第1章 计算语言学的兴起和发展

本章首先说明计算语言学的性质，然后按萌芽期、发展期、繁荣期的顺序向读者展示计算语言学这个学科的发展历程。

第一节 计算语言学的性质

计算语言学(computational linguistics)是用计算机研究和处理自然语言的一门新兴的边缘学科。由于计算语言学的研究对象是自然语言，因此，语言学家把它算为语言学的一个分支；由于计算语言学要采用先进的计算机科学技术来研究和处理自然语言，因此，计算机科学家把它算为计算机科学的一个分支；由于计算语言学要研究自然语言的形式结构和自然

语言处理的算法，因此，数学家把它算为应用数学的一个分支。这种情况说明，计算语言学不是一门单纯的学科，而是一门边缘性学科。

为什么计算语言学会出现这种跨学科的性质呢？这是由计算语言学的研究目标决定的。计算语言学既然以用计算机研究和处理自然语言为其目标，那么，它就必定要认真地研究计算机处理自然语言的全过程，提出有效的理论和方法。

一般地说，计算机对自然语言的研究和处理应当经过如下四个过程：

第一，从语言学的角度提出自然语言处理的问题和理论（linguistic problem）；

第二，把需要研究的语言学问题加以形式化（linguistic formalism），使之能以一定的数学形式或者接近于数学的形式，严格而规整地表示出来；

第三，把这种严格而规整的数学形式表示为算法（algorithm），使之在计算上形式化（computational formalism）；

第四，根据算法编写计算机程序，使之在计算机上加以实现（computer implementation）。

因此，为了研究计算语言学，研究人员不仅要具备语言学的知识，而且还要具备数学和计算机科学方面的知识，这样，计算语言学就成为了一门介乎于语言学、数学和计算机科学之间的边缘性交叉学科，它同时涉及到文科、理科和工科三大

领域，使它具有明显的跨学科性质。

上述的第一、第二个过程属于计算语言学的理论部分，第三和第四个过程属于计算语言学的方法部分。我们有时把第三和第四个过程叫做自然语言的计算机处理（Natural Language Processing by Computer），简称自然语言处理(Natural Language Processing)。本书主要讲述计算语言学的基础理论而不过多地讨论方法，至于自然语言处理的各种方法，本书仅在说明理论时作简略的介绍，不作详尽的叙述。

我们希望计算语言学研究人员同时具备语言学、数学和计算机科学的知识，成为文理兼通、博学多识的人才。对于不可能同时具备语言学、数学和计算机科学知识的研究人员，至少对于自己原来所学的专业是精研通达的内行，对于另外两个专业不是似懂非懂的外行，这样，才有可能有效地从事计算语言学研究。因此，我们应该提倡计算语言学的研究人员不断地进行更新知识的再学习:“活到老,学到老”,对于计算语言学研究人员来说，决不是一句装扮门面的空话，而应该成为身体力行的座右铭。

第二节　计算语言学的萌芽期

同其他任何学科一样，作为一门新兴边缘科学的计算语言学是在一定的社会历史条件下应时代的要求而逐渐成长起

来的，计算语言学的成长历程可以分为萌芽期、发展期和繁荣期三个时期。

人类对于计算语言学的研究早在“计算语言学”这个名称出现之前就开始了。人类历史上最早的计算语言学研究就是机器翻译(machine translation)。

《圣经·创世纪》中说，古代人类说原是一种统一的语言，交流思想非常方便，劳动效率也很高，他们曾经想建立一座高达天庭的通天塔，叫做“巴比塔”，来显示他们的丰功伟绩。建造巴比塔的壮举震惊了上帝，上帝便施伎俩，让不同的人说不同的语言，使人们难以交流思想，无法协调工作，以此来惩罚异想天开的巴比塔建造者。结果，巴比塔没有建成，而语言的不同，却成为人们相互交往的极大障碍。这样的传说当然是不可信的，但是，语言的障碍却时时刻刻在困扰着人们。

在 17 世纪，一些有识之士提出了采用机器词典来克服语言障碍的想法。笛卡儿（Descartes）和莱布尼兹(Leibniz)都试图在统一的数字代码的基础上来编写词典。在 17 世纪中叶，贝克(Cave Beck)、基尔施(Athanasius Kircher)和贝希尔(Johann Joachim Becher)等人都出版过这类的词典。由此开展了关于“普遍语言”的运动，一些人试图在逻辑原则和图形符号的基础上，创造出一种无歧义的语言，这样一来，人们就不必再由于误解而产生交际方面的困惑了。维尔金斯（John Wilkins）在《关于真实符号和哲学语言的论文》(An Essay towards a Real Character and Philosophical Language, 1668)中

提出的中介语(Interlingua)是这方面最著名的成果，这种中介语的设计试图将世界上所有的概念和实体都加以分类和编码，有规则地列出并描述所有的概念和实体，并根据它们各自的特点和性质，给予不同的记号和名称。

1903 年,古图拉特(Couturat)和洛(Leau)在《通用语言的历史》一书中指出，德国学者里格(W. Rieger)曾经提出过一种数字语法(Zifferngrammatik)，这种语法加上词典的辅助，可以利用机械将一种语言翻译成其他多种语言，首次使用了“机器翻译”(德文是 ein mechanisches Uebersetzen)这个术语。

20 世纪 30 年代之初，亚美尼亚裔的法国工程师阿尔楚尼(G.B. Artsouni)提出了用机器来进行语言翻译的想法，并在 1933 年 7 月 22 日获得了一项‘翻译机’的专利,叫做‘机械脑’(mechanical brain)。这种机械脑的存储装置可以容纳数千个字元，通过键盘后面的宽纸带，进行资料的检索。阿尔楚尼认为它可以应用于记录火车时刻表和银行的账户，尤其适合于作机器词典。在宽纸带上面，每一行记录了源语言的一个词项以及这个词项在多种目标语言中的对应词项，在另外一条纸带上对应的每个词项处，记录着相应的代码，这些代码以打孔来表示。要查询的词项也利用键盘打孔来表示，检索一个词项的时间大约时长 10 到 15 秒。阿尔楚尼的原型机于 1937 年正式展出，引起了法国邮政、电信部门的兴趣。但是，由于不久爆发了第二次世界大战，阿尔楚尼的机械脑无法安

装使用。

1933 年，苏联发明家特洛扬斯基(П.П.ТРОЯНСКИИ)设计了用机械方法把一种语言翻译为另一种语言的机器，并在同年 9 月 5 日登记了他的发明。特洛扬斯基认为翻译可以分为三个阶段，第一个阶段由只懂源语言的编辑，将输入的原文分析成特定的逻辑形式，将带有屈折词尾的变形词还原成原形词，并分析出各个单词的句法功能，为此，他创造了一套逻辑分析符号。第二阶段是利用他的翻译机，把源语言的原形词和逻辑符号转换成目标语言的原形词和符号。第三阶段由只懂目标语言的编辑，把目标语言的原形词和符号转换成目标语言。特洛扬斯基认为，他的翻译机只能在第二阶段作为自动词典来使用。不过他相信，只要能够建造出一部专门处理逻辑分析过程的机器，总有一天，上述的整个翻译程序都能够用机器来实现。特洛扬斯基这种认识，已经超越了“机器词典”的简单想法，比阿尔楚尼又进了一步。1939 年,特洛扬斯基在他的翻译机上增加了一个用“光元素”操作的存储装置；1941 年 5 月，这部实验性的翻译机已经可以运作；1948 年，他计划在此基础上研制一部“电子机械机”(electromechanical machine)。但是，由于当时苏联的科学家和语言学家对此反应十分冷淡，特洛扬斯基的翻译机没有得到支持，最后以失败告终了。

1946 年，美国宾夕法尼亚大学的埃克特（J. P. Eckert）和莫希莱(J.W.Mauchly）设计并制造出了世界上第一台电子计

算机 ENIAC ,电子计算机惊人的运算速度,启示着人们考虑翻译技术的革新问题。因此，在电子计算机问世的同一年，英国工程师布斯(A. D. Booth) 和美国洛克菲勒基金会副总裁韦弗(W. Weaver) 在讨论电子计算机的应用范围时，就提出了利用计算机进行语言自动翻译的想法。 1947 年 3 月 6 日,布斯与韦弗在纽约的洛克菲勒中心会面，韦弗提出，“如果将计算机用在非数值计算方面，是比较有希望的”。在韦弗与布斯会面之前，韦弗在 1947 年 3 月 4 日给控制论学者维纳 (N. Wiener)写信,讨论了机器翻译的问题,韦弗说“:我怀疑是否真的建造不出一部能够作翻译的计算机？即使只能翻译科学性的文章（在语义上问题较少），或是翻译出来的结果不怎么优雅(但能够理解),对我而言都值得一试。’可是,维纳给韦弗泼了一瓢冷水，他在 4 月 30 日给韦弗的回信中写道：“老实说，恐怕每一种语言的词汇，范围都相当模糊；而其中表示的感情和言外之意，要以类似机器翻译的方法来处理，恐怕不是很乐观的。”不过韦弗仍然坚持自己的意见。1949 年，韦弗发表了一份以《翻译》为题的备忘录，正式提出了机器翻译问题。在这份备忘录中，他除了提出各种语言都有许多共同的特征这一论点之外，还有两点值得我们注意：

第一，他认为翻译类似于解读密码的过程。他说：“当我阅读一篇用俄语写的文章的时候，我可以说，这篇文章实际上是用英语写的，只不过它是用另外一种奇怪的符号编了码而已,当我在阅读时,我是在进行解码。’备忘录中记载了一个有

趣的故事：布朗大学数学系的吉尔曼（R. E. Gilmam）曾经解读了一篇长约一百个词的土耳其文密码，而他既不懂土耳其文，也不知道这篇密码是用土耳其文写的。韦弗认为，吉尔曼的成功足以证明解读密码的技巧和能力不受语言的影响，因而可以用解读密码的办法来进行机器翻译。

第二，他认为原文与译文“说的是同样的事情”，因此，当把语言 A 翻译为语言 B 时，就意味着，从语言 A 出发，经过某一“通用语言”（Universal Language）或“中间语言”（Interlingua），然后转换为语言 B，这种“通用语言”或“中间语言”，可以假定是全人类共同的。

可以看出，韦弗把机器翻译仅仅看成一种机械的解读密码的过程，他远远没有看到机器翻译在词法分析、句法分析以及语义分析等方面的复杂性。

由于学者的热心倡导、实业界的大力支持，美国的机器翻译研究一时兴盛起来。1954 年，美国乔治敦大学在国际商用机器公司（IBM 公司）的协同下，用 IBM－701 计算机，进行了世界上第一次机器翻译试验，把几个简单的俄语句子翻译成英语，接着，苏联、英国、日本也进行了机器翻译试验，机器翻译出现热潮。

早期机器翻译系统的研制受到韦弗的上述思想的很大影响，许多机器翻译研究者都把机器翻译的过程与解读密码的过程相类比，试图通过查询词典的方法来实现词对词的机器翻译，因而译文的可读性很差，难于付诸实用。

1964 Äê£¬ÃÀ¹ú¿ÆÑ§Ôº³É Á¢ÓïÑÔ×Ô¶¯´¦ Àí×ÉÑ¯Î¯ Ô±»á£¨ Automatic Language Processing Advisory Committee £¬¼ò³Æ ALPAC Î¯ Ô±»á£©£¬µ÷²é »úÆ÷ ·ÒëµÄÑÐ¾¿Çé¿ö£¬²¢ ÓÚ 1966 Äê 11 ÔÂ ¹«²¼ÁËÒ»¸ö Ìâ Îª ¡¶ÓïÑÔÓë»úÆ÷ µÄ±¨¸æ£¨¼ò³Æ ALPAC ±¨¸æ£©¶Ô »úÆ÷ ·Òë²É È¡·ñ ¶¨ µÄÌ¬ ¶È£¬±¨¸æÐû³Æ£º ¡° ÔÚÄ¿Ç°¸ø»úÆ÷ ·ÒëÒÔ ´ó Á¦Ö§³ÖÖ»¹ Ã»ÓÐ¶àÉÙÀíÓÉ£»±¨¸æ»¹Ö¸³ö£¬»úÆ÷ ·ÒëÑÐ¾¿Óöµ½ ÁËÄÑÒÔ¿ËË·þ µÄ ¡°ÓïÒåÕÏ° ¡±(semantic barrier)。

ÔÚ ALPAC ±¨¸æµÄÓ°Ïì ÏÂ £¬Ðí¶à¹ú ¼ÒµÄ»úÆ÷ ·ÒëÑÐ¾¿µÍ ³± £¬Ðí¶àÒÑ¾½¨Á¢Æð À´ µÄ»úÆ÷ ·ÒëÑÐ¾¿µ¥Î» Óöµ½ÁËÐÐÕþ ÉÏ ºÍ ¾·ÑÉÏ µÄÀ§ÄÑ£¬ÔÚÊÀ½ç ·¶Î§ ÄÚ£¬»úÆ÷ ·ÒëµÄÈÈ³± Í» È»ÏûÊ§ÁË£¬ ³ö ÏÖÁË¿ÕÇ°ÏÔ Ìõ µÄ¾°Ïó£

¾ÍÔÚÕâÑù¿ÕÇ°ÏÔ Ìõ µÄÇé¿öÏÂ £¬Ò»Ð©ÓÐÊ¶Ö®Ê¿ÇåÐÑµØÈÏÊ¶ µ½´ÓÆËã»ú´¦ Àí ×ÔÈ»ÓïÑÔµÄ½Ç¶ÈÑÐ¾¿ÓïÑÔµÄÖØÒªÐÔ£¬¾ÍÔÚ ALPAC ±¨¸æÖÐ£¬Ê×Î ³ö ÏÖÁË¡° ¼ÆËãÓïÑÔÑ§(computational linguistics £©Õâ¸ö ÊõÓï¡£¾ÝËµÕâ¸ö ÊõÓïµÄÌá ³ö ÕßÊ¾ÍÊÇ ALPAC ±¨¸æµÄ±àÎ¯ Ö®Ò»ÃÀ¹ú ÖøÃûÓïÑÔÑ§¼ÒºÉ£ Ë¹£¨ D. G. Hays) ¡£ËûÊÇ Ê×ÏÈ Ìá ³ö ÒÀ´æÓï·¨ÀíÂÛ£¨ Dependency Theory £©À´½øÐÐ»úÆ÷ ·ÒëµÄ ×Ô¶¯¾ä ·¨ ·ÖÎö £¬ÊÇ»úÆ÷ ·ÒëµÄÏÈ ÇýÕß Ö®Ò»¡£ºÉ£Ë¹µ±Ê±Ìá ³ö ¡° ¼Æ ËãÓïÑÔÑ§¡±Õâ¸ö ÊõÓï£¬ÊÇÏ£ Íû ÑÐ¾¿ÈËÔ±Ç½¸ÈÕÚÔÚ»úÆ÷ ·ÒëÒÔÍâ ÖÊ Á¿µÍ ÁÓÄÑµÄÌõ¼þ£¬»¹¸ü ¶àµÄ¾«Á¦ºÍ Ê±¼ä¾ä×ÔÈ»ÓïÑÔ¼ÆËã»ú ´¦ Àí µÄ»ù±¾ÀíÂÛ£¬¼ÆËãÓïÑÔÑ§¾ÍÊÇ×ÔÈ»ÓïÑÔ¼ÆËã»ú´¦ Àí µÄ»ù ±¾ÀíÂÛ¡£Õâ Ñù£¬Ò»Ð©ÑÐ¾¿µÄ¼ÆËãÓïÑÔÑ§±ãÔÚ»ú Æ÷ ·Òë¿ÕÇ°ÏÔ Ìõ µÄ¾°ÏóÖÐ ÆðÆ¾³ö ÁËËü¸ö Õâ¸ö ÊõÓïÆðÆðÀ´µÄ¡¢ÊÇÇ¼Æ

1966 ALPAC

Understanding Natural Lan-
guage Man – Machine Dialogue)

舞的进展。因此,当 60 年代末期机器翻译困难重重、一筹莫展的时侯,自然语言理解的研究却左右逢源,后来居上,而当机器翻译东山再起、重振旗鼓的时侯,自然语言理解却已获得了累累的硕果。这种局面使计算语言学进入了它的发展期。

自然语言理解系统的发展可以分为第一代系统和第二代系统两个阶段。第一代系统建立在对词类和词序分析的基础之上,分析中经常使用统计方法;第二代系统则开始引进语义甚至语用和语境的因素,几乎完全抛开了统计技术。

第一代自然语言理解系统又可分为四种类型:

(1) 特殊格式系统:早期的自然语言理解系统大多数是特殊格式系统,根据人机对话内容的特点,采用特定的格式来进行人机对话。1963 年,林德赛(R. Lindsay)在美国卡内基技术学院用 IPL-V 表处理语言设计了 SAD-SAM 系统,就采用了特定格式来进行关于亲属关系方面的人机对话,系统内建立了一个关于亲属关系的数据库,可接收关于亲属关系方面的问题的英语句子提问,用英语作出回答。 1968 年,波布洛(D. Bobrow)在美国麻省理工学院设计了 STUDENT 系统,这个系统把高中代数应用题中的英语句子归纳为一些基本模式,由计算机来理解这些应用题中的英语句子,列出方程求解,并给出答案。60 年代初期,格林(B. Green)在美国林肯实验室建立了 BASEBALL 系统,也使用 IPL-V 表处理语言,系统的数据库中存贮了关于美国 1959 年联邦棒球赛得分记录的数据,可回答有关棒球赛的一些问题。该系统的句

法分析能力较差，输入句子十分简单，没有连接词，也没有比较级形式的形容词和副词，主要靠一部机器词典来进行单词的识别,使用了 14 个词类范畴，所有的问题都采用一种特殊的规范表达式来回答。

(2) 以文本为基础的系统：某些研究者不满意在特殊格式系统中的种种格式限制，因为就一个专门领域来说，最方便的还是使用不受特殊格式结构限制的系统来进行人机对话，这就出现了以文本为基础的系统， 1966 年西蒙斯（R. F. Simmons）、布尔格(J. F. Burger ）和龙格(R. E. Long ）设计的 PROTOSYNTHEX－I 系统，就是以文本信息的存贮和检索方式工作的。

(3 有限逻辑系统：有限逻辑系统进一步改进了以文本为基础的系统。在这种系统中，自然语言的句子以某种更加形式化的记号来替代，这些记号自成一个有限逻辑系统，可以进行某些推理。1968 年，拉菲尔(B. Raphael ）在美国麻省理工学院用 LISP 语言建立了 SIR 系统,针对英语提出了 24 个匹配模式，把输入的英语句子与这些模式相匹配，从而识别输入句子的结构，在从存贮知识的数据库到回答问题的过程中，可以处理人们对话中常用的一些概念，如集合的包含关系、空间关系等等，并可进行简单逻辑推理，机器并能在对话中进行学习，记住已学过的知识，从事一些初步的智能活动。1965 年，斯莱格勒(J. R. Slagle)建立了 DEDUCOM 系统,可在情报检索中进行演绎推理。1966 年,桑普逊(F. B. Thompson) 建立

了 DEACON 系统，通过英语来管理一个虚构的军用数据库，设计中使用了环结构和近似英语的概念来进行推理。1968 年，凯罗格(C. Kellog)在 IBM 360/67 计算机上，建立了 CONVERSE 系统，该系统能根据关于美国 120 个城市的 1000 个事实的文件来进行推理。

(4)一般演绎系统：一般演绎系统使用某些标准数学符号(如谓词演算符号)来表达信息。例如：

Some girls are pretty
(有些女孩是漂亮的)

这个英语句子可表示为

$$\exists x(\mathrm{Girl}(x) \ \& \ \mathrm{Pretty}(x)),$$

Every girl is pretty
(所有的女孩都漂亮)

这个英语句子可以表示为

$$\forall x(\mathrm{Girl}(x) \rightarrow \mathrm{Pretty}(x)).$$

其中，$\exists$ 是存在量词，$\exists x$ 表示存在某个 x，$\forall$ 是全称量词，$\forall x$ 表示对于一切的 x，&是合取符号，$\rightarrow$ 是蕴涵符号，表示“如果……，则……”。

这样一来，逻辑学家们在定理证明工作上取得的全部成就，就可以用来作为建立有效的演绎系统的根据，从而能够把任何一个问题用定理证明的方式表达出来，并实际地演绎出所需要的信息，用自然语言作出回答。一般演绎系统可以表达那些在有限逻辑系统中不容易表达出来的复杂信息，从而进一步提高了自然语言理解系统的能力。1968－1969 年，格

林和拉菲尔建立的 QA2, QA3 系统，采用谓词演算的方式和格式化的数据(formated data) 来进行演绎推理，解答问题，并用英语作出回答，这是一般演绎系统的典型代表。

1970 年以来，出现了一定数量的第二代自然语言理解系统，这些系统绝大多数是程序演绎系统，大量地进行语义、语境以至语用的分析。其中比较有名的系统是 LUNAR 系统、SHRDLU 系统、MARGIE 系统、SAM 系统、PAM 系统。

LUNAR 系统是伍兹(W. Woods)于 1972 年设计的一个自然语言情报检索系统，其目的在于帮助地质学家们比较和评价从阿波罗 - 11 火箭得到的关于月球岩石和土壤的组成成分的化学分析数据，这个系统采用形式提问语言 (formal query language) 来表示所提问的语义，从而对提问的句子作出语义解释，最后把形式提问语言执行于数据库，产生出对问题的回答。

SHRDLU 系统是维诺格拉德(T. Winograd)于 1972 年在美国麻省理工学院建立的一个用自然语言指挥机器人动作的系统。该系统把句法分析、语义分析、逻辑推理结合起来，大大地增强了系统在语言分析方面的功能。该系统对话的对象是一个具有简单的“手”和“眼”的玩具机器人，它可以操作放在桌子上的具有不同颜色、尺寸和形状的玩具积木，如立方体、棱锥体、盒子等，机器人能够根据操作人员的命令把这些积木捡起来，移动它们去搭成新的积木结构，在人机对话过程中，操作人员能获得他发给机器人的各种视觉反馈，实时地观

察机器人理解语言、执行命令的情况。在电视屏幕上还可以显示出这个机器人的模拟形象以及它同一个真正的活人在电传机上自由地用英语对话的生动情景。

MARGIE 系统是杉克(R. Schank) 于 1975 年在美国斯坦福人工智能实验室研制出来的。该系统的目的在于提供一个自然语言理解的直观模型。系统首先把英语句子转换为概念依存表达式，然后根据系统中有关信息进行推理，从概念依存表达式中推演出大量的事实。由于人们在理解句子时，总要牵涉到比句子的外部表达多得多的内容，因此，该系统的推理有 16 种类型,如原因、效应、说明、功能等等,最后,把推理的结果转换成英语输出。

SAM 系统是阿贝尔森(R. Abelson)于 1975 年在美国耶鲁大学建立的。这个系统采用“脚本”(script) 的办法来理解自然语言写的故事。所谓脚本，就是用来描述人们活动（如上饭馆、看病）的一种标准化的事件系列。

杉克和阿贝尔森假定，每个人在他自己的生活实践中，会自然而然地意识到这样的脚本，在理解故事时，这些脚本可以用来建立时间发生的语境，因而也就可以用来预料它所代表的事件的情况，并以这些脚本为背景来理解自然语言，对故事中的人物、地点、事件进行推理,在推理过程中,给它们补充新的信息,最后采用‘同义互训’或‘释句’(paraphrase) 的方法，根据计算机理解的结果，由计算机复述原来的故事。复述时，由于在推理过程中补充了许多新的信息，因而所复述的故事

的内容会比原来的故事要丰富得多。计算机似乎像一个有理智的活人，把在推理过程中所推出的新信息加到故事中，添油加醋地把原来的故事说得更加精彩。

PAM 系统是威林斯基（R. Wilensky ）于 1978 年在美国耶鲁大学建立的另一个理解故事的系统。 PAM 系统也能解释故事情节，回答问题，进行推论，作出摘要。它除了“脚本”中的事件序列之外，还提出了“计划”(plan）作为理解故事的基础。所谓“计划”，就是故事中的人物为实现其目的所要采取的手段。如果要通过“计划”来理解故事，就要找出人物的目的以及为完成这个目的所采取的行动。系统中设有一个“计划库”(plan box) ，存贮着有关各种目的的信息以及各种手段的信息。这样，在理解故事时，只要求出故事中有关情节与计划库中存贮的信息相重合的部分，就可以理解到这个故事的目的是什么。当把一个一个的故事情节与脚本匹配出现障碍时，由于“计划库”中可提供关于一般目的的信息，就不致造成故事理解的失败。

杉克等学者还进一步研究语言理解和记忆的关系，概括各种具体知识结构为一般经验，综合句法、语义、知识、推理为一体,建成 FRUMP 和 IPP 两个快速阅读系统。这两个系统存贮 2000 多个英语单词，对输入故事无须逐字逐句地分析，而是跳过某些词语提取故事中的主要信息。这样的系统可以对报刊上一些新闻故事自动地作出摘要。

上述的系统都是书面的自然语言理解系统，输入输出都

是用书面文字。口头的自然语言理解系统，还牵涉到语音识别、语音合成等复杂的技术，显然是更加困难的课题，口头自然语言理解系统的研究近年来也有进展。

在计算语言学的发展期，机器翻译经过萧条之后也逐渐复苏，机器翻译的研究者们从失败中汲取教训，他们痛定思痛，普遍认识到：原语和译语两种语言的差异，不仅只表现在词汇的不同上，而且，还表现在句法结构的不同上，为了得到可读性强的译文，必须在自动句法分析上多下功夫。

早在 1957 年，美国学者英格维(V. Yingve)在《句法翻译的框架》(Framework for Syntactic Translation)一文中就指出，一个好的机器翻译系统，应该分别地对原语和译语都作出恰如其分的描写，这样的描写应该互不影响，相对独立。英格维主张，机器翻译可以分为三个阶段来进行。

第一阶段：用代码化的结构标志来表示原语文句的结构；

第二阶段：把原语的结构标志转换为译语的结构标志；

第三阶段：构成译语的输出文句。

第一阶段只涉及原语，不受译语的影响，第三阶段只涉及译语，不受原语的影响，只是在第二阶段才设计到原语和译语二者。在第一阶段，除了作原语的词法分析之外，还要进行原语的句法分析，才能把原语文句的结构表示为代码化的结构标志。在第二阶段，除了进行原语和译语的词汇转换之外，还要进行原语和译语的结构转换，才能把原语的结构标志变成译语的结构标志。在第三阶段，除了作译语的词法生成之外，