

格框在機器翻譯的應用

黃金仁 王良志

工業技術研究院電子工業研究所

一、前言

工研院電子所於1986年7月開始一個英譯中機器翻譯系統的計畫。經過2年的發展，目前(1988年7月)此系統在英中轉換、中文合成部份引用了格框(case frame)的作法，期能對句意有更深一層的瞭解，從而產生更佳的翻譯品質。本文主要將討論格框在此翻譯系統中的使用情形，包括：格位分析、英中轉換、中文合成，並將提出實際應用的心得，及探討格框做法對機器翻譯的益處。

此系統採取轉換式的作法，最基本就是對語法結構轉換，然而引用格框作法到底有些什麼好處呢？請先看一個例子：

例. You can avoid loading two separate disks to perform either of these tasks by using an AUTOEXEC.BAT file.

[譯]：你可以用一個AUTOEXEC.BAT檔，避免載入2片不同磁碟，來執行其中一項工作。 [註 1]

如果可以分析出格框：

(can) VP 施事A (You)
 avoid 結果RST (loading....) [註 2]
 目的p (to perform....)
 方法mns (by using....)

A : Agent 施事
RST : Result 結果
p : purpose 目的
mns : means 方法

又有中文規則： A aux mns * RST p
 施事 助動詞 方法 動詞 結果 目的
 你 可以 用.. 避免 載入.. 來執行..
則可得出上述譯文。

也就是說，英文順序為： 結果 目的 方法

而漢語一般習慣卻採用： 方法 結果 目的

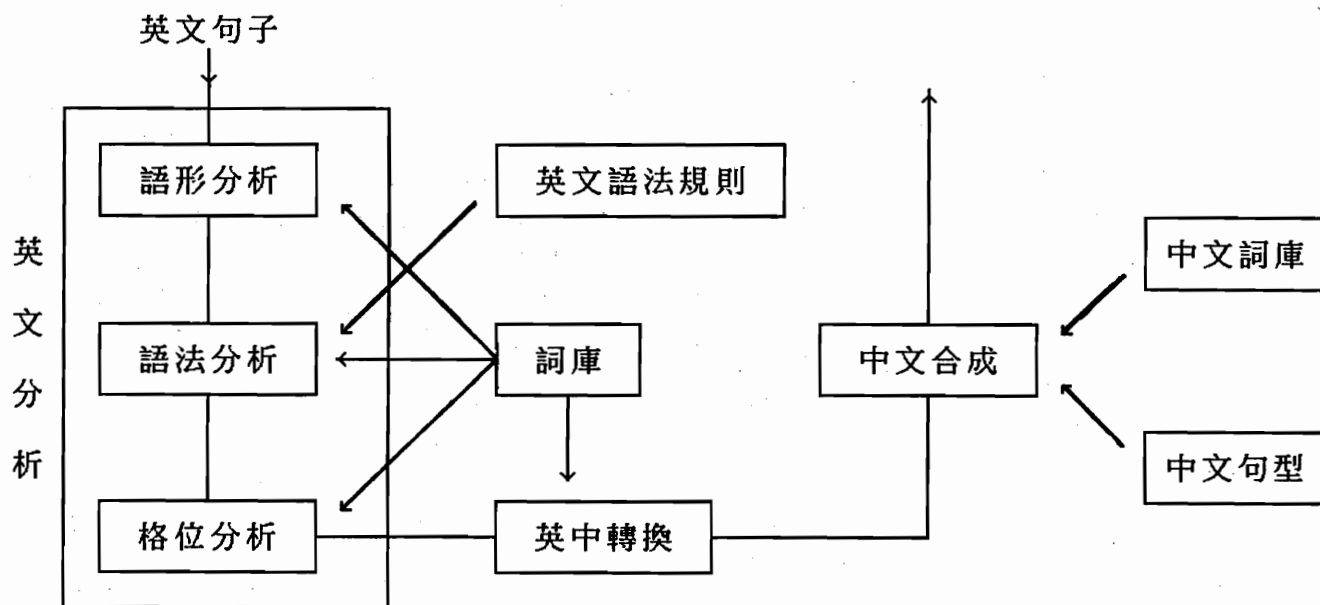
[註 1] 本文中 [譯] 均表示電子所機器翻譯系統的輸出結果。

[註 2] 格框中大寫英文表內格位(inner role)，小寫英文表外格位(outer role)。

如果比單純依靠語法結構，對於句子中各成份的意義，能夠更深入瞭解一層，分析出足以反映句意的格位(case role)，轉換、合成部份又有適當的規則可循，就可以得出較好的翻譯品質。

二、模組介紹

此機器翻譯系統之模組安排如下：



其中英文語法規則、詞庫、中文詞庫、中文句型都是知識庫(Knowledge base)，分別支援箭頭所指的模組。詞庫為記在英文word之下的知識，給分析與轉換部份用。每項詞目(entry)記載詞類、意義。

例. provide (詞類) CS V
 (意義) mng 0 1
 (格框知識) cf
 (預設中文) Chi 提供 規定
 (轉換規則) tr

cf記載該項mng所用的格框知識，詳見第三節。tr記載英中轉換模組選詞所用的法則，在不同語境，這個英文word的此項mng可能有意義相近的不同譯法(詳見第五節)，如果tr沒有成立或tr沒記，就用chi欄所記的中文。

例. Call-V CS V
 mng 0 1 2
 cf
 chi 稱呼 呼叫 打電話
 tr....稱為

例. Chinese CS N
 mng 0 1
 sm (lang) (huma)
 chi 中文 中國人
 sm為語意標記(semantic marker)

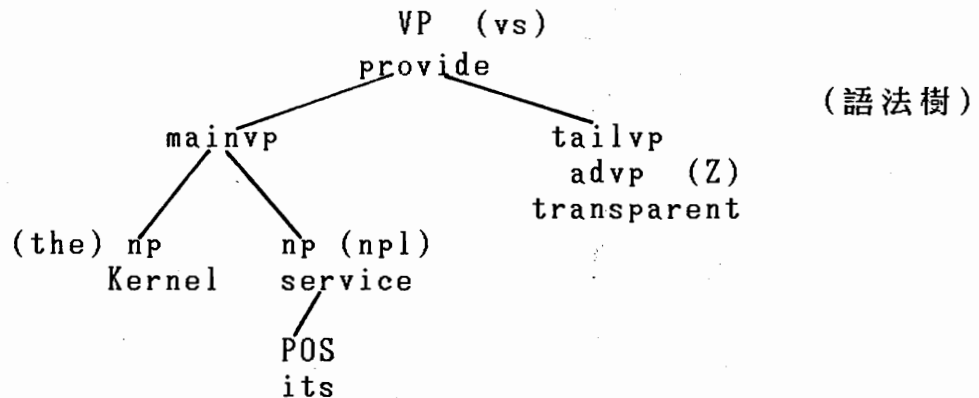
在英文語法分析之後，得出一棵正確的英文語法樹，格位分析則針對此語法樹，進行內格位分析，外格位分析，並選擇各詞的意義。經過格位分析模組，得到一英文語意結構(即格框)。

接下來是英中轉換，主要工作為選出適當的中文詞，填入這個結構之中。在中文合成階段，利用所記的中文句型、內格位、外格位知識，產生合適的中文句子。

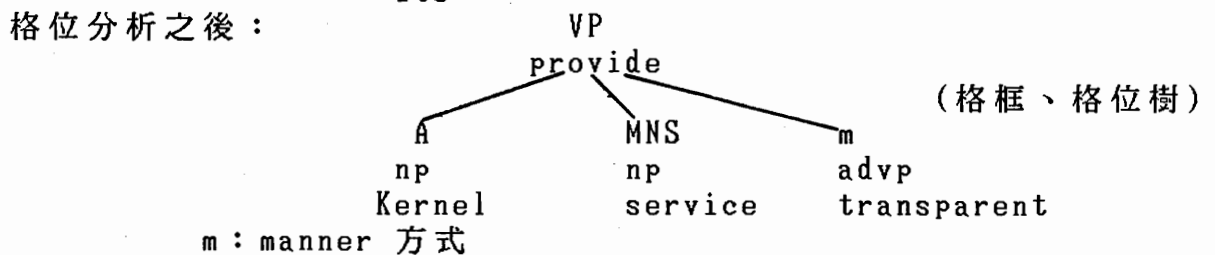
例. The kernel provides its services transparently.

[譯]：核心明顯地提供服務。

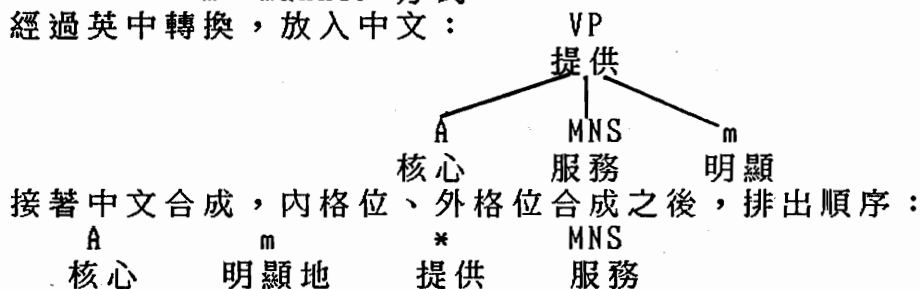
英文語法分析之後產生一語法樹：



格位分析之後：



經過英中轉換，放入中文：



三、內格位分析

在語法分析之後，英文句子已經表示成語法樹的結構，有些系統會直接以此語法樹為轉換成目的語言的輸入，而在我們的系統裡加入了格位分析這個模組，希望能在轉換模組之前，多分析一下句中各片語(phrase)之間的功能關係(functional relationship)，也就是這些片語在句中所扮演的角色(role)。在以下各節我們會一一地說明。

首先，我們來看看在格位分析這個模組要完成哪些動作：

- (1) 內格位的匹配(match) — 依據格框知識所記的判斷條件，將格位(case role) 匹配給適當的格位元(case-element)
- (2) 選擇意義(meaning) — 經由內格位分析，解決動詞及其內格位名詞一辭多義的問題。
- (3) 依據外格位規則，做外格位分析。

基本上，格框是以動詞為主的一種表示法，它的基本單位是格位元，這些格位元可能是內格位或外格位，也可能只是個修飾語而已，而格位元之間相互連接成一個樹狀結構，所以我亦稱格框為格位樹。在格位樹上，保留了語法樹上的資訊，如時態、語態、語形變化...等，關於語法樹及格位樹的區別，請參考上一節的例子。

3.1 內格位與外格位

在討論內格位及外格位的分析之前，讓我們先來看看內格位與外格位是如何區分的。所謂內格位是動詞所支配的格位元，例如主語、賓語、補語....等，從語法樹來看，它們可能是一個名詞片語(NP)介系詞片語(PP)，或不定詞片語(INF)...等等，這些格位元是動詞在某一動詞類型(vptype)所經常存在的。至於一個句子中，動詞的類型在語法分析時就已經決定好了，格位分析首先就是要把這些內格位找出來，而我們記錄動詞下的格框知識，也就是這些內格位的格位及判斷的條件。

相反地，外格位就是對這個動詞不常見的格位元，或是可省略的，例如，表示時間、地點、條件的片語或子句等，從語法樹來看，它們可能是介系詞片語，或從屬子句等。而外格位分析的規則，我們是依它們的語法結構來做分類，在每一分類中，記錄格位及判斷的條件。在下一節中我們會更詳細的描述外格位分析。

3.2 格框知識表示法

這節中，我們將主要敘述內格位分析的動作，及如何利用它來選擇意義，首先讓我們來看看格框知識的表示法。

```
(verb CFM0 CFM1 CFM2...)
CFM := (CF0 CF1 CF2...)
CF := (vptype RT0 RT1 RT2...)
RT := (label (role1 conditions1)
        (role2 conditions2)
        :
    )
```

其中CFM 表示動詞在某一個意義下所記的格框知識，而CF表示在同一意義不同動詞類型所記的格框知識，所以我們可以看得出來，格框知識是與動詞的意義及類型息息相關。而判斷的條件(conditions)通常包含有，語法結構、名詞的語意標記、格標(case marker)等。

3.3 內格位分析

瞭解格框知識的表示法後，我們就可以來看看內格位分析的動作。

- ①根據動詞類型，把記在各個意義下的格框知識找出來。
- ②對每一格框知識做內格位匹配，並記錄結果。
- ③選擇一個匹配結果最佳的格框。
- ④設定格位給每一個內格位元，同時決定動詞及內格位名詞的意義。

接下來，我們舉兩個例子，來說明上述的動作。

假設take和taxi在詞庫中記了下列的資訊。

```

take  cs  v
      mng 0
      cf  ((vp1
            (sub (a huma))
            (obj (o vehi)))
            )
      chi 搭
taxi   cs  n
      mng 0
      sm  (vehi)
      chi 計程車

```

在例句：I take a taxi.

take的意義可經由內格位分析，正確的選出。因為take在vp1 有兩個格框知識，而它們分別匹配 'I' 和 'taxi' 時，第一個兩項都符合，而第二個，只符合一項，所以我們會選擇take的意義應是'搭'。這個例子說明了內格分析如何幫助決定動詞的意義，下面我們再看另一個例子。

假設read格chinese在詞庫中記了下列的資訊

```

read  cs  v
      mng 0
      cf  ((vp1
            (sub (a huma)
                  (ai (sys mach)))
            (obj (o *)))
            )
      chi 讀

chinese cs  n
        mng 0
        sm  (lang)
        chi 中文

```

[註] *表示不限制

在例句：The chinese can read chinese.

到底這兩個chinese 哪一個要譯成'中文'，哪一個譯成'中國人'呢？內格位分析，找出read在vp1 的格框知識，在這個格框知識中限制主格必須是人類(huma)、系統(sys) 或機器(mach)，所以要符合這個格框知識的話，第一個chinese 必須要選擇'中國人'，而第二個chinese 由於受格沒有限制，所以就固定地選了第一個意義'中文'，因此，這句譯出來就是，"中國人可以讀中文"。

當然，假如第一個意義是'中國人'，這句話就有可能譯成"中國人可以讀中國人"，不過，這個譯法，我們可以藉著限制受格的語意標記而將它排除。上面這個例子，我們說明了，內格位分析如何幫助選內格位名詞的意義。

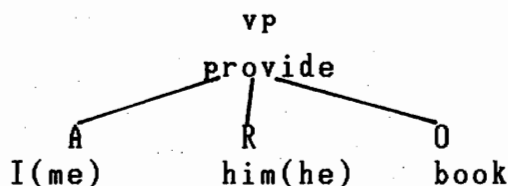
從上面的例子中，我們可以看出語意標記的限制，對內格位分析相當重要，但由實際工作經驗，我們發現要設定這些語意標記，並不是一件容易的事，所以我們還必須要有其他的判斷條件，其中最重要的就是格標，所謂格標主要是指介系詞而言。這些格標往往就可顯示出它後面的名詞片語在句子中的格位，以下我們來看看由provide這個動詞所構成的幾個句子。

- (1) I provide a book to him.
- (2) I provide him a book.
- (3) He is provided with a book by me.

而provide 在詞庫中記了以下的資訊：

```
provide cs v
      mng 0
      cf ((vp1
            (sub (a *))
            (obj (mns *)))
            (vp18
              (sub (a *))
              (o1 (r * (for to))
                 (r *))
              (o2 (o * with)
                 (o *)))
            (vp19
              (sub (a *))
              (o1 (r *))
              (o2 (o *)))
          )
      chi 提供
      規定
```

由provide 所記的格框知識，我們知道provide 在vp1 的格位是 (A MNS)，在vp18及vp19是(A R O) 上面三個句子，(1)(3)是vp18，(2) 是vp19，而這三個句子，雖然在語法結構上各不相同，但經過格位分析，所得到格框是相同的。



判斷(1)(3)的格位，主要是依據格標—'to'和'with'，而(2) 句是依照語法結構及格位的順序而決定。另外我們也從(3) 句中看出來，並不是在動詞之前的名詞片語就一定是句子的主格，尤其在被動式時，它常常是動詞的一個受詞，這是內格位分析所必須注意的。

當然這三個句子，雖然分析出相同的格框，並不表示它們譯文就會完全一樣，對於這方面的討論，我們將在內格位合成一節中提出。

除了動詞之外，對於形容詞，我們也記錄格框知識，例如：

```
synonymous    cs      a
                mng     0
                cf      ((vp2
                          (sub  (n *))
                          (obj  ((o-n * with)))
                          )
                chi     同義
```

對形容詞也做格位分析，主要是因為有些形容詞，具有相當強的動詞性，產生格位後對於中文的生成，將會有許多的幫助，在內格位合成一節我們會看到它的應用。

四、外格位分析

外格位分析的知識是依據外格位規則，它的表示法如下所示：

```
(type
  (role1 conditions1 )
  (role2 conditions2 )
  :
)
```

對於介系詞片語、從屬子句，我們可以根據它們的介系詞及連接詞來判斷格位。另外對於副詞片語，我們則是根據副詞的分類，如時間副詞、頻率副詞....等，來設定格位，而這些資訊都記錄在詞庫中，該副詞之下。

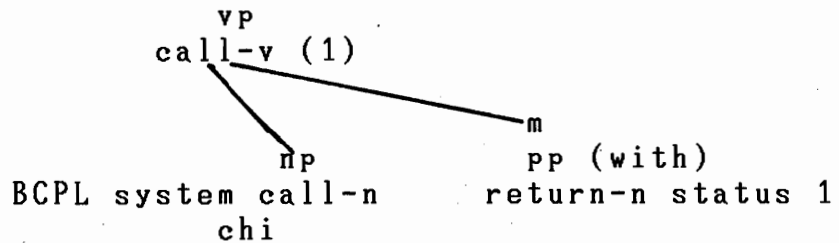
外格位分析的動作，首先是依據語法結構才找相關的外格位規則再由判斷條件決定格位。雖然流程很簡單，但知識整理卻相當困難，因為它是句子中可省略的部分，所以變化很多，判斷的條件也就不好制定。至於外格位在中譯時的應用，我們將在外格位合成一節中提出。

五、英中轉換

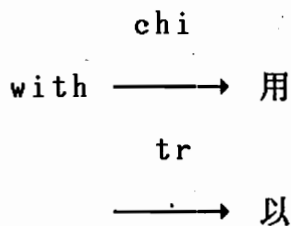
英文詞(word)的某一意義可能有幾種不同譯法，須依照所處的語境來判斷。轉換模組，即根據前面傳來的英文格框結構，選出恰當的中文詞。做法為查看所處的語境，判斷條件包括：表層結構(surface structure)、格位(case role)、語意標記(sm)等。選詞的動作對各種詞類都做。

例. If the B or D calls fail, the program prints a message and calls the BCPL system call with return status 1, terminating its execution and indicating that something went wrong.

[譯]: 假如 B 或 D 呼叫失敗, 程式列印一個訊息, 並且以返回狀態 1 呼叫 BCPL 系統呼叫, 終止它的執行, 並指出某事出錯了。



這句 call-v 在格位分析選出 mng 1 → 呼叫

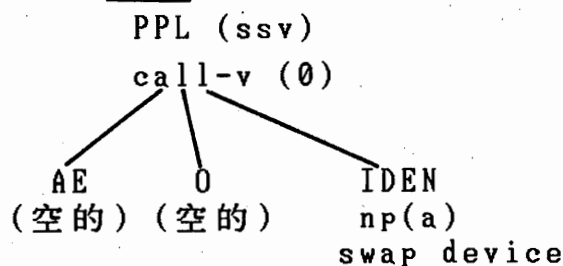


這條轉換法則(tr)為: ((role m outer t) 以)

用到了格位(case role) 作為判斷條件, 所以格框不只用在動詞片語(vp)合成, 對於轉換部份的選詞動作也有幫助。

例. If the system runs low on free memory, the kernel frees memory by writing a process temporarily to secondary memory, called a swap device.

[譯]: 假如系統缺乏自由記憶體, 核心可以把程序暫時寫到輔助記憶體, 來釋放記憶體, 稱為交換裝置。



AE: Agent-Experiencer	經驗
O: Object	對象
IDEN: Identity	名稱

call-v			
mng	0	1	2
chi	稱呼	呼叫	打電話
tr	...稱為		

六、內格位合成

6.1 知識庫與流程

這個模組做：內格位排序、格標(case marker)譯法、被動處理。
用到 中文句型知識庫，

及記在詞(英文verb，中文動詞)之下的知識：

主動句型代號、被動句型代號、被動一般規則。

另外還有主動通則及單純語法運作。

處理的過程為：

①如句型(主動、被動)比對成功，就套用。

②如果句型代號(主動、被動)沒有記，或者句型比對不成功，就進行格標(case marker)譯法。(如果①句型比對成功，格標譯法就含在中文句型之中了。)

③主動：內格位都在主動通則之中，就套用主動通則，不然，則做單純語法處理。

被動：檢查詞(英文verb、中文動詞)之下有沒有記被動一般規則，如有，而且句子條件合，則套用。
不然，就做單純語法處理。

例．主動通則：

A AE AI N SNC I CO-N SRC * R O MNS AS P ACV L

AI: Agent Instrument	施事工具	SRC: Source	來源
N: Neutral	通格	R: Receptient	受物
SNC: Since	始點	AS: As	為格
I: Instrument	工具	ACV: Action with Verb	動作
CO-N: Co-Neutral	共通格	L: Location	場作

格標通則	SNC	自從 (內格位、外格位均適用)
	CO-N	跟
	SRC	從
	AS	當作
	P	來
	p3	於

例．中文句型：

:
4 ——((a r o) (a 把 o * 給 r) (> (lc 'o) 4))
5 ——((ai o acv) (ai 讓 o * acv) (eq '使 (1st 'o)))
6 ——((ai o acv) (ai 使 o * acv))
:
11 ——((a o ws) (a 把 o * 成 ws))
12 ——((a o dst) (a 把 o * 到 dst))
:
103 ——((o dst) (o * 到 dst))
104 ——((a o mns) (o 是 a 用 mns * 的))
105 ——((o a) (o 是 a * 的))
:

中文句型的規格為：格位欄 中文型式 測驗條件

"格位欄" 所記的要跟句子的內格位比對。

"測驗條件" 檢查句子情況，決定是否適用這條句型。

這項可有可無，「沒有："格位欄" 比對成功即可，

「有：尚須"測驗條件"成立才算比對成功。

"中文型式" 在 "格位欄" 及 "測驗條件"(如有)比對成功之後套用。

例. 中文詞庫：

提供 pat (2 4 32 33 37 82) psv (102 105)

稱為 ssv (0)

寫 pat (100 12 34) psv (111 104 105) ssv (2 6)

用 pat (1 84 58 210 92) psv (119 122 133) ssv (3 8)

移植 pat (12) psv (103)

結合 pat (11)

用來 ssv (0)

:

(英文)詞庫 develop

mng 0

chi 發展

chip (11)

[註] chip為記在英文verb之下的主動句型代號，

chis為記在英文verb之下的被動句型代號。

enable

mng 0

chi 能

chip (5 6)

parse-v

mng 0

chi 解析

chip (11)

由此約略可以看出中文句型的共用現象，

共用 [ipat 4] 的有：提供 供應 分配 ...

共用 [ipat 11] 的有：develop parse-v 結合 改變 ...

共用 [ipat 12] 的有：pipe-v extend 寫 移植 映射 ...

共用 [ipsv 105] 的有：提供 寫 強加 ...

[註] [ipat 4] 表主動句型4，

[ipsv 105] 表被動句型105。

	(英文)詞庫	中文詞庫			
主動	chip	pat	主動句型	主動通則	語法
被動	chis	psv	被動句型		語法
	ssv	ssv		被動一般規則	

記在中文動詞的用意是：不同來源對映到同一個中文動詞都可使用同樣部份的中文知識。

例. provide(0) $\xrightarrow[\text{tr}]{\text{chi}}$ 提供
 supply(0) $\xrightarrow[\text{tr}]{} \rightarrow$ 提供
 give(0) $\longrightarrow$ 提供

則這3個英文verb如果在英中轉換之後變成了"提供"，就可以使用記在"提供"下的中文知識，及所引用的句型，而不必在3個英文verb分別記。

記在英文verb的用意是：某一個英文verb的中譯，可能有好幾種句型；而對於這個英文verb的譯文類型來說，只有一部份適合它。

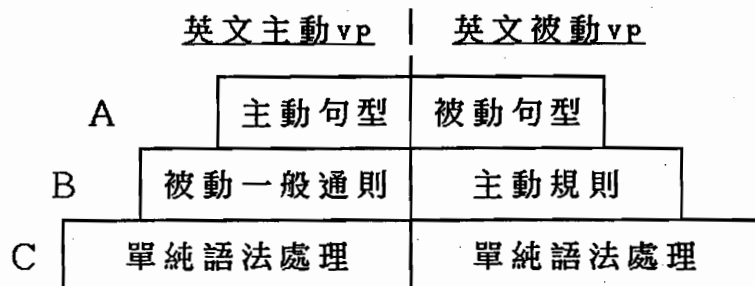
例. (英文)詞庫 enable
 mng 0
 chi 能
 chip (5 6)
 中文詞庫： 能 pat (2 5 6 70 248)

中文動詞"能"有5種主動句型，但是對於enable的譯文類型來說，只有5、6適用。也就是說，在"enable $\longrightarrow$ 能"這個翻譯過程中，enable實際上只對映到"能"的一部份用法，而非全部。當然也有可能enable的其他用法不是"能"所能涵蓋，那時候就用轉換法則(tr)來處理，將enable的其他用法對映到別的中文動詞。

主動句型代號、被動句型代號、

及被動一般規則代號都是記在詞(英文verb、中文動詞)之下的知識，而主動通則、單純語法處理(主動、被動)卻完全對句子情況來做，不管詞的情形如何，只管格位(inner role)或語法結構。

內格位合成的做法，也可用下面層狀塔的觀點來看：

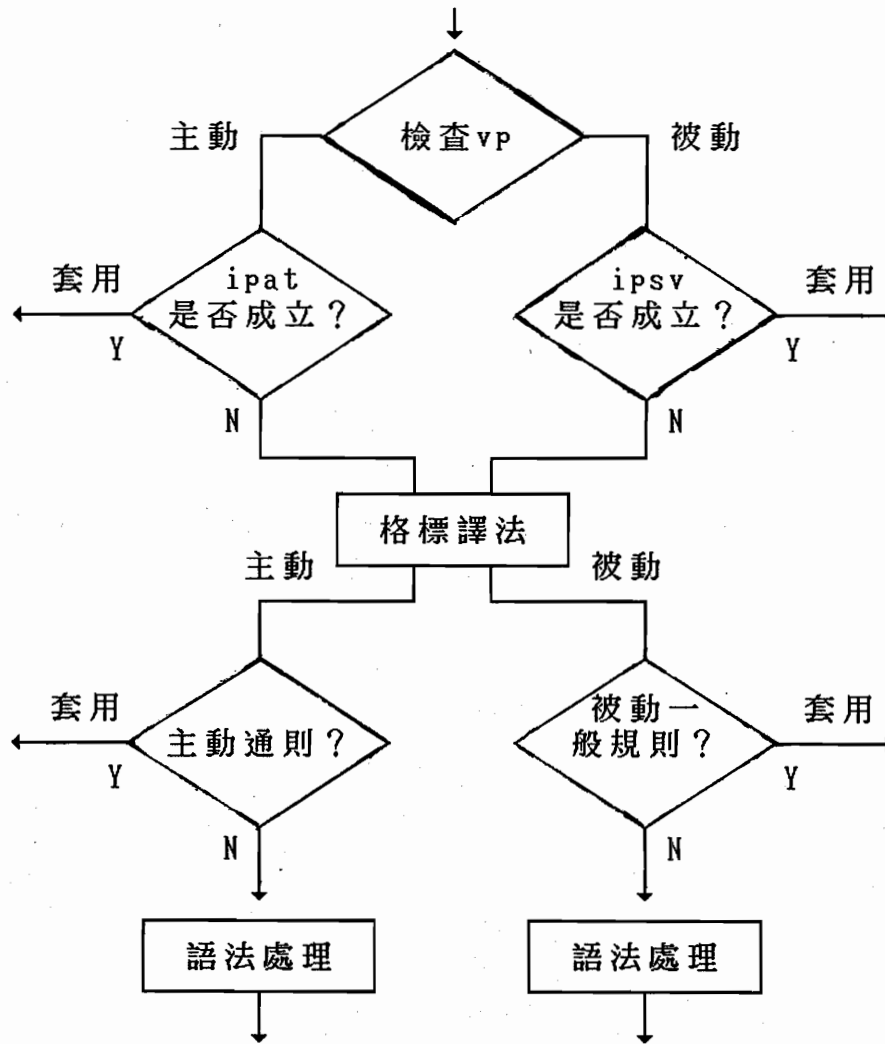


A成立，則套用。

A不成立，試驗B。B成立，則套用。

B也不成立，就落到C。

內格位合成的流程如下：



關於格標的翻譯，如果句型比對(主動、被動)成功，格標譯法就含在句型之中了。

要不然，就用 格標譯法模組(主、被動共用)來處理。

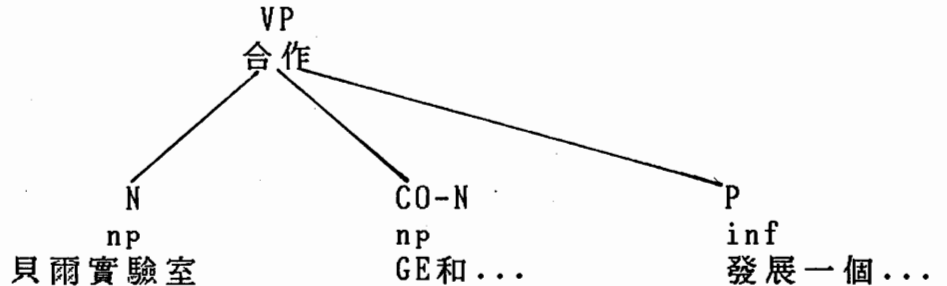
[註]格標譯法是針對格位而言的，該格位在中文用法有此格標譯法需要。格標譯法並不同於英文介詞(preposition)譯法，因為該格位不一定是PP，可能是：np、inf...。

格標譯法模組：先試驗英文介詞的轉換規則(tr) (如果該格位語法結構是PP)。不成功，就試用格標通則 (針對某一格位加上一格標譯法) 如果格標通則沒有記，用英文介詞的chi (如果該格位語法結構是PP)。

6.2 主動通則

例. In 1965, Bell Telephone Laboratories joined an effort with the GE and the Project MAC of the Massachusetts Institute of Technology to develop a new operating system called Multics.

[譯]: 在1965年, 貝爾實驗室跟GE和MIT 計劃, 合作來發展一個稱為 Multics 的新作業系統。



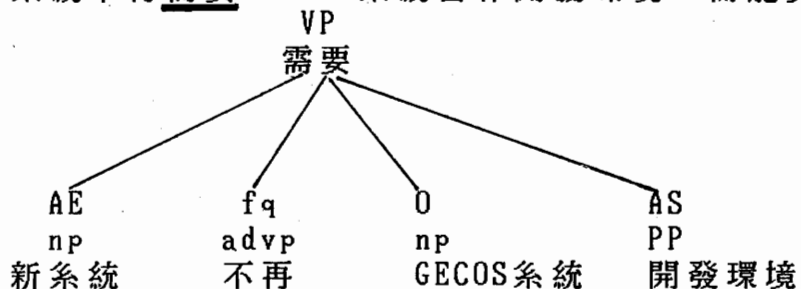
因為中文詞庫並沒有記 "合作" 這一項, 所以這句是走主動通則出去的。格位順序依照主動通則(見6.1), "跟" 來自格標通則譯法 (碰到CO-N就加 "跟") (見6.1)。"來" (格位P 的格標譯法) 也源於格標通則。在內格位合成之後, 這個vp得出結果為: N 跟CO-N * 來P。

"主動通則" 是重要的一部份, 因主動通則及語法處理能夠涵蓋的類型, 就不需要再多記相關的東西了。

而 "中文句型" 卻可以在某些較困難的類型, 做比較細緻、精巧的處理, 達到優良的翻譯效果。

例. Eventually, the new system no longer needed the GECOS system as a development environment but support itself.

[譯]: 最後, 新系統不再需要GECOS 系統當作開發環境, 而能夠支持本身。



中文詞庫沒有記 "需要" 這一項, 所以這句格標譯法按照格標通則, 內格位順序依據主動通則。(見6.1)

在內格位合成之後, 這個格框(vp)的內格位部份為: AE * O 當作AS

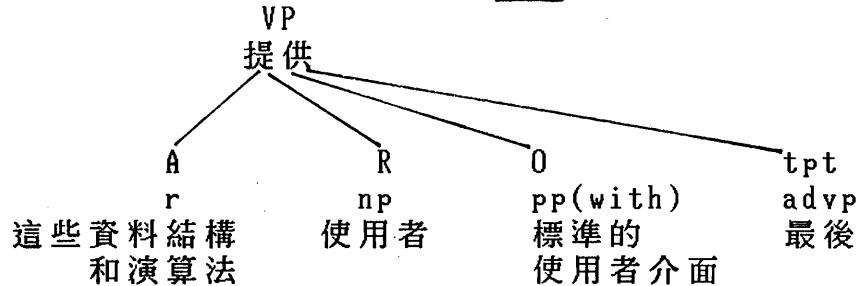
fq: frequency 頻率

6.3 主動句型 被動句型

接下來介紹幾個利用 ipat、ipsv 句型的例子。

例. It examines the major data structures and algorithms used in the operating system that ultimately provide users with the standard user interface.

[譯]：它討論在作業系統用的主要資料結構和演算法，這些資料結構和演算法最後把標準的使用者介面提供給使用者。



[中文詞庫] 提供 pat (2 4 32 33 37 82)

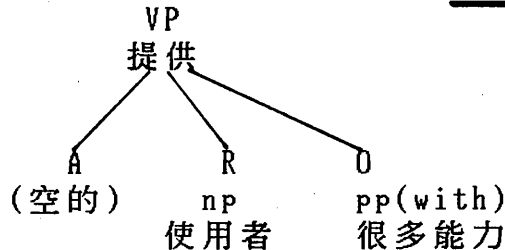
↑ ↑ ↑
"格位欄"都是(A R O)

[ipat4]——((a r o) (a 把 o * 給 r) (> (lc 'o) 4))

[ipat4]"測驗條件"：格位 O 漢字長度需大於 4，對這個 vp 來說成立。在內格位分析之後，此格框(vp)的內格位部份即為[ipat4]的"中文型式"：A 把O * 給R。

例. They have simple options that make them easy to use but provide the user with a lot of power.

[譯]：他們有簡單選擇項，使他們容易用，但是提供很多能力給使用者。



[ipat 4]——((A R O) (A 把 O * 給 R) (> (lc 'O) 4))

[ipat 32]——((a r o)(a * r o)(and (= (lc 'r) 1)(> (lc 'o) 1)))

[ipat 33]——((a r o) (a * o 給 r))

[ipat 32] "測驗條件" 為：格位 R 是 1 個漢字，格位 O 的中文長度大於 1。

對於這個中文格框來說，格位 O 長度並未大於 4 個漢字，[ipat 4] 不成立。

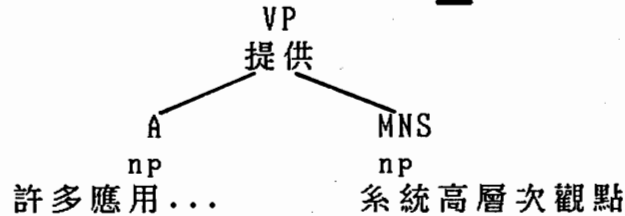
格位 R 並非 1 個漢字，[ipat 32] 也不成立。

所以用[ipat 33]。

內格分析之後，此格框變為[ipat 33] 的"中文型式"：A * O 給R。

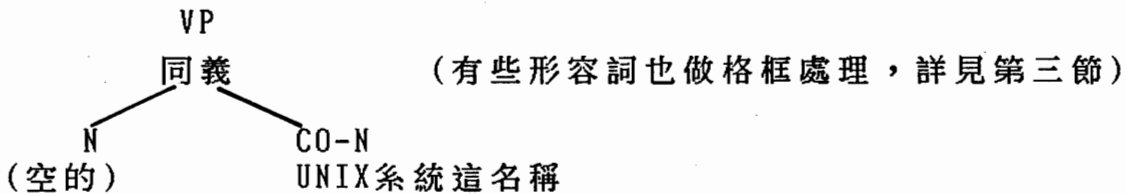
例. Many application subsystems and programs that provide a high level view of the system such as the shell, editors, SCCS, and document preparation files, have gradually become synonymous with the name UNIX system.

[譯]: 許多應用子系統和程式提供系統高層次觀點, 如SHELL、編輯器、SCCS和文件準備檔案, 逐漸變為跟UNIX系統這名稱同義。



由於 "提供" pat項 所記的 6 個主動句型, 它們的 "格位欄" 都不是 (A MNS), 所以 格框"提供" 走主動通則出去。

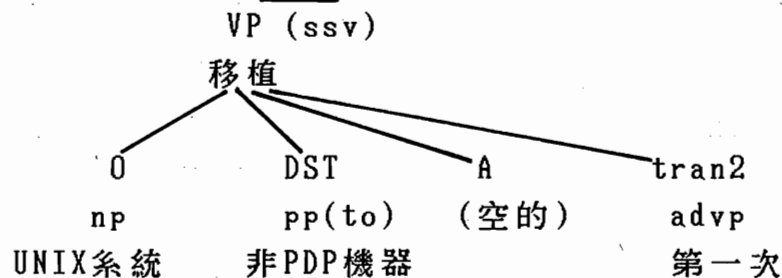
可能中文句型知識庫中沒有一條 "格位欄" 是 (A MNS), 也可能 "提供" 不適用那一條, 以致 pat項 未記該條。



格框"同義" 也是走主動通則出去。(參見6.2 "合作"例)

例. 1977 was the year that the UNIX system was first ported to a non-PDP machine.

[譯]: 1977是UNIX系統第一次移植到非PDP機器的一年。



DST: Destination終點

這是一個被動格框。

[中文詞庫] 移植 pat(12) psv(103)

[psv 103]——((o dst) (o * 到 dst))

[psv 103] "格位欄" 是(O DST), 在跟句子的內格位比對時, 句子中空的格位可以不計。

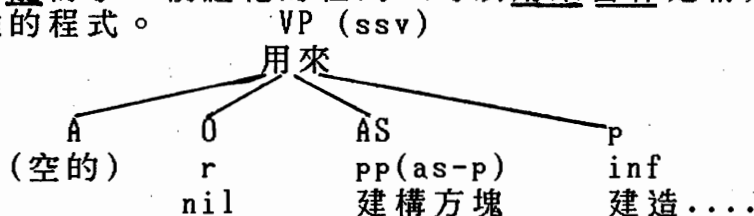
但某一條句型 "格位欄" 如是(A O DST), 內格位比對也會成功。

在內格位 合成之後, 此格框的內格位部份為: O * 到DST (格位A已刪)

6.4 一個動詞多種句型 (一對多)

例. As described earlier, the philosophy of the UNIX system is to provide operating system primitives that enable users to write small, modular programmes that can be used as building blocks to build more complex programs.

[譯]: 如較先前所描述的, UNIX系統的理念是要提供作業系統基本指令, 讓使用者能寫小、模組化的程式, 可以用來當作建構方塊來建造更複雜的程式。

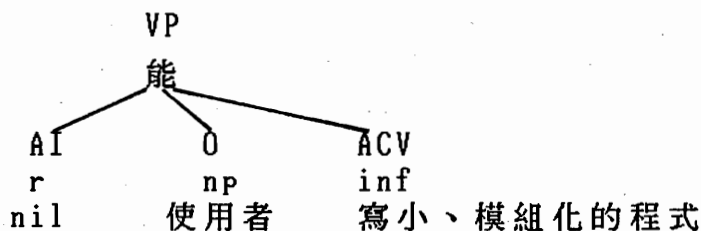


chi tr
use-v(0) → 用 → 用來

[中文詞庫] 用來 SSV(0)

"用來" 這個中文動詞沒有記 psv項, 因此被動格框不會用被動中文句型。接下來走 格標譯法模組, 格位 AS 用到了格標通則。(參見6.1 流程) 再接著檢查被動一般規則, 記了 ssv(0), 則套用。

ssv(0) 為可通用被動, 順序依照原來順序, 不加"被"或任何其他東西。



[詞庫] enable
 mng 0
 chi 能
 chip (5 6)

[ipat 5] — ((ai o acv) (ai 讓 o * acv) (eq '使 (1st 'o)))

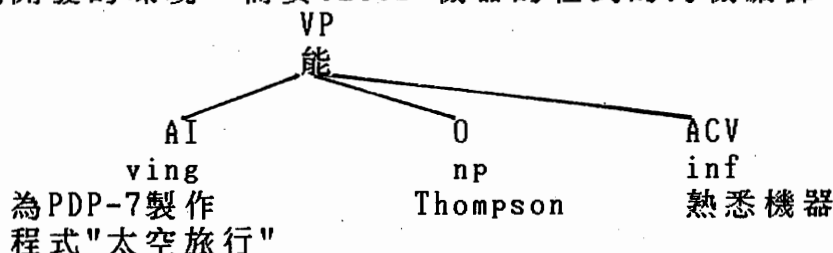
[ipat 6] — ((ai o acv) (ai 使 o * acv))

[ipat 5] "測驗條件": 格位 O 的第一個漢字是"使"。

這個格框符合此項條件, 所以引用[ipat 5], 得出: AI 讓 O * ACV

例. Programming "the Space Travel" for the PDP-7 enabled Thompson to learn about the machine, but its environment for program development required cross-assembly of the program on the GECOS machine.

[譯]: 為PDP-7 製作程式"太空旅行", 使THOMPSON能熟悉機器, 但是它程式開發的環境, 需要GECOS 機器的程式的跨機編譯。



[ipat 5] "測驗條件" 未成立, 所以引用[ipat 6], 得到: AI 使 O * ACV

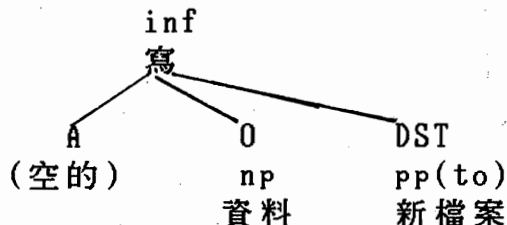
6.5 句型共用 (多對一)

有些不同動詞共用同一條中文句型，請見下面的例子：

例. [ipat 12]——((a o dst) (a 把 o * 到 dst))

①The program then calls the subroutine Brian, which goes into a loop, invoking the B system call to read a buffer's worth of characters from the existing file, and invoking the D system call to write the data to the new file.

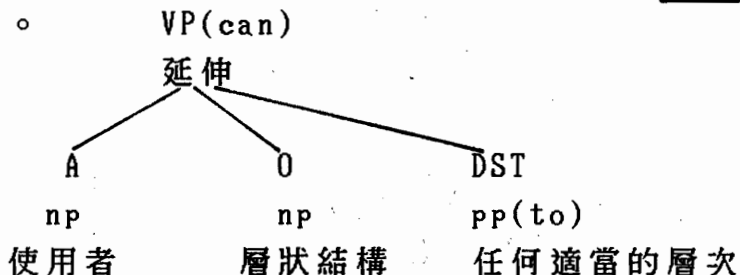
[譯]：然後程式呼叫副程式Brain，它進入迴圈，啟動B系統呼叫來從現有檔案讀一緩衝器字元，並啟動D系統呼叫來把資料寫到新檔案。



[中文詞庫] 寫 pat (100 12 34)

②Although the figure depicts a two level hierarchy of application programs, users can extend the hierarchy to whatever levels are appropriate.

[譯]：雖然圖描述應用程式的二層塔，使用者可以把層狀結構延伸到任何適當的層次。



[詞庫] extend

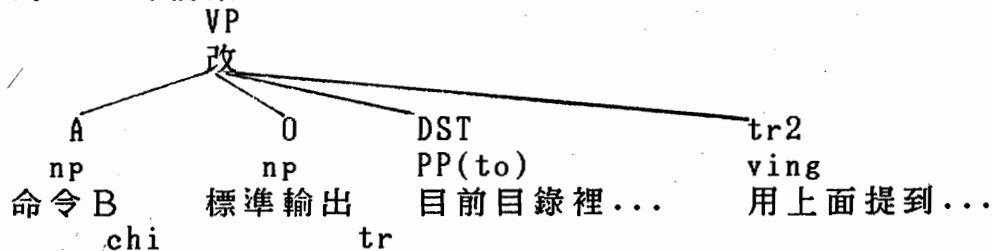
mng 0

chi 延伸

chip (12)

③For instance, the command line D lists all files in the current directory on the standard output, but the command line B redirects the standard output to the file called BCPL in the current directory, using the MCC system call mentioned above.

[譯]：例如，命令D把目前目錄的全部檔案列印在標準輸出，但是命令B用上面提到的MCC系統呼叫，把標準輸出改到在目前目錄裡稱為BCPL的檔案。



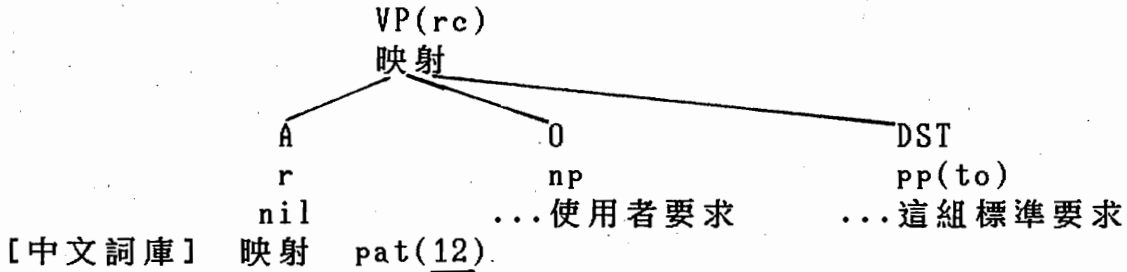
[詞庫] redirect → 重導

→ 改

[中文詞庫] 改 pat(12)

④The UNIX system can support any language that has a compiler or interpreter and a system interface that maps user requests for operating system services to the standard set of requests used on UNIX system.

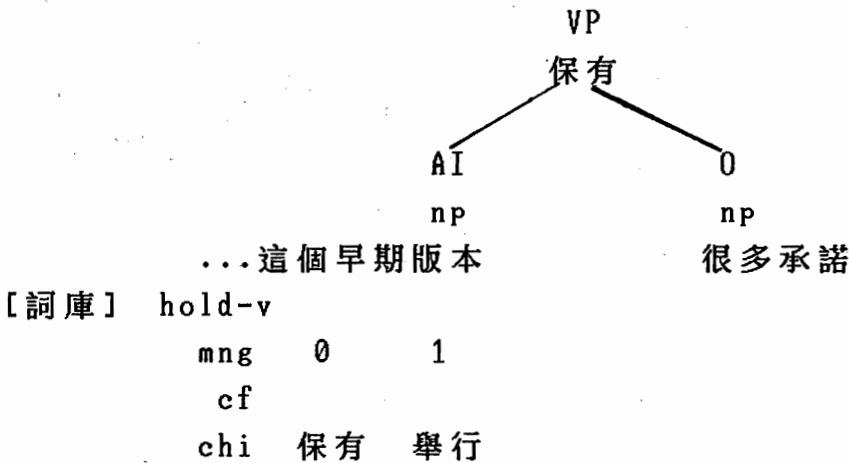
[譯]: UNIX系統可以支持任何語言、只要它有編譯器或直譯器, 也可以支持一個系統介面, 把對作業系統服務的使用者要求, 映射到UNIX系統上用的這組標準要求。



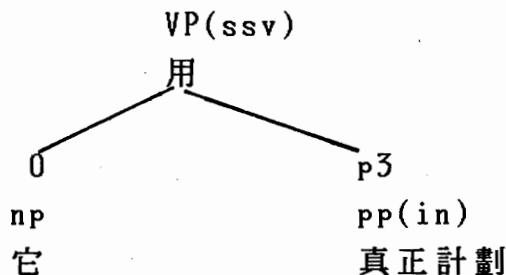
6.6 被動一般規則(ssv)

例. Although this early version of the UNIX system held much promise, it could not realize its potential until it was used in a real project.

[譯]: 雖然UNIX系統的這個早期版本保有很多承諾, 直到它用於真正計劃, 它才能實現其潛能。



經過格位分析模組, 選行了正確的格框(也就是正確的mng, 參見第三節)。



[中文詞庫] 用 psv (119 122 133) ssv (3 8)

psv 中所記的都未能跟這個格框比對成功,

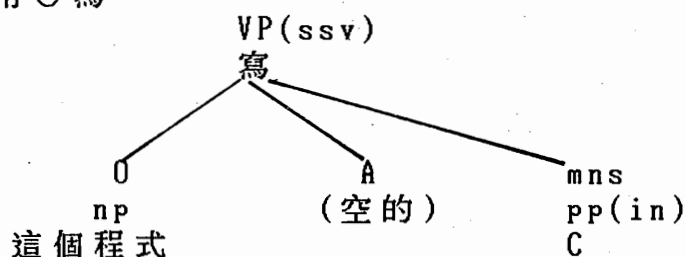
經過 格標譯法模組, 進而檢查被動一般規則, 發現ssv(8)成立, 則套用。

ssv(8) 為檢查有外格位p3, 就可以通用被動, 不須加"被"或其他東西。

"於"來自格標通則, 在外格位合成加上的。(參見6.1, 及第七節)

例. ① The program was written in C.

[譯]: 這個程式用 C 寫。



[中文詞庫] 寫 psv (111 104 105) ssv (2 6)

psv 中句型沒有比對成功，ssv(2)成立，則引用。

ssv(2) 是說如果有外格位修飾語PP，則可以通用被動。

漢語中有如下的現象：

有此動詞在加上某些修飾語(如：PP、AUX、Comp...)，就可以通用被動。

這個程式用C寫。 (O) [實際上就是："被寫"]
PP

這個程式可以寫。 (O)
AUX

這個程寫好了。 (O)
Comp

這個程式可以在明天寫好。 (O)
AUX PP Comp

這個程式可以用C寫。 (O)
AUX PP

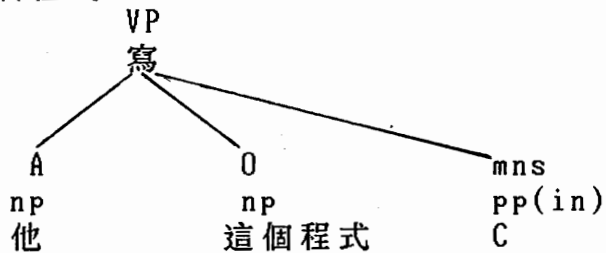
這個程式寫。 (X)

這種現象可以用在機器翻譯上，因為英文技術文章的被動句子相當多，而漢語一般習慣卻不喜歡被動，所以如果查知原句中有這類修飾語，則可以通用被動。

6.7 原文的順序

② He wrote the program in C.

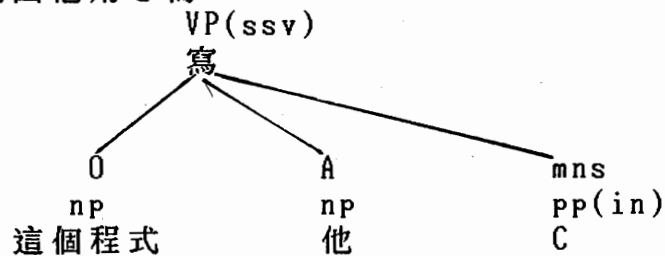
[譯]：他用C寫這個程式。



此格框走 "主動通則"。

③ The program was written by him in C.

[譯]：這個程式由他用C寫。



此格框套用 被動一般規則 ssv(2)。

請比較②與③，格框中的格位(role)完全相同，順序卻不一樣，譯文結果也不一樣(順序不同)。對於格框的看法，有些觀點認為role完全一樣(不管主動、被動)，就是意義相同，用同樣的格框(如將②與③的格框視為一樣)。但是此機器翻譯系統，引用了格框做法，卻把格位(role)視為有順序，將原文的順序考慮在內。如例③很明顯 "這個程式" 是中譯主題，也反映了原句的主題。因為"忠於原文"，是翻譯的基本要求。此機器翻譯系統，希望儘量做到2方面的主要原則：譯文合於漢語自然習慣，又能充份表現原句的意思(包括：主題、順序.....)。

7.1 處理方式

外格位的處理可以說都是通則（相對於內格位合成的主動通則，參見 6.1），也就是說針對格位及句子結構來做，而不管詞的情況。

(句首) Sub Aux ... * ... ←内格位末尾
主語 助動詞 ↑ ↑ ↑
uu0 uu1 uu2 uu3

tran	:transition	轉折
if-cl	:if-clause	條件子句
top	:topic	主題
tpt	:time-point	時間點
fq	:frequency	頻率
pla	:place	場所
src	:source	來源
m	:manner	方式
cmp	:compare	比較
mns	:means	方法
subcl	:subclause	附屬子句

如 P 在 6.2"合作"例 中，是內格位； 在第 1 節例中，是外格位。

又如 MNS 在 6.3"提供"例 中，是內格位；在 6.6、6.7"寫"例、第 1 節例中 是外格位。

如果是動詞常用的部份（也就是在許多句子中普遍出現的情形），就

當作內格位處理。如果是不常用的部份，就當作外格位處理。(而不是"必要"或"非必要")這是一個從實用觀點出發的做法。如果一個動詞的某一成份使用時為必要成份，而其出現率僅1%，我們就可能將它歸為外格位處理。如動詞的某一成份雖可省略(非必要)，但常出現，我們也可能將其當成內格位。因為外格位的處理較難掌握，內格位處理比較有明確的控制，我們按出現頻率來安排內、外格位，對於全面應用(大量文章)比較有利。

依外格位名稱(outer role name)，分別屬於uu0、uu1、uu2、uu3中的那一族，塞進前一節得出的內格位中文句型之中。如果一個格框裡有2個不同外格位屬於同一族，則它們彼此的順序依照族中的順序。又如果一格框中有2個外格位名稱相同，則它們彼此的順序就依照原文裡的順序。(原句各成份的順序仍保留在格框中，參見6.7)。

如果某一外格位沒有外格位名稱、或外格位名稱不認得(uu0、uu1、uu2、uu3都找不到)時，就走nil(uu2中的一項)這條路，這也可說是語法預設規則(syntax default rule)。

在外格位合成階段，中文主語的認定依照以下2個原則：

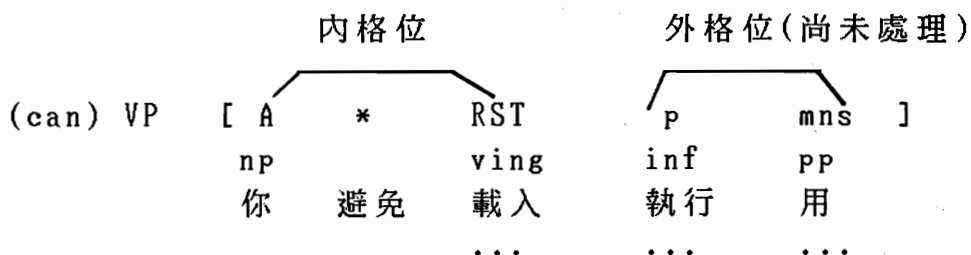
- ①原來英文句子的主語(subject)，如果仍在中文動詞左邊，則以它和它左邊的內格位當做中文主語。
- ②如原英文subject這時已在中文動詞之右，則以此時內格位中文句型的第1項做為中文主語。

進入外格位合成模組時，就先做外格位的格標翻譯，直接呼叫內格位合成部份"格標譯法"那個模組。(參見6.1)再把外格位分別歸入它所屬的一族，助動詞的中譯也用轉換法則(tr)處理，最後將各成份依7.1所記模式合併。

例. You can avoid loading two separate disks to perform either of these tasks by using an AUTOEXEC.BAT file.

[譯]: 你可以用一個AUTOEXEC.BAT檔，避免載入2片不同磁碟，來執行其中一項工作。

內格位合成之後：



uu2 : mns
uu3 : p

經過外格位合成模組，得出結果：

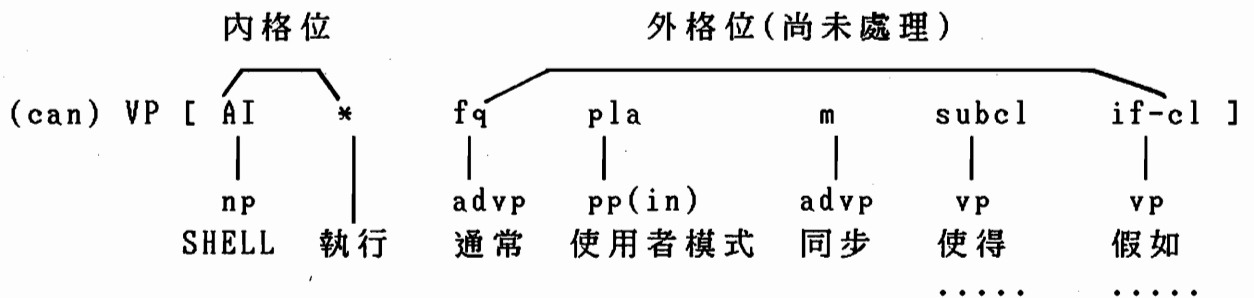
Sub Aux (uu2) * ... (uu3)

[A 可以 mns * RST 來p]
你 用.. 避免 載入.. 來執行..

例. The shell can usually execute synchronously in user mode so that application programs could invoke system calls, if it resides in the machine.

[譯]：假如SHELL 駐留在機器，它通常可以在使用者模式同步執行，使得應用程式能啟動系統呼叫。

內格位合成之後：



uu0 : if-cl
uu1 : fq
uu2 : pla m
uu3 : subcl

經過外格位合成模組，得出結果：

(uu0) Sub (uu1) Aux (uu2) * (uu3)

[if-cl AI fq 可以 在pla m * subcl]
假如.. SHELL 通常 在使用者.. 同步 執行 使得..

7.2 格框及單純語法處理

內格位合成模組有 "單純語法處理" 做為底層，外格位合成模組也有純粹語法的路(沒有role或role不認得可以走外格位nil 的路)，所以可以說，中文合成部份，是將格框作法架在 單純語法處理 之上。單純語法

處理可以補格框之不足，而格框在純粹語法操作較弱的地方、很難處理的方面，又能夠予以加強，達成良好的翻譯品質。另一方面，把“純粹語法處理”做為底層，又可以省掉非常大量建格框知識的人工，以內格位合成來說，某些句子如果中譯順序跟英文一樣，又不需要比較特別的格標譯法，就根本不必多費工夫去建相關的格框知識，因為單純語法操作已足夠了。英文與中文有某些相似的部份，一些較簡單的操作(語法)就可以了，不需用到格框這種比較複雜、精巧的處理。對於外格位合成模組，如某些句子類型走語法預設規則(Syntax default rule)就可過了，也就不必多費心思去考慮那些句子怎樣長外格位、怎樣合成。這樣把相當比例的東西丟給語法預設規則去做，而用格框來對付更難的問題，單純語法不容易做得好的問題。

八、 結論

8.1 多對一

在這個機器翻譯系統中，常可以見到“多對一”的關係。多對一可以節省大量建知識的人工，不需要記許多重複的東西。

例如 格位分析 對不同verb或相同verb不同語境，解析出了表達句意的同一格框，在 合成部份 就可能引用 主動通則(見6.2)，或外格位通則(請見7.1)，而得出良好的翻譯結果，這就是一種 多對一 的關係。

不同英文verb，對映到同樣的中文動詞(經過chi 或tr)，而能夠共用 記在中文知識庫的東西，這也是一種 多對一 的關係。

chi
例. permit ———> 允許

chi
allow ———> 允許

則可以共用記在“允許”這個中文動詞之下的知識。

或如6.1，provide(0)、supply(0)、give(0)都對映到“提供”。

有些不同的動詞，可以共用相同的句型，這也是 多對一 的關係。
如 [ipat 11]——((A 0 WS) (A 把 0 * 成 WS)) (參見6.1)
用到 [ipat 11] 的有：develop parse-v 結合 改變 ...

或如 6.5 例中 ①寫 ②extend ③改 ④映射 都用到 [ipat 12]。

8.2 一對多

在這個機器翻譯系統中，也常可見到 "一對多" 的關係。一對多可以在需要時有各種不同選擇，應付翻譯上多種變化的需求。

一個英文 verb 在不同環境，可能分析出反映語意的不同格框，這就是一種 一對多 的關係。（參見 3.3）

英文 word 之意義在 格位分析 決定（參見第 3 節），同一 word 的同一意義 在不同語境，也會有各種變化的翻譯需求，這在英中轉換部份做，也是一種 一對多 的關係。

	chi		
例. with	→	用	
	tr		
	→	以	
	tr		
	→	附帶	(請見第 5 節)
	:		
	:		

同一個動詞的不同格框，或同一格框在不同情況，可能使用不同的 中文句型(主動、被動)，由 "格位欄"、"測驗條件" 及 句子情況而決定，這也是一種 一對多 的關係。

如 6.3 "提供" 在 3 個例子中 分別用到了 [ipat 4]、[ipat 33]、及 "主動通則"。

6.4 "enable" 例 用到了 [ipat 5]、[ipat 6]。

8.3 格框應用

機器翻譯是不是需要很深的語意分析，一直是個爭議頗多的問題。因為如果語意分析做得深入而且詳盡，又要應付各種不同類型，知識庫及程式部份必然變得龐大而且繁複，難以掌握及維護，又不容易加進真正適合廣泛用途的東西。然而如果只做到很表層的結構，又不容易得到好的翻譯品質。用格框來處理機器翻譯，感覺上還算是一個適當的結構，不會繁雜到無法運作、不切實際，又能達成許多希望達到的效果。

九、感謝

此機器翻譯系統承蒙經濟部的電腦與通訊技術發展計劃 (計劃編號: *ITRI-048-P001 (77)) 支持，特此誌謝。