proc. of National Computer Symposium, 1987, pp.423-431.

[4]劉英茂等，常用中文詞的出現次數，六國出版社，1975.

[5]趙元任，中國話的文法，中文大學出版社，1982.

[6] DeRose, S.J., Grammatical Category Disambiguation by Statistical Optimization, Computational Linguistics 14, 1988, pp.31-39.

[7] Dechter, R. and J. Pearl, Network-Based Heuristics for Constraint-Satisfaction Problems, J. of Artificial Intelligence 34, 1988, pp.1-38.

[8] Mackworth, A.K. and E.C. Freuder, The Complexity of Some Polynomial Network Consistency Algorithms for Constraint Satisfaction Problem, J. of Artificial Intelligence 25, 1985, pp.65-73.

[9] Richard Sproat and Chilin Shih, A Statistical Method for Finding Word Boundaries in Chinese Text, Coumputer Processing of Chinese & Oriental Languages, Vol. 4, March 1990.

[10]工業技術研究院電子工業研究所，中文電子詞細部設計手册，1990.

[11]中央研究院計算中心中文知識庫小組，國語的詞類分析，技術報告T0002，1989.

[12]陳克健等，國語中的複合詞和語言剖析，76年全國計算機會議論文集，1987，415-422頁.

測試語料來源

a-1.中東戰爭相關報導

中國時報 第9版80.1.19.

中國時報 第10版80.1.19.

工商時報　第2版80.1.19.

b-1.社論

中國時報 第四版80.1.13.

工商時報 第二版80.1.13.

工商時報 第二版80.1.19.

g-1.中國現代散文評析

蒲公英的歲月　余光中　長安出版P743-P748

e-1.天文望遠鏡使用與製作

丸山秀明　　世歲出版P130-151

j-11.焦慮與精神官能病

馬丁博士　桂冠出版P20-P27

j-14.野生動物世界

1.虎　2.豹　3.美洲獅　時代公司出版

l-2.推理雜誌69期 P.193-P.200

l-3.推理雜誌70期　創作推理 P.49-P.59

l-4.推理雜誌72期　違者必死 P.177-P185

l-8.樹與女· 冥府客棧　胡品清譯　爾雅出版P24-P31

p-1.當代中國大陸作家叢刊　女作家卷2

雨，沙沙沙-荒山之戀　王安憶 P192-198

(註:檔案的分類仿照BrownCorpus)

自動化中文電話總機輔助系統

ACTOA: Automatic Chinese Telephone Operator Assistant

許仁榮

交通部電信總局電信研究所

摘要

在通訊發達的現代生活中，電話是不可或缺的通訊工具，對一個具有多具電話分機之部門或公司，總機的角色尤其重要，因此自動化的電話總機輔助系統就成為電信應用上一項新的嘗試，也是 ACTOA 系統的終極目標。本文將介紹以文字做輸出入的 ACTOA 系統的架構和內容，包括：詞典、以詞典和語境為本的「概念分析器」、對話概念的整合、「語境解譯器」，和對話劇本等，並分析系統的測試結果。

1. 簡介

在通訊發達的現代生活中，電話是不可或缺的通訊工具，對一個具有多具電話分機之部門或公司，總機值機人員就成了該部門或公司的通話門房，擔任該部門或公司與外界通話接觸的第一線工作。由於總機值機員的工作繁重，又十分枯躁乏味，所以在最近幾年內，總機值機員的轉接工作有些已經逐漸由機器所取代，打電話去的人可直接撥分機號碼，而不用透過總機值機員轉接，只有該分機在忙線的狀況下才由總機值機員接聽；但是這種半自動化的功能有其限制，因為它只能接受分機號碼，對於不知道分機號碼的轉接工作，乃至於詢問事項等複雜對話的處理，仍然必須以人工的方式（由總機值機員接聽）來完成，因

此自動化的電話總機輔助系統就成為電信應用上一項新的嘗試，也是ACTOA (Automatic Chinese Telephone Operator Assistant) 的終極目標。

在自動化的電話總機輔助系統中，客人可以用語音和交換機系統溝通，只有在通話狀況不良（如有雜音），或一直都聽不懂的情形下，系統才自動轉由總機值機員接聽。因此，一個具有總機值機員能力的交換機系統，就必須具有接受語音、加以辨認、理解，並產生適當應對文句，以語音的形式輸出等功能，具有這些功能的電話自動轉接系統，就稱之為「國語對談式電話總機輔助系統」。而在此系統中，自然語言處理和語音辨認的技術是系統成功的關鍵。

近年來，自然語言處理方面的各種理論和應用系統不斷的推出，有的應用在以自然語言為輸出入介面的資料庫查詢系統上，如GUS[1]、TEAM[2]、LUNAR[3]、LADDER[4]和CIDA[5] 等；有的則用於故事篇章的理解與問題的回答，如SAM和PAM[6,7]。這些系統，除了TEAM強調「可攜性」(transportability)外，都應用於特定的領域上。

GUS是一個以「框架驅動」(frame-driven) 的對話系統，它透過和使用者以自然語言形式的英文對話，替客人預訂機票。在此系統中，輸入文句先經過利用「轉移網路文法」(transition-network grammar)和「圖表」(chart)的語法分析器，產生語法結構，再交由「格框分析」(case-frame analysis)。GUS雖然可瞭解一些「交戶啓動」(mixed initiative)的對話，但整個對話過程仍是由系統所引導。

TEAM以「邏輯形式」(logical form)做為語意表達的方式，再經由「綱要轉換器」(schema translator)將邏輯形式轉換成資料庫系統的查詢語言。它著重於資料庫系統自然語言處理介面的可攜性，對於「省略」(ellipsis) 和「指涉」(reference) 部分則沒有處理。

LUNAR利用「增強型轉移網路」(Augmented Transition Network, ATN)剖析輸入文句，產生以「意義表達語言」(Meaning Representation

Language, MRL)描述的中間表示式。它利用句子的語法結構和語意解釋，處理指涉和省略的問題，但是未能提供一般性的解答。

LADDER使用「語意語法」(semantic grammar)。語意語法中所使用的並非語法上的範疇，而是語意類別，以避免一些沒有意義的冗贅。在LADDER系統中，處理了簡單的省略問題，也簡化的將最近提到的物體，作為指涉文句的所指。

CIDA與前述四個系統不同，是一個中文的圖書資料查詢輔助系統[8]。它先對輸入文句做斷詞，再利用「以中心語驅動的詞組結構語法」(Head-Driven Phrase Structure Grammar, HPSG)加以剖析，產生「複雜特徵結構」(complex feature structure)做為中間表示式；其次，CIDA的「語境解譯器」(contextual interpreter)利用「規則庫」(rule base)中的規則和「對話庫」(dialogue base)中的資訊，解釋特徵結構。至於指涉和省略的問題，則未完成處理。

SAM和PAM利用「概念分析器」(conceptual analyzer)產生「概念相依表示式」(CD representation)。與前述系統不同的是，SAM和PAM並未對語法做深入的分析，只是利用部分的語法訊息，輔助概念分析；至於對話的處理，SAM使用「劇本」(script)，PAM則利用「計劃」(plan)和「目標」(goal)，來瞭解故事。

ACTOA在自然語言處理部分所面臨的是十分口語化的對話輸入，這是因為使用者在打電話時，常常是邊說、邊想、邊改，甚至脫口而出，未加以思索，一但說錯，不像文字輸入可以用BACKSPACE鍵修改。在這種情形下，如何避開錯誤的訊息，找出客人真正的意圖，是ACTOA面臨的最大問題。

至於語音辨認方面，由於技術仍未完全成熟，因此我們在ACTOA系統中乃以文句取代語音，並模擬交換機的功能，先行發展「國語文句對談式電話總機輔助系統」，除了輸出入部分改用文句外，其餘部分仍舊不變，亦即，這是一個有理解、應對能力的電話轉接系統。

ACTOA的系統架構如圖1 所示，以中間表現為界，可將系統粗略的分成兩部分，其上是「理解子系統」，其下則是「應答生成子系統」；應答生成子系統又可細分為「語境解譯器」和「應對產生器」兩個子系統。客人輸入的文句經過「中文代碼轉換器」，轉成中文代碼，經過斷詞、概念分析後，產生中間表現；語境解譯器將中間表現與客人上次所說的，做概念上的整合、資料庫的查詢、對話環境的更新等；根據當時的對話環境和對話劇本，應對產生器產生適當的應答或動作。在此系統中，對話劇本透過「語境參數」(contextual parameter)，提供「概念產生器」相關的語境資料，並由語境解譯器更新這些語境參數。

本文第2節介紹ACTOA的系統功能和對話模式；第3 節介紹ACTOA的理解子系統與其中間表示式；第4節討論應答生成子系統；第5 節是測試結果的分析；最後一節則是結論與未來發展方向。

2. 系統功能與對話模式

2.1 系統功能

總機的工作依電話來源和目的地所在位置的不同，可分成：「外線打入」、「內線轉內線」、和「由內線打出到外線」等三種，其中除了由內線打出要求轉接外線的情形，由於牽涉到的知識庫和資料庫太大，且無法掌握外，其餘兩種都是ACTOA系統所欲處理的範圍。

外線打入的電話中，最常見的就是轉接要求[9]。客人提供人名、分機、地點，而要求總機轉接；有時客人甚至不知道該找誰，僅知欲辦理或詢問的事項，總機都必須根據客人所給的訊息，給予適當的轉接。除了轉接要求之外，回答客人詢問，如詢問研究所全名、地址、是否有上班，或是詢問同仁電話、電話狀態等，都是總機應具備的功能。

由內線打給總機的要求轉接電話，多半是因為同仁不知道分機號

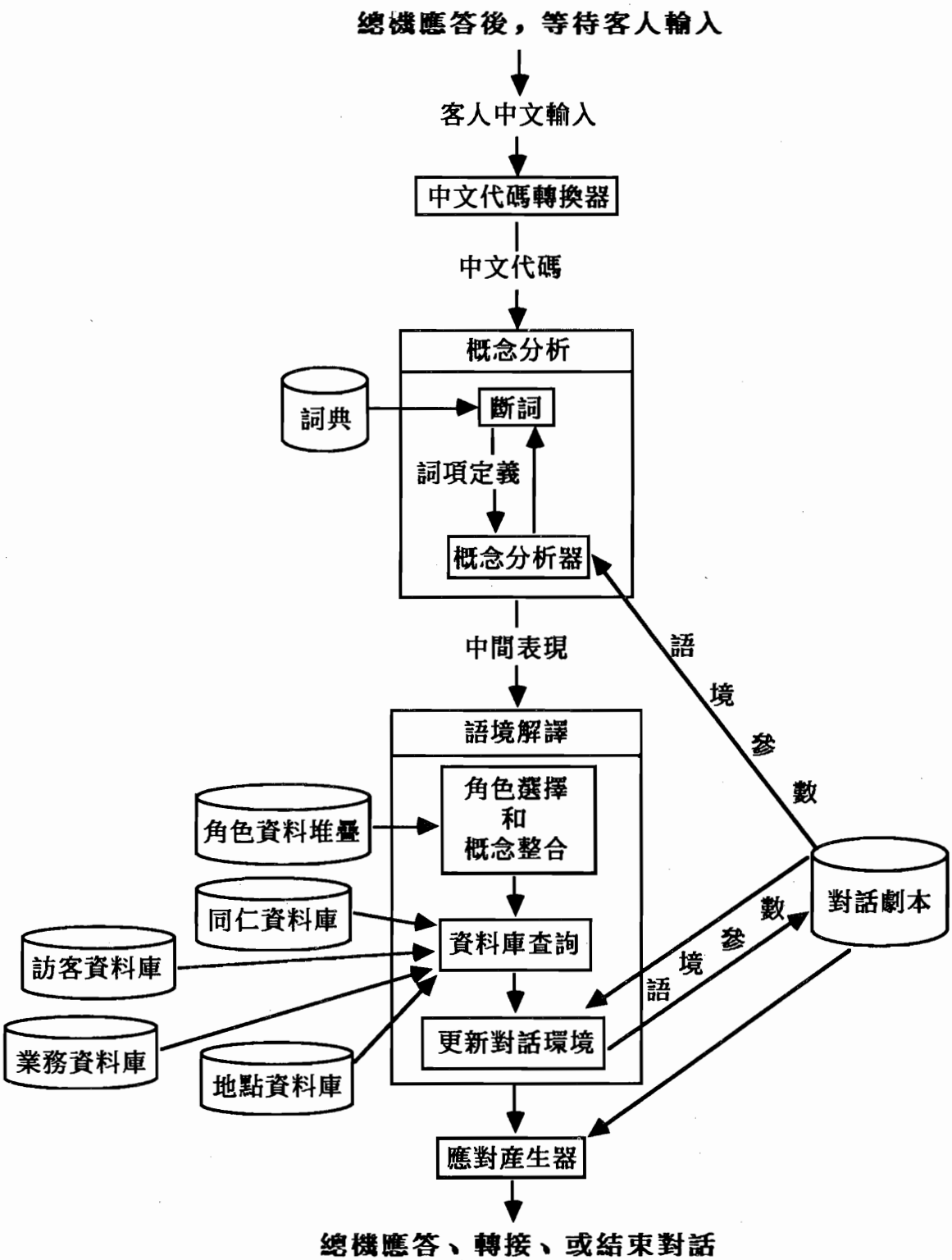


圖1. ACTOA 的系統架構

碼，僅知人名或地點，因此要求總機代轉；或是欲辦理或詢問所內事務，但不知道是誰負責的，也透過總機幫忙。除此之外，ACTOA 尚提供一項「話機跟隨」的功能。現有的話機跟隨功能必須由人按鍵設定，此種方式使所有打到該分機的電話，不論找誰，全部轉到所設定的分機上，這種方式並不適用於多人共用一個分機號碼的情形。ACTOA 所提供的話機跟隨的功能，使同仁可向總機設定自己目前所在的分機，而不妨礙同一分機的其他同仁。

根據以上的說明，我們將 ACTOA 欲提供的功能列於圖2。

系統功能	子功能
電話轉接	○ 已知人名、分機、地點的電話轉接 ○ 僅知欲辦理事項的電話轉接
回答詢問	○ 詢問本所全名、地址、上班狀態 ○ 詢問總機電話 ○ 詢問同仁電話、上班狀態 ○ 詢問地點電話 ○ 詢問電話狀態
狀態設定	○ 同仁上班狀態設定 ○ 話機跟隨

圖 2. ACTOA 的系統功能

2.2 對話模式

雖然電話轉接時，總機值機員和要求轉接的客人間的談話領域有所限制，我們仍必須就總機值機員和客人間各種不同類型的對話實例，配合系統所欲提供的功能，加以探討、分析，才能找出「對話模式」。這部分的工作已做過[10]，我們在此僅做摘要性的介紹。

所謂的「對話模式」就是指：在對話的過程中，對話的雙方為達成某種共同的目標（例如轉接電話至某分機）所採用的相互問答的方式、

總機值機員的動作等與完成此目標有關的整個過程。

不論那一類對話，都可以將整個對話，依其部分對話所要達到的目標分為三部分，首先是「對話對象識別」部分，其次是「服務對話」部分，最後則是「結尾」部分。

「對話對象識別」部分的對話主要目的在互相識別說話的雙方是否正是彼此想要說話的對象，因此除了有「問候」、「宣告或確認總機所屬單位名稱」等功能的句子外，「對話對象識別」部分也可以包括客人「表明自己的身份」的句子。經過或略過對話對象識別後，整個對話就進入了對話的核心－－－「服務對話」部分。「服務對話」部分是由客人提供資訊，總機根據客人提供的訊息，或詢問客人，以取得足夠的資訊，或做出適當的回應。客人提出服務要求之後，有些客人還會以「謝謝」…等<客人致謝語>表示謝意。「結尾」部分可有可無。它是「服務對話」之後，依據當時對話的環境，客人可能說的，用以結束對話的語句，如「再見」、「我晚一點再打好了」等。

下面就是我們蒐集的對話資料中的一個例子，Op代表總機值機員；Cus則代表客人[9]，其中「對話對象識別」的部分以「→」標示，其餘的部分屬於「服務對話」部分，「結尾」部分則無。

對話1：
→ Op：研究所。
→ Cus：ㄟ，研究所是不是？
→ Op：是。
Cus：ㄟ，麻煩你跟我接257好不好？
Op：257啊？
Ding-T〔總機值機員撥分機號碼後，分機的鈴聲。〕

ACTOA也是一個「總機服務對話理解系統」，所謂的「理解系統」必須能根據輸入文句，產生適當的應對或採取適當的行動，其前提是：能瞭解客人所輸入文句的意義。根據上述的對話模式和ACTOA的系統功能，客人所說語句對ACTOA的意義、說明、及例句，如圖3所示。

客人所說語句對 ACTOA 的意義	例句或說明
1. 無輸入	客人沒有任何訊息輸入
2. 未定	客人所說語句中未指明要求什麼
3. 不瞭解	客人所說的字系統都不懂
4. 問候語	例：你好
5. 表明身份	例：我是許仁榮
6. 狀態設定	例：我在機房
7. 轉接要求	例：請轉資訊室的機房
8. 隱含轉接要求	例：許仁榮在不在機房
9. 詢問事物	例：請問一下你們的地址
10. 要求總機重覆	例：你說什麼
11. 誌謝	例：謝謝你
12. 晚點打	例：那我晚一點再打好了
13. 打錯	例：對不起，我打錯了
14. 再見	例：再見

圖 3. 客人所說語句對 ACTOA 的意義、說明、或例句

圖 3 中，前三項在對話的任何部分都可能發生；4-5 項屬於「對話對象識別」；6-10 項屬於「服務對話」範圍；其餘的則是「結尾語」。

3. 理解子系統

理解子系統由「中文代碼轉換器」、「斷詞模組」、和「概念分析器」構成。由於本系統所使用的程式語言不能直接處理中文，因此所有的中文輸入，都必須先經過中文代碼轉換器，轉換成系統可處理的符號，再經斷詞與概念分析，產生句子的中間表現。

如同前面所述，ACTOA 所處理的是十分口語化的輸入文句，包括許多的口頭語、贅語、省略、說錯的、或是因雜訊而導致不完整的句子。欲以有限的文法規則去規範所有可能的輸入文句，是一件很難的

工作，同時，對話系統對於所輸入的文句，不論在何種情形下，都應有所反應，因此，儘管句子不完整，系統仍應儘可能的從片段中取得資訊，以產生適當的回應，一般的語法分析顯然不能滿足此項需求，所以本系統乃捨棄一般的語法剖析，偏重於句子概念的抽取。

3.1 詞典

ACTOA的詞典格式與 MICRO ELI 的字典格式類似[7]，都有 TEST、ASSIGN、NEXT-PACKET等結構，不同的是，ACTOA系統存的是詞，且加入了ADD-PROP結構，對詞的定義也採用不同的表現方式。ACTOA的詞項結構如圖4 所示。

```
(def-lex   W0
   ((W11     (( ...   ((W1n  (($   (  (TEST ...)
                                      (ASSIGN ...)
                                      (ADD-PROP ...)
                                      (NEXT-PACKET ...))))) ... )))
    (W21      (( ... ((W2m  (($   ...)))))... )))))
```

圖 4. ACTOA 的詞項結構

在圖4 中，W_0、W_{11}、W_{12}、…W_{1n} 代表一個由(N+1)個字構成的詞，W_0、W_{21}、…W_{2m} 則是一個由(M+1)個字所構成的詞，都以W_0為詞的第一個字，而以"$"表示詞的結尾。由於每一個詞可能具有多個意義，因此有許多的測試條件，由TEST 結構表示；一旦條件成立，附於其下的ASSIGN和ADD-PROP，就設定一些語意標記，供其他詞參考，或加入一些特徵到句子的概念中；有時候，一個詞的某些意義必須在第一次處理該詞時就決定，而其他意義則不能完全由出現在它之前的訊息所決定，必須看它後面詞句的內容，NEXT-PACKET就提供了這種功能，其下也是由 TEST、ASSIGN 、ADD-PROP、NEXT-PACKET 等構成，是一個遞迴

的結構。

ACTOA的詞典與MICRO ELI的字典另一個重大的差異是，ACTOA並未採用「概念相依」的語意訊息，而著重於詞在句子中的「詞序」和「與其他詞的語意關係」。以「在不在」(*0A6*62 *0A4*0A3 *0A6*62) 和「在」(*0A6*62)為例，它們的詞項定義如圖5(a) 所示。「在不在」不論出現在句子的何處，通常用於詢問某人狀態，這在ACTOA中歸類為「隱含轉接要求」；而「在」後面常跟隨著地點，但是確實的意義必須視後面接的詞而定，它可能是「在講話啊」（詢問電話狀態）、「在開會啊」（詢問同仁狀態）、「他今天不在」（設定上班狀態）、「在機房」（話機跟隨）、或是「在機房嗎」（隱含轉接要求）等。

由於ACTOA要處理大量的句子片段，尤其在續問句時，客人多使用不完整的句子，因此詞的意義有時必須依靠語境訊息來決定。以「幾號」(*0B4*58 *0B8*0B9) 為例，客人可能以它詢問本所地址、同仁電話、地點電話等，除了由句子內其他詞項決定其詞意外，在沒有其他訊息之下，語境訊息就成了決定詞意唯一的憑藉。圖5(b)是「幾號」的詞項定義，其中*query_type* 就是語境參數。

3.2 斷詞

詞是一個具有完整語意的單位。在英文裡，單字就可成詞，有自己的意義；但是在中文裡，有許多單字並不成詞，尤其在句子中，那些字該構成一個詞對句子的意義有很大的影響，因此對中文句子做分析之前，必須先對句子做切割，將句子分成一些適當的語意單位。除了字典的查詢之外，斷詞常用的方法主要有「結構性的方式」(structural approach) 和「統計性的方式」(statistical approach) 兩種[11]。

結構性的方式通常採用一些「經驗法則」(heuristic rules)做為選詞的標準，常見的經驗法則有：「長詞優於短詞」、「與左邊詞的結合優於與右邊詞的結合」等。以「他馬上來」為例，因為「馬上」（兩個

```
(a). (def-lex  *0A6*62
       ((*0A4*0A3
         ((*0A6*62
            (($  ( (assign *job_type*  'intent_to_switch
                           *cur_part*  'location)))))))
        ($ (assign *cur_part* 'location)
           (next_packet
             ( (test (equal *stus* 'speaking))
                 (assign *stus* nil))
             ( (test (equal *stus* 'meeting))
                 (assign *stus* nil))
             ( (test (and   (equal *word* '*end*)
                            (equal *job_type* 'question)))
                 (assign  *job_type*  'intent_to_switch))
             ( (test (and   (equal *word* '*end*)
                            (equal *mode* 'negative)))
                 (assign  *job_type*  'status_setting)
                 (add-prop  *cur_role*  ((person status (-)))))
             ( (test (equal *word* '*end*))
                 (assign  *job_type*  'status_setting))))))

(b). (def-lex  *0B4*58
       ((*0B8*0B9
         (($ ( (test (equal *query_type* 'addr))
                 (assign *job_type* 'question)
                 (add-prop *lab* ((query lab addr (ADDR_NO NIL)))))
             ( (test (and (equal *cur_part* 'person)))
                 (assign *job_type* 'question)
                 (add-prop   *cur_role* ((query person tel))))
             ( (test (equal *cur_part* 'location))
                 (assign *job_type* 'question)
                 (add-prop *cur_role* ((query location tel))))
             ( (test (equal *cur_part* 'lab))
                 (assign *job_type* 'question)
                 (add-prop *lab* ((query lab tel))))
             ( (test T)
                 (assign *job_type* 'question)))))))
```

圖5. (a)「在不在」、「在」的詞項定義　(b).「幾號」的詞項定義

字）較「馬」（一個字）長，而「上」雖然可與左邊的「馬」或右邊的「來」結合，但以「與左邊結合優先」，所以「他馬上來」可斷為｛他、馬上、來｝。這些經驗法則適用於大多數的例子，但是仍然會有問題。以「這名記者會說國語」為例，依照上述的經驗法則，斷詞的結果是｛這、名、記者會、說、國語｝，顯然與句子本意不符。

統計性的方式利用詞的統計資料，透過數學模式的分析，來決定最有可能的詞組組合，這種方示式產生的結果不錯[12]，但是由於以詞的出現頻率斷詞，和結構性的方式一樣，缺乏理論基礎，且無法說明句子在何種情況下會斷錯，造成何種錯誤，因此有的系統就加入了語言上的知識為輔助，以提高斷詞的正確性。

葉和李提出以規則為主(rule-based)的斷詞方式[11]，他們除了採用上述的經驗法則和詞類的出現頻率來決定詞界外，對於一些有歧義的句子，他們也配合語法分析和語意分析來解決，也就是說：一個完整的斷詞系統必須以語法、語意、甚至語用分析為後盾。

ACTOA 的詞項僅有數百到數千個（視資料庫的大小而定），且並不複雜，所以斷詞模組採用結構性方式中的「長詞優於短詞」、「與左邊詞的結合優於與右邊詞的結合」等常見的經驗法則，結果十分良好。

3.3 以詞典和語境為本的概念分析器

ACTOA 的概念分析器利用詞典和斷詞模組，取得詞項定義加以分析，產生中間表現。在對話的過程中，客人的輸入文句可能含有多個概念，以「請轉 908 ，謝謝」為例，句子中包含了「轉接要求」和「致謝語」兩個概念，在這種情形下，如同一般人所採取的方式，概念分析器只取出一個最重要的概念，而這種重要概念的選擇完全取決於詞項的定義；如同上述，概念分析時，也利用了語境參數，所以ACTOA的概念分析乃是以詞典和語境為本。概念分析器的流程、與斷詞、詞典的關係，如圖6 所示。

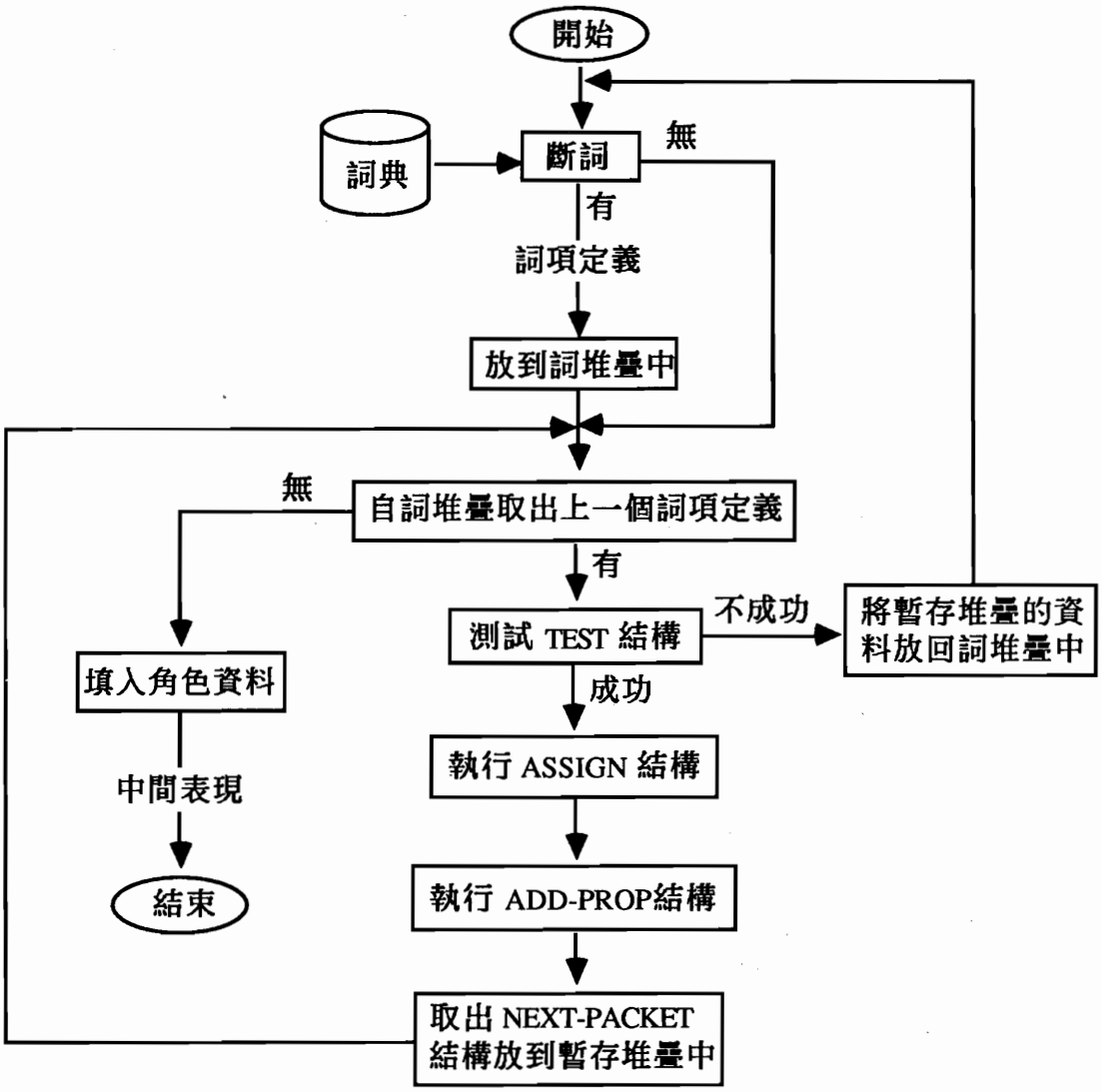


圖 6. 概念分析器的流程、與斷詞、詞典的關係

3.4 中間表現

ACTOA 的中間表現由句子的主要概念、客人提及的人、事、地等角色、及與此角色相關的一些特徵結構組成。句子可能的主要概念列於圖 3；而客人提及的角色乃是指客人自己，或是所提到的欲轉接的目的地，或欲詢問的對象，因此，這些角色包括客人本身、所內同仁、地點、和總機所代表的單位（本所）。客人在提到各角色時，都會提供一

些與該角色相關的資訊，如人名、職稱等，這些都屬於特徵結構。以「請轉資訊許仁榮」為例，其中間表現如下：

```
(switch (col1 ((person   (lastname (0B3 5C))
                         (firstname (0A4 0AF 0BA 61))
                         (department (0B8 0EA 0B0 54))))))
```

其中，主要概念是switch，客人提及的角色是col1，而lastname 、firstname 和department 都是特徵結構。

ACTOA的部分特徵類別、特徵結構、及其意義，如圖7 所示。

特徵類別	意義	特徵結構
個人資料	○許 ○仁榮 ○資訊	((person (lastname (0B3 5C)))) ((person (firstname (0A4 0AF 0BA 61)))) ((person (department(0B8 0EA 0B0 54))))
個人狀態	○沒上班	((person (status (-))))
地點資料	○影印室 ○資訊 ○4301室	((loc (name (0BC 76 0A4 4C 0AB 0C7)))) ((loc (department (0B8 0EA 0B0 54)))) ((loc (room_no 10CD)))
詢問同仁電話	○某人電話幾號	((query (person (tel))))
詢問地點電話	○某地電話幾號	((query (loc (tel))))
詢問同仁狀態	○有沒有上班	((query (person (status))))
詢問本所狀態	○有沒有上班	((query (lab (status))))
詢問電話狀態	○沒人接嗎	((query (tel (status (noans)))))
詢問所內事物	○勞保	((query (lab (affair (affair_class7)))))

圖7. ACTOA 的部分特徵類別、特徵結構、及其意義

4. 應答生成子系統

4.1 語境解譯器

語境解譯器解釋中間表現在對話環境中的意義，由「角色選擇與概念整合」、「資料庫查詢」、「對話環境更新」等三個模組所構成。

4.1.1 角色選擇與概念整合

角色的選擇主要在解決指涉的問題。雖然絕大部分的總機服務對話，客人只提到一個角色，但是在較複雜的對話中，客人所提及的角色就可能有很多個，因此，如何找出客人真正所指的角色是一個重要的課題，我們將以一些例子，來說明本系統對角色選擇的處理方式。

對話2：

Op: 研究所
Cus: 請問一下那個唐憲章的電話幾號
Op: 他的電話是908
Cus: 那那個梁冠雄呢
Op: 90X
Cus: 你說多少 （客人沒聽清楚）
Op: 907
Cus: 907 啊
Op: 是
Cus: 他在不在
…

對話3：

Op: 研究所
Cus: 請轉梁冠雄
…
Op: 講話中
Cus: 那唐憲章他有沒有來
…

對話2和對話3中，客人都提到唐憲章和梁冠雄，也都用「他」來指稱，但對話2的「他」指的是梁冠雄，而對話3的「他」則是唐憲章。本系統對此問題的解決方式是：一個句子中有指稱詞，又提及一個角色，該角色即視為是指稱詞所指的，否則，系統就到角色堆疊中找出最近提到，符合所指類別（人或地點）的資料，當做指稱詞所指的對象。

在對話4 中，我們假設有兩位王大明，分別屬於資訊和交換兩個部門。當客人第一次提及時，總機要求客人提供部門訊息；當後面再次提及時，總機就不再詢問，而以「資訊的王大明」做為客人所指的對象。

對話5表達的是：總機第一次取得錯誤且不完整的訊息，產生了一個新的不完整的角色，經過再次詢問後，總機取得新的訊息，舊的、不完整的角色資料就被放棄。

選擇正確的角色之後，系統必須將輸入文句的概念，和該角色已有的訊息做整合。以對話4 為例，客人所說第二句中的「資訊」和客人所說第一句中的「王大明」整合後，就變成「資訊王大明」。

對話4 ：

Op: 研究所
Cus: 請問一下那個王大明的電話幾號
Op: 那一個部門的
Cus: 資訊的
Op: 他的電話是900
Cus: 李大海的呢
Op: 907
Cus: 那幫我轉一下王大明 謝謝
…

對話5 ：

Op: 研究所
Cus: 許XXXX
Op: 找那位
Cus: 李臻儀
…

4.1.2 資料庫查詢

資料庫查詢模組主要的工作是查出概念中所提到的人、事、地等資料。若整合後的概念中所含訊息不足，或有錯誤，資料庫查詢模組就設定必須的標記，將資料不足的訊息傳出。

本系統共有同仁、訪客、地點、和業務等四個資料庫，除了訪客資料可機動更新外，其餘資料庫的修改或刪增，都必須在總機執行服務之前完成。

4.1.3 對話環境和省略

在對話的過程中，依據對話的內容，會建立起對話環境，這種對話環境，使得說話的一方可以使用一些省略或不完整的句子，卻無礙聽者的理解，因此對話環境和省略的關係十分密切。在ACTOA中，對話環境是由一些語境參數和對話劇本所構成，而對話環境的更換實際上就是語境參數和對話劇本的更換。

依據客人所提要求的不同，系統所需的語境參數和對話劇本也就不同。例如，在詢問所內事務、同仁電話時，系統必須了解客人所欲詢問的相關對象（如本所、同仁、或地點）和欲詢問的項目，此時所需的是一個詢問方面的對話劇本；在轉接要求時，客人可以以人名、分機、或地點要求轉接，不同的方式，總機應對的方式也不同，這也是語境參數必須提供的。對話劇本將在下一節討論，本節則著重於本系統所能處理的省略情況之分析。

依據語境參數和對話劇本，系統可處理一些電話轉接對話中的省略問題：

服務類別的省略

如對話6，客人只給分機號碼，總機仍了解客人要求「轉」907。

轉接對象的省略

如對話7，客人詢問電話號碼後直接請總機轉接。要求轉接的句子中，客人並未指明欲轉接的對象，但總機由對話環境可知是胡祖櫻。

詢問對象的省略

如對話8，客人詢問本所全名，而後詢問地址，總機知道「地址」指的是「本所地址」。

詢問事項的省略

如對話2，客人只說「那那個梁冠雄呢」，總機回答客人梁冠雄的電話。

狀態的省略

如對話9，客人首先設定自己的狀態，再設定另一位同仁的狀態。

對話6：

Op: 研究所
Cus: 請轉唐憲章
…
Op: 喂 電話在講話中
Cus: 那 907
…

對話7：

Op: 研究所
Cus: 請問一下胡祖櫻的電話幾號
Op: 他的電話是908
Cus: 可不可以幫我轉一下
…

對話8：

Op: 研究所
Cus: 研究所嗎
Op: 是
Cus: 你們的全名是什麼
…
Cus: 那地址呢
…

對話9：

Op: 研究所
Cus: 你好 我是資訊許仁榮
Op: 是
Cus: 我現在在703
Op: 好的
Cus: 唐憲章也是
…

4.2 對話劇本與應對產生器

對話劇本與劇本不同，劇本主要描述一連串有順序關係的事件，並用於故事篇章的瞭解；而對話劇本主要是由語料分析，配合系統所欲提供的功能所產生的，除了透過語境參數，幫助概念分析，建立對話環境，並解決省略問題之外，對話劇本主要在提供系統決定應答所需的訊息。由於ACTOA 並未完全具有人一般的能力，同時為了避免一些無

謂的冗贅對話，因此對於某些狀況，ACTOA並未完全模擬語料中總機值機員的應對方式，而採用更簡潔的策略。

如圖8，用於回答詢問本所地址的對話劇本，所示，對話劇本由「劇本名」(qlab_addr_script)、Entry_Cond、Init_Action 、Rule_Packet 和 Exit_Cond 五個部份所組成。

```
(def-script  qlab_addr_script
    (Init_Action
      (assign *lab_pointer* 0 *lab_flag* T *expected_part* 'lab))
    (Entry_Cond
      (and (member *job_type* '(question what unclear confirm null nil))
    (Rule_Packet
      ((cond (member *job_type* '(null confirm)))
       (action (oprepeat 'T '(*lab_pointer* (COUNTY TAU_YUAN)
                              (TOWN YANG_MEI) (ROAD MING_CHU)
                              (SEC THREE) (LANE 551) (ADDR_NO 12)) 'SO)))
      ((cond (or (null *inlab_addr*)(member *job_type* '(what unclear))))
       (action (oputter *lab_flag* '((0 OUR_ADDR BLANK)))
               (oprepeat 'T '(*lab_pointer* (COUNTY TAU_YUAN)
                              (TOWN YANG_MEI) (ROAD MING_CHU)
                              (SEC THREE) (LANE 551) (ADDR_NO 12)) 'SO)
               (assign *lab_flag* nil)))
      ((cond (equal *inlab_addr* *lab_addr*))
       (action (oputter 'T '((0 YES)))
               (assign *inlab_addr* nil *lab_pointer* 6)))
      ((cond T)
       (action (oputter *lab_flag* '((0 NO COMMA OUR_ADDR BLANK)))
               (oprepeat 'T '(*lab_pointer* (COUNTY TAU_YUAN)
                              (TOWN YANG_MEI) (ROAD MING_CHU)
                              (SEC THREE) (LANE 551) (ADDR_NO 12)) 'SO)
               (assign *inlab_addr* nil *lab_flag* nil))))
    (Exit_Cond *ending_mark*))
```

圖8. 用於回答詢問本所地址的對話劇本

Entry_Cond 是對話劇本的入口，用以檢查目前的對話劇本是否可以解釋新輸入的概念；Init_Action是起始動作，在啓動該對話劇本時執行；Rule_Packet 是對話劇本的重心，由一組規則所組成，每一條規則包括「條件」和「動作」，描述各種情況下系統應採取的應對措施，如回答、轉接、設定離開標記、或到，對當時的對話環境和對話輸入系統所應採取的措施，有更詳細描述的子對話劇本中，去執行；Exit_Cond 在執行 Rule_Packet 中的任一條規則後，用來測試是否離開此對話劇本，或讀入下一個輸入文句。

應對產生器利用對話劇本產生適當應對，它和對話劇本的關係如圖 9 所示。應對產生器首先測試對話劇本中的Entry_Cond，若不成功，就離開此對話劇本，回到叫用它的劇本上；若是測試成功，應對產生器就執行Init_Action 和Rule_Packet，執行的結果可能會叫用子對話劇本；最後，測試Exit_Cond，以決定是回到叫用它的劇本上，或是回到理解子系統。

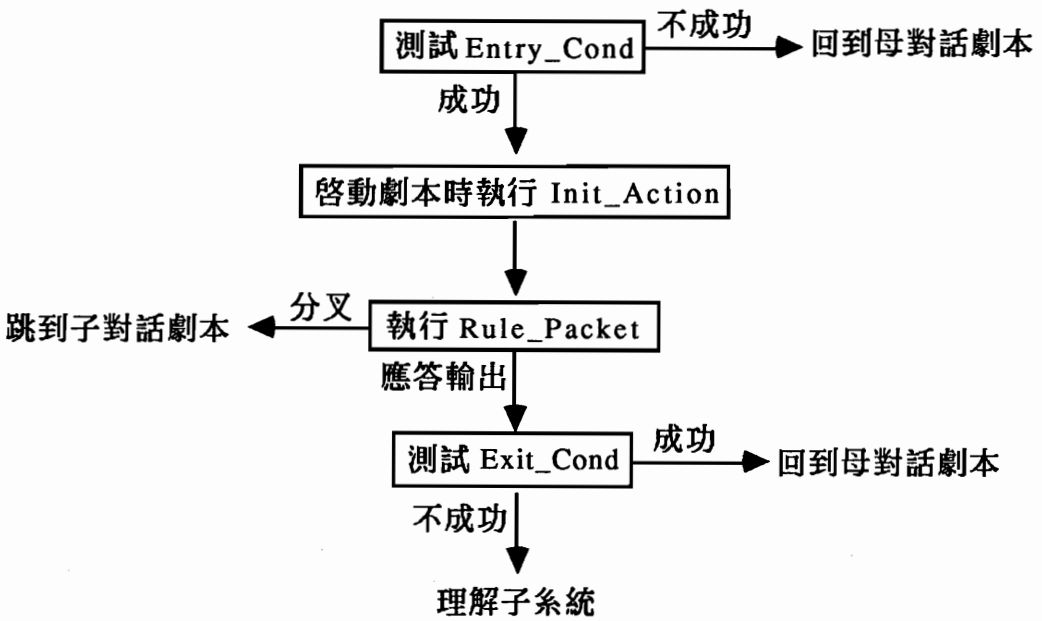


圖 9. 應對產生器和對話劇本的關係

5. 系統測試結果分析

在一個斷詞系統中，有的人以詞，有的人則以句子為衡量正確率的單位；在逐句翻譯的翻譯系統裡，所翻出句子的正確性是最基本的考量；而像ACTOA這樣的對話系統，句子的正確率並非最重要的，「對話目標」的達成率才是衡量一個對話系統的主要依據。

ACTOA是一個總機與客人交互應答的對話系統，每一次的對話中，都有一個或多個主目標，有時還包括多個次目標。在應對的過程中，客人可能以多個句子逐步達成傳達主要訊息的目的，如對話10所示，客人分兩次才將他要「詢問李臻儀電話」的訊息完整的表達出來，在這種情況下，客人所說的每一個句子可能是達成主目標所不可或缺的，因此若是系統誤解一個句子，又不能提供訊息讓客人適時更正，可能造成整個對話的錯誤；有時，客人所說句子所傳達的訊息與主要目標的達成並無直接關聯，即使系統誤解也沒影響；或者，雖然有影響，可是系統提供的訊息可讓客人適時更正，在這種情形下，誤解這樣的句子對主要目標的達成並無影響。

根據以上的分析，我們將對話系統可能產生的錯誤分成兩個層次，一個是句子層次，另一個則是對話層次。句子層次包括由客人輸入文句，到系統產生應對文句或執行轉接動作之間的範圍；對話層次則是包括整個對話目標的達成。圖10就是對話中句子層次和對話層次可能的正誤關係。

句子層次	對話層次
O	O
O	X
X	O
X	X

圖10. 對話中句子層次和對話層次可能的正誤關係

對話10 ：

Op: 研究所
Col: 總機，請問李臻儀的X嘀咕X
Op: 你有什麼事?
Col: 他的電話多少?
Op: 他的電話是908
Col: 謝謝・ㄏㄟ。
Op: 不客氣

對話11 ：

Op: 研究所
Cus: 總機ㄟ那個胡X嘀咕X的電話幾號你知不知道
Op: 誰的電話?
Cus: 胡祖櫻
Op: 他的電話是908
Cus: 908 啊 謝謝
Op: 是
Cus:
Op: 就這樣
Cus:
Op:

以句子層次錯誤，對話層次正確的情形為例。有時在對話的過程中，系統因不瞭解而誤解客人所說的話，以致產生句子層次上的錯誤，卻由於系統的應對子系統在有些情況下，能產生具有主動意味的應對，引導客人提供更多的訊息，以利對話的進行，達成對話的目標；另一種情形是，客人所說句子所傳達的訊息與主要目標的達成並無直接關聯，即使系統誤解也沒影響，對話11就是一個例子，其中對話中的粗體字即是系統理解錯誤的句子和系統所產生的應對。

在對話11中，「908啊謝謝」包括了客人「確認分機號碼」和「感謝語」兩種概念，而系統選擇了前者。直到多次應答都得不到客人回應後，系統才停止產生應對。此時客人「詢問同仁分機號碼」的對話目標早已達成。

我們將ACTOA的263個測試對話中，句子層次和對話層次的正誤與其所佔個數和比例的關係，列於圖11。由圖11可知，對於這263個測試對話，ACTOA的對話目標達成率是97.72％。

句子層次	對話層次	對話個數	百分比
O	O	252	95.82%
O	X	0	0
X	O	5	1.90%
X	X	6	2.28%
總　數		263	100.00%

圖11. 對話中句子層次和對話層次的正誤與其所佔個數和比例的關係

6. 結論與未來發展方向

本文提出一個以文字輸出入的自動化中文電話總機輔助系統的架構和它的製作方式，包括：總機和客人間各種不同類型的對話模式；著重於詞在句子中的「詞序」和「與其他詞的語意關係」的詞典結構、以詞典和語境為本的「概念分析器」、句子的中間表現、對話概念的整合、「語境解譯器」，和對話劇本等，同時也討論了本系統所遭遇的指涉和省略的問題，並提出了解決的方法。

儘管本系統已能處理總機所處理的大部分工作，有一些問題仍然有待解決。首先是詞典的維護。ACTOA 的詞典著重於詞在句子中的「詞序」和「與其他詞的語意關係」，這使得詞的新增和修改都十分麻煩，尤其當新增資料庫的人名或地點的詞彙，與轉接系統中的關鍵詞相同時，該詞的加入就必須修改與該詞有關的詞組定義，因此一個輔助性的詞典輸入介面有其必要性。其次，雖然概念分析器對絕大部分的輸入文句，都能產生適當的中間表現，但是，不用語法訊息使得它對複雜片語結構的處理能力受限。

本系統在發展時，雖然儘量模仿以語音輸入時的電話轉接對話，但是與實際情形仍有差距，因此和語音辨認、語音合成、及交換機系統密

切配合，以完成一個線上系統，做為發展自動化104查號輔助系統的基礎，是我們未來的發展方向。

誌謝

本文係交通部電信總局電信研究所80年度計畫(計畫代號：80312)中自然語言處理部分所得成果之一。感謝本所呂學錦所長、王金土副所長、電腦通訊與人工智慧計畫主持人盧清松博士、人工智慧分項計畫 主持人蕭振木博士對有助於本文完成的各項先導工作的支持， 特別感謝同仁胡祖櫻完成本系統的測試工作。此外，與其他同仁的討論對完成本文也有很大的貢獻，特此申謝。

參考資料

[1] D. G. Bobrow, R. M. Kaplan, M. Kay, D. A. Norman, H. Thompson, and T. Winograd, "GUS, a frame-driven dialog system," *Artificial Intelligence*, vol. 8, pp. 155-173, 1977.

[2] B. J. Grosz, " TEAM : an experiment in the design of transportable natural-language interfaces," *Artificial Intelligence*, vol. 32, pp.173-243, 1987.

[3] W. A. Woods, "Semantics and quantification in natural language question answering," *Advances in Computers*, vol. 17, pp.1-87, 1978.

[4] G. G. Hendrix, "Developing a natural language interface to complex data," *ACM Transactions on Database Systems*, vol.3,pp.105-147, 1978.

[5] S. H. Lee and H. J. Lee, "A Unification-Based Approach for Chinese Inquiry Sentences Processing," *Proc. of ROCLING III*, pp. 441-466, 1990.

[6] R. C. Schank and R. Abelson, *Scripts Plans Goals and Understanding*, Hillsdale, New Jersey, 1977.

[7] R. C. Schank and C.K. Riesbeck ed., *Inside Computer Understanding*, Hillsdale, New Jersey, 1981.

[8] 李錫堅，「中文智慧型資料庫輔助系統之研究」，電子與資訊研究中心技術報告，TR-MIST-E76006，1987。

[9] 胡海旭編輯，李臻儀、胡祖櫻、胡海旭、唐憲章、許仁榮語料謄寫及校對，1990b，電話總機值機員和客人對話分析研究報告，第二卷：「對話原始語料」第一冊至第五冊，中華民國交通部電信總局電信研究所。

[10] 胡海旭、李臻儀、胡祖櫻、許仁榮，「電話轉接的部分句型與對話模式的分析」，*Proc. of ROCLING III*, pp.271-294, 1990.

[11] C. L. Yeh and H. J. Lee, "Rule-based Word Identification for Mandarin Chinese Sentences," *Proc. 1990 Intern. Conf. on Computer Processing of Chinese and Oriental Languages*, pp. 27-32, 1990.

[12] C. K. Fan and W. H. Tsai, "Automatic Word Identification in Chinese Sentences by the Relaxation Technique, " *Proc. of National Computer Symposium*, pp.423-431, National Taiwan University, Taipei, Taiwan, 1987.

Lexicon-Driven Transfer In English-Chinese Machine Translation

Chung-Teng Sun and Jyun-Sheng Chang

Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan 30043

Abstract

This paper identifies the major differences between English and Chinese due to lexical idiosyncrasy and describes a proposed mechanism for bridging the differences in the generation phase of an English-Chinese machine translator. The method involves uniform intermediate representation for clauses and noun phrases, a minimal set of transfer operations and an active bilingual lexicon that encodes the needed transfer. Using the method, we are able to deal with lexical idiosyncrasy in both languages in a modular and efficient manner.

1. Introduction

Much effort has been devoted to research and development of machine translation since 1950s (Slocum 1985). However, the quality of the output produced by most machine translation systems is not high enough to have any marked effect on translation productivity.

A machine translation system produces a variety of expressions in the target language, including *good*, *fair*, and *poor* expressions [Tsutsumi, 1990]. To improve the quality of translation and get the *good* sentences which can be easily understood or postedited, the following major functions should be implemented with great care.

1. Selection of equivalents for words;
2. Reordering of words; and
3. Improvement of sentences styles.

In this paper, we will propose a practical way of designing a generator in machine translation system that bridges the differences between English and Chinese languages. Using this generator we hope to achieve the above functions and improve translation quality.

1.1 Rule-Based Machine Translation

Modern rule-based machine translation systems use either *transfer* approach or *interlingual* approach.

Transfer approach is characteristic of a system(e.g., TAUM) in which the internal representations of a grammatical unit (e.g., sentence) in analysis and synthesis are different depending on the source and target languages. This implies the existence of a third translation stage which maps one language-specific representation into another : this stage is called *Transfer*. Thus, the overall transfer translation process is *Analysis* followed by *Transfer* and then *Synthesis*. **Interlingual approach** is characteristic of a system (e.g., CETA) in which the internal representation of the source language input is intended to be independent of any language, and the same representation is used to analysis the source language and to synthesize the target language output [Slocum 1985].

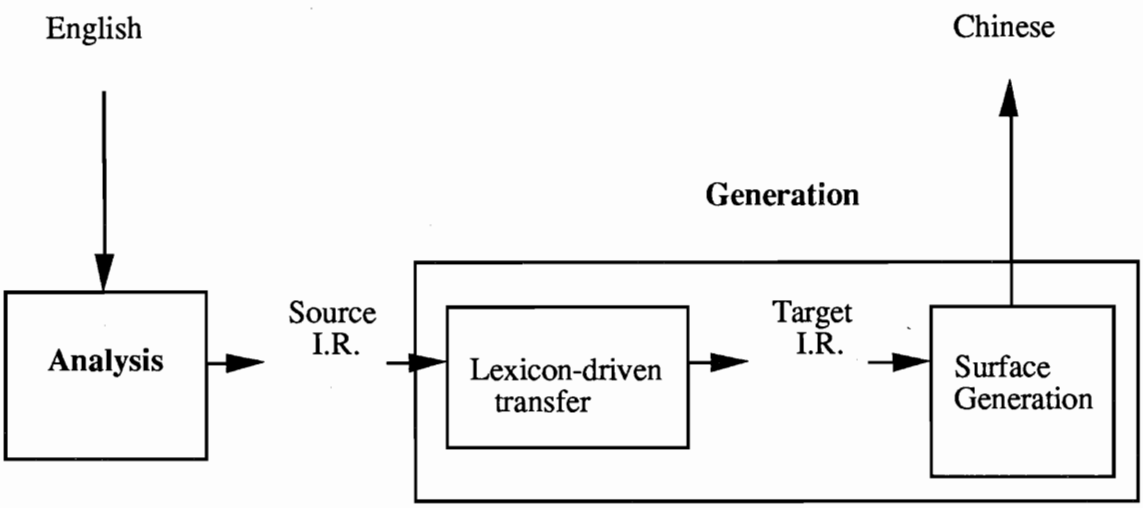
The differences between two languages can be classified into two kinds
(1) syntactical differences: the general differences in word order.
(2) differences that are caused by the idiosyncrasy of individual words in the two languages.

As for the syntactical difference, various systems use different approaches to deal with them. The transfer approach uses structure transfer rules to express the differences. The interlingual (or pivot) approach use a non-syntactical representation, and provide mapping mechanisms between syntax and the representation, so the differences can be resolved via a language independent representation.

We are currently developing a machine translation system that uses a mixed approach. On the syntactical and semantic levels, it is interlingual and on the lexical level it takes the transfer approach.

1.2 System Model

In this paper, we concentrate on the generation process of our MT system. The generator consists of two phases: the **lexicon-driven transfer phase** and the **surface generation phase**. The first phase deals specially with global reorganization of intermediate representations in order to bridge the differences caused by lexical idiosyncrasy between source and target languages. The reorganization includes structural and lexical transfer. Currently, transfer is done sentence by sentence, using only information from sentential analysis. No analysis and reorganization on the discourse level is performed. The system overview is shown in Figure 1.



I.R. : Intermediate Representation

Figure 1. system outline

This paper focuses on the problem lexical idiosyncrasy. Our approach emphasizes the role of lexicon, and in particular, it shares the idea with [Tsujii, 1990] which proposes that the bilingual lexicon play the central role in the transfer phase.

1.3 Paper Outline

This paper describes research from continuing previous works on Chinese sentence generation [Kuo 1989, Chen 1990 and Liao 1990] and focuses on the following:

1. Adding lexical transfer phase before sentence generation.
2. Designing a bilingual lexicon which controls the transfer by means of necessary tests and actions.

3. Rewriting the driver in Prolog, making various changes in the input format and extending the Chinese systemic grammar

In the following, we will concentrate on the first two areas. The rest of the paper is organized as follows: Section 2 identifies the most prominent lexical idiosyncrasy between English and Chinese. Section 3 describes our proposal for resolving these differences in order to obtain fluent target Chinese text. Section 4 compares our approach with previous works and summarizes the paper.

2. Differences between English and Chinese

In discussions on the differences caused by lexical idiosyncrasy between English and Chinese, it is necessary to consider the differences in the ways people recognize things and express their ideas about them. It is observed that there is a difference in viewpoint [Wu, 1990] between English and Chinese. So we sometimes have to restructuring them to give an appropriate translation. Each language also has its own specific word constructions, which are used to express specific meanings. These specific constructions can not be directly translated into other languages. We will describes the differences caused by *lexical idiosyncrasy* and specific constructions caused by target words in following two subsections.

2.1 Structure Transfer Caused by Lexical Idiosyncrasy

Most contrastive linguistic analysis of English and Chinese are quick in pointing out that the most prominent difference is in the way that nouns and verbs are used. In English, every sentence has at most one finite verb, so for complex information to come across, most of the information has to be expressed in terms of nouns. This explains the abundance of English nouns. There are more nouns than verbs and a verb can turn into a nominal counterpart through inflexion transformation. On the contrary, Chinese sentences are common to have more than one verb, and for this reason, verbs abounds in Chinese [Chen, 1988]. Consequently, the most appropriate translation of an English noun often turns out to be a verb in Chinese. Some English verbs turn into other parts of speech other than a noun, such as adjective or adverbs in English. However, because of heavy reliance on verbs in Chinese, the most suitable translation is again a verbal counterpart in Chinese.

The following are some examples of structural transfer due to translating an English noun into a Chinese verb. In each example, the first sentence (a) is the source English, the second one (b) is

the direct translated Chinese sentence without any structure transfer, and the last one (c) is the Chinese sentence generated with structural transfer.

(1-a): He pretended illness yesterday.
(1-b):昨天他假裝病。
(1-c): 昨天他假裝生病。

(2-a): He is a good speaker of English.
(2-b): 他是個好的英語講者。
(2-c): 他講英語講得很好。

(3-a): New Lab animals reduce testing of drugs on humans.
(3-b): 新的實驗動物減少在人身上藥物的試驗。
(3-c): 新的實驗動物減少在人身上試驗藥物。

(4-a): I have a severe headache.
(4-b) 我有厲害的頭痛。
(4-c):我頭痛得厲害。

(5-a): The arrival of a train at the station.
(5-b): 火車的抵達車站
(5-c): 火車抵達車站

(6-a): He quits the job surprisingly.
(6-b): 他令人吃驚地辭掉工作。
(6-c): 他辭掉工作令人吃驚。

(7-a): He is talkative.
(7-b): 他是愛說話的。
(7-c): 他很愛說話。

(8-a): This sentence is untranslatable.
(8-b): 這個句子是無法翻譯的。
(8-c): 這個句子無法翻譯。

English prepositions sometimes are best translated into verbs in Chinese. For instance,

(9-a): in uniform
(9-b): _
(9-c): 穿制服

(10-a): in hat
(10-b):_
(10-c): 戴帽子

(11-a): by train
(11-b):_
(11-c): 坐火車

(12-a): a path by the river
(12-b): _
(12-c): 沿河道路

(13-a): a telegram with bad news
(13-b):_
(13-c): 帶著壞消息的電報

(14-a): a man with glasses
(14-b): _
(14-c): 戴眼鏡的人

Because the different way that verbs are used in the two languages, even when a verb is translated into a verb in the target language, there could be incompatibility in their argument structure and that calls for some kinds of structural transfer too. For instance,

(15-a): we shall give special consideration to your opinion.
(15-b): 我們將給你的意見特別考慮。
(15-c): 我們將特別考慮你的意見。

(16-a): this surprised everybody°.
(16-b): 這震驚大家。
(16-c): 這使大家震驚。

(17-a): he skied twice this year.
(17-b): 他今年滑雪兩次。
(17-c): 他今年滑兩次雪。

(18-a): they lived a happy life.
(18-b): 他們過著快樂的生活。
(18-c): 他們生活得很快樂。

We have described some types of structure differences caused by lexical idiosyncrasy. Generally, the major structural changes are caused by the differences in part of speech of a equivalent concept in the two languages. It is clear that if we can deal adequately with these structure differences, we can improve translation quality considerably.

2.2 Lexical Influences on Construction of Target Sentence

Individual words can sometimes influence the selection of structure in construction of a target sentence. The lexical influence on constructions is complex, including wide-range and local-range restructuring. The following are some examples of lexical influence on sentence construction.

2.2.1 Lexical Influence on Order

The selection of a target word may influence the order of phrases in a sentence. For example,

(19-a) She sang sadly.
(19-b) 她憂傷地唱歌。
(19-b) 她唱歌唱得很憂傷。

(20-a) She sang that song well.
(20-b) 她很好地唱那首歌。
(20-c) 她唱那首歌唱得很好。

Both (19-b) and (19-c) are appropriate translation for (19-a). But for (20-a), (20-c) is appropriate while (20-b) is not a good translation. This is because there is the adverb 很好 unlike 憂傷地 must locate after the verb in Chinese.

2.2.2 Lexical Influence on Selection of Sentence Construction

The conceptual content of a lexical item may determine the structure of the target sentence. For example,

(21-a) That book has been stolen
(21-b) 那本書被偷了。

(23-a) That book has been stolen by him.
(23-b) 那本書被他偷了。

(22-a) That book has been published.
(22-b) 那本書出版了。

(24-a) That book has been published by Tsin-Hua bookstore.
(24-b) 那本書是清華書局出版的。

The *bei-construction* (被字句) in Chinese is used essentially to express an *adverse* situation, one in which something unfortunate has happened, and also express *disposal*. That is, the *bei* sentence describes an event in which an entity or person is dealt with, handled, or manipulated in some way [Li-Thompson, 1982]. So we must check adverse and disposal features of each verb to decide whether to use the *bei-construction*. For example, since the verb *publish* in Chinese is not a *adverse* verb, (24-b) is the *shi-de-construction* (是-的-句), not the *bei-construction*.

2.2.3 Lexical Influence on Lexical Selection

The selection of target words may influence each other. For example,

(25-a) That book is not valuable.
(25-b) 那本書沒有價值。

There are four negative forms in common use in Chinese: 不, 別, 沒, 沒有. The scope position and form of negative particles in Chinese are decided mostly according to the features collected from the analysis phase. But there are words such as *valuable* in (25-a), Chinese counterpart "有價值" carrying the head "有". In this case, the negative form is "沒有". Other example is interesting and "有意思".

These cases can only be handled appropriately using a bilingual lexicon.

3. Lexicon-Driven Transfer

This section discusses the main idea of bilingual lexicon-driven transfer, and show how our framework treats structural changes caused by lexical idiosyncrasy described in Section 2.

3.1 Main Idea about Lexicon-Driven Transfer

We propose to resolve the structure difference in translation due to lexical idiosyncrasy according to the following considerations:

1. The transfer needed should be captured in the lexicon.
2. The intermediate representation should encode a clause, a verb phrase and a noun phrase as similar as possible so that easily interchangeable.This idea is similar to [Allen 1987] which propose that the logic form for NPs that describe events should be virtually identical to the representation of sentences that describe event.
3. The set of transfer operations should be kept minimal for simplicity and efficiency.

The intermediate representation has three layers: *event*, *entity*, and *lexeme* corresponding to the syntactical structures of clause, noun phrase and word. The lexeme layer is atomic containing only the target word while the event and entity layers share the same structure with a head and various thematic cases and modifiers. The head of an event is of course the main verb and that of an entity is the head noun. And to operate on this representation, we proposed four basic operations for encoding transfer:

1. **Raise** : a constituent in the intermediate representation can be raised one step up the constituency structure without changing the slot names (functional role) of its subconstituents.
2. **Modify** : the slot name of a constituent can be changed into another.
3. **Insert** : a constituent of any slot name can be inserted on the same level of the lexical item being considered.
4. **Delete** : a constituent can be deleted from the intermediate representation.

3.2 How to Deal with Structure Differences

Following are some examples which show what structure changes caused by lexical idiosyncrasy between English and Chinese languages have been done using these four operations.

3.2.1. *Raise* operation

Example (2-a): He is a good speaker of English.

The source intermediate representation using Direct Acyclic Graph (DAG)notation is shown in Figure 2.:

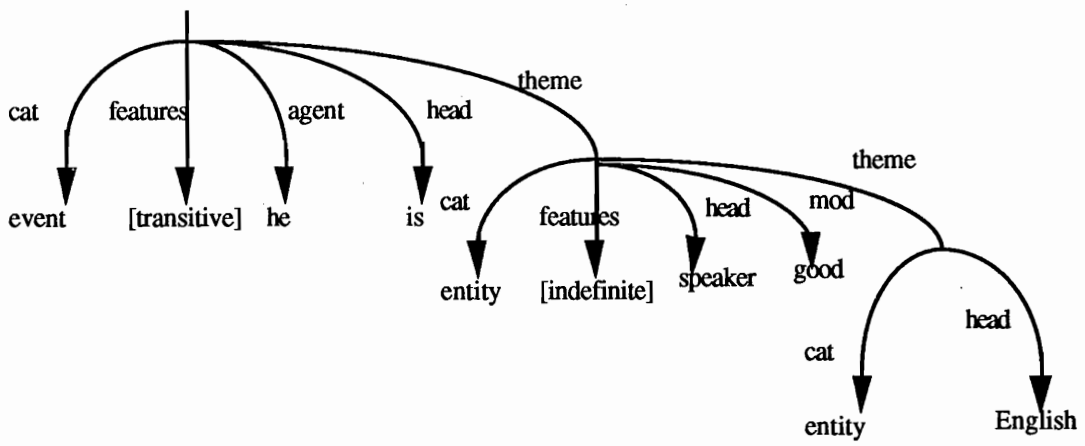


Figure 2. DAG of *He is a good speaker of English*

The linear format of representation is the following:

[cat : event, features : [transitive],
agent : [cat : pro, lex: he],
head: [cat: bv, lex: is],
theme:[cat:entity, features:[indefinite],
mod:[cat: adj, lex: good],
head:[cat: n,lex: speaker],
theme: [cat: n, lex: English]]].

If we generate a Chinese sentence, using this intermediate representation directly, we will get "他是個好的英語講者".That is an inappropriate sentence in Chinese. To get the appropriate sentence, we need to raise the noun phrase to the verb phrase position, resulting the DAG in Figure 3 as the target intermediate representation, with Chinese lexical item inserted in the DAG.

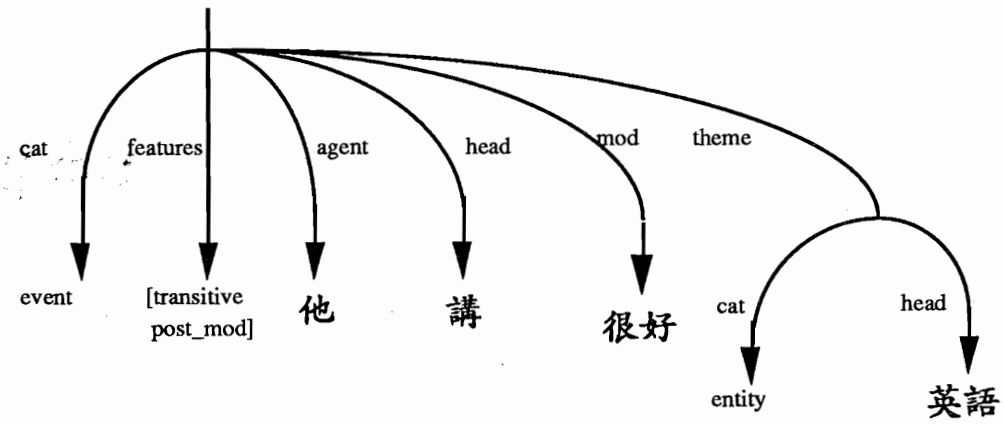


Figure 3. DAG of 他講英語講得很好

The corresponding linear form is

[cat : event, features : [transitive, post_mod],
agent : [cat : pro, lex: 他)],
mod:[cat: adj, lex 很好],
head: [cat: v, lex: 講],
theme:[cat: n, lex: 英語]].

Using this target representation, we can generate a fluent Chinese sentence. The crucial point is that how and when we can raise Figure 2 to Figure 3 to get suitable target intermediate representation and then generate a appropriate Chinese sentence. Let us consider, for example, some possible lexical entries for the noun "speaker" :

lex(speaker, hn, [human],[head_of(theme),is(head_of(sentence):cat,bv))],
[delete(head), raise(*)],講).
lex(speaker, hn, [human], [head_of(theme_mod)], [raise(*),insert_f(theme,transitive),
modify(theme:cat:entity, theme:cat:event)],講).
lex(speaker, hn, [inanimate], [], [], 喇叭)

Six arguments in each lexical entry are (1)English word, (2)part of speech, (3)semantic attribute list, (4)condition test list, (5)transfer action list and (6)Chinese word. When we unify the first lexical entry of "speaker", the condition list, *[head_of(theme), is(head_of(sentence):cat, bv))]*,will be instantiated. If the logic form satisfies this condition list , that is, "speaker" is the head of theme

and the *cat* of the head of theme is *bv*, we then execute the action described in transfer rule list, raising theme to upper level of DAG and automatically replacing the original head of sentence with "講".

Example (2'): We consider him a good speaker of English.

The source DAG is shown in Figure 4.

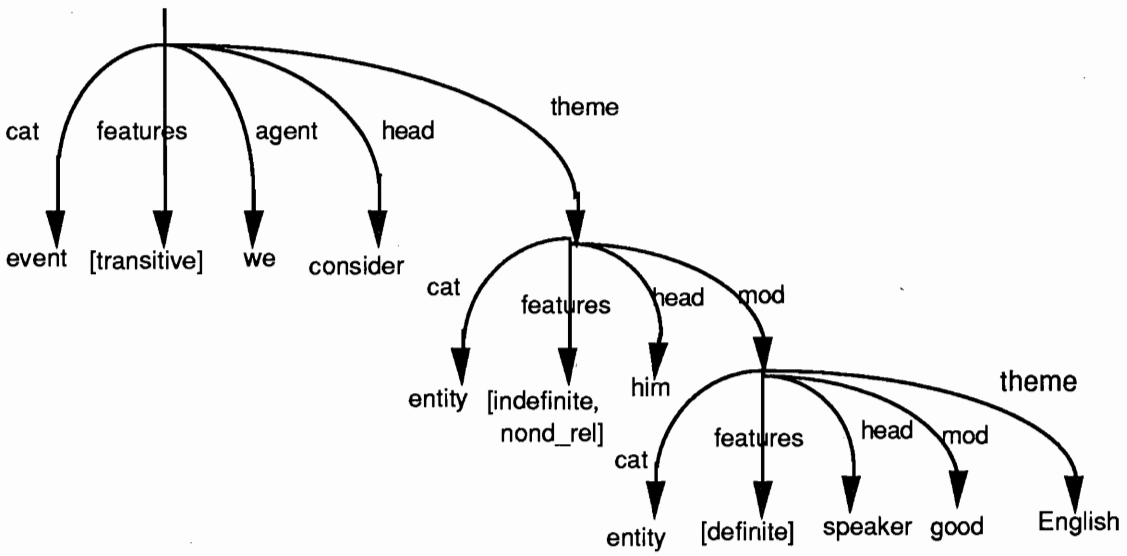


Figure 4. DAG of *We consider him a good speaker of English*

It is like above example (a), we execute the *raise* operation. But in this case, when we raise "講" up to clause level, it can not replace the original sentence head "認爲", because "認爲" is a verb with substance unlike "is". So the right thing to do seems to be raise the np containing this lexical item to a clausal level, by changing the *cat* slot from entity to event and inserting transitivity in the features slot.

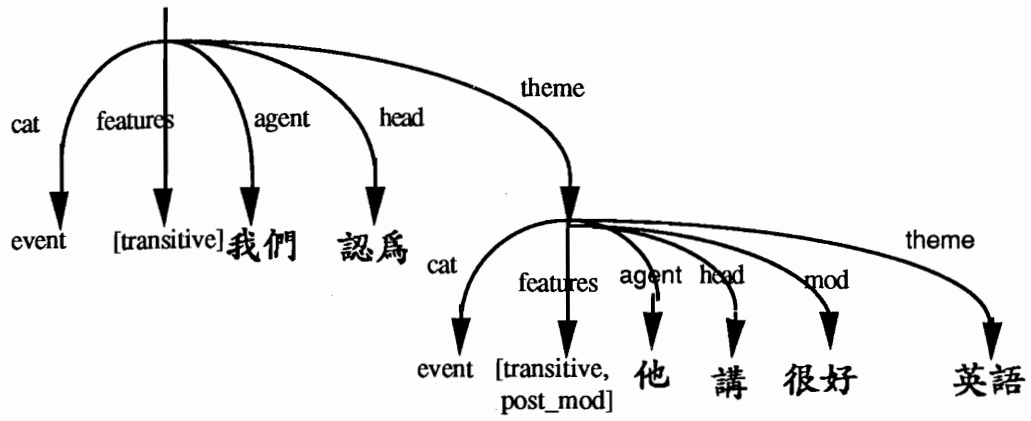


Fig 5. DAG of 我們認爲他講英語講得很好

3.2.2. *Modify* and *Insert* operations

Example (16-a): This surprised everybody.

The source DAG is shown in Figure 6.

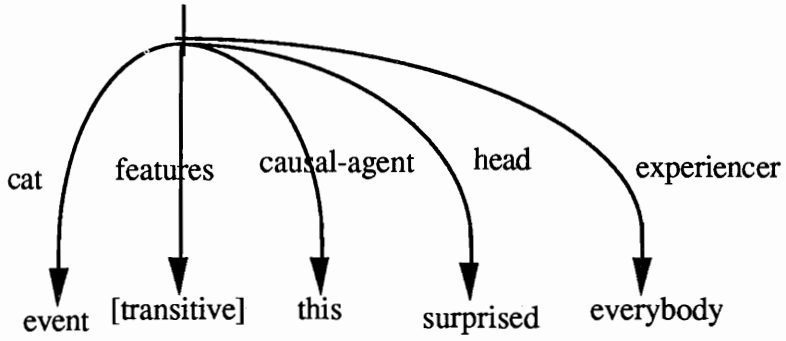


Figure 6. DAG of *This surprised everybody*

The equivalent Chinese lexical item for "surprise" is "使．．．震驚", so the "使" need to occur before the *experiencer* role. Let us consider the lexical entry of "surprise".

lex(surprise,v,_,[],[modify(head,head'),insert(head,"使")],震驚).

Executing the transfer operations described in Transfer-list, we will get new DAG shown in Figure 7.

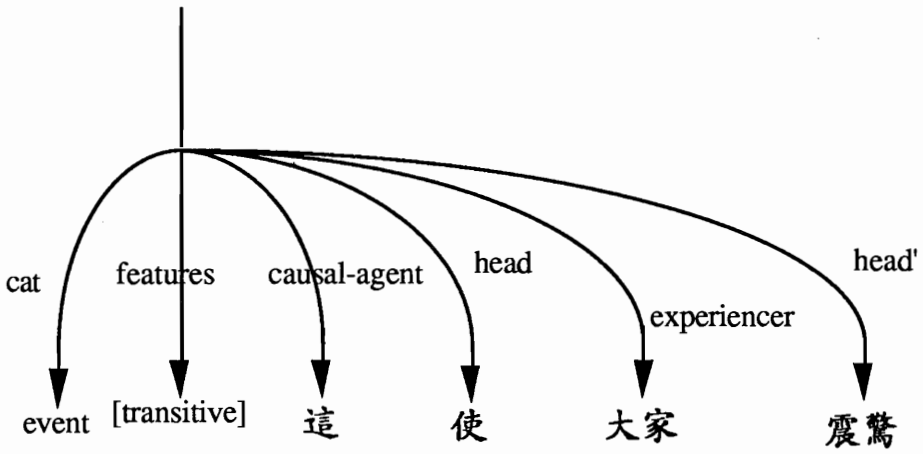


Figure 7. DAG of 這使大家震驚

3.2.3. *Delete* operation

Example (18-a): They lived a happy life.

The source DAG is shown in Figure 8.

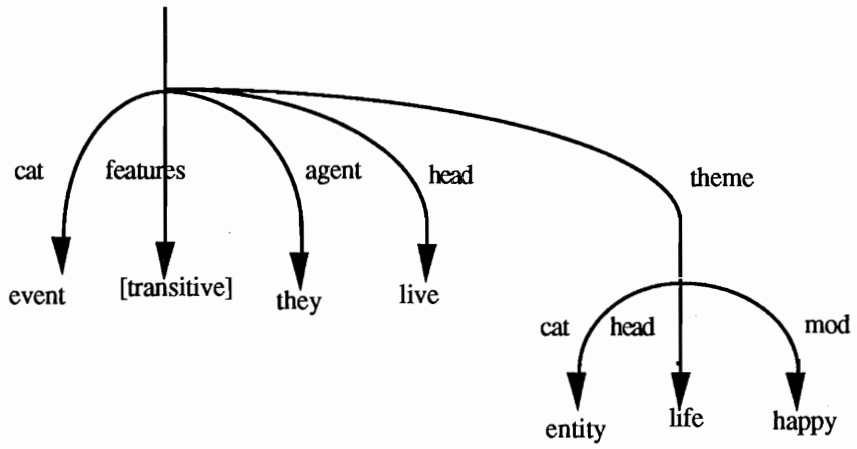


Figure 8 DAG of *They lived a happy life*

Live in English is transitive while "生活" in Chinese is intransitive, so the "life" must be deleted in target DAG. So the lexical entry of "live" is as follows:

lex(live,v,_,[is(head_of(theme):lex, "life")],
[raise(theme:mod),delete(theme),modify(f:transitive,f:intransitive)]
,生活).

Executing the *delete* operation, we will get the new DAG as Figure 9.

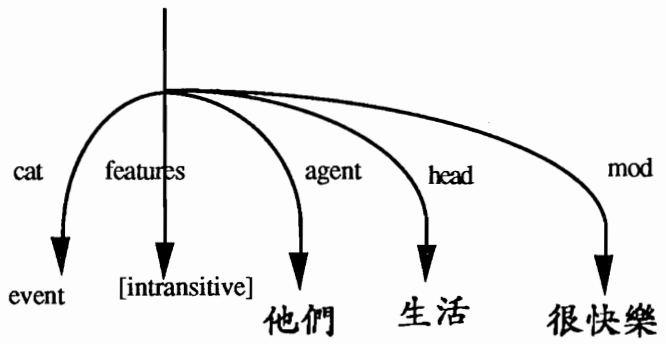


Figure 9. DAG of 他們生活得很快樂

3.2.4 Relative Clauses

Many relative clauses begin with a **relative pronoun**. The relative pronoun can acts as the subject or object in the relative clause, which always influences the translation of a sentence. In following, we will discuss the translation of relative clauses using the proposed mechanism. We will consider the two kinds of relative clauses, *defining* and *non-defining* relative clauses.

3.2.4.1 The translation of Defining Relative Clauses

Defining relative clauses explain which person or thing you are talking about. For example, if you say 'the teacher', it might not be clear who you mean, so you might say, 'The teacher who taught me English married yesterday'. In this sentence, 'who taught me English' is a defining relative

clause. Defining relative clause is a kind of quantifier, so the whole sentence should be translated as '教我英文的那個老師，昨天結婚了'

3.2.4.2 The Translation for Non-defining Relative Clause

Non-defining relative clauses give further information which is not needed to identify the person, thing, or group you are talking about. For example, 'The teacher, who taught me English, married yesterday' should be translated as '那個老師，教我英文，昨天結婚了' Here 'who taught me English' is only the added information, so we can take advantage of the topic-comment construction of Chinese, treat it as a comment in Chinese sentence and place it after the topic/subject.

Another important issue is the translation of the relative pronoun itself in non-defining relative clauses. For example, 'I like English, which is an interesting language' should be translated as '我喜歡英文，英文是種有趣的語言'. The relative pronoun 'which' refers to the preceeding object 'English' and often appears in non-anaphoric form in Chinese.

3.2.4.3 The Informations in Lexicon

We can then encode the above informations about the translation of relative clauses in lexicon as following.

lex(who, ip, _,[],[],誰)

lex(who, rp, _,[with(d_rel)],[delete],_)
lex(who, rp, _,[with(↑,subj)],[delete,modify(↑,comment),raise(↑),
    insert_f(↑,topic)],_)
lex(who, rp, _,[], [modify(↑,comment),raise(↑)], ~(↑:head:lex))

The '↑' sign means the upper level (parent node) of current node.

By these lexical transfer rules, the following English intermediate representation can be transferred to appropriate Chinese intermediate representation and then turn into suitable translation. For example:

[cat:event, f:[transitive],
agent:[cat:entity,f:[*defining(non-defining), (subj)*],
head:[cat:n,lex:teacher],
mod:[cat:event, f:[transitive],
agent:[cat:rp,lex:who],
head:[cat:v,lex:teach],
receipent:[cat:pro,lex:I],
theme:[cat:n,lex:English]]],
head:[cat:v,lex:marry],
time:[cat:n,lex:yesterday]]

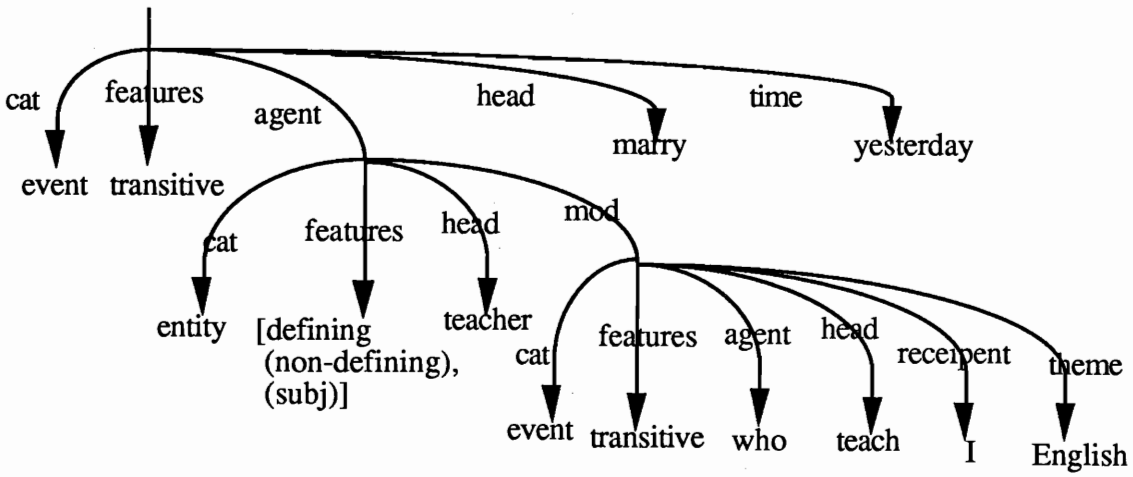


Figure 10. The DAG of English relative clause

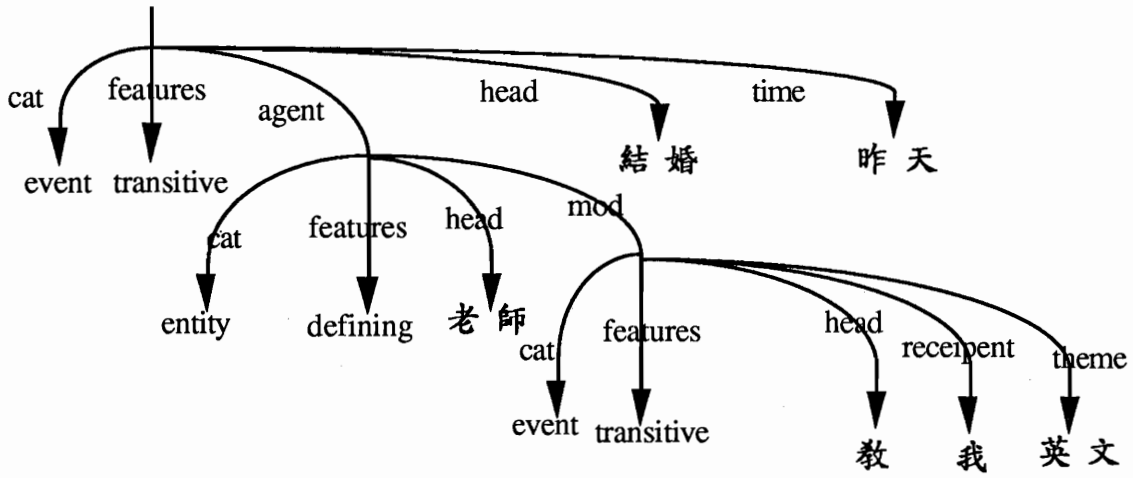


Figure 11. The DAG of Chinese defining relative clause

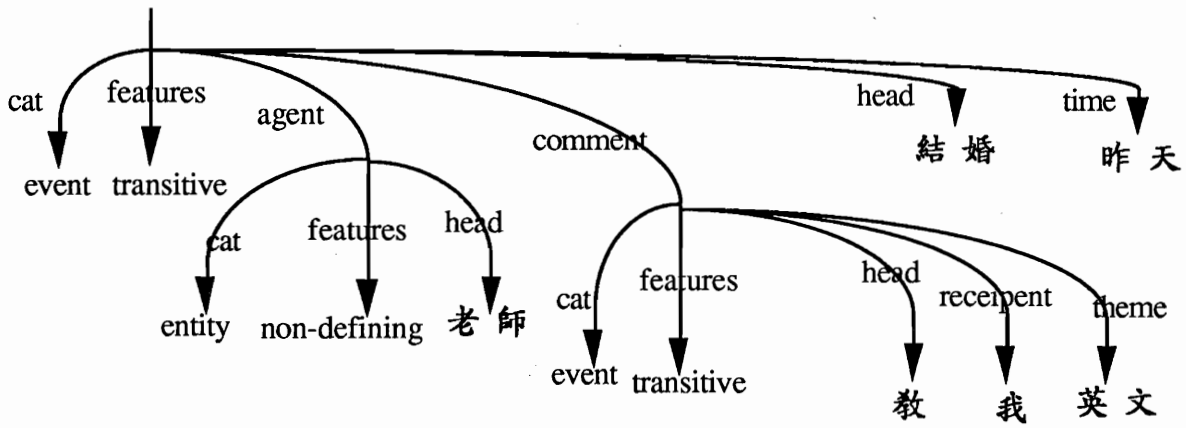


Figure 12. The DAG of Chinese non-defining relative clause

3.3 How to Deal with Special Constructions

In addition to processing structure mismatches problem, our lexical transfer rules in lexicon can also deal with lexical influences on constructions that we described in Section 2. To consider the sentences (19-a), *She sang sadly*, and (20-a), *She sang that song well*, the default structure building rule in our Chinese systemic grammar specify that the adverb should be located before verb, such as sentence (19-b),她憂傷地唱歌. But this default rule is not suitable for the case of

(20-a), because (20-b), 她很好地唱歌, is inappropriate. Let us consider the lexical entries for the adverb "well":

lex(well,adv,_,[], [insert_f(post_mod)],很好).

The condition list is empty, so the transfer action in transfer rule list will be fired, inserting feature *post_mod* in current level in DAG. Using this new DAG to generate Chinese sentence, we will get modifier into the right position, such as (20-b), 她唱歌唱得很好.

Similarly, Let us examine the sentences (21)-(24). Whether an English sentence with passive voice should be translated into *bei_construction* in Chinese or not is determined by the attributes of the Chinese head verb. So the lexical entries for the verbs "steal" and "publish" are as follows :

lex(steal,v,_,[with(passive)],[insert_f(bei)],偷).
lex(publish,v,_,[with(passive),has(agent)], insert_f(shi_der), 出版).
lex(publish,v _,[],[],出版).

As we mentioned in the previous section, since the verb "偷" has the attributes of *adverse* and *disposal*, we will generate bei-construction (被字句) for the passive sentence.

Besides structure changes caused by lexical items, selection of suitable target words has been problematic in MT. For example, the negative forms in English can be translated into different target expressions, sometimes depending on the target word of what item be negated. So we need a lexical transfer rule to decide what negative form to use in surface generation phase. Consider the lexical entry of adjective "valuable" :

lex(valuable, a, _, [with(negative)],[insert_f(mei)],有價值).

In the surface generation phase we will generated "不" as default negative form based on the syntactic and semantic analysis on the word "valuable". But the equivalent Chinese word containing the head "有", so it is necessary to replace "不" with "沒" in lexical transfer phase, otherwise, the negative form is inappropriate in Chinese.

4. Conclusions

4.1 Summary

Existing transfer-based MT systems, deal with all the differences of the two languages in one complex transfer phase [Chen 1988 b.], In this paper we present an approach that only deal with lexical idiosyncrasy using an active bilingual lexicon in transfer, to minimize the complexity of the transfer unit and ease the task of retargeting of a translator. It is a mixture of the interlingua and transfer approach [Tsutsumi 1991]. Let us discuss the advantage of this approach over the conventional transfer method in the following.

Our idea is similar to [Alonso, 1990], in making the intermediate representation as universal as possible based on case grammar [Fillmore, 1971],[Tang, 1975],[Huang and Wang 1988]. But this representation deviates from the interlingual approach in that it does not include a universal representation for lexical items [Nirenburg 1990]. For the characteristic of Interlingua, our system guarantees the independence of analysis and generation grammars, which is a basic requirement for practical multilingual MT systems, and at the same time, minimizes the size and complexity of the transfer modules, by using a bilingual lexicon.

Using this approach, we intend to achieve the following goals:

(1) The transfer module for a language-pair is reduced to the bilingual lexicon. The global syntactical reorganization is dealt with using a generator with an explicit grammar of the target language [Kuo 1989 ,Chen 1990 and Liao 1990].
(2) The source language analysis module is target-language independent.The analysis module produces an intermediate representation as output which is as interlingua as possible.
(3) The target language generation module is based on an explicit grammar which is completely source-language independent.

4.2 Future Work

Our system can be improved in the following respect:
(1) **Extending the scope of the grammar :** The grammar used in our system does not have a very large scope. We feel that the inclusion of the following is most urgent in improving translation quality:
1. Interogative sentences: Question word questions.
A-not-A question.
Particle question.
2. Serial verb construction.

3. Nominalization.
Besides, some existing parts should also be extended, such as arrangement of various cases in different type of sentences and selection of conjunction.

(2) **Implementing macros in transfer rules :** we can further define macros to represent the relative tests and actions for transfer. It can minimize the size and ease the maintainence of the transfer modules.

(3) **Using corpus to train transfer rules** : we plan to use a large English-Chinese bilingual corpus to train transfer rules stochastically. In this way, we hope to ease the work of analyzing and formulating transfer rules between these two languages.

5. Acknowledgement

This research was supported by the National Science Council under Contracts NSC79-0408-E007-17 and NSC80-0408-E007-13 and by the Telecommunication Laboratories under Contract TL-NSC-79-016.

References

[1] James Allen, Natural Language Understanding, The Benjamin/Cummings Publishing Company, Inc. pp 215, 1987.

[2] Juan Alberto Alonso, *Transfer InterStructure: Designing an "Interlingua" for Transfer-based MT Systems*, COLING 1990.

[3] D.A. Chen, *English and Chinese-A Comparative Study*, 新學識文教出版中心, 1988.

[4] J.R. Chen, And I.P.Lin, 利用ATN製作英中機器自動翻譯轉換生成子系統, Master Paper, Department of Computer Engineering, National Taiwan University, Taipei, Taiwan, 1988.

[5] C.C. Chen and J.S.Chang, *A Multi-Lingual Sentence Generator Based on Systemic Grammar*, Master Paper, Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan, 1990.

[6] C.J. Fillmore, *Some Problems for Case Grammar*, Gerogetown University Monograph Series of Languages and Linguistics 22. PP 35-36, 1971.

[7] C.R Huang and L.J Wang, 格框在機器翻譯的應用, ROCLING I, pp 99-125, 1988.

[8] H.W. Kuo and J.S Chang, *Systemic Generation of Chinese Sentences*, ROCLING II, pp. 187-212, 1989.

[9] J.W.Liao and J.S.Chang, Computer Generation of Chinese Commentary on Othello Games, ROCLING III, pp. 393-415, 1990.

[10] C.N. Li, and S.A. Thompson, *Mandarin Chinese - A Functional Reference Grammar*, University of California Press, California.

[11] Sergei Nirenburg, *Lexical and Conceptual Structure for Knowledge-Based Machine Translation*, ROCLING III, pp 105-130, 1990.

[12] Slocum, J. 1985. *A survey of Machine Translation: Its History, Current Status, and Future Prospects*. AJCL 11(1):1-17.

[13] Taijiro Tsutsumi, *Wide-Range Restructuring of Intermediate Representations in Machine Translation*, AJCL 16(2):71-78.

[14] T.C.C. Tang, *A Case Grammar Classification of Chinese Verbs*, Hai-Guo Book Company, Taipei, Twiwan 1975.

[15] Gr. Thurmair, *Complex Lexical Transfer in METEL*, COLING 1990.

[16] Jun-ichi Tsujii and Kimikazu Fujita, *Lexical Transfer based on Bilingual Signs -Towards Interaction during Transfer-*, ROCLING III, pp 141-157, 1990.

[17] C.C. Wu, Chinese-English Translation through Contrastive analysis, 文鶴出版有限公司, 1990 2nd Edition.

AUTOMATIC CHINESE TEXT GENERATION BASED ON INFERENCE TREES

Hing-Lung Lin
Benjamin K. T'sou
Hing-Cheung Ho
Tom Bong-yeung Lai
Suen Caesar Lun
Chi-yuen Choi
Chun-yu Kit
City Polytechnic of Hong Kong

Abstract

This paper describes a method for the generation of a coherent and continuous Chinese text from an inference tree. We argue that it is important to include information of rhetorical relations as part of the knowledge representation scheme in a rule-based expert system shell, in order to facilitate text generation of the inferred relationships. Applying the Rhetorical Structure Theory(RST) defined by Mann and Thompson[5,6], a set of rhetorical relations for Chinese rule-based inferencing is proposed. We observe that the rhetorical structure for an inference tree will be transformed after the inference tree is reasoned(or proved) by an expert system. Rules governing such transformation are derived. We also give an algorithm that can systematically generate multiple sentences of coherent Chinese text on the basis of the transformed rhetorical structure involving conjunctively and disjunctively conjoined constituents in Chinese.

1. Introduction

Natural language text generation (NLTG) can be viewed as a decision process, which determines what information to communicate, when to do it, and which syntactic structures and words might best express the author's intent. Generally speaking, NLTG can be divided into two stages, the strategic stage and the tactical stage [7]. Given a set of communicative goals, the strategic stage determines the content and structure of the discourse. At this stage, relevant information to be included in a text is determined, discourse strategy to control of the order information in the text is selected, and focus mechanism isused to monitor the progress of succeeding utterances so that the text can be easily understood.

On the other hand, the tactical stage uses a grammar and dictionary to realize in some natural language a single utterance produced by the strategic stage. An utterance can consist of one or more related propositions, which, in turn, corresponds to one or more simple sentences(or clauses) in the text. It is recognised that generating multiple sentences for a text is far more difficult than that of a single sentence, as the text generator must tackle such problems as pronominal reference and the use of conjunctions in order to produce a coherent and rhetorically sound text.

This paper addresses the tactical problem in Chinese text generation, where an utterance is represented by a proof tree of a rule-based expert system, as defined below.

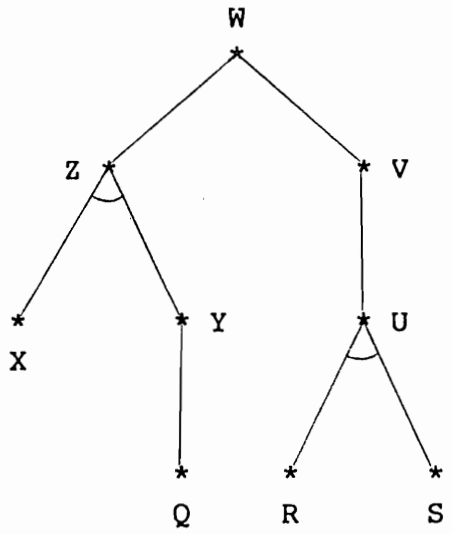
Generally speaking, an expert system is a computer program that is capable of reasoning and arriving at conclusions based on the knowledge it possesses. A rule-based expert system represents knowledge in terms of facts and rules. Facts are permanent or temporary knowledge that is unconditionally true. On the other hand, rules represent knowledge in a form that can be used for inference. Specifically, in a rule-based system, knowledge is represented as a series of "If-Then" rules based on propositional or predicate logic. In this paper, we are interested in two types of rules, namely, the AND rules (Conjunction) and the OR rules(Disjunction) as shown below.

Q :- P_1 AND P_2

Q :- P_1 OR P_2

Rules can be combined with facts to deduce new facts or arrive at conclusions. This process is known as inference. We can view an inference as a process of constructing a tree structure whose nodes are the clauses used in rules and whose branches are arrows connecting the clauses. When an AND rule is encountered, we have an "AND node". Otherwise, we have an "OR node". The branching in such a tree reflects the structure of a set of rules used in an inference. The tree so constructed is referred to be an AND/OR inference tree.

Fig. 1 shows an AND/OR inference tree (IT) for a set of rules in a knowledge base of some expert system.



W :- Z OR V
Z :- X AND Y
Y :- Q
V :- U
U :- R AND S

Figure 1

A proof P is an association of either the value True (T) or False (F) with each node of IT in such a way that all the rules in IT are not violated. A proof tree is an inference tree with an associated proof.

Our problem is to derive an algorithm that generates a paragraph of coherent and rhetorically sound Chinese text for a given proof tree. Our concern in this paper is not on how to generate an isolated Chinese sentence[2]. Instead, we are mainly concerned with clause concatenation, conjunctions and related issues of form and function in generating multiple-sentence Chinese text [1,6].

2. RST Analysis of AND/OR Rules

2.1 Review of Rhetorical Structure Theory

Rhetorical Structure Theory (RST) provides a theoretical basis for computational text planning and generation [5,6]. RST describes a text by assigning a rhetorical structure to it. Specifically, a

rhetorical structure represents a text as a tree, whose terminal nodes represent independent clauses appearing in the text, and non-terminal nodes represent instances of rhetorical relations, also called "schemas", which indicate how a particular unit of text structure is decomposed into other smaller units. Fig. 2 shows a generic rhetorical relation.

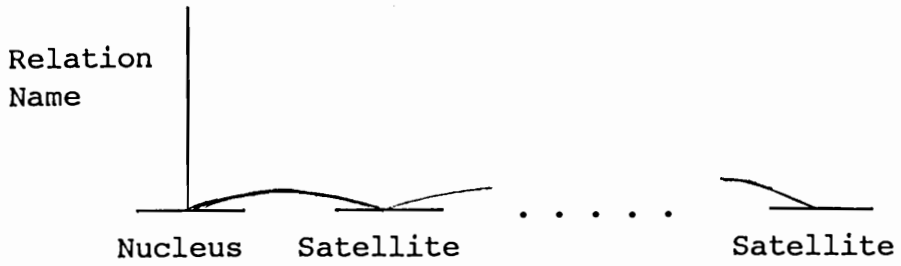


Figure 2

There are two or more text spans covered by a rhetorical relation. The text span pointed by a vertical line labeled with the relation name is called the nucleus, while the other spans are called satellites. A rhetorical relation can be symmetric or asymmetric. In a symmetric relation, functions of all the spans are of equal importance, but in an asymmetric relation, one span is more essential to the text than the other. The prominent and essential core span is the nucleus, and the other spans the satellites. The identity of the nucleus is part of therelation definition.

As pointed out by Mann and Thompson [6], the set of rhetorical relations is reasonably stable for any particular purpose, and they are, to certain extent, language-specific and culture-specific.

2.2 Rhetorical Relations for Rule-based Inferencing

As pointed out in Section 1, we are interested in two kinds of rules in a rule-based expert system, i.e. the AND rules and the OR rules. Without loss of generality, we assume that the rule body of an AND/OR rule consists of no more than two predicates.

To define a set of rhetorical relations for AND/OR rules, we propose that each rule should be represented as a two-level

rhetorical structure as shown in Fig. 3.

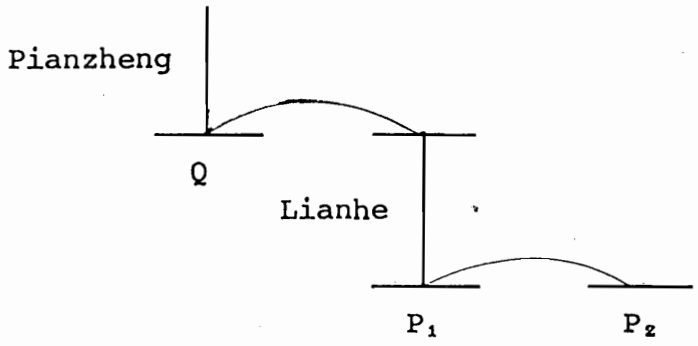


Figure 3

In Fig. 3, the upper rhetorical relation corresponds to the logical implication in a rule, and the lower rhetorical relation corresponds to the conjunction or disjunction of the two predicates that constitute the rule body. The set of rhetorical relations that can be used in the former shall be called Pianzheng relationship (偏正關係) and those used in the latter Lianhe relationship (聯合關係) .

Pianzheng relationship for a rule can be one of the following rhetorical relations:

1. Sufficient condition (假設關係) : This rhetorical relation applies to a rule whose body is the sufficient condition of its head. In Chinese, this relation is indicated by a pair of discontinuous constituents in a conjunction, e.g." 如果 ..... 那末 .....".

2. Necessary and sufficient condition (條件關係) : This rhetorical relation applies to a rule whose body is both the necessary and sufficient condition of its head. In Chinese, this relation is indicated by the discontinuous conjunction, e.g. " 只有 ..... 才 .....".

Both of the above rhetorical relations are asymmetric relations, where the text span corresponding to the head of a rule is the nucleus, while the other span corresponding to the body of a rule is the satellite.

On the other hand, Lianhe relationship for a rule includes the following rhetorical relations:

1. Disjunction (選擇關係) : This rhetorical relation applies to the rule body of any OR rule. This is a symmetric relation. In Chinese, this relation is indicated by the discontinuous conjunction, e.g. "或者 ..... 或者 .....".

2. Conjunction (並列關係) : This rhetorical relation applies to the rule body of an AND rule, where the two constituent predicates are semantically related. This is also a symmetric relation. In Chinese, this relation is indicated by the discontinuous conjunction, e.g. "一方面 ..... 一方面 .....".

3. Progression (遞進關係) : This rhetorical relation applies to the rule body of an AND rule, where the two constituent predicates are semantically related but one is more prominent and essential than the other. This is an asymmetric relation. In Chinese, this relation is indicated by the discontinuous conjunction,
e.g. "不但 ..... 而且 .....".

To facilitate text generation, each AND/OR rule in the knowledge base will be associated with two tags, denoted as {TAG1, TAG2}, where TAG1 indicates Pianzheng relationship and TAG2 indicates Lianhe relationship. If the rule body has only one predicate, TAG2 will be left blank.

2.3 Rhetorical Structure for Inference Tree

It is a straightforward procedure to construct a rhetorical structure for an inference tree. Every node of an inference tree always corresponds to some AND/OR rule with an associated rhetorical relationtags, {TAG1, TAG2}, as discussed in Subsection 2.2.

Starting at the root node N of an inference tree, we replace N by the two-level rhetorical structure shown in Fig. 3. The upper vertical line is labeled with TAG1 and the lower vertical line TAG2 of the rule corresponding to N. The nucleus span of the upper rhetorical relation is labeled with the head of the rule, which is the same as the predicate associated with N. The two spans of the lower rhetorical relations are connected to the left daughter and

the right daughter of N. The rhetorical relation indicated by TAG2 must be used to determine which daughter of N is connected to the nucleus span and which to the satellite span. Then, consider the left subtree of N, followed by the right subtree of N, using the same procedure above to replace their root nodes by the appropriate rhetorical structures. This procedure continues until all the nodes of the inference tree are exhausted.

3. Rhetorical Structure Transformation

Given the following rule with the associated rhetorical relations:

$Q(X)$:- $P_1(X)$ AND $P_2(X)$ { Sufficient condition , Progression }

Let P_1 stands for the clause " 投入更多的人力 " ,
P_2 stands for the clause " 提高機械化程度 " ,
Q stands for the clause " 農業現代化就會成功 " .

Furthermore, assume that X stands for some country.

Using the proper Chinese conjunctions for the associated rhetorical relation, we can generate the following text from the above rule. Note that in all the texts followed, conjunctions that are used to link clauses within a single sentence or across multiple sentences are underlined. How the texts are generated will be discussed in Section 4.

如果一個國家不但投入更多的人力，而且提高機械化程度，那末該國農業現代化就會成功。

It is known that P_1 and P_2 for country A are both true. After inferencing, it is deduced that Q is also true. This inferred result can be stated by the following text:

因爲A國家不但投入更多的人力，而且提高機械化程度，所以A國農業現代化就會成功。

The rhetorical relations associated with the above text stated in the same format as a rule should be { cause and effect (因果關係), progression }.

On the other hand, if it is proved that P_1 is true while P_2 is false for country B, we can no longer use the previous rhetorical relations to generate an easily understood text. Instead, the inferred result should be generated using another rhetorical relation combination{ possible effect(或然因果關係), concession(讓步關係)} so that proper Chinese conjunctions and order of clauses can be determined. Note that since the premise of the rule is found to be false, we can infer that the conclusion is probably false. The generated text is:

因爲B國家没有提高機械化程度，所以，縱然投入更多的人力，B國農業現代化大概仍然不會成功。

From the above discussion, we observe that : (1) Rhetorical relations are essential to select different conjunctions for clause linking and to determine the order of clauses in the text generated, and (2) the rhetorical relations associated with a rule can be changed after that rule is reasoned by an expert system.

3.1 Transformation of Rhetorical Relations

In the previous subsection, we have observed that the rhetorical relations associated with a rule will be transformed after that rule is reasoned by an expert system. The rules governing such transformationare described in Table 1 and 2.

after / before	Satellite is true	Satellite is false
sufficient condition	cause and effect	possible effect
necessary and sufficient condition	premise and condition	premise and condition

Table 1 Transformation rules for Pianzheng relationship

after / before	If P_1 and P_2 are both true or both false	If one is true and the other is false
disjunction	conjunction	adversativity / concession
conjunction	conjunction	adversativity / concession
progression	progression	adversativity / concession

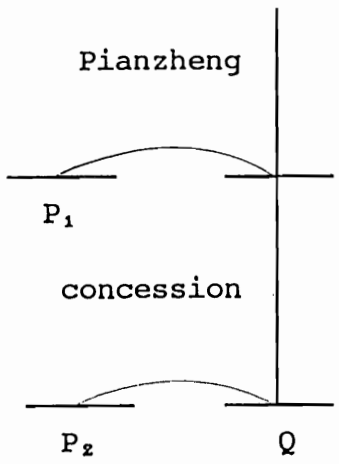
Table 2 Transformation rules for Lianhe relationship

In Table 2, if the truth values of P_1 and P_2 are different, the transformed rhetorical relations, namely, adversativity (轉折關係) or concession is applied to the whole rhetorical structure of a rule, not only the rule body. Furthermore, there is a mutual duality property between this pair of rhetorical relations as discussed in the following subsection.

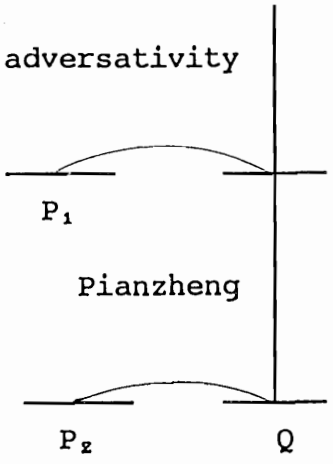
3.2 Transformation Rules for concession and adversativity

Concession and adversativity are good examples that different text can be generated for the same piece of information. These two rhetorical relations occur when, after reasoning, the truth value of P_1 and P_2 are found to be different for an AND/OR rule. The rhetorical structure of the inferred result can be represented in one of the two forms shown in Fig. 4.

Choosing which rhetorical structure of Fig. 4 depends on whether the rule is an AND rule or an OR rule, as well as the truth values of P_1 and P_2, as shown in Table 3.



(a) Rhetorical structure with concession



(b) Rhetorical structure with adversativity

Figure 4

Rule Type	P_1	P_2	Q	Rhetorical Structure
AND	F	T	F	Use Fig. 4(a)
AND	T	F	F	Use Fig. 4(b)
OR	T	F	T	Use Fig. 4(a)
OR	F	T	T	Use Fig. 4(b)

Table 3

Example 3.1

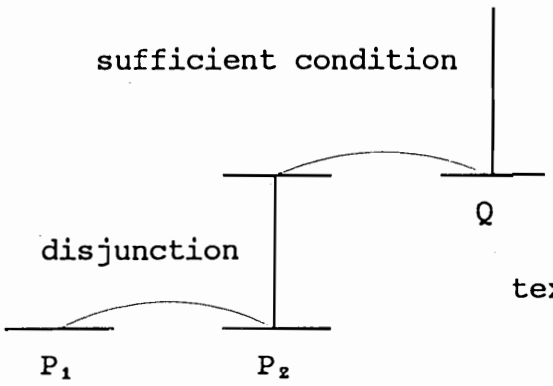
Given the following rule with the associated rhetorical relations:

Q(X) :- P_1(X) OR P_2(X) { sufficient condition, disjunction}

Let P_1 stands for the clause " 工資高 "
P_2 stands for the clause " 物價低 "
Q stands for the clause " 生活水準可以維持高水平 "

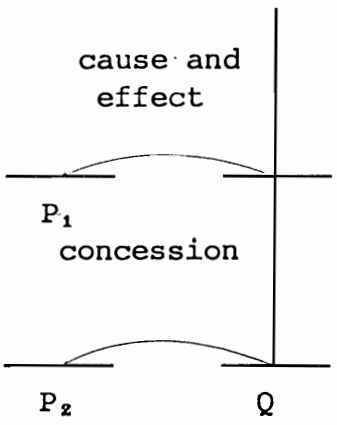
Fig. 5(a) shows the rhetorical structure for this rule, and the corresponding text generated using the appropriate conjunctions.

After this rule is reasoned by an expert system, it is discovered that P_1 is true and P_2 is false. Using Table 3, we can transform the rhetorical structure of Fig. 5(a) to that shown in Fig. 5(b) or 5(c).The texts for Fig. 5(b) and 5(c) are also given.



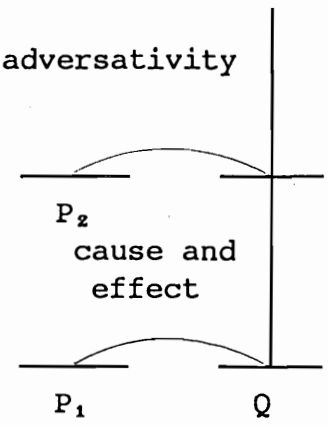
text: 如果或者工資高，或者物價低，那末生活水準可以維持高水平。

(a) Rhetorical structure for an OR rule with text



text: 因爲工資高，所以即使物價不低，生活水準仍然可以維持高水平。

(b) Transformed rhetorical structure with concession



text: 雖然物價不低，但是工資高，所以生活水準仍然可以維持高水平。

(c) Transformed rhetorical structure with adversativity

Figure 5

3.3 Rhetorical Structure for Proof Tree

Using the transformation rules developed above, we can generate many different rhetorical structures (accordingly, generate different texts) for the same inference tree, each corresponding to a different proof. Further, even for the same proof, the rhetorical structure generated is not unique. This non-uniqueness property is due to the followings. Firstly, for any symmetric rhetorical relation, we can randomly select one clause to be the nucleus and the other to be the satellite. The resulting rhetorical structure will be different.

Secondly, as discussed in Subsection 3.2, we can select either concession or adversativity to express an inferred rule. This non-uniqueness property of text generation allows us to select a text to be generated according to designated optimization criteria, or writing styles. Text selection and optimization will not be addressed in this paper.

We adopt the following deterministic top-down procedure to transform a rhetorical structure for an inference tree, given a proof. Starting at the root of the rhetorical structure, every step of the transformation takes a two-level rhetorical structure, corresponding to a rule in the original inference tree, and applies to it the transformation rules presented in Subsections 3.1 and 3.2. The transformation should preserve the structure of the original rhetorical structure as follows: (1) If the transformed rhetorical structure includes the concession or adversativity, then the satellite span in the bottom level of the original rhetorical structure will become the satellite span in the top level of the transformed rhetorical structure. (2) For the other rhetorical structures, the transformation should preserve the original identity of nucleus and satellite spans. See Fig. 6(a) to 6(c) for an example.

4. Chinese Text Generation for Rhetorical Structure

4.1 Rhetorical Relations and Chinese Conjunctions

Two or more simple sentences (or clauses) can be linked to form a compound or complex sentence by means of suitable conjunctions. Let x and y be a pair of discontinuous constituents in a conjunction, and

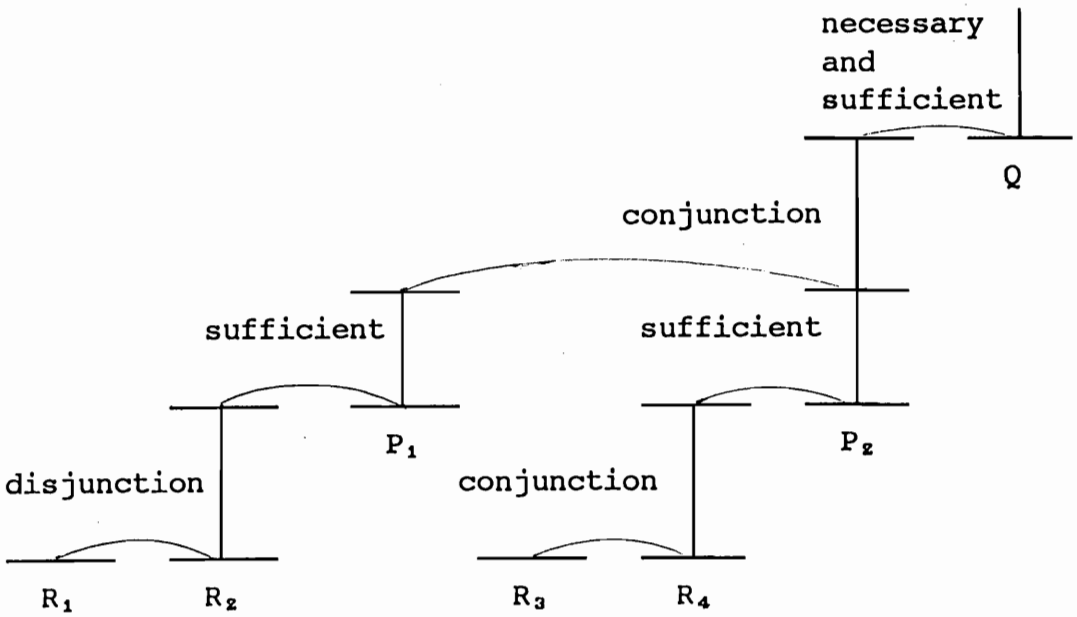


Figure 6(a) Rhetorical structure for an inference tree

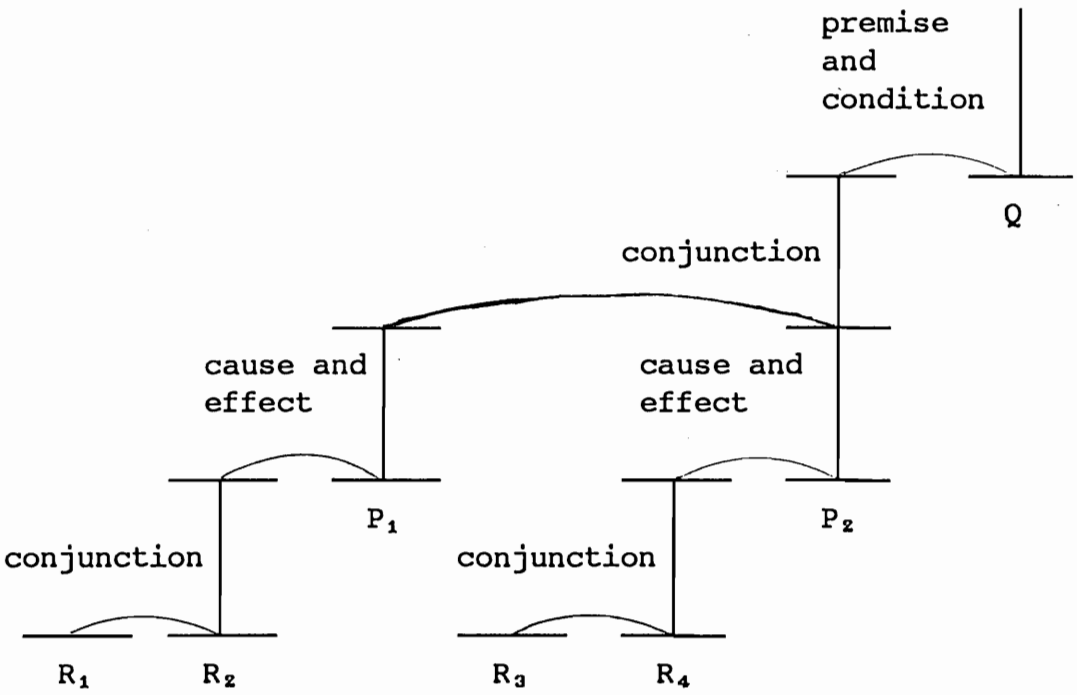


Figure 6(b) Rhetorical structure for the inference tree shown in Fig. 6(a) with an associated proof = {(R_1,T), (R_2,T), (R_3,T), (R_4,T), (P_1,T), (P_2,T), (Q,T)}

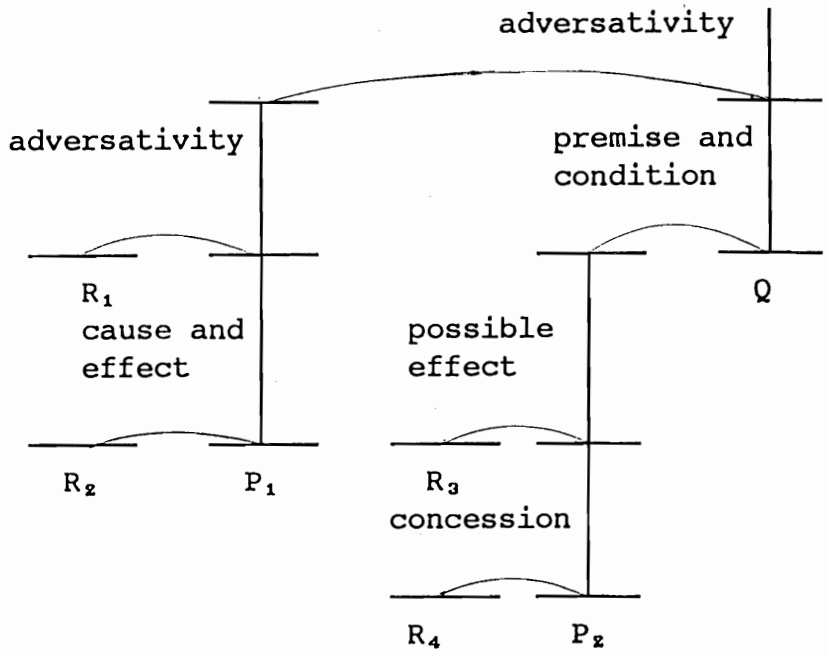


Figure 6(c) Rhetorical structure for the inference tree shown in Fig. 6(a) with an associated proof = {(R_1,F), (R_2,T), (R_3,F), (R_4,T), (P_1,T), (P_2,F), (Q,F)}

A and B be two clauses. Then, to join A and B together, we can use either one of the following two formats:

Format 1: xA, yB
Format 2: A, yB

In Format 2, there is the omission of the first discontinuous constituent in the conjunction, associated with the first clause (which is usually the satellite span in an asymmetric rhetorical relation). Generally speaking, with few exceptions, it is grammatically incorrect in Chinese to omit the constituent associated with the second clause (which is usually the nucleus span). Omitting both constituents will usually make the meaning of the resulting sentence logically ambiguous.

Mapping from the rhetorical relations discussed in Section 3 to their corresponding Format 1 or 2 of the generally paired conjunctions are given in Appendix A. Note that this mapping is a one to many mapping.

4.2 A Chinese Text Generation Algorithm for Rhetorical Structure

A rhetorical relation (RR) can be represented by:

RR_Name (Satellite, Nucleus).

A rhetorical structure (RS) can be defined by its root rhetorical relation, whose two spans are themselve s rhetorical structures. We denote the rhetorical structures connected to the satellite and the nucleus RS_Satellite and RS_Nucleus respectively. Therefore,

RS = Root_RR_Name (RS_Satellite, RS_Nucleus)

This definition of rhetorical relation can be applied recursively until a terminal node is reached. In that case, the RS is set to be the clause p associated with that node. Furthermore, if p is assigned a truth value according to a given proof, then the RS is set to be p (for True) or -p (for False) accordingly. For example, the transformed rhetorical structure shown in Fig. 6(c) can be represented by:

RS1 = adversativity(adversativity(-R_1,cause and effect(R_2,P_1)),
premise and condition(possible effect(-R_3,concession(R_4,-P_2)),
-Q))

The above list representation for a rhetorical structure will be the basis of the following text generation algorithm.

1. The list representation is processed in a left-to-right order. Scanning from the left,when the first relation is encountered, its relation name is used to search the conjunction table shown in Appendix A. If the first argument of this relation is a simple predicate, then a Format 1 conjunction pair is selected, otherwise, a Format 2 conjunction pair is selected. Note that the first constituent of a Format 2 conjunction pair is always absent.

2. Drop the relation name and the corresponding parentheses. Append the first conjunction to the first argument(i.e. the satellite) of this relation, and the second conjunction to the second argument (i.e. the nucleus).

3. Punctuations are assigned according to the following rules:

a. If the first argument is a clause, then insert a comma after the first argument.

b. If the second argument is not a clause, then insert a comma after the second conjunction.

c. Insert a fullstop after the second argument, if no punctuation has been assigned to this position.

4. Repeat Steps 1 to 3 until there is no more relation name and parentheses left in the generated text.

Using this algorithm, we obtain the following text for RS1.

雖然 $-R_1$，但是，因爲 R_2，所以仍然 P_1。然而，由於 $-R_3$，因此，即使 R_4，大抵仍然 $-P_2$。由此推論 $-Q$。

5. A Method for Chinese Text Generation for Inference Tree

To generate a coherent and rhetorically sound Chinese text for a given proof tree (i.e. an inference tree with an associated proof), we have to carry out the following steps:

1. Generate the rhetorical structure for the inference tree.

2. For each terminal node (corresponding to a predicate of the inference tree) of the generated rhetorical structure, associate its corresponding truth value from the given proof.

3. Transform the rhetorical structure with the associated truth values to a new rhetorical structure using the transformation rules discussed in Section 3.

4. Generate a Chinese text for the transformed rhetorical structure using the algorithm presented in Section 4.2.

We use the following example to illustrate the above method for Chinese text generation:

Example 5.1

Given the following three rules and their corresponding rule texts.

rule_0: Q :- P_1 AND P_2 { necessary and sufficient condition, conjunction }
只有當一個人身體弱，加上經濟差，他才需要別人救濟。

rule_1: P_1 :- R_1 OR R_2 { sufficient condition, disjunction}
如果一個人或者患病，或者年紀老邁，那末他身體弱。

rule_2: P_2 :- R_3 AND R_4 { sufficient condition, conjunction}
如果一個人沒有積蓄，加上失業，那末他經濟差。

Fig. 7 shows the inference tree for the above rules. We are required to generate a paragraph of Chinese text for the following purpose.

已知張三患病，年紀老邁，沒有積蓄，失業，請説明張三的現況。

To generate the required text, we have to carry out the 4 steps shown above.

1. The inference tree is replace by its corresponding rhetorical structure shown in Fig. 6(a).

2. According to the requirement stated above, the inference tree is reasoned by the expert system and the following proof is obtained.

Proof = {(R_1,T), (R_2,T), (R_3,T), (R_4,T), (P_1,T), (P_2,T), (Q,T)}

3. The rhetorical structure of Fig. 6(a) is transformed to a new rhetorical structure shown in Fig. 6(b).

4. The Chinese text generated for the transformed rhetorical structure is as follows.

張三除了患病，兼之年紀老邁，因此身體弱。同時，他一方面沒有積蓄，一方面失業，從而他經濟差。由此推論他需要別人救濟。

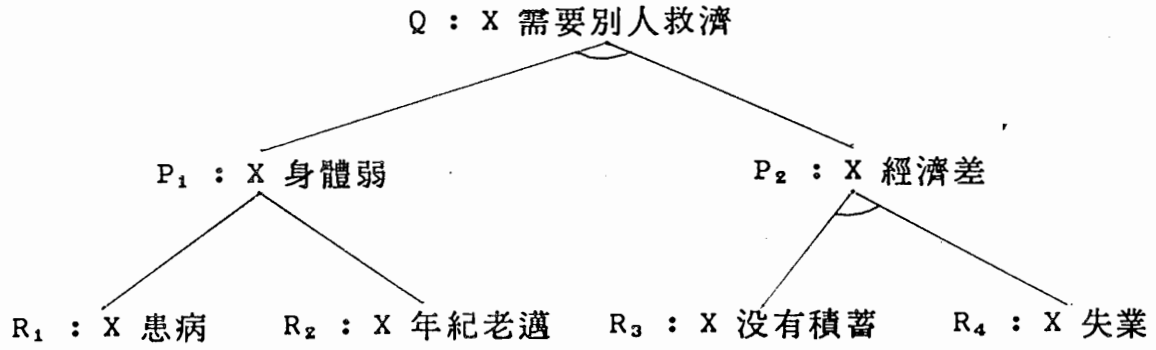


Figure 7

Example 5.2

For the same inference tree shown in Fig. 7, we are required to generate another paragraph of Chinese text to describe the followings.

已知李四没有患病，年紀老邁，有積蓄，失業，請説明李四的現況。

To generate the required text, we carry out the 4 steps as before.

1. The rhetorical structure of the inference tree is the same as before.

2. According to the requirement stated above, the inference tree is reasoned by the expert system and the following proof is obtained.

Proof = {(R_1,F), (R_2,T), (R_3,F), (R_4,T), (P_1,T), (P_2,F), (Q,F)}

3. The rhetorical structure of Fig. 6(a) is transformed to a new rhetorical structure shown in Fig. 6(c).

4. The Chinese text generated for the transformed rhetorical structure is as follows.

雖然李四没有患病，但是，因爲年紀老邁，所以身體仍然弱。然而，由於他有積蓄， 因此，即使他失業，經濟大抵仍然不差。自然他還不需要別人救濟。

6. Conclusions

Traditional study of knowledge representation emphasizes the impact of a representation on the process of inferencing, and disregards its effect on text generation. However, as pointed out by Mann and Thompson, "the relations of RST reflect a set of distinct kinds of knowledge that are given special treatment in text generation. It is therefore essential to represent these in the knowledge notations underlying a general text comprehender or generator." [6] Therefore, we propose in this paper a way of including information on rhetorical relations as part of the knowledge representation scheme in a rule-based expert system shell. By means of this rhetorical knowledge, this paper describes a method to generate Chinese text for proof trees which are the results of inferencing carried out by a rule-based expert system.

This study is the outgrowth of a research project which attempts to design and develop an automated Chinese text abstraction system (ACTAS) using a human-machine co-operative approach [8]. Inshort, ACTAS operates according to the followings. Information concerning a designated text is first digested by ahuman informant, who will then interact with ACTAS by means ofanswering a series ofquestions, which ACTAS automatically generated, with the assistance of a domain knowledge base and an inference engine, in order to acquire knowledge on the significant facts and the flow of argumentation in the original text. This acquired knowledge, or the abstract of the original text, is represented in the form of a proof tree in ACTAS. This proof tree is then transformed into a paragraph of rhetorically sound and easily understood Chinese text using the method presented in this paper.

References

[1] Bree, D.S. and Smith, R.A., "Linking Propositions," Proc. 1986 COLING/ACL Conference, Bonn, 1986.

[2] Kuo, Hwei-Ming and Chang, J.S., "Systemic Generation of Chinese Sentences," Proc. ROCLING II, R.O.C., 1989.

[3] Mann, W.C., "Discourse Structure for Text Generation," Proc. 1984 COLING/ACL Conference, Stanford, 1984.

[4] Mann, W.C. and Thompson, S.A., "Text Generation: The Problem of Text Structure," Proc. 1986 COLING/ACL Conference, Bonn, 1986.

[5] Mann, W.C. and Thompson, S.A., "Rhetorical Structure Theory: A Theory of Text Organization," in Discourse Structure, L. Polanyi (Ed.), Ablex, N.J., 1987.

[6] Mann, W.C. and Thompson, S.A., "Rhetorical Structure Theory: Description and Construction of Text Structures," In Natural Language Generation -- New Results in Artificial Intelligence, Psychology andLinguistics, G. Kempen(Ed.), Martinus Nijhoff Publishers, 1987.

[7] McKeown, K.R., "Discourse Strategies for Generating Natural-Language Text," Artificial Intelligence, vol.27, 1985.

[8] Tsou, B.K., Ho, H.C., Lin, H.L., Liu, G., Lun, C.S. and Heung, A., "Automated Chinese Text Abstraction: A Human-Machine Co-operative Approach," Proc. 1990 Int'l Conf. on Computer Processing of Chinese andOriental Languages, Changsha, 1990.

[9] 吳競存、侯學超，"現代漢語句法分析，" 北京大學出版社，北京，1982。

[10] 王福祥，"漢語話語語言學初探，" 商務印書館，北京，1989。

[11] "邏輯與語言研究，" 中國邏輯與語言研究會主編，中國社會科學出版社，1989。

Appendix

A. Conjunction Table

Note that only selected conjunction-pairs are included for each rhetorical relation in the following table. This is not an exhaustive listing.

Rhetorical Relation	Format	Conjunction-pair		Remarks
		1	2	
sufficient condition	1	如果	那末	(a)
		倘若	那末	
		要是	就	
necessary and sufficient condition	1	只有	才	(a)
		惟有	才	
		但凡	總	
cause and effect	1	因爲	所以	(b)
		由於	因此	
	2	----	因此	
		----	從而	
		----	以致	
premise and condition	1	既然	那麼	
	2	----	由此推論	
		----	可見	
		----	自然	
adversat-ivity	1	雖然	但是	(c)
		雖然	然而	
	2	----	但是	
		----	然而	
		----	不過	

Rhetorical Relation	Format	Conjunction-pair		Remarks
		1	2	
concession	1	即使	仍然	
		縱然	也	
		盡管	還	
		就算	仍然	
		誠然	可是	
conjunction	1	除了	兼之	
		一方面	另一方面	
		一來	二來	
	2	----	同時	
		----	加上	
disjunction	1	或者	或者	
	2	----	或者	
progression	1	不但	而且	
		不只	甚至	
	2	----	進而	
		----	甚至於	
		----	乃至於	

Remark

(a) This rhetorical relation will not appear in any transformed rhetorical structure.

(b) Possible effect will use the same conjunction pair, except that the clause of its nucleus span should include such adverbs as 大概, 大抵, 可能, 多半 etc.

(c) The clause of the nucleus span of the Pianzheng relationship that is coupled with this rhetorical relation should include such adverbs as 仍然, 還 etc.

A TRACE & UNIFICATION GRAMMAR FOR CHINESE

Hans Ulrich Block

Siemens AG, Corporate Research, ZFE IS INF 23

Otto Hahn- Ring 6, D-8000 München 83

block@ztivax.uucp

Ping Peng

Department of Computer Science

Courant Institute of Mathematical Sciences

New York University

715 Broadway, # 715, New York, NY 10003

peng@cs.nyu.edu

ABSTRACT

This paper presents a design and an implementation of a unification grammar for Chinese. Furthermore the Chinese grammar is designed as a reversible grammar which serves both parsing and generation. The Chinese grammar is developed under the system of Trace & Unification Grammar that compiles the grammar into an efficient parser and an efficient generator. The implementation shows that a set of Chinese grammar rules used for parsing and generation can be stated elegantly by the unification. Some examples illustrate how to formulate Chinese sentences by reversible grammar rules.

KEYWORD: Chinese grammar, Reversible grammar, Unification formalism.

1 Introduction

During recent years there has been a growing interest in NL systems that can be used for both parsing and generation. The ideas of a unification grammar that allows for a declarative description of language have made it possible to use the same grammar for both tasks. The main goal of designing a grammar then is to describe a relation between normalized (semantic) representations and language strings. If a grammar can be used in both directions of parsing and generation, we call it a "reversible grammar".

This paper discusses the design of a Chinese reversible grammar and describes its implementation in the system "Linguistic Kernel Processor" developed by Siemens AG, Corporate Research [3]. It has been tested as one component of a machine translation system "Multilingual Conversation Interpreter" which translates dialog-style texts between any pair of languages among English, German, Chinese and Swedish [2].

The reversibility of a grammar requires consideration of two aspects. One is the different procedural interpretations of a grammar in parsing and generation. This can be handled by a mechanism which automatically associates a control interpretation with each of the two opposite directions of computation (see [12], [11], [3] for detailed descriptions). The other one is a way how to formulate a natural language sentence by a grammar so that the grammar can be used both in a process of parsing and in a process of generation. This paper focuses on the second aspect of designing a Chinese grammar.

The mechanism with which the reversible Chinese grammar is written is the "Linguistic Kernel Processor". It provides natural language grammar writers with a tool for designing a reversible grammar, which serves for both natural language sentence parsing and generation. The formalism adopted by the "Linguistic Kernel Processor" is a variant of Unification Grammar combined with "movement rules" based on Government & Binding Theory, called "Trace & Unification Grammar" (TUG). A set of Chinese grammar rules written in this formalism are stated declaratively as context-free productions and PATR-II style feature equations. Context-free productions describe the surface structure of Chinese language strings. A mechanism of "movement rules" is built into production rules to

specify discontinuous dependencies within a language string. Feature equations specify a relation among features of phrases within a production. Unification is prescribed as the sole operation on the feature equations to make a bi-directional computation possible. The equations play roles both in composition of a semantic representation of a phrase from its children during parsing and in decomposition of a semantic representation of a phrase into its children during generation. The mechanisms of feature typing, mixing of attribute-value pairs and Prolog-terms unification, macros, and general disjunctions combine with feature equations to increase flexibility in information-combining and information passing, as well as syntactical and semantical composition and decomposition. The written Chinese reversible grammar is then compiled by the system into a LR parser [13] and a semantic-head-driven generator [10] to enhance the dynamic performance of the parser and the generator (See [3] for details of the introduction.) In the following chapters, we will first give a description of the TUG Formalism, then describe the basic features of the Chinese grammar and finally give some examples of paraphrases generated by the system for Chinese sentence inputs.

2 The TUG Formalism

The design of Trace and Unification Grammar has been guided by the following goals:

- **Perspicuity**. We are convinced that the generality, coverage, reliability and development speed of a grammar are a direct function of its perspicuity, just as programming in Pascal is less error-prone than programming in assembler. In the optimal case, the grammar writer should be freed of reflections on how to code things best for processing but should only be guided by linguistic criteria. These goals led for example to the introduction of unrestricted disjunction into the TUG formalism.

- **Compatibility to GB Theory**. It was a major objective of the LKP to base the grammar on well understood and motivated grounds. As TUG was originally applied to German and most of the newer linguistic descriptions on German are in the

framework of GB theory, it was designed to be somehow compatible with this theory though it was not our goal to "hardwire" every GB principle.

- **Efficiency**. As the LKP is supposed to be the basis of systems for interactive usage of natural language, efficiency is a very important goal. Making efficiency a design goal of the formalism led e.g. to the introduction of feature types and the separation of the movement rules into head movement and argument movement.

The basis of TUG is formed by a context free grammar that is augmented by PATR II-style feature equations. Besides this basis, the main features of TUG are feature typing, mixing of attribute-value-pair and (PROLOG-) term unification, flexible macros, unrestricted disjunction and special rule types for argument and head movement.

2.1 The framework

As a very simple example we will look at the TUG version of the example grammar in [8].

```
% type definition
s       => f.
np      => f(agr:agrmnt).
vp      => f(agr:agrmnt).
v       => f(agr:agrmnt).

agrmnt => f(number:number,person:person).

number => {singular,plural}.
person => {first,second,third}.

% rules
s ---> np, vp |
     np:agr = vp:agr.
```

```
vp ---> v, np |
     vp:agr = v:agr.

% lexicon
lexicon('Uther',np) |
     agr:number = singular,
     agr:person = third.
lexicon('Arthur',np) |
     agr:number = singular,
     agr:person = third.
lexicon(knights,v) |
     agr:number = singular,
     agr:person = third.
lexicon(knight,v) |
     ( agr:number = singular,
       ( agr:person = first
       ; agr:person = second
       )
     ;
       agr:number = plural
     ).
```

There are two main differences from PATR II in the basic framework. First, TUG is less flexible in that it has a "hard" context free backbone, whereas in PATR II categories of the context free part are placeholders for feature structures, their names being taken as the value of the cat feature in the structure. Second, TUG has a strict typing. For a feature path to be well defined, each of its attributes has to be declared in the type definition.

Besides defined attribute-value-pairs, TUG allows for the mixing of attribute-value-pair unification with arbitrary structures like PROLOG terms using a back-quote notation. This can be regarded as the unificational variant of the BUILDQ operation known from ATNs. As an example consider the following lexicon entry of *each* that constructs a predicate logic notation out of `det:base`, `det:scope` and `det:var`.

```
lexicon(each,det) |
     det:sem =
          'all(det:var,det:base ->
                         det:scope)
```

The usefulness of this feature for the construction of semantic forms will be shown in the section on the Chinese grammar.

TUG provides templates for a clearer organization of the grammar. The agreement in the above mentioned grammar might have been formulated like the following:

```
agree(X,Y) short_for
     X:agr = Y:agr.
...
s ---> np, vp |
     agree(np,vp).
```

TUG allows for arbitrary disjunction of feature equations. Disjunctions and Conjunction may be mixed freely. Besides well known cases as in the entry for *knight* above, we found many cases where disjunctions of path equations are useful, e.g. for the description of the extraposed relative clauses[1].

[1] [4] describes our processing technique for disjunctions.

2.2 Features

Features are defined at the beginning of the grammar. Features of a noun phrase (`np`) can e.g. be defined as:

```
np => f(semantics:sem, class:npclass, cmw).
```

By this definition, a noun phrase has three features. The feature `semantics` carries a semantic representation of the noun phrase. The feature `class` indicates a user defined semantic classification to which the noun phrase belongs, which helps to disambiguate syntactic structures of sentences. The feature `cmw` specifies that a designated classifier (see the discussion of 3.2. 1) is required by the noun phrase. Strict typing is used for the definitions of `semantics` and `class`. In operations on features, values of `semantics` and `class` are restricted to be an element of the pre-defined set `sem` and `npclass` correspondingly.

Features are used in grammar rules. The symbol `':'` is used as an infix operator for feature indexing. For example, `np:class` should be read as a value of the feature `class` of the noun phrase `np`.

2.3 Movement rules

Besides these more standard UG-features, TUG provides special rule formats for the description of discontinuous dependencies, so called "movement rules". Two main types of movement are distinguished: argument movement and head movement. The format and processing of argument movement rules is greatly inspired by [5] and [6], the processing of head movement is based on GPSG like slash features.

2.3.1 Head Movement

A head movement rule defines a relation between two positions in a parse tree, one is the landing site, the other the trace position. Head movement is constrained by the condition

that the trace is the head of a specified sister (the root node) of the landing site[2]. Trace and Antecedent are identical with the exception that the landing site contains overt material, the trace doesn't. Suppose, that v is the head of vk, vk the head of vp and vp the head of s, then only the first of the following structures is a correct head movement, the second is excluded because np is not head of vp, the third because antecedent and trace are unequal.

$[_{s'}$ v_i $[_s$ ... $[_{vp}$ ...

$[_{vk}$ ... trace(v)$_i$ ...]...]...]...]

$[_{s'}$ np_i $[_s$ ... $[_{vp}$ trace(np)$_i$ ...

$[_{vk}$ ... v ...]...]...]

$[_{s'}$ np_i $[_s$ ... $[_{vp}$ ...

$[_{vk}$ ... trace(v)$_i$ ...]...]...]...]

To formulate head movement in TUG the following format is used. First, a head definition defines which category is the head of which other.

```
v is_head_of vk.
vk is_head_of vp.
vp is_head_of s.
```

Second, the landing site is defined by a rule like

```
s' ---> v+s | ...
```

To include recursive rules in the head path, heads are defined by the following head definitions. In a structure $[_M$ D_1 ... $D_n]$ D_i is the head of M if either D_i **is_head_of** M is defined or D_i has the same category as M and either D_i **is_head_of** X or X **is_head_of** D_i is defined for any category X.

Head movement rules are very well suited for a concise description of the positions of the finite verb in German (sentence initial, second and final) as in

[2]Here, "head of" is a transitive relation s.t. if x is head of y and y is head of z then x is head of z.

Hat$_i$ der Mann der Frau das Buch gegeben t$_i$?

Has$_i$ the man the woman the book given t$_i$

Der Mann hat$_i$ der Frau das Buch gegeben t$_i$

The man has$_i$ the woman the book given t$_i$

... daß der Mann der Frau das Buch gegeben hat

... that the man the woman the book given has

All that is needed are the head definitions and the rule that introduces the landing site[3].

2.3.2 Argument Movement

Argument movement rules describe a relation between a landing site and a trace. The trace is always c-commanded by the landing site, its antecedent. Two different traces are distinguished, anaphoric traces and variable traces. Anaphoric traces must find their antecedent within the same bounding node, variable trace binding is constrained by subjacency, e.a. the binding of the trace to its antecedent must not cross two bounding nodes. Anaphoric traces are found for example in English passive constructions [$_s$ [$_{np}$ The book of this author]$_i$ was read t$_i$] whereas variable traces are usually found in wh-constructions and topicalization. Similar to the proposal in [5], argument movement is coded in TUG by a rule that describes the landing site, as for example in

```
s2 ---> np:ante<trace(var,np:trace), s1 |
     ante:fx = trace:fx,
     ...
```

[3]On a first glance, one might be tempted to consider head movement as a speciality of German syntax. This is not necessarily true, as it can e.g. also be used for the description of English Subj-Aux inversion.

Peter has been reading a book

Has$_i$ Peter t$_i$ been reading a book

As to Chinese syntax, the existence of head movement remains unclear at the moment.

This rule states that np:ante[4] is the antecedent of an np-trace that is dominated by s1. This rule describes a leftward movement. Following Chen's proposal, TUG also provides for rightward movement rules. A rightward movement rule might look like this.

```
s2 ---> s1, trace(var,np:trace)>np:ante |
     ante:fx = trace:fx,
     ...
```

The first argument in the trace-term indicates whether the landing site is for a variable (var) or for an anaphoric (ana) trace. Other than head movement, where trace and antecedent are by definition identical, the feature sharing of argument traces with their antecedents has to be defined in the grammar by feature equations (ante:fx = trace:fx, ...). Furthermore, it is not necessary that the antecedent and the trace have the same syntactic category. This is important for e.g. the rule for pronoun fronting in German might which can be stated along with rules like the following:

```
spr ---> pron<trace(ana,np), s | ...
```

The current version of the formalisms requires that the grammar contains a declaration on which categories are possible traces. In such a declaration it is possible to assign features to a trace, for example marking it as empty:

```
trace(np) | np:empty = yes.
```

Bounding nodes have to be declared as such in the grammar by statements of the form

```
bounding_node(np).
bounding_node(s) | s:tense = yes.
```

[4]The notation Cat:Index is used to distinguish two or more occurrences of the same category in the same rule in the equation part. :ante and :trace are arbitrary names used as index to refer to the two different nps.

As in the second case, bounding nodes may be defined in terms of category symbols and features[5]. Typical long distance movement phenomena are described within this formalism as in GB by trace hopping. Below is a grammar fragment to describe the sentence *Which books$_i$ do you think t_i John knows t_i Mary didn't understand t_i*:

```
bounding_node(s).
bounding_node(np).

s1 ---> np<trace(var,np), s | ...
s ---> np, vp | ...
s ---> aux, np, vp | ...
np ---> propernoun | ...
np ---> det, n |
vp ---> v, s1 | ...
vp ---> v, np | ...

trace(np).
```

The main difference of argument movement to other approaches for the description of discontinuities like extraposition grammars [7] is that argument movement is not restricted to nested rule application. This makes the approach especially attractive for a scrambling analysis of the relative free word order in the German *Mittelfeld* as in

Ihm$_i$ hat$_j$ das Buch$_k$ keiner t_i t_k gegeben t_j. The usefulness of this feature for the description of Chinese is described in [5] and [6].

3 Description of Chinese

In designing a reversible grammar, it is important to find an adequate description of linguistic knowledge that we would like to use for both parsing and generation. To accom-

[5]Currently, only conjunction of equations is allowed in the definition of bounding nodes.

plish this, it is necessary to have a grammar formalism be completely declarative and its interpretation order-independent. On the other hand, grammar rules should be tailored accurately, not only to provide large coverage for a sentence analysis but also to restrict overgeneration of language strings.

Concerning the Chinese language, its sentences are less structured than those of western languages. There are no relative pronouns or inflections. Sometimes, an active sentence and a passive sentence may share the same surface structure. Compare the following two sentences:

1. The English sentence "I walked." has a Chinese sentence equivalence:

   wo zou le.
   "I walk"

2. The English sentence "The book is bought." has a Chinese sentence equivalence:

   shu mai le.
   "book buy"

In these two sentences, correct constructions of the syntactic trees and the semantic representations are mainly derived from the lexical semantics within the sentences. Word order in a sentence can be very flexible, though the average length of a sentence is shorter than that in western languages. An object without any inflected marker in a sentence is dislocated frequently. To disambiguate syntactic structures and to build up correct semantic representations, semantic information in lexicon and word orders in sentences play very important roles.

In the Chinese grammar, features and feature structures are defined for each phrase. They carry necessary syntactic, semantic and pragmatic information for parsing and generation. These features are instantiated or passed through feature equations. Furthermore those feature structure are composed or decomposed level by level by feature equations.

3.1 A semantic representation

In our Chinese language processing, Quasi Logical Form, which is a contextually-sensitive logical form language [1], is chosen for the semantic representation of a sentence. Several sentences with different surface structures may be mapped into the same semantic representation. For example, Chinese yes/no questions appear regularly in different sentential forms:

- ni mai shu ma?
  "you buy book"
- ni mai bu mai shu?
  "you buy not buy book"
- ni mai shu bu mai?
  "you buy book not buy"
- ni mai shu bu mai shu?
  "you buy book not buy book"

In the analysis, the same semantic representation is produced from any of the above four sentences. In the generation, all of these four sentences are produced from the semantic representation. A tendency in the analysis of sentences is to discard some information which is not related to syntactic and semantic representations. In the above case, information about certain sentential style in a Chinese question is ignored while the information is necessary to designate which one of the above should be generated.

3.2 Selected examples

3.2.1 Classifiers

In Chinese, when a quantity word is used to describe a quantity of a noun, a classifier which is also called a count measure word must be inserted in between the quantity word and the noun. For an English phrase "one book", its Chinese equivalency is "yi (one) *ben* shu

(book)". Here, *ben* is a classifier associated with a quantity word for describing a quantity of the noun "shu" (book). For another English phrase "one car", its Chinese equivalency is "yi (one) *liang* che (car)". Here, *liang* is a classifier associated with a quantity word for describing a quantity of the noun "che (car)". Classifiers vary with nouns. Each classifier has to match the noun which a quantity word modifies. A selection of a classifier is not determined by s surface structure of a phrase, but by a lexical item of nouns.

The following fragment of grammar rules is used for parsing and generating a noun phrase with a quantity modifier.

```
np ---> cmwp, noun |                                          (1)
        cmwp:form = noun:cmw,
        np:sem = cmwp:sem,
        cmwp:restr = noun:sem.
cmwp ---> quantity, cmw |                                     (2)
          cmwp:form = cmw:form,
          cmwp:sem = 'qterm(quantity:sem,cmwp:restr).

lexicon(yi, quantity)| quantity:sem = 1.
lexicon(ben, cmw) | cmw:form = ben.
lexicon(shu, noun) | noun:sem = shu,
                     noun:cmw = ben.
```

Two points can be observed from the fragment of grammar rules.

1. feature passing and equality testing:

   In order to enforce a semantic restriction upon a classifier and a noun, we use the equation "`cmwp:form = noun:cmw`", which checks whether a value of the feature `form` of `cmwp` is the same as a value of feature `cmw` of `noun` in *rule (1)*. The value of the feature `form` of `cmwp` represents the value of the feature `form` of `cmw`. It is defined in the lexical item "`lexicon(ben,cmw) | cmw:form = ben`" and is passed to `cmwp` through the equation "`cmwp:form = cmw:form`'' in *rule (2)*.

2. classifier generation:

   Since a classifier is not coded into the semantic representation, a classifier generation can not be done with the input semantic representation. The solution in the fragment of grammar rules is to use lexical information. A value of the feature `cmw` is found in a lexical item of noun after the noun is selected. It is then passed to `cmw` through the two equations "`cmwp:form = noun:cmw`" and "`cmwp:form = cmw:form`". Some values which are discarded in a process of parsing but are useful to select a lexical item in a process of generation can be recovered correctly.

3.2.2 Topicalization

An usual word order of a Chinese declarative sentence is similar to that in English, that is, subject-verb-objects. Quite frequently, an object can be topicalized. The topicalized object is preceded by a syntactic marker "ba" ("ba" is called a "virtual particle" in Chinese) and is placed between a subject and a verb. The English sentence "I have bought a book." can be interpreted as:

- wo mai shu le.

  (*usual*): subject_verb_object

- wo ba shu mai le.

  (*topicalized*): subject_ba_object_verb

Overgeneration may arise. The problem is that an object topicalization sentence is allowed when the verb in the sentence has two objects or an adjunct such as the particle "le" (completed). For instance, a topicalized sentence "wo ba shu mai." is not adequate in daily dialog except in a Peking opera. To overcome the overgeneration, a feature is set up to detect an appropriate form. The following fragment of grammar rules shows how the feature play the role.

```
s ---> np, vp.                    (1)
vp ---> db.                       (2)
```

```
vp ---> ba, np<trace(ana,np:np_trace),db |          (3)
        db:weight = heavy.
db ---> v, np |                                     (4)
        db:weight = v:weight.
db ---> v, np, np |                                 (5)
        db:weight = heavy.
v ---> verb |                                       (6)
       v:weight = light.
v ---> verb, le |                                   (7)
       v:weight = heavy.
```

Here, a left movement rule gives the landing site of the np in rule (3). The similar treatment of the movement transformation has been proposed in [6]. An interesting point in this grammar rule is that the same movement rule used for parsing a ba-structure sentence is used for generating a surface structure of ba-sentence. In order to overcome the overgeneration mentioned above, the feature "*weight*" is created in a verb phrase. A value of the feature "*weight*" indicates the adequate surface structure that should be generated from a quasi-logic form. When the value of the feature "*weight*" is "*light*", the possibility of generating a ba-structure is eliminated. Only when the value of the feature "*weight*" is "*heavy*", a ba-structure can be derived from an internal semantic representation, that is, a quasi-logic form in our grammar.

4 Using a Paraphraser for grammar testing

In the above system, the declarative content of the Chinese grammar is shared by both the parser and the generator. The Chinese grammar is compiled dually into a parser and a generator automatically. The parser transforms a Chinese sentence into a quasi-logic form which we use for our internal semantic representations of languages in our machine translation system. The generator produces a Chinese sentence from

a quasi-logic form. We define the predicate *paraphrase(X, Y)* to show the reversible computation. The first argument X of the predicate *paraphrase* is bound to an input of a Chinese sentence. The second argument Y of the predicate *paraphrase* is any output of a Chinese sentence generated from the quasi-logic form to which the input Chinese sentence is transformed.

```
?- paraphrase(['wo^','ke^yi','bangzhu','ni^','ma'],O).
S E M A N T I K:
ynq(ke3yi(bangzhu(qterm(qcat(-,-,ex,sg),X,[event,X]),
      a_term(ref(_,th,perspron,1,_,sg,_),Y,[personal,Y]),
      a_term(ref(_,rh,perspron,2,_,sg,_),Z,[personal,Z]))))
O = ['wo^','ke^yi','bangzhu','ni^','ma'];
O = ['wo^','ke^yi','bu`','ke^yi','bangzhu','ni^'];
O = ['wo^','ke^yi','bangzhu','ni^','bu`','ke^yi'];
O = ['wo^','ke^yi','bangzhu','ni^','bu`','ke^yi','bangzhu','ni^'];

no
```

This example shows how the sentence

- wo keyi bangzhu ni ma?
  "Can I help you?"

is analysed and is generated from its semantic representation. The generator enumerates all possible paraphrases that are covered by the grammar for one semantic structure.

5 Conclusion

We have discussed some issues in designing a reversible grammar. We have shown how a reversible Chinese grammar can be designed under the formalism of Trace

& Unification Grammar. The examples illustrate how some Chinese language phenomena can be handled by the Chinese grammar. There are about one hundred grammar rules in our current Chinese grammar. It takes 0.2 to 1.2 seconds to parse and generate a sentence up to 10 words. The result shows that a reversible Chinese grammar not only is possible but also performs effectively in practical applications.

References.

[1]: Alshawi, H. "Resolving Quasi Logical Forms". *Computational Linguistics*, Vol. 16, pp. 133-144, 1990.

[2]: Alshawi, H., H. U. Block, D. Carter, B. Gambäck, R. Hunze, P. Peng, M. Rayner, S. Schachtl and L. Schmid. "Communication Multilingue par Forme Quasi Logique". *Proc. Expert Systems and their Applications, Avignon*, 1991.

[3]: Block, H. U. "Compiling Trace & Unification Grammar for Parsing and Generation". *Proc. The Reversible Grammar Workshop, ACL, 1991.*

[4]: Block, H. U. and L. A. Schmid. "Using Disjunctive Constraints in a Bottom-Up Parser". *Forthcoming.*

[5]: Chen, H.-H., I-P. Lin and C.-P. Wu. "A new design of Prolog-based bottom-up Parsing System with Government-Binding Theory". *Proc. 12th International Conference on Computational Linguistics* (COLING-88), pp. 112-116, 1988.

[6]: Chen, H.-H. "A Logic-Based Government-Binding Parser for Mandarin Chinese". *Proc. 13th International Conference on Computational Linguistics* (COLING-90), pp. 1-6, 1990.

[7]: Pereira, F. "Extraposition Grammar". *Computational Linguistics* Vol. 7, pp. 243-256, 1981.

[8]: Shieber, S.M. "The design of a Computer Language for Linguistic Information". *Proc. 10th International Conference on Computational Linguistics* (COLING-84), pp. 362-366, 1984.

[9]: Shieber, S.M. "A Uniform Architecture for Parsing and Generation". *Proc. 12th International Conference on Computational Linguistics* (COLING-88), pp. 614-619, 1988.

[10]: Shieber, S.M., G. van Noord, F.C.N. Pereira and R.C. Moore . "Semantic-Head-Driven Generation". *Computational Linguistics*, Vol. 16, pp. 30-43, 1990.

[11]: Strzalkowski, T. and Ping Peng. "Automated Inversion of Logic Grammars for Generation". *Proc. Conf. of the 28th Annual Meeting of the ACL*, (ACL-90) pp. 212-219, 1990.

[12]: Strzalkowski, T. "How to Invert a Natural Language Parser into an Efficient Generator: An Algorithm for Logic Grammars". *Proc. 13th International Conference on Computational Linguistics* (COLING-90), pp. 347-352, 1990.

[13]: Tomita, M. *Efficient Parsing for Natural Language: A fast Algorithm for Practical Systems.* Boston: Kluwer Academic Publishers, 1986.

CONSTRUCTING A PHRASE STRUCTURE GRAMMAR BY INCORPORATING LINGUISTIC KNOWLEDGE AND STATISTICAL LOG-LIKELIHOOD RATIO

Keh-Yih Su*, Yu-Ling Hsu, and Claire Saillard*****

*Department of Electrical Engineering
National Tsing Hua University
Hsinchu, Taiwan, R.O.C.
email: kysu@ee.nthu.edu.tw

**Behavior Design Corporation
2nd. Fl., 28 R&D Road II
Science-Based Industrial Park
Hsinchu, Taiwan, R.O.C.

*** Institute of Linguistics
National Tsing Hua University
Hsinchu, Taiwan, R.O.C.

ABSTRACT

Phrase structure grammar is one of the most important components in a syntax-oriented parsing system. However, constructing an adequate PSG is an arduous task. Either the traditional linguistic approach or the fully automatic inference approach has encountered several difficulties.

Thus, a human-machine cooperative method is suggested in this paper as a better approach. A statistical tool, **Log-Likelihood Ratio**, is proposed to enhance the productivity of human grammar writers. The Log-Likelihood Ratio of co-occurring tags is automatically computed by the computer to indicate the strength of linear association. The task of linguists is then to verify the relevance of groupings based on their linguistic knowledge. The advantages of this approach over other methods are pointed out, and the actual procedures are illustrated by a pilot experiment of constructing a Mandarin PSG. The experimental result shows the feasibility of the proposed approach.

1. Introduction

In a syntax oriented parsing system, parsing usually amounts to consulting a phrase structure grammar (PSG, hereafter) to check the well-formedness of the input strings and to generate their corresponding syntactic structures accordingly. Thus, PSG is one of the most important components of the whole parsing system.

However, constructing an adequate PSG is an arduous task. Traditional approaches resort only to linguists' own knowledge, and therefore are extremely labor-intensive and prone to incompleteness and incoherence in practical large-scale systems. Recently, owing to the advance of computer technology in providing cheap and fast computational power and the increasing

availability of machine-readable corpora, corpus-based statistical approaches are gaining prevalence in the community of computational linguistics. Plenty of systems propose to use statistics in their researches, including lexical analysis, category disambiguation, semantic models etc.[1] However, as for PSG construction, there are still no satisfactory methods available, neither traditional nor statistical.

Thus, a statistical tool, Log-Likelihood Ratio, is proposed in this paper to provide clues for linear association and enhance the productivity of human grammar writers. This approach intends to incorporate statistical information and linguistic knowledge in order to benefit from both the simple, objective, consistent characteristics of statistics and the better deductive power of ready-made linguistic analyses.

In the next section, two previous approaches of constructing a PSG are presented. Their advantages and drawbacks are demonstrated in detail. Then the Log-Likelihood-Ratio statistic is introduced. A fully automatic approach based on a so-called Generalized Mutual Information will also be discussed, with its shortcomings. Finally, the proposed cooperative approach will be presented. The actual procedures will be illustrated by a pilot experiment of constructing a Mandarin PSG. The experimental result shows the feasibility of the proposed approach.

2. Previous Approaches

In this section, we will describe two extremely different approaches of constructing a PSG. Their advantages and drawbacks will be discussed in detail.

2.1 Relying on Linguistic Knowledge Only

This approach has been traditionally used as a dominant way for constructing a PSG. In the initial phase, the cost of this method is minimal, because no collection of a large machine-readable database or preprocessing of the database is needed. A few linguists can build a preliminary PSG of a language in a relatively short time, by incorporating the ready-made theoretical linguistic analyses of this language. Since well-known linguistic analyses have gone through rigorous argumentations and been well-tested by lots of empirical data, they provide much insight about the language and their descriptive power is relatively strong.

However, theoretical linguistic researchers are apt to focus their attention on theoretically interesting phenomena and sweep the residual problems under the carpet. Unfortunately, theoretically interesting phenomena do not necessarily correlate to frequently occurring phenomena in real texts. Thus, although many aspects of grammatical structure are well-known and uncontroversial, authentic material still includes massive amounts of phenomena which have been

[1] See [9], [12], and [16], etc.

ignored or have not yet received consentient linguistic analyses. Especially in a language like Mandarin Chinese, where the linguistic phenomena are poorly studied, the contribution of the ready-made linguistic analyses to the construction of a Mandarin PSG is even more limited.

If ready-made linguistic analyses offer no guidance to the construction of a PSG, linguists have to work based on their own linguistic knowledge. In most cases, linguists start with a small set of data which is the basis for the first formulation of the grammar. Then, they gradually expand the data under consideration, using new data to test their original hypothesis and make decisions among competing analyses. The grammar is under reformulation until it covers most of the sentences in consideration. This method works well in theoretical researches or for small scale systems. However, when the set of data has been enlarged to thousands or millions of sentences, human simply can no longer successfully handle all the trivial linguistic phenomena, let alone the complicated interrelations among rules. Consequently, a purely linguistic approach to grammar construction arouses several problems in large scale systems.

Firstly, the PSG constructed in this way is prone to errors of omission. Human is not good at managing massive amounts of data. Without the help of the computer, linguists may ignore many trivial phenomena they do want to cover, and occasional mistakes are also inevitable.

Secondly, no simple and objective measure of the data is available for linguists to make tradeoffs between the coverage and the efficiency of the PSG. Ideally, a good PSG should define the class of "all and only" well-formed sentences of the language. But since authentic language is much more complex than theoretical linguists' descriptions commonly imply, this goal is hard to be achieved in practical systems. It may be clearer from what Sampson says : " If the activity of revising a generative grammar in response to recalcitrant authentic examples were ever to terminate in a perfectly leak-free grammar, that grammar would surely be massively more complicated than any extant grammar, and would thus pose correspondingly massive problems with respect to incorporation into a system of automatic analysis."[2] That is to say, attempting to construct a grammar accounting for all constructions in real-life texts is not feasible. Thus, some "omissions" of data are required. Most practical NLP systems will define the subject domain and style for their input texts and evaluate the importance of each construction according to the frequency of its real occurrences. If certain constructions have few occurrences in their domain, they will be discarded to avoid causing extra-complication of the system. However, without statistical information as a reference, the tradeoffs are difficult to be made.

Thirdly, linguists in this way do not have a general view of the linguistic phenomena involved during the process of grammar construction, and therefore modification shall very likely have to be made on preceding decisions if new data triggers new arguments in favor of a different solution. But without an objective measure of the real coverage of the PSG, grammar

[2] See [12], Chap. 2, 20.

writers cannot predict the actual influences caused by the modification of rules. Back-and-forth modifications are therefore hard to be avoided because linguists cannot guarantee their successive alterations of the rules lead to a global enhancement of the whole system. Consequently, the revision process will be full of back-and-forth operations, and it would be hard to imagine how the process should ever be concluded.

2.2 A Fully Automatic Approach Based on Grammatical Inference

Opposed to the purely linguistic approach mentioned above, a fully automatic approach based on grammatical inference has also been proposed. The principle of grammatical inference is to extract a grammar from a set of sentences, i.e. the sample or the learning set, which generates a set of sentences containing the sample. This procedure is an important subject in the study of syntactic pattern recognition because of its automatic learning capability. Several algorithms have been proposed and discussed.[3] Potential engineering applications of grammatical inference include areas of information retrieval, translation and compiling, and artificial intelligence, etc.

Generally speaking, the inferred grammar is a set of rules for describing the given finite set of strings from *L(G)*, the language generated by *G*, and predicting other strings which in some sense are of the same nature as the given set. A model for the inference of string grammars is shown in Figure 1. A set of sample terminal strings $\{x_i\}$ is fed into an adaptive learning algorithm, represented by the box in Figure 1, and a grammar *G* which is compatible with the given strings is obtained from the output.[4]

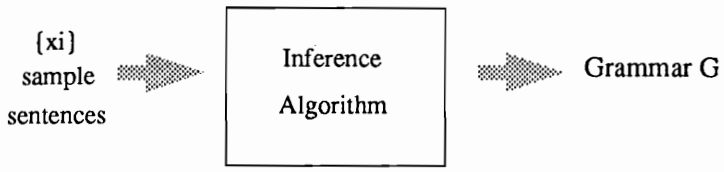


Figure 1 grammatical inference of string grammars

Thus, it is possible to directly infer a PSG from a set of sample sentences. In doing grammatical inference, the most popular method is to deduce the grammar in Chomsky Normal-Form. The Chomsky Normal-Form Theorem states that every context-free language can be generated by a grammar in which all productions are of the form $A \rightarrow BC$ or $A \rightarrow a$. Here A, B, and C are variables and a is a terminal. By the strategy of grammatical inference, a grammar with Chomsky Normal Form can be automatically inferred from the sample corpus.

At the first step, a set of sentences is selected as the corpus. Then, the corpus is tagged with lexical categories to reduce the number of terminal symbols of grammar rules. Finally,

[3] Detailed discussions on grammatical inference can be found in [11].

[4] See [11] and [17].

the inference algorithm is performed by the computer to automatically infer a grammar from the tagged corpus.

This automatic approach has some advantages. Firstly, since the inferred procedures are performed by the computer, it can be proved that the inferred grammar will perfectly cover all the sentences in the sample corpus.

Secondly, a fully automatic approach can reduce human intervention to a minimum. As humans are recognized as the most precious, yet most costly, resources in NLP systems, reducing human intervention will greatly enhance the cost-effectiveness of a system.

Nevertheless, this approach has several serious drawbacks. Firstly, since the the automatic construction of PSG does not take semantic relevance into consideration, the constituents constructed in this way are just ad hoc groupings which may not correspond to any traditional semantic concept. For many applications of NLP, such as text understanding and machine translation, semantic interpretation is an important process after syntactic parsing. Thus, the mismatch between the automatically-trained syntactic model and the traditional semantic model will cause difficulties for human linguists to attach semantic information to the syntactically analyzed structures.

Secondly, since the syntactic grammar inferred by automatical procedures is dramatically different from that of standard linguistic researches, the inferred grammar will not be able to couple with existing linguistic theories and thus to take advantage of the achievements of linguistic researches. As mentioned previously, most of the linguistic analyses are well-motivated and well-tested. They are valuable resources for related researches. Thus, it is a mistake to overlook the value of linguistic information and adopt a thoroughgoing automatic approach.

Thirdly, it is clear that the choice of the initial sample is critical in this approach. If the sample is too small, since all the rules are acquired exclusively from the corpus, the grammar may not be able to account for phenomena outside the sample space. But if the size of the sample is large, the number of inferred rules may become astronomically large and greatly increase the complexity of processing.

III. Using Log-Likelihood Ratio to Construct a PSG

As we have discussed, previous approaches for constructing a PSG have encountered several serious problems. Thus, a statistical tool, called Log-Likelihood Ratio, is proposed in this section to fertilize the construction of a PSG.

3.1 What Is Log-Likelihood Ratio

Log-Likelihood Ratio (LLR, hereafter) is a statistic measure of word associations. It compares the probability of a group of tags to occur together (joint probability) to their probability of occurring independently.

The bigram (with window size of 2) LLR, also called Mutual Information in the literatures, is computed by the formula:[5]

$$LLR_2(x,y) = I(x;y) = \log_2 \frac{P(x,y)}{P(x) \times P(y)}$$

where x and y are two tags in the corpus, and $LLR_2(x, y)$ (or $I(x; y)$) is the bigram Log-Likelihood Ratio (or Mutual Information) of the two tags x and y (in this order). $P(x)$ is evaluated as the relative frequency of the number of occurrences of x with respect to the number of total instances of singletons.

If there is a genuine association between x and y, then the joint probability $P(x, y)$ will be much larger than the chance $P(x)\times P(y)$, and consequently $LLR_2(x, y) >> 0$. If there is no interesting relationship between x and y, then $P(x, y) = P(x)\times P(y)$, and thus $LLR_2(x, y) = 0$. If x and y are in complementary distribution, then $P(x, y)$ will be much less than $P(x)\times P(y)$, and thus $LLR_2(x; y) << 0$.

3.2 An Automatic Approach Using Generalized Mutual Information

In the past few years, Mutual Information has been used in many areas of natural language processing, and has shown its success in different applications.[6] Recently, based on so-called Generalized Mutual Information (GMI, hereafter), an automatic constituent boundary parsing algorithm has been developed, which can derive a syntactic (unlabelled) bracketing for input tagged texts. In this approach, the tag sequences are processed using an n-ary-branching recursive function which branches at the minimum GMI value of the given window. Besides, for exceptional cases, a distituent grammar is constructed to specify a list of tag pairs which cannot be adjacent within a constituent.

Unfortunately, this approach has some drawbacks. Firstly, the formula of the GMI is not theoretically well-supported. It is heuristically expressed as a weighted sum of the Mutual Information based on the substring of the given context.[7]

Secondly, as mentioned, the bracketing of sentences in this approach is majorily determined by the value of GMI. A local minimum suggests the place to bracket. But this way of constructing

[5] For more details, readers are referred to [9].

[6] For example, [6], [7], [8], [9], and [18] have shown that Mutual Information is helpful in their researches.

[7] Interested Readers are referred to [2] for more details.

constituents still deviates from that of the standard linguistic researches. Conventionally, linguists determine the constituency of words not only by the strength of their linear co-occurrences, but more importantly also by their semantic relevance, or their substitutability and movability. It should be noted that tags in the same constituents should have higher GMI, but tags with higher GMI do not necessarily imply they are belonging to the same constituents. For example, verbs and determiners frequently occur together in sentences, thus the GMI for verb-determiner pair will be relatively high. However, linguists will never group verbs and determiners into the same constituent because they do not correspond to any semantic concept and do not act as a unit in syntactic operations (e.g. movement). Although the distituent grammar is constructed to make up this shortcoming, the adequate list of distituents is hard to be defined and is nevertheless source of inaccuracies.[8] As a consequence, a cooperative approach which incorporate both linguistic knowledge and statistical information is proposed in this paper to construct a PSG.

3.3 A Cooperative Approach Combining Linguistic Knowledge and LLR

This approach combines the advantages of the conventional linguistic knowledge-based method and those of the corpus-based, statistical approach. Firstly, a corpus with lexical tags is still required. Secondly, the LLR of co-occurring tags is automatically computed by the computer. The task of linguists is then to decide whether the linearly highly associated tags are belonging to the same constituents, or to highly associated but distinct constituents. That is, the grouping of tags indicated by the computer is further confirmed by linguists' knowledge about the syntactic constituency.

On the one hand, the advantage of incorporating linguists' knowledge about constituency is to eliminate the drawbacks of the automatic construction of PSG, so as to couple the syntactic model with traditional linguistic analyses.

On the other hand, the advantage of using LLR is manyfold. Although the strength of linear co-occurrence does not necessarily correspond to the membership of syntactic constituents, a list of co-occurring tags with their statistical LLR is extremely helpful for grammar writers.

Firstly, the list focuses grammar writers' attention on really occurring phenomena. Thus, the PSG constructed in this way will not result from abstract invention of examples, but from quantifiable facts in the real corpus.

Secondly, the list provides an overview of all the distributional phenomena involved before linguists start to write the PSG. The list of all co-occurring tags can prevent linguists from committing manual omissions or errors. The relevant statistical information equips linguists with a simple and objective measure. The values of LLR highlight the strongly associated tags,

[8] The distituent grammar in [2] contains only four rules of two tokens each. And these distituent rules do not remain accurate in every pass (or level) of construction.

providing a good set of candidates to form constituents. The values of probability (count) enable linguists to focus on phenomena which are statistically significant (i.e. with frequent occurrence).

Thirdly, when corpus are enlarged, the tag sequences and their LLR can be automatically reconstructed and compared with the old ones to show what new phenomena need to be handled in the PSG. If some modifications of rules should be made, the influences of modifications can be predicted from relevant statistical information of relevant tag sequences.

This approach, of course, may have some weak points similar to those of other corpus-based approaches. Firstly, the deduction power of the PSG will be poor with a small corpus. However, with the increasing availability of machine readable corpora, this kind of capability can be easily improved by enlarging the corpus. Moreover, if there are indeed well-known linguistic phenomena which fail to occur in the small corpus, it will be adequate for linguists to add the corresponding rules to the PSG in order to increase the descriptive power of the PSG in testing sets. Since the orginal set of constituents has been confirmed by linguists, the manual addition or modification of syntactic rules is easier to be accomplished.

Secondly, the manual category-tagging process is still too time-consuming. However, with the aid of computer tools, the tagging process can be more conveniently and systematically undertaken.[9] Besides, once the tagged corpus is constructed, many useful models can be trained from the same corpus.

IV. Incorporating Linguistic Knowledge and Statistical LLR

In this section, our proposed cooperative approach will be illustrated by a pilot experiment of constructing a Mandarin PSG. The actual procedures are demonstrated as follows:

4.1 Constructing a Tagset

Appropriately classifying the lexical items and constructing an adequate tagset are important tasks for the whole tagging process. However, owing to the brevity of this paper, we will not pursue this issue any further, but simply present our tagset in the Appendix as a reference.

4.2 Tagging the Corpus

The sample sentences of this experiment are selected from computer technical manuals. In order to retrieve syntactical LLR from this corpus, all the sentences in this corpus have to be preprocessed. A tag will be associated to each word, representing the category (part of speech) it belongs. The LLR will be computed from the tag sequences thus obtained.

---

[9] For example, the stochastic tagger proposed in [6] is an automatic tagger.

4.3 Bootstrapping

Because tagging the corpus is still a time-consuming task, we decided to start our pilot experiment with a relatively small database (2,000 sentences). In order to reduce the estimation error for sparse data, a statistical method, called "bootstrapping", is applied before LLR is computed.[10] The bootstrapping method calculates the statistics over much more samples of data created by resampling from the original database. Each sample is taken independently from the original sample in order to be fair or representative of the population. In this experiment, 20,000 sentences were randomly drawn with replacement from the original 2,000 sentences to form a new database. During the sampling process, each sentence has equal chance to be selected. The new bootstrapping sample (with total number of 20,000 sentences) serves as the database for LLR calculation.

4.4 Calculating LLR from the Corpus

After applying the bootstrapping technique, the LLR of tags is automatically calculated with three different window sizes. The window size parameter allows us to look at different scales. Enlarging the window size enables linguists to build constituents with more elements. However, for the sake of reliability, the larger the window size is, the larger the corpus must be. To be compromised with the size of our database, the window sizes we chose in this experiment are 2, 3, and 4.

The formula of bigram LLR has been presented in section II. Intuitively, the original bigram LLR measure can be regarded as a measure function for a hypothesis testing problem of two events. The probability in the numerator corresponds to the event that the observed (x, y) are generated by a random source in which x and y are generated as an atom. The probability in the denominator, on the other hand, corresponds to the event that (x, y) are generated by a random source in which the generation of x and y is independent. By the same argument, the general n-gram LLR measure can also be treated as a measure function of a hypothesis testing problem. The numerator corresponds to the hypothesis that the observed data $(x_1, x_2, \cdots, x_n)$ is generated by a source in which $(x_1, x_2, \cdots, x_n)$ is generated as an atom. And, the denominator corresponds to the hypothesis that $(x_1, x_2, \cdots, x_n)$ are generated by the other sources in which the sequence $x_1, x_2, ...x_n$ is generated in coincidence. The formulas with window size of 3 and 4 can thus be defined as follows:

$$\mathrm{LLR}_3(x, y, z) \equiv \log_2 \frac{P_D(x, y, z)}{P_I(x, y, z)}$$

[10] Readers are referred to [10] for a review of the nonparametric estimation of statistical errors.

$$\mathrm{LLR}_4(w,x,y,z) \equiv \log_2 \frac{P_D(w,x,y,z)}{P_I(w,x,y,z)}$$

where $P_D(x, y, z)$ is definided as the probability for x, y, z to occur jointly, and $P_I(x, y, z)$ is defined as the probability for x, y, z to occur by chance. That is:

$$P_D(x,y,z) \equiv P(x,y,z)$$

$$\begin{aligned} P_I(x,y,z) \equiv\ & P(x) \times P(y) \times P(z) \\ & + P(x) \times P(y,z) + P(x,y) \times P(z) \end{aligned}$$

Similarly, the formula for $P_D(w, x, y, z)$ and $P_I(w, x, y, z)$ are shown below:

$$P_D(w,x,y,z) \equiv P(w,x,y,z)$$

$$\begin{aligned} P_I(w,x,y,z) \equiv\ & P(w) \times P(x) \times P(y) \times P(z) \\ & + P(w) \times P(x,y,z) + P(w,x) \times P(y,z) \\ & + P(w,x,y) \times P(z) + P(w) \times P(x) \times P(y,z) \\ & + P(w) \times P(x,y) \times P(z) \\ & + P(w,x) \times P(y) \times P(z) \end{aligned}$$

We can interpret P_I as the chances that $(x_1, x_2, \cdots, x_n)$ is generated by sources which happen to be able to generate the n-gram by chance.

After computation, the number of patterns obtained with window size of 2, 3, and 4 is 451, 1893, and 4828, respectively.

4.5 Verification of the Relevance of the Groupings by Linguists

Once groups of tags have been attested with LLR, linguists will use their linguistic knowledge to decide whether these groups really form constituents or not. The information obtained in the bigram model is presented in two different forms. One is ranking the tag pairs containing the same first tag (T1) by the value of LLR, called Bigram LLR Form I. The other is ranking all the tag pairs by the value of LLR, called Bigram LLR Form II. For illustration, part of these tables are shown in Table 1 and Table 2.

T1	T2	T1_cnt	T2_cnt	T1-T2_cnt	P(T1)	P(T2)	P(T1, T2)	LLR(T1, T2)
d	cl	6299	7480	3221	0.0201	0.0239	0.01028	4.420385
d	q	6299	5844	863	0.0201	0.0187	0.00276	2.876390
d	vr	6299	270	15	0.0201	0.0009	0.00005	1.465989
d	nc	6299	72194	2110	0.0201	0.2305	0.00674	0.539350
d	a	6299	2390	12	0.0201	0.0076	0.00004	-2.001918
d	vi	6299	7171	34	0.0201	0.0229	0.00011	-2.084581
d	d	6299	6299	13	0.0201	0.0201	0.00004	-3.284553
d	adv	6299	16187	12	0.0201	0.0517	0.00004	-4.761671
d	vv	6299	13532	10	0.0201	0.0432	0.00003	-4.766245
d	vn	6299	9	9	0.0201	0.1131	0.00003	-6.307170

Table 1 A part of the Bigram LLR Form I
(Ranking tag pairs with the same first tag by the value of LLR)

T1	T2	T1_cnt	T2_cnt	T1-T2_cnt	P(T1)	P(T2)	P(T1, T2)	LLR(T1, T2)
q	cl	5844	7480	4241	0.0187	0.0239	0.01354	4.925447
vp	p	1275	12892	1275	0.0041	0.0412	0.00407	4.602633
vxnp	p	1488	12892	1469	0.0048	0.0412	0.00469	4.584093
d	cl	6299	7480	3221	0.0201	0.0239	0.01028	4.420385
vnv	np	1073	7403	479	0.0034	0.0236	0.00153	4.239375
,	cjs	16556	10964	6239	0.0529	0.0350	0.01992	3.428368
vv	vnv	13532	1073	407	0.0432	0.0034	0.00130	3.134185
np	vv	7403	13532	2417	0.0236	0.0432	0.00772	2.917841
a	ctm	2390	21415	1210	0.0076	0.0236	0.00386	2.888484
d	q	6299	5844	863	0.0201	0.0076	0.00276	2.876390

Table 2 Top ten tag patterns in Bigram LLR Form II
(Ranking all the tag pairs by the value of LLR)

Table 1 provides an overview of which tags may accompany which tags in the corpus, and equips linguists with associated statistical information. If necessary, linguists can make tradeoffs between the coverage of the grammar and the efficiency of the system by consulting the joint probabilities (or co-occurrence counts) of tag pairs. When the value of the joint probability is small, which means the real occurrences of the tag pair are few, it will be relatively adequate to ignore the distribution of the tag pair in order to reduce the complexity of the grammar and simplify the processing of the system. This table is also helpful for identifying errors in the tagged corpus or finding some important phenomena which have been overlooked by theoretical studies.

Table 2 can focus linguists' attention on strongly associated tag pairs which are more likely to be combined into constituents. To indicate tags with genuine association, patterns with LLR less than 1.0 are automatically discarded. Furthermore, because LLR becomes unreliable when the real occurrences are few, patterns with joint probabilities less than 0.0005 are also ignored.

Besides, according to linguists' intuition, certain constructions will more naturally be analyzed as tri-branching or quadri-branching instead of bi-branching. (e.g. the bi-transitive con-

struction). Thus, the trigram model (with window size of 3) and quadrigram model (with window size of 4) will serve as convenient guides for linguists to construct constituents with more than two members. Since the list of patterns obtained in trigram and quadrigram models is too long (1893 and 4828 respectively), thresholds are also set on LLR (1.0) and joint probability (0.0005). The number of patterns thus obtained is 78 and 60 for trigram and quadrigram models respectively. These patterns are also ranked by the value of LLR. Top ten tag patterns in the trigram model and the quadrigram model are shown in Table 3 and Table 4, respectively.[11] It is clear that many meaningful groupings do appear in the top of these tables.[12] So, these LLR tables provide valuable clues for linguists to form constituents and help making the analyses quicker and more accurate.

T1	T2	T3	T1-T2-T3_cnt	P(T1,T2,T3)	LLR(T1,T2,T3)
「	nc	」	5832	0.018619	4.582876
、	nc	、	877	0.002800	4.469508
」	,	cjs	1885	0.006018	3.232748
、	nc	cjw	656	0.002094	2.950891
p	nc	vxnp	726	0.002318	2.883788
d	q	cl	744	0.002375	2.774897
p	nc	loc	2202	0.007030	2.219192
adv	vp	p	389	0.001242	2.200802
np	vv	p	724	0.002311	2.082381
p	nc	vxn	240	0.000766	2.057487

Table 3 Top ten tag patterns in the Trigram LLR Table
(Ranking all the trigram tag patterns by the value of LLR)

T1	T2	T3	T4	T1-T2-T3-T4_cnt	P(T1,T2,T3,T4)	LLR(T1,T2,T3,T4)
、	nc	、	cjw	293	0.000935	3.485713
p	nc	vxnp	p	726	0.002318	2.462845
np	adv	vn	ctm	1105	0.003528	2.417586
vxnp	p	nc	loc	363	0.001159	2.413632
vv	p	nc	vxnp	284	0.000907	2.397562
「	nc	」	,	2086	0.006660	2.354226
q	cl	vi	ctm	412	0.001315	2.148030
」	,	cjs	vn	1353	0.004320	2.132438
vp	p	nc	loc	240	0.000766	2.055334
vn	「	nc	」	3352	0.010702	2.018944

Table 4 Top ten tag patterns in the Quadrigram LLR Table
(Ranking all the Quadrigram tag patterns by the value of LLR)

[11] The tag "、" stands for the Chinese punctuation mark "、", and the tags "「" and "」" stand for the Chinese qutation marks "「" and "」", respectively.

[12] For example, d-q-cl is a good candidate for forming a quantifier phrase.

After having checked over the tag patterns, linguists pick out groups which should be treated as constituents, and assign phrasal tags to them. A substitution tool will automatically replace all the relevant tag patterns with new tags, or automatically locate the relevant tag patterns for linguists to confirm the substitution. Then LLR is computed again with the newly changed corpus (with new phrasal tags), yielding new LLR tables. In this experiment, we firstly constructed the quantificational phrases (Q1, consisting of "(d) (q) (cl)"), the low level coordinate phrases, and substituted "{ nc }" with N0. Part of the resulting new tag patterns in different n-gram models are shown in Table 5, Table 6, and Table 7.

T1	T2	T1_cnt	T2_cnt	T1-T2_cnt	P(T1)	P(T2)	P(T1, T2)	LLR(T1, T2)
q	cl	281	417	271	0.0010	0.0015	0.00098	9.323793
d	cl	296	417	146	0.0011	0.0015	0.00053	8.356441
NJ	ctn	3894	205	141	0.0141	0.0007	0.00051	5.613008
vp	p	1275	12892	1275	0.0046	0.0465	0.00460	4.425786
vxnp	p	1488	12892	1469	0.0054	0.0465	0.00530	4.407246
vnv	np	1073	7403	479	0.0039	0.0267	0.00173	4.062527
vnp	p	259	12892	163	0.0009	0.0465	0.00059	3.757706
adv	vns	16209	222	175	0.0585	0.0008	0.00063	3.752262
,	cjs	16485	10945	6233	0.0595	0.0395	0.02249	3.258835
VN0	Q1	496	10467	148	0.0018	0.0378	0.00053	2.981671

Table 5 LEVEL II Top ten tag patterns in Bigram LLR Form II
(Ranking all the tag pairs by the value of LLR)

T1	T2	T3	T1-T2-T3_cnt	P(T1,T2,T3)	LLR(T1,T2,T3)
cjs	vv	VNJ	342	0.001234	3.253585
N0	,	cjs	1922	0.006936	3.062737
vp	p	N0	229	0.000826	2.838479
p	nc	vxnp	726	0.002620	2.829509
p	N0	loc	419	0.001512	2.738720
vv	vnv	np	230	0.000830	2.688979
,	adv	vp	206	0.000743	2.271391
p	nc	loc	2202	0.007947	2.178622
p	nc	vxn	240	0.000866	2.095507
vn	N0	,	2086	0.007528	2.085268

Table 6 LEVEL II Top ten tag patterns in the Trigram LLR Table
(Ranking all the trigram tag patterns by the value of LLR)

Linguists then check the new tag patterns to look for higher level constituents. This procedure is recursively applied until there is only one phrasal tag (S) left in every sentence. A complete PSG for this corpus is thus obtained.

When the size of the corpus is small, many constructions may not be included in this corpus. Thus, they will fail to appear in the LLR tables. However, if linguists are aware of their importance in the applicational domain, and there are indeed well-justified linguistic analyses

T1	T2	T3	T4	T1-T2-T3-T4_cnt	P(T1,T2,T3,T4)	LLR(T1,T2,T3,T4)
N0	,	cjs	VNJ	193	0.000697	3.424608
adv	vp	p	N0	149	0.000538	2.573180
p	Q1	nc	vxnp	141	0.000509	2.434070
p	nc	vxnp	p	726	0.002620	2.374187
vxnp	p	nc	loc	363	0.001310	2.335425
VNJ	nc	,	adv	232	0.000837	2.284488
vv	p	nc	vxnp	284	0.001025	2.257929
np	adv	vn	ctm	1105	0.003988	2.233253
vn	N0	,	cjs	1885	0.006803	2.184932
vp	p	nc	loc	240	0.000866	1.975087

Table 7 LEVEL II Top ten tag patterns in the Quadrigram LLR Table
(Ranking all the Quadrigram tag patterns by the value of LLR)

for them, it will be convenient for linguists to directly incorporate the existing analyses into the PSG. The descriptive power of the PSG for testing sets can be enlarged by incorporating linguistic knowledge in this way.

V. Conclusion

This paper discusses several methods of constructing a PSG, including the purely linguistic approach, the purely automatic approach, and the proposed human-machine cooperative approach. The advantages of the proposed approach over other methods are briefly summarized as follows:

1. The corpus-based statistical approach focuses linguists' attention on authentic material instead of invented examples.

2. The LLR tables equip linguists with an overview of the distributional phenomena involved, preventing linguists from committing manual errors or omissions.

3. The LLR statistic highlights the strongly associated tag sequences, providing a good set of candidates for forming constituents.

4. The statistical information provides an objective measure for linguists to make tradeoffs, and enables linguists to focus on phenomena of statistical importance rather than of theoretical interest.

5. When modifications are made, the tagged corpus and the relevant statistical information can be automatically and systematically reconstructed.

6. The syntactic model can be coupled with traditional semantic models.

7. The grammar is able to incorporate achievements of linguistic researches.

According to our experience in the pilot experiment, the LLR statistic really helps making the analyses quicker and more accurate. As most of us believe, human grammar writers could do a better job if they had access to better tools. LLR statistic is suggested in this paper as the right tool.

REFERENCES

[1]. Anick, P., and J. Pustejovsky, "An Application of Lexical Semantics to Knowledge Acquisition from Corpora," *Coling 90*, vol. 2, 7-12, 1990.

[2] Brill, Eric, David Magerman, Mitchell Marcus, and Beatrice Santorini, "Deducing Linguistic Structure from the Statistics of Large Corpora," *Proceedings of the Third DARPA Workshop on Speech and Natural Language*, U.S.A, 1990.

[3] Brown, Peter F., John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin, "A Statistical Approach to Machine Translation," *Computational Linguistics*, vol. 16, 79-85, 1990.

[4] Chao, Yuen-Ren, *A Grammar of Spoken Chinese*, Berkeley: University of California Press, 1968.

[5] Chinese Knowledge Information Processing (CKIP), *Kuo2 Yu3 De0 Tsi2 Lei4 Fen1 Shi1 (Shiou1 Ding4 Ban3)*, Nankang: Academia Sinica, 1988.

[6] Church, Kenneth, "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text," *Proceedings of Second Conference on Applied Natural Language Processing*, Austin, Texas, 1988.

[7] Church, Kenneth and Patrick Hanks, "Word Association Norms, Mutual Information, and Lexicography," *Proceedings of ACL-27*, Vancouver, 76-83, 1989.

[8] Church, Kenneth, William Gale, Patrick Hanks, and Donald Hindle, "Parsing, Word Associations, and Typical Predicate-Argument Relations," *Proceedings of the International Workshop on Parsing Technologies*, C.M.U, 1989.

[9] Church, Kenneth, William Gale, Patrick Hanks, and Donald Hindle, "Using Statistics in Lexical Analysis," in U.Zernik (ed.), *Lexical Acquisition: Exploiting On-Line Resources*, Hillsdale, N.J. Lawrence: Erlbaum Associates, 1990.

[10] Efron, Bradley and Gail Gong, "A Leisurely Look at the Bootstrap, the Jackknife, and Cross-Validation," *The American Statistician*, vol 37, No. 1, 36-48, 1983.

[11] Fu, King-Sun, "Syntactic Pattern Recognition and Applications," Englewood Cliffs, N.J.: Prentice-Hall, Inc, 1982.

[12] Garside, Roger, Geoffrey Leech and Geoffrey Sampson, *The Computational Analysis of English : A Corpus-Based Approach*, London: Longman, 1987.

[13] Hopcroft, John E. and Jeffrey D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, U.S.A.: Addison-Wesley, 1979.

[14] Hsu, Yu-Ling and Keh-Yih Su, "Criteria for the Classification of Lexical Categories in a Syntax-Oriented Parsing System," *Proceedings of ROCLING I*, 215-227, 1988.

[15] Hsu, Yu-Ling and Keh-Yih Su, "Lexical Categorization in a Syntax-Oriented Parsing System," In preparation.

[16] Su, Keh-Yih and Jing-Shin Chang, "Some Key Issues in Designing MT Systems," *Machine Translation 5*, 265-300, 1990.

[17] Tou, Julius T. and Rafael C. Gonzalez, *Pattern Recognition Principles*, U.S.A.: Addison-Wesley, 1974.

[18] Zernik, U. and P. Jacobs, "Tagging for Learning : Collecting Thematic Relations from Corpus", *Coling 90*, vol 1, 34-39, 1990.

APPENDIX

The tagset used in this experiment is listed below (punctuation marks are not included). Readers are referred to [4], [5], [14], and [15] for detailed discussions on lexical categorization.

****

nc : common nouns

np : proper names or pronouns

d : determiners

q : quantifiers

cl : classifiers

p : prepositions

loc : locatives

ref : reflexives

vi : intransitive verbs

vn : verbs followed by a single nominal argument

vnn : verbs followed by double nominal arguments

vs : verbs followed by a sentential argument

vv : verbs followed by a verbal argument

vp : verbs followed by a prepositional phrase argument

vr : verbs introducing an obligatory relative clause

vns : verbs followed by nominal and sentential arguments

vnv : verbs followed by nominal and verbal arguments

vnp : verbs followed by a nominal object and a prepositional clause argument

vxn : verbs preceded by a preposed nominal object

vxnn : verbs preceded by a preposed nominal object and followed by a second object

vxnp : verbs preceded by a preposed nominal object and followed by a prepositional phrase argument

vxnv : verbs preceded by a preposed nominal object and followed by a verbal object

vxns : verbs preceded by a preposed nominal object and followed by a sentential object

a : adjectives

asp : aspect markers

adv : adverbs

cjs : conjunctions for sentences

cjv : conjunctions for verb phrases

cjw : conjunctions for words or other phrases

ctm : modifier clitics

cts : sentential clitics

ctn : noun clitics

excl : exclamatives

****

DEVELOPMENT OF AN AUTOMATIC ENGLISH GRAMMAR DEBUGGER FOR CHINESE STUDENTS: A PROGRESS REPORT[1]

Hsien-Chin Liou, Hui-Li Hsu, Yong-Chang Huang, and Von-Wun Soo

National Tsing Hua University

Abstract

This paper reports the research about development of an automatic English grammar debugger (on a personal computer) for Chinese college students, based on an analysis of the errors they made in their compositions. The first stage of development is devoted to an error analysis of 125 writing samples and classification of the errors into 14 main types and 93 subtypes. To implement the grammar debugger we first built a small dictionary with 1402 word stems and necessary features, as well as a suffix processor which accommodates morpho-syntactic variants of each word stem. We then built up an ATN parser, which is equipped with correct phrase structure rules and error patterns. In addition, a set of disambiguating rules for multiple word categories was designed to eliminate the unlikely categories to increase the precision power of the parser. The current implementation enables detection of seven types of error for a text input and response of corresponding diagnostic feedback messages. Future research will be focused on refining the grammar debugger to detect more types of mistakes with more precision and on providing appropriate feedback messages to diagnose students' deficiency as well as operations for students to edit their errors.

Introduction

One of the reasons which make English writing classes formidable for language teachers in Taiwan is the seemingly endless correction task of many learners' grammatical mistakes such as subject-verb agreement and article usage. This experience motivated our

[1] The research project is sponsored by National Science Council (#NSC80-0301-H007-15) in Taiwan, Republic of China. We would like to acknowledge our research assistant's (Kuei-Ping Hsu) full-time dedication to the project.

use of computer software to alleviate teachers' burden. If a computer program can help detect or even correct grammatical mistakes in students' compositions, it will reduce the tiring part of revision process and leave more time for human teachers to work on higher-level re-writing tasks such as revision of contents, organization, or expressions.

For this purpose, we have tested to what extent a commercial software package, *Grammatik IV* [1] could help our students. It was found that only 14 percent (10 out of 70) of all the categories that *Grammatik IV* had detected are substantive grammatical errors which writing teachers are serious about. What is worse, the package misses some of the significant errors frequently made by students due to the package's limited capacity, and generates false positives and misleading messages such as those in the following brackets.

(1) *Having listening _ the teachers' word, I was not surprised at the poor score I got as I didn't do the question with caution.* [Passive voice: 'was surprised' Consider revising using active]

(2) *There <u>were</u> great man in the world whom I respected forever.* [The context of 'whom' indicates you may need to use 'who']

(3) *These occupy successively lower vanges on the scale of computer translation ambition.* [Usually 'these' should be followed by a plural noun.]

The failure in *Grammatik IV* is due to erroneous analysis of sentence structures (as in all the three above) or rigid conformity to rhetorical conventions (as in (1) and (2)). The disappointment with *Grammatik IV* motivated our research on the development of an automatic English grammar debugger which can detect the major mistakes unique in our students' compositions. Concurrent efforts such as Chen and Xu [2] have been initiated, which, as complementary to the present research, has much to be desired regarding their global design and achievements. For example, the error types their debugger handles are not based on corpus but on the researchers' intuition.

The Research

This project proceeds in two stages. Stage I is devoted to analysis of errors in students' compositions, categorization of error types, and formulation of computer

processable rule patterns based on the categorization. Stage II concentrates on implementation of the grammar debugger on a personal computer.

Stage I Error Analysis and Categorization

We have collected over 1000 hand-written compositions, the corpus of this project, written by our students mainly with engineering backgrounds. The average length of the essays is about 200 words. To facilitate future testing of the debugger, we have keyed in some 194 essays (hereafter referred to as the sample database); the rest of them will be keyed in by the time the project is finished. A textual analysis mainly for syntactic mistakes has been conducted on 125 essays from the corpus. We have found 1659 errors and used a database package, dBASE III Plus to manage the errors. We then classified the mistakes into 14 major types and 93 subtypes for all the data we have analyzed. The rest of the corpus will be analyzed by the end of this project to update or refine the categorization. To measure the gravity of the error types, we adopted two criteria: frequency of occurrence and levels of hindering comprehensibility. Frequency of occurrence is measured by dividing the number of a certain error type by 1659, the total number of errors. The results, descending distribution of frequency in both raw numbers and proportion for major types and subtypes, are shown in Appendix A. To obtain a measure for the second criterion, level of hindering comprehensibility, we asked two native speakers (associate professors in linguistics) to grade examples taken from each subtype on a scale of one to four (meaning *bad* to *very bad* for comprehension). Results from the two criteria were used to screen all the error types. Lastly, we chose those categories which had higher frequency, were perceived worse and more easily processed by the computer -- under mainly a syntactic approach, before we formulated the categories into rule patterns.

To make the error types processable by a computer, we have tried to formulate error patterns or represented the errors as explicitly as possible so that computer programs may recognize/detect them. Here, we use the subtypes under Verbs as examples to illustrate how error patterns and pattern matching rules were formulated. All the subtypes are listed in Table 1.

Table 1

Subtypes of Errors Under the Verb Category

V1 (be V; redundant be verb; double finite verbs)
People could contact with friends when they were lived away.
V2 (modal + past verb)
If you use it carefully, it could made many work for you.
V-sub (verb subcategorization errors)
They try their best to stop them happen_ again.
VT (wrong tense/aspect)
If the war happened, we can never live a good life.
VT-1 (verb tense disagreement between clauses)
If we were not interested in the basic research, then we will not go ahead any more.
VT-2 (tense disagreement in a compound)
...we must avoid hazardous by-product of science and utilized the good points of science.
VT-3 (tense disagreement at discourse level)
On holidays, I often went out of Taipei. I usually ride my motorcycle enjoying the speed of wind.
VT-4 (contracted form fails to show plural form)
It's rainy last weekend.
VF (wrong verb forms -- passive/progressive forms)
The classmates and the teacher are all keep in my mind.

We then pulled out all context fields of each error type from our database and examined how the errors were manifested. For instance, all the contexts of V1 errors are listed in Table 2.

Table 2

All Contexts of V1 Errors

Record#	context
29	... people could contract with their friends and daily when they were lived away.
68	... then many dangerous thing will be happened.
506	Scientists have done a lot of works which made our living pattern is different from those days.
639	... the earth would be die at last.
692	... although they are not necessary improve our material life directly.
782	Because the scient is progress too fast.
817	It is seem great for the results coming out from science.
833	Although science makes our lives more comfortable, is it all do good to us?
885	Science has occupied a part of our life, and we are enjoy the development and achievement that science bring to us.
911	... I was fortunately passed the entrance examination ....
964	All of them made the earth never be suitable to be lived.
1040	All my life was began to be contained in the textbooks....

Here we take three error types, V1, V2, and V-sub as examples to illustrate how error patterns were formulated, or what we designed as a solution if the pattern could not be represented in a formal way. First, the **V1** error pattern can be described as a be verb plus another non-be verb, which has a feature of [intransitive], or [transitive] followed by a noun phrase at the verb phrase level. The only exception is the error in Record number 506 which requires another pattern to describe: causative verb, make plus finite verb be. More explicit rules can be written as follows:

a' V[b] X - V[vi]
|_ V[vt] NP
b' V[c] X V[b]
(V[b]: be verbs; X: wildcard symbol; V[vi]: intransitive verbs; V[vt]: transitive verbs; NP: noun phrase; V[c]: causative verbs)
(Note: Tentatively X is defined as an arbitrary number of words.)

Likewise, the error pattern for **V2** can be described as:

modal V-ed/V-en
(read as a modal such as should, could followed by the past tense or past participial form of a verb).

V-sub concerns problems with verb subcategorization. Referring to categorization in the framework of generalized phrase structure grammar [3], we have classified verbs into 33 categories (see Appendix B for detail). Since we have found that it is impossible to formulate error patterns for the V-sub type, we have attempted to represent the correct patterns instead. The correct representation enables mapping of verb patterns of the erroneous input onto the correct representation. As we have not completed this part, it will not be discussed in the remaining part of this paper.

Stage II On-line Implementation

Before we explain the work in this stage, we would make a note: the current project does not deal with misspellings that spelling checkers in commercial word processing packages have achieved to a very satisfying extent. For this project, the implementation work is divided into three phases and programmed in C language. Phase one concerns preparation of a small machine readable dictionary. Phase two involves construction of the electronic grammar debugger itself. Phase three pertains to phrasing and delivery of feedback messages.

Phase I. A survey of literature indicates that there are several comprehensive machine readable dictionaries available such as Longman Dictionary of Contemporary English, Webster's Seventh Collegiate Dictionary, Collins Bilingual Dictionary, and Collins

Thesaurus [4, 5, 6]. As our learners have limited English vocabulary and the project is exploratory in nature, we decided to make a small dictionary on our own to meet the immediate needs. Our experiences with this small dictionary, however, will help selection of the crucial information and access methods when we adopt an electronic comprehensive dictionary in the future. For our own dictionary, first, a program was written to extract word types from our sample database and formed the core of our dictionary entries. There are currently 1402 entries in the dictionary. Proper nouns like Chang, Tsing Hua are tentatively listed in the dictionary alphabetically. Each of the words is attached with (a) its part-of-speech information and (b) necessary features. Note that we have selected only the more likely part-of-speech information which our learners use in their English writing; we have not encoded rare usage in our dictionary. Ideally we hope the selection and ordering of word categories reflect the frequency of occurrence of each word, yet this requires further research. This selective approach has the disadvantage of encountering more unknown words if a learner's essay happens to be of higher quality. However, the reason why we adopted the simplification strategy is to save the dictionary space and increase the parser's precision. A sample of the dictionary entries and their affiliated features is shown in Table 3.

Table 3

A Sample of Word Entries and Their Selected Features in the Dictionary

Noun: count/noncount; vowel/consonant in the initial phoneme (V/C)
Adjective: single/multiple syllable (S/M); V/C
Adverb: subcategories (8 classes); S/M; V/C
Verb: subcategories (33 classes)
Pronoun: singular/plural/both (S/P/B); person (1st, 2nd, 3rd); case (subject/object/possessive)
Determiner: S/P/B

The entries in our dictionary are mainly stems of words, or headwords. To accommodate suffix changes of word stems, we have designed a suffix processor as

suggested in the EPISTLE text critiquing system [7] by adopting the concept called a distributional lexicon [8]. The processor is equipped with information about (a) rules of changes concerning word categories (e.g. from verb to noun) or the inflectional features (e.g. from plural noun to singular noun), and (b) associated actions (e.g. omitting -s can reform a noun stem). By means of a search procedure to correlate rules and suffix changes between the variants and headwords, the suffix processor ensures that the dictionary can identify the following three types of morpho-syntactic variants of each corresponding headword built in the dictionary: (a) the inflectional suffixes such as -ing, -ed, -s (for both verbs and nouns), (b) the derivational suffixes such as -ly in happily (from happy), -ful in cheerful (from cheer), and (c) markers of comparative and superlative degrees, -er, -est (such as hotter, or fastest). In this way, our dictionary can cope with natural English texts without building all the derivations as respective entries in our dictionary. To increase the processing efficiency, we grouped the rules above so that when a word like getting is encountered, it is assigned to the -ing group. This can save the searching time among all the suffix rules.

To cope with irregular forms of verbs, we have designed a table which lists the root form, and irregular changes of verbs. If an irregular verb like began is found in this table, it is associated with the feature past tense for later processing and its root form begin. Then, the processing directs to our dictionary and attaches affiliated features of begin to began.

In addition, we plan to build up a phrase dictionary and a dictionary of common problematic words to cope with errors in, for instance, sentences (4) and (5).

(4) *The misuse of the science results to the terrible thing of the rest part of the earth.* (should be results in)

(5) *We know that science is effected to human life seriously.* (should be science affects human life seriously)

Whether these dictionaries are to be integrated into a parsing process (to be described shortly) or remain individual processors is to be explored, though Stock [9] suggests the

former being more profitable in their system.

Phase II. In phase two, a parser was built and augmented by pattern matching of error types in order to automatically detect grammatical mistakes in a text input. Most of the work has been completed, whereas the other has been planned or under way.

The error patterns obtained from the analysis in the above section were tentatively classified into eight levels of processing, based on ease of manipulation by the computer or linguistic analysis, if applicable. The classification will be revised as we analyze more students' essays, generalize more and finer error patterns, and encounter bottlenecks.

(I) matching strings: For instance, the mistake in (6) can be easily detected when we simply search for the words 'Although/Though' and 'but'.

(6) Although my high school years were full of pressure, but I still found my ways to relax myself.

(II) matching strings and sets: For instance, the mistake in (7) can be detected when we search for the words No matter and a set of question words such as when, where, who.

(7) No matter eating, clothing, living, and walking, we rely on science.

(III) using the suffix processor to cope with errors related to a certain category of words: The technique can, for example, handle the problem of pluralizing uncountable nouns. After failing to match the word informations as in (8) in our dictionary, the suffix processor (designed to extract a possible stem, or root form for a word) can be used to reform the stem information. Since the countability feature for information indicates that it is uncountable, we can detect the nature of its error: an uncountable noun should not have a plural form.

(8) We must depend on some instruments like radio, computer to receive informations.

(IV) incorporating information in the dictionary into string matching: For instance, the mistake in (9) can be detected by matching the word more and searching for part-of-speech information of the following word in the dictionary. During the latter process, the suffix processor is activated to attach the feature [simple] or [comparative] degree to the word. This corresponds to the error pattern, 'more' + comparative degree of

adjective/adverb, and the debugger can flag this mistake.

(9) The weather becomes more hotter than before.

(V) looking the problem up in a dictionary for common problematic words or phrases: As mentioned before, some of the students' mistakes are related to a specific word or phrase. This phenomenon will lead to construction of a specific dictionary with the hope of detecting such types of errors more effectively. In addition to problematic words, resolution techniques for detection will be built in the dictionary. This approach may help solve some of semantic problems which are not very meaning-dependent such as (10). With the help of parsing, the program can detect the mistake: misuse of an adjective for an adverb. With the special dictionary, the program enables specific diagnosis of a common error type, confusion between everyday and every day (because of very similar forms).

(10) A lot of people feel nervous everyday.

(VI) using syntactic parsing and pattern matching: This level will be explained in more detail shortly as it is the main mechanism by which the most of the implementation work has been accomplished.

(VII) using semantic processing: Most of the diction problems fall into this category. This will be a very challenging problem as the information conveyed in the essays of our corpus is not within a limited domain. We have not yet had a clear idea of how to cope with such problems.

(VIII) using discourse strategies: Some of the errors concerning the scope of discourse such as anaphora may be too complex to be resolved in this project; however, we will explore the possible directions for future study.

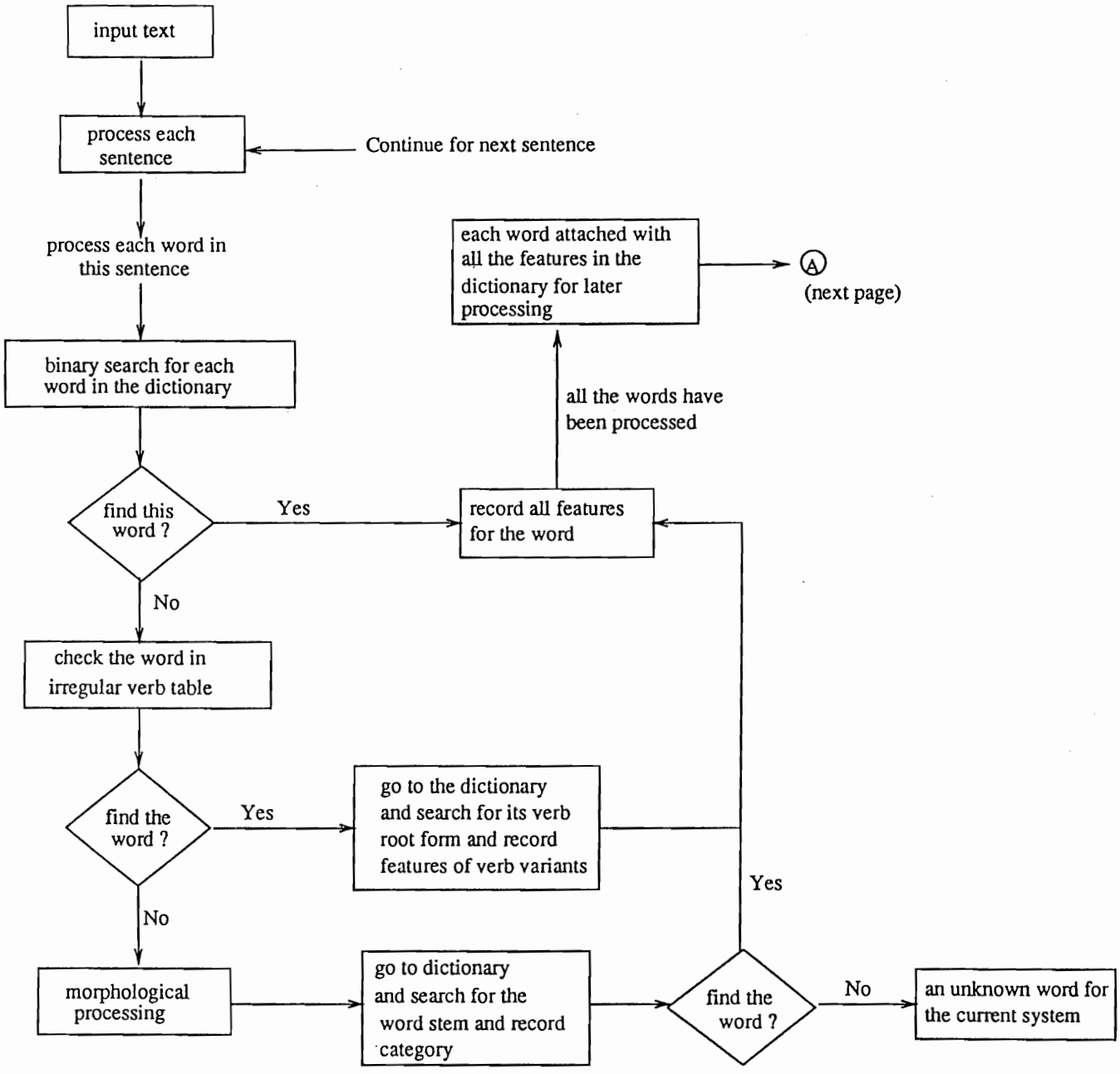
Pattern matching, as an efficient technique from the programming perspective, has been shown limited in developing grammar checkers [see 10, for example]. Thus, a syntactic parser is one of the ultimate solutions to natural language understanding/analysis. To structurally analyze the input text, a top-down parser has been constructed. It was formulated in the augmented transition network (ATN) grammar [11]. To increase its precision of analysis, a set of word category disambiguation (WCD) rules has been devised

to pre-process multiple word categories of some input words. The rules cut down the possibility of multiple word categories, and reduce the number of ambiguous sentence structures as well as processing time. For example, if a word has two categories, verb and adjective, and it is preceded by a determiner and followed by a noun, then the category, adjective is chosen such as falling in the falling rock. For the parser to be able to debug grammatical errors (besides judging whether the sentence is grammatical or not), two types of information have been encoded in the program: an expert model and a bug model. The expert model represents all the structural possibilities of correct sentences, whereas the bug model represents the error patterns we have formulated. For the expert model, a small segment of phrase structure rules by which we need to generate the structure of a correct sentence look like the following.

S -> NP VP
NP -> (Det) (AP) N ({PP, S'})
AP -> (Det) ("more") A {PP, S'}
VP -> V (NP) ({NP, PP})
PP -> P NP
S' -> Comp S
(S: sentence; NP: noun phrase; VP: verb phrase; Det: determiner; AP: adjective phrase; N: noun; S': embedded sentence; A: adjective; V: verb; PP: prepositional phrase; P: preposition; Comp: complementizer; ( ): optional symbol; {}: selectional symbol)

The bug model currently has three groups of error patterns: those manifested at noun phrase, verb phrase, and clause levels. Each of the groups is activated while the parser is analyzing/reconstructing its corresponding constituent. There are cases whose bug structure is unlikely to be represented, due, for instance, to its sporadic or idiosyncratic nature. In those cases, we map the expert model onto the input sentence and try to diagnose the nature of the problem by some devised heuristic. The dual-model mechanism is similar to a meta-rule concept described in Weischedel and Sondheimer [12].

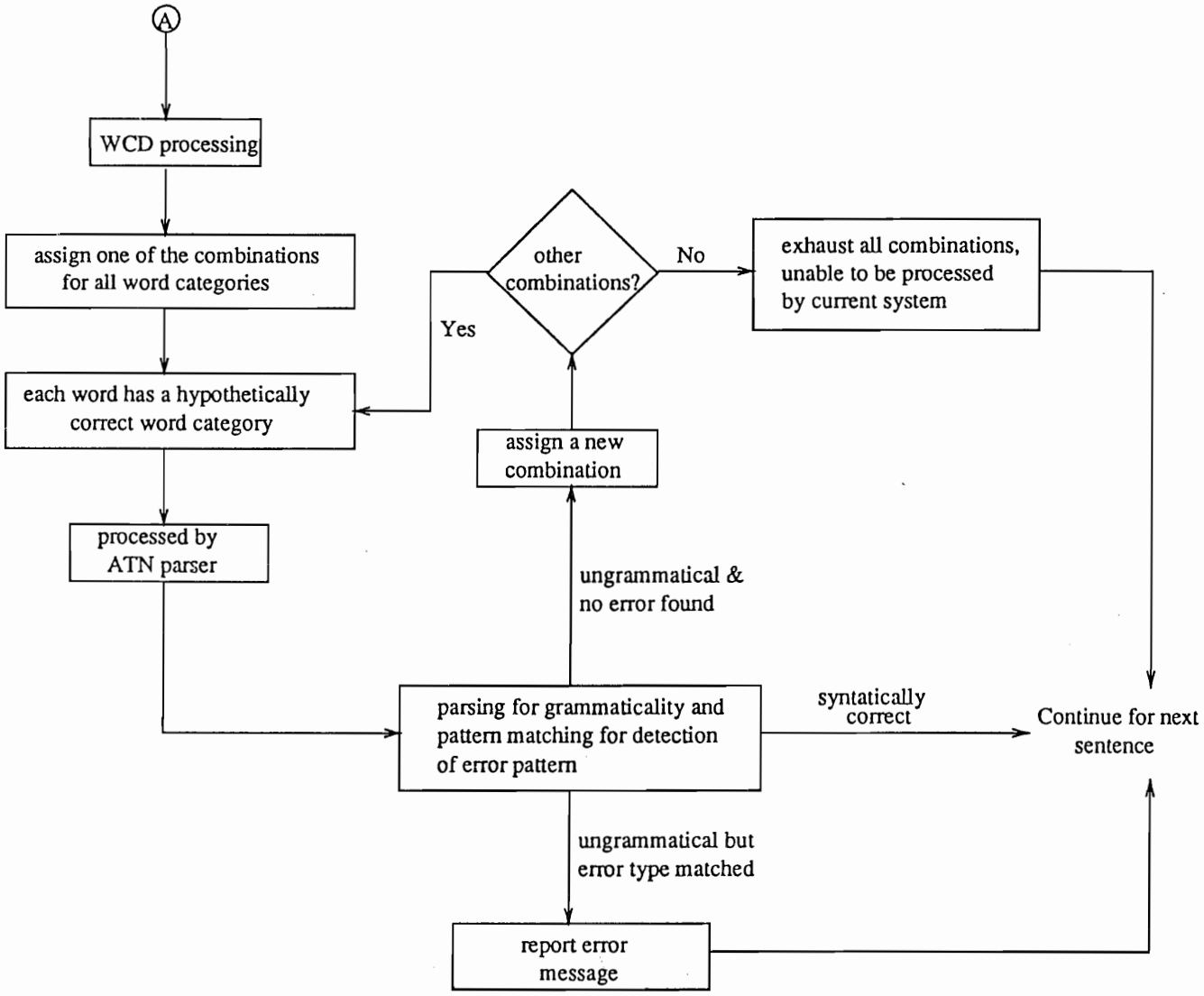
Figure 1 is a flow chart that demonstrates the procedures by which the grammar debugger processes each sentence and detects errors. The program allows regular English texts as its input and processes sentence by sentence. For each sentence, the program first uses the binary search algorithm to locate each word in the dictionary. If the program finds the word, it then records all associated features of this word. If the program fails, it proceeds to search for the word in the irregular verb table. If it finds the word, then the program goes to the dictionary to locate its root form and obtain features as well. If the program still can not find the word, it activates the suffix processor to do morphological processing. Notice that the category of a word before morphological processing is unknown and the word does not exist in the dictionary. After the word is processed by the suffix processor, it may be reformed and obtain its category information from this process. If the program still fails at this stage, the word is recognized as an unknown one for our current system. Up to this stage, each word, except unknown ones, is assigned its word category/categories and associated features. At the error detection level, i. e. after each word has been assigned categories, the program activates word category disambiguation (WCD) rules to cut down unlikely categories if a word has more than one category. After WCD processing, each sentence obtains a hypothetically correct combination of word categories to be processed by the parser. If the parser determines the sentence as grammatical, the program proceeds to the next sentence. If the sentence is determined as ungrammatical and detected by any of the error patterns, the program reports the error/feedback message and continues for the next sentence. If neither the parser nor pattern matching can determine the status of the input sentence, the sentence is assigned by another combination, if any, of word categories and the program repeats the parsing/pattern-matching processing. After the program exhausts all the possible combinations of word categories but still can not determine the status of the sentence (grammatical or ungrammatical) nor the nature of errors made, then the sentence is determined unable to be understood by the debugger/the current system.



(to be continued)

Figure 1. The flowchart of processing a sentence in the program.

(to continue)


A
WCD processing
assign one of the combinations for all word categories
each word has a hypothetically correct word category
processed by ATN parser
other combinations?
No
Yes
exhaust all combinations, unable to be processed by current system
assign a new combination
ungrammatical & no error found
parsing for grammaticality and pattern matching for detection of error pattern
syntatically correct
Continue for next sentence
ungrammatical but error type matched
report error message

The operation of the grammar debugger is basically an interaction between the parsing and the matching of error pattern processes. Each sentence is presumed to be ungrammatical, or erroneous. The program thus activates the pattern matching process first. As previously mentioned, the bug model has three groups of error patterns. Whenever a corresponding constituent in a sentence is built by the parser, that group of error patterns is tested to match whether the input sentence has any of the error patterns. For example, the debugging process of sentence (11) can be illustrated in the following trace, run by our current system.

(11) No matter _ he say_, he like_ these job_.

Table 4

An Output Trace

==========

Parse sentence : **No matter he say, he like these job.**

Searching in the dictionary ....
WORD : no
CATEGORY : <av>
WORD : matter
CATEGORY : <n v>
WORD : he
CATEGORY : <ppn>
WORD : say
CATEGORY : <v>
WORD : he
CATEGORY : <ppn>
WORD : like
CATEGORY : <v pp>
WORD : these
CATEGORY : <d pn>
WORD : job
CATEGORY : <n>

Using WCD-rules ....
WORD : no
CATEGORY : <av>

```
WORD     :  matter
CATEGORY :  <n v>
WORD     :  he
CATEGORY :  <ppn>
WORD     :  say
CATEGORY :  <v>
WORD     :  he
CATEGORY :  <ppn>
WORD     :  like
CATEGORY :  <v pp>
WORD     :  these
CATEGORY :  <d>
WORD     :  job
CATEGORY :  <n>

Assigning category ....
no <av> matter <n> he <ppn> say <v> he <ppn> like <v> these <d> job <n>

Syntax Error !!  --->  No matter .....
no matter ( ? ) he say, he like these job.

Syntax Error !!  --->  Number disagreement: determiner -- noun
no matter he say, he like ( these ) ( job ).

Syntax Error !!  --->  Subject-verb disagreement
no matter ( he ) ( say ), he like these job.
no matter he say, ( he ) ( like ) these job.

This is not a correct sentence.  There are four errors.
```

==========

First of all, pattern matching of clause level errors is activated. Error types such as although ... but or no matter are classified under the clause level errors. This sentence matches the error pattern of no matter, which is thus flagged. Since there is only one noun phrase (NP), these job, error types at the noun phrase level are attempted and found matched with the type determiner-noun disagreement. The correct noun phrase should be these jobs, so these job is flagged. Subject-verb (S-V) agreement is checked for each NP and VP (verb phrase) in each clause. The program first locates the head of each NP and

VP and returns the number values (singular or plural) of both. Then, a comparison process is made to see whether they agree. In the case above, two incidents of S-V disagreement are found. If none of the error types are matched in any of the constituents, the parser proceeds and determines whether this is grammatical under our current phrase structure representation.

Currently, our checker can locate the following seven types of errors:

(1) although ... but combination

(11) Although he is poor, but he is happy.

(2) erroneous usage of no matter

(12) People can produce many things, no matter bad or good.

(3) determiner-noun disagreement

(13) We can know many informations.

(14) This is a books.

(15) I like an book[2].

(4) unbalanced coordinated phrases

(16) He likes a dog but hate_ a cat.

(5) capitalization misuse

(17) There are not the exist of Television, computer, airplane, and so on.

(6) erroneous morphological changes in verb phrase, and

(18) I should went with you.

(7) subject-verb disagreement

(19) Human create_ the science.

(20) Human already have the ability to research the phenomena of space.

(21) But the development in science have bring great change.

(22) A man who like_ art like_ books.

---

[2] The initial phoneme of book is encoded in the dictionary.

Phase III. Phase three concerns what and how feedback messages should be given. When the program detects a grammatical error, giving appropriate feedback messages is essential for a grammar checker to achieve its educational goal. For this, we plan to design a message generating routine which basically matches a flag that is attached to each processing rule with a message file, and outputs the message to the users, possibly with some examples. The way we design the message is to use a template; namely, the message consists of some variables (as the underlined words in the following brackets) and literal texts (those in plain texts). For example, a feedback message for sentence (23) may look like that in the brackets.

(23) *The development in scientific technologies have bring great change.*

[development is the subject of the verb have. The subject is in 3rd person singular form. The following are 2 correct examples:

The baby in the living room watches television.

The lady who sits next to me teaches English.]

For technical terms, we consider using Chinese. In addition, the correction and feedback given should be set up with a user-friendly interface environment so that language teachers and learners will not encounter confusion -- which may seem reasonable or common to computer-literate people, though.

Future Research

As an exploratory but ambitious research study, the current project has its drawbacks to be improved. Since we are aiming to treat the errors manifested in natural English texts, the coverage of English grammar, of both correct and incorrect ones, is much wider than much of the previous research work. Thus, the error detection tasks are accomplished in an dissatisfying piecemeal manner. In the future, we, therefore, will try to formulate the global mechanism of the grammar debugger in a more generalized, from the linguistic perspective, framework. Possible directions we will refer to are those in Jensen, Heidorn, Miller, and Ravin [13], Kwasny and Sondheimer [14], Weischedel and Black [15], and

Weischedel and Sondheimer [see 12].

For the short-term goal, first, we will complete the analysis of the remaining corpus. Second, we will polish the programming tasks in detecting errors, as pattern matching is likely to fail for most of the error types and parsing is overloaded with problems in the long tradition of natural language processing. In addition, the grammar debugger's performance is still waiting to be tested. Last, we will consider at which point to give appropriate feedback messages and what actions allowed for the user to edit the mistake after the debugger detects an error.

References

[1] G. D. Price, *Grammatik IV: User's Guide,* Reference Software International, San Francisco, CA, 1989.

[2] S. Chen, L. Xu, "Grammar-Debugger: A Parser for Chinese EFL Learners," *CALICO Journal,* vol. 8, no. 2, pp. 63-75, 1990.

[3] G. Gazdar, E. Klein, G. Pullum, I. Sag, *Generalized Phrase Structure Grammar,* Harvard University Press, Cambridge, MA, 1985.

[4] B. Boguraev, T. Briscoe, "Large Lexicons for Natural Language Processing: Utilising the Grammar Coding System of LDOCE," *Computational Linguistics,* vol. 13, no. 3-4, pp. 203-218, 1987.

[5] B. Boguraev, T. Briscoe, Eds., *Computational Lexicography for Natural Language Processing,* Longman, London, 1989.

[6] R. J. Byrd, N. Calzolari, M. S. Chodorow, J. L. Klavans, M. S. Neff, O. A. Rizk, "Tools and Methods for Computational Lexicography," *Computational Linguistics,* vol. 13, no. 3-4, pp. 219-240, 1987.

[7] G. E. Heidorn, K. Jensen, L. A. Miller, R. J. Byrd, M. S. Chodorow, "The EPISTLE Text-Critiquing System," *IBM Systems Journal,* vol. 21, no. 3, pp. 305-326, 1982.

[8] A. Beale, "Towards a Distributional Lexicon," in *The Computational Analysis of English,* R. Garside, G. Leech, and G. Sampson, Eds., Longman, London, 1987, pp. 149-162.

[9] O. Stock, "Parsing with Flexibility, Dynamic Strategies, and Idioms in Mind," *Computational Linguistics,* vol. 15, no. 1, pp. 1-18, 1989.

[10] G. Hull, C. Ball, J. L. Fox, L. Levin, D. McCutchen, "Computer Detection of Errors in Natural Language Texts: Some Research on Pattern Matching," *Computers and the Humanities,* vol. 21, no. 2, pp. 103-118, 1987.

[11] W. A. Woods, "Transition Network Grammars for Natural Language Analysis," *Communications of the ACM,* vol. 13, no. 10, pp. 591-601, 1970.

[12] R. M. Weischedel, N. K. Sondheimer, "Meta-Rule as a Basis for Processing Ill-Formed Input," *American Journal of Computational Linguistics,* vol. 6, no. 3-4, pp. 161-177, 1983.

[13] K. Jensen, G. E. Heidorn, L. A. Miller, Y. Ravin, "Parsing Fitting and Prose Fixing: Getting a Hold on Ill-Formedness," *American Journal of Computational Linguistics,* vol. 9, no. 3-4, pp. 147-160, 1983.

[14] S. C. Kwasny, N. K. Sondheimer, "Relaxation Techniques for Parsing Grammatically Ill-Formed Input in Natural Language Understanding Systems," *American Journal of Computational Linguistics,* vol. 7, no. 2, pp. 99-108, 1981.

[15] R. M. Weischedel, J. E. Black, "Responding Intelligently to Unparsable Inputs," *American Journal of Computational Linguistics,* vol. 6, no. 2, pp. 97-109, 1989.

APPENDIX A

Descending Distribution of Errors in Main Types and Subtypes

MAIN TYPE	N	PER CENT
Det	326	19.65 %
Verb	231	13.92 %
Noun	178	10.73 %
PS	174	10.49 %
Concord	168	10.13 %
Sent	158	9.52 %
Prep	123	7.41 %
Lex	115	6.93 %
Conj	67	4.04 %
Mech	54	3.25 %
Adv	27	1.63 %
Adj	23	1.39 %
Pron	9	0.54 %
Aux	6	0.36 %
Total	1659	100.00 %

MAIN TYPE	SUBTYPE	N	PER CENT
Det	A-3	154	9.28 %
Noun	CN	129	7.78 %
Det	A-1	105	6.33 %
Lex	Dict	94	5.67 %
Prep	Prep-1	81	4.88 %
Concord	3S-1	75	4.52 %
Sent	Run-on	65	3.92 %
PS	PS-nadj	65	3.92 %
Verb	V-sub	59	3.56 %
Sent	Frag	57	3.44 %
Verb	VT-1	55	3.32 %
Conj	Conj-1	55	3.32 %
Verb	VT-3	51	3.07 %
Mech	Cap	49	2.95 %
Det	Det-a	49	2.95 %
PS	PS-adjn	41	2.47 %
Verb	VF	39	2.35 %
Concord	SV	39	2.35 %
Noun	UN	27	1.63 %
Adj	Comp-1	23	1.39 %
Prep	Prep-2	23	1.39 %
Concord	3S-4	21	1.27 %
Noun	NN	20	1.21 %
Prep	Prep-3	19	1.15 %
Concord	3S-5	15	0.90 %
PS	PS-nv	14	0.84 %
PS	PS-adjadv	12	0.72 %
Verb	V1	12	0.72 %
PS	PS-advadj	12	0.72 %
Det	A-2	9	0.54 %
Sent	E	9	0.54 %
PS	PS-vn	8	0.48 %
Concord	3S/paral	8	0.48 %

(to be continued)

(to continue)

Adv	ED	8	0.48 %
Sent	2S	8	0.48 %
Sent	Paral	8	0.48 %
Conj	NM	7	0.42 %
Verb	VT-2	7	0.42 %

MAIN TYPE	**SUBTYPE**	**N**	**PER CENT**
Pron	Pron-1	7	0.42 %
Adv	Adv-2	6	0.36 %
Concord	SP	5	0.30 %
Conj	AB	5	0.30 %
PS	PS-vadj	5	0.30 %
Adv	OS	5	0.30 %
Sent	Rel-1	5	0.30 %
Verb	V2	4	0.24 %
Aux	Aux-to	4	0.24 %
PS	PS-prepv	4	0.24 %
Det	Det-O	4	0.24 %
Lex	Dict-v	4	0.24 %
Lex	2V-1	4	0.24 %
Det	A-4	3	0.18 %
Adv	ASP	3	0.18 %
Mech	Ap	3	0.18 %
Lex	Dict-p	2	0.12 %
Verb	VT-4	2	0.12 %
Lex	Red	2	0.12 %
Sent	WH	2	0.12 %
PS	PS-adjv	2	0.12 %
Concord	3S-2	2	0.12 %
PS	PS-advconj	2	0.12 %
Adv	very/much	2	0.12 %
Verb	VT	2	0.12 %
Sent	Rel-3	2	0.12 %
Noun	One-N	2	0.12 %
Pron	anaf	2	0.12 %
Lex	SM	2	0.12 %
Concord	3S-3	2	0.12 %
Mech	Punct	2	0.12 %
PS	PS-conjprep	2	0.12 %
PS	PS-nadv	2	0.12 %
Lex	Sem-1	1	0.06 %
PS	PS-prepconj	1	0.06 %
PS	PS-N.PP	1	0.06 %
Lex	to/too	1	0.06 %
Lex	Dict-Es	1	0.06 %
Lex	A/E	1	0.06 %
Det	some/any	1	0.06 %
Det	Num-a	1	0.06 %
Concord	WS	1	0.06 %
Sent	Rel-2	1	0.06 %
PS	PS-infprep	1	0.06 %
PS	Red-Comp	1	0.06 %
Lex	PH	1	0.06 %
Sent	WHi	1	0.06 %
PS	N-adj	1	0.06 %

(to be continued)

(to continue)

Aux	Aux-2	1	0.06	%
Aux	Aux-1	1	0.06	%
Lex	Dict-mb	1	0.06	%
Lex	Dict-e	1	0.06	%
Adv	TA	1	0.06	%
Adv	SA	1	0.06	%
Adv	Adv-1	1	0.06	%
Total		1659	100.00	%

Appendix B

Verb Subcategorization

1	vp --> v	die
2	vp --> v np	love
3	vp --> v np pp[to]	give
4	vp --> v np pp[for]	buy
5	vp --> v np np	spare
6	vp --> v np pp[+loc]	put
7	vp --> v np s[fin]	persuade
8	vp --> v (pp[to]) s[fin] e.g. ... concede to the scientists that John has contact with the patient	concede
9	vp --> v s[bse] e.g. ... insisted (that) the job be given to John	insist
10	vp --> v (pp[of]) s[bse] e.g. ... require of them that they write a paper	require
11	vp --> v vp[inf] e.g. ... continue to be unhappy	tend
12	vp --> v vp[inf, +norm] e.g. I tried to leave.	try
13	vp --> v (pp[to]) vp[inf] e.g. ... seems (to us) to be unhappy	seem
14	vp --> v np vp[inf] e.g. ... believe John to be unhappy	believe
15	vp --> v np vp[inf +norm] e.g. ... persuade them to give themselves up	persuade
16	vp --> v (np) vp[inf +norm] e.g. ... promise Mary to do the homework	promise
17	vp[agr s] --> v np e.g. It bothered Li that Tom was chosen.	bother
18	vp[+it] --> v (pp[to]) s[fin] e.g. It seems (to us) that Mary is unhappy	seem
19	vp[agr np[there, PLU]] --> v np[PLU] e.g. There was a lion in the zoo. There were three wolves in the zoo.	be
20	vp --> v s[fin] e.g. Mary believes that it is true.	believe
21	vp --> v s[+Q] e.g. He inquired which way to go	inquire
22	vp --> v np s[+Q] e.g. Tell us why you did it.	tell
23	vp --> v pp[of]	approve

24 vp[+AUX] --> v vp[-AUX bse] do
e.g. I do like it be true.

25 vp --> v pp[to] pp[about] talk

26 vp --> v adj/n feel
e.g. I felt stupid/ a fool.

27 vp --> v np adj believe
e.g. They believe her guilty.

28 vp --> v np np consider
e.g. They consider this offer a big improvement.

29 vp --> v v+ing give up
e.g. She's given up smoking.

30 vp --> v np v+ing hear
e.g. They heard someone laughing.

31 vp --> v np v-ed want
e.g. I want this work finished by tomorrow.

32 vp --> v np vp[bse] make
e.g. ... make her cry ....

33 vp --> v np pp compare
e.g. ... compare it with a book.

Abbreviation

fin: finite concede to the scientists that Jone has rings

bse: bare infinitive
insisted (that) the job be given to John

inf: infinitive continue to be unhappy

+norm means the noun is not in it or there dummy form

Q: question marker Tell us why you did it

AUX: auxiliary

agr: agreement

plu: plural

TRAINING A RECURRENT NEURAL NETWORK TO PARSE SYNTACTICALLY AMBIGUOUS AND ILL–FORMED SENTENCES

Ssu–Liang Lin and Von–Wun Soo

Department of Computer Science

National Tsing–Hua University, Hsin–Chu, Taiwan, 30043.

ABSTRACT

We are investigating to what extent can neural networks learn to parse a natural language. In particular, we present a recurrent neural network architecture and the learning experiments used to train the neural network. We train the recurrent neural network using the extended error backpropagation method by giving a sequence of lexicons as input whose categories may be ambiguous (more than one category is possible). Instead of encoding the parse tree within the neural network, the correct phrasal links as well as the lexical categories are clamped at the output layer of the network at the training phase while lexical categories are being fed into the neural network. With phrasal links, however, a complete parse tree can be easily reconstructed. Our results indicate that with a few training examples, the neural network can parse not only syntactically ambiguous sentences but also some ill–formed sentences that it has never seen before.

1. Introduction

Parsing is an important step in natural language processing. The main function of parsing is to produce the structural relationships among lexicons from a given input sentence. Traditional syntactic parsing methods such as chart–parsing, WASP (Wait–And–See parser), ATN (Augmented Transition Network), etc. [1,7], were somewhat successful in parsing well–formed sentences. However, they all encountered the difficulty in dealing with high complexity of ambiguities and potentially ill–formed sentences. This can be due to the reason that traditional methods are so restricted in their accuracy constraints that they do not accept any noise in their input. Nevertheless, in natural language processing, ambiguity and ill–formness are ubiquitous and unavoidable. New parsing techniques seem to be desirable to overcome these problems. This motivates us to look for more flexible parsing models that can achieve high efficiency and adaptability.

Artificial neural networks have recently raised much attention in its capabilities of carrying out computation of parallel constraint satisfaction and learning[6,9,10]. From a problem solving standpoint, parsing can be viewed as a constraint satisfaction process which must reconcile with constraints coming from both data (bottom–up from lexicons) and models (top–down from syntactic grammars). Therefore, training a neural network to perform parsing can be viewed as a process of incrementally encoding the structural relationships between lexicons and grammar rules in the inter–connections of the neural network. In this paper, we propose a system called SPARK (Syntactic PArser with Recurrent neural networK) to show the process of training a recurrent neural network to parse a subset of natural language from a context–free grammar. In section 2, we briefly summarize previous neural network approaches dealing with the problems of natural language parsing. In section 3, we describe the architecture of SPARK and the extended error backpropagation method that it adopts to achieve learning. In section 4, we explain the learning experiments that were used to train SPARK in order to make it acquire syntactic parsing skills. In section 5, we show the performance of the trained neural network by testing several different cases and discuss their implications. In section 6, we give our conclusions, discuss the limitations of SPARK and future work.

2. Previous work

Fanty [3] proposed a connectionist model which used neural networks to parse an English sentence in terms of a sequence of syntactic categories. His approach is to embed all possible parse trees in the neural network by pre–encoding a huge number of all possible phrasal links (which he called matching units). His parsing model can only handle sentences with a fixed number of lexicons. The disadvantage of this model is that the number of interconnections can be quite large for even a simple grammar. Santos [11] proposed a system called PALS which used a seven–by–six matrix of cells to represent the partial structure of a parse tree. Each cell in the matrix consists of all possible phrasal nodes. PALS needs additional rule nodes to represent the linking relationships between constituents in two adjacent cells of the same column in the matrix. PALS used the idea of snapshots with a size of seven constituents to break a long sentence into several chuncks. The disadvantages of PALS, however, are two folds: (1) its learning ability is locally restricted which can be difficult when handling embedding clauses and (2) its built–in rules prohibit the possibility of rule acquisition. Giles [4] trained a second–order single–layer recurrent neural network to recognize the languages produced by a regular grammar and a pushdown automata. Although Giles's work is not related with parsing a language, using activation patterns of context units in a recurrent network to represent the transition states seems to reflect a similar situation encountered in learning language parsing where intermediate parsing statuses are often needed to be trained and retained for subsequent parsing processes. St. John and McClelland [12] trained recurrent neural networks to achieve semantic comprehension of a natural language. Their neural network consisted of two stages of processing: a Gestalt–pattern building process which accumulates the syntactic and lexical information of a given input sentence and a role–filling process which assigns semantic roles to the corresponding constituents based on the Gestalt–patterns. The Gestalt–patterns can achieve the expectation and prediction on the semantic roles for the incoming constituents at a certain parsing context and situation. Other works such as Cottrel [2], Jain et. al. [5], McClelland et. al.[8] and Wermter [14] also discussed several different proposals and techniques to apply neural networks to problems at differ-

ent levels of language parsing. We were motivated by previous work and decided to choose the recurrent neural network model to investigate the natural language parsing problems.

3. The Architecture of SPARK and The Extended Error Backpropagation Learning Method

SPARK's architecture consists of four units: the input units, output units, hidden units and the context units which are shown in Fig. 3.1. The input layer for this feedforward network consists of 8 category input units (CIU), i. e. "noun","pronoun", "aux", "verb–i", "verb–t", "det", "adj" and "prep". Each unit represents a syntactic category of the input lexicon. We assume that all lexicons in a sentence have been assigned to their corresponding categories before the sentence is given to the recurrent neural network. For example, the sentence "The young boy will go with her" would be converted to its categorial form "det adj noun aux verb–i prep pronoun". When a given input lexicon has more than one syntactic category, more than one unit can be activated. When the first word category, say "det", is read in, the weight of the det–CIU will be set to 0.7 and all weights of the rest of CIU's will be set to 0.2. When a syntactic ambiguous lexicon is read in, more than one CIU's will be set to 0.7 depending on the corresponding categories of the lexicon. The output layer of the recurrent neural network also has 8 category output units (COU) which correspond to CIU's

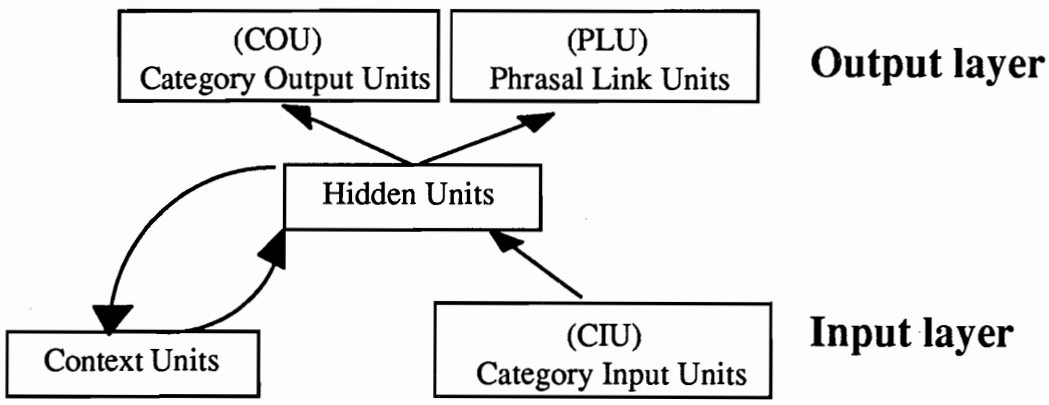


Fig. 3.1 The structure of the recurrent neural network

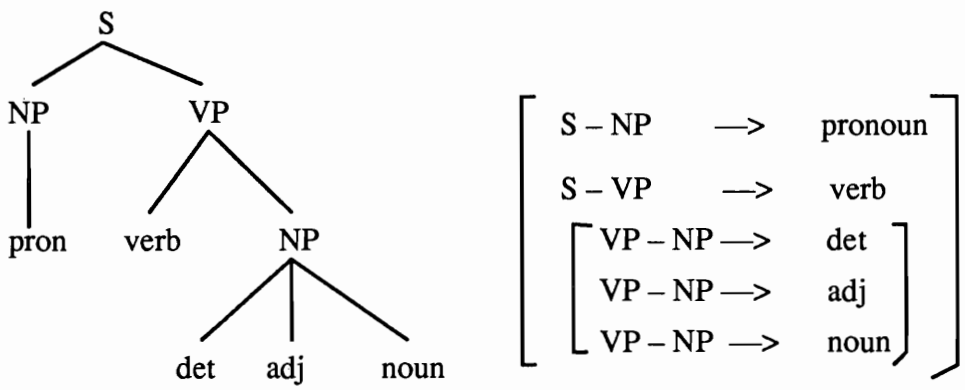


Fig. 3.2 The parsing tree and its corresponding phrasal link representation

as well as 6 phrasal link units (PLU), i. e., "S–NP" (S–N), "S–VP" (S–V), "VP–NP" (V–N),"VP–PP" (V–P),"PP–NP" (P–N) and "NP–PP" (N–P), each of which represents a piece of partial parse–tree information. An parse tree can be represented in terms of these phrasal links. For example, in Fig. 3.2, a parse tree of a sentence with five lexicons on the left can be represented by five phrasal links on the right and each of the link corresponds to a lexicon. The number of the hidden units and that of the context units are the same and can be varied.

The extended error backpropagation differs from conventional error backpropagation of Rumelhart in that there is a feedback connections from hidden units to context units [14]. This feedback mechanism in SPARK is to temporalily store the current parsing status in terms of activation patterns of hidden units into the context units so that the subsequent parsing step can take it into account. The learning algorithm for training SPARK can be derived by unfolding the temporal sequence of feedforward passes into a multi–layer feedforward network that grows one layer at each pass as shown at the right hand side of Fig. 3.3. To carry out the extended error backpropagation learning procedure, we let first the network run through the time interval $[t_0, t]$ and save all inputs, activation patterns of hidden units, and target vectors at each time step into a history buffer. Then the temporal error backpropagation according to the history proceeds. The process is described in terms of a set of equations that are defined in Fig. 3.3. First, we define the error generated over time as E(t) which

(1) $e_k(t) = d_k(t) - y_k(t) \quad k \in U$

(2) $E(t) = \frac{1}{2} \sum [\, e_k(t)]^2$

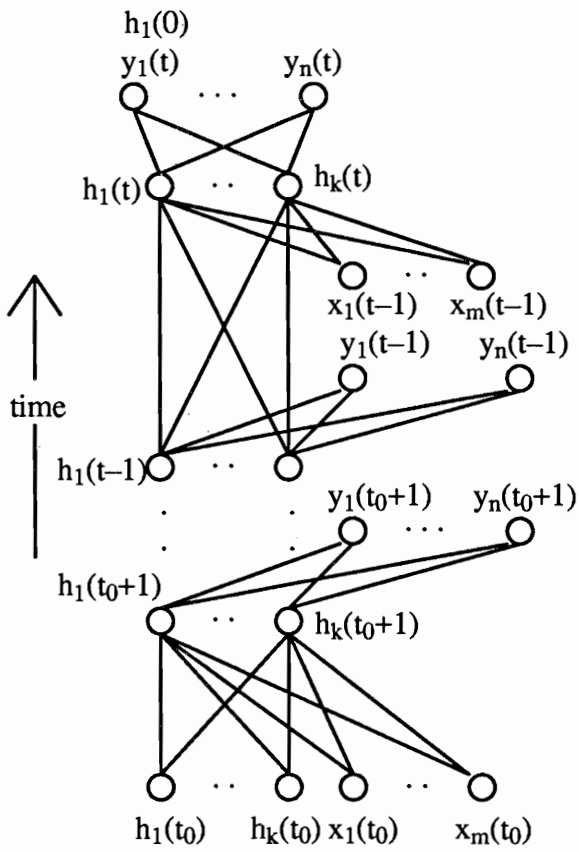
(3) $E^{total}(t_0,t) = \sum_{\tau = t_0+1}^{t} E(\tau)$

(4) $\nabla_w E^{total}(t_0,t) = \sum_{\tau = t_0+1} \nabla_w E(\tau)$

(5) $\Delta W_{ji} = -\eta \frac{\partial E^{total}(t_0,t)}{\partial W_{ji}}$

(6) $\delta_k(\tau) = f_k'[s_k(\tau)][\sum_{j \in U} W_{jk} \cdot \delta_j(\tau) + \sum_{l \in I} W_{kl} \cdot \delta_k(\tau+1)]$
$k \in H$

(7) $\Delta W_{ji} = -\eta \sum_{\tau = t_0+1}^{t} \delta_j(\tau) \cdot x_i(\tau - 1)$



$s_k(\tau)$: the net input for k–th hidden units at time τ

U : the index set for output units
I : the index set for external inputs
H : the index set of hidden units

Fig. 3.3 Equations used in the extended error backpropagation and the unfolded neural network architecture for a temporal sequence

is a sum of square of the difference between desired target $d_k(t)$ and output $y_k(t)$ [eq. (1) and (2)]. The learning goal of SPARK is to minimize the total error function $E^{total}(t_0, t)$ [eq. (3)]. The weight updating formula is computed by taking the negative gradient of $E^{total}(t_0, t)$ with respect to weights [eq. (4) & (5)]. Since the errors are propagated through the whole time sequence, the error at hidden layer must take into account both errors from the output layer at current time step and those from subsequent step [eq. (6)]. Once the temporal error backpropagation has been propagated to the time t_0+1, the actual connection weights can then be updated by the generalized delta rule [eq. (7)].

4. The Training Experiments

In this section, we show how to train a recurrent neural network to acquire the parsing skills given a set of training sentences. We create training sentences according to a set of context–free grammar rules as shown in Table 4–1. Note that the (adj)* represents that the number of adjectives can vary from zero to several while (aux) represents that an auxiliary is optional.

Table 4.1 A set of phrase structure rules

S	←	NP VP			
NP	←	det (adj)* noun PP	VP	←	(aux) verb–t NP
NP	←	pronoun	VP	←	(aux) verb–t NP PP
NP	←	det (adj)* noun	VP	←	(aux) verb–i PP

We tentatively prepare two training sets S1 and S2. S1 consists of 16 sentences shown in Table 4.2 whose lexicon categories are all uniquely assigned while the number of the input lexicons ranges from 2 to 11. Training set S2 includes S1 and has additional 9 sentences shown in Table 4.3 whose lexical categories can be ambiguous. For example, "n/v" represents a word whose category can be either a noun, an intransitive verb (vi) or a transitive verb (vt). In the second columns of Table 4.2 and Table 4.3, the corresponding partial parse–tree information in terms of phrasal links for each sentence is also shown.

Table 4.2 Training set 1

Training sentences with lexicon categories	partial parse tree information (phrasal links)
1. det noun vi *The child laughed.*	S–N S–N S–V
2. det noun vt det noun *The girl saw every movie.*	S–N S–N S–V V–N V–N
3. det noun vi prep det noun *The baby cried in the bedroom.*	S–N S–N S–V V–P P–N P–N

4. det noun prep det noun vi *The stranger with a hat disappeared.*	S–N S–N N–P P–N S–V
5. det noun prep det noun vt det noun *The girl with the unbrella broke her leg.*	S–N S–N N–P P–N P–N S–V V–N V–N
6. det noun prep pron vt det noun prep det noun *The man over there drank some wine in the afternoon.*	S–N S–N N–P P–N S–V V–N V–N V–P P–N P–N
7. pron vi *He succeeded.*	S–N S–V
8. pron vt pron *I like you,*	S–N S–V V–N
9. det adj noun vt det adj noun *A young girl found this little cat.*	S–N S–N S–N S–V V–N V–N V–N
10. det adj noun prep det adj noun vt det adj noun *The old man with the wooden stick discovered a new life.*	S–N S–N S–N N–P P–N P–N P–N S–V V–N V–N V–N
11. pron vt pron prep pron *He saw her with me.*	S–N S–V V–N V–P P–N
12. pron vi prep pron *He sat over there.*	S–N S–V V–P P–N
13. det noun prep pron aux vi *The clerk over there will manage.*	S–N S–N N–P P–N S–V S–V
14. det noun aux vi *Their dog won't bite.*	S–N S–N S–V S–V
15. det noun aux vt det noun *The students must read this textbook.*	S–N S–N S–V S–V V–N V–N
16. det noun prep det noun aux vt det noun *The students in the classroom must took an examination.*	S–N S–N N–P P–N S–V S–V V–N V–N

In Table 4.4 and 4.5, we show the effects of learning speed (in terms of epochs) versus various number of hidden/context units. It can be seen that when the number of hidden units reach around 30's, the efficiency of learning seems to become stable. Since the number of the hidden/context units might slightly affect the performance, for comparing the testing results in this paper, however, we kept the number at 10.

Table 4.3 Additional sentences with ambiguous lexicon categories for Training set 2

Training sentences with lexicon categories	partial parse tree information (phrase links)
1. det n/v vi *The program halted.*	S–N S–N S–V
2. det noun vi/n *The animals escaped.*	S–N S–N S–V
3. det noun vt det n/v *The boy caught one fish.*	S–N S–N S–V V–N V–N
4. det noun vt/n det noun *The tanks attacked the city.*	S–N S–N S–V V–N V–N
5. det noun vi/n prep det noun *His wife worked in the company.*	S–N S–N S–V V–P P–N P–N
6. det noun prep det n/v vi *The book with no cover disappeared.*	S–N S–N N–P P–N P–N S–V
7. pron vi/n *He danced.*	S–N S–V
8. pron vt/n pron *She helped me.*	S–N S–V S–N
9. det adj/v noun vt det adj n/v *Her close friend told a funny joke.*	S–N S–N S–N S–V V–N V–N V–N
10. det n/v aux vt/n det n/v *This report may influence his score.*	S–N S–N S–V S–V V–N V–N

Table 4.4 The convergence rates in terms of number of training epochs vs number of hidden units for Training set S1. The threshold is set at 0.7 for total sum of square error.

No.of hidden units	**8**	**10**	**15**	**20**	**30**	**40**	**50**
Epochs	**>3000**	**575**	**187**	**84**	**62**	**56**	**56**

Table 4.5 The convergence rates in terms of number of training epochs vs number of hidden units for Training set S2. The threshold is set at 1.0 for total sum of square error.

No.of hidden units	**10**	**15**	**20**	**30**	**40**	**50**	**60**
Epochs	**814**	**158**	**117**	**86**	**79**	**78**	**56**

5. Testing Results

After training, we used several testing sentences to evaluate the performance of SPARK.We found that SPARK could successfully parse many sentences with ambiguous categories. In particular, SPARK can tolerate syntactic ill–formed sentences and can produce a plausible parsing structure to account for a given input sentence. Since it is impossible to explore all possible legal and illegal sentences to evaluate the performance of SPARK, we show only a few testing cases to explain how SPARK performs parsing. This will be discussed from three different aspects in (A), (B) and (C) respectively.

(A) Testing Results Using New and Syntactic Ambiguous Sentences

SPARK can parse successfully those sentences with ambiguous categories which it has never seen before. Although SPARK can handle many sentences of this kind, we only illustrate one example in detail. In Fig. 5. 1, we show the activation patterns which were taken from run–time execution results for sentence "det adj/v n/v aux vi prep det n/v".

We found that the ambiguous categories for three lexicons were assigned correctly and the corresponding parse tree directly constructed from the PLU patterns shown in Fig. 5.2. Similarly, for the

	COU								PLU					
	n	pn	ax	vi	vt	d	a	p	S–N	S–V	V–N	V–P	P–N	N–P
1	.	.	.	.	.	#	.	.	#	.	.	.	.	.
2	.	.	.	.	.	.	#	.	#	.	.	.	.	.
3	#	.	.	.	.	.	.	.	#	.	.	.	.	.
4	.	.	#	.	.	.	.	.	.	#	.	.	.	.
5	.	.	.	#	.	.	.	.	.	#	.	.	.	.
6	.	.	.	.	.	.	.	#	.	.	.	#	.	.
7	.	.	.	.	.	#	.	.	.	.	.	.	#	.
8	#	.	.	.	.	.	.	.	.	.	.	.	#	.

Fig. 5.1 The activation patterns for the sentence "det adj/v n/v aux vi prep det n/v", the # represents the activation values ≥ 0.8 and the . represents values ≤ 0.2.

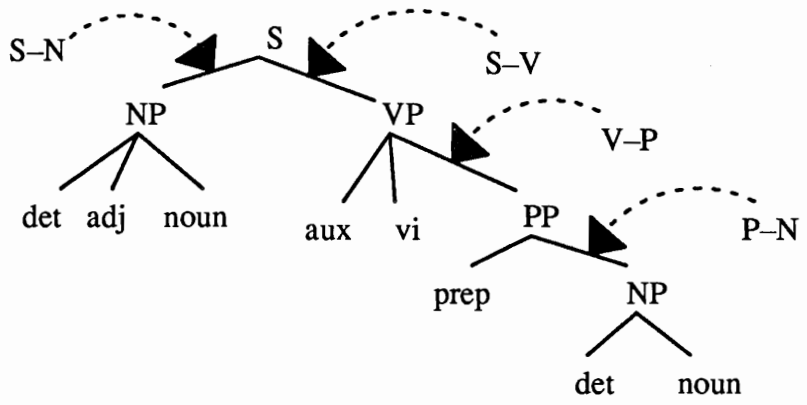


Fig. 5.2 The corresponding parse tree for the sentence "det adj/v n/v aux vi prep det n/v". The dashed arrows point to the corresponding phrasal links.

sentence with more categorial ambiguities like "det adj/v n/v vt/n det adj/v n/v", SPARK also produced the correct category assignment "det adj noun vt det adj noun" as well as a correct parsing tree. For a longer sentence such as "det adj/v adj adj n/v prep det adj adj noun vi prep det adj noun" SPARK also performed well and produced "det adj adj adj n prep det adj adj noun vi prep det adj noun" as a result as we desired.

(B) Testing Results Using Sentences with Syntactic Noises

In this experiment, we show that SPARK can tolerate some syntactic ill–formed sentences. In Fig. 5.3, we illustrate three sentences and their activation patterns of the output units generated by SPARK. The first sentence is a well–formed sentence with respect to the context–free grammar in Table 4.1, while the second and the third sentences are ill–formed. In the second sentence, a noun follows a determiner is tentatively omitted, SPARK can still produce the plausible parse tree with an expectation for a noun after the determiner. In the third sentence, both a noun and a preposition are omitted, the most plausible parse tree shows an expectation of a noun after the determiner, however with a erroneous preprositional link which shows up as indicated in the right bottom parse tree in Fig. 5.3.

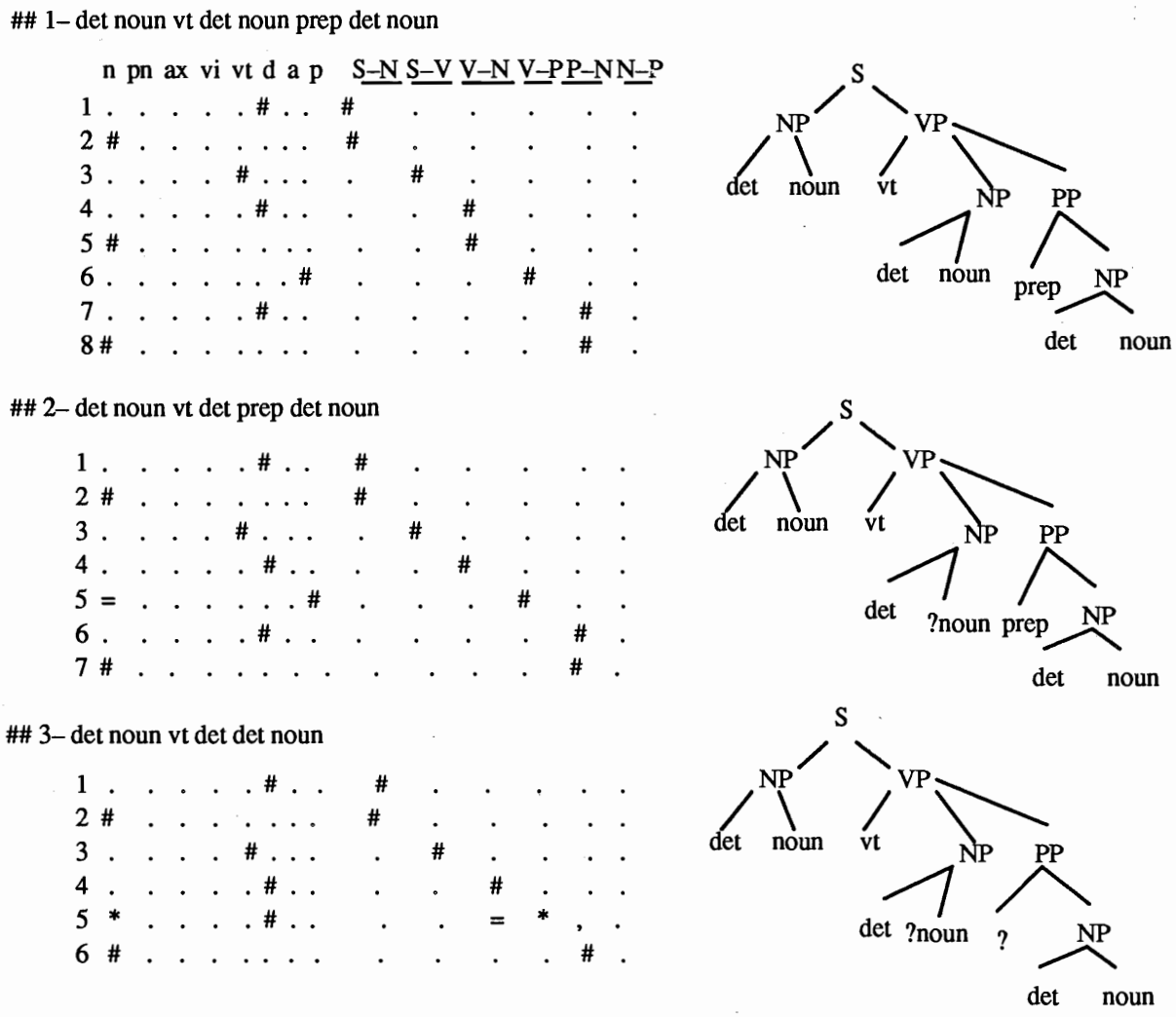


Fig. 5.3 ## 2 and ## 3 are examples of syntactic ill–formed sentences derived from sentence ## 1.
The notations for "#", "*" , "=" , "·" , and "·" represent ranges of activation values [0.8,1], [0.6, 0.8), [0.4, 0.6), [0.2, 0.4), and [0, 0.2) respectively. On the right hand side, the corresponding parse trees are shown. The ? in the parse tree represents the expectation of a constituent reflected in the activation patterns of COU.

(C) The Limitations and Problematic Cases

Of course, when there is too much noise involved, SPARK might fail too. However, we could argue that for certain bad data even human experts might be confused too. SPARK would maintain those that were successfully parsed and only fail at places where troubles got in. Here we illustrate a case to explain how SPARK might fail. For example, the sentence illustrated in Fig. 5.4 has "n/v" ambiguities at three places. As we can see in Fig. 5.4, SPARK handled well on the first entry of

15 det n/v n/v prep det n/v

	n	pn	ax	vi	vt	d	a	p	S–N	S–V	V–N	V–P	P–N	N–P
1	.	.	.	.	.	#	.	.	#	.	.	.	.	.
2	#	.	.	.	.	.	.	.	#	.	.	.	.	.
3	.	.	.	*	#	.	.	.	.	#	.	.	.	.
4	.	.	.	.	.	.	.	#	.	.	,	.	.	.
5	.	.	.	.	.	#	.	.	.	.	=	.	*	.
6	#	.	.	.	.	.	.	.	.	.	,	.	*	.

Fig. 5.4 A case where SPARK cannot produce a complete parse tree.

"n/v" (row 2)which it predicted as a noun. However, SPARK hesitated as for whether the second entry of "n/v" (row 3) was a transitive or an intransitive verb. This influenced the next entry when a lexical category "prep" was entered (row 4), SPARK got loss and could not predict any plausible phrasal link (no high enough activation values for PLU in row 4).

6. Discussions and Conlcusions

We have demonstrated a way of using recurrent neural network to perform syntatic parsing. Although we cannot claim that current SPARK can outperform traditional parsing methods, we do show the potentials of the approach. The advantages of SPARK are its ability to cope with ambiguities and ill–formness and its ability of learning (without explicitly specifying the grammar rules). Using only a few training sentences, we have obtained a plausible parser to parse many sentences that are generated by a context–free grammar. There are several extensions that can further enhance SPARK to become a truly natural language parser. First, the classification of categories and the phrasal links can be further elaborated. Second, semantic features or case roles can be included in the training in order to resolve those ambiguities such as a prepositional phrase attachment problem.

Reference

[1] Allen, James (1987). Natural Language Understanding, Benjamin/Cunmmings.

[2] Cottrel, G. W. (1989). A Connectionist Approach to Word Sense Disambiguation, Morgan Kaufmann Publishers.

[3] Fanty, Mark (1985). Context Free Parsing in Connectionist Networks, Technical Report 96, Computer Science Department, University of Rochester.

[4] Giles, C. L., Sun, G. Z., Chen, H. H., Lee, Y. C. & Chen, D. (1990). Higher Order Recurrent Networks & Grammatical Inference, In D.S. Tourestzky (ed), Advances in Neural Information Systems 2, Morgan Kaufmann, pp. 381–387.

[5] Jain, A. N. & Waibel, A H. (1990). Incremental Parsing by Modular Recurrent Connectionist Networks, In Tourestzky, D.S. (ed), Advances in Neural Information Systems 2, Morgan Kaufmann, pp. 364–371.

[6] Khanna, Tarun (1990). Foundations of Neural Networks, Addison–Wesley.

[7] Marcus, Mitchell P. (1980). A Theory of Syntactic Recognition for Natural Language, MIT Press.

[8] McClelland, J. L. & Kawamoto, A. H. (1986). Mechanisms of Sentence Processing: Assigning Roles to Constituents, In J.L. McClelland, D.E. Rumelhart & the PDP Research Group (eds), Parallel Distributed Processing: Explorations in the Microstructure of Cognition vol–II: Applications, MIT Press, pp. 272–325.

[9] Pao, Yoh–Han (1989). Adaptive Pattern Recognition and Neural Networks, Addison–Wesley.

[10] Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986). Learning Internal Representations by Error Propagation, In J.L. McClelland, D.E. Rumelhart & the PDP Research Group (eds), Parallel distributed processing: Explorations in the microstructure of cognition 1: Foundations , MIT Press.

[11] Santos Jr, E. (1989). A Massively Parallel Self–Tuning Context–Free Parser, In D.S. Tourestzky (ed), Advances in Neural Information Systems 1,Morgan Kaufmann, pp. 537–544.

[12] St. John, M. F. & McClelland, J. L. (1990). Learning and Applying Contextual Constraints in Sentence Comprehension, Artificial Intelligence 46, pp. 217–257.

[13] Wermter, S. (1989). Integration of Semantic and Syntactic Constraints for Structural Noun Phrase Disambiguation, Proceeding of the 11th IJCAI, vol. 2, pp. 1486–1491.

[14] Williams, R. J. and Peng, I. (1990). An Efficient Gradient–based Algorithm for On–line Train ing of Recurrent Network Trajectories, Neural Computation 1, pp. 490–501.