中 文 信 息 学 报
第 14 卷 第 1 期 JOURNAL OF CHINESE INFORMATION PROCESSING Vol. 14 No. 1

汉语自动分词词典机制的实验研究*

孙茂松　左正平　黄昌宁

清华大学计算机科学与技术系　北京　100084

摘要　分词词典是汉语自动分词系统的一个基本组成部分。其查询速度直接影响到分词系统的处理速度。本文设计并通过实验考察了三种典型的分词词典机制：整词二分、TRIE 索引树及逐字二分，着重比较了它们的时间、空间效率。实验显示：基于逐字二分的分词词典机制简洁、高效，较好地满足了实用型汉语自动分词系统的需要。

关键词　中文信息处理　汉语自动分词　汉语自动分词词典机制

An Experimental Study on Dictionary Mechanism for Chinese Word Segmentation

Sun Maosong　Zuo Zhengping　Huang Changning

The State Key Laboratory of Intelligent Technology and Systems,
Department of Computer Science and Technology, Tsinghua University　Beijing　100084

Abstract　The dictionary mechanism serves as one of the basic components in Chinese word segmentation systems. Its performance influences the segmentation speed significantly. In this paper, we design and implement three typical dictionary mechanisms, i.e. binary-seek-by-word, TRIE indexing tree and binary-seek-by-characters, from word segmentation point of view, and compare their space and time complexity experimentally. It can be seen that the binary-seek-by-characters model is the most appropriate one being capable of fulfilling the need for speed of practical Chinese word segmenters to the maximum extent.

Keywords　Chinese information processing　Chinese word segmentation　Dictionary mechanism for Chinese word segmentation

一、引言

分词词典是汉语自动分词系统的一个基本组成部分[1]。自动分词系统所需要的各类信息(知识)都要从分词词典中获取，分词词典的查询速度直接影响到分词系统的速度。而现实

* 本研究得到国家自然科学基金资助(合同号：69433010)
本文于 1999 年 4 月 6 日收到

应用(如因特网上的中文文本检索、汉字与汉语语音识别系统的后处理以及中文文语转换系统的前处理等)均对分词速度提出了迫切要求,因此建立高效快速的分词词典机制势在必行。

针对分词系统的特点,可以将分词词典的查询操作大致分为三种基本方式:

查询方式 1:在分词词典中查找指定词 w_0(词在分词词典中的定位)

这是一种最基本的查询方式。给定词 w_0,返回 w_0 在分词词典中的位置,以便得到 w_0 的各类附属信息。此时 w_0 是确定的,所以可以简单地通过二分查找给出结果。

查询方式 2:根据分词词典,在汉字串 S 中查找从某一指定位置 i 开始的最长词 $w_{i,\max}$(对应最大匹配分词法)

有别于查询方式 1,这里最长词 $w_{i,\max}$ 无法预知,需要在查询过程中动态地确定。通常的做法是尝试始于位置 i 的所有可能长度的词,多次运用查询方式 1 来完成查询。

查询方式 3:根据分词词典,在汉字串 S 中查找从某一指定位置 i 开始的所有的词 $w_{i,1}$, $w_{i,2}$, …, $w_{i,\max}$(对应全切分分词法)

类似查询方式 2,但返回结果通常不唯一。

本文设计并通过实验考察了三种典型的分词词典机制:(1)基于整词二分(2)基于 TRIE 索引树及(3)基于逐字二分,着重比较了它们的时间、空间效率。

二、三种典型的分词词典机制

实验利用了一个包含 112 967 个不同词的分词词典 THDic。THDic 中最短词只有一个汉字(单字词),最长词则达到 17 个汉字(长词多为成语、惯用语、习用语等),平均词长度约 2.5 个汉字。

2.1 基于整词二分的分词词典机制

这是一种广为使用的分词词典机制。其结构通常分为三级,前两级为索引(见图 1)。

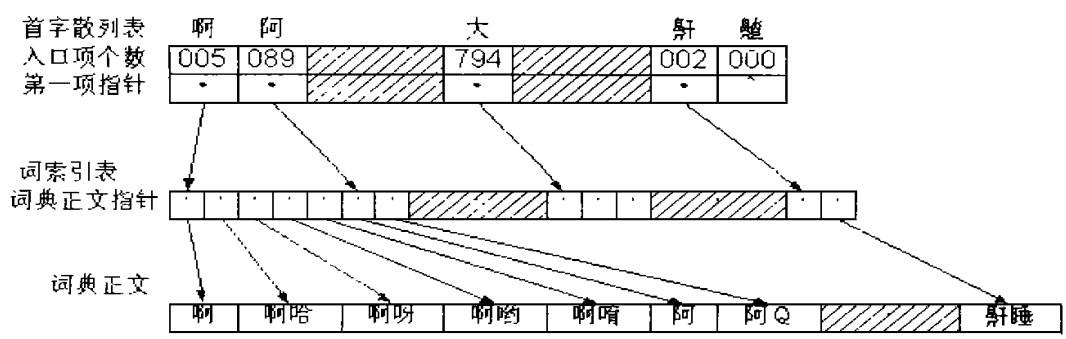

图 1 基于整词二分的分词词典机制

1. 首字散列表

词首字散列函数根据汉字的国标区位码给出。通过一次哈希运算即可直接定位汉字在首字散列表中的序号。

首字散列表的一个单元包括两项内容:

入口项个数(4 字节):以该字为首字的词的个数;

第一入口项指针(4 字节):指向第一入口项在词索引表中的位置。

2. 词索引表

因词的长度可变(实际系统中还包括附属于该词的各类信息),故以选择不定长存储为宜;

此外必须实现对词的随机访问。这两条决定了必须建立词索引表。

词索引表的一个单元仅含一项内容：

词典正文指针(4 字节)：指向词在词典正文中的位置。

3. 词典正文

以词为单位的有序表。通过词索引表和词典正文的配合，很容易实现指定词在词典正文中的整词二分快速查找。

此种分词词典机制比较适合于查询方式 1。而在查询方式 2、3 中，对任一位置 i，通常我们不得不预先主观设定一个词的可能最长长度 l(一般 l 取词典中最长词的长度。THDic 为 17 个字)，然后截取子汉字串 $S[i, i+l-1]$ 作为假想词，在词典中进行整词二分查找。不成功则词长 l 递次减一并循环，直至匹配成功。由于一、二、三、四字词分别占 THDic 的 3.6%、64.0%、16.0%、14.0%，它们更动态覆盖了真实文本绝大部分，因此这种由长词及短词的盲目尝试方法效率很低。

[例] 查询 S“水怪大白天现形一个多小时这个令人惊异的消息不径而走。”中从“大”字开始的最长的词。

(1) 取从“大”字开头最长可能的汉字串：$w_{i,\max}$ =“大白天现形一个多小时这个令人惊异的”(词典中最长词为 17 个汉字)；

(2) 用整词二分法在词典中查找候选词 $w_{i,\max}$，失败；

(3) 去掉 $w_{i,\max}$ 最后一个字，重复步骤(2)，失败；

(4) ……

(5) 经过 14 次的错误尝试，$w_{i,\max}$ 最终消减到“大白天”，查找成功，于是返回 $w_{i,\max}$ =“大白天”。

2.2 基于 TRIE 索引树的分词词典机制

TRIE 索引树是一种以树的多重链表形式表示的键树[4]。面向英文的 TRIE 索引树一般以 26 个字母作为关键字，树结点包含个数相同的指针。汉字接近 7 000 个，如果采用同样的策略构造中文词典，显然将造成指针的大量浪费。面向中文的 TRIE 索引树的结点应允许指针个数变化。

基于 TRIE 树的分词词典由两部分组成(见图 2)。

1. 首字散列表

同基于整词二分的分词词典机制。首字散列表的一个单元是所对应汉字的 TRIE 索引树的根结点。

2. TRIE 索引树结点

TRIE 索引树结点是以下述结构为单元的、按关键字排序的数组：

关键字(2 字节)：单一汉字；

子树大小(2 字节)：以从根结点到当前单元的关键字组成的子串为前缀的词的个数；

子树指针(4 字节)：子树大小非 0 时，指针指向子树；否则指向叶子。

在 TRIE 索引树上查询任意一个词 W[1, n] 的过程为：

(1) 根据首字散列表得到 W[1] 的 TRIE 索引树，沿相应指针移动至目标结点 NODEx；i=2；

(2) 在 NODEx 的关键字域中对汉字 W[i] 进行二分查找。

如果与 NODEx 的第 j 个单元的关键字匹配成功，

则　沿该单元的子树指针移至目标结点，并令该结点为新的 NODEx；

i＝i＋1；

否则查询失败，退出此过程。

（3）　重做（2），直至 NODEx 为叶子结点。

（4）　如果抵达叶子结点时 i＞ n，

则查询成功，W[1，n] 为分词词典中的一个词

否则查询失败。

与整词二分形成鲜明对照的是，基于 TRIE 索引树的分词词典机制每次仅比较一个汉字，不需预知待查询词的长度，且在对汉字串 S 的一遍扫描过程中，就能得到查询方式 2、3 的结果。这种由短词及长词的确定性工作方式避免了整词二分分词词典机制中不必要的多次试探性查询。

由于 TRIE 索引树已蕴涵了词条信息，因此词典中不必再显式地罗列词条，可直接存储词的附属信息（叶子指针直接指向这些信息）。

[例]　查询 S“水怪大白天现形一个多小时这个令人惊异的消息不径而走。”中从“大”字开始的最长词（及所有词）。

（1）　首先通过首字散列表得到以“大”字开头的词的 TRIE 索引树结点 T；

（2）　由于结点 T 中包含关键字“ ”（表示空字符，下文同），因此“大”是一个词。在结点 T 中用二分法查找关键字“白”，其指针指向的目标结点设为 T_1。

（3）　结点 T_1 中包含关键字“ ”，因此“大白”也是一个词。在结点 T_1 中继续用二分法查找关键字“天”，此时“天”的目标结点已是叶子结点，“大白天”也是一个词，查询结束。最后得到“大白天”为最长词，中间过程识别的“大”、“大白”、“大白天”均为从“大”字开始的词。

TRIE 索引树分词词典机制的主要缺点是其构造及维护比整词二分复杂。

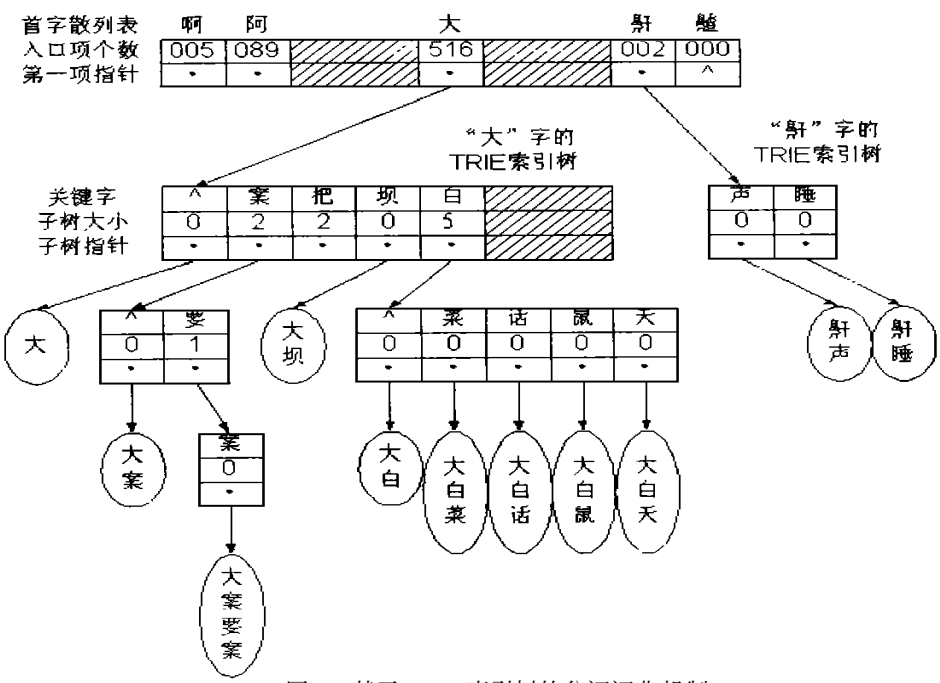

图 2　基于 TRIE 索引树的分词词典机制

2.3　基于逐字二分的分词词典机制

针对整词二分和 TRIE 索引树的不足，这里设计了一种改进方案——基于逐字二分的分词词典机制（见图 3）。逐字二分与整词二分在数据结构上完全一致，不同的仅仅在于查询过程：不再将整个词作为关键字进行比较，而是类似 TRIE 索引树的情形，每次仅仅比较单个的汉字。因而其效果同 TRIE 索引树完全一样。

[例]　查询 *S*“水怪大白天现形一个多小时这个令人惊异的消息不径而走。”中从“大”字开始的最长词（及所有词）。

（1）　通过首字散列表可知：以“大”字开头的词位于词索引表的第 14285～第 15078 项范围内，并且其中的头一项“大”为一个单字词；

（2）　通过二分法在第 14285～第 15078 项的范围内查找第二个字为“白”的词，可知：以“大白”开头的词位于词索引表的第 14292～第 14297 项范围内，并且其中的头一项“大白”为一个二字词；

（3）　通过二分法在第 14292～第 14297 项的范围内查找第三个字为“天”的词，可知：以“大白天”开头的词位于词索引表的第 14296～第 14297 项范围内，并且其中的头一项“大白天”为一个三字词；

（4）　通过二分法在第 14296～第 14297 项的范围内查找第四个字为“现”的词，此范围为空，查询结束。最后得到“大白天”为最长词，中间过程识别的“大”、“大白”、“大白天”均为从“大”字开始的词。

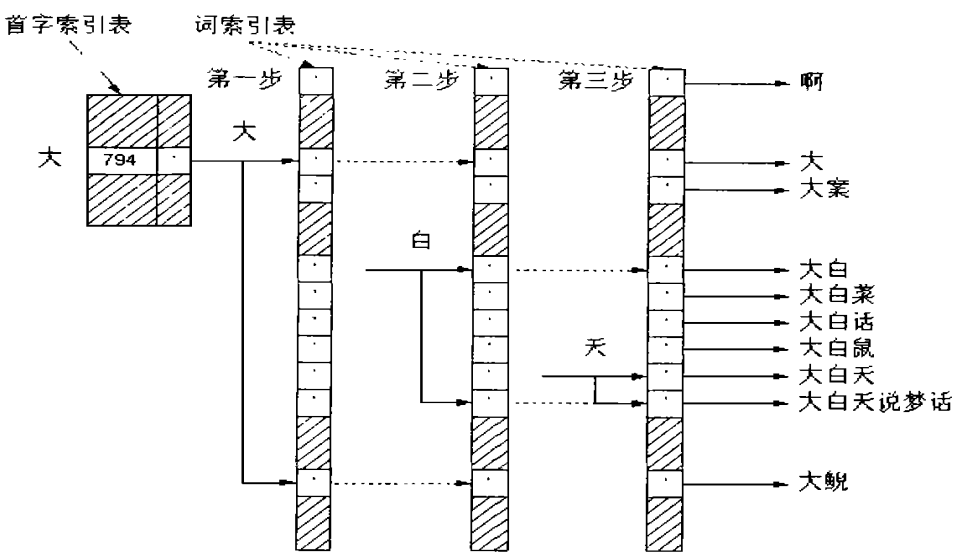

图 3　基于逐字二分的分词词典机制（查询过程）

三、分词词典机制的实验比较

我们从时间和空间角度对第二节所述的分词词典机制进行了实验比较：

1. 空间

（1）　词表（仅含词条，不计词条所含的各类信息）：占用空间 782 678 字节；

（2）　首字散列表：每个单元占 8 字节，共 7 614 个单元（包括一、二级汉字及某些图形符号），故首字散列表占空间 7 614 *8＝60 912 字节；

（3） 词索引表：每个词占 4 字节，共 112 967 个词，故词索引表占空间 112 967 *4＝451 868 字节；

（4） TRIE 索引树：每个 TRIE 索引树结点单元占 8 字节，共含 129 588 个索引项，故 TRIE 索引树占空间 129 588 *8＝1 036 704 字节。

基于整词二分和逐字二分的分词词典机制包括（1）、（2）、（3）项开销。而基于 TRIE 索引树的分词词典机制包括项（2）、（4）项开销。

2. 时间

对每种分词词典机制，我们均设计了四个不同类型的测试：

（1） 对 THDic 的所有词依次查询 1 次。（第一类查询）；

（2） 对 THDic 的所有词依次查询，查询次数与词频成正比。（第一类查询，模拟真实文本环境）；

（3） 从语料库中任意取一段测试文本，用最大匹配分词法切分该文本。（第二类查询）；

（4） 从语料库中任意取一段测试文本，用全切分分词法切分该文本。（第三类查询）。

针对（3）（4），我们从《人民日报》中随机选取了长度为 3 066K 字节的文本作为测试文本。则关于空间、时间的实验结果总汇如下：

查询方法	词典空间（字节）	测试 1（单位时间）	测试 2（单位时间）	测试 3（单位时间）	测试 4（单位时间）
整词二分	1 295 458	500	3 950	105 460	168 950
TRIE 索引树	1 097 616	270	2 140	6 750	8 950
逐字二分	1 295 458	330	980	6 480	9 610

从实验结果来看，三种分词词典机制的空间效率差不多。而基于 TRIE 索引树和逐字二分的分词词典机制的时间效率大致相同，较基于整词二分的词典机制均有大幅度的改善。由于基于逐字二分的词典机制的实现比 TRIE 树简单，所以我们认为就综合性能而言，前者较后者更为优越。由表中可见，对最大匹配分词法和全切分分词法，基于逐字二分的分词词典机制的处理速度较基于整词二分的处理速度分别提高了 15.3 倍和 16.6 倍，效果十分显著。

综上所述，我们的结论是：基于逐字二分的分词词典机制是一种简洁、高速的词典组织模式，最大程度地满足了实用型汉语自动分词系统的现实需要。

参 考 文 献

[1] 孙茂松，邹嘉彦. 汉语自动分词中的若干理论问题. 语言文字应用，1995，（4）

[2] 梁南元. 书面汉语自动分词系统——CDWS. 中文信息学报，1987，2（2）

[3] 马晏. 基于评价的汉语自动分词系统的研究与实现. 见：语言信息处理专论，北京：清华大学出版社，1996

[4] 严蔚敏，吴伟民. 数据结构. 北京：清华大学出版社，1992