第 5 章

CHAPTER 5

计算机学会说话

5.1 ELIZA 会识别字符串模式

在以知识为基础的系统背景下，图灵所提出的打动了早期人工智能研究人员的这一著名问题频频被再次提起：这些系统能“思考”吗？它们“具有智能”吗？分析表明，基于知识的专家系统和传统的计算机程序都是基于算法的。即使将知识库和问题解决策略分离也不会改变这一特点，因为专家系统的两个组成部分都必须用算法数据结构表示，最终才能在计算机上编程实现。

这也适用于通过计算机实现自然语言。一个例子是魏茨鲍姆(J. Weizenbaum)的 ELIZA 语言程序。作为人类专家，ELIZA 模拟精神病医生与病人交谈。这些是关于如何用“精神病医生”的特定句型对病人的某些句型做出反应的规则。一般来说，这是关于规则在各种情况下的适用性的识别或分类。在最简单的情况下，必须确定两个符号结构的相等性，这是由关于符号列表的 LISP 编程语言中的 EQUAL 函数确定的(见 3.3 节)。如果符号表达式中包含术语和变量，则存在一种扩展，例如：

$$(xBC)$$

$$(ABy)$$

这两项相等，因为变量 x 对应于原子元素 A，变量 y 对应于原子元素 C。通过引入变量，模式匹配可以识别一个模式的结构。为此，在 LISP 中定义了一个单独的运算符 MATCH(匹配)。日期与结构对应的最简单情况是常数相等，例如列表 AB 和 AB 相等：

$$(\mathrm{MATCH}'(AB)'(AB))=\mathrm{T}$$

不同的生成元在逻辑或数字中的作用类似于变量，导致类的形成。所以在符号“?”帮助下，任何对象都可以调整，例如：

$$(\mathrm{MATCH}'(?\ B)'(AB))=\mathrm{T}$$

$$(\mathrm{MATCH}'(?\ B)'\ ((AB)B))=\mathrm{T}$$

因此，该结构将每个具有两个元素的列表与 B 作为第二个元素进行比较。* 生成器比

较任何元素序列(包括空序列),例如:

$$(\mathrm{MATCH}'(*B)'(B))=\mathrm{T}$$
$$(\mathrm{MATCH}'(*B)'(ABCB))=\mathrm{T}$$

上述给定的结构将所有列表与结束元素 B 相匹配。这已经解释了理解算法结构的基本原理,该结构反映了 ELIZA 中精神病医生的知识。病人 P 和 ELIZA 之间的对话可以想象为这样:

【举例】

病人:MEN ARE ALL ALIKE(男人都是一个样子)

ELIZA:IN WHAT WAY(何出此言)

病人:THEY'RE ALWAYS BUGGING US ABOUT SOMETHING OR OTHER(他们总是为某些事情烦扰我们)

ELIZA:CAN YOU THINK OF A SPECIFIC EXAMPLE(你能说出一个具体的例子吗)

病人:WELL;MY BOYFRIEND MADE ME COME HERE(是我男朋友让我来的)

ELIZA:YOUR BOYFRIEND MADE YOU COME HERE(你男朋友让你来的)

病人:YES;I AM VERY SAD(是的,我很伤心)

ELIZA:YOU ARE VERY SAD(你很伤心)

病人:SAD AND DEPRESSED(悲伤且沮丧)

ELIZA:PLEASE GO ON(请继续)

…

如果患者使用了某些句型,这些句型通过模式匹配被 ELIZA 识别,则以下规则允许推导出 ELIZA 的答案:

```
'(((*AM *B)→(APPEND'(YOU ARE)B))
((*ARE*)→'(IN WHAT WAY))
((*ALWAYS*)→'(CAN YOU THINK OF A SPECIFIC EXAMPLE))
((*MY * B ME * C)→(APPEND'(YOUR)(APPEND B(CONS'YOU C))))
((*L)→'(PLEASE GO ON)))
```

第二条规则是:如果病人的句子中含有 ARE,用"WHAT WAY"来回答。在输入句子"MEN ARE ALL ALIKE"(男人都是相似的)中,* 运算符代表的 MEN 出现在 ARE 之前,而 ALL ALIKE 出现在 ARE 之后。

第四条规则是:如果在病人的记录中,单词 MY 和 ME 被列表 * B 隔开,并且记录以列表 * C 结尾,那么 ELIZA 会做出反应,首先将 YOU 和 C 部分组合在一起(CONS'YOU C),然后应用 B 部分,最后应用'(YOUR)。

因此,在程序语言 LISP 示例中,与 ELIZA 的对话只是语法符号列表的派生。从语义上讲,结构的选择与口语娱乐习惯相对应。最后一条规则是一种典型的尴尬反应,因为它也发生在实际的对话中:如果任何符号列表(*L)没有被专家识别(也就是谈话中的噪声等),那么它就会装作一副聪明的嘴脸,说"请继续"。

决不能把孩子和洗澡水一起扔出去，从这个对话的简单算法结构中得出结论，模拟图灵测试只是一个魔术。ELIZA 这个简单的例子清楚地表明，派对上的讨论以及对人类专家的提问都是由基本模式决定的，只能在一定程度上有所不同。这些各自的基本模式由许多专家系统描述，算法捕捉得不多不少。然而，与专家系统不同的是，人类不能被简化为单独的算法结构。

5.2　自动机和机器识别语言

计算机基本上把文本处理成某种字母表的对应符号序列。计算机程序是计算机键盘字母表（即键盘按键上的符号）所构成的文本。这些文本在计算机中自动翻译成机器语言的字节序列，即由 0 和 1 两个数字组成的字母表的符号序列，这两位数字代表计算机的替代技术状态。通过这些文本及其技术过程的翻译，计算机的物理机器开始运行。下面，首先讨论一个由不同类型的自动机和机器所理解的形式语言构成的一般系统。人类的自然语言，以及其他生物的交流手段，都被认为是在特殊情境下的特殊情况。

【定义】 字母表 Σ 是一组有限的（非空的）符号（根据应用，也称为字符或字母）。例如：

$\Sigma_{\text{bool}}=\{0,1\}$是机器语言的布尔字母表；

$\Sigma_{\text{lat}}=\{a,b,\cdots,z,A,B,\cdots,Z\}$是一些自然语言的拉丁字母；

Σ_{keyboard} 由 Σ_{lat} 和键盘上的其他符号组成，如 B.！、′、§、$ 和空格字符（作为符号之间的空白）。

关于 Σ 的一个词是一个有限的或空的符号序列。ε 被称为空词。单词 w 的长度 $|w|$ 表示一个单词的符号数（对于空单词，$|\varepsilon|=0$；而对于键盘空格，$|␣|=1$）。单词的例子包括：

布尔字母表 Σ_{bool} 上的“010010”，

键盘的字母表 Σ_{keyboard} 上的“Here we go!”。

表示字母表 Σ 上所有单词的集合。

例如：$\Sigma^{*}_{\text{bool}}=\{\varepsilon,0,1,00,01,10,11,000,\cdots\}$

字母表 Σ 上的一种语言 L 是 Σ 的子集。

来自 Σ^{*} 的单词 w 和 v 的联结与 wv 结合在一起。因此，L_1L_2 是语言 L_1 和 L_2 的连接，这两种语言是由连接词 wv 派生的，w 来自 L_1，v 来自 L_2。

自动机或机器什么时候能识别一种语言？

一种算法（如图灵机或计算机——根据丘奇的论文）识别一个字母表 Σ 上的一种语言 L，如果它能从 Σ^{*} 中判定所有符号序列 w 是否是 L 中的一个词，我们认为它具备识别语言 L 的能力。

区分自动机和能够识别不同复杂度语言的机器。有限自动机是一种特别简单的自动机，它可以在有限内存的基础上毫不延迟地描述过程。例如电话线路、加法、操作咖啡机或控制电梯。乘法不能用有限自动机进行，因为在处理过程中有延迟的中间计算是必要的。

这也适用于单词的比较，因为它们可以是任意长度的，并且不能再被缓冲在有限的内存中。

一个有限自动机如图 5.1 所示。这里有一个存储程序，一个带输入单词的磁带和一个读磁头，它们只能在磁带上从左向右移动。这个输入磁带可以理解为输入的线性存储器，它分为几个部分，每个字段作为一个存储单元，包含字母表 Σ 的符号。

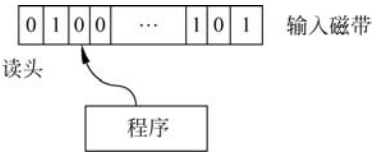

图 5.1　有限自动机示例图

对于语言识别，有限自动机的工作是从字母表 Σ 上输入一个单词 w 开始的。在输入过程中，有限自动机处于某种状态 s_0，每一个有限自动机都有一组接受状态（或结束状态）的特征。在进一步的处理步骤中，符号序列和机器各自的状态发生变化，直到最后，经过许多有限的步骤，到达状态 s 中的空字 ε。当这个最终状态属于自动机的可分辨接受状态时，则有限自动机已经接受了这个词。在另一种情况下，单词 w 被自动机拒绝。因此，有限自动机在读取输入字的最后一个字母后，如果它处于接受状态，则接受该输入单词。

【定义】 一个有限自动机 FA 所接受的语言 L(FA)由 Σ^* 中的接受单词 w 组成。

有限自动机 FA 所接受的所有语言的类 L(FA)称为正则语言类。

正则语言以正则表达式（单词）为特征，通过交替、联结和重复从字母表的符号中产生。例如，考虑字母表 $\Sigma=\{a,b,c\}$，正则语言的一个例子包含由任意数量的 a（a 的重复，例如 a、aa、aaa，…）或任意数量的 b（b 的重复，例如 b、bb、bbb、…）所组成的所有单词的语言；正则语言的另一个例子包含所有以 a 开头、以 b 结尾、中间只有 c 的重复的单词，例如 acb、$accccb$。

为了证明一种语言不是正则的，只要证明没有接受它的有限自动机就足够了。除了当前状态外，有限自动机没有其他的存储可能性。因此，如果一个有限自动机在读了两个不同的单词后，再次以相同的状态结束，它就不能再区分这两个单词了：它“忘记”了这个区别。

【定义】 确定性有限自动机是由确定性过程决定的。每个配置都由机器的状态和单词定义。一个程序完全而明确地从机器状态和相关联的单词确定配置的顺序。

非确定性有限自动机允许在某些配置中选择几个可能的后续配置。

因此，一个不确定的算法可能导致指数级多的可能性。一般来说，没有比用确定性算法模拟所有可能方案更有效的方法了。即使在有限自动机的情况下，也可以证明语言识别可能性的非确定性扩展并没有带来任何新的东西：确定性有限自动机接受的语言与非确定性有限自动机相同。

一个图灵机可以理解为一个有限自动机的扩展，包括：

（1）包含程序的有限控件；

（2）作为输入磁带和内存的无限长的磁带；

（3）可双向移动磁带的读/写磁头。

图灵机类似于有限自动机，它在有限的字母表上工作，使用的磁带在开头包含一个输入字。与有限自动机不同，图灵机也可以使用无限的磁带作为存储器。有限自动机可以扩展到图灵机，方法是用读/写头代替读头，并将其向左移动。

一个图灵机(TM)由初始状态、接受状态和拒绝状态决定。当 TM 到达接受状态时，它接受输入单词，而不管读/写头在磁带上的什么位置。当 TM 到达拒绝状态时，它拒绝输入单词并停止。然而，即使一个单词在有限步数的输入后没有停止，也会被 TM 拒绝。

【定义】 一个图灵机 TM 所接受的语言 L(TM)由 Σ^* 中的接受单词 w 组成。一个图灵机 TM 所接受的所有语言的类 L(TM)称为递归可枚举语言类。

一种语言被称为递归的或可判定的，如果有一个图灵机 TM，它可以用于 Σ^* 中的所有单词 w，来决定 w 是否被接受(并且属于该语言)或不被接受(因此不属于该语言)。

根据丘奇论题(见第 3.4 节)，图灵机是计算机的逻辑数学原型，与它作为超级计算机、笔记本电脑或智能手机的技术实现无关。然而，实际的计算机具有所谓的冯·诺依曼体系结构，其中用于程序和数据的存储器、中央处理单元和输入在技术上是独立的单元。在图灵机中，输入和存储器合并成一个磁带单元，读和写合并成一个读/写磁头。这在理论上不是问题，因为多磁带图灵机有多个自己的读/写头的磁带，然后它们接管了冯·诺依曼体系的独立功能。从逻辑上讲，单磁带图灵机等价于多磁带图灵机，这意味着单磁带机可以模拟多磁带机。

与有限自动机类似，确定性图灵机可以扩展到非确定性图灵机。一个非确定性图灵机最终可以在一个输入单词之后跟随许多选择。这些加工操作可以图形化地想象为一棵分枝树。如果这些操作中至少有一个结束于图灵机的接受状态，则接受输入单词。深度搜索作为这种分支树的加工策略与广度搜索不同。在深度搜索中，会逐个测试分支树的每个分支，以检查它是否以可接受的最终状态结束。在广度搜索中，所有分支同时被测试到一定深度，以判断其中一个分支是否达到接受状态。这个过程逐步地重复，直到机器随之停止运转。通过分支树的广度搜索，非确定性图灵机可以被确定性图灵机模拟。

一般来说，没有更有效的非确定性算法的确定性模拟被称为非确定性算法的所有计算的逐步模拟。然而，这是有代价的：当非确定性被确定性模拟时，计算时间呈指数级增长。到目前为止，还不知道是否存在更有效的模拟，然而这种模拟是否存在还尚未得到证实。

从自然语言中已经习惯了这样一个事实：它们的单词和句子是由语法规则决定的。每种语言都可以由一种语法决定，即一套适当的规则体系。a、b、c 等终端符号和非终端符号(非终端)的数字 A、B、C，…；X、Y、Z，…不同。非终端符号的使用类似于变量(空格)，可以用其他词代替。

【举例】 一个语法示例：

终端符号：a，b

非终端符号：S

规则：

R_1：$S \rightarrow \varepsilon$

R_2：$S \rightarrow SS$

R_3：$S \rightarrow aSb$

R_4：$S \rightarrow bSa$

单词 *baabaabb* 的派生词：

$S \rightarrow R_2\ SS \rightarrow R_3\ SaSb \rightarrow R_3\ SaSSb \rightarrow R_4\ bSaaSSb \rightarrow R_1\ baassb$

$\rightarrow R_3\ baabSaSb \rightarrow R_1\ baabaSb \cdots \rightarrow R_3\ baabaaSbb \rightarrow R_1\ baabaabb$

显然，语法是生成符号序列的非确定性方法。几个规则允许有相同的左侧。此外，如果存在多个选项，则不指定首先将哪个规则应用于单词中的替换。

在语言学中，语法是用来从句法上描述自然语言的。句法范畴如<*sentence*>、<*text*>、<*noun*>和<*adjective*>作为非终结语引入。可以用适当的语法规则派生出文本。

【举例】 用语法规则推导文本：

```
<text>→<sentence><text>
<sentence>→<subject><verb><object>
<subject>→<adjective><noun>
<noun>→<tree>
<adjective>→[green]
```

乔姆斯基(N. Chomsky)认为，可以给出不同复杂度的语法层次。由于相应的语言是由语法规则生成的，他也把它们称为生成语法，如以下定义。

【定义】

1. 正则语法

最简单的类是正则语法，它精确地创建了正则语言这个类别。一个规则语法的规则形式为：$X \rightarrow u$ 和 $X \rightarrow uY$，对于一个终端 u 和非终端 X 和 Y。

2. 上下文无关语法

所有规则都有 $X \rightarrow \alpha$ 的形式，其中有一个非终端 X 和一个来自终端和非终端的单词 α。

3. 上下文相关语法

在规则中，$\alpha \rightarrow \beta$ 是单词长度不大于单词 β 长度的单词 α 的长度。因此，导出结果中没有部分词 α 可以被一个短的部分词 β 代替。

4. 无限语法

这些规则不受任何限制。

上下文无关语法与正则语法的不同之处在于，一个正则规则的右侧最多包含一个非终结符。与无限语法不同，上下文相关语法不包含左侧单词长度大于右侧单词的规则。因此，在计算机中，无限语法可以生成任意内存内容，从而模拟任意派生词。

不同的语法与机器和识别这些语言的机器有什么关系？可以为每个正则语法指定一个等价的有限自动机，它可以识别相应的正则语言。相反，可以为每一个有限自动机指定一个

等价的正则语法，用它生成相应的正则语言。

上下文无关语法创建上下文无关的语言。可以引入下推自动机，作为识别上下文无关语言的一种合适的自动机类型。

【定义】 下推自动机，如图 5.2 所示，有一个输入磁带，它在开始处包含输入单词。与有限自动机一样，读取头只能从左向右读取和移动。因此，磁带只能用来读取输入，而不能像图灵机那样作为内存。然而，与有限自动机不同，下推自动机在读取符号后不必随读头向右移动，它可以保持在磁带的同一区域，并对存储区中的数据进行编辑。下推自动机只能访问和读取存储区的顶部符号，如果要访问更深层次的数据，则必须不可撤销地删除以前的数据。原则上，存储区是一个不受限制的磁带，最终有许多访问的可能性。

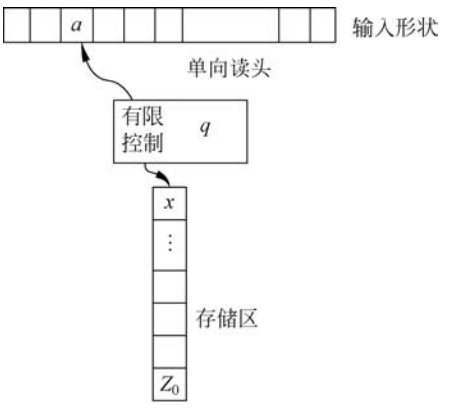

图 5.2　下推自动机的结构

因此，存储区的自动机从输入磁带上的读符号、最终控制的状态和存储区的顶部符号开始工作。在进一步的操作中，它改变状态，用读头向右移动一个字段，并用一个单词 α 替换存储区最上面的符号 X。

非确定性下推自动机准确识别上下文无关语言的类别。因此，非确定性的存储区机等价于上下文无关语法，后者正好生成上下文无关语言。在计算机科学中，上下文无关语法适合于表示编程语言。上下文无关语法生成的单词对应于所构建的编程语言的正确程序，因此上下文无关语法适合构建编译器。编译器是计算机程序，它将用特定编程语言编写的另一个程序翻译成可以由计算机执行的形式。

在乔姆斯基层次结构中，遵循上下文相关语言，这些语言是由上下文相关语法生成的。上下文相关语言被识别为图灵机的一种受限机类型：一个线性有限自动机是一种图灵机，其工作磁带受输入单词长度的限制。使用两个附加符号来标记输入字的左端或右端，并且在处理过程中不能超过这两个符号。

非确定性线性约束自动机识别的语言集等价于上下文相关语言集。到目前为止，还没有证明确定性线性有限自动机是否接受与非确定性自动机相同的语言类。

【重点】 无限的语法能精确地生成由图灵机识别的递归可枚举语言。因此，递归可枚举语言集正是可以由语法生成的所有语言的类别。

因此，不能递归枚举的语言只能被位于图灵机之外的机器所识别，也就是说，直观地“可以比图灵机做得更多”。这是人工智能的核心问题，即智能是否可以简化为作为计算机原型的图灵机，或者更多。

生成语法不仅生成句法符号序列，它们也决定句子的意思。乔姆斯基首先将句子的表面当作短语和短语组成的结构来分析，它们被进一步的规则分割成更多的部分，直到最终一个自然语言句子中的单个单词可推导。然后，句子由名词性短语和动词性短语组成，名词性短语由冠词和名词组成，动词性短语由动词和名词性短语组成，等等。因此，句子可以有不同的语法深度结构，以表示不同的意义。

因此，同一个句子可以有不同的意思，这是由不同的语法深度结构决定的。在图 5.3 中，“She drove the man out with the dog”，这个句子的意思是女人在狗的帮助下把男人赶出去(a)；但是，这个句子也可以有这样的意思：一个女人赶走了一个带着狗的男人(b)。产生规则如下，其中$<S>$指代 sentence(句子)，$<NP>$指代 nominal phrase(名词性短语)，$<VP>$指代 verbal phrase(动词短语)，$<PP>$指代 prepositional phrase(介词短语)，$<T>$指代 article(冠词)，$<N>$指代 noun(名词)，$<V>$指代 verb(动词)，$<P>$指代 preposition(介词)，$<Pr>$指代 pronouns(代词)。

$<S>\rightarrow<NP><VP>$	$<Pr>\rightarrow$[she]
$<NP>\rightarrow<T><N>$	$<V>\rightarrow$[drove]
$<NP>\rightarrow<Pr>$	$<T>\rightarrow$[the]
$<NP>\rightarrow<NP><PP>$	$<T>\rightarrow$[the]
$<VP>\rightarrow<V><NP>$	$<N>\rightarrow$[man]
$<VP>\rightarrow<VP><PP>$	$<N>\rightarrow$[dog]
$<PP>\rightarrow<P><NP>$	$<P>\rightarrow$[with]

生成语法是对这种递归产生式规则的计算，它也可以通过图灵机来实现。利用这一生成语法，导出了 a 和 b 不同意义的两个深度结构，如图 5.3 所示。

自然语言只在句子的表层结构上有所不同。乔姆斯基认为，生产规则的使用是普遍通用的。使用一个模拟有限多个递归产生式规则的图灵程序，可以生成任意数量的句子及其深度语法。

语言哲学家福多(J. Fodor)仍然相信乔姆斯基理论，因为他就语言的深度结构和普遍性假设了心理上真实的认知结构，这是所有人类固有的。精神被理解为一个具有普遍性和内在性的语义表征系统，所有概念都可以分解到其中。福多说的是一种“思想的语言”。

然而，人与人之间的交流绝不局限于对事实的看法交换，交流是由追求意图并引发环境变化的言语行为组成的。继英国语言哲学家奥斯汀(J. L. Austin)之后，美国哲学家赛厄(J. Searle)引入了言语行为的概念。一种言语行为，如“你能告诉我某个人的情况吗?”是由各种动作成分决定的。首先，必须观察反应的传递过程(言语行为)。言语行为与说话人的

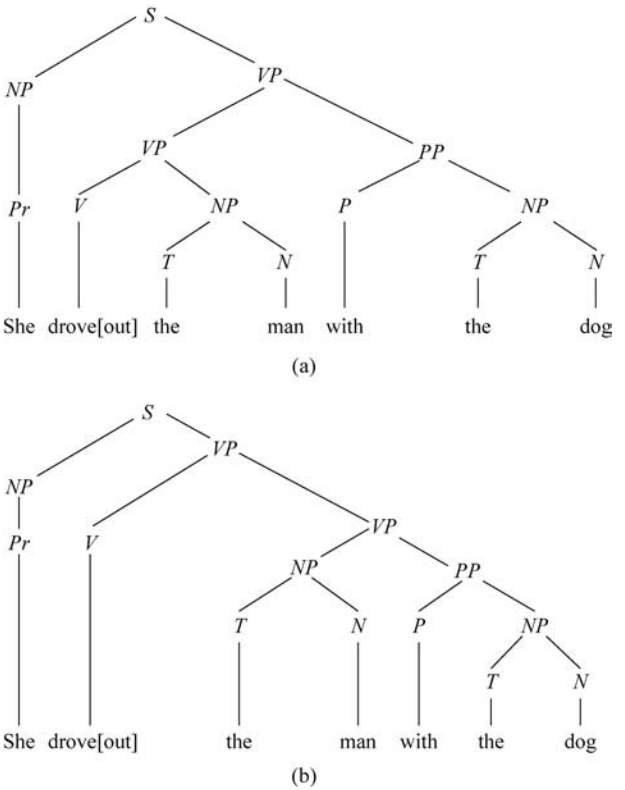

图 5.3　基于乔姆斯基语法体系的语义深度结构

某些意图有关，例如请求、命令或疑问（言外行为）。语效行为记录了言语行为对信息接收者的影响，例如是否愿意提供关于某个人的信息。

言语行为理论成为了计算机语言中知识和查询管理语言（Knowledge and Query Manipulation Language，KQML）的模型，它定义了互联网上搜索程序（代理）之间的通信。KQML 代理语言提供了相互识别、建立联结和交换消息的协议。在消息级别，定义了言语行为类型，这些类型可以用不同的计算机语言来表达。

在技术上，第一步是开发最有效的部分解决方案，即利用计算机程序识别、分析、传输、生成和合成自然语言通信。这些技术解决方案不必模仿人脑的语音处理，而是还可以通过其他方式实现可类比的解决方案。因此，为了达到计算机程序有限通信的目的，没有必要对所有语言层（包括意识层）进行技术模拟。

事实上，在技术高度发达的社会，也依赖于隐性和程序性的知识，而这些知识只能在有限程度上被捕获为规则。与人打交道时的情感、社会和情境知识只能在有限范围内用规则来表述。然而，为了为计算机等技术设备设计用户友好的用户界面，这些知识是必要的。人工智能也应该面向用户的需求和直觉，而不是用复杂的规则让用户承受过重的负担。

5.3 我的智能手机何时会理解我

人类的语言理解是由大脑相应的功能实现的。因此，使用模拟大脑的神经网络和学习算法是有意义的(见第 7.2 节)。计算神经科学家塞诺夫斯基(T. J. Sejnowski)提出了一种神经网络，它可以模拟在类似大脑的机器中学习阅读时的神经交互作用。人类大脑中的神经元是否真的以这种方式相互作用还不能从生理学上加以确定。然而，一个称为 NETalk 的人工神经网络能够从相对较少的神经元构建块中，产生类似于人类的学习过程，这仍然是一个惊人的成就(见图 5.4)。如今的计算机能使系统的速度大大提高，如果这个系统不像以前的人工神经网络那样在传统的(顺序工作的)计算机上模拟，而可以通过相应的硬件或活细胞的“湿件”来实现，NETalk 也会变得有趣。

【举例】 作为 NETalk 的输入，文本被逐字输入，如图 5.4 所示。由于周围其他字符对于一个字符的发音很重要，所以在所讨论字符之前和之后的三个符号也会被标记。每一步读取的七个字符中的每一个都由对应于字母表中字母、标点符号和空格的神经元来检查。输出表示文本的语音发音，每个输出神经元负责声音形成的一个组成部分，普通的常规合成器将这些声音成分转换成可听见的声音。阅读的学习过程是决定性的，它在输入的文本和输出的发音之间自组织起来。还要插入第三层神经元，其与输入和输出神经元的突触联结用数值权重模拟。

在训练阶段，系统首先学习样本文本的发音。系统中并没有明确的声音形成规则的程序。文本的发音是由神经元间的突触联结储存起来的。在未知文本的情况下，它最初随机发音的声音会与标准文本的期望声音进行比较。如果输出不正确，系统将反向工作到内部层次，并检查为什么网络会导致此种输出、哪些联结的权重最高(因此对输出的影响最大)，然后改变权重以逐步优化结果。因此，NETalk 是在采取了鲁梅尔哈特(D. Rumelhart)等提出的反向传播学习算法之后才开始如此工作的(见第 7.2 节)。

这个系统通过“做中学”而不是以规则为基础，以与人类相似的方式学习阅读。在新的阅读尝试中，该系统像小学生一样改进了发音，最终错误率约为 5%。

但是，真的首先需要大脑神经语音处理的知识，才能使用人工智能软件进行语言处理吗？随着计算机性能不断提高，过去伽利略(Galilei)和托马斯·冯·阿奎那(Thomas von Aquinas)等人的个人作品已经被数字化存储和编目。谷歌公司为全球文学的系统数字化开辟了新的可能性，现在称之为“数字人文”。实际上，数字人文的方法不仅仅是文本的数字化，而是利用大数据的方法(见第 10.1 节)，即不必详细了解内容，就可以从数据中获取某些信息。在生态医学的研究领域，旧手稿的元数据是通过算法创建的，目的是得出关于它们的起源地、生产条件和上下文关系的结论。元数据很重要，例如页面格式、铭文、寄存器或旁注。

ePeotics 项目研究文学术语在一个历史时期的传播，由此可以得出这一时期文学理论发展的结论。一个科学家只能阅读有限数量的文本，为了捕捉时代和风格并分类，成千上万

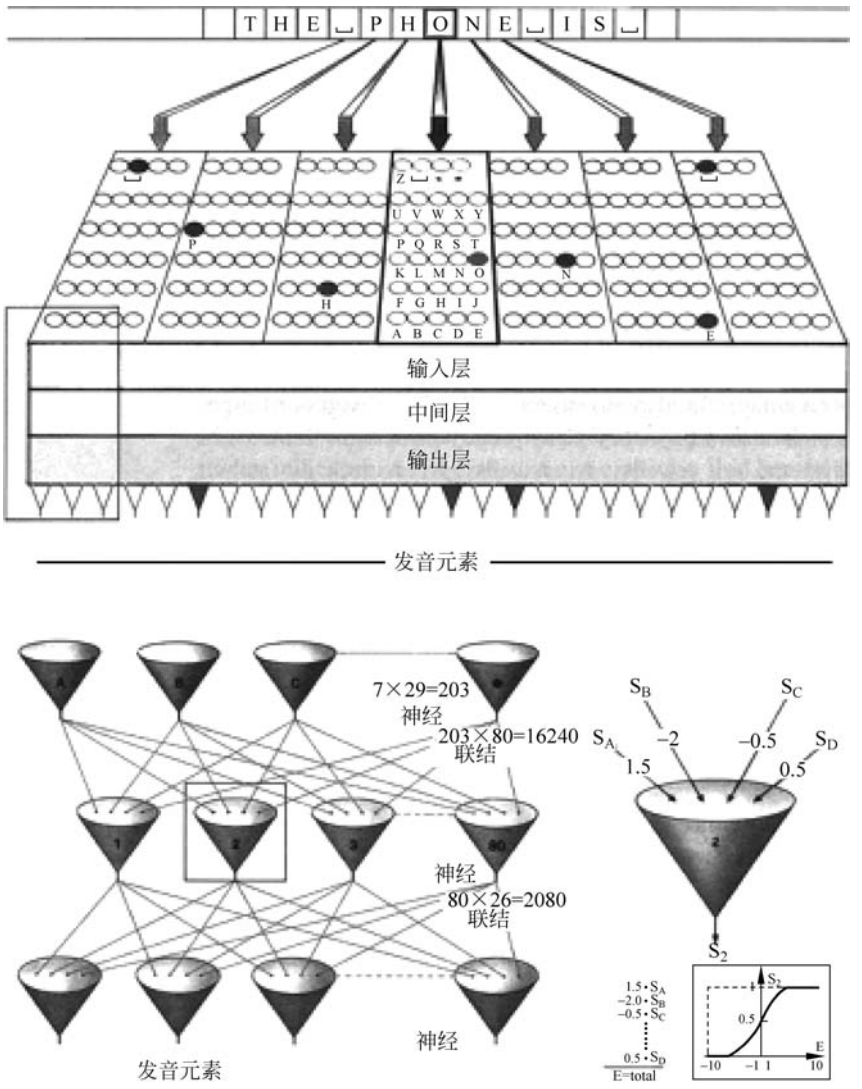

图 5.4　NETalk 学习阅读

的小说和短篇小说可能是必要的。适当的软件可以快速地传递相关性，并用图表生动地加以说明。然而，有一个批判性的保留：最终，超级计算机并不能取代文学学者的评价和解读。然而，正如语义网所显示的，合适的软件能够识别语义上下文。文学学者仍然认为计算机“只”在句法上改变符号，但他们还没有理解这种情况的严重性及其目标。

下一步是使用自动撰写文本的软件代理（机器人）。撰写简单的文本，就像在社交媒体上常见的那样，这一点也不奇怪。是不是已经用机器人而不是人类发推特了？并且，即使在新闻业的某些领域，机器人也会取代或至少支持文案撰稿人。叙事人科学（Narrative Science）公司提供了一种软件，可以自动在期刊上发表文章。公司可以使用软件撰写内容，

例如自动证券交易报告。写作程序可以根据作者的风格进行调整。通过联结到数据库，可以快速发布文本。银行使用这些文本，能对新数据立即做出反应，以便比竞争对手更快地获利。同样，对于大数据来说，重要的不是数据的正确性，而是反应的速度。只要各方使用相同的数据，信息的质量和可靠性就不会影响取胜的机会。

自从 ELIZA 问世以来，基于模式识别的文本比较已经为人们所熟知。如今的软件现在将句子分解成单独的短语，并计算出问题的适当答案模式或以光速翻译成其他语言的可能性。VERBMOBIL 就是一个高效翻译程序的例子。

【举例】 VERBMOBIL 是由德国人工智能研究中心(DFKI)于 1993—2000 年协调的一个项目。具体来说，通过两个麦克风将口语传输到德语、英语或日语的语音识别模块，并进行韵律分析(语音度量和节奏分析)。在此基础上，通过对句子的语法深度分析和对话处理的规则，纳入语义信息并进行综合加工。VERBMOBIL 由此实现了从口语识别到对话语义的转变，对话语义不仅仅局限于短语块的交换，还包括长语块，因为它们是典型的自然语言。

人类的语音处理经历了不同层次的表征。在技术系统中，人们试图逐个实现这些步骤。在计算机语言学中，这个过程被描述为一个流水线模型：从声音信息(听觉)开始，下一步将生成文本形式，相应的字母串被记录为单词和句子。在词法分析中，分析人称形式，并将文本中的单词追溯到基本形式。在句法分析中，强调句子的语法形式，如主语、谓语、宾语、形容词等，如乔姆斯基语法(见第 5.2 节)。在语义分析中，句子被赋予意义，就像乔姆斯基语法的深度结构一样。最后，在对话和语篇分析中，考察诸如提问和回答、计划、目的和意图之间的关系。

正如将在后面看到的那样，高效的技术解决方案绝不需要贯穿这个流水线模型的所有阶段。如今强大的计算能力，再加上机器学习和搜索算法，开启了数据模式的开发，这些模式可用于所有级别的高效解决方案。用于深度结构语义分析的生成语法很少用于此目的，而且定向在人类的语义信息加工中也不起作用。人类中的语义过程通常与意识相联系，这绝不是必需的，如下例所述。

【举例】 IBM 的 WATSON 程序是一个语义问答系统，它利用并行计算机的计算能力和维基百科的存储能力。WATSON 理解语境和语言游戏的语义是与 ELIZA 不同的。WATSON 是一个语义搜索引擎(IBM)，它捕捉自然语言提出的问题，并短时间内在大型数据库中找到合适的事实和答案。它基于海量数据(大数据)的计算和存储能力，集成了许多并行语言算法、专家系统、搜索引擎和语言处理器，如图 5.5 所示。

WATSON 程序不是面向人脑的，而是依赖于计算能力和数据库能力。然而，系统通过了图灵测试。风格分析适应说话人或作家的习惯，因此写作风格的个性化不再是一个不可逾越的障碍。

WATSON 现在成了在 IBM 平台用于认知工具及其在商业和企业中的不同应用。根据摩尔定律(见第 9.4 节)，在可预见的将来，WATSON 的服务将不需要超级计算机，智能手机中的一款应用程序也能提供同样的性能。终于可以用我们的智能手机来实现这些功能

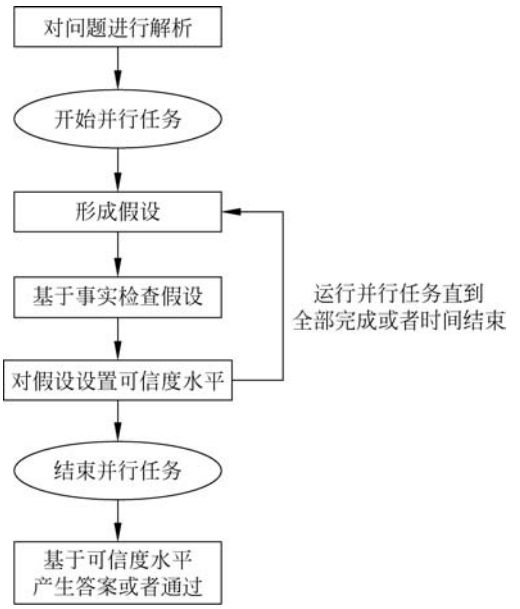

图 5.5　WATSON 的体系结构

了，这些服务不再需要通过键盘，而是通过智能语音程序来进行。即使是关于亲密感情的谈话也不能排除，正如魏泽鲍姆所担心的那样。

2013 年斯派克·琼兹(Spike Jonze)执导的美国科幻电影《她(Her)》讲述了一个内向腼腆的男人爱上了一门语言学课程。从专业角度讲，这个男人给那些很难与对方交流感情的人写信，为了减轻自己的痛苦，他购买了一个新的操作系统，配备了女性身份和悦耳的声音。通过耳机和摄像机，他与萨曼莎(Samantha)进行关于系统如何自我调用的交流，萨曼莎很快学会了社交互动，行为越来越人性化。在频繁、长时间和激烈的交谈中，最终发展起来一种亲密的情感关系。

智能写作程序的使用不仅在媒体和新闻业中是有广泛前景的，同时也可涉及商业新闻、体育报道或小报公告等常规文本。委托给机器人程序的常规文本也可用于管理或司法领域。还将体验自动写作程序在科学中的应用，在医学、技术和自然科学杂志上发表的文章已经如此众多，以至于即使是在特殊的研究领域，各自的专家也无法看到它们的细节。为了在竞争中生存，必须迅速发表研究成果。因此，可以想象，科学家和学者只需以通常的语言结构(例如预印本)输入数据、论据和结果，机器人便会根据作者的写作风格，通过数据库发布。

在金融领域，书写机器人日益成为日常工作的一部分。像叙事科学(Narrative Science)或者自动化透视(Automated Insight)这样的企业使用智能软件，将投资银行的季度数据翻译成新闻文本，而此前则是靠记者们辛苦地在季报中撰写这样的文字。自动机器会在几秒钟内生成多个由人类编写的报告。在金融领域，算法以光速为分析部门生成公司简介；自动编写程序可以告诉客户基金经理投资股票市场的策略以及基金的表现；保险公司使用智

能写作程序来衡量销售业绩，并解释改进建议；自动生成的文本使客户能够确认其投资策略是否正确；由自动编写程序提供的支持也为个人客户建议创造了更多的时间。随着RoboAdvice的出现，人工智能在投资咨询和资产管理领域的发展也越来越迅速。如果该系统现在除英语外，还作用于德语、法语和西班牙语，那么适用范围将扩大。人力投资顾问并没有被取代，但数字化产品的速度非常快，并且与信息技术工具的指数级增长相协调。

参考文献