

自然語言通訊




本章我們看到人們如何以自然語言相互溝通，並且電腦代理人如何可以加入對話。

通訊(communication)乃透過產生和感知一套共通慣例系統中的**信號**(sign)，造成有意圖的資訊交換。大多數動物用信號表示重要資訊：這裡有食物、附近有掠奪者、前進、撤退、來交配吧。在一個部分可觀察的世界裡，通訊可以幫助代理人成功，因為代理人能夠學到其他代理人觀察到或推論的資訊。人類是所有物種最健談的，且若電腦代理人要有幫助，它們會需要學習使用語言。在本章我們觀察用來溝通的語言模型。目標為對溝通作深刻理解的模型必然需要比目標為比如垃圾郵件分類的簡單模型更為複雜。我們由句子的片語結構的文法模型開始，增加語意於模型，然後應用其於機器翻譯與語音辨識。

23.1 片語結構文法

第 22 章的 n 元語言模型是基於文字序列。這模型最大的問題為**資料稀疏**——含 10^5 個單字的字彙，並有 10^{15} 個三元機率需估計，並且所以甚至 1 兆個字的語料庫將無法支援對於它們所有的可靠度估計。我們可以透過一般化來解決稀疏的問題。對於「black dog」比「dog black」還來得頻繁的事實與相似的觀察，我們可以形成在英文裡形容詞往往放在名詞之前的一個歸納(其中他們傾向跟隨法語名詞：「chien noir」是最頻繁的)。當然總是有例外的，「galore」是個跟在所修飾名詞之後的形容詞。儘管有例外，**詞彙範疇**(也稱為**部分語音**)的想法(如名詞或形容詞)是個很有用的一般化——本身就很有用，但更有用的原因在於，當我們將這些詞彙範疇串連在一起形成**句法範疇**(如名詞片語或動詞片語)，並且合併這些句法範疇成為句子的**片語結構**之樹狀表示：巢狀片語，每個以目錄標示。

有許多基於片語結構的概念之競爭語言模型，我們將描述較為受歡迎的模型，稱之為**機率的語境自由語法**(the probabilistic context-free grammar 或 PCFG )。文法是一個規則的集合，定義著語言中所允許單字的串列。「語境自由」描述於底下的「產生能力」(generative capacity)專欄，「機率的」表示文法指定機率於每個字串。這是個 PCFG 法則：

$$VP \rightarrow Verb [0.70]$$

$$| VP NP [0.30]$$

這裡 VP (動詞片語)和 NP (名詞片語)是**非終端符號**。這文法也指實際單字，其中稱為**終端符號**。這個法則指出動詞片語有 0.7 的機率由單獨的動詞組成，並且有 0.3 的機率是一個跟在名詞片語後的動詞片語。附錄 B 描述非機率的語境自由文法。

我們為英語的一小片段定義文法，其中適合於探索魔獸世界(wumpus world)代理人之間的通訊。我們稱這語言為 ϵ_0 。後面的章節將會改進 ϵ_0 ，使其更接近於真正的英語。我們不可能設計出英語的完整文法，因為我們甚至找不到兩個人能在正確英語的文法這點上有完全相同的意見。

產生能力

文法的形式可以根據**產生能力**(generative capacity)，即其所能表示的語言集合，來加以分類。Chomsky(1957)僅根據重寫規則形式的不同就定義了 4 種文法形式。這些類型可以按照層次結構排列，其中每範疇型都可以用來描述能力比它低的類型所能描述的全部語言，以及某些額外的語言。下面我們列出這個層次結構，由描述能力最強的類型開始：

遞迴可列舉文法(recursively enumerable grammar)使用無限制的重寫規則：重寫規則的左右兩側都可以包含任意數量的終端符號和非終端符號，就像規則 $ABC \rightarrow DE$ 中一樣。這種文法與圖靈機具有同等的表達能力。

語境限定文法(context-sensitive grammar)的限制只是重寫規則的右邊必須包含至少與該規則左邊同樣數量的符號。「語境限定」這個名稱來自這樣的事實：規則如 $AXB \rightarrow AYB$ 的意思就是，在前 A 後 B 的語境中，則 X 可以重寫為 Y 。語境限定文法能夠表示諸如 $a^n b^n c^n$ (a 的 n 個副本後面跟著相同數量的 b 和 c 的一個序列)這樣的語言。

語境自由文法(或稱「文本無關文法」，context-free grammar，或簡稱 CFG)中，每個重寫規則的左邊由一個單獨的非終端符號組成。因此，每條規則許可在任何語境中把該非終端符號重寫為規則的右邊。CFG 在自然語言和程式語言的文法中很流行，雖然現在廣泛接受的觀念是至少有些自然語言不是語境自由的(Pullum, 1991)。語境自由文法能夠表示 $a^n b^n$ ，但不能表示 $a^n b^n c^n$ 。

正規文法(regular grammar)是限制最強的一種類型。每條重寫規則的左邊是一個單獨的非終端符號，而右邊是一個終端符號，其後還可以跟著一個任意的非終端符號。正規文法與有限狀態自動機具有同等的表達能力。它們並不適合程式語言，因為它們不能表示諸如對稱括弧一類建構子的結構 ($a^n b^n$ 語言的一種變形)。它們最多只能表示 $a^* b^*$ ，由任意數量的 a 後面跟著任意數量的 b 組成的一個序列。

儘管層次等級越高的文法表達能力越強，但是處理它們的演算法效率也越低。到 20 世紀 80 年代中期之前，語言學家一直把注意力集中在語境自由和語境限定語言上。在那之後，出於對上百萬字甚至上十億字的線上文章進行非常快速處理的需要(即使以不完備的分析為代價)，致使正規文法逐漸成為研究重點。正如 Fernando Pereira 指出的那樣：「年紀越大，我研究的 Chomsky 文法層次越低」。要瞭解他的意思，比較 Pereira 和 Warren (1980)與 Mohri、Pereira 和 Riley(2002)(並且注意這三位作者目前都於 Google 研究大型的文件語料庫)。

23.1.1 ϵ_0 的詞典

首先我們定義一個**詞典**(lexicon)，即允許的辭彙列表。這些詞語以字典使用者熟悉的範疇或詞性來分類：名詞、代名詞、指物的名稱、指事的動詞、修飾名詞的形容詞，以及修飾動詞的副詞。冠詞(像是 the)、介系詞(in)和連接詞(and)。圖 23.1 表示語言 ϵ_0 的一個小型詞典。

每個以「...」結尾的範疇表示在這個範疇中還有其他詞語。對於名詞、動詞、形容詞和副詞而言，將它們全部列出在原則上是不可行的。不僅每類中有成千上萬詞語，而且經常還有新詞加入——諸如 MP3 或者 anime。這 5 類被稱為**開放類**(open classes)。對於代名詞、關係代名詞、冠詞、介系詞和連接詞的範疇，我們可以花點時間得到一個所有詞彙的列表。這些被稱為**封閉類**(closed classes)。它們只包含少量詞語(幾個到數十個)。封閉類歷經數個世紀才會逐漸變遷，而不是幾個月內就產生改變。例如，「thee」和「thou」這兩個詞語在十七世紀是經常使用的代名詞，到了十九世紀才逐漸少用，而今天僅僅在詩歌和某些地區方言中才能見到。

<i>Noun</i>	→	stench [0.05] breeze [0.10] wumpus [0.15] pits [0.05] ...
<i>Verb</i>	→	is [0.10] feel [0.10] smells [0.10] stinks [0.05] ...
<i>Adjective</i>	→	right [0.10] dead [0.05] smelly [0.02] breezy [0.02] ...
<i>Adverb</i>	→	here [0.05] ahead [0.05] nearby [0.02] ...
<i>Pronoun</i>	→	me [0.10] you [0.03] I [0.10] it [0.10] ...
<i>RelPro</i>	→	that [0.40] which [0.15] who [0.20] whom [0.02] ...
<i>Name</i>	→	John [0.01] Mary [0.01] Boston [0.01] ...
<i>Article</i>	→	the [0.40] a [0.30] an [0.10] every [0.05] ...
<i>Prep</i>	→	to [0.20] in [0.10] on [0.05] near [0.10] ...
<i>Conj</i>	→	and [0.50] or [0.10] but [0.20] yet [0.02] ...
<i>Digit</i>	→	0 [0.20] 1 [0.20] 2 [0.20] 3 [0.20] 4 [0.20] ...

圖 23.1 ϵ_0 的詞典。*RelPro* 是相對代名詞縮寫，*Prep* 是介系詞，*Conj* 是連接詞。每個類別的機率總和為 1

23.1.2 ϵ_0 的文法

下一步則是將詞語組成片語。圖 23.2 表示 ϵ_0 的文法，對於 6 個句法範疇的每個規則和每個重寫規則的實例^[2]。圖 23.3 表示一個句子「Every wumpus smells.」的**剖析樹**。剖析樹給出個建設性的證明，其中詞彙的字串確實是個根據 ϵ_0 規則的句子。 ϵ_0 的文法產生如下列廣大的英語句子：

John is in the pit

The wumpus that stinks is in 2 2

Mary is in Boston and the wumpus is near 3 2

不幸的是，該文法會**過度產生**(overgenerate)，也就是說，它可以產生不符合文法的句子，例如「Me go Boston」和「I smell pit gold wumpus nothing east.」。同時它也可能**生產不足**(undergenerate)：很多英語中的句子會被它排除，例如「I think wumpus is smelly.」。我們稍後將會看到如何產生較佳的文法，對於現在我們注意在現有的文法能做些什麼。

23.4 人工智慧－現代方法 3/E

\mathcal{E}_0 :	$S \rightarrow NP VP$	[0.90]	I + feel a breeze
	$S Conj S$	[0.10]	I feel a breeze + and + It stinks
NP	$\rightarrow Pronoun$	[0.30]	I
	$Name$	[0.10]	John
	$Noun$	[0.10]	pits
	$Article Noun$	[0.25]	the + wumpus
	$Article Adjs Noun$	[0.05]	the + smelly dead + wumpus
	$Digit Digit$	[0.05]	3 4
	$NP PP$	[0.10]	the wumpus + in 1 3
	$NP RelClause$	[0.05]	the wumpus + that is smelly
VP	$\rightarrow Verb$	[0.40]	stinks
	$VP NP$	[0.35]	feel + a breeze
	$VP Adjective$	[0.05]	smells + dead
	$VP PP$	[0.10]	is + in 1 3
	$VP Adverb$	[0.10]	go + ahead
$Adjs$	$\rightarrow Adjective$	[0.80]	smelly
	$Adjective Adjs$	[0.20]	smelly + dead
PP	$\rightarrow Prep NP$	[1.00]	to + the east
$RelClause$	$\rightarrow RelPro VP$	[1.00]	that + is smelly

圖 23.2 \mathcal{E}_0 的文法，其中每條規則都給出了一個片語例子。句法類別 S (句子)， NP (名詞片語)， VP (動詞片語)， $Adjs$ (形容詞表)， PP (介係詞片語)及 $RelClause$ (關係子句)

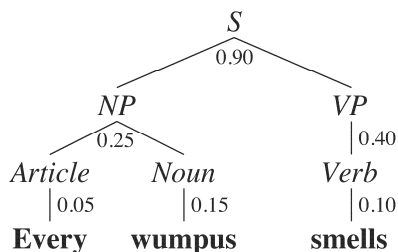


圖 23.3 根據文法 \mathcal{E}_0 的句子「Every wumpus smells」的剖析樹。樹的每個內部節點標以其機率。整個樹的機率為 $0.9 \times 0.25 \times 0.05 \times 0.15 \times 0.40 \times 0.10 = 0.0000675$ 。因為這棵樹是唯一的句法剖析，這個數值也就是這個語句的機率。這個樹也可寫為線性形式，如： $[S [NP [Article \text{every}] [Noun \text{wumpus}]] [VP [Verb \text{smells}]]]$

23.2 句法分析(剖析)

剖析是一個根據文法的規則，分析詞彙的串列來發現它的片語結構的程序。圖 23.4 表示我們從符號 S 開始，並且對於具有詞彙在葉結點的樹由上往下搜尋，或者我們可以詞彙開始並且由下往上搜尋到頂點 S 。由上而下以及由下而上剖析兩者都可能沒有效率，然而因為他們可能最後在重複搜尋空間的區域導致死胡同。考慮下面兩個句子：

Have the students in section 2 of Computer Science 101 take the exam.

Have the students in section 2 of Computer Science 101 taken the exam?

儘管它們的前 10 個詞語是相同的，但是這兩個句子卻有截然不同的剖析結果，這是因為第一個句子是一個命令句，而第二句則是一個問句。從左至右的剖析演算法必須猜測第一個詞語是否為某個命令句或是問句的組成部分，並且至少要分析到第 11 個詞語的時候——即 *take* 或者 *taken*——才能知道這個猜測結果是否正確。若演算法猜測錯誤則所有路徑將必須原路折回到第一個詞彙，並且以別的詮釋重新分析整個句子。

項目列表	規則
<i>S</i>	
<i>NP VP</i>	$S \rightarrow NP VP$
<i>NP VP Adjective</i>	$VP \rightarrow VP Adjective$
<i>NP Verb Adjective</i>	$VP \rightarrow Verb$
<i>NP Verb dead</i>	$Adjective \rightarrow \mathbf{dead}$
<i>NP is dead</i>	$Verb \rightarrow \mathbf{is}$
<i>Article Noun is dead</i>	$NP \rightarrow Article Noun$
<i>Article wumpus is dead</i>	$Noun \rightarrow \mathbf{wumpus}$
<i>the wumpus is dead</i>	$Article \rightarrow \mathbf{the}$

圖 23.4 根據文法 ϵ_0 對於作為語句的字串「The wumpus is dead」來尋找其片語的追溯步驟。從上而下地剖析，我們從 *S* 這一系列項目開始，在每一步以形式 $(X \rightarrow \dots)$ 的規則來配項目 *X*，並且以 (\dots) 更換列表中的項目 *X*。由下而上地剖析，我們從屬於句子中的詞彙之列表項目開始，且在每一步，將列表中 (\dots) 的字串匹配形式 $(X \rightarrow \dots)$ 的規則，並且將 (\dots) 換為 *X*。

為了避免這個沒效率的來源我們可使用動態規劃：每分析一個子字串，就保存結果，這樣以後就不必再重新分析該子字串。例如，我們一旦發現「the students in section 2 of Computer Science 101」是一個 *NP*，那麼我們就將這個結果記錄在一個被稱為圖(chart)的資料結構中。能執行這項功能的演算法被稱為圖剖析器(chart parser)。由於我們處理的是語境自由文法，因此在搜尋空間裡某個分支的語境中發現的任何片語在其他分支中一樣可用。有很多種類行的圖剖析器，我們描述由下而上的版本稱之為 **CYK 演算法**，由它的發明者 John Cocke、Daniel Younger 和 Tadeo Kasami 命名。

該演算法如圖 23.5。注意它需要兩個非常特別格式之一在所有的規則的文法：格式 $X \rightarrow \mathbf{word}$ 的詞彙規則，和格式 $X \rightarrow Y Z$ 的語法規律。這個文法格式稱為**杭士基正規形式**(Chomsky Normal Form)，看似個限制，但卻不是：任何文本無關文法可自動地轉換到杭士基正規形式。習題 23.8 引導讀者通過此過程。

```

function CYK-PARSE(words, grammar) returns P, a table of probabilities

  N ← LENGTH(words)
  M ← the number of nonterminal symbols in grammar
  P ← an array of size [M, N, N], initially all 0
  /* Insert lexical rules for each word */
  for i = 1 to N do
    for each rule of form (X → wordsi [p]) do
      P[X, i, 1] ← p
  /* Combine first and second parts of right-hand sides of rules, from short to long */
  for length = 2 to N do
    for start = 1 to N − length + 1 do
      for len1 = 1 to N − 1 do
        len2 ← length − len1
        for each rule of the form (X → Y Z [p]) do
          P[X, start, length] ← MAX(P[X, start, length],
                                         P[Y, start, len1] × P[Z, start + len1, len2] × p)
  return P

```

圖 23.5 CYK 演算法用於剖析。給定一個單字序列，它找出對於所有序列與每個子序列的最可能推導。它回傳整個表 P ，其中一個單元 $P[X, start, len]$ 是在 $start$ 位置開始的具有長度 len 的最可能 X 的機率。若在該位置沒有該大小的 X ，則機率為零。

CYK 演算法為表格 P 使用 $O(n^2m)$ 的空間，並且耗費 $O(n^3m)$ ，其中 n 是句子中單字的數量，而且 m 是文法中非終端符號的數目[當 m 對於一般文法是個常數時，通常描述為 $O(n^3)$]。沒有演算法可以為一般文本無關文法做得更好，即使在很多的限制文法中有更快的演算法。事實上，對於在 $O(n^3)$ 時間完成有一個把戲用於演算法，因為有可能給出個句子是有指數時間的剖析樹。試想此語句：

Fall leaves fall and spring leaves spring.

此句是有歧義的，因為每個字(除了「and」)都可為名詞或動詞，而「fall」和「spring」還可以作形容詞(例如，「Fall leaves fall」的一個意義和「Autumn abandons autumn」是相同)。在 ϵ_0 下，這句話共有 4 種剖析結果：

[S [S [NP Fall leaves] fall] 和 [S [NP spring leaves] spring]

[S [S [NP Fall leaves] fall] 和 [S spring [VP leaves spring]]

[S [S Fall [VP leaves fall]] 和 [S [NP spring leaves] spring]

[S [S Fall [VP leaves fall]] 和 [S spring [VP leaves spring]] .

如果我們有 c 個雙向歧義又可相組合的子句，那麼我們將會有 2^c 種方式來選擇子句的剖析結果^[3]。CYK 演算法如何地處理 2^c 剖析樹於 $O(c^3)$ 時間？答案是這並不會驗證所有剖析樹，所有它要做的是計算最有可能樹的機率。子樹都用 P 表格表示，並且以一點作業我們可列舉它們所有的(在指數時間)，但除非我們想要，否則以美好的 CY 演算法我們不會想列舉它們。

實際上我們通常不對所有的剖析有興趣，僅在第一個或是前幾個。考慮 CYK 演算法以「使用文法規則」運算子定義完整狀態空間。使用 A^* 搜尋，僅搜尋部分空間是可能的。在這空間的每個狀態是項目的列表(單字或範疇)，如同表示在由下而上剖析表(圖 23.4)。起始狀態是單字的列表，並且終點狀態是單一項目 S 。狀態的成本是如同目前已經應用的規則所定義它的機率的相反，並且有不同的試探法來估計到終點的剩餘距離，最佳的試探法來自於機器學習應用在語料庫判別。以 A^* 演算法我們不需要搜尋整個狀態空間，並且保證首次剖析搜尋將會是最有可能的。

23.2.1 學習 PCFG 的機率

PCFG 具有許多規則，每個規則具有它的機率。這暗示著，從資料中學習語法可能會比知識工程方法要好。若我們給出正確的剖析句子的語料庫(一般稱為樹狀庫)，學習會最容易。Penn 樹狀庫 (Marcus *et al.*, 1993) 是最為人所知，它由 3 百萬字所組成，其中以詞性和剖析樹結構標註，使用人工輔助一些自動化工具。圖 23.6 表示從 Penn 樹狀庫的有註解之樹。

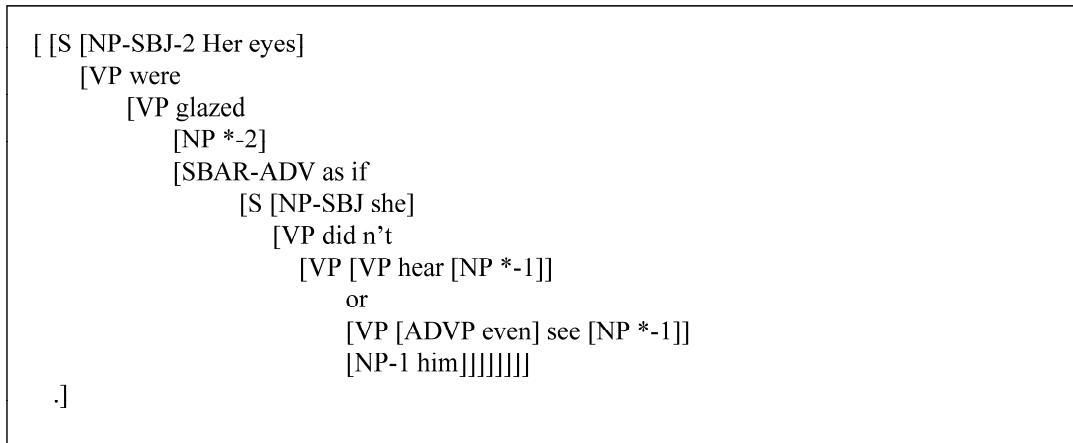


圖 23.6 對 Penn 樹狀庫的句子「Her eyes were glazed as if she didn't hear or even see him.」的註解樹。注意到，此文法中有區別受詞的名詞片語(NP)和主詞的名詞片語($NP-SBJ$)。也注意到，目前我們還沒提及的一個語法現象：一個片語從樹的一部份移動到另一部份。這個樹將片語「hear or even see him」分析為是由 2 個要素 VP 所組成，即 $[VP \text{ hear } [NP *-1]]$ 和 $[VP \text{ [ADVP even] see } [NP *-1]]$ ，兩者皆有遺失的受詞(標為 $*-1$)，其將樹中標於別處之 NP 稱為 $[NP-1 \text{ him}]$ 。

給定一個句法分析樹的語料庫，我們能夠透過計數(和平滑)建立 PCFG：在前述例子，形式 $[S[NP \dots][VP \dots]]$ 有兩個節點。我們會數這個，和語料庫中關於根 S 的所有其他子樹。若有 10 萬個 S 節點且其中在這個形式有 6 萬個，則我們建立規則：

$$S \rightarrow NP VP [0.60]$$

如果樹狀庫不存在，但我們有原先未標記的句子的語料庫為如何？從這樣的語料庫學習文法仍然是可能的，但是會比較困難。首先，我們實際上有兩個問題：學習語法規則的結構和學習與每條規則相關的機率。(在學習貝氏網路中也存在同樣的區分)。我們將假設給定詞彙的和句法的範疇名稱。(若不是，我們僅可假設範疇 X_1, \dots, X_n 和使用交叉驗證來找出 n 的最佳值)。然後可以假設文法包含每個 $(X \rightarrow YZ)$ 或 $(X \rightarrow \text{word})$ 可能規則，即使這些規則大部分都有 0 的機率或是趨近於 0。

就像在學習 HMM 中一樣，我們可以採用期望最大化(EM)方法。我們嘗試著學習的參數是規則的機率，我們以隨機或是一致的值開始。而隱變數則是句法分析樹：我們不能確定由 $w_i \dots w_j$ 組成的字串是否由規則($X \rightarrow \dots$)產生。E 步驟估計每條規則產生每個子序列的機率。然後 M 步驟估計每條規則的機率。整個計算過程以動態規劃的方式實作，所採用的是**內外演算法**(inside-outside algorithm)，類似於 HMM 中的前後演算法。

內外演算法令人不可思議的地方，在於能夠從未經過句法分析的文本中歸納出語法。但是，它有很多缺點。第一，歸納出的語法所產生的剖析往往難於理解，且不能令語言學家滿意。這使得把手工得到的知識與自動推導得到的知識結合起來非常困難。第二，該演算法速度很慢：複雜度是 $O(n^3 m^3)$ ，其中 n 是語句中單字的數目， m 是語法類別數目。其次，機率值的空間很大，在實驗中的一個嚴重問題是它會被卡在局部極大點。另一個像是模擬退火可更接近於全域最大化，在於成本或是更多的計算。Lari 和 Young(1990)推論出其中內-外是「對於真實問題棘手的計算」。

然而，若我們希望單獨地加強從未剖析的文章學習之外部限制則能夠取得進展。一個方法是從**原型學習**：以十幾種或 2 種規則來種下程序，類似於在 ϵ_1 的規則。從那裡，更複雜的規則可更容易地學習，並且從整體回顧結果的文法剖析英語，和句子的精準度約為 80%(Haghighi 和 Klein, 2006)。另一個方法是使用樹狀庫，但除了直接地從括弧學習 PCFG 規則，同樣地學習沒有位於樹狀庫之中學習差異。例如，並非在圖 23.6 的數造成介於 NP 和 NP-SBJ 之間的差異。接下來會使用代名詞「she」，前者的代名詞則使用「her」。我們將在 23.6 節發現此一問題；現在我們僅說有許多方式其中可能對於**分割**類別有用，像是 NP——文法感應系統使用樹狀庫且自動地分割類別，比起堅持原始分類集合做得更好(Petrov 和 Klein, 2007c)。自動學習文法的錯誤率仍然是大約 50%高於手動建構文法，但是這差距是逐漸降地。

23.2.2 語境自由與馬可夫模型的比較

PCFG 的問題在於它們是語境自由的。也就意味著 $P(\text{「eat a banana」})$ 和 $P(\text{「eat a bandanna」})$ 之間的差別僅僅取決於 $P(\text{Noun} \rightarrow \text{「banana」})$ 和 $P(\text{Noun} \rightarrow \text{「bandanna」})$ 之間的差異，而不依賴於「eat」和其他受詞之間的關係。二階或更高的馬可夫模型，給出個充分大的語料庫，將知道關於「eat a banana」更為可能。我們可以合併 PCFG 與馬可夫模型來得到兩者的最佳。這個簡單的方法是以藉由兩個模型所計算出來的機率之幾何平均，估計一個句子的機率。則我們會知道關於「eat a banana」是可能從文法的或是詞彙兩者的觀點。但它仍不回在「eat a slightly aging but still palatable banana」拾起介於「eat」和「banana」之間的關係，因為這裡的關係比兩個字還來得多。增加馬可夫模型的階數並不會得到準確地關係，為了達到這我們可以使用**詞彙化** PCFG，如同下一個章節所要描述的。

另外一個問題是 PCFG 傾向於對較短的語句有太強烈的偏好。在類似《華爾街日報》(*Wall Street Journal*)這樣的語料庫中，語句的平均長度大約是 25 個單字。這意味著 PCFG 會給予很多諸如「He slept」這樣的短句較高的機率，而在《華爾街日報》中我們更常見的是「It has been reported by a reliable government source that the allegation that he slept is credible」。看起來，《華爾街日報》中的片語其實不是語境自由的，而是寫作者對語句長度有一個概念，並以這個長度為他們的語句的軟性全局限制。這在 PCFG 中很難反映出來。

23.3 擴充文法和語意解釋

在本節我們看到如何延伸文本無關文法——也就是說，例如並非每個 NP 都獨立於文本，而是包含 NP 更可能出現於一個文本，而其他在另外的文本。

23.3.1 詞彙化 PCFGs

爲了得到關於動詞「eat」與名詞「banana」相對於「bandanna」之間的關係，我們可以使用詞彙化 PCFG，其中規則的機率決定於剖析樹中兩個單字之間的關係，並非僅在句子中鄰近的單字。當然，我們無法得到關於樹之中關於每一個單字的機率，因為我們不會有足夠的訓練資料來估計所有這些的機率。片語引導開頭的觀念是很有用的——最重要的單字。因此，「eat」是動詞片語(*VP*)「eat a banana」的開頭以及「banana」是名詞片語(*NP*)「a banana」的開頭。我們使用格式 $VP(v)$ 來表示一個片語其中開頭單字爲 v 分類爲動詞片語(*VP*)。我們稱這個動詞片語的分類是以開頭變數 v 的擴充。這裡是個描述著動詞-受詞關係的擴充文法：

$$VP(v) \rightarrow Verb(v)NP(n) \quad [P_1(v, n)]$$

$$VP(v) \rightarrow Verb(v) \quad [P_2(v)]$$

$$NP(n) \rightarrow Article(a) Adjs(j) Noun(n) \quad [P_3(n, a)]$$

$$Noun(banana) \rightarrow banana \quad [p_n]$$

...

在此機率 $P_1(v, n)$ 取決於 v 和 n 開頭的單字。我們會設定這個機率爲相對地高，當 v 是「eat」並且 n 是「banana」，而當 n 是「bandanna」時是相對低。注意自從我們僅考量開頭，介於「eat a banana」和「eat a rancid banana」將不會被這些機率所捕捉。另一個這方法的問題是在於，對 2 萬個名詞和 5 千個動詞， P_1 需要 1 億個機率估計。這些僅有少數比率可來自於語料庫，剩餘的部分將必須得自於平滑(參考 22.1.2 節)。例如，我們可以爲 (v, n) 對估計 $P_1(v, n)$ ，其中未曾看過時常(或根本)反饋至僅和 v 相關的模型。這些無受詞機率仍然非常有用，他們可捕捉介於及物動詞像是「eat」之間的差異——其中對於 P_1 將有較高的值且 P_2 對於低的值——和不及物動詞像是「sleep」其中將有顛倒。這是對於從樹狀庫學習這些機率十分可行的。

23.3.2 擴充文法規則的正式定義

擴充規則是複雜的，所以將給它們一個正式的定義，藉由表現出如何將擴充規則可轉換到邏輯句子。這句子將有確定子句的形式(參考第 7.5.3 節)，因此結果稱爲確定子句文法或 DCG。我們將使用個例子的規則版本，以一個新的符號片段，從名詞片語的詞彙化文法：

$$NP(n) \rightarrow Article(a) Adjs(j) Noun(n) \{Compatible(j, n)\}$$

新的層面在這是個符號 $\{constraint\}$ 來標記邏輯限制於某些變數，這規則僅在限制為真的時候成立。在此預測 $Compatible(j, n)$ 是意味著測試無論形容詞 j 和名詞 n 是相容的，它將被一連串像是 $Compatible(black, dog)$ 的斷言所定義。我們可轉換這個文法規則至明確的條文，藉著：(1) 迴轉左右手方向順序；(2) 造成所有的組成和制約連結在一起；(3) 增加變數 s_i 至參數的列表對於每個表達藉由組成展開的單字序列之組成；(4) 為單字的串連增加項目， $Append(s_1, \dots)$ ，增加到參數的列表到樹的根。這給了我們

$$Article(a, s_1) \wedge Adjs(j, s_2) \wedge Noun(n, s_3) \wedge Compatible(j, n) \Rightarrow NP(n, Append(s_1, s_2, s_3))$$

這明確的條文提到若預測的 $Article$ 對於開頭單字 a 和字串 s_1 為真，並且 $Adjs$ 對於開頭單字 j 和字串 s_2 相似地為真，以及 $Noun$ 對於開頭單字 n 和字串 s_3 為真，而且若 j 和 n 是相容的，則預測 NP 對於開頭單字 n 和附加字串 s_1 、 s_2 和 s_3 的結果為真。

DCG 轉換遺漏了機率，但可將它們補回去：增加每個成分以更多的變數來表達成份的機率，並且增加根部的變數其中是成分機率乘上規則機率的結果。

從文法規則轉換到明確條文，讓我們討論到像是邏輯推理的剖析。這使得我們能夠用多種方式對語言和字串進行推論。例如，它意味著我們可以用前向連結進行由下而上的剖析，或是用後向連結進行由上而下的剖析。事實上，以 DCG 剖析自然語言是 Prolog 邏輯程式語言最先的應用之一。有時可以反向執行程序和剖析一樣執行語言生成。例如，忽略圖 23.10 前頭，邏輯程式可給出個語意形式 $Loves(John, Mary)$ 和應用明確條文規則來推演

$$S(Loves(John, Mary), [John, loves, Mary])$$

這對小型的例子有用，但重要的語言生成系統需要透過程序控制，則單獨地由 DCG 規則所提供。

23.3.3 格位一致與主詞-動詞一致

我們看到在 23.1 節簡單的文法對於 ε_0 過度產生，產生非句子像是「Me smell a stench.」。為了避免這個問題，我們的文法必須知道當「me」作句子的主詞時，「me」不是合法的 NP 。語言學家主張代名詞「I」是主格^[4]，而「me」是受格。我們可以透過將 NP 分成 NP_S 及 NP_O 這兩類來修正這個問題，它們分別表示主格和受格的名詞片語。我們還需要把 $Pronoun$ 範疇劃分為兩個範疇 $Pronoun_S$ (包括「I」) 和 $Pronoun_O$ (包括「me」)。圖 23.7 的上半部顯示了格位一致的文法，我們稱產生出的語言為 ε_1 。應該注意的是，所有 NP 的規則都必須加倍，一份用於 NP_S ，一份用於 NP_O 。

不幸的是， ε_1 仍然存在過度產生的問題。英語句子中對於主詞的人稱或數目和主要動詞需要是主詞-動詞一致。例如，如果「I」是主詞，那麼「I smell」是合文法的，而「I smells」則否。如果「it」是主詞，那麼結果正好相反。在英語中，一致性區別是最小的：大多數動詞針對第三人稱單數作主詞的情況(he, she, 或 it)有一種形式，而對其他所有的人稱和數量的組合是另一種形式。有一個例外：動詞「to be」有三個形式，「I am / you are / he is」。因此一個差異(格位)將名詞片語分為 2 類，另一個差異(人稱和數目)將名詞片語區 NP 分為 3 類，並且未含蓋到其他差異，我們將最終以名詞片語形式下標變量的指數值結束，若使用 ε_1 的方法。擴增是較佳的方法：他們可代表一個指數數目形式，作為單一的規則。

$\mathcal{E}_1 :$	$S \rightarrow NP_S VP \mid \dots$
	$NP_S \rightarrow Pronoun_S \mid Name \mid Noun \mid \dots$
	$NP_O \rightarrow Pronoun_O \mid Name \mid Noun \mid \dots$
	$VP \rightarrow VP NP_O \mid \dots$
	$PP \rightarrow Prep NP_O$
	$Pronoun_S \rightarrow \mathbf{I} \mid \mathbf{you} \mid \mathbf{he} \mid \mathbf{she} \mid \mathbf{it} \mid \dots$
	$Pronoun_O \rightarrow \mathbf{me} \mid \mathbf{you} \mid \mathbf{him} \mid \mathbf{her} \mid \mathbf{it} \mid \dots$
	\dots
<hr/>	
$\mathcal{E}_2 :$	$S(head) \rightarrow NP(Sbj, pn, h) VP(pn, head) \mid \dots$
	$NP(c, pn, head) \rightarrow Pronoun(c, pn, head) \mid Noun(c, pn, head) \mid \dots$
	$VP(pn, head) \rightarrow VP(pn, head) NP(Obj, p, h) \mid \dots$
	$PP(head) \rightarrow Prep(head) NP(Obj, pn, h)$
	$Pronoun(Sbj, 1S, \mathbf{I}) \rightarrow \mathbf{I}$
	$Pronoun(Sbj, 1P, \mathbf{we}) \rightarrow \mathbf{we}$
	$Pronoun(Obj, 1S, \mathbf{me}) \rightarrow \mathbf{me}$
	$Pronoun(Obj, 3P, \mathbf{them}) \rightarrow \mathbf{them}$
	\dots

圖 23.7 上半部： \mathcal{E}_1 語言文法的一部分，其處理名詞片語中的主格和受格，因此不會造成嚴重如 \mathcal{E}_0 的過產生。其中省略了與 \mathcal{E}_0 相同的部分。下半部：對 \mathcal{E}_2 的擴充文法的一部份，有 3 個擴增：格位一致、主詞-動詞一致、開頭單字。 Sbj 、 Obj 、 $1S$ 、 $1P$ 和 $3P$ 是常數，且小寫名稱為變數。

在圖 23.7 的底下我們看到(部分的)一個擴展語法的語言 \mathcal{E}_2 ，其中處理格位一致、主詞-動詞一致和開頭單字。我們只有個名詞片語範疇，而 $NP(c, pn, head)$ 具有 3 個擴增： c 是對於格位的參數， pn 是人稱和數目的參數，且 $head$ 是個片語的開頭單字。其他的範疇也是以開頭和其他論證來擴展。讓我們仔細考慮一個規則：

$$S(head) \rightarrow NP(Sbj, pn, h) VP(pn, head)$$

這條規則是由右至左最容易理解的：當名詞片語和動詞片語合併形成 S ，但僅若名詞片語具有主格(Sbj)，且名詞片語和動詞片語的人稱和數量(pn)是相同的。若這個成立，則我們有個 S 其中開頭是跟動詞片語的開頭相同。注意到名詞片語的開頭，由虛擬變數 h 所註記的，並非 S 擴展的一部份。對於 \mathcal{E}_2 的詞彙規則填寫參數的值並且也最好從右至左讀取。例如，規則

$$Pronoun(Sbj, 1S, \mathbf{I}) \rightarrow \mathbf{I}$$

「I」在第一人稱單數主格可解釋為代名詞，當聽到「I」。為了簡單起見我們忽略對於這些規則的機率，但是擴增是以機率作用著。擴增也可作用於自動學習機制。Petrov 和 Klein(2007c)表示學習演算法如何可自動將名詞片語分裂為 NP_S 和 NP_O 的範疇。

23.3.4 語意解釋

爲表示如何增加語意於文法中，我們從一個比英語還簡單的例子開始：語意的算數表示。圖 23.8 表示個算數表示的文法，其中每個規則是以變數擴增指出片語的語意解釋。數字的語意像是「3」是數字本身。表示式的語意像是「3 + 4」運算元「+」用於片語「3」和片語「4」的語意。這規則遵守著合成語意的原則——片語的語意是個分詞的語意函數。圖 23.9 表示根據此文法對於 $3 + (4 \div 2)$ 的剖析樹。這棵剖析樹的根節點是 $Exp(5)$ 這個運算式，其語意解釋是 5。

$$\begin{aligned}
 Exp(x) &\rightarrow Exp(x_1) \text{ Operator}(op) Exp(x_2) \{x = \text{Apply}(op, x_1, x_2)\} \\
 Exp(x) &\rightarrow (Exp(x)) \\
 Exp(x) &\rightarrow Number(x) \\
 Number(x) &\rightarrow Digit(x) \\
 Number(x) &\rightarrow Number(x_1) Digit(x_2) \{x = 10 \times x_1 + x_2\} \\
 Digit(x) &\rightarrow x \{0 \leq x \leq 9\} \\
 Operator(x) &\rightarrow x \{x \in \{+, -, \div, \times\}\}
 \end{aligned}$$

圖 23.8 數學運算式的文法，擴充了語意部分。每個變數 x_i 表示一個成分的語意。注意到，使用 $\{test\}$ 標記法是為了定義必須滿足的邏輯述詞，但那並不是成分

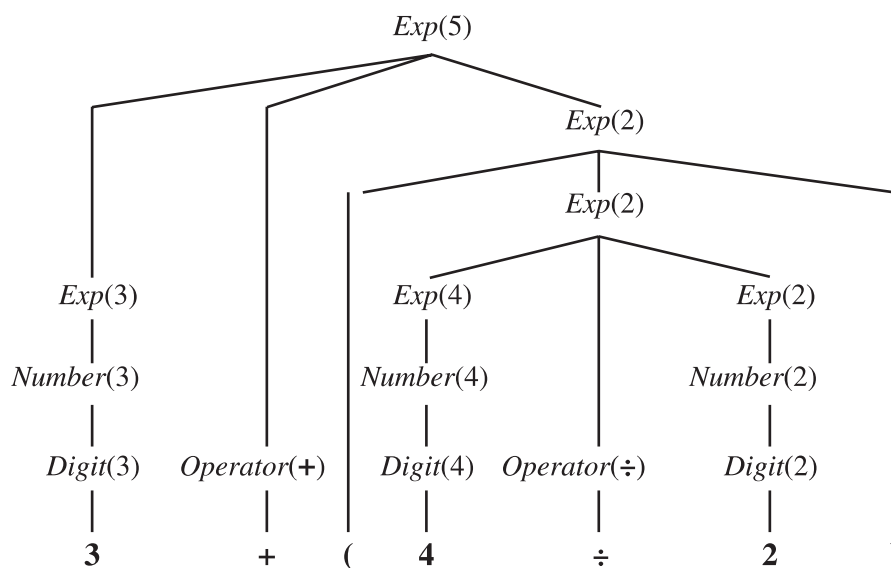


圖 23.9 對字串「3 + (4 ÷ 2)」的語意解釋的剖析樹

現在讓我們進入英語的語意或是至少 ϵ_0 。我們從決定哪些語意表示法要關聯至哪些片語開始。我們採用最簡單的例句「John loves Mary」。NP「John」的語意表示法應為邏輯項 *John*，而整個句子的詮釋則為邏輯句子 *Loves(John, Mary)*。這些部分似乎是很明顯的。複雜的部分在於 VP「loves Mary」。這個片語的語意詮釋既不是一個邏輯項，也不是一個完整的邏輯句子。直覺上，「loves Mary」是一個敘述，可以用於或不用於特定人物。(在這個範例中，它用於 John)。這表示「loves Mary」是一個述詞(predicate)，當它與表示某個人的邏輯項(愛的行為主體人)相結合時，就會產生一個完整的邏輯句子。藉由使用 λ -標記法(參見 8.2.3 節)，我們可將「loves Mary」表為述語。

$$\lambda x \text{ Loves}(x, \text{Mary})$$

現在我們需要一條規則，說明「具有語意 *obj* 的一個 *NP* 後面跟隨一個具有語意 *rel* 的 *VP* 會產生一個句子，該句子的語意就是將 *rel* 套用於 *obj* 的結果」：

$$S(\text{pred}(\text{obj})) \rightarrow NP(\text{obj}) VP(\text{pred})$$

這條規則告訴我們「John loves Mary」的語意詮釋是

$$(\lambda x \text{ Loves}(x, \text{Mary}))(\text{John})$$

這與 $\text{Loves}(\text{John}, \text{Mary})$ 是等價的。

其餘的語意則是根據我們到目前為止所做的選擇以直接的方式得到的。因為 *VP* 被表示為述詞，所以把動詞也一致地表示為述詞不失為一個好主意。動詞「loves」被表示為 $\lambda y \lambda x \text{ Loves}(x, y)$ ，如果已知了參數 *Mary*，那麼則回傳述詞 $\lambda x \text{ Loves}(x, \text{Mary})$ 。我們以圖 23.10 所示的文法以及圖 23.11 所示的剖析樹作為本小節的結束。我們可僅輕易地增加語意到 ε_2 ，選擇與 ε_0 所以讀者可以每次集中於擴展的其中一個類型。

以手工加入語意擴展至文法是吃力並且容易出錯。因此，有數個專案從實例來學習語意擴展。CHILL(Zelle 和 Mooney, 1996)是一個歸納邏輯程式設計(ILP)的程式，它能夠從實例中學習文法以及針對該文法的專門剖析器。其目標領域是以自然語言進行資料庫查詢。訓練樣本由詞語串以及相對應的查詢配對組成——例如：

哪一個國家的首都有最大的人口密度？

$$\text{Answer}(c, \text{Capital}(s, c) \wedge \text{Largest}(p, \text{State}(s) \wedge \text{Population}(s, p)))$$

CHILL 的任務是學習一個述詞 $\text{Parse}(\text{words}, \text{query})$ ，讓它能與範例一致，並希望能將其一般化，套用於其他實例。直接用 ILP 學習這個述詞的效能很差：歸納出的剖析器正確率只有約 20%。幸運的是，ILP 學習器可透過新增知識而得到改進。在這種情況下， Parse 述詞的大半部分都被定義為邏輯程式，而 CHILL 的任務則簡化為歸納控制規則，使其能夠導引剖析器挑選出合適的剖析結果。透過這個額外的背景知識，CHILL 能夠在不同的資料庫查詢工作中達到 70%到 85%的準確率。

$$\begin{aligned} S(\text{pred}(\text{obj})) &\rightarrow NP(\text{obj}) VP(\text{pred}) \\ VP(\text{pred}(\text{obj})) &\rightarrow \text{Verb}(\text{pred}) NP(\text{obj}) \\ NP(\text{obj}) &\rightarrow \text{Name}(\text{obj}) \end{aligned}$$

$$\begin{aligned} \text{Name}(\text{John}) &\rightarrow \mathbf{John} \\ \text{Name}(\text{Mary}) &\rightarrow \mathbf{Mary} \\ \text{Verb}(\lambda y \lambda x \text{ Loves}(x, y)) &\rightarrow \mathbf{loves} \end{aligned}$$

圖 23.10 能夠推導出「John loves Mary」(及另外 3 個句子)的剖析樹和語意解釋的文法。每個範疇都使用一個表示語意的單一擴充

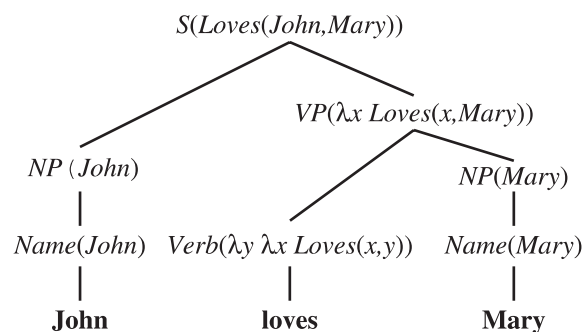


圖 23.11 對字串「John loves Mary」的語意解釋的一棵剖析樹

23.3.5 複雜因素

實際英語的文法是無止境地複雜。我們將簡單地提及一些實例。

■ 時間和時態

現在假設我們想要表示「John loves Mary」和「John loved Mary」之間的區別。英語採用動詞的時態(過去式、現在式和未來式)來顯示與一個事件相關的時間。一個表示事件時間的好選擇是 12.3 節中的事件演算標記法。在事件計算有

John loves mary: $E1 \in Loves(John, Mary) \wedge During(Now, Extent(E1))$

John loved mary: $E2 \in Loves(John, Mary) \wedge After(Now, Extent(E2))$

這提示了我們對於詞語「loves」和「loved」的辭彙規則應該是：

$Verb(\lambda y \lambda x e \in Loves(x, y) \wedge During(Now, e)) \rightarrow \text{loves}$

$Verb(\lambda y \lambda x e \in Loves(x, y) \wedge After(Now, e)) \rightarrow \text{loved}$

除了這些改變之外，文法中的其餘內容都保持不變，這是十分令人振奮的；這顯示如果我們能夠輕易地新增一些動詞時態之類的複雜內容，那麼我們就走對方向了(儘管我們只是蜻蜓點水地討論了表示時間及時態的完整文法)。這也鼓勵了程序和離散事件之間的差異，其中在 12.3.1 節我們討論到知識表達是會實際地反映到所用的語言。我們可說「John slept a lot last night」其中 *Sleeping* 是個程序範疇，但說「John found a unicorn a lot last night」則是很怪的，其中 *Finding* 是個離散事件範疇。文法會反映出由對於離散事件增加副詞片語「a lot」具有低的機率這個事實。

■ 量化

考慮句子「Every agent smells a wumpus」。這句子在 ε_0 底下有唯一的句法剖析，但它這個句子實際上是有歧義的：它的優先含義是「對於每個代理人存在著代理人感覺到的氣息」，另一個可接受的意思是「存在著每個代理人感覺到的氣息」^[5]，兩個解釋可被表示為

$\forall a a \in Agents \Rightarrow \exists b b \in Breezes \wedge \exists e e \in Feel(a, b) \wedge During(Now, e)$;

$\exists b b \in Breezes \forall a a \in Agents \Rightarrow \exists e e \in Feel(a, b) \wedge During(Now, e)$

標準的量化方法是為文法定義個非實際邏輯語意句子，而是藉由演算法以外的剖析程序轉變為邏輯句子的**準邏輯形式**。這些演算法有優先規則比起另一個偏好的計量器範圍——偏好不需要直接地反映到文法。

■ 語用

我們已經說明了一個代理人如何能夠感知一個由詞語組成的字串，並使用文法來導出一組可能語意詮釋。現在，我們要處理的問題是，透過為每個詮釋選項，增加關於當前狀況的語境從屬資訊，來完成解釋。最明顯的需要語用資訊的地方在於解決**索引詞**(indexical)的含義，這是一些直接涉及當前處境的片語。例如，一個句子「I am in Boston today」是由「I」和「today」所標示的。單字「I」會被表示為一位流暢的演說者，這會取決於聽者來解讀流暢的意思——這不是一個文法的考量部分而是流暢的問題，使用現有狀態的文本來說明流暢。

語用的另一部份是說明講者的意向。說話者行動被視為**言詞行動**(speech act)，並且取決於聽者如何解讀它是什麼樣的行動——一個問題、敘述、承諾、警告、命令...等等。類似「go to 2.2」的命令句隱含地指涉到聆聽者。到目前為止，關於 *S* 的文法僅僅包含了陳述句。我們可以很容易地擴展它以涵蓋命令句。命令句是從 *VP* 形成的，其中的主詞已隱含為聆聽者。我們需要區分命令句和陳述句，所以我們對 *S* 的規則進行修改，以包含言辭行動：

$$S(\text{Statement}(\text{Speaker}, \text{pred}(\text{obj}))) \rightarrow NP(\text{obj}) VP(\text{pred})$$

$$S(\text{Command}(\text{Speaker}, \text{pred}(\text{Hearer}))) \rightarrow VP(\text{pred})$$

■ 遠距離從屬資訊

問題引導了一個文法上的複雜度。在「Who did the agent tell you to give the gold to?」的最後一個字「to」應該被剖析為[PP to \square]，其中「 \square 」註記為名詞片語所缺少的差距或是**軌跡**，缺少的名詞片語是由句子中第一個字「who」所授權的。一個複雜的擴增系統被用於確保缺少名詞片語和許可詞在正確的方法是匹配的，並且禁止在錯誤地方之差距。例如，不能在一個名詞片語連接的分支有差距：「What did he play [NP Dungeons and \square]?»是不合乎語法的。但可以在名動詞片語的分支有相同的差距：「What did you [VP [VP smell \square] and [VP shoot an arrow at \square]]?»

■ 歧義

某些情況下，聆聽者能意識到發言中所包含的歧義。以下列出有一些從報紙中摘錄的頭條範例：

Squad helps dog bite victim.

Police begin campaign to run down jaywalkers.

Helicopter powered by human flies.

Once-sagging cloth diaper industry saved by full dumps.

Portable toilet bombed; police have nothing to go on.

Teacher strikes idle kids.

Include your children when baking cookies.

Hospitals are sued by 7 foot doctors.

Milk drinkers are turning to powder.

Safety experts say school bus passengers should be belted.

不過大多數時候我們所聽到的語言似乎是沒有歧義的。因此，在 20 世紀 60 年代，當研究人員開始使用電腦對語言進行分析的時候，他們對此非常驚訝：儘管對於以該語言為母語的人而言，許多歧義都並不明顯，然而幾乎每句發言都含有很強的歧義。一個使用大型文法和詞典的系統可能會為一個非常普通的句子找到上千種解釋。**辭彙歧義**很常見：「back」可以是一個副詞(go back)、形容詞(back door)、名詞(the back of the room)或是動詞(back up your files)。「Jack」則可以是一個人名、一個名詞(一張紙牌名、拋接子遊戲、航海用的旗子、狗魚、公驢、插座或者是起重設備)或者是一個動詞(頂起汽車、用燈打獵、用力揮擊棒球)。**句法歧義**指出片語有多重的剖析：「I smelled a wumpus in 2.2」

具有 2 個剖析：其中一個是介系詞片語「in 2,2」修飾了名詞和另一個是修飾了動詞。句法歧義導致了語意歧義(semantic ambiguity)，因為其中一種剖析意味著 wumpus 出現在 2,2 的位置，而另一種分析則意味著在 2,2 的位置有臭氣。在這種情況下，採用錯誤的解釋可能造成致命的失誤。

最後，在字面含義和喻義中也會存在歧義。比喻在詩歌中很重要，不過很令人驚訝的是，在日常用語中也一樣常見。轉喻(metonymy)是用一種事物表示另外一種事物的修辭手法。當我們聽到「Chrysler announced a new model(克萊斯勒公司公佈了一種新型車)」時，我們並不會將其解釋為公司能講話，相反地，我們知道這是代表該公司的發言人宣佈了這個公告。轉喻很常見而且經常被人類聆聽者下意識地進行解釋。不幸的是，我們所寫出的文法卻不易做到這點。為了正確地處理轉喻的語意，我們需要引入全新層次的歧義。我們透過為句子中的每個片語的語意解釋提供兩個物件來實作上述工作：一個是片語在字面上指涉的物件(Chrysler)，另一個是轉喻指涉的對象(即發言人)。那麼，我們就可以說在此二者之間存在某種關係。在我們目前的文法中，「Chrysler announced」被解釋成

$$x = \text{Chrysler} \wedge e \in \text{Announce}(x) \wedge \text{After}(\text{Now}, \text{Extent}(e))$$

我們需要把它變成

$$x = \text{Chrysler} \wedge e \in \text{Announce}(m) \wedge \text{After}(\text{Now}, \text{Extent}(e)) \wedge \text{Metonymy}(m, x)$$

上式的意思是，存在一個等同於 Chrysler 的實體 x ，和另外一個可以發佈公告的實體 m ，這二者之間存在轉喻關係。下一個步驟是定義能夠發生何範疇的轉喻關係。最簡單的情況是不存在轉喻——字面物件 x 和轉喻物件 m 是相同的：

$$\forall m, x (m = x) \Rightarrow \text{Metonymy}(m, x)$$

在 Chrysler 這個例子中，一種合理的一般化方法是一個組織可以被用於代表其發言人：

$$\forall m, x (x \in \text{Organizations} \wedge \text{Spokesperson}(m, x) \Rightarrow \text{Metonymy}(m, x))$$

轉喻還包括用作者指涉作品(如：I read *Shakespeare*)，或者更一般的是用生產者指涉產品(如：I drive a *Honda*)，以及用部分指涉整體(如：The Red Sox need a strong *arm*)。某些類似「The *ham sandwich* on Table 4 wants another beer」這樣的轉喻例子則比較奇特，需要參考語境才能解釋。

隱喻(metaphor)是將有某種字面含義的片語，透過類推的方式用於暗示另一種涵義的修辭法。因此，隱喻可被視為一種相似關係的其中之一的借代。

排歧是個最有可能話語本意的回復程序。在某種意義上我們已經具有解決此問題的框架：每個規則具有與它相關的機率，所以解釋的機率是導致解釋規則的機率乘積。不幸地，這機率反映出學到的文法是在語料庫中很一般的片語，因此反應一般的知識，並非現行狀態的特定知識。為了正確地消除歧義，我們需要合併 4 個模型：

1. 世界模型(world model)：

一個命題在世界中發生的可能性。給定我們所知的相關世界，演講者有可能說「I'm dead」表示「I am in big trouble」，而不是「My life ended, and yet I can still talk」。

2. 心智模型(mental model)：

說話者形成欲將事實傳達給聆聽者之意圖的可能性。這個方法結合了有關說話者相信什麼的模型，以及說話者相信聆聽者相信什麼的模型等等，依此類推。例如，在聽到某政治家說「I am not a crook」(我不是騙子)後，我們可能只會賦予命題「該政治家不是罪犯」50%的機率，而把 99.999% 的機率賦給命題「該說話人不是一根彎曲的牧羊人手杖」。然而，我們選擇前者解釋因為它是較有可能說的事。

3. 語言模型(language model)：

已知說話者具有傳達某個特定事實的意圖，某個特定的詞語字串被選中的可能性。

4. 聲學模型(acoustic model)：

已知說話者已經選擇了一個特定的詞語字串，則產生一個特定聲音序列的可能性。第 23.5 節涵蓋了語音辨識。

23.4 機器翻譯

機器翻譯是把用某種自然語言(原文語言)寫成的文本自動翻譯成另外一種語言(目的語言)。這是對於電腦最先應用領域展望(Weaver, 1949)，但僅在過去的數十年這項技術才被廣泛地使用。底下這是本書第 1 章第 1 頁的一小段：

人工智慧是個在科學與工程之中的最新領域。真正的研究工作在第二次世界大戰結束後迅速展開，直到 1956 年，它被正式命名為「人工智慧」。如同分子生物學，AI 往往被其他領域的科學家譽為「我最想參與的研究領域」。

(AI is one of the newest fields in science and engineering. Work started in earnest soon after World War II, and the name itself was coined in 1956. Along with molecular biology, AI is regularly cited as the “field I would most like to be in” by scientists in other disciplines.)

在此使用 Google 翻譯的線上工具來做翻譯：

AI er en af de nyeste områder inden for videnskab og teknik. Arbejde startede for alvor lige efter Anden Verdenskrig, og navnet i sig selv var opfundet i 1956. Sammen med molekylær biologi, er AI jævnligt nævnt som “feltet Jeg ville de fleste gerne være i” af forskere i andre discipliner.

對於不會丹麥語，這裡可將丹麥語翻譯回英語。不同的單字以斜體字表示：

AI is one of the newest fields *of* science and engineering. Work *began* in earnest *just* after the *SecondWorldWar*, and the name itself was *invented* in 1956. *Together* with molecular biology, AI is *frequently mentioned* as “field I would most like to be in” by *researchers* in other disciplines.

差異都是所有合理的改述，像是經常地被提及對於定期被引用。唯一實際錯誤是冠詞 *the* 的省略，被註記為 \sqsubset 的符號。這是典型精準度：在兩個句子，對於一個說母語的人不會犯這樣的錯誤，而意思可以清楚地傳達。

從歷史的角度，對於機器翻譯有 3 個主要的應用。粗略翻譯(Rough translation)，如同免費的線上服務所提供，給出個外語句子或文件的「要點」，但包含錯誤。預編輯翻譯(Pre-edited translation)是用在公司要以多重語言發表他們的文件和銷售商品。原始來源文件是以受限的語言所撰寫，其會較易於自動地翻譯，並且結果通常是由人類來修正任何錯誤。限定來源翻譯(Restricted-source translation)完全自動地運作，但僅在高度刻板語言，像是氣象報告。

翻譯是困難的，因為在完全一般的情況，需要對文本有深入的瞭解。即使對於非常簡單的文本而言——甚至只包含一個單字的「文本」——也是困難的。考慮出現在某個商店門上的單字「Open」^[6]。它傳達的意思是當時該商店是允許顧客進入的。現在考慮同一個單字「Open」，它出現在某個新近建好的商店外的大橫幅上。它意味著該商店現在進入日常運營，但是如果晚上商店關門後而沒有拆除該橫幅，那麼看到它的人也不會被誤導。這兩個旗標用了同樣的單字卻表達了不同的含義。在德國，門上的標記是「Offen」，而橫幅上的標記則是「Neu Eröffnet」。

問題在於不同的語言對世界的分類是不同的。例如，法語單詞「doux」涵蓋了寬泛的含義，近似對應於英語單詞「soft」、「sweet」以及「gentle」等。相似的，英語單詞「hard」事實上涵蓋了德語單詞「hart」（身體對抗的，殘酷的）的所有用法以及單詞「schwierig」（困難的）的部分用法。因此，表示語句的含義對翻譯而言比對單一語言的理解更困難。單一語言的句法分析系統可以使用類似 $Open(x)$ 的謂詞，但是對於翻譯，其表示語言要必須進行更多區別，也許用 $Open_1(x)$ 表示「Offen」的含義，而 $Open_2(x)$ 表示「Neu Eröffnet」的含義。能夠在語言集合中進行所有必要區分的表示語言被稱為中間語言(interlingua)。

一個(人類或機器)翻譯員通常需要瞭解敘述於來源的實際狀況，並非僅獨立單字。例如，要將英語單字的「him」翻譯成韓語，就必須選擇是謙卑語或是尊敬語的形式，這樣的選擇取決於講者和談話對象「him」的之間的社交關係。在日本語，尊敬語是相對的，所以選擇是取決於講者、被談論者和聽者之間的社交關係。有時翻譯者(包括機器和人類)會發現很難作選擇。另一個例子，翻譯「The baseball hit the window. It broke.」成為法語，對於「it」我們必須選擇陰性「elle」或是陽性「il」，所以我們必須決定「it」為棒球或是窗戶。為了正確翻譯，其中除了語言之外必須瞭解實體。

有時對於可產生完全滿意的翻譯是沒有得選擇。例如，義大利語的情詩使用陽性的「il sole」(sun)和陰性「la luna」(moon)將兩位戀人符號化，當要翻譯至德文將會需要些變換，其中名詞的性被保留的，並且當翻譯成一種名詞的性是相同的語言有進一步的改變^[7]。

23.4.1 機器翻譯系統

所有的翻譯系統必須塑模來源和目標語言，但是系統隨著它們所用得模型類型有所不同。一些系統嘗試分析原文語言文本，一直到將其表示為中間語言，然後從中間語言表示再產生目標語言的語句。這是困難的，因為它包含了三個未解決的問題：建立完整知識來表達任何事，剖析表達且從該表達生成句子。

有的系統是基於轉換模型(transfer model)。利用翻譯規則(或是實例)資料庫，一旦規則(或實例)能夠匹配，就直接翻譯。轉換可以發生在詞法、句法以及語義層次上。例如，一條嚴格的句法規則將英語的 [Adjective Noun]([形容詞 名詞])映對為法語的 [Noun Adjective]([名詞 形容詞])。一條混合的句法和詞法規則將法語的 [S₁ “et puis” S₂] 映對為英語的 [S₁ “and then” S₂]。圖 23.12 圖解了各種轉換點。

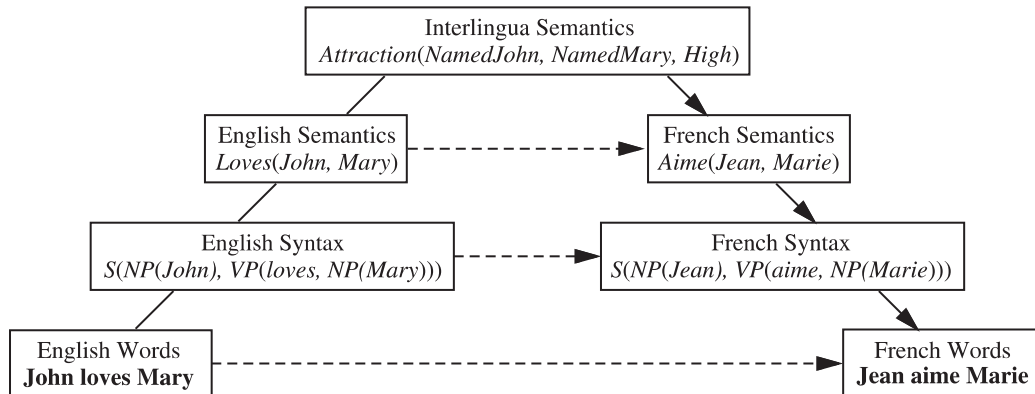


圖 23.12 沃古瓦(Vauquois)三角形：機器翻譯系統中各種選擇的示意圖(Vauquois, 1968)。我們從頂部的英語文本開始。中間語言導向的系統是沿著實線前進，首先對英語進行剖析成為句法形式，接著表示為語義及中間語言，然後產生法語的語義、句法及辭彙形式。基於轉換的系統利用虛線作為捷徑。不同系統在不同的點進行轉換；有些系統可以在多個點進行轉換

23.4.2 統計機器翻譯

現在我們已經看過翻譯任務可能如何地複雜，應該不會驚訝於許多成功的機器翻譯系統，是藉由使用統計蒐集巨大的文件語料庫之訓練機率模型。這方法不需要國際語觀念的複雜本體，也不需要手工製作來源的文法和目標語言，或是手工標示樹狀庫。所有它需要的是資料——從可能已學習的翻譯模型中樣本翻譯。為了翻譯一個句子，English(e)至 French(f)，我們找到單字 f^* 的字串最大化

$$f^* = \underset{f}{\operatorname{argmax}} P(f | e) = \operatorname{argmax} P(e | f)P(f)$$

因數 $P(F)$ 是法語的語言模型(language model)，它的含義是一個給定的語句出現在法語中的可能性有多大。 $P(e | f)$ 是翻譯模型(translation model)，它的含義是給定一個法語語句，某個英語語句作為其翻譯結果的可能性。相同地， $P(f | e)$ 是一個翻譯模型由 English 到 French。

我們是否該直接地實行於 $P(f | e)$ ，或應用貝氏法則並且實行在 $P(f | e)P(F)$ ？在像是醫療的診斷應用，在因果關係方向易於塑模其領域： $P(\text{symptoms} | \text{disease})$ 而不是 $P(\text{disease} | \text{symptoms})$ 。但雙方向的翻譯是十分地容易。在統計機器翻譯的早期研究中，應用貝氏法則——部分地因為研究者有好的語言模型 $P(f)$ ，且希望利用它，而且在部分因為他們的背景是來自於語音辨識，其中牽涉診斷問題。在本章隨著他們的引導，但我們注意到近期於統計機器翻譯的研究通常直接地 $P(f | e)$ 最佳化，使用更為複雜的模型，其中考慮到許多由語言模型的特徵。

語言模型 $P(f)$ ，可以解決任何在圖 23.12 右手邊的 level(s)，但是在最早期和最一般的方法是從法語語料庫建立個 n 元模型，如同先前已經看過的。這僅擷取部分，區域理想的法語句子，然而這對於粗略翻譯經常是充分的^[8]。

翻譯模型是從**雙語語料庫**學習——平行文本的蒐集，每對英語/法語。現在，若我們有無限大的語料庫，翻譯個句子僅需要做查詢的工作：在語料庫中我們應該已經看過英語句子，所以可只回傳對應的法語句子。但是當然我們的資源是有限，並且被要求翻譯的多數句子會是小說。然而它們會是之前已看過**片語**的組成(即使若某些片語僅是短短的一個單字)。例如，在這本書一般片語包含「in this exercise we will」、「size of the state space」、「as a function of the」和「notes at the end of the chapter」。若要求翻譯小說句子「In this exercise we will compute the size of the state space as a function of the number of actions」為法語，我們會需要能將句子拆成片語，在英語語料庫中尋找該改片語(本書)，找出相對應的法語片語(從本書法語版的翻譯)，並且以合理的法語語順重組這些法語片語。換句話說，給定個來源的英語句子 e ，找出個法語翻譯 f 是關於三個步驟：

1. 將英語句子拆散為片語 e_1, \dots, e_n 。
2. 對於每個片語 e_i 選定個相對應的法語片語 f_i 。我們使用 $P(f_i | e_i)$ 的符號代表片語機率，其中 f_i 是來自於 e_i 的翻譯。
3. 選擇個片語的排列 f_1, \dots, f_n 。我們將會以一個方式說明這個看起來有點難懂的排列，但是設計有一個簡單的機率分佈：對於每個 f_i ，我們選擇一個**失真** d_i ，這是個單字的號碼其中片語 f_i 的移動相對於 f_{i-1} ，正是往右移動，負是往左移動，並且零是 f_i 緊跟著 f_{i-1} 。

圖 23.13 表示程序的實例。在頂端，句子「There is a smelly wumpus sleeping in 2 2」是被拆散為 5 個片語 e_1, \dots, e_5 。它們每個都被翻譯至相對應的片語 f_i ，並且排列為順序 f_1, f_3, f_4, f_2, f_5 。我們解釋這個排列根據每個法語片語的失真 d_i ，定義為

$$d_i = \text{START}(f_i) - \text{END}(f_{i-1}) - 1$$

其中 $\text{START}(f_i)$ 是在法語句子中片語 f_i 的第一個單字的序數，且 $\text{END}(f_{i-1})$ 是片語 f_{i-1} 的最後一個字的序數。在圖 23.13 我們看到 f_5 ，「à 2 2」緊跟著 f_4 ，「qui dort」因此 $d_5 = 0$ 。片語 f_2 ，然而移動一個字到 f_1 的右邊，所以 $d_2 = 1$ 。像是特別的例子我們有 $d_1 = 0$ ，因為 f_1 起始於位置 1，且 $\text{END}(f_0)$ 是定義為 0(即使 f_0 不存在)。

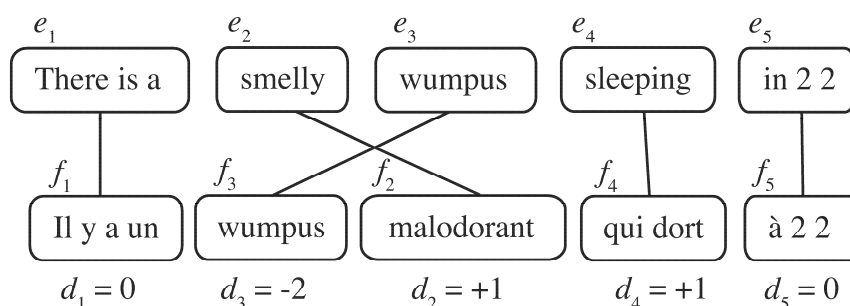


圖 23.13 候選法語片語對應於一個英語句子中的各個片語，以 $\text{distortion}(d)$ 的值代表每個法語片語

現在我們已經定義失真 d_i ，可以定義失真的機率分佈 $P(d_i)$ 。注意句子侷限於長度 n ， $|d_i| \leq n$ 並且全機率分佈 $P(d_i)$ 僅有 $2n + 1$ 元素，學習的數量遠少於數字的排列 $n!$ 。這就是為何我們以此迂迴方式定義排列。當然，這是一個失真的比較貧困模型。當我們從英文翻譯到法語，不會說在名詞後面形容詞通常出現扭曲——這事實是在所被表達的法語模型 $P(f)$ 。失真的機率在片語中是完全獨立於單字——僅和整數數值 d_i 相關。機率分佈提供一個排列反覆的總和，例如怎麼可能 $P(d = 2)$ 的失真和 $P(d = 0)$ 相比。

我們現在準備將它們整合：我們可以定義 $P(f, d | e)$ ，具有失真 d 的片語 f 序列是片語 e 序列翻譯的機率。我們可以做個假設每個片語翻譯和每個失真個別獨立，且因此我們可包括表達的因素為

$$P(f, d | e) = \prod_i P(f_i | e_i) P(d_i)$$

這給了我們一個計算 $P(f, d | e)$ 機率的方式，對於候選翻譯 f 和失真 d 。但要找最佳的 f 和 d 不能僅列舉句子，或許語料庫中 100 個法語片語對應每個英語片語，有 100^5 種不同 5 片語翻譯，且對於其中每個有 5! 種排列。我們將會找尋好的解。可估計機率的啟發式局部剪枝搜尋法(4.1.3 節)已被證實，可有效尋找接近最可能翻譯。

所有剩餘的是學習片語和失真機率。我們描繪這個程序，詳細請參考在章末的部分。

1. 找平行文本：

首先蒐集平行雙語語料庫。例如，**議事錄**^[9]是議會辯論的紀錄。加拿大、香港以及其他一些國家都製作雙語的議事錄，歐盟用 11 種語言出版它的官方文件，聯合國出版的更是多語言文件。雙語文本也存在於線上，某些網頁以統一位置資源(URL)平行公布平行目次，像是 /en/ 對於英文頁面、/fr/ 對應相符的法語頁面。領導統計翻譯系統訓練平行文本的億萬單字和單一語言文本的數十億單字。

2. 切分句子：

翻譯的單元是語句，因此我們將語料庫分解成句子。句號是最強的句子結束旗標，但是要考慮到「Dr. J. R. Smith of Rodeo Dr. arrived.」；只有最後一個句號才表示一條句子的結束。一個方法來決定若句號結束個句子，訓練個模型需要單字周圍和他們的詞性做為特徵。這個方法達到大約 98% 的精確度。

3. 句子對齊：

對英語版本的每條語句，判斷它所對應的法語語句有哪(些)條。通常，英語的下一句話和法語的下一句話以 1:1 的匹配相對應，但是有時也有變化：一種語言中的一條語句會被分割成 2:1 的匹配，或者兩個語句的順序要交換，造成一個 2:2 的匹配。單看語句長度(亦即，短句應與短句對齊)，利用 Viterbi 演算法的一個變形，就可能以 90% 到 99% 間的準確度對齊它們(1:1, 1:2 或 2:2 等)。如果採用兩種語言都常用的界標，諸如數詞或專用名詞，或者一些我們知道從雙語字典中的翻譯無歧義的單字，甚至能得到更好的對齊結果。例如，若第 3 個英語和第 4 個法語句子包含字串「1989」且近鄰句子沒有，這是個句子該排列在一起的好證據。

4. 排列片語：

在一個句子的範圍，片語可能以相似於所使用的句子排列用一個程序排列，但需要不斷改善。當開始時我們沒有方法知道「qui dort」和「sleeping」排列在一起，但我們可藉由證據的聚集程序來達成排列。透過所看過的所有實例句子，我們注意到「qui dort」和「sleeping」頻繁地共同出現，並且在排列的句子對，除了「qui dort」在其他句子沒有如此頻繁地共同出現「sleeping」。透過我們的語料庫一個完整的片語排列，給出我們一個片語的機率(經過適當的潤飾)。

5. 摘錄失真：

有一個片語的排列我們可定義失真機率。簡單地計算語料庫中對於每個距離失真多常發生 $d = 0, \pm 1, \pm 2, \dots$ ，並且使用潤飾。

6. 隨著 EM 提高判斷：

使用期望最大化來提高 $P(f|e)$ 和 $P(d)$ 的判斷值。我們在步驟 E 以當前值計算這些參數最好的排列，然後在步驟 M 更新判斷並且反覆過程至收斂。

23.5 語音識別

語音識別這個任務是根據給定聲學信號來辨識說話人所說的單詞序列。這已經成為人工智慧的主流應用其中之一——數以百萬的人們每天使用語音識別互動來操控語音郵件系統、從手持裝置搜尋網頁、以及其他應用。當需要免持操控時例如操作機器，語音是具有吸引力的選擇。

語音識別是困難的因為由講話者發出的聲音是混淆且有雜訊。眾所周知的實例，片語「recognize speech」當說快一點的時候聽起來幾乎像是「wreck a nice beach」。即使這些短的實例表示數個爭議造成語音的困難。首先是**分段**：寫下的英語單字之間具有空格，但在「wreck a nice」快速的語音沒有間隔，將需要區別多個單字片語還是單個字「recognize」。第二，**連音**：當講者將「nice」尾端的「s」發音快速地與「beach」的開頭髮音「b」合併，則會些近於「sp」。另一個不會顯示在這個實例的問題是**同音詞**——單字像是「to」、「too」和「two」聽起來一樣但是在意思不同。

我們可看到語音識別在最有可能序列解釋上的問題。如同我們在 15.2 節看到，這是一個計算狀態變數 $\mathbf{x}_{1:t}$ 的最有可能序列的問題，給定一個觀察的序列 $\mathbf{e}_{1:t}$ 。在這情形狀態變數是單字且觀測是聲音。更準確地，一個觀測是擷取自音效訊號的特徵向量。依照慣例，最有可能序列可以依貝氏法則來做計算：

$$\operatorname{argmax}_{word_{1:t}} P(word_{1:t} | sound_{1:t}) = \operatorname{argmax}_{word_{1:t}} P(sound_{1:t} | word_{1:t}) P(word_{1:t})$$

這裡 $P(sound_{1:t} | word_{1:t})$ 是**聲學模型**。它描述了詞語的發音——「ceiling」(天花板)由一個發絲音的字母「c」開頭，並且和「sealing」密封的發音相同。 $P(word_{1:t})$ 被稱為**語言模型**。它詳列了每個話語的事前機率——例如「ceiling fan」比起「sealing fan」有 500 倍的更有可能為單字序列。

這個方法由 Claude Shannon(1948)命名為**雜訊通道模型**。他描述了一個狀況其中原始訊息(在我們的實例中的單字)透過雜訊通道(像是電話線)傳輸這樣導致走樣的訊息(在我們實例的聲音)由另一端所接收。Shannon 表示無論通道如何地具有雜訊，以任意小的錯誤回復到原始訊息是有可能的，若我們將原始訊息經過足夠的冗餘方式來編碼。雜訊通道方法已被應用到語音識別、機器翻譯、拼字校正和其他任務。

我們定義聲學和語言模型，我們可使用為特比演算法解決單字的最有可能序列(15.2.3 節)。多數的語音識別系統使用語言模型做馬可夫假設——當前狀態 $Word_t$ 僅取決於前一個狀態的固定數字 n ——並且表達 $Word_t$ 為單一隨機變數，注意到值的有限集，成為隱馬克夫模型(Hidden Markov Model, HMM)。因此語音識別成為一個 HMM 方法的簡單應用，如同在 15.3 節所述——是簡單的，我們定義一個聲學和語言模型。接下來談論到它們。

23.5.1 聲學模型

聲波是透過空氣傳播的壓強的週期性變化。當這些波撞擊到麥克風的震動膜，往復的運動產生電流。一個類比信號到數位信號的轉換器在由**取樣率**所決定的離散區間間隔上測量電流的強度——電流強度對應著聲波的振幅。語音聲響主要的範圍在 100Hz(每秒 100 次)到 1000 Hz，典型的取樣率是在 8 kHz(CDs 和 mp3 檔案是以 44.1 kHz 取樣)。而每次測量的精度取決於**量化因數**；典型的語音識別系統使用 8 到 12 位元(二進位位元)。這意味著一套使用 8 位元量化和 8 千赫茲取樣率的低端系統每分鐘的語音需要大約

即使我們只想知道說了哪些單字，並不確定它們聽起來像什麼，我們不需要保留這些所有的資訊。我們只需要區別出不同語音之間的差異。語言學家已分辨出 100 種語音或是**音素(phone)**，可組成所有已知人類語言中的所有單字。粗略地說，一個音素是與單個母音或者輔音相對應的發音，但也有一些複雜情況：諸如「th」和「ng」之類的字母組合產生單個音素，而有些字母在不同的上下文中產生不同的音素(例如，單詞 *rat* 和 *rate* 中的字母「a」)。圖 23.14 列出了英語中使用的音素，並為每個音素附上了例子。**音位(phoneme)**是對使用某種特定語言的說話人具有獨特意義的最小發音單位。例如，「t」在「stick」聽起來像是在「tick」的「t」對英語的講者來說它們是相同的音位。但是這個差別在泰語就非常明顯，所以它們是兩個音位。為了表達說英語，我們希望一個表達可以區別出兩個不同的音位，但不需要區別聲音中的非音位變化：大或小聲、快或慢、男聲或女聲…等。

首先，我們發現雖然語音中的聲音頻率可能達到幾千赫茲，但信號內容中發生的變化卻不那麼頻繁，變化頻率一般不超過 100Hz。因此，語音系統概括(summarization)了在被稱為**碼框(frame)**的擴展區間內的信號特徵。一個長度約 10 毫秒的碼框(即 8kHz 取樣率下 80 個樣本)是短得足以保證幾乎沒有短期音素被遺失。重疊碼框是用來確定我們沒有遺漏訊號，因為它恰好落在碼框的邊界。

每個碼框由**特徵**向量所加總。從一個語音信號中挑出特徵就如同在聽交響樂，並說：「這裡大聲演奏的是法國號，而很輕柔的則是小提琴。我們將給出典型系統特徵的簡要概述。首先，使用傅立葉轉換在大約數十種頻率來算出聲學能量的範圍。然後對於每個頻率我們算出測量稱為**梅爾式頻率聲譜係數**(mel frequency cepstral coefficient)或 MFCC。我們同樣也計算在碼框的總能量。這給出 13

個特徵，對於每個我們計算出介於此碼框和前一個碼框的差異，並且各差異之間的差異，對於 39 個特徵的全部。這些都是連續值，最簡單安置它們於 HMM 框架的方法是將這些值離散化。(這也可能延伸 HMM 模型來處理高斯的連續混合)。圖 23.15 顯示了由原始聲音到碼框序列的變換過程。

母音		輔音		輔音	
音素	例子	音素	例子	音素	例子
[iy]	<u>b</u> eat	[b]	<u>b</u> et	[p]	<u>p</u> et
[ih]	<u>b</u> it	[ch]	<u>Ch</u> et	[r]	<u>r</u> at
[eh]	b <u>e</u> t	[d]	d <u>e</u> bt	[s]	<u>s</u> et
[æ]	b <u>a</u> t	[f]	<u>f</u> at	[sh]	<u>sh</u> oe
[ah]	b <u>u</u> t	[g]	<u>g</u> et	[t]	<u>t</u> en
[ao]	b <u>o</u> ught	[hh]	<u>h</u> at	[th]	<u>th</u> ick
[ow]	b <u>o</u> at	[hv]	<u>h</u> igh	[dh]	<u>th</u> at
[uh]	b <u>o</u> ok	[jh]	<u>j</u> et	[dx]	b <u>u</u> tter
[ey]	b <u>a</u> it	[k]	<u>k</u> ick	[v]	<u>v</u> et
[er]	B <u>e</u> rt	[l]	<u>l</u> et	[w]	<u>w</u> et
[ay]	b <u>u</u> y	[el]	bott <u>l</u> e	[wh]	<u>wh</u> ich
[oy]	b <u>o</u> y	[m]	<u>m</u> et	[y]	<u>y</u> et
[axr]	din <u>e</u> r	[em]	botto <u>m</u>	[z]	<u>z</u> oo
[aw]	d <u>o</u> wn	[n]	<u>n</u> et	[zh]	mea <u>s</u> ure
[ax]	<u>a</u> bout	[en]	butto <u>n</u>		
[ix]	ros <u>e</u> s	[ng]	sing <u>g</u>		
[aa]	c <u>o</u> t	[eng]	wash <u>g</u>		
				[-]	<i>silence</i>

圖 23.14 ARPA 音標符號表(或 ARPAbet)，列出了所有美式英語中用到的音素。另外還有幾種可選的符號表示方法，包括國際音標符號表(International Phonetic Alphabet, IPA)，其中包含了所有已知語言中的音素

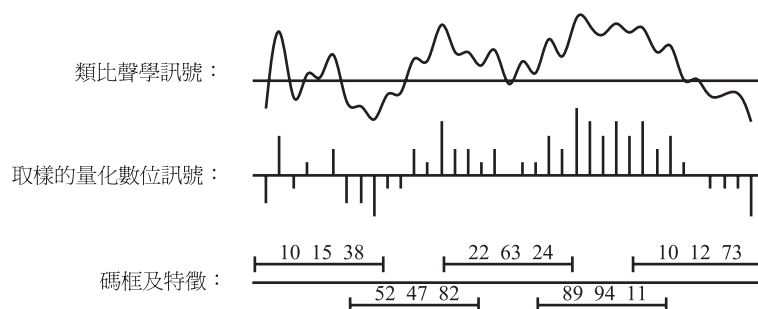
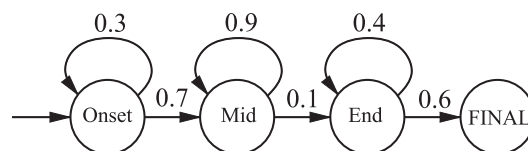


圖 23.15 轉換聲學訊號到碼框。在這個圖中，每個碼框是以 3 個聲學特徵的離散值描述；真實的系統會有數十種特徵

我們已經看過如何從原始聲學訊號到一連串的觀察 \mathbf{e}_t 。現在我們必須描述(無法觀測)HMM的狀態,並且定義轉換模型 $\mathbf{P}(\mathbf{X}_t | \mathbf{X}_{t-1})$ 和感測模型 $\mathbf{P}(\mathbf{E}_t | \mathbf{X}_t)$ 。轉移模型可以被區分為兩個階段:單字和音素。我們將從底部開始:

音素模型將音素描述為 3 個狀態:結首音、中間和結尾。例如,[t]音素在開始為靜音,中間為些許的爆音,以及(通常)在結尾是嘶聲。圖 23.16 表示對於音素[m]的實例。注意到,在普通的語音中,平均音素會持續約 50 到 100ms,或者 5 到 10 碼框。這個自我循環在每個狀態的期間允許變動。藉著使用許多自我循環(特別是中間狀態),我們可表示個長的「mmmmmmmmmmmm」聲音。忽略過自我循環產生個短的「m」聲音。

[m]的音素HMM:



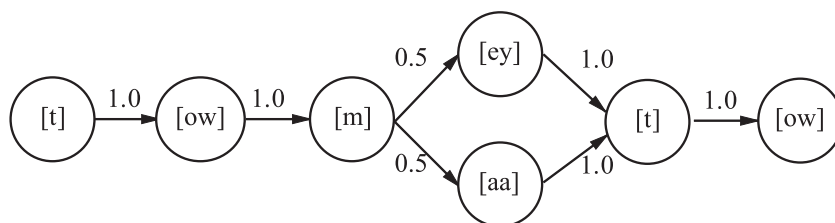
音素HMM的輸出機率:

Onset:	Mid:	End:
C_1 : 0.5	C_3 : 0.2	C_4 : 0.1
C_2 : 0.2	C_4 : 0.7	C_6 : 0.5
C_3 : 0.3	C_5 : 0.1	C_7 : 0.4

圖 23.16 三態音素[m]的一個 HMM(隱馬可夫模型)。每種狀態都有若干可能的輸出,每個輸出都有自己的機率。從 C_1 到 C_7 的 MFCC 特徵標示是任意的,代表著某些特徵值的組合

圖 23.17 音素模型是從發音模型串成一個單字。根據 Gershwins(1937)的著述,你會將其讀作 [t ow m ey t ow],而我會讀作[t ow m aa t ow]。圖 23.17(a)顯示了支援這種方言變化的轉移模型。圖中的每個圓圈表示個音素模型,像是圖 23.16 其中之一。

(a) 單字模型,具有方言變化:



(b) 單字模型,具有諧音及方言變化:

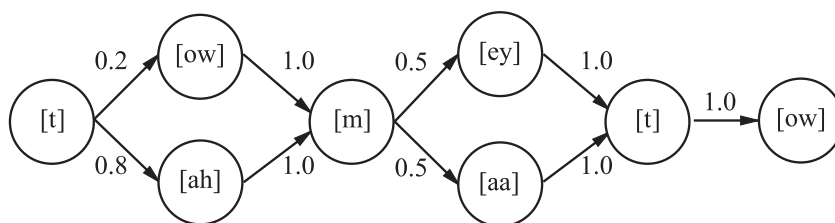


圖 23.17 單詞「tomato」的兩個發音模型。每個模型都表示為一個轉移圖,其中圓圈表示狀態,箭頭顯示允許的轉移過程及其相關的機率。(a) 考慮到方言差異的模型。這裡的值 0.5 是基於本書的兩位作者所偏好的發音估計得到的。(b) 考慮到第一個母音上的音素重疊效應的模型,允許其發[ow]或[ah]音

爲了方言的變化，單字可以有諧音變化。例如，發[t]這個音素時舌頭是在口腔的上部，然而發[ow]時舌頭接近於口腔的底部。當說快一點，舌頭沒有時間到位發[ow]，並且以[t ah]結束而不是[t ow]。圖 23.17(b)給出個「tomato」的模型考慮到諧音影響。更複雜的音素模型考慮到音素周圍的文本。

在單字的發音可能有本質的變化。「because」最一般的發音是[b iy k ah z]，但僅佔 1/4 的使用者。另外(接近)1/4 以[ix]、[ih]或[ax]替換第一個母音，並且以[ax]或[aa]替換第二個母音，[zh]或[s]取代最後的[z]，或者是去除「be」整個，留下「cuz.」。

23.5.2 語言模型

對於一般用途語音識別，語言模型可爲一個從學習自寫出的句子語料庫的文本之 n 元模型。然而，口說的語言有著和寫作的語言不同的特徵，所以最好得到一個口說語言的文字記錄語料庫。對於特定任務語音識別，語料庫必須爲特定任務：要建立航空訂位系統，得到事前電話的文字記錄。這也對於特定任務字彙有幫助，像是列出所有服務的機場與城市以及所有航班號碼。

部分語音使用介面的設計迫使使用者從選項的限制集合中說出事項，所以語音識別器將有嚴謹的機率分佈來處理。例如，詢問「What city do you want to go to?」引出個高度約束的語言模型，當詢問「How can I help you?」則否。

23.5.3 搭建語音識別器

語音識別系統的品質依賴於其所有組成部分的品質——這些組成部分包括語言模型、詞語發音模型、音素模型以及用於從聲學信號中提取頻譜(spectral)特徵的信號處理演算法。我們已經討論過如何建構語言模型，而關於信號處理的細節則留給了其他教科書。還剩下發音模型和音素模型。關於發音模型的結構——諸如圖 23.17 中所示的 tomato 模型——通常是手工建立起來的。現在對於英語和很多其他語言，我們已經可以找到很多大型讀音詞典，儘管其精確度良莠不齊。三態音素模型的結構對所有的音素都相同，如圖 23.16 所示。另外，剩下機率本身。

如同一般，我們會從語料庫得到機率，此時語料庫的語音。語料庫所包含的最一般型態，是對於每個搭配單字的文字記錄的句子之語音訊號。從這個語料庫建立模型是比建立文本的 n 原模型還困難，因為我們必須建立隱馬克夫模型(HMM)——每個單字的音素順序和每個時間碼框的音素狀態是隱變數。在語音識別的早期，引變數是由費時的手工標示譜儀提供。近代的系統使用期望最大化來自動地提供這個遺失的資料。其想法非常簡單：給定了一個隱馬可夫模型和一個觀察序列，我們可以透過第 15.2 節和第 15.3 節所介紹的平滑演算法計算出在每個時間步上每個狀態的機率，並透過一個簡單的擴展，得到相繼時間步上的狀態-狀態對的機率。可以把這些機率看作是不確定標記。根據這些不確定標記，我們能夠估計出新的轉移機率和感測器機率，並重複執行期望最大化演算法。該方法保證在每次迭代中不斷提高模型和資料之間的一致性，並且通常收斂到一個比最初的手工標記估計所提供的參數值好得多的參數值集合。

系統在藉著每個講者訓練不同的模型會有最高的準確度，因此捕捉差異除了方言、男女和其他的種類。訓練可需要數小時與講者的互動，系統以最廣的散佈所採用，不建立特定講者模型。

系統的精確度基於因素的數目。首先，訊號的品質問題：高品質指向性麥克風指著在填充空間的固定口，將會比起便宜的麥克風在車上傳送訊號透過電話線，並隨著收音機播放來得好。單字大小問題：當識別數字字串以 11 個單字的詞彙(1-9 加上「oh」和「zero」)，單字錯誤率將降到 0.5%，儘管它在一個具有 20,000 單字詞彙的新故事提高到大約 10%，並且 20%對於具有 64,000 單字詞彙的語料庫。任務也有關：當系統嘗試著完成一個特殊任務——預約個航班或是給出餐廳的方向——任務通常可完美地達成，即使單字錯誤率為 10%或更高。

23.6 總結

自然語言理解是人工智慧中最重要的子領域之一。與人工智慧的其他領域不同的是，自然語言理解需要有關實際人類行為的經驗化研究——結果證明這是複雜且有趣的。

- 正規語言理論以及**片語結構**(phrase structure)文法(特別是**語境自由文法**)都是處理自然語言的有用工具。機率化文本無關文法(PCFG)形式論被廣泛地使用。
- 在文本無關語言的句子可以在 $O(n^3)$ 的時間被剖析，利用像是 CYK 演算法的**圖形剖析器**，其中需要文法規則在**杭士基正規形式**(Chomsky Normal Form)。
- **樹狀庫**可以用於學習文法。從未剖析語料庫的句子也可以學習到文法，但這很少是成功地。
- **詞彙化 PCFG** 允許表示某些介於單字間的關係，比起其他的更為一般。
- 爲了處理類似主詞與動詞一致以及代名詞格位的問題，**擴充**(augment)一個文法是很方便的方法。**確定式文法**(DCG)是一種允許擴充的形式化方法。透過 DCG，剖析和語意解釋(甚至是句子產生)都可以用邏輯推導實作。
- **語意解釋**(semantic interpretation)也可以用擴充文法處理。
- **歧義**(ambiguity)是自然語言理解中一個很重要的問題；大多數句子有多種可能的解釋，但是通常僅有一個是合適的。而排歧則依賴於有關世界的、當前語境的以及語言用法的知識。
- **機器翻譯**系統已經採用了一系列技術進行實作，從完整的句法和語義分析到基於單字出現頻度的統計技術。當前的統計模型是最受歡迎與最成功的。
- **語音識別**系統也是主要基於統計原理。語音系統儘管不完善，仍是受歡迎與有用。
- 機器翻譯和語音識別是自然語言技術的兩個大的成就。其中一個原因是若存在個大的語料庫模型有好的表現——翻譯和語音是「自然環境下」每天由人們所表現的任務。反之，像是剖析句子的任務則較不成功，部分原因是沒有存在於「自然環境下」個被剖析句子的語料庫，且部分原因是剖析本身就是沒有用的。

◎ 參考文獻與歷史的註釋 BIBLIOGRAPHICAL AND HISTORICAL NOTES

如語意網路，語境自由文法(也稱片語結構文法)是一種技術的再發明，該技術首先被印度文法家(尤其是 Panini, 約西元前 350)用於學習印度法典中的梵語(Ingerman, 1967)。杭士基(Noam Chomsky, 1956)重新發明了 CFG，用於對英句子法的分析，而巴克斯(John Backus)也獨立地發明了 CFG，用於 Algol-58 句法的分析。瑙魯(Naur, 1963)擴展了巴克斯的標記法，現在獲得了 BNF 中的「N」的榮譽(Backus, 1996)，這個字母原本代表的是「Backus Normal Form(巴克斯範例)」中的「Normal」。Knuth(1968)定義了一種被稱為屬性文法(attribute grammar)的擴充文法，該文法對程式語言很有用。Colmerauer(1975)引入了確定式文法，Pereira 和 Warren(1987)則對其進行了發展和推廣。

PCFG 是由 Booth(1969)和 Salomaa(1969)研究提出的。Charniak(1993)對於 PCFG 其他的演算法發表了絕佳的短篇論著，且由 Manning 和 Schütze (1999)以及 Jurafsky 和 Martin(2008)著有極優的長篇教科書。Baker(1979)引入了向內向外演算法，Lari 和 Young(1990)則描述了它的用途和局限性。Stolcke 和 Omohundro(1994)表現如何以貝氏模型合併來學習文法規則，Haghighi 和 Klein(2006)基於原型描述個學習系統。

詞彙化 PCFGs(Charniak, 1997、Hwa, 1998)合併 PCFG 和 n 元模型的最佳觀點。Collins(1999)描述以開頭特徵來詞彙化 PCFG 剖析。Petrov 和 Klein(2007a)表示如何得到詞彙化的優點沒有實際詞彙擴增，從具有一般類別的樹狀庫藉由學習特殊句法範疇，例如樹狀庫有名詞片語範疇，從更具體類別像是 NPO 和 NPS 可被學習。

不論是在「純粹的」語言學還是在計算語言學中，都已經對寫出自然語言的形式文法進行了很多嘗試。還有一些關於英語的很全面但是非形式化的文法(Jespersen, 1965; Quirk 等人, 1985; McCawley, 1988; Huddleston 和 Pullum, 2002)。從 20 世紀 80 年代中期開始，將更多的資訊放入詞典而不是文法中成為一種發展趨勢。辭彙功能文法，即 LFG(Bresnan, 1982)是第一個主要的高度辭彙化的文法形式化方法。如果我們把辭彙化推到極致，那麼我們最終得到可以只包含少至兩條文法規則的範疇文法(categorial grammar)(Clark 和 Curran, 2004)，或只包含詞語關係而沒有語法類別的依存文法(dependency grammar)(Smith 及 Eisner, 2008; Kübler *et al.*, 2009)。Sleator 和 Temperley(1993)描述依存分析器。Paskin(2001)表示依存關係文法的版本比起 PCFG 還容易學習。

第一個電腦化的剖析演算法是由 Yngve(1955)提出的。有效的演算法在 20 世紀 60 年代晚期逐步發展起來，不過從那時起，走了一些彎路(Kasami, 1965; Younger, 1967; Graham 等人, 1980)。Maxwell 和 Kaplan(1993)證明了擴充的圖剖析器能夠在平均情況下提高效率。Church 和 Patil(1982)則強調了句法歧義的消解問題。Klein 和 Manning(2003)描述 A*剖析，且 Pauls 和 Klein(2009)擴展 K 最佳 A*剖析，其中結果不是個單一剖析但 K 最佳。

現今的領導分析器包含這些，在 Petrov 和 Klein(2007b)其中在華爾街日報語料庫精準度達到 90.6%，Charniak 和 Johnson(2005)達到 92.0%，和 Koo 等人(2008)其中在 Penn 樹狀庫達到 93.2%。這些數字不可直接地比較，有些領域的評論是，在少數選定的語料庫焦點太過於狹窄，對於它們或許會過適配。

自然語言的形式語意解釋起源於哲學和形式邏輯之中，尤其是 Alfred Tarski(1935)在正規語言的語意學方面的研究。Bar-Hillel 則是第一個考慮語用問題的人，並提出了可以用形式邏輯來處理語用的想法。例如，他把 C. S. Peirce(1902)的術語 *indexical*(索引詞)引入到語言學(Bar-Hillel, 1954)中。雖然 Richard Montague 的文章《作為正規語言的英語》(*English as a Formal Language*, 1970)是關於語言的邏輯分析的某種宣言，但是 Dowty 等人所著的書籍(1991)以及 Lewis 的文章(2002)相對來說更通俗易懂。

第一個用於解決實際任務的 NLP 系統很可能就是 BASEBALL 問答系統(Green 等人, 1961)，該系統能夠處理有關棒球統計資料庫的問題。這之後不久，Wood(1973)研製了 LUNAR 系統，用於回答有關透過阿波羅計畫從月球帶回來的岩石的問題。Roger Schank 和他的學生搭建了一系列完成自然語言理解任務的程式(Schank 和 Abelson, 1977; Wilensky, 1978; Schank 和 Riesbeck, 1981; Dyer, 1983)。語意解釋的現代方法通常假設從語法到語意將從實例學習(Zelle 和 Mooney, 1996; Zettlemoyer 和 Collins, 2005)。

Hobbs 等人(1993)敘述對於解釋定量非機率框架。近期的研究跟隨著明確概率框架(Charniak 和 Goldman, 1992; Wu, 1993; Franz, 1996)。在語言學中，最佳化理論(Kager, 1999)是建立在這樣的想法基礎上的：在文法中構建軟限制，給予解釋一個自然的等級，而不是讓文法以相同的等級產生所有機率。諾維格(Norvig, 1988)討論了考慮多個同時出現的解釋，而不是固定採用唯一的最大可能性解釋的問題。文藝評論家(Empson, 1953; Hobbs, 1990)則提出了「歧義應該被消解還是應該被保留」的疑問。

Nunberg(1970)勾畫了一個有關轉喻的形式化模型。Lakoff 和 Johnson(1980)則致力於英語中常見暗喻的分析和編目研究。Martin(1990)和 Gibbs(2006)提供隱喻解釋的計算模型。

文法歸納(grammar induction)研究一開始成果是負面的：Gold(1967)證明了在已知一個來自語境自由文法的字串集合的情況下，可靠地學習到正確的該文法是不可能的。很多著名的語言學家，如杭士基(1957, 2003)和 Pinker(1989, 2000)等採用 Gold 的結果來論證必定存在一個先天的、每個小孩從出生開始就擁有的**普遍文法**(universal grammar)。所謂的**刺激貧乏**(Poverty of the Stimulus)論據認為沒有給小孩足夠的輸入學習 CFG，所以他們必須已經知道「文法」並且微調其中部分的參數。當這個論點在很多杭士基學派的語言學家中繼續保持支配地位時，其他語言學家(Pullum, 1996; Elman 等人, 1997)和大多數電腦科學家則已經拋棄了這個論點。早在 1969 年，Horning 就指出在 PAC 學習的意義上學習一種機率的語境自由文法是可能的。從那時起，湧現出許多僅從正例中進行學習的令人信服的實驗證實，諸如 Mooney(1999)的 ILP 研究、Muggleton 和 De Raedt(1994)的研究、Schütze(1995)出色的博士論文以及 de Marcken(1996)的研究等。這是語法推理年度國際會議(International Conference on Grammatical Inference, ICGI)。學習其他語法形式主義是可能的，像是正規語言(Denis, 2001)和有限狀態自動機(Parekh 和 Honavar, 2001)。Abney(2007)是一本對於語言模型半受監督學習導論的教科書。

Wordnet(詞網)(Fellbaum, 2001)是一個面向大眾的字典，包含大約 100,000 個詞語和片語，以詞性標注對詞語劃分範疇，並用諸如同義、反義以及部分等語意關係將各範疇聯繫起來。Penn 樹狀庫(Marcus *et al.*, 1993)提供 3 百萬單字英語的語料庫之剖析樹。Charniak(1996)、Klein 和 Manning(2001)討論了用樹狀庫(treebank)語法進行句法分析。英國國家語料庫(British National Corpus)(Leech *et al.*, 2001)延伸到大約 1 億個單字，而全球資訊網則包含超過數兆的單字，(Brants 等人, 2007)透過 2 兆單字網頁語料庫描述 n 元模型。

在 20 世紀 30 年代，Petr Troyanskii 為一台「翻譯機器」申請了專利，但是沒有能實作他的想法的電腦。1947 年 3 月，洛克菲勒基金的 Warren Weaver 致信維納(Norbert Weiner)，提出機器翻譯也許是可能的。借鑒密碼學和資訊理論，Weaver 寫道：「當我看到一篇俄語文章時，我對自己說：『這實際是用英語寫的，但是它用了一些奇怪的符號進行編碼。我現在將進行解碼。』」下一個十年間，大家都試圖用這種方法進行解碼。IBM 在 1954 年展示了一個基本的系統。Bar-Hillel(1960)描述了這個時期的熱情。然而，美國政府隨後報告(ALPAC, 1966)其中「有用的機器翻譯沒有直接的或可預期的前景」。然而，有限的研究工作持續，並 1980 年代開始，電腦的能力大大提高，ALPAC 的發現不再是正確的。

本章中描述的統計方法是基於 IBM 團隊的早期研究(Brown *et al.*, 1988, 1993)以及由 ISI 和 Google 研究團隊(Och 和 Ney, 2004、Zollmann *et al.*, 2008)近期的研究。統計機器翻譯的導論教科書是由 Koehn(2009)，以及 Kevin Knight(1999)所著具有影響力的短篇指南。早期的語句切分的研究工作是由 Palmer 和 Hearst(1994)完成的。Och 和 Ney(2003)以及 Moore(2005)則論及了雙語語句的對齊問題。

語音識別的史前史開始於 20 世紀 20 年代的一隻聲音驅動的玩具狗「無線電雷克斯」(Radio Rex)。雷克斯跳出牠的狗屋並回應「Rex!」一個字。(或實際上是任何足夠響亮的字。)一些比較嚴肅的工作開始於二戰之後。在 AT&T 公司的貝爾實驗室，建立了一個透過聲學特徵的簡單模式匹配的手段識別孤立數位的系統(Davis 等人, 1952)。從 1971 年開始，美國國防部的國防高級研究計畫局(DARPA)資助了 4 個競爭性的開發高性能語音識別系統的 5 年計畫。其勝出者，也就是其中唯一一個實作了「能夠在 1000 單詞辭彙量下達到 90%精確度」的目標的系統，是卡內基梅隆大學(CMU)的 HARPY 系統(Lowerre, 1976；Lowerre 和 Reddy, 1980)。HARPY 的最後版本是從一個卡耐基梅隆大學的研究生 James Baker(1975)所建立的被稱為 DARGON 的系統發展而來的；DARGON 第一個在語音處理中使用了隱馬可夫模型。幾乎與此同時，國際商用機器公司(IBM)的 Jelinek(1976)開發了另外一個基於隱馬可夫模型的系統。最近幾年發展的特點則是各種漸增的進步，更大的資料集與模型，以及在更多現實語音任務上的更加激烈的競爭。在 1997 年 Bill Gates 預測「五年後的個人電腦——你將不需要認識它，因為語音將會成為它的介面。」這並不完全發生，但在 2008 年他預測「再五年微軟預計更多的網路搜尋藉由語音完成，而非用鍵盤打字。」歷史將會知道這一次他是否為正確。

在語音識別方面有一些很好的教科書(Rabiner 和 Juang, 1993; Jelinek, 1997; Gold 和 Morgan, 2000; Huang 等人, 2001)。本章使用的表示方法引用了 Kay, Gawron 和 Norvig(1994)的綜述以及 Jurafsky 和 Martin(2008)的教科書。關於語音識別研究的論文發表於《電腦語音和語言》(*Computer Speech and Language*)，《語音通訊》(*Speech Communications*)，《IEEE 聲學、語音及信號處理學報》(*IEEE Transactions on Acoustics, Speech, and Signal Processing*)等期刊，以及「DARPA 語音和自然語言處理專題研討會」，Eurospeech，ICSLP，以及 ASRU 等會議上。

Ken Church(2004)表示自然語言研究介於集中在資料(經驗主義)和集中在理論(理性主義)。語言學家 John Firth(1957)宣稱「你必須知道公司保留的字」並且 1940 年代與 1950 年代早期的語言學家主要是根據單字頻率，即使在沒有現今我們所具備的計算能力。而後 Noam(Chomsky, 1956)表示有限狀態模型的限制，並且鼓舞語法規則的理論學習的興趣，忽視頻率計算。這個方法主宰了二十年，直到經驗主義基於一連串統計語音辨別的研究捲土重來。現在多數的研究接受統計框架，但最有興趣於建立統計模型，其中考量高階模型像是句法樹和語意關係，並非只是一連串的單字。

關於實用語言處理的研究工作發表在兩年一次的「應用自然語言處理會議」(Applied Natural Language Processing conference, ANLP)、「自然語言處理的經驗方法會議」(the conference on Empirical Methods in Natural Language Processing, EMNLP)，以及《自然語言工程》(*Natural Language Engineering*)期刊上。廣泛的 NLP 研究發表於計算語言學會暨會議(ACL)和計算語言學國際會議(COLING)。

❖ 習題 EXERCISES

23.1 閱讀一遍下面的短文並加以理解，盡可能多地記下其中的內容。後面會有一個測試：

這程序實際上十分簡單。首先將事情安排到不同的群組。當然一堆可能完全地關於是有做多少，若因為下一步缺少資源必須出走，否則是相當不錯的一套。不要過度努力是很重要的。也就是說一次做過少的事比做太多來得好。在這短期執行可能不會看起來很重要但複雜度可能輕易地提高。錯誤的代價是相當昂貴的。首先整個流程將看起來非常複雜。很快地，然而它將會變成生活的另一方面。在不久的將來預見對於此必要之任務的任何結束，但不會有人提出。在程序結束，再次將材料分配至不同群組。則可放置於它們適當的位置。最後它們將使用更多且將必須重複整個循環。然而，這是人生的一部份。

23.2 HMM 文法本質上是標準 HMM，其狀態變數是 N (非終端，具有像是 *Det*, *Adjective*, *Noun* 諸如此類的值)並且其證據變數是 W (單字，有著像 *is*, *duck*...等等的值)。HMM 模型包含事前 $P(N_0)$ ，轉移模型 $P(N_{t+1} | N_t)$ 和感知模型 $P(W_t | N_t)$ 。試證每個 HMM 文法可被改寫為 PCFG。(提示：從思考 HMM 事前對於句子符號可以如何地被表示成 PCFG 法則。你可能發現以值 A 、 B 對於 N 以及值 x 、 y 對於 W 有助於說明為特定 HMM)

23.3 考量對於簡單動詞片語的下列 PCFG：

0.1: $VP \rightarrow Verb$ 0.2: $VP \rightarrow Copula\ Adjective$ 0.5: $VP \rightarrow Verb\ the\ Noun$ 0.2: $VP \rightarrow VP\ Adverb$ 0.5: $Verb \rightarrow is$ 0.5: $Verb \rightarrow shoots$ 0.8: $Copula \rightarrow is$ 0.2: $Copula \rightarrow seems$ 0.5: $Adjective \rightarrow unwell$ 0.5: $Adjective \rightarrow well$ 0.5: $Adverb \rightarrow well$ 0.5: $Adverb \rightarrow badly$ 0.6: $Noun \rightarrow duck$ 0.4: $Noun \rightarrow well$

- a. 以下哪一個具有非零的機率？(i) shoots the duck well well (ii) seems the well well (iii) shoots the unwell well badly
- b. 產生「is well well」的機率會多少？
- c. 何種的歧義被(b)中的片語展出？
- d. 給定任意 PCFG，是否可能計算出 PCFG 產生出剛好 10 個單字的機率？

23.4 概述 Java 語言(或者任何其他你所熟悉的電腦語言)和英語的主要區別，評論每種情況下的「理解」問題。首先可以考慮如下方面：文法、句法、語意、語用、合成性、語境依賴性、辭彙歧義、句法歧義、指涉發現(包括代名詞)、背景知識以及「理解」意味著什麼。

23.5 本習題是有關非常簡單語言的文法的。

- a. 寫出語言 $a^n b^n$ 的語境自由文法。
- b. 寫出回文語言的語境自由文法。即所有字串的後半部分是前半部分的逆串。
- c. 寫出複製語言的語境限定文法。即所有字串的後半部分與前半部分相同。

23.6 考慮句子「Someone walked slowly to the supermarket」和詞典：

代名詞 \rightarrow someone	動詞 \rightarrow walked
副詞 \rightarrow slowly	介系詞 \rightarrow to
冠詞 \rightarrow the	名詞 \rightarrow supermarket

列 3 個文法中的哪個與該詞典束縛能夠產生上述句子？畫出相對應的剖析樹。

(A):	(B):	(C):
$S \rightarrow NP VP$	$S \rightarrow NP VP$	$S \rightarrow NP VP$
$NP \rightarrow Pronoun$	$NP \rightarrow Pronoun$	$NP \rightarrow Pronoun$
$NP \rightarrow Article Noun$	$NP \rightarrow Noun$	$NP \rightarrow Article NP$
$VP \rightarrow VP PP$	$NP \rightarrow Article NP$	$VP \rightarrow Verb Adv$
$VP \rightarrow VP Adv Adv$	$VP \rightarrow Verb Vmod$	$Adv \rightarrow Adv Adv$
$VP \rightarrow Verb$	$Vmod \rightarrow Adv Vmod$	$Adv \rightarrow PP$
$PP \rightarrow Prep NP$	$Vmod \rightarrow Adv$	$PP \rightarrow Prep NP$
$NP \rightarrow Noun$	$Adv \rightarrow PP$	$NP \rightarrow Noun$
	$PP \rightarrow Prep NP$	

上述的每個文法，分別寫出由文法產生的 3 句英語句子和 3 句非英語句子。每個句子應該是明顯地不同，最少會是 6 個單字長，且應包含某些新的詞彙元素(其中你必須定義)。提出改進每種文法以避免產生非英語句子的建議。

- 23.7 搜集一些時間運算式的例子，例如「two o'clock」、「midnight」以及「12:46」等。並虛構一些不符合語法的例子，例如「thirteen o'clock」以及「half past two fifteen」等。給時間語言編寫一個語法。
- 23.8 本題中將轉換 ϵ_0 到杭士基正規形式(CNF)。有 5 個步驟：(a) 加入新的起始符號，(b) 排除 ϵ 規則，(c) 排除右手邊多重單字，(d) 形式($X \rightarrow Y$)的排除規則，(e) 轉換長右手邊到二元法則。
- 起始符號 S 僅會發生在 CNF 中的左手邊。增加格式 $S' \rightarrow S$ 的新法則，使用新的符號 S' 。
 - 空白字串 ϵ 無法出現於 CNF 的右手邊。 ϵ_0 無法以 ϵ 具有任何規則，所以這不是個問題。
 - 在規則中只有形式($X \rightarrow word$)單字會出現於右手側。使用新符號 W' 以($X \rightarrow \dots W' \dots$)和($W' \rightarrow word$)更換每個規則的形式($X \rightarrow \dots word \dots$)。
 - 規則($X \rightarrow Y$)在 CNF 是不被允許的，必須為($X \rightarrow YZ$)或($X \rightarrow word$)。更換形式($X \rightarrow Y$)的每個規則，以形式($X \rightarrow \dots$)的規則集合對應到每個規則($Y \rightarrow \dots$)，其中(\dots)表示 1 個或更多的符號。
 - 以兩個規則($X \rightarrow YZ'$)和($Z' \rightarrow Z \dots$)更換每個規則的形式($X \rightarrow YZ \dots$)，其中 Z' 是個新符號。請表示程序的每個步驟與規則的最後集合。
- 23.9 用 DCG 標記法，為一種語言編寫文法，這種語言正如同 E_1 ，除了它要加強句子的主詞和動詞之間的一致性，從而不會產生類似「I smells the wumpus」的句子之外。
- 23.10 考慮以下問題：

$S \rightarrow NP VP[1.0]$
 $NP \rightarrow Noun[0.6] \mid Pronoun[0.4]$
 $VP \rightarrow Verb NP [0.8] \mid Modal Verb [0.2]$

 $Noun \rightarrow can [0.1] \mid fish[0.3] \mid \dots$
 $Pronoun \rightarrow I [0.4] \mid \dots$
 $Verb \rightarrow [0.01] \mid fish [0.1] \mid \dots$
 $Modal \rightarrow can [0.3] \mid \dots$

「I can fish」這個句子依照這個文法有兩個剖析樹。試表示給定該此句子的兩個樹、它們的事前機率和條件機率。

- 23.11 一個具有參數的文本無關文法可表示一般文本無關文法所不能表示的語言。寫出語言 $a^n b^n c^n$ 的語境自由文法。對於擴展變數所允許的值是 1 和 $SUCCESSOR(n)$ ，其中 n 是個值。在這個語言中句子的規則是

$$S(n) \rightarrow A(n) B(n) C(n)$$

對於每個 A 、 B 和 C 表示出 rule(s)。

- 23.12 擴充 ε_1 的文法，使其能夠處理冠詞-名詞一致性問題。也就是說，確保「agents」及「an agents」是 NP ，而「agent」和「an agents」不是。
- 23.13 考慮下列摘自 2008 年 7 月 28 日紐約時報的句子：

Banks struggling to recover from multibillion-dollar loans on real estate are curtailing loans to American businesses, depriving even healthy companies of money for expansion and hiring.

- 句子中哪一個自是詞彙上模稜兩可？
 - 找出句子中兩個句法上的模稜兩可(這有兩個以上)。
 - 舉出個這個句子隱喻的實例。
 - 可否找出語意上的歧義？
- 23.14 不要翻看習題 23.1，回答下列問題：
- 習題 22.1 中提到的 4 個步驟是什麼？
 - 其中被剔除的是什麼？
 - 文中提到的「the material」是什麼？
 - 要對什麼樣的錯誤付出很高的代價？
 - 做得太少還是做得過多比較好？為何？
- 23.15 選擇 5 個語句提交給一個線上翻譯伺服器。把它們從英語翻譯成另一種語言，然後再翻譯回英語。評價結果語句的語法性和所保留的含義。重複上述過程：第二輪的結果是更糟了還是相同？中間語言的選擇對結果的質量是否有影響？如果你懂一種外國語，看一下翻譯至該語言的一個段落。計算並描述所犯的錯誤，並且推測為何這些錯誤會犯。
- 23.16 對於圖 23.13 中每個句子 D_i 值總和為零。這對於每個翻譯對將是否為真？證明或舉出反例。
- 23.17 [改編自 Knight(1999)的著述]。我們的轉移模型假設，在片語轉移模型選定片語且失真模型交換它們，語言模型可還原出交換。本習題則研究這個假設是否明智。試著將下列所示的詞組恢復成正確的順序：
- have, programming, a, seen, never, I, language, better
 - loves, john, mary
 - is the, communication, exchange of, intentional, information brought, by, about, the production, perception of, and signs, from, drawn, a, of, system, signs, conventional, shared
 - created, that, we hold these, to be, all men, truths, are, equal, self-evident

你能完成哪些語句的正確排列？你需要利用何種知識？在訓練語料庫上訓練一個二元模型，用該模型從測試語料庫中找到某些語句的最高的換位機率。報告該模型的準確率。

- 23.18 對輸出序列 $[C_1, C_2, C_3, C_4, C_5, C_6, C_7]$ 計算穿過圖 23.16 所示的隱馬可夫模型的最可能路徑，並給出其機率。
- 23.19 我們忘記提到，習題 23.1 中的短文標題是「洗衣服」。重新閱讀短文，並回答習題 23.13 中的問題。這次你有沒有做得好一些？Bransford 和 Johnson(1973)把這篇短文用於一個良好控制的實驗中，並發現標題有顯著的幫助作用。這個告訴你什麼樣的關於語言和記憶如何地運作？

本章註腳

- [1] PCFG 也被稱為隨機語境自由語法，或 SCFG。
- [2] 關係子句跟在名詞片語之後，修飾該名詞片語。由關係代名詞(如「who」或者「that」)及緊接其後的動詞片語組成。關係子句的一個例子是「The wumpus *that stinks* is in 2,2.」中的「*that stinks*」。另一種關係子句沒有關係代名詞，像是「the man *I know*」中的 *I know*。
- [3] 在元件之間的相互組合的方式中還可能有 $O(c!)$ 的歧義性——比如， $(X \text{ and } (Y \text{ and } Z))$ 與 $((X \text{ and } Y) \text{ and } Z)$ 。不過這是另外一件事情，Church 和 Patil(1982)進行了詳細的敘述。
- [4] 主格原文為 subjective case，有時也被寫為 nominative case；受格原文為 objective case，有時也被寫為 accusative case。很多語言在間接賓語的位置上還有詞語的與格。
- [5] 如果這個解釋看起來不太合理，那麼就想想這一句的解釋：「Every Protestant believes in a just God」。
- [6] 本例是由 Martin Kay 提出的。
- [7] Warren Weaver (1949)報告關於 Max Zeldner 指出偉大的希伯來詩人 H. N. Bialik 說過的翻譯為「is like kissing the bride through a veil」。
- [8] 對於翻譯細微處， n 元明顯地不足夠。一個著名的例子是，馬塞爾·普魯斯特(Marcel Proust)的 4000 頁的小說《追憶似水年華》(*A la recherche du temps perdu*)以同一個詞開始和結尾，所以一些翻譯者決定照樣去做，這樣，對結尾這個詞的翻譯就是基於出現在大約 2 百萬個詞之前的那一個的。
- [9] 以 William Hansard 的姓氏命名，他於 1811 年第一次出版了英國議會辯論記錄。

