第一届学生计算语言学研讨会(SWCL2002)专题讲座

汉语词法分析和句法分析技术综述

刘群
北京大学计算语言学研究所
中国科学院计算技术研究所
liuqun@ict.ac.cn

引言

本文主要介绍一些常用的汉语分析技术。

所谓语言的分析，就是将一个句子分解成一些小的组成部分（词、短语等等）并了解这些部分之间的关系，从而帮助我们把握这个句子的意义。

语言的研究，一般而言存在四个层面：词法层、句法层、语义层和语用层。

同样，语言的分析也存在四个层面：词法分析、句法分析、语义分析和语用分析。

本文主要介绍汉语的词法分析和句法分析技术。这两种技术是汉语分析技术的基础，而且已经发展得比较成熟。文中也会少量提及语义层面和语用层面的一些问题，但不会做深入的探讨。

汉语是一种孤立语（又称分析语），与作为曲折语和黏着语的其他一些语言相比，汉语在语法上有一些特点，仅仅从形式上看，这种特点主要体现在以下几个方面：

1. 汉语的基本构成单位是汉字而不是字母。常用汉字就有 3000 多个（GB2312 一级汉字），全部汉字达数万之多（UNICODE 编码收录汉字 20000 多）；
2. 汉语的词与词之间没有空格分开，也可以说，从形式上看，汉语中没有“词”这个单位；
3. 汉语词没有形态上的变化（或者说形态变化非常弱），同一个词在句子中充当不同语法功能时，形式是完全相同的；
4. 汉语句子没有形式上唯一的谓语中心词。

这些特点对汉语的分析造成了一定的影响，使得汉语分析呈现出和英语（以及其他一些语言）不同的特点。

不过也不能过分夸大这种不同。我认为，那种以为汉语完全不同于英语，因此有必要重新建立一套分析体系的想法是没有道理的。从现有的研究看，汉语分析所使用的技术和其他语言分析所使用的技术并没有本质的不同，只是应用方式上有所区别（主要体现在词法分析方面）。而且从应用的效果看，没有证据表明，这些技术用来分析汉语比用来分析英语效果更差。

本文结合我们自己的一些工作，比较全面的介绍一下汉语词法分析和句法分析中所使用的各种技术。

1 汉语词法分析

前面说过，汉语在形式上，并没有“词”这一个单位，也就是说，汉语的语素、词、短语、甚至句子之间（词也可以直接成句，称为独词句），都没有明确的界限。

这是不是说，汉语就没有必要做词法分析，可以直接做句法分析呢？

实际并不是这样。因为如果这样做的话，会导致句法分析的搜索空间急剧膨胀，以致无法承受。实际上，根据我们的统计，未定义词在汉语中真实文本中所占的比例并不大，可见绝大部分词都是可以在词典中找到的，如果这些词都要从头开始分析，势必给句法分析带来太多的负担。

不过汉语的词法分析与英语（或其他屈折型语言）的词法分析有很大不同。就英语来说，采用确定的有限状态自动机就已经能基本解决问题，而对于汉语词法分析来说，需要更为复杂的计算工具。就问题的复杂性而言，我认为汉语的词法分析大致相当于英语的词法分析和基本短语分析之和。

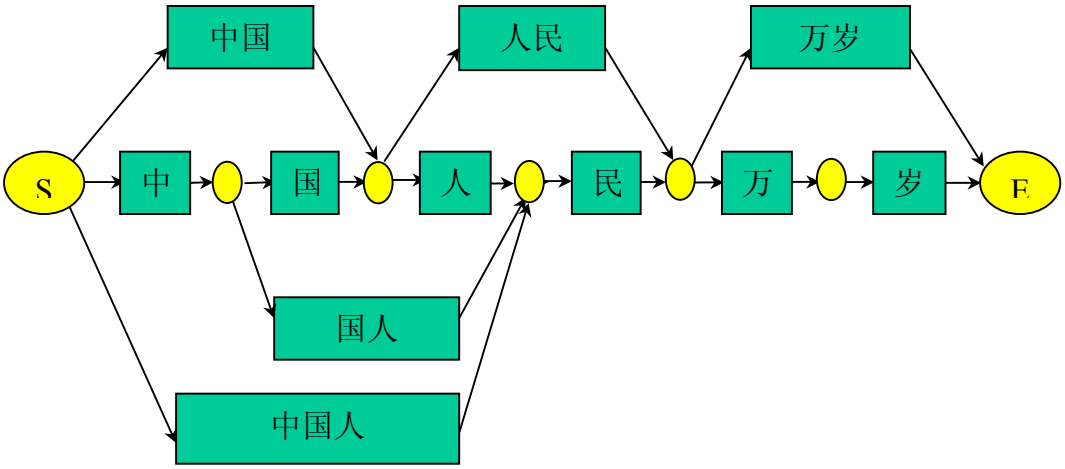
1.1 汉语词法分析的任务

汉语词法分析包括一下几个任务：

1. 查词典
2. 处理重叠词、离合词、前后缀
3. 未定义词识别
   a) 时间词、数词处理
   b) 中国人名识别
   c) 中国地名识别
   d) 译名识别
   e) 其他专名识别
4. 切分排歧
5. 词性标注

1.2 数据结构：词图（Word Graph）

对于一个汉语句子，如果把两个汉字之间的间隔作为结点，把一个汉语词作为连接两个结点的有向边，那么我们就可以得到一个无环有向图：



根据这个数据结构，我们可以把词法分析中的几种操作转化为：

1. 给词图上添加边（查词典，处理重叠词、离合词和前后缀）；
2. 寻找一条起点 S 到终点 E 的最优路径（切分排歧）；
3. 给路径上的边加上标记（词性标注）；

1.3 词典查询与重叠词、离合词和前后缀的处理

词典查询主要考虑分词词典的数据结构与查询算法的时空消耗问题。

在词典规模不大的时候，各种词典查询算法对汉语词法分析的效率整体影响并不大。不过当词典规模很大时（几十万到上百万数量级），词典查询的时空开销会变得很严重，需要详细设计一个好的词典查询算法。

（孙茂松，2000）一文比较详细的总结了汉语词法分析中使用的几种词典查询算法。（Aho&Corasick，1990）提出的算法（简称 AC 算法）实现了一种自动机，可以在线性的时间里用一组关键词去匹配一个输入字符串，（Ng&Lua，2002）一文对 AC 算法中提出的自动机（实际上就是一种词典索引的组织方式）进行了改进，可以快速实现输出汉语句子的多种切分候选结果。对词典查询算法感兴趣的同学可以去查阅这几篇文章，这里不再做详细的介绍。

汉语重叠词的重叠方式有很强的规律，处理起来并不困难。例如汉语的双字形容词的重叠现象主要有三种：AABB、ABAB、A 里 AB。遇到这种形式的词，只要还原成词语原形 AB 并查词典即可。

汉语词的前后缀不多，处理也不困难，通过简单的规则，即可这里不做介绍。

离合词的处理稍微复杂一些。现在一般的词法分析器都没有对离合词进行处理，仅仅把分开的离合词作为两个词对待，实际上这样做是不太合理的。离合词中，通常有一个语素的自由度比较差，可以通过这个语素触发，在一定的上下文范围内查找另一个语素，即可发现离合词。

1.4 不考虑未定义词的切分排歧

1.4.1 切分歧义的分类

不考虑未定义词的切分排歧问题，也就是我们一般说的切分问题。

一般把切分歧义分为两种结构类型：交集型歧义（交叉歧义）和组合型歧义（覆盖歧义）。

交集型歧义（交叉歧义）：“有意见”：我对他有意见。总统有意见他。

组合型歧义（覆盖歧义）：“马上”：我马上就来。他从马上下来。

其中交集型歧义占到了总歧义字段的 85%以上。

实际语料中出现的情况并不都这么简单，有时会出现非常复杂的歧义切分字段。例如：

公路局正在治理解放大道路面积水问题

其中“治理”“理解”“解放”“放大”“大道”“道路”“路面”“面积”“积水”都是词，考虑到这些单字也都可以成词，这就使得这个句子可能的歧义切分结果非常多。

1.4.2 切分排歧算法概述

这里我们介绍几种最主要的歧义切分算法：

1. 全切分：全切分算法可以给出一个句子所有可能的切分结果。由于全切分的结果数随着句子长度的增加呈指数增长，因此这种方法的时空开销非常大；
2. 最大匹配：从左到右或从右到左，每次取最长词，得到切分结果。分为前向最大匹配、后向最大匹配和双向最大匹配三种方法。很明显，最大匹配法无法发现组合型歧义（覆盖歧义），对于某些复杂的交集型歧义（交叉歧义）也会遗漏；
3. 最短路径法：采用动态规划方法找出词图中起点到终点的最短路径，这种方法比最大匹配法效果要好，但也存在遗漏的情况；
4. 交叉歧义检测法：（王显芳，2001-1）给出了一种交叉歧义的检测方法，可以快速给出句子中所有可能的交叉歧义切分结果，对于改进切分的效率非常有效；
5. 基于记忆的交叉歧义排除法：（孙茂松，1999）考察了一亿字的语料，发现交集型歧义字段的分布非常集中。其中在总共的 22 万多个交集型歧义字段中，高频的 4,619 个交集型歧义字段站所有歧义切分字段的 59.20%。而这些高频歧义切分字段中，又有 4,279 个字段是伪歧义字段，也就是说，实际的语料中只可能出现一种切分结果。这样，仅仅通过基于记忆的方法，保存一种伪歧义切分字段表，就可以使交集型歧义切分的正确率达到 53%，再加上那些有严重偏向性的真歧义字段，交集型歧义切分的正确率可以达到 58.58%。
6. 规则方法：使用规则排除切分标注中的歧义也是一种很常用的方法。规则的形式定义可以非常灵活，如下所示：

   @@ 的话(A+B, AB)

   CONDITION FIND(R,NEXT,X) {%X.ccat=~w} SELECT 1

   CONDITION FIND(L,NEAR,X) {%X.yx=听|相信|同意} SELECT 1

   CONDITION FIND(L,NEAR,X) {%X.yx=如果|假如|假设|要是|若|如若} SELECT 2

   OTHERWISE SELECT 1

   可以看到，通过规则可以在整个句子的范围内查找对于排歧有用的信息，非常灵活。规则方法的主要问题在于知识获取。如果单纯依靠人来写规则，无疑工作量太大，而且也很难总结得比较全面。也可以通过从语料库学习的方法来获取规则，如采用错误驱动的基于转换的学习方法。
7. n 元语法：利用大规模的语料库和成熟的 n 元语法统计模型，可以很容易将切分正确率提高到很高的正确率。（王显芳，2001-2）和（高山，2001）都说明，使用三元语法，在不考虑未定义词的情况下，就可以将切分的正确率提高到 98%以上。
8. 最大压缩方法：（Teahan et. al. 2000）提出了一种基于最大压缩的汉语分词算法。这是一种自适应的算法，其基本思想是，首先用一个标注语料库进行训练，在实际标注过程中以最大压缩比为指导来决定切分方式。这种方法的主要优点是其自适应的特定，可以切分出一些词典中没有出现的词。

上面这些方法中，前四种方法不需要人工总结规则，也不需要语料库；规则方法需要人工总结规则，比较费时费力；其他几种方法需要大规模的切分语料库为训练的基础。好在目前这种语料库已经可以得到，如（俞士汶等， 2000）。

1.4.3 n 元语法

从上面的介绍可以看到，在有大规模语料库切分语料库的情况下，采用简单的 n 元语法，就可以使切分正确率达到相当高的程度。所以我们在这里简单介绍一下 n 元语法在汉语分词中的应用。首先简单介绍一下 n 元语法的原理。

n 元语法的作用之一，是可以预测一个单词序列出现的概率。n 元语法假设一个单词出现的概率分布只与这个单词前面的 n-1 个单词有关，与更早出现的单词无关。这样，为了描述这个概率分布，我们需要使用一个 n 维数组，这个数组中每一维长度为单词的个数 m，这个数组中元素的个数为 m^n，其中元素 $a_{i_1 i_2 \ldots i_n}$ 的含义为：在单词串 $w_{i_1} w_{i_2} \ldots w_{i_{n-1}}$ 后面出现单词 w_{i_n} 的概率，也就是 $p(w_{i_n} \mid w_{i_1} w_{i_2} \ldots w_{i_{n-1}})$。

假设我们的单词表中有 50,000 个单词，如果我们使用一元语法，就是说，假设每个单词出现的频率与其他单词无关，那么所使用的参数实际上就是每个单词出现的词频，参数个数等于 50,000。如果我们使用二元语法，就是说，假设每个单词出现的频率只与上一个单词相关，那么所使用的参数就是一个单词后面出现另外一个单词的转移概率，参数个数为 50,000×50,000。如果采用三元语法，参数的个数将是 50,000 的三次方。实际上，由于很多的单词序列在实际的语料库中并不会出现，所以实际上有效的参数数量会少的多。不过，如果这些在训练语料中没有出现的单词序列出现在测试文本中，会导致该文本的预测概率为 0。为了避免这种情况，我们就要采用某种策略将这些为概率为 0 的单词序列赋予一个很小的猜测值，这种策略叫做数据平滑。由于数据稀疏问题的大量存在，数据平滑在任何一种统计模型中都是必须采用的。数据平滑有很多种技术，这里不再一一介绍。

n 元语法是一种非常成熟的语言模型，而且在自然语言处理中被证明是非常有效的。Internet 上有现成的 n 元语法的源代码可以下载（如 The CMU-Cambridge Statistical Language Modeling toolkit），而且即使自己编写，也并不太复杂。

我们的实验表明，仅仅使用一元语法（也就是仅使用词频信息），切分的正确率就可以达到 92%以上。

1.4.4 基于 n 元语法的切分排歧方法

前面我们说了，所谓切分排歧过程，可以看作从词图中选择一条最优路径的过程。利用 n 元语法，我们可以对任何一条路径进行概率评分：

$$p(w_1 w_2 \ldots w_l) = p(w_1 \ldots w_{n-1}) \prod_{i=n}^{l} p(w_i \mid w_{i-1} \ldots w_{i-n+1})$$

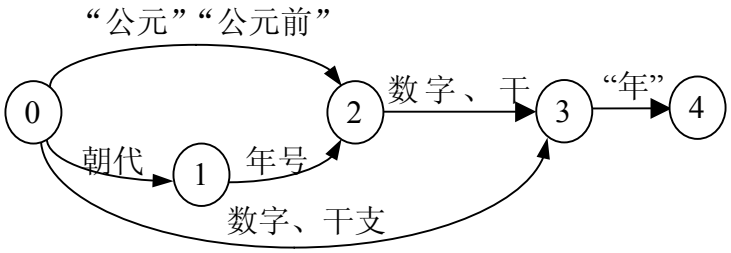
算法可采用动态规划方法实现。算法的时间开销与句子长度成正比。

1.5 未定义词识别

汉语中，由于词与词之间没有形式上的边界，而且绝大多数的汉字都可以独立成词，因此未定义词的识别问题非常严重。

1.5.1 时间词、数词处理

由于时间词、数词的组成规律性较强，识别起来比较简单。一般采用一个简单的确定性有限状态自动机即可。例如采用下面的有限状态自动机可以识别年份：



1.5.2 中国人名、中国地名、译名和其他专名的识别

中国人名是未定义词中最常见，也是比较容易识别的一类，因为中国人名的姓名用字都有比较强的规律。中国地名的规律性稍差一些。译名的用字非常集中，不过短译名比较容易和其他类型的未定义词混淆。其他专名主要包括组织机构名、企业商标字号等等，这些专名的用字分布也有一定规律，但规律性不是很强，目前识别准确率都不高。

这些类型的未定义词识别，仅仅使用规则方法很难达到好的效果，一般都要进入统计方法。我们这里仅以中国人名为例说明这些类型未定义词的识别方法。有关中国人名识别的研究已经很多：（李建华，2000）、（孙茂松， 1995）、（张俊盛，1992）、（宋柔，1993）等。所使用的方法包括规则加统计的方法和纯统计的方法。其中（孙茂松，1995）一文对中国人名用字的分布有比较详细的统计结果。

这里我们主要介绍我们自己使用的一种采用隐马尔可夫模型（HMM）的中国人名识别方法（Zhang et al.，2002，张华平等，2002），我们称之为“基于角色标注的中国人名自动识别方法”。在我们的词法分析系统中，这种方法达到了很好的效果。

1) 人名识别的输入输出

人名识别的输入是一个已经经过粗切分的句子，只是其中的未定义词都没有被识别出来，如：

馆 内 陈列 周 恩 来 和 邓 颖 超生 前 使用 过 的 物品

注意，这个粗切分结果可能是有错误的，如这个句子就把超生合并成了一个词。

在通过中国人名识别程序后，应该把句子中“周恩来”和“邓颖超”这两个人名识别出来。

2) 隐马模型

关于隐马模型，我们这里不再做详细的介绍，只给出一个直观的解释。感兴趣的同学可以取参考（Rabiner，1989）和（翁富良，1998）。隐马模型作为一种简单而有效的数学工具，在自然语言处理、语音识别、生物信息学很多领域得到了广泛的应用。后面我们将要介绍的词性标注，也要使用隐马模型这个工具。隐马模型目前已经发展得非常成熟，在网上也能找到完整的带源代码的软件工具（如 HMM Toolkit）。

我们这里以词性标注问题为例，对隐马模型给出一个直观的解释。

隐马模型要解决的问题，就是对于一个单词串（句子），要给这个单词串中的每一个单词做一个标记（例如单词的词性）。并假设从统计规律上说，每一个词性的概率分布只与上一个词的词性有关（也就是一个词性的二元语法），而每一个单词的概率分布只与其词性相关。

如果我们已经有了一个已经标记了词性的语料库，那么我们就可以通过统计得到以下两个矩阵（实际上还有一个初始词性概率分布矩阵）：

词性到词性的转移概率矩阵：$A = \{a_{ij}\}$，$a_{ij} = p(X_{t+1} = q_j \mid X_t = q_i)$

词性到单词的输出概率矩阵：$B = \{b_{ik}\}$，$b_{ik} = p(O_t = v_k \mid X_t = q_i)$

这里 $q_1,\ldots,q_n$ 是词性集合，$v_1,\ldots,v_m$ 是单词的集合。

对于词性标注问题而言，转移概率矩阵中的一个元素 a_{ij} 含义的就是上一个词的词性为 q_i 时，下一个词的词性为 q_j 的概率；输出概率矩阵中的一个元素 b_{ik} 的含义就是对于一个词性 q_i 来说，对应的词语为 v_k 的概率。

在有了这两个矩阵之后，对于任何一个给定的观察值序列（单词串），我们总可以通过一个 Viterbi 算法，很快得到一个可能性最大的状态值序列（词性串）。算法的复杂度与观察值序列的长度（句子中的单词个数）成正比。对于 Viterbi 算法，我们这里不再详细描述。

3) 中国人名识别中的角色定义

通过上面的介绍我们可以看到，隐马模型处理的问题就是一个标注问题，也就是给一个单词串中的每一个单词做一个标注的问题。

对于词性标注问题而言，这个标注就是词性。

对于中国人名识别问题，我们要标注的是这个单词在人名识别中充当的角色。

我们定义的角色如下表所示：

编码	意义	例子
B	姓氏	张华平先生
C	双名的首字	张华平先生
D	双名的末字	张华平先生
E	单名	张浩说：“我是一个好人”
F	前缀	老刘、小李
G	后缀	王总、刘老、肖氏、吴妈、叶帅
K	人名的上文	又来到于洪洋的家。
L	人名的下文	新华社记者黄文摄

M	两个中国人名之间的成分	编剧邵钧林和稽道青说
U	人名的上文和姓成词	这里有关天培的壮烈
V	人名的末字和下文成词	龚学平等领导, 邓颖超生前
X	姓与双名的首字成词	王国维、
Y	姓与单名成词	高峰、汪洋
Z	双名本身成词	张朝阳
A	以上之外其他的角色	

4) 语料库的训练

隐马模型的训练需要标记好的语料库。由于这里的标记是我们自己定义的，显然没有现成的语料库可用。不过这个问题并不难解决。由于我们已经有了《人民日报》的切分标记语料库，这个语料库中所有词都标注了词性，其中人名也有专门的标记（nr），我们设计了一个半自动转换程序，只需要很少的人工干预，就可以将《人民日报》语料库的词性标记转换为我们设定的中国人名角色标记。

5) 人名的识别

通过语料库的训练，我们可以得到中国人名识别的隐马模型（三个个概率矩阵）。这样，对于输入的任何一个粗切分结果，我们都可以进行中国人名的角色标注。

为了解决人名与其上下文组合成词的问题，在人名识别之前，我们要对角色 U（人名的上文和姓成词）和 V（人名的末字和下文成词）进行分裂处理。相应地分裂为 KB、DL 或者 EL。然后，我们在得到的角色序列中寻找一些特定的模式：{BBCD, BBE, BBZ, BCD, BEE, BE, BG, BXD, BZ, CD, EE, FB, Y, XD}，凡是匹配成功的序列，我们都认为是一个人名。

以前面的句子为例，我们得到的标注结果（分裂后）就是：

馆/A 内/A 陈列/K 周/B 恩/C 来/D 和/M 邓/B 颖/C 超/D 生/L 前/A

使用/A 过/A 的/A 物品/A

通过模式匹配，得到两个成功的模式（都是 BCD），对应的人名就是“周恩来”和“邓颖超”。

6) 实验结果

我们利用两个月的《人民日报》语料进行初步的测试，结果如下表所示：

类别	封闭测试语料 1	封闭测试语料 2	开放测试语料
来源 均为《人民日报》	98 年 1 月	98 年 2 月 1 日-20 日	98 年 2 月 20 日-28 日
语料库大小（字节）	8,621K	6,185K	2,605K
实际人名数	13360	7224	2967
识别出的人名数	17505	10929	4259
正确识别的人名数	13079	7106	2739
准确率	74.72%	65.02	64.32%
召回率	97.90%	98.37%	92.32%
F 值	84.75%	78.29%	75.81%

需要说明的是，我们这里的测试是在完全真实的语料环境下进行的。如果仅

对 12,507 个含人名的句子重新进行识别测试实验，无论是封闭测试还是开放测试，准确率、召回率均超过 92%以上。因为这种方法下，排除了原来没有人名的句子被识别出人名的错误情况。另外，由于我们的实验规模比已有的一些类似工作的规模都要大得多，其实验结果的可信度也更高。

我们的实验结果中，召回率比较高，而准确率较低，这也对我们的整个词法分析是有利的。由于人名识别只是整个词法分析过程的一个阶段，错误识别出的人名在后续的过程中还有可能被排除掉，但被忽略掉的人名在后续过程中却不可能被重新发现。众所周知，召回率和准确率是互相矛盾的。高召回率、低识别率对于整个词法分析过程是有利的。

1.6 考虑未定义词的切分排歧

前面介绍的切分排歧方法，都没有考虑到未定义词问题。如果把未定义词的因素考虑进来，切分排歧算法应该如何调整呢？

前面我们介绍了，采用 n 元语法，我们可以从一个词图中选取一条最优路径，作为最好的分词结果。其中，n 元语法的参数可以事先从语料库中训练得到。在未定义词词识别以后，词图中加入了新识别出来的未定义词。不过，由于未定义词可能是语料库没有出现的，无法事先得到未定义词的 n 元语法参数。

我们采用的做法是：把每一种类型的未定义词（如中国人名、中国地名等）作为同一个词进行 n 元语法的参数估计。在实际计算词图中的一条路径的概率评分时，除了要利用 n 元语法的概率评分之外，还要乘上句子中每一个未定义词在该类未定义词中出现的概率。也就是评分函数修改如下：

$$p(w_1 w_2 ... w_l) = p(w_1 ... w_{n-1}) \prod_{i=n}^{l} p(w_i \mid w_{i-1} ... w_{i-n+1}) \prod_{t=1}^{T} \prod_{s=1}^{l_t} p(w_{ts} \mid Type(t))$$

其中，*Type(t)*，*t=1...T*，表示 *T* 种类型的未定义词，$w_{t1}...w_{tl_t}$ 为句子中识别出来的类型为 *Type(t)*的 l_t 个未定义词。其中 $p(w_{ts} \mid Type(t))$ 可以由前面的未定义词识别算法得到。

1.7 词性标注

词性标注也是研究得比较充分的一个课题。

总体上说，汉语的词性标准和英语的词性标注在方法上没有明显的不同。

在有大规模标注语料库的情况下，很多方法（特别是统计方法）都可用于解决词性标注问题，而且结果通常也都很好。我们这里不一一列举，只给出常用的两种方法：

1. 隐马模型（HMM）：前面已经介绍了；
2. 错误驱动的基于转换的规则方法（TBL）：这是一种从语料库中学习规则的方法，由于篇幅所限，这里不做详细介绍；

我们使用《人民日报》标注语料库，采用隐马模型，也得到了很好的结果。

1.8 词法分析的流程

大家可以看到，词法分析是一个很复杂的过程，其中有很多子任务，而这些子任务又是互相交织在一起的。作为一个完整的词法分析程序，应该如何组织这个过程？子任务之间又应该如何衔接呢？

在具体实现上，各个子任务的顺序并没有明确的规定，例如，前后缀的处理可以在查词典阶段进行，也可以在未定义词识别阶段进行；人名识别可以在查词典之前进行（基于字的模型），也可以在查词典以后进行（基于词的模型），切分和标注可以分别进行，也可以同时进行（高山，2001）。

不过，根据我们的经验，我们在这里提出几条原则，应该说对整个词法分析流程的设计有一定的指导意义：

1. 采用一致的数据结构（如词图），有利于各个阶段之间的衔接。这个数据结构应该有一定的冗余表达能力，能够同时表示多种切分标注的结果；
2. 每一个阶段最好能输出几个候选结果，有些歧义现象在某一个阶段无法排除，可能在下一阶段却很容易解决，提供多个候选结果，有利于总体上减少错误率；
3. 如果采用统计模型，应该尽量在各个统计模型之间建立一定的联系，也就是前面得到的概率评分值能在后面的阶段中有效的利用起来，最理想的是建立统一的概率模型，可以得到总体最优的结果。

在我们的系统中，我们采用的词法分析流程如下所示：

1. 查词典，重叠词处理；
2. 数词、时间词、前后缀处理；
3. 粗切分（采用基于词一元语法，保留多个结果）；
4. 未定义词识别（采用基于角色标注的隐马模型，识别中国人名、中国地名、译名、其他专名）；
5. 细切分（采用基于词的二元语法，利用 PCFG 计算未定义词概率，输出多个结果）；
6. 词性标注（采用隐马模型，输出一个或多个结果）。

我们开发的系统 ICTCLAS 通过大规模的开放测试，实际切分正确率在 97%以上，标注正确率约为 95%。该系统的源代码可以在自然语言处理开放平台(http://www.nlp.org.cn)下载。

2 汉语句法分析

词法分析的作用是从词典中划分出词，而句法分析的作用是了解这些词之间的关系。所以，句法分析的输入是一个词串（可能含词性等属性），输出是句子的句法结构。

就句法分析所面临的问题而言，汉语和英语及其他语言，都没有太大的不同。二者所采用的技术也都大体一致。

2.1 形式语法体系

句法分析一般都依赖于某种语法体系。语法体系的形式丰富多彩，各种语法

形式都有各自的特点。这里简要介绍几种典型的语法形式，主要目的是让读者对语法形式的多样性有一个直观的感受。

不同的语法体系产生的句法结构形式不尽相同。最常见也最直观的句法结构形式是句法树。其他主要的形式有依存关系树（依存语法、范畴语法）、有向图（链语法）、特征结构（HPSG、LFG）等等。

2.1.1 乔姆斯基层次体系

所谓乔姆斯基层次体系（Chomsky Hierarchy），指的是乔姆斯基定义的四种形式语法，这四种语法，这四种语法所产生的语言依据包含关系构成了严格的层次体系。

乔姆斯基层次体系第一次严格地描述了形式语法、语言和自动机之间的关系，在数学、计算机科学和语言学建立起了一道沟通的桥梁。

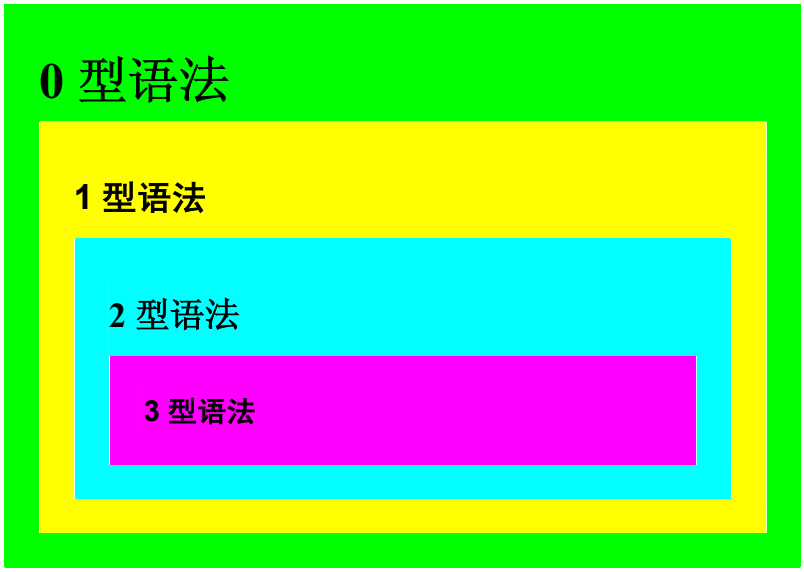
在乔姆斯基的语法层次体系中，一共定义了四种层次的形式语法，这四种语法可统称为短语结构语法（PSG）。一个 PSG 形式定义如下：

一个 PSG 是一个四元组：{ V, N, S, P }，其中 V 是终结符的集合（字母表），N 是非终结符的集合，S∈N 是开始符号，P 是重写产生式规则集。

乔姆斯基语法层次体系中的四种语法形式具体说明如下：

层级	语法	识别自动机	产生式规则形式	例子
0 型	不受限短语结构语法	图灵机	α -> β	
1 型	上下文敏感语法	线性有界自动机	$\alpha A\beta$ -> $\alpha\gamma\beta$	$a^nb^nc^n$
2 型	上下文无关语法	下推自动机	A -> γ	a^nb^n
3 型	正规语法	有限状态机	A -> aB A -> a	a^*

一个 PSG 所接受的语言就是由开始符号 S 通过 P 中的规则所可以导出的所有终结符串的集合。乔姆斯基四种形式语法所导出的语言具有以下关系：



正规语法的语法形式最严格，生成的语言最简单，分析起来也最容易（时间复杂度是线性的），可以用有限状态自动机进行分析。有限状态自动机现在广泛

应用于各种语言的词法分析中。由于有限状态自动机的高效性，也有人使用它来进行句法分析（见后面部分分析的介绍），甚至有人用来做机器翻译(Alshawi et al.，2000)。

上下文无关语法虽然不足以刻划自然语言的复杂性，但由于其形式简单，分析效率高（多项式时间复杂度），实际上是句法分析中使用最广泛的一种语言形式。我们后面将要介绍的句法分析算法大多也都是基于上下文无关语法的。

上下文敏感语法分析的时间复杂度是非多项式的（NP 问题），而 0 型文法的分析甚至不是一个可判定性问题（实际上是一个半可判定问题），所以这两种语法形式在实际中都无法得到应用。

2.1.2 乔姆斯基的形式句法理论

乔姆斯基的形式语法理论不仅是现代计算机科学的基础之一，也为语言学的研究打开了一个暂新的局面，对自然科学和社会科学的很多领域都产生了深远的影响，被称为“乔姆斯基革命”，在科学史上具有里程碑式的重要地位。

乔姆斯基的形式语法理论是一个不断演变、不断发展的过程。在 1957 年，乔姆斯基提出了“转换生成语法理论（TG）”，1970 年代，发展成为“标准理论”，在 1981 年，乔姆斯基又提出了“管辖－约束理论（GB）”，1992 年，提出了“最简方案（MP）”。

乔姆斯基的形式语法理论有一个核心思想，就是“普遍语法”的思想。他认为人有先天的语言习得机制，生来就具有一种普遍语法知识，这是人类独有的生理现象。人类各种语言之间共性（原则）是主要的，语言之间的个性（参数）是次要的。因此乔姆斯基后期的语言学理论（GB 以后）又称为“原则＋参数”的语言学理论。

乔姆斯基早期的转换生成语法还比较简单，后来乔姆斯基语法理论越来越复杂，使得形式化的工作变得非常困难。所以现在计算语言学领域的研究中，已经很少有人采用乔姆斯基的形式语法体系。不过乔姆斯基的形式句法理论在语言学界还是很有生命力的，因为它确实可以解释很多其他理论很难解释的语言现象。

2.1.3 中心词驱动的短语结构语法（HPSG）和词汇功能语法（LFG）

HPSG 和 LFG 属于非乔姆斯基阵营的语法理论中比较有生命力的两种。

他们与乔姆斯基语法理论的本质差别在于没有转换规则（乔姆斯基后期的理论中又称为 α －移动），没有浅层结构和深层结构的区别。

从计算机实现的角度看，这两种理论都采用了特征结构这种形式来表达复杂的语言学知识并采用合一算法进行规则的推导。与乔姆斯基的语法理论不同，这两种语法理论都又很好的可实现性。因此这两种理论的发展一直和计算机的结合非常紧密。

有关这两种语法的详细资料，可到互联网上查询相应网站。

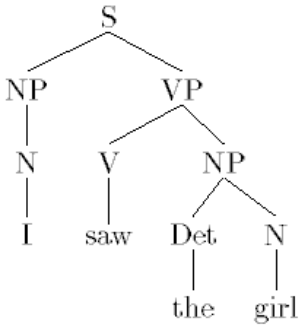
LFG：Stanford：http://www-lfg.stanford.edu/lfg/

Essex： http://clwww.essex.ac.uk/LFG/

HPSG：http://hpsg.stanford.edu/

下面仅通过几个图示简单介绍一下 LFG，使读者对 LFG 有一个直观的映像。

在 LFG 中，一个句子的结构除了用一棵句法树（c-structure），还用一个特征结构（f-structure）来刻划这个句子的各种句法特征，如下图所示：



$$\begin{bmatrix} \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'pro'} \\ \text{PERS} & 1 \\ \text{NUM} & \text{SG} \end{bmatrix} \\ \text{TENSE} & \text{PAST} \\ \text{PRED} & \text{'see}\langle(\uparrow \text{SUBJ}), (\uparrow \text{OBJ})\rangle\text{'} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'girl'} \\ \text{DEF} & + \\ \text{PERS} & 3 \\ \text{NUM} & \text{SG} \end{bmatrix} \end{bmatrix}$$

相应的，LFG 的规则（包括词典中的词条）除了通常的短语结构规则形式外，还附带一些合一表达式，如下图所示（↑其中表示父结点的特征结构，↓表示本结点的特征结构）：

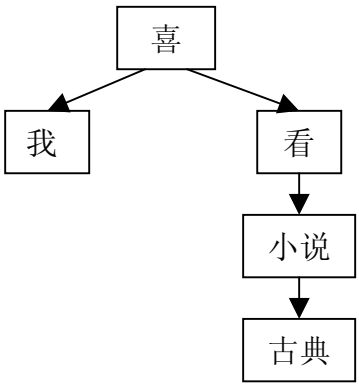
规则：S →	NP	VP
	↑SUBJ =↓	↑ = ↓

词条：*Gives* V	(↑PRED) = 'give <Agent, Theme, Recipient>'
	(↑TENSE) = present
	(↑SUBJ NUMBER) = sing
	(↑SUBJ PERSON) = 3

2.1.4 依存语法

依存语法也是一种使用非常广泛的语法形式。

与短语结构语法（PSG）的最大不同在于，依存语法的句法结构表示形式不是一棵句法层次结构的句法树，而是一棵依存树：依存树上的所有结点都是句子中的词，没有非终结符结点。例如句子“我喜欢看古典小说”的依存结构如下图所示：



可以看到，在依存关系树中，丢失了句子中词与词之间的顺序关系。

应该说，依存语法并不是一种严格定义的语法形式。依存语法没有明确定义的规则形式。也没有明确规定依存关系是否要加上标记。实际的应用系统中，一般都会给依存关系加上句法或语义的标记。

1970 年，美国计算语言学家 J. 罗宾孙(J. Robinson)提出了依存语法的 4 条公

理：

1. 一个句子只有一个成分是独立的；
2. 句子中的其它成分直接从属于某一成分；
3. 任何一个成分都不能从属于两个或两个以上的成分；
4. 如果成分 A 直接从属于成分 B，而成分 C 在句子中位于 A 和 B 之间，那么，成分 C 或者从属于 A，或者从属于 B，或者从属于 A 和 B 之间的某一成分。

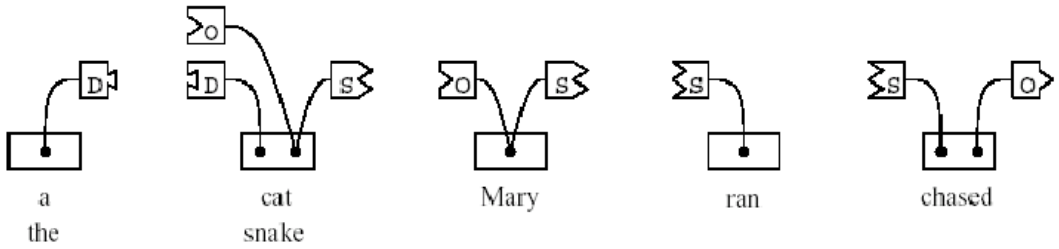
这四条公理比较准确界定了一个依存树所要满足的条件，得到了依存语法研究者的普遍接受。

2.1.5 链语法

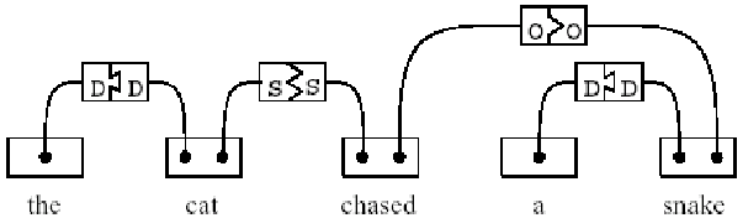
链语法由美国CMU计算机学院的Daniel Sleator和美国Columbia Uiversity（的Davy Temperley共同提出，最早的文章是1991年的一个技术报告，题目是“Parsing English with a Link Grammar”。

链语法的网址是：http://www.link.cs.cmu.edu/link。

链语法词典中的词条如下图所示：



上面的一些词组成的一个句子通过句法分析得到下面的结构：



链语法的一个显著特点是分析的结果不是一棵句法树，而是一个有向图。

链语法的另一个特点是没有句法规则，只有几条简单的原则，用于规定句法成分之间互相结合的方式。链语法的语法知识都存放在词典中。

链语法的网站上提供了链语法分析器的完整源代码。

2.1.6 范畴语法

范畴语法语法的特点在于，把句法分析的过程变成了一种类似分数乘法中进行的“约分”运算。

举一个简单的例子：我喜欢红苹果。

在词典中，句子中的几个词分别表示为：

我：N
喜欢：N/S\N
红：N/N
苹果：N

句法分析的过程表现为：

红＋苹果：N/N + N => N
喜欢＋红苹果：N\S/N + N => N\S
我＋喜欢红苹果：N+N\S => S

和链语法一样，在范畴语法中，也没有规则，只有几条简单的原则，规定范畴之间如何进行“约分”，所有的语法信息都表现在词典中。

范畴语法在现在的形式语义学理论中有很重要的作用。

范畴语法的网站是：http://www.cs.man.ac.uk/ai/CG/。

2.2 句法分析算法

句法分析的过程就是将小的语法成分组合成大的语法成分的过程。虽然各种语法的形式相差很大，不过在句法分析的过程中采用的分析算法都是类似的（也有少数语法有自己特有的句法分析算法）。

2.2.1 常见的分析算法

常见的句法分析算法包括：

1. 自顶向下分析算法；
2. 自底向上分析算法；
3. 左角分析算法；
4. CYK 算法；
5. Marcus 确定性分析算法；
6. Earley 算法；
7. Tomita 算法（GLR 算法）；
8. Chart 算法；

等等。

这些算法都有各自的优缺点和适用的场合，由于篇幅关系，我们难以一一介绍。

目前应用得最为广泛的句法分析算法是 Tomita 算法和 Chart 算法。

Tomita 算法是传统的 LR 分析算法的一种扩展，所有又被称为 Generalized LR（GLR）算法。和 LR 算法一样，GLR 算法也是一种移进－规约（Shift-Reduce）算法。GLR 算法对传统 LR 算法的改进主要体现在：

1) GLR分析表允许有多重入口（即一个格子里有多个动作），这样就克服了传统LR算法无法处理歧义结构的缺点；
2) 将线性分析栈改进为图分析栈处理分析动作的歧义（分叉）；
3) 采用共享子树结构来表示局部分析结果，节省空间开销
4) 通过节点合并，压缩局部歧义。

对于 Tomita 算法，我们这里不做详细的介绍。我们主要介绍的是 Chart 分析算法。实际上，Chart 分析算法是非常灵活的，通过修改 Chart 算法中的分析策略，很容易模拟很多种形式的其他算法，例如自顶向下的分析算法、自底向上的

分析算法和左角分析算法等等。这也是 Chart 分析算法得到广泛应用的原因之一。

2.2.2 Chart 算法

1) 一个简单的文法

算法的介绍，最直观的做法莫过于通过一个例子来说明。我们这里也不例外。

考虑一个句子[1]：我 是 县长 派 来 的。

词典中的词条有：

(1) R → 我

(2) N → 县长

(3) V → 是 | 派 | 来

所使用的规则为：

(1) S → NP VP

(2) NP → R

(3) NP → N

(4) NP → Sϕ 的

(5) VP → V NP

(6) Sϕ → NP VPϕ

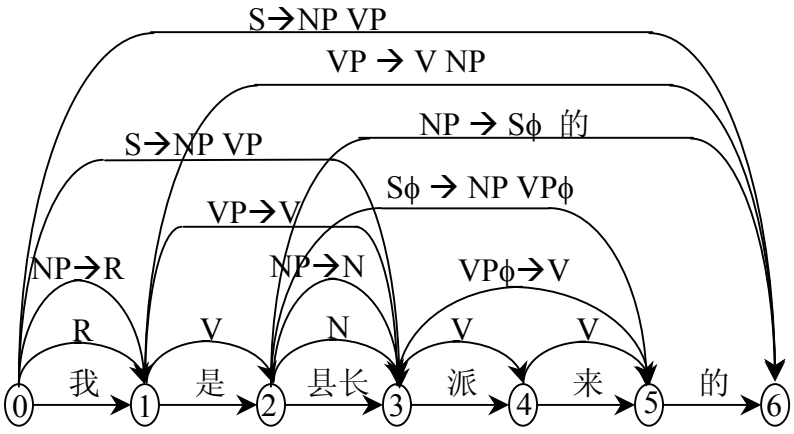
(7) VPϕ → V V

其中 Sϕ、VPϕ分别表示带空位的 S 和 VP，这里大家可以不必管它的含义，只要把 Sϕ和 VPϕ分别看成两个独立的短语类型即可。

2) Chart 数据结构

Chart（有人译为线图）是 Chart 算法中最重要的数据结构。

与前面介绍的词图表示法有点类似，线图是把词与词之间的间隔作为结点，把词和短语当作连接结点的边。于是这个句子可以用词图表示为：



这个图上，我们不仅标出了每条边的标记，还标出了产生该边的规则。

注意：“我是县长”和“我是县长派来的”都是句子。

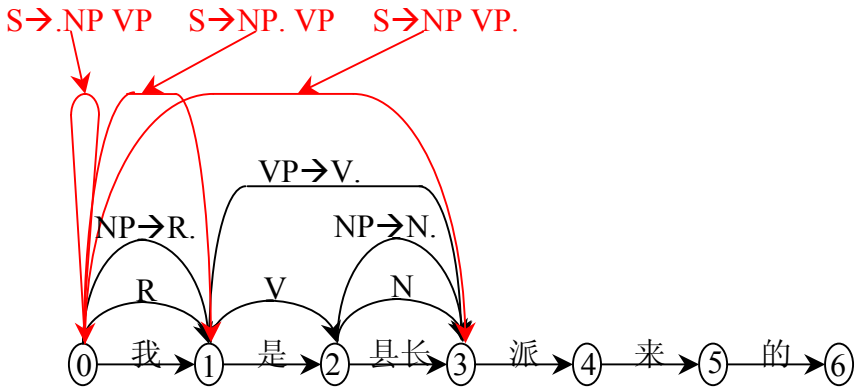
[1] 这里借用了白硕 2001(计算语言学教程讲义)中的例子，特此向白硕研究员表示感谢。

3) 活跃边与非活跃边

我们注意到，“我是县长”和“我是县长派来的”都是由规则 S → NP VP 生成的，而且其中“NP”都是对应同一个结点（“我”）。也就是说，这两次规则使用的过程中，有一个冗余的操作：将规则右部的第一个结点 NP 与同一个结点（“我”）进行匹配。如果规则很多，Chart 的结构很复杂，这种冗余是很严重的。那么，我们能不能消除这种冗余操作呢？答案是可以。在 Chart 算法中，将边分为两种，一种叫做非活跃边，就是上图中我们已经见过的这种边。另一种叫做活跃边，用于记录一条规则部分被匹配的情形。于是，规则 S → NP VP 生成结点“我是县长”的匹配过程可以记录为两条活跃边和一条非活跃边：

记录方式	边状态	匹配程度	起点	终点	对应词串
<0,0, S→.NP VP>	活跃	S → .NP VP	0	0	
<0,1, S→NP. VP>	活跃	S → NP. VP	0	1	我
<0,3, S→NP VP.>	非活跃	S → NP VP.	0	3	我是县长

其中“匹配程度”用规则中加入句点来表示，其中句点的位置表示规则已经匹配成功的位置（从左边开始）。用 Chart 表示如下：



4) 日程表（Agenda）

在 Chart 算法中，还有一个重要的数据结构，称为“日程表（Agenda）”。

Chart 分析的过程就是一个不断产生新的边的过程。但是每一条新产生的边并不能立即加入到 Chart 中，而是要放到日程表（Agenda）中。

日程表（Agenda）实际上是一个边的集合，用于存放已经产生，但是还没有加入到 Chart 中的边。日程表（Agenda）中边的排序和存取方式，是 Chart 算法执行策略的一个重要方面（后面将要介绍）。

5) Chart 算法的基本流程

Chart 算法就是一个由日程表驱动的不断循环的过程：

(1) 按照**初始化策略**初始化日程表（Agenda）；
(2) 如果日程表（Agenda）为空，那么分析失败；
(3) 每次按照**日程表组织策略**从日程表（Agenda）中取出一条边；
(4) 如果取出的边是一条非活跃边，而且覆盖整个句子，那么返回成功；
(5) 将取出的边加入到 Chart 中，执行**基本策略**和**规则调用策略**，将产生

的新边又加入到日程表（Agenda）中；

(6) 返回第(2)步。

这个算法流程当中，各项基本策略都是可以调整的，通过调整这些策略，可以得到不同的分析算法。下面我们主要介绍如果通过调整这些策略来改变分析算法。

6) 初始化策略

Chart 分析算法开始执行以前，要先将日程表（Agenda）初始化。对于自底向上和自顶向下的分析算法，要采用不同的初始化策略：

自底向上分析的规则调用策略：

(2) 将所有单词（含词性）边加入到日程表（Agenda）中。

自顶向下分析的规则调用策略：

(1) 将所有单词（含词性）边加入到日程表（Agenda）中；

(2) 对于所有形式为：S→W 的规则，产生一条形式为<0, 0, S→.W>的边，并加入到日程表（Agenda）中；

7) 基本策略

在 Chart 算法中，边是逐条被加入到 Chart 中的。每一条边在被加入到 Chart 中时，都要执行以下**基本策略**：

(1) 如果新加入一条活跃边形式为：<i, j, A→W_1. B W_2>
那么对于 Chart 中所有形式为：<j, k, B→W_3>的非活跃边，生成一条形式为<i, j, A→W_1 B. W_2>的新边，并加入到日程表（Agenda）中；

(2) 如果新加入一条活跃边形式为：<j, k, B→W_3>
那么对于 Chart 中所有形式为：<i, j, A→W_1. B W_2>的活跃边，生成一条形式为<i, j, A→W_1 B. W_2>的新边，并加入到日程表（Agenda）。

上面 A、B 为非终结符，W_1、W_2、W_3 为终结符和非终结符组成的串，其中 W_1、W_2 允许为空，W_3 不允许为空。

8) 规则调用策略

自底向上的分析和自顶向下的分析中，要使用不同的规则调用策略：

自底向上分析的规则调用策略：

如果要加入一条形式为<i, j, C→W_1.>的边到 Chart 中，
那么对于所有形式为 B→C W2 的规则，产生一条形式为<i, i, B→.C W2>的边加入到日程表（Agenda）中。

自顶向下分析的规则调用策略：

如果要加入一条形式为<i, j, C→W_1.B W_2>的边到 Chart 中，
那么对于所有形式为 B→W 的规则，产生一条形式为<j, j, B→.W>的边，并加入到日程表（Agenda）中。

9) 日程表组织策略

通过日程表组织的不同策略，可以分别实现深度优先和广度优先等句法分析策略：

深度优先的日程表组织策略：

将日程表按照堆栈的形式，每次从日程表中取出最后加入的结点；

广度优先的日程表组织策略：

将日程表按照队列的形式，每次从日程表中取出最早加入的结点；

10) 细节处理

前面的讨论中忽略了两个细节，在实现一个系统时应该考虑到：

(1) 考虑到可能通过多种途径生成一条完全相同的边，所以每次从日程表（Agenda）中取出一条新边加入 Chart 时，要先检查一下 Chart 中是否已经有相同的边，如果有，那么删除这条边，直接进入下一个循环；

(2) 为了生成最后的句法结构树，每一条边中还应该记录其的子句法成分所对应的边。

11) 例子

下面我们按照自底向上的初始化策略和规则调用策略以及深度优先的日程表组织策略，给出上述例句（“我是县长派来的”）的分析过程（略）。

12) 讨论

通过上面的介绍，大家可以看到，Chart 分析算法是一种非常灵活的分析算法，通过修改分析过程中的一些具体策略，Chart 分析算法可以模拟很多种其他句法分析算法。

如果你有兴趣，完全可以自己尝试修改这些策略，以实现新的句法分析算法。

另外，（白硕&张浩，2002）中，把 Tomita 算法中“向前看（look ahead）”的思想结合到 Chart 分析算法中，提出了一种“角色反演算法”，可以减少 Chart 分析算法中垃圾边的数量而又不影响最后的分析结果，提高分析的效率。

2.2.3 基于统计的句法分析算法

随着统计方法在 NLP 中的复兴，各种统计的句法分析算法也开始得到广泛的研究，并取得了很大的进展。

纯粹基于规则的句法分析算法有以下缺点：

1. 歧义问题：如何总众多的歧义结构中选择合理的结构？规则方法无法给出满意的答复；
2. 鲁棒性问题：对于不符合语法的句子，规则方法无法给出满意的猜测；
3. 规则冲突问题：规则增加时规则之间的冲突变得非常严重，规则调试非常困难，后面的规则往往会抵消前面规则的作用，使得系统总体效果无法改善。

由于基于统计的概率句法分析算法都需要句法树库作为训练数据（无指导的统计句法分析算法也有人尝试过，效果非常糟糕），这使得句法树库的建设成为了实现统计句法分析算法的前提。好在现在已经开始有了一些这种语料库，如 LDC 提供的英语和汉语句法树库。其中汉语的句法树库规模较小，含 10 万汉语词语，约 4 千个汉语句子，主要的数据来源是新华社新闻稿。

下面我们我们先介绍统计句法分析方法的两种类型的模型，然后介绍几种典型的统计的句法分析算法：

1) 分析模型与语言模型

任何统计模型，最基本的都是一个归一性假设。

统计句法分析的两类模型的区别就在于归一性假设上。

在分析模型中，假设对于任何一个句子，其所有的可能的分析树的概率之和为 1：

$$P(t \mid s,G), where \sum_{t} P(t \mid s,G) = 1$$

$$\hat{t} = \arg\max_{t} P(t \mid s,G)$$

其中，G 表示该分析模型，s 表示一个句子，t 表示该句子的一种可能的分析结果（句法树）。

而在语言模型中，假设从一种语言中推导出的所有句子结构（句法树）的概率为 1，而一个句子的概率为其所有可能的句子结构（句法树）的概率之和：

$$\sum_{\{t:yield(t)\in L\}} P(t) = 1$$

$$P(s) = \sum_{t} P(s,t) = \sum_{\{t:yield(t)=s\}} P(t)$$

$$\hat{t} = \arg\max_{t} P(t \mid s) = \arg\max_{t} \frac{P(t,s)}{P(s)} = \arg\max_{t} P(t,s)$$

初看上去，好像分析模型比较符合我们的推理过程。不过，在实际的研究工作中，语言模型应用更多。因为实现的时候，分析模型需要正例和反例同时进行训练，这在处理上比较困难。而语言模型只需要即可进行训练。从已有的研究工作看，语言模型的效果也更好一些。

2) 统计句法分析的评价标准

在统计句法分析研究中，一般使用以下几个参数作为评价标准：

标记正确率（Labeled Precision）

$$LP = \frac{number\ of\ correct\ constituents\ in\ proposed\ parse}{number\ of\ constituents\ in\ proposed\ parse}$$

标记召回率（Labeled Recall）

$$LR = \frac{number\ of\ correct\ constituents\ in\ proposed\ parse}{number\ of\ constituents\ in\ treebank\ parse}$$

交叉括号数（Crossing Brackets）

$$CB = number\ of\ constituents\ which\ violate\ constituent\ boundaries\ with\ a\ constituent\ in\ the\ treebank\ parse$$

所谓交叉括号数，就是与标准语料库中发生边界冲突的结点数目，类似于汉

语词切分中的交叉歧义字段数。

3) 概率上下文无关语法

概率上下文无关语法的基本思想就是给传统的上下文无关语法加上概率信息。其基本思想非常简单：所有由同一个句法标记导出的规则的概率之和为 1：

$$\sum_{a} P(A \to \alpha) = 1$$

简单的概率上下文无关语法有三个基本假设：

位置无关(Place invariance)

上下文无关(Context-free)

祖先无关(Ancestor-free)

根据这三个基本假设，可以推导出：分析树的概率等于所有施用规则概率之乘积。

实验表明，这种简单的概率上下文无关语法的使用效果并不理想。在作为一个语言模型使用时（判断一个句子出现的概率），效果还不如简单的 n 元语法。我们自己的实验表明，即使采用这种简单的概率上下文无关语法，分析的正确率大约可以达到 50～60%。这已经比不采用概率时效果要好得多。因为在不采用概率信息时，分析的结果经常是成千上万棵句法树，而从中任选一棵为正确的概率显然非常低。

目前的概率上下文无关语法研究，主要集中在如何突破上述的几个基本假设。通过逐步的放宽这些假设，分析的正确率可以得到很大的提高。现在也有很多研究者在研究词汇化的概率上下文无关语法，为句法树上的每个结点标上中心词信息，并分别计算每条规则在不同中心词搭配下的概率。这样做确实非常有效，可以达到很高的正确率。不过随之而来的问题是数据稀疏问题和搜索空间过大的问题。特别是引入词汇信息以后，数据空间变得非常巨大，数据稀疏问题也极为严重。

对概率上下文无关语法感兴趣的同学可以参考（Charniak. 1996, 1997）以及（Collins, 1996, 1997）。

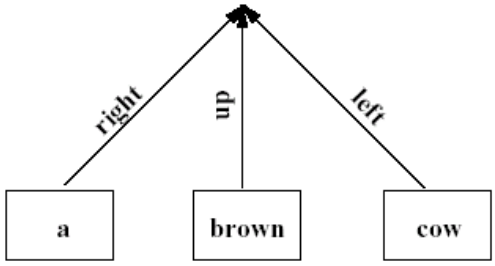
本次会议我们提交的论文（张浩、刘群、白硕，2002）详细介绍了我们在概率上下文无关语法方面的一些初步的工作，在前天晚上的自由讨论中也做了演示，该软件的源代码可以在我们的自然语言处理开放平台中下载，这里不再做详细介绍。

4) 基于统计模式识别的句法分析算法

（Magerman, 1995）中提出了一个 SPATTER 句法分析器，其特点是把句法分析的过程理解为一系列的排歧决策的过程：

$$P(T \mid S) = \prod_{d_i \in T} P(d_i \mid d_{i-1} d_{i-2} \ldots d_1 S)$$

分析从左到右，自底向上地进行。依次决定当前结点是“向左、向上、向右”进行规约。然后还要决定各结点的句法标记和中心词信息。机器学习采用决策树方法。



可以看到，这是一种非常有意思的方法，虽然没有利用任何句法规则，也不采用任何传统的句法分析算法，居然也可以得到很不错的分析效果。不过由于决策树的训练费时费力，这种方法现在使用的并不多。

（WONG&WU，1999）使用了另外一种机器学习的算法进行句法分析。这种方法还是采用传统的移进－规约算法，不过，在发生决策表多重入口冲突的时候，使用一棵决策树来决定采用哪一个动作。

2.2.4 组块分析算法

由于句法分析的效果总是不理想，于是有人开始考虑用“分而制之”的方法来解决句法分析的问题。其基本思想是将完整的句法分析分为两个过程：

(1) 组块的识别：从句子中识别出组块；

(2) 组块之间关系的判断：将组块结合成句子。

现在一般所说的组块分析、浅层分析等等，都是指的前一个阶段，也就是组块的识别的工作。也有很多人在研究某一类特定组块的识别工作，如基本名词短语等等。

组块实际上也就是一种短语。组块的定义，具体到每一种语言都不尽相同。（Abney，1991）中把英语的组块定义为“从句范围内的一个非递归的核心成分”。这种成分包含中心成分的前置修饰成分，而不包含后置附属结构。这个定义非常明确，不过与一般对于英语短语的解释有一定的冲突。

汉语的语法结构与英语差异很大，Abney 的组块定义在汉语中显然难以实施。汉语的组块定义也有不同的方式。我个人认为（孙宏林，2001）中使用的“实语块（content Chunk）是一种比较好的汉语组块定义方法。

英语的组块分析在国际会议 CoNLL-2000(Computaional Natural Language Learning)上作为共同任务(shared task)被提出[Eric et al, 2000]。该会议上对组块分析结果进行评测。它的训练和测试语料是 Upenn 树库的 WSJ(Wall Street Journal)。下表列出了 2000 年参加评测的 10 个系统的性能，包括各个系统对组块的精确率、召回率和综合指标 Fβ。Baseline 表示只根据词性标注划分组块得到的结果，作为系统性能的底线。系统[KM00][Hal00][TKS00]采用了组合系统，系统[ZST00][Koe00][Osb00][PMP00]采用了统计方法，其他的采用了规则方法，这次评测结果表明了统计方法和机器学习的方法在组块分析中是主流趋势，而基于规则的方法也有着不可忽视的作用。

系统名	准确率	召回率	$F_{\beta=1}$
[KM00]	93.45%	93.51%	93.48
[Hal00]	93.13%	93.51%	93.32

[TKS00]	94.04%	91.00%	92.50
[ZST00]	91.99%	92.25%	92.12
[Dej00]	91.87%	91.31%	92.09
[Koe00]	92.08%	91.86%	91.97
[Osb00]	91.65%	92.23%	91.94
[PMP00]	90.63%	89.65%	90.14
[Joh00]	86.24%	88.25%	87.23
[VD00]	88.82%	82.91%	85.76
Baseline	72.58%	82.14%	77.07%

我们这里介绍两种做组块分析的典型思路，一种是基于规则的，一种是基于统计的。

1) 基于层叠有限状态自动机的组块分析方法

（Abney，1996）提出了用 Finite State Cascade 的方法进行英语的组块分析。他这里所说的组块分析包括了上面所说的两层意思，也就是组块的识别和组块间关系的判断，实际上可以实现完整的句法分析。

这种方法的基本思想是：

把句法分析的过程分成很多个层次，每个层次都只输出一个结果，而在每个层次内部只使用简单的有限状态自动机进行分析。

例如使用以下层叠有限状态自动机：

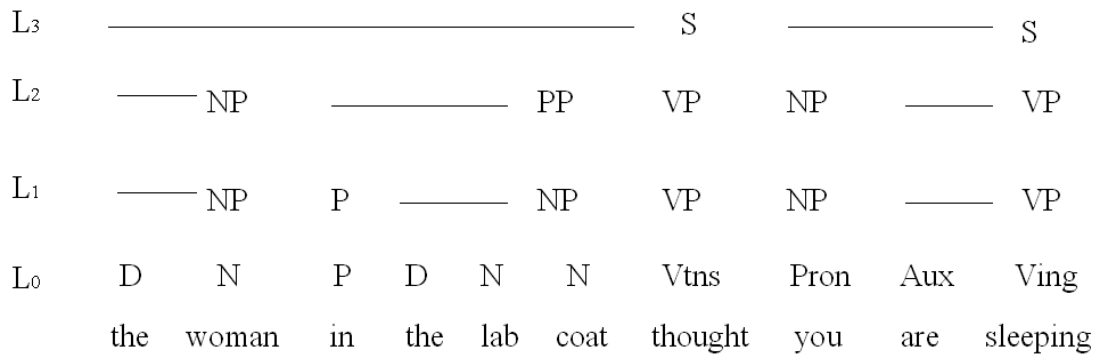
$$T_1:\begin{Bmatrix} \mathrm{NP} \rightarrow (\mathrm{D})\ \mathrm{A}^*\ \mathrm{N}^+ \\ \mathrm{VP} \rightarrow \mathrm{Vtns} \mid \mathrm{Aux\ Ving} \\ \mathrm{NP} \rightarrow \mathrm{Pron} \end{Bmatrix}$$

$$T_2:\{\mathrm{PP} \rightarrow \mathrm{P}\ \ \mathrm{NP}\}$$

$$T_3:\{\mathrm{S}\ \ \rightarrow \mathrm{PP}^*\ \ \mathrm{NP}\ \ \mathrm{PP}^*\ \ \mathrm{VP}\ \ \mathrm{PP}^*\}$$

可以看到，在这个层叠有限状态自动机中，第一层 T_1 可以识别名词短语和动词短语，第二层可以识别介词短语，第三层可以识别句子。

下图是采用这个层叠有线状态自动机识别一个句子的过程：



层叠有限状态自动机是一种比较有效的句法分析算法，不仅分析效率高，而且可以在不使用概率信息的情况下达到较高的正确率。

2) 基于标注的组块分析方法

组块分析的另一种常见的方法就是把组块分析问题转化成类似于词性标注的问题。

以句子“中国十四个边境开放城市经济建设成就显著”为例，词性标注的结果为：

中国/NR 十四/CD 个/M 边境/NN 开放/NN 城市/NN
经济/NN 建设/NN 成就/NN 显著/VV

我们希望得到的组块结构为：

NP(中国) QP(十四个)NP(边境开放城市)NP(经济建设成就)VP(显著)

为了得到这个组块分析的结果，我们可以给这些词加上组块标记：

中国/NP 十四/QP 个/QP 边境/NP 开放/NP 城市/NP
经济/NP$ 建设/NP 成就/NP 显著/VP

其中 NP$表示两个 NP 相连时后一个 NP 的起点。通过这些标记，我们可以很容易得到组块分析的结果。这样我们就把组块分析的问题转换成了一个组块标记的问题，这种标记问题可以用类似词性标记中使用的各种成熟的方法加以解决，例如 HMM 方法、最大熵方法、基于转换的学习方法等等。

（周强，1999,2001）采用了另外一种基于标注的组块分析方法，其基本思想是在词与词之间的间隔进行标注，标注类型为组块的左、中、右边界，然后再根据这些边界信息确定组块的划分及类型。

结语

汉语分析技术是中文信息处理的基础，在机器翻译、信息检索、信息抽取、语音处理等各种应用系统中都有着广泛的应用。全面的了解这些技术，对于从事相关的研究工作是非常重要的。

目前，汉语词法分析技术已比较成熟，组块分析技术也得到了越来越广泛的应用。完全句法分析技术虽然的结果还不能令人满意，不过随着各种语言资源的日益丰富，算法研究的逐步深入，相关的研究工作进展也非常快。相信不久的将来，汉语分析技术将为各种中文信息处理的各种应用提供更加有力的支持。

参考文献：

孙茂松，左正平，黄昌宁，2000，汉语自动分词词典机制的研究实验，中文信息学报，Vol.14，No.1，2000

孙茂松，左正平，邹嘉彦，1999, 高频最大交集型歧义切分字段在汉语自动分词中的作用，中文信息学报，Vol.13，No.1，1999

Hong I Ng and Kim Teng Lua, 2002, A Word Finding Automation for Chinese Sentence Tokenization, submitted to ACM Transaction of Asian Languages Processing. (Can be

downloaded from: http://cslp.comp.nus.edu.sg/luakt/paper/publication.html)

Aho A.V. and Corasick M.J. Handbook of Theoretical Computer Science, Volume A, Algorithms and Complexity, The MIT Press, Cambridge, Massachusetts, 1990. 273-278.

王显芳，2001，一种基于“长词优先”原则的能够检测所有交叉歧义的汉语自动分词算法

王显芳，2001，利用覆盖歧义检测法和统计语言模型进行汉语自动分词

高山，张艳，徐波，宗成庆，韩兆兵，张仰森，2001，基于三元统计模型的汉语分词标注一体化研究，全国第五届计算语言学联合学术会议（JSCL2001）

李建华，王晓龙，2000，中文人名自动识别的一种有效方法，高技术通讯，High Technology Letters，Vol.10，No.2，P.46-49

孙茂松，黄昌宁，高海燕等，1995，中文姓名的自动辨识，中文信息学报，Vol.19, No.2

张俊盛，陈舜德，郑萦，1992，多语料库做法之中文姓名辨识，中文信息学报，Vol.16, No.3

郑家恒，刘开瑛，1993，自动分词系统中性质人名处理策略探讨，陈力为主编，《计算语言学研究与应用》，北京语言学院出版社

宋柔，朱宏，潘维桂，尹振海，1993，基于语料库和规则库的人名识别法，陈力为主编，《计算语言学研究与应用》，北京语言学院出版社

孙茂松，张维杰，1993，英语姓名译名的自动辨识，陈力为主编，《计算语言学研究与应用》，北京语言学院出版社

Rabiner L. R., 1989, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE, vol. 77, no. 2, Feb. 1989, pgs 257 - 285. There is a lot of notation but verbose explanations accompany.

翁富良，王野翊，计算语言学导论，中国社会科学出版社，1998

Huaping Zhang, Qun Liu, Hao Zhang,Xueqi Cheng, 2002, Automatic Recognition of Chinese Unknown Words Based on Role Tagging, SigHan Workshop, 19th International Conference on Computational Linguistics.

张华平，刘群，2002，基于角色标注的中国人名自动识别研究，第七届研究生学术研讨会

白硕，2001，计算语言学教程（讲义）

詹卫东，2002，计算语言学概论（讲义）
http://icl.pku.edu.cn/doubtfire/Course/CL/2001_2002_2.htm

白硕，1998，论语义重心偏移，待发表

The HTK HMM Toolkit，
http://svr-www.eng.cam.ac.uk/research/projects/HTK_HMM_Toolkit

The CMU-Cambridge Statistical Language Modeling toolkit，
http://svr-www.eng.cam.ac.uk/~prc14/toolkit.html

Hiyan Alshawi, Srinivas Bangalore, Shona Douglas, 2000, Learning Dependency Translation Models as Collections of Finite State Head Transducers, Computational Linguistics, Vol.26, No.1, Page 45-60

Steven Abney (1996). Partial Parsing via Finite-State Cascades. In *Proceedings of the ESSLLI '96 Robust Parsing Workshop.*冯志伟，2000，基于短语结构语法的句法分析方法，当代语言学，Vol.2，No.2

白硕，张浩.角色反演算法. 软件学报. 已录用. 2002.

张浩，刘群，白硕，2002，结构上下文相关的概率句法分析，第一届学生计算语言学研讨会（SWCL2002）

Gerald Gazdar, Chris Mellish, 1989, Natural Language Processing in Lisp, Addison-Wesley Publishing Company, ISBN 0-201-17825-7, page 181-214

Abney Steven, 1991,Parsing by Chunks, In: Robert Berwick, Steven Abney and Carol Tenny (eds.), Principle-Based Parsing, Kluwer Academic Publishers, 1991, pp.257-278

李素建，2002，汉语组块计算的若干研究，中国科学院计算技术研究所博士论文

孙宏林，2001，现代汉语非受限文本的实语块分析，北京大学博士论文

俞士汶、朱学锋、段慧明，大规模现代汉语标注语料库的加工规范，《中文信息学报》，2000 年 6 期, PP58-64

Charniak, E. 1996. Treebank grammars. Technical Report CS-96-02, Department of Computer Science, Brown University.

Charniak, E. 1997. Statistical parsing with a context-free grammar and word statistics. In Proceedings of NCAI-1997, pp 598—603

Collins, M. J. 1996. A new statistical parser based on bigram lexical dependencies, In Proceedings of 34th Annual Meeting of the Association for Computational Linguistics, pages 184-191

Collins, M. J. 1997. Three generative, lexicalised models for statistical parsing. In Proceedings of 35th Annual Meeting of the Association for Computational Linguistics, pages 17-23

Magerman, D. M. 1995. Parsing as Statistical Pattern Recognition. IBM Technical Report No. 19443. December 1993.

WONG Aboy, WU Dekai, 1999. Are Phrase Structured Grammars Useful in Statistical Parsing, in Proceedings of 5th Natural Language Processing Pacific Rim Symposium, Page 120-125

Erik F. Tjong Kim Sang and Sabine Buchholz, 2000, Introduction to the CoNLL-2000 Shared Task: Chunking. In: Proceedings of CoNLL-2000 and LLL-2000, Lisbon, Portugal, 2000

W.J. Teahan, Yingying Wen, Rodger McNab, Ian H. Witten, 2000, A compression-based algorithm for Chinese word segmentation Computational Linguistics, Vol.26, No.3, Page 375-393

周强，2001，汉语句法知识的自动获取研究，中文信息学会二十周年学术会议论文集，北京，156-165.

周强，黄昌宁，1999，汉语结构优先关系的自动获取，《软件学报》, 10(2), 149-154.

Qiang Zhou, Fuji Ren, 1999, Automatic Inference for Chinese Probabilistic Context-free Grammar, In Proceedings of 5th Natural Language Processing Pacific Rim Symposium 73-78.