第 5 章

网页信息预处理技术

对于信息预处理系统来说，最主要的工作就是从抓取的网页中提取有价值的，能够代表网页的属性（如网页的 URL、编码类型、标题、正文、关键词等），并将这些属性组成一个网页的对象。然后根据一定的相关度算法进行大量复杂的计算，得到每一个网页针对页面内容及链接每一个关键词的相关度，并用这些信息建立索引数据库。

从网页中提取关键词，至少要做两部分的工作：一是将网页的源代码整理成一个有层次的、利于分析的、包含原始网页中的各种属性的网页对象，即 DOM 树；二是从整理出来的网页对象中提取文本内容，将这些文本内容切分成以词为单位的集合。

本章主要介绍网页信息结构化、文本处理技术和 PageRank 算法。

5.1 网页信息结构化

结构化数据是指被标签定义了其内容、意义和用法的数据。结构化数据除了包含数据本身之外，一般还包含对数据的描述信息，而且其中的数据与描述信息都按照严格的规则进行组织。语法上不具有层次特点的数据称为非结构化数据。介于结构化数据和非结构化数据之间的数据称为半结构化数据。

结构化的数据模型可以用二维表（关系型）来表示，半结构化数据模型可以用树和图来表示，非结构化数据没有数据模型。结构化数据的特点是先有结构、再有数据，半结构化数据的特点是先有数据、再有结构。典型的结构化数据的格式有 XML、XHTML、INI 等，半结构化数据的格式有 HTML。

5.1.1 网页结构化的目标

网页结构化的目标是根据搜索的需要，将半结构化的 HTML 网页中的数据按照约定的基本属性组合成一个网页的对象。一个网页对象至少有以下 5 个属性。

（1）锚文本（anchor text）：锚文本除了网页标题可以描述网页之外，还会有一些锚文本来描述它。例如，百度的主页可能被另外一些网页中存在的锚所指向。

（2）标题（title）：标题是指 HTML 中＜title＞＜/title＞中间的文字部分，这部分文字表达了网页的基本含义，和锚文本一样都是用来描述网页内容的属性。

（3）正文标题（content title）：在 HTML 中，由于一般的网页在＜title＞标签中都不写内容，因此要抽取正文中的适当文字作为正文的标题。

（4）正文（content）：正文是一个网页的主题内容，它完整地表述了网页中的主体内

容。一般出现在<body> </body>中。

(5) 正向链接(link)：正向链接是网页引导用户浏览下一个页面的锚点，这些链接的文字也是其他网页的锚文本。

例如，建立一个简单的 HTML 文件，文件名为 index.html。

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>搜索引擎技术与应用</title>
</head>
<body>
<table >
  <tr><td>搜索引擎技术与应用:第 1 章</td></tr>
  <tr><td>搜索引擎技术与应用:第 2 章</td></tr>
  <tr><td>搜索引擎技术与应用:第 3 章</td></tr>
</table>
</body>
</html>
```

用 IE 浏览器打开后，如图 5-1 所示。

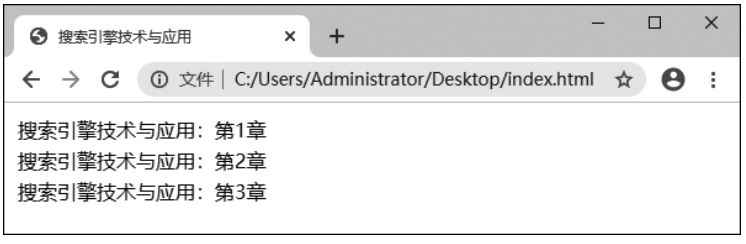

图 5-1 简单网页示例

从图 5-1 中可以看出，在浏览器标题栏中显示的是标题，在浏览器中显示的是网页的正文。为了完成结构化的目标，首先要建立 HTML 标签树，即 DOM 树，然后要对网页的正文进行投票来识别正文的文本，并按照深度优先的遍历规则组织正文。

5.1.2 建立 DOM 树

DOM 的全称是 Document Object Model，即文档对象模型。在使用过程中，首先要将 HTML 网页转换成 XML 格式，然后使用 XML 分析器将一个 XML 文档转换成一个对象模型的集合(通常称 DOM 树)。应用程序正是通过对这个对象模型的操作，来实现对 XML 文档数据的操作。通过 DOM 接口，应用程序可以在任何时候访问 XML 文档中的任何一部分数据，因此，这种利用 DOM 接口的机制也被称作随机访问机制。

DOM 接口提供了一种通过分层对象模型来访问 XML 文档信息的方式，这些分层对象模型依据 XML 的文档结构形成了一棵节点树。无论 XML 文档中所描述的是什么类型的信息，利用 DOM 所生成的模型都是节点树的形式。也就是说，DOM 强制使用树模

型来访问 XML 文档中的信息。由于 XML 本质上就是一种分层结构，所以这种描述方法是相当有效的。

1. 网页内容的 DOM 树表示

建立 DOM 树的过程就是将网页中的标签按照出现的顺序整理出来，并用适当的结构记录过程。由于标签之间的嵌套关系，因此整理结果是一个树状结构。

在用户浏览的网页中，那些 HTML 标签，如＜HTML＞和＜TABLE＞在显示过程中都不会以字符的方式显示出来，这是因为浏览器在解析网页过程中也进行了建立 DOM 树的操作。因此，我们的系统需要建立一个 DOM 树来对抓取的网页进行分析。如图 5-2 所示为 index.html 文件的 DOM 树。

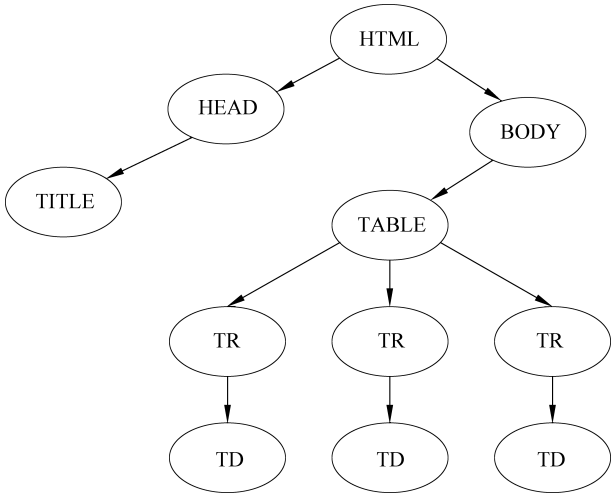

图 5-2　HTML 文件对应的 DOM 树

2. DOM 树的建立过程

HTML 语法中，各种标签都是成对出现的，这样就可以分析一个标记的始末。因此，在解析网页的过程中需要一个标签分析栈的数据结构。栈结构是一种先进后出的线性表结构，栈结构的这种特性为分析工作提供了可能。具体步骤如下。

(1) 建立标签分析栈。

(2) 顺序读取网页标签并依次入栈。

(3) 文本节点不入栈。

(4) 成对标签同时退栈。

DOM 树建立以后，遍历树中的每个节点，将其中的文本送到分词模块进行处理。下面是用 Java 编写的一段遍历 DOM 树的程序。

```
public class RecurDOM(NodeList nodelist)
{
    Node node;
    int i;
    if(nodelist.getLength() ==0)
```

```
        {
            //该节点没有子节点返回
            return;
        }
        for(i=0;i<nodelist.getLength();i++)
        {
            node =nodelist.item(i);
            if(node.getNodeType() ==Node.ELEMENT_NODE)
                RecurDOM(node.getChildNodes());     //递归调用
        }
    }
```

5.1.3 网页内容的获取

由于网页半结构化的特性，得到一个完整的网页内容非常复杂。首先，网页中没有明显的标签标识出网页的正文，其次，正文可能分散在多个 HTML 标签中，问题是如何组合才能获得完整的网页正文。

1. 正文分块

正文具有分块保存的特性，因此引入文本块的概念，对于那些诸如＜P＞＜/P＞等标签间的文本认为是一个文本块。例如，“＜TD＞搜索引擎技术与应用：第 1 章＜/TD＞”称为一个文本块。一般来说，网页会出现以下 3 种类型的文本块。

1）主题型文本块

主题型文本块是大段文字的文本块，如“＜TD＞搜索引擎技术与应用：第 1 章＜/TD＞”。

2）目录型文本块

目录型文本块是描述链接的文本块，如“＜a href＝" "＞搜索引擎技术与应用：第 1 章＜/a＞”。

3）图片型文本块

图片型文本块是描述图片的文本块，如“＜img src＝" "＞搜索引擎技术与应用：第 1 章＜/img＞”。

2. 投票算法

目录型文本块和图片型文本块相对容易被区分；而主题型文本块中可能包含广告等其他内容，必须与正文相区别。判断哪个文本块是正文采用称为“投票算法”的计算方法，这种方法在搜索引擎中特别常用。

投票算法的过程是：首先定义一系列规则，然后通过这些规则为每一个文本块打分。得分最高的被认为是正文的可能性足够大，并且可以接受。

假定一个规则集合中包含以下几条规则，实际的规则往往更加繁多和复杂。

（1）如果文本块文本的长度少于 10 个字，得分为 0；10～50 个字得分为 5 分；50～250 个字，得分为 8 分；超过 250 个字，得分为 10 分。

（2）如果文本块文本的位置在右侧，得分为 0 分；在顶部，得分为 3 分；在左侧，得分

为5分；在中间，得分为10分。

投票算法的过程是依据不同规则从不同的角度依次打分，文本块得分高的是正文的一部分。

除此之外，规则的定义还需要通过足够多的网页进行反馈，之后才能得到一个公正客观的打分。最后还需要注意：如果规则A打分为0～1000分，规则B为0～10分，那么很显然，规则B对最后打分结果影响力几乎可以不考虑，因此如何平衡规则对最后结果的影响也需要充分考虑。这是因为规则的重要性不同可以适当拉开距离，但有时需要在可以接受的范围内保持这种距离。

3. 提取正文

打分之后的工作就是将一个个文本块组织成一个正文。深度优先遍历DOM树并依次记录主题类型的文本块，即可得到该网页的正文。如图4-2所示，按照深度优先，可以依次提取文本块并按照顺序组织成正文“搜索引擎技术与应用：第1章搜索引擎技术与应用：第2章搜索引擎技术与应用：第3章”。

对于其他的网页属性抽取，例如正文标题等也大多采用相同或类似的方法。

5.2　文 本 处 理

不是所有的单词都能等同地表示一个文本的语义。在书面语言中，一些词汇与其他词汇相比能够表达更多的意思。一般说来，名词是最能够表达文档的内容的，这样就有必要对文档进行预处理，以决定对哪些词汇建立索引。在对文档进行预处理的过程中，还有一些其他有用的文本操作，如无用词汇的删除、词干提取技术、词典的生成和文本的压缩等。

文本处理的过程可以分为如下5个步骤。

(1) 文本的词法分析。主要是对文本中的数字、连接符、标点符号和字符的大小写进行处理。

(2) 无用词汇的删除。主要是过滤掉那些对于信息获取过程来说区分能力低的词汇。

(3) 词干提取。主要是去除词缀(前缀和后缀)，这样可以允许所获取的文档包含一些查询词条的变换形式。

(4) 索引词条/词干的选择。在选择的时候通常按照单词的习惯用法，实际上名词往往要比形容词、副词和动词包含更多的语义。

(5) 构造词条的分类结构。例如，词典或者结构抽取，利用它可以进行查询的扩展。

在中文信息的获取过程中，还需要利用中文分词技术对文本进行预处理。

5.2.1　词法分析

词法分析的过程是将字符串(文档中的文本)转换成词条的过程，这些词条可能被用来作为索引词条。因此词法分析的主要目的就是识别文本中的词条。

在对英文进行分词的过程中，除了空格分隔符，还有几种特殊的情况要处理：数字、

连字符、标点符号和字母的大小写。

数字一般不适合用作索引词条，因为对于数字来说，如果不参考上下文，它就没有明确的含义。因此一般说来，信息获取系统都不对数字进行索引，但是常常碰到数字和单词混合的情况，如“2020B.C.”，而该单词是个非常重要的索引词条。因此现在常用的方法是保留一些专门指出的（通过与正则表达式的匹配）数字，而将其他数字过滤掉。对于保留下来的数字，词法分析程序将把日期和一些重要的数字进行规范化，以统一数字的格式。

连字符是造成词法分析困难的又一原因。一种方法是将连字符都忽略掉，例如，“state-of-the-art”等同于“state of the art”。但是对于那些将连字符作为一个整体中的一部分的词条来说就显然不可行了，如“PS-42D8”等。对于连字符的处理，目前常用的是首先采用一定的规则选出那些对词义有影响的连字符号，然后将其他的连字符都过滤掉。

对于文本中的标点符号，一般说来，在词法分析过程中将被全部去除。但是，对于那些成为单词中一部分的标点符号来说，一般不可以去除，例如，如果在“2020B.C.”中去掉标点符号“.”的话，该单词的意思就完全改变了。然而这种情况不一定会影响信息获取的性能，因为现在大多数信息获取系统中，如果用户在查询串中输入“2020B.C.”，那么在查询串与文档中都去掉标点符号“.”，将不会影响文档的获取。但是还有一种特殊情况，就是如果一个程序片段出现在文本中，这时候就需要区分变量 x.id 与 xid 了。在这种情况下，标点符号应该给予保留。

字母的大小写对于区分索引词条说来一般不是很重要，因此可以将文本中的所有词条都转换成大写或者小写。但是在某些特殊情况下，也需要对大小写进行区分，例如，对于描述 UNIX 命令的文档。这时候用户其实不希望改变文档中的大小写，因为这里的大小写都是约定俗成的。

以上这些问题都是词法分析过程中需要考虑的问题，它们将对文档获取所花费的时间产生重要影响。

5.2.2 中文分词技术

分词指的就是将一个完整的句子划分为一个个词条的过程。这种词条应当满足某种语言规则，以便于为其建立索引。只有通过这样的方式，才能完成对一种语言的分析和检索。

1. 中文分词的方法

关键词查询的前提是将查询条件分解成若干关键词。对于英文来说，分词是一件很容易的事，因为空格就是它们天然的分隔符。一个软件可以很轻易地根据英文文本中的分隔符为之切分出一个一个的单词来。然而对于中文来说，情况就复杂多了。主要的问题是中文的词与词之间没有分界符，需要人为切分。此外，汉语中存在大量的歧义现象，对几个字分词可能有好多种结果。例如，对“中华人民共和国”这样一个词进行分词，那么可以分为“中华”“人民”“共和国”，或是“中”“华人”“民”“共和国”。

很显然，前一种分词方法较好。可是如何来判别分词的好坏呢？如果是人，则可以通过大脑进行分词识别，可是如何才能让机器知道对词组、对句子进行词语的切分呢？可以根据语料库进行总结，获得每个词的出现概率以及词与词的关联信息，这样就可能有效地

排除各种歧义，大幅度提高分词的准确性，从而准确地表述查询请求和文档信息。

中文分词技术采用了统计方法和基于规则的方法来识别词边界和专有名词。下面就具体讲述中文分词的方法。

想对中文进行分词，通常情况下有以下几种方式。

1）单字切分

单字切分，顾名思义，就是按照中文一个字一个字地进行分词。以这样的方式切分出来的词再进行索引，称为字索引。很显然，这不是一种很好的分词方式，因为随着索引的增大，相应索引条目的内容会不断增大，严重影响效率。另外，当用户对索引进行检索时，如果用户输入5个字，则相当于要对索引进行5次检索，严重地影响效率。

2）二分法

二分法，就是指每两个字进行一次切分。例如，对于“北京林业大学”这样一个词组进行二分法切分，则结果如下。

北京/京林/林业/业大/大学

这种切分方式完全不考虑词义、语境，仅机械地对语句进行处理。但在很长一段时间内，它一直是中文分词的一种很方便的方式。根据这样的分词效果建立起来的索引会存有大量垃圾词汇，有些可能是用户根本不可能检索的词。因此，它也不是一种最好的方式。

3）词库分词

一直以来，词库分词被认为是最理想的一种中文分词方式。词库分词其实就是用一个已经建立好的词的集合（按某种算法）去匹配目标，当遇上集合中已经存在的词时，就将其切分出来。例如，词库中已经存在了“天涯若比邻”这个词时，分词器就会把它当作一个词条加入索引。

很显然，对于这种分词方式，词库的建立便成了关键。通常，词库的建立需要统计大量的内容，然后根据各种词出现的频率、概率再来进行筛选，最终决定什么词应当放入词库。

另外，一些更加高级的词库还加入了语义和词性的标注，甚至还有不同词的权重。使用这样的词库进行分词的效果应该是很理想的。

2. 中文分词的系统评价

通常对于分词系统性能的鉴定主要依据以下几种评价指标。

1）用户响应度

主要指用户对这项技术的满意度，顾客的满意度也只有系统的质量、系统的性能、系统的稳定性和系统的兼容性才能达到。

2）兼容性

兼容性非常重要，因为不同的企业或个人所使用的系统并不完全相同，所以希望能在不同的系统中都可以毫无障碍地使用，而且能给各行各业都带来方便。

3）准确率

$$\text{准确率} = \text{切分结果中正确分词数} / \text{切分结果中所有分词数} \times 100\%$$

准确率是分词系统性能的核心指标，系统的准确率越高越好，能接近100%是开发者所期望达到的，也是用户所希望的。目前，分词系统的准确率已经达到了98%，但是这并

不是最理想的。若这种分词系统被用来支持句法分析，假定平均每句话有 10 个汉语词，那么 10 句话中会错切 2 个词，含有切分错误的 2 个词就不可能被正确处理。因此仅由于分词阶段的准确度不够，语言理解的准确率就会减少 20%。所以对于分词技术来讲，准确率尽可能接近 100%是最好的。

4）运行效率

在分词系统中分词是各种汉语处理应用系统中共同的、基础性的工作，这步工作消耗的时间应尽量少，使用户没有等待的感觉，在普遍使用的平台上大约每秒钟处理 1 万字或 5000 词以上为宜，这样实现高效率。

5）适用性

适用性是普遍性的基础，汉语分词也是一样，它只是一种实现方法而不是目的，任何分词系统产生的结果都是为某个具体的应用服务的。好的分词系统具有良好的适用性，可以方便地集成在各种各样的汉语信息处理系统中。

6）通用性

随着网络信息化的飞速发展，以及网络信息使用的普遍化，我们研究的中文平台的处理能力不能仅限于国内，仅限于日常应用领域。作为各种高层次中文处理的共同基础，中文分词系统必须具有很好的通用性。中文分词系统应支持不同地区的汉语处理；应能适应不同地区的不同用字、用词，不同的语言风格，不同的专用名构成方式等；支持不同的应用目标，包括各种输入方式、简繁转换、语音合成、校对、翻译、检索、文摘等；支持不同领域的应用，包括社会科学、自然科学和技术，以及日常交际、新闻、办公等；为了做到足够通用又不过分庞大，必须做到在词表和处理功能、处理方式上能灵活组合装卸，有充分可靠和方便的维护能力，有标准的开发接口。同时，系统还应该具有良好的可移植性，虽然不能做到完全通用，但是也要相对全面。

3. 中文分词算法

中文分词的算法主要有：正向最大匹配、逆向最大匹配、双向最大匹配、最佳匹配法、最少分词法、词网格算法、逐词遍历法、设立切分法、有穷多层次列举法、二次扫描法、邻接约束法、邻接知识约束法和专家系统法等。

现有的分词算法可分为 3 大类：基于字符串匹配的分词方法、基于理解的分词方法和基于统计的分词方法。

1）基于字符串匹配的分词方法

这种方法又叫作机械分词方法，它是按照一定的策略将待分析的汉字串与一个“充分大的”机器词典中的词条进行匹配，若在词典中找到某个字符串，则匹配成功（识别出一个词）。按照扫描方向的不同，串匹配分词方法可以分为正向匹配和逆向匹配；按照不同长度优先匹配的情况，可以分为最大（最长）匹配和最小（最短）匹配；按照是否与词性标注过程相结合，又可以分为单纯分词方法和分词与标注相结合的一体化方法。常用的几种机械分词方法如下。

（1）正向最大匹配法（Forward Maximum Matching method，FMM）。FMM 算法是正向最大匹配算法，它是基于字符串匹配的一种分词方法，其主要的算法思想是，选取包含 6～8 个汉字的符号串作为最大符号串，把最大符号串与词典中的单词条目相匹配，如

果不能匹配，就减掉一个汉字继续匹配，直到在词典中找到相应的单词为止。匹配的方向是从左向右。下面应用一个简单的例子来说明算法的过程。

假设要分词的语料为“管理学基础课程是十个学时”，设定一个最大词长为5。

P_1=“管理学基础课程是十个学时”，MAXLEN=5，P_2=“ ”。

① P_2=“”；P_1 不为空，从 P_1 左边取出候选子串 M=“管理学基础”。

② 查词表，“管理学基础”在词表中，将 M 加入到 P_2 中，P_2=“管理学基础/”，并将 M 从 P_1 中去掉，此时 P_1=“课程是十个学时”。

③ P_1 不为空，从 P_1 左边取出候选子串 M=“课程是十个”。

④ 查词表，M 不在词表中，将 M 最右边一个字去掉，得到 M=“课程是十”。

⑤ 查词表，M 不在词表中，将 M 最右边一个字去掉，得到 M=“课程是”。

⑥ 查词表，M 不在词表中，将 M 最右边一个字去掉，得到 M=“课程”。

⑦ 查词表，M 在词表中，将 M 加入到 P_2 中，P_2=“管理学基础/课程/”，并将 M 从 P_1 中去掉，此时 P_1=“是十个学时”。

⑧ P_1 不为空，从 P_1 左边取出候选子串 M=“是十个学时”。

⑨ 查词表，M 不在词表中，将 M 最右边一个字去掉，得到 M=“是十个学”。

⑩ 查词表，M 不在词表中，将 M 最右边一个字去掉，得到 M=“是十个”。

⑪ 查词表，M 不在词表中，将 M 最右边一个字去掉，得到 M=“是十”。

⑫ 查词表，M 不在词表中，将 M 最右边一个字去掉，得到 M=“是”，这时 M 是单字，将 M 加入到 P_2 中，P_2=“管理学基础/课程/是/”，并将 M 从 P_1 中去掉，此时 P_1=“十个学时”。

⑬ P_1 不为空，从 P_1 左边取出候选子串 M=“十个学时”。

⑭ 查词表，M 不在词表中，将 M 最右边一个字去掉，得到 M=“十个学”。

⑮ 查词表，M 不在词表中，将 M 最右边一个字去掉，得到 M=“十个”。

⑯ 查词表，M 不在词表中，将 M 最右边一个字去掉，得到 M=“十”，这时 M 是单字，将 M 加入到 P_2 中，P_2=“管理学基础/课程/是/十/”，并将 M 从 P_1 中去掉，此时 P_1=“个学时”。

⑰ P_1 不为空，从 P_1 左边取出候选子串 M=“个学时”。

⑱ 查词表，M 不在词表中，将 M 最右边一个字去掉，得到 M=“个学”。

⑲ 查词表，M 不在词表中，将 M 最右边一个字去掉，得到 M=“个”，这时 M 是单字，将 M 加入到 P_2 中，P_2=“管理学基础/课程/是/十/个/”，并将 M 从 P_1 中去掉，此时 P_1=“学时”。

⑳ P_1 不为空，从 P_1 左边取出候选子串 M=“学时”。

㉑ 查词表，M 在词表中，将 M 加入到 P_2 中，P_2=“管理学基础/课程/是/十/个/学时/”，并将 M 从 P_1 中去掉，此时 P_1=“”。

㉒ P_1 为空，输出 P_2 作为分词结果，分词过程结束。

以上是对一个简单语料进行正向最大匹配算法的完整过程。

最大匹配算法也同样存在一些不足，对于词长的确定无法做到恰当，掩盖了分词歧义；能发现部分交集性歧义，并且可以通过回溯机制改进算法。

(2) 逆向最大匹配法(Backward Maximum Matching method,BMM)。逆向最大匹配法也是基于字符串匹配的一种分词方法,基本算法和正向最大匹配法相似,只是匹配的方向是从左到右,它的算法比 FMM 的精确度高一些。

(3) 双向匹配法(Bi-direction Matching method,BM)。将 FMM 法和 BMM 法结合起来的算法称为双向匹配法,这种算法通过比较两者的切分结果,来决定正确的切分,而且可以识别出分词中的交叉歧义。

(4) 最少匹配算法(Fewest Words Matching,FWM)。FWM 算法实现的分词结果中含词数最少,它和在有向图中搜索最短路径很相似。控制首先要对所选的语料进行分段,然后逐段计算最短路径,得到若干个分词结果,最后进行统计排歧,确定最理想的分词结果。

(5) 网格分词算法。网格分词算法是基于统计性的一种分词算法,它的算法思想是:首先构造候选词网格,利用词典匹配,列举输入句子所有可能的切分词语,并且以词网格形式保存;然后计算词网格中的每一条路径的权值,权值是通过计算图中每一节点(每一个词)的一元统计概率和节点之间的二元统计概率的相关信息计算出来的;最后根据搜索算法在图中找到一条权值最大的路径,作为最后的分词结果。

另外,还有一种方法是改进扫描方式,称为特征扫描或标志切分。该方法优先在待分析字符串中识别和切分出一些带有明显特征的词,以这些词作为断点,可将原字符串分为较小的串再来进行机械分词,从而减少匹配的错误率。另一种方法是将分词和词类标注结合起来,利用丰富的词类信息对分词决策提供帮助,并且在标注过程中又反过来对分词结果进行检验、调整,从而极大地提高切分的准确率。

2) 基于理解的分词方法

这种分词方法是通过让计算机模拟人对句子的理解,达到识别词的效果。其基本思想就是在分词的同时进行句法、语义分析,利用句法信息和语义信息来处理歧义现象。它通常包括 3 个部分:分词子系统、句法语义子系统、总控部分。在总控部分的协调下,分词子系统可以获得有关词、句子等的句法和语义信息来对分词歧义进行判断,即它模拟了人对句子的理解过程。这种分词方法需要使用大量的语言知识和信息。由于汉语语言知识的笼统、复杂性,难以将各种语言信息组织成机器可直接读取的形式,因此目前基于理解的分词系统还处在实验阶段。

3) 基于统计的分词方法

从形式上看,词是稳定的字的组合,因此在上下文中,相邻的字同时出现的次数越多,就越有可能构成一个词。因此字与字相邻共现的频率或概率能够较好地反映成词的可信度。可以对语料中相邻共现的各个字的组合的频度进行统计,计算它们的互现信息。定义两个字的互现信息,计算两个汉字 X、Y 的相邻共现概率。互现信息体现了汉字之间结合关系的紧密程度。当紧密程度高于某一个阈值时,便可认为此字组可能构成了一个词。这种方法只需对语料中的字组频度进行统计,不需要切分词典,因而又叫作无词典分词法或统计取词方法。但这种方法也有一定的局限性,会经常抽出一些共现频度高但并不是词的常用字组,例如,“这一”“之一”“有的”“我的”“许多的”等,并且对常用词的识别精度差,时空开销大。实际应用的统计分词系统都要使用一部基本的分词词典(常用词词典)