维普资讯 http://www.cqvip.com

全文检索模型综述*)

申 展 江宝林 陈 祎 唐 磊 胡运发
(复旦大学计算机与信息技术系 上海200433)

摘 要 全文检索的应用导致了信息检索领域的一场革命,是文档数据库研究的核心。全文检索的首要问题是全文检索模型的选择。本文介绍了目前国内外主要的全文检索模型,总结了全文检索模型的评价标准,按照此标准对各种模型进行了分析比较,并对全文检索模型构建中一些关键问题进行了分析。

关键词 全文检索,信息检索,倒排表,Pat-树,互关联后继树

A Survey on Full-Text Index Model

SHEN Zhan JIANG Bao-Lin CHEN Yi TANG Lei HU Yun-Fa
(Department of Computing and Information Technology, Fudan University, Shanghai 200433)

Abstract The application of full-text index which is the key research work on text databases, leads to a revolution in information retrieval. The primary problem of full-text index is how to choose an effective full-text index model. A full-text index model evaluation criterion is summarized. Then, the main models nowadays are introduced and analyzed. In addition, key techniques are explored.

Keywords Full-text retrieval, Information retrieval, Inverted files, Pat-tree, ISTree

1 引言

全文检索模型的应用有着广泛的意义,它不仅仅导致了信息检索领域的革命,对XML数据库、数据挖掘[1,2]和图像检索等许多方面的研究也都起着推动作用。

目前,国际上流行的全文检索模型主要有:倒排表模型(Inverted Files)[3]、署名文件(Signature Files)、位图(Bit Map)、Pat树[4]和Pat数组[5]。前三种索引模型从实质上讲,是同一基本观念的变体,即把文档看成索引项的集合,因此具有同样的弱点:索引数据必须具有文档-索引项结构,且只能实现简单的查询。Pat树和Pat数组将索引数据看成一组半无限串的叠加,避免了这个弱点。

目前,国内对全文检索模型的研究也方兴未艾,第五届World Multiconference on Systemics, Cybernetics and Informatics(SCI 2001)海量中文信息管理分会上展示了许多全文检索方面的优秀成果[6,7]。Σ^2相邻矩阵模型[6]和互关联后继树模型(ISTree)[8]都是针对中文提出的全文检索模型。这两种模型都能够利用索引生成原文,因而可以完全抛弃原文,节约存储空间。互关联后继树比其他模型有着更加广泛的应用,除了用于全文检索,还可以应用在时间序列查询[1,2],关联规则挖掘[10]等许多方面。

本文首先简单介绍了以上几种模型的原理,总结了一套评价索引模型的标准,对索引模型进行了比较分析,最后列出了全文检索中一些关键技术。

2 全文索引模型介绍

2.1 位图

位图为每个文档存储一个矢量(bitvector),矢量的每个位对应一个索引项。如果文档中出现某个索引项,该位就置1,否则置0。查询包含某索引项的文档时,只需察看哪些文档矢量对应那个索引项的位为1即可。查询仅仅涉及简单的位操作,因而位图使用简便,查询快捷,尤其适合布尔检索。但是即使使用了一些高效的压缩方法后,位图的空间开销还是很大。一些实验表明,位图的空间开销可能是原文本的几十倍。

2.2 署名文件

署名文件有时也称散列函数法。它为每个文档设定一个宽度为w的矢量,把该文档的每个索引项都按照某个散列函数映射到矢量中的某几位上,并将那几位置1。所得结果,即为该文档的署名。

要检测一个查询项是否在给定的文档中出现,首先要计算此查询项的散列函数值。如果某文档署名中所有的对应位均被设定,则此查询项可能出现在该文档中。散列函数的使用,使署名文件的查询结果具有一定的不确定性。为了排除误匹配,必须扫描结果文档以检查查询项是否真正出现。

存在误匹配是署名文件的一个主要缺点。可以通过加大文档署名宽度w来降低误匹配率,当然这是以索引空间的膨胀为代价的。但无论如何,总是需要进行误匹配检查,这在相当程度上增加了查询的开销。假定文档中不同索引项个数为t,散列函数将每个索引项映射为s位,文档署名宽度为w,则误匹配率可以表示为[11]:

$$p(w,s,t)=(1-(1-\frac{1}{w})^{st})^{s}$$

2.3 倒排表模型

在倒排表模型中,字符表Σ中的每个字符都对应一个链

*)基金项目:国家自然科学基金(60173027)。申 展 硕士研究生,主要研究领域为数据库与知识库;江宝林 硕士研究生,主要研究领域为数据库与知识库;陈 祎 硕士研究生,主要研究领域为数据库与知识库;唐 磊 本科生,主要研究领域为数据库与知识库;胡运发 博士生导师,主要研究领域为人工智能。

维普资讯 http://www.cqvip.com

表，链表的每个节点记录了该字符所在的一个文档的文档号、该字符在该文档中的出现频度以及在所有出现位置等信息。具体结构如图1所示，其中 T_{ij}($1\leqslant j\leqslant m$)表示含有字符 a_i 的文档的内部代号，N_{ij}表示文档 T_{ij}中字符 a_i 出现的次数，$[O_{ija}$，O_{ijb}，…]指出了文档 T_{ij}中字符 a_i 出现的具体位置。

a_1	→ $\{T_{11},N_{11},[O_{11a},O_{11b}\cdots]\},\{T_{12},N_{12},[O_{12a},O_{12b}\cdots]\},\cdots\cdots$
a_2	→ $\{\{T_{21},N_{21},[O_{21a},O_{21b}\cdots]\},\cdots\cdots$
……	……
a_n	→ $\{T_{n1},N_{n1},[O_{n1a},O_{n1b}\cdots]\},\cdots\cdots$

图1　倒排表模型结构

检索时，设待查字符串为 $a_1a_2,\cdots a_r$，首先通过字符列表定位各字符的索引数据，然后对数据进行分析：若 $a_1a_2\cdots a_r$ 的索引数据中均含有文档 T 的索引记录，在 r 个关于文档 T 的索引记录中又含有 $O_1,O_2,\cdots,O_r$(O_i 是属于字符 a_i 的索引数据)，且 O_i 和 O_{i+1}($1\leqslant i\leqslant r-1$)的差值刚好是字符 a_i 所占字节数，则文档 T 为一个命中文档。找到所有的命中文档后(或分析完毕后仍找不到命中文档)，检索完成。

2.4　Pat 树和 Pat 数组

下面，我们用一个简单的例子来说明 Pat 树和 Pat 数组的原理。设文档为："ababc"。从每个字符到文档尾字符形成一个字符串(这种字符串称为半无限串)，此文档包含了5个字符，也包含了5个半无限串，分别如下：1)ababc，2)babc，3)abc，4)bc，5)c。我们用半无限串开始字符的位置标记该半无限串，对这些半无限串做词典排序，排序结果是1)3)2)4)5)。文档对应的 Pat-树和 Pat-数组如图2所示。

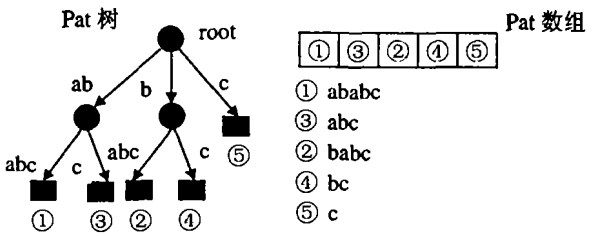


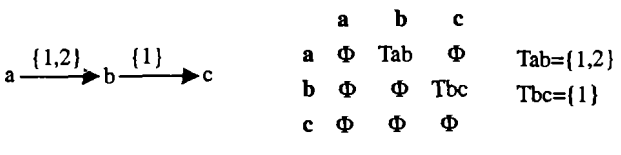
图2　Pat 树和 Pat 数组示例

对于 Pat 树，如果查询"abc"，则从 Pat 树的树根按照分支的标志下降，首先沿着根节点的第一个分支(标志 ab)找到第一层的第一个中间节点，而后再从该中间节点的第二个分支(标志 c)找到 abc 的匹配③。Pat 数组是由 Pat 树的叶结点串行化得到的，反映了文档中所有半无限串的词典排序结果，查询可以直接在数组中进行。

2.5　Σ^2相邻矩阵模型

Σ^2相邻矩阵模型将索引数据看成无环句串的连接，所谓无环句串实际上就是同一字符至多只出现一次的字符串。如文档"ababc"可以分解为无环句串1)"ab"和2)"abc"。如果用有向图来表示该文档如图3a 所示。

有向图的顶点为构成文档的基本字符；边连接文档中两个邻接字符，且由前面的字符指向后面的字符。边的标记为该边对应的字符有序对所在的句串的标号。如果用邻接矩阵表示此有向图就是该文档的 Σ^2相邻矩阵模型(图3 b)。如果查询"abc"从邻接矩阵中取出 ab 和 bc 的值，分别是{1,2}和{1}，然后进行集合求交得到1即产生一个解。



a. 文档有向图　　b. 相邻矩阵

图3

Σ^2相邻矩阵模型中句串的无环性保证了相邻矩阵确定的每个句串是唯一的，各句串的连接，实际上就是模型中加入的原文档，因而利用相邻矩阵模型可以进行原文生成。

2.6　互关联后继树模型

任意文档 T 中的任意字符串 a_1a_2，a_1称为 a_2的前驱；a_2称为 a_1的后继。我们在整个文档的末尾人为添加一个不在该文档中出现的字符，做为该文档的结束符，则文档最后一个字符的后继就是文档结束符。

对于文档 ababc#(其中 # 为索引库的结束)。a 有两个后继都是 b，b 有两个后继分别是 a 和 c，c 的后继为 #。对每个字符建立一棵树，该字符的后继作为树的分支，如图4所示。图中各分支后继后面的序号是该后继的后继在以该后继为根的树中的分支号。如 a 树的第二个分支是(b，2)，因为这个 b 在原文中的后继为 c，而 c 是 b 树的第二个分支。实际上，整个互关联后继树构成的森林记载了文档中所有字符的先后位置，因而利用 ISTree 索引可以还原出原来的文档。

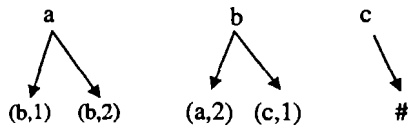


图4　ISTREE 示例

如果查询"abc"，首先查看以 a 为根的树，它有两个后继为 b 的分支，再根据这两个分支中的序号，跳转到 b 树的1，2分支察看其后继是否为 c，发现只有2分支的后继为 c，则 abc 在文档中仅存在一个匹配。

3　全文索引模型的评价标准

目前，世界上还没有一套统一的标准对全文检索模型进行比较全面的功能与性能评估。本文对国内外在评价比较全文检索模型时常用的一些标准进行总结，归纳如下：

1)时间复杂度：包括创建索引的时间效率以及检索的时间效率。对于索引模型更重要的是检索性能，所以，我们更常考虑的是模型的检索速度。

2)空间复杂度：由于现在索引数据的海量性，索引模型中任何能够减少索引空间消耗的机制均可导致不可估量的实用效果。通常，我们用膨胀比(ER)来评价索引模型的空间性能。

ER＝(IS＋FS)/FS (IS 为索引大小，FS 为原文大小)

索引模型的膨胀比与模型的具体实现方式和使用的压缩算法也有很大关系。同时又会影响索引效率。

3)查询完备性：一个好的全文检索模型应该能够实现多种查询，如：

a)字符串查询：包括关键词查询，或较长的字符串查询，比如一个句子。

b)前缀查询：在索引库中查询具有某给定前缀的字符串。

c)临近查询：查询所有字符串 S，S 以 S1字符串开始，以 S2结尾，而 S1，S2之间的最多间隔 n 个字符(n 由用户给定)。

维普资讯 http://www.cqvip.com

d)范围查询:查询所有在某一词典范围内的字符串。如:"abrasion"在"abc"和"acc"范围之内,而"abacus"和"acrimonious"不在。

e)最长重复查询:在文档的两个不同位置寻找相同的字符串,而这个字符串在整个文档中是最长的。

f)最频繁匹配查询:查询在文档中出现次数最多的字符串。

4)适应性:好的索引模型应该可以处理各种不同类型,具有不同特点的数据,如:长度相差很大的记录,或者没有文档,关键词结构的数据。

5)动态性:索引模型最好能够适应数据的动态改变,包括关键词词典中词条的增加,或索引记录的改变等。

4 几种全文检索模型分析比较

下面对上述模型按照评价标准进行分析,由于位图和署名文件使用较少,将不对其进行讨论。

4.1 时间复杂度

查询时间复杂度和具体的查询有关。对于全文检索来说,最常用的查询是字符串匹配,所以这里就以这类查询的时间复杂度来进行比较。设查询字符串 P 的长度为 M,索引库中文档大小为 N。倒排表的检索效率与其存储结构有关。通常,可以按照数组、B^+树、散列函数等方式来实现倒排表,那么查找索引项时需要的比较次数前两者为 O(logN),而后者为 O(1)。如果查询串是单个的关键词,可以直接得出该索引项所在的文档;如果查询词组或句子,则需要对各所有查询关键词位置信息进行分析合并,因而效率较低,分析处理所需时间与索引库中关键词的匹配个数有关。Pat 树中查询需要比较 M 次以确定 P 在树中的位置,其时间复杂度为 O(M)。由于 Pat 数组实际上是文档中半无限串的词典排序,查询时间复杂度为 O(M+logN)[12]。

Σ^2相邻矩阵模型将 P 分解为 M−1个相邻字符对分别进行查询,需要时间 O(M−1),而后需要合并查询结果。在 ISTree 中查询 P 需要遍历所有与 P 具有相同前缀的字符串,直到该字符串与查询串不符或者完全相符。设 $P=a_1,a_2,\cdots,a_n$,在索引库中 a_1出现 f_1次;a_1a_2出现 f_2次;……;$a_1,a_2,\cdots,a_n$出现 f_n 次,则算法需读取磁盘 $f=\sum_{i=1}^{n} f_i$ 次。若所有与 P 具有相同前缀的字符串的平均长度为 r,则算法时间复杂度 O(rf)。此模型的检索效率高于倒排表模型[13]。

4.2 空间复杂度

假设原文档大小为 N,倒排表和 Pat 数组基本上都是对文档中的每个字符存储一定位置信息,索引约占空间 N,则膨胀比为2。Pat 树需要存储大量的中间节点,占用空间较大,为 4N～5N[5],膨胀比为5～6。倒排表和 Pat 树模型出现较早,对其研究也更加广泛,产生了许多压缩算法。虽然解压缩对查询的时间影响很大,但是由于这两种模型的压缩比很高,且索引减小而引起磁盘交换的减少,从而节约的查询时间实际上比解压缩需要消耗的时间更多。倒排表模型在压缩以后膨胀比仅为1.06～1.1[14],Pat 树则仍需要2。

对于Σ^2邻接矩阵模型来说,基本上也是为每个字符建立一个索引,索引存储空间为 N。本文中介绍的 ISTree 模型为每个字符存储一个后继和一个位置信息,而另一种精简 ISTree 模型可以仅仅存储位置信息,而省略后继,则索引空间也为 N。由于Σ^2邻接矩阵模型和 ISTree 能够还原原文,无需保留原文档,所以它们的膨胀比都是1。

4.3 查询完备性

倒排表模型只能够存储具有文档-索引项结构的数据,按照索引项的不同,有三种形式的倒排表:按字倒排,按词倒排和按关键词(在按词倒排的基础上去掉停用词)倒排。基于字或词的倒排表保留了文档中的所有词的位置信息,可以查询任意字符串;而基于关键词的倒排表只能查询关键词,若查询长字符串,需要先将该字符串分解为多个关键词,分别查询后,合并查询结果。用数组、B^+树存储的倒排表实际上反映了索引项的词典顺序,所以也可以进行前缀查询和范围查询。同样,由于记录了索引项在文档中的位置信息,还可以进行临近查询。但是不能够进行最频繁匹配查询和最长重复查询,这是因为通常这两种查询的结果都不限于一个索引项。Pat 数组,Pat 树的查询功能十分强大,都可以实现评价标准中的所有查询,当然,在实现最大重复查询的时候需要增加一定的辅助信息。此外,利用 Pat 树还可以进行规则挖掘[15]。

Σ^2邻接矩阵模型把文档切成无环句串,实际上把所有查询都限制在无环句串之内,因而查询功能受到很大的影响。ISTree 和 Pat 树一样能够实现所有的查询,并且可以进行规则挖掘。

4.4 适应性

倒排表的结构也导致其仅仅能处理有文档和关键词结构的数据;Pat 树和 Pat 数组将处理数据看成半无限串的叠加,所以不需要索引数据有任何结构。而Σ^2邻接矩阵模型和 ISTree 模型处理的数据既可以看作像倒排表一样具有文档-关键词结构,又可以像 Pat 树一样是无结构的字符串。因而适应能力最强。

4.5 动态性

目前全文检索模型的动态性能并不十分理想,利用 B^+树存储的倒排表能够实现动态操作,但是膨胀比会增大为1.14～1.15[8]。在 Pat 树或者 Pat 数组中能够实现数据的动态改变,但代价非常高,这是因为若原数据为 UVW,将其改变为 UZW 后(UVWZ 均为字符串),W 中半无限串的序号也将改变,所以整棵树将作很大的变动。我国研究人员提出过一种动态的 Pat 数组模型[16],但也是以索引空间的增大为代价的。Σ^2邻接矩阵模型的动态性能也不好,ISTree 模型可以很容易实现文档的添加和删除,对膨胀比的影响不是很大,对于文档的修改只能够通过先删除再添加来实现。

4.7 几种索引模型的性能比较

表1 几种全文索引模型的性能比较

	倒排表	Pat 数组	Pat 树	Σ^2	ISTree
查询时间复杂度	O(logN)+合并时间	O(M+logN)	O(M)	O(M−1)+合并时间	O(rf)(倒排表)
膨胀比(压缩前)	2	2	5～6	1	1
查询完备性	不完备	较完备	完备	不完备	完备
适应性	弱	中	中	强	强
动态性	强	弱	弱	弱	强
对原文依赖程度	生成原文依赖	生成原文查询依赖	生成原文查询依赖	不依赖	不依赖

5 关键技术研究

在全文检索中,有许多关键性技术问题,其解决将对检索

维普资讯 http://www.cqvip.com

效率的提高产生很大帮助。

1. 并行性：由于目前对检索速度的要求越来越高，单纯从模型的检索算法上来提高速度已经不能满足需要，这就迫切地要求索引模型提供并行处理机制。有两种并行，一种是检索请求间的并行，对每个检索请求串行处理，但是同时处理多个请求；另一种是请求内部并行。处理请求的过程一般包括搜索索引库和处理合并搜索结果。如果是署名文件还需要一个查询文档消除误匹配的过程，这三部分之间可以并行处理。索引模型的并行机制应该充分利用到这两种并行方式。

2. 相关度排序(Ranking)：很多时候，无需将查询结果全部返回给用户，而是对查询结果进行相关度排序，仅把相关度最高的 N 条记录返回。若能够在不检索全部文档的情况下就把相关度最高，或者近似最高的 N 条记录返回，则可使查询效率大大提高。实际上，我们可以预先设定关键词的权重，首先处理那些具有较大权重的关键词。

3. 对不同长度文档的处理：有些全文检索模型的性能会受到文档长度变化的影响，为了获得比较好的性能，我们可以有两种方法来解决这个问题：一、把需要索引的文档分类，相同长度的放在一起处理作为一个子索引库，查询的时候，分别查询所有子索引库，而后合并查询结果。二、将长文档切分为多个部分，作为多个短文档来索引。

结束语 全文检索技术的出现，导致了信息检索领域的一场革命。它不仅可以实现情报检索的绝大部分功能，而且还能直接根据数据资料的内容进行检索，实现了多角度、多侧面地综合利用信息资源。但是目前的全文检索模型并不完善，最广泛使用的倒排表模型膨胀比小，创建和检索速度都比较快，但要受到索引数据格式的限制，查询功能不完备。而 Pat 树和 Pat 数组能够解决倒排表的问题，但是膨胀比又很大，动态性能也很差。ISTree 兼顾了倒排表和 Pat 树的优点，但是对 ISTree 的研究还刚刚起步。所以，对 ISTree 模型的研究与完善应该成为我们今后研究工作的重点。

参 考 文 献

1 Zeng Haiquan, Shen Zhan, Hu Yunfa. Mining Sequence Pattern from Time Series Based on Inter-Relevant Successive Trees Model. In:Proc. of 9th. Intl. Conf. on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC'2003), LNCS/LNAI, Spring-Verlag, Chongqing, China, 2003

2 曾海泉，刘永丹，宋扬，胡运发. 基于互关联后继树的多时间序列关联模式挖掘. 计算机研究与发展，(已录用)2003,1

3 Knuth D E. The Art of Computer Programming, Sorting and Searching. 1st edition. Addision-Wesley Pub. Co., 1973

4 Weiner P. Linear pattern matching algorithm. In: Proc. 14th IEEE Symposium on Switching and Automata Theory, 1973. 1～11

5 Manber U, Myers E. Suffix arrays: A new method for on-line string searches. In: Proc. of the FISTREE Ann. ACM-SIAM Symp. on Discrete Algorithms, 1990. 319～327

6 Hu Yunfa, Zhou Shuigeng. A New Model of Chinese Full-text databases. In: Proc. World Multiconference on Systemics, Cybernetics and Informatics, Florida, USA, 2001. 528～533

7 Tao Xiaopeng, Hu Yunfa, Zhou Shuigeng. Subsequent Array: A New Full Text Index. In: Proc. World Multiconference on Systemics, Cybernetics and Informatics, Florida, USA, 2001. 551～556

8 胡运发. 互关联后继树—— 一种新型全文数据库数学模型：[技术报告]. 复旦大学计算机与信息技术系，2002. 3

9 曾海泉，宋扬，申展，胡运发. 基于互关联后继树的时间序列相似性查询. 计算机研究与发展，(已录用)2002. 12

10 杜莉. 基于互关联后继树的关联规则挖掘研究：[复旦大学硕士学位论文]. 2003. 5

11 Zobel J, Moffat A, Ramamohanarao K. Inverted files versus signature files for text indexing. Transactions on Database Systems, 1998,23(4):453～490

12 Grossi R, Vitter J S. Compressed suffix arrays and suffix trees with applications to text indexing and string matching (extended abstract). STOC 2000. 397～406, 1999

13 申展，王建会，吴爱华，胡运发. 互关联后继树模型——一种新颖的全文检索模型. 见：全国数据库学术会议，湖南长沙，2003

14 Moffat A, Zobel J. Self-Indexing Inverted Files for Fast Text Retrieval. ACM Transactions on Information System, 1996, 14(4):349～379

15 Gonnet G H, Baeza-Yates R. Lexicographical indices for text: Inverted files vs pat trees: [Technical Report TR-OED-91-01]. University of Waterloo, 1991

16 陶晓鹏. 面向(中文)全文数据库的全文索引的研究：[复旦大学博士学位论文]. 1999

17 Zobel J, Moffat A, Davis S. An Efficient Indexing Technique for Full-Text Database Systems. In: Proc. of the 18th Inter. Conf. on VLDB, 1992. 352～362

18 Golomb. Run-Length Encodings. IEEE Transactions on Information Theory, 1996, IT-12(3): 399～401

19 Faloutsos C. Access methods for text. Computing Survey, 1985, 17(1)

20 Roberts C. Partial-match retrieval via the method of superimposed code. Proceeding of the IEEE, 1979, 67(12):1624～1642

21 Christodoulakis S, Faloutsos C. Design Considerations for a Message File Server. TSE, 1984, 10(2):201～210

22 Farach M, Ferragina P, Muthukrishnan S. Overcoming the Memory Bottleneck in Suffix Tree Construction. In: Proc. of IEEE FOCS, 1998. 174～183

23 McCreight E. A space-economical suffix tree construction algorithm. Journal of the ACM, 1976, 23(2):262～272

24 Muthukrishnan S. Efficient Algorithms for Document Retrieval Problems. In: Proc. ACM-SIAM SODA, 2002. 657～666

25 Manber U, Myers G. Suffix arrays: a new method for on-line string searches. SIAM journal on Computing, 1993, 22(5):935～948

26 Boss B, Helmer S. Indexing a Fuzzy Database Using the Technique of Superimposed Coding - Cost Models and Measurements: [Technical Report TR-96-002]. Department for Mathematics and Computer Science, University of Mannheim, 1996

27 Gonnet G H, Baeza-Yates R A, Snider T. New Indices for Text: PAT Trees and PAT Arrays. In Information Retrieval: Data Structures And Algorithms, chapter 5, Prentice-Hall, 1992. 66～82

28 Baeza-Yates R, Ribeiro-Neto B. Modern Information Retrieval. Addison-Wesley, 1999

29 刘学文，陶晓鹏，于玉，胡运发. 一种全新的全文索引模型——后继数组模型. 软件学报，2002,(13)2:150～158

(上接第60页)

5 陈滢，徐宏炳，王能斌. 基于标记图的 Web 数据模型. 计算机学报，1999,22(3): 306～312

6 XML Path Language (XPath) 2.0. W3C Working Draft 02 May 2003. http://www.w3.org/TR/2003/WD-xpath20-20030502/

7 XQuery 1.0: An XML Query Language. W3C Working Draft 02 May 2003. http://www.w3.org/TR/2003/WD-xquery-20030502/

8 XQuery 1.0 and XPath 2.0 Data Model. W3C Working Draft 02 May 2003. http://www.w3.org/TR/2003/WD-xpath-datamodel-20030502/

9 王宁，徐宏炳，王能斌. 基于带根连通有向图的对象集成模型及代数. 软件学报，1998,9(12): 894～898

10 HTML Tidy. http://www.w3.org/MarkUp/#tidy