基于正则路径表达式的 XML 查询优化技术研究*)

陈继明[1,2]　鞠时光[1]　潘金贵[2]
(江苏大学计算机科学与通信工程学院　镇江 212013)[1]
(南京大学计算机软件新技术国家重点实验室　南京 210093)[2]

摘　要　支持正则路径表达式的查询技术，被认为是半结构化数据模式下的 XML 查询研究领域中一种颇具有研究价值的 XML 查询计算方法。本文对基于正则路径表达式的 XML 查询计算方法及其特点进行了分析，在此基础上详尽地介绍了目前所提出的各种查询优化技术，最后讨论了 XML 查询优化技术研究中存在的问题以及今后的研究方向。

关键词　半结构化数据，XML，正则路径表达式，查询优化

Research for XML Query Optimization Technology Based on Regular Path Expression

CHEN Ji-Ming[1,2]　JU Shi-Guang[1]　PAN Jin-Gui[2]
(School of Computer Science and Telecommunications Engineering, Jiangsu University, Zhenjiang 212013)[1]
(State Key Lab for Novel Software Technology, Nanjing University, Nanjing 210093)[2]

Abstract　Nowadays, the query techniques that support regular path expression gain wide attention in the research area of XML query in semi-structured data mode. Followed by analysis of the characteristic of regular path expression based XML query techniques, this paper introduces several existing query optimization techniques in details, discusses the problems which need to be improved and points out the future research at the end.

Keywords　Semi-structured data, XML, Regular path expression, Query optimization

1　引言

随着因特网技术及应用的发展，XML(extensible markup language)以其标准、简洁、结构严谨、可高度扩展的特点获得广泛的应用，并迅速成为因特网上数据表示和数据交换一种的新标准。XML 具有自我描述的特性，是一种半结构化数据，与传统数据库的结构化数据在数据模式上有所不同，不能直接使用传统的数据库查询技术进行查询计算，因此研究和探讨如何有效地查询 XML 数据变得尤为重要。

数据的查询技术与存储方式有着密切的关系。XML 数据的存储方式主要分为两类：第一类通过映射关系将 XML 存储在传统的数据库系统中，如关系型数据库或面向对象数据库；第二类存储方式则根据 XML 数据的特点，使用对象的概念来处理 XML 阶层性数据，如 NXD(Native XML Database)[1]。对于第一类存储方式，在查询时可直接采用传统数据库查询及其优化技术，但在存储时利用了指针或关联的方式来体现 XML 的数据阶层关系，因此在查询时也必须反复使用 join 的机制，来完成阶层性数据的查询对应。第二类更符合 XML 数据的存储特点，将 XML 数据看作一种半结构化数据，利用基于正则路径表达式的查询方式如 Xpath[6]、XML-QL[7]进行查询计算。当运用第二类查询方式，无论是在技术的使用以及效率的体现，其效果都更为突出，近来越来越受到研究者的关注。

2　XML 查询计算

2.1　XML 的描述及查询方式

XML 文件由描述文件结构的 DTD 和 XML 中所包含的数据两部分组成，由于其具有模式的可变动性，数据未赋予严格的类型等特点，被认为是一种具有良好结构的半结构化数据。目前对于半结构化数据模式的描述大多采用带标记的有向图，最典型的是 OEM(Object Exchange Model)图[2]。数据在 OEM 下使用带标记的有向图来表示，其中每一个对象都由一个标识和一个值构成。文[3]中描述了一种建立 XML 与 OEM 图之间映射关系及其相关规则的方法，即使用 OEM 图中的节点表示 XML 数据中的元素、子元素以及元素的属性，而它们之间的关系在 OEM 图中则使用带标签的边进行表达(如图 1 所示)。另外，文[4]中则提出一种基于 OEM 的 XML DTD 模式定义和形式描述的方法，从而实现 XML 与 OEM 图之间的"无缝"转换。

随着越来越多的信息用 XML 存储、交换和表示，智能地查询 XML 数据源的能力变得越来越重要。存在大量的关于 XML 查询技术的研究，提出了 Lorel[5]、Xpath[6]、XML-QL[7]等 XML 查询语言，由于 XML 文件具有阶层性和树状结构的特征，目前绝大多数 XML 查询语言都是基于正则路径表达式的查询方式。所谓正则路径表达式，是指能够以 Like-SQL 语言的方法表示查询的内容，但却可以结合一些特殊字元或一些特殊符号，来描述 XML 文件的阶层式及树状资料内容，如树状结构子孙关系的描述等。举例来说，基于正则路径表达式的查询 video·film·_*·name，描述的是查找 video 的下一代为 film，且 film 的任一子孙中有出现 name 的元素。目前正则路径表达式的应用极为广泛，不论在 XML

*)本课题得到国家自然科学基金项目(60473113)、国家自然科学基金重点项目(60533080)、江苏省高校自然科学指导性计划项目(05KJD520051)资助。陈继明　博士研究生，主要研究领域为 XML、分布式虚拟环境；鞠时光　教授，博士生导师，主要研究领域为数据库、XML；潘金贵　教授，博士生导师，主要研究领域为多媒体信息处理、多媒体远程教育系统。

应用系统的构建还是在查询语言的技术方面，大部分都支持正则路径表达式的应用。

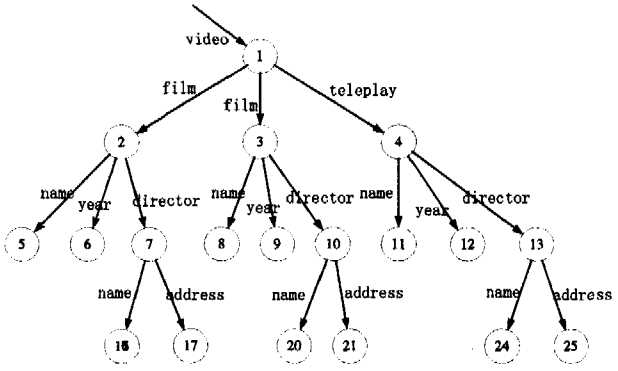

图 1　OEM 图

2.2　XML 查询计算方法

XML 查询是将 XML 数据描述成图状的半结构数据模型(OEM 图)，采用半结构化数据的特殊的查询机制，即基于正则路径表达式的查询机制进行查询计算。下面给出正则路径表达式的查询的定义。

定义 1　给定正则路径表达式查询 r 以及数据图 D，r 在数据图 D 上进行查询得到的结果为在图 D 中满足此路径表达式 r 的一组对象的集合。

正则路径表达式查询结果可以通过以下的方法进行计算[9]。首先，将正则路径表达式 r 转化成相应的非确定有限状态自动机 A_R 来表示。然后从非确定有限状态自动机 A_R 的初始状态到最终状态进行遍历，同时查找数据图 D 中的相关的节点，将当前自动机的状态和数据图中的相关节点一起存入集合中。最后从集合中取得最终状态所对应的节点，即为此正则路径表达式在数据图中查询的结果。算法 1 描述了查询计算的具体过程。

举例来说，假设给定数据图 D(如图 1 所示)及正则路径表达式 Video · film · director(name | address)，首先构建与此正则路径表达式相关的非确定有限状态自动机(如图 2 所示)，然后利用 Collection 集得到查询结果。表 1 描述了 Collection 集在算法 1 循环执行的变化情况。从表 1 中我们可以看到，经过 4 次循环后，Collection 集达到一个固定的值。由于状态 q12 是最终状态，因此最后得到计算结果为{16、17、20、21}的一组对象。

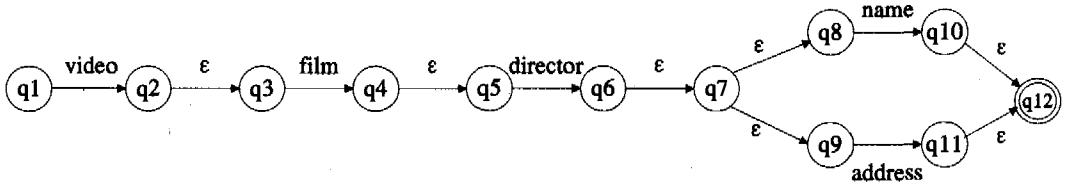

图 2　正则路径表达式对应的非确定有限状态自动机

表 1　Collection 集合变化的情况

Iteration	Collection
0	{ (root,q1) }
1	{ (root,q1),(1,q2),(1,q3) }
2	{ (root,q1),(1,q2),(1,q3),(2,q4),(3,q4),(2,q5),(3,q5) }
3	{ (root,q1),(1,q2),(1,q3),(2,q4),(3,q4),(2,q5),(3,q5),(7,q6),(10,q6),(7,q7),(10,q7)(7,q8),(10,q8),(7,q9),(10,q9) }
4	{ (root,q1),(1,q2),(1,q3),(2,q4),(3,q4),(2,q5),(3,q5),(7,q6),(10,q6),(7,q7),(10,q7)(7,q8),(10,q8),(7,q9),(10,q9),(16,q10),(20,q10),(16,q12),(20,q12),(17,q11),(20,q11),(17,q12),(21,q12) }

算法 1　正则路径表达查询的计算

输入：正则路径表达式 r 以及数据图 D；

输出：满足路径表达式 r 的数据图 D 中一组对象的集合

Procedure Query-Evaluation(r, D)

(1)　根据正则路径表达式 r 构建自动机 AR；

(2)　Collection＝{(a_1, s_1)}，其中{a_1, a_2, …}表示 D 中所有节点的集合，a_1 为根；s_1 为自动机 A_R 的初始状态；}

(3)　**WHILE** Collection 中的值不在变化 **DO**

(4)　　任选一对(a, s)∈ Collection；

(5)　　**IF** 存在($a \xrightarrow{R} a'$) in D **and** 存在($s \xrightarrow{R} s'$) in A_R **THEN**

(6)　　将(a', s')加入到 Collection 集合中；

(7)　　**END IF**

(8)　**END WHILE**

(9)　Result＝{a: (a, s)∈Collection，其中 s 为 A_R 的最终状态}；

(10)　return Result

值得注意的是，利用上述的方法进行正则路径表达式的查询计算时，使用了非确定有限状态自动机来实现，因此在查询时必须检索 XML 数据(OEM 图)中所有可能的路径，相应的 Collection 集合中将存储所有可能的组合，查询计算的空间也将会根据正则路径表达式对应的非确定有限状态自动机的状态数呈指数级增长。当计算处理较为复杂的正则路径表达式查询时，查询计算的空间会变得很大，导致查询速度的降低。如何提高查询效率，实现查询优化成了一个亟待解决的问题。

3　XML 查询优化技术

3.1　XML 索引技术

索引是数据库中重要的数据结构，它是提高查询效率一种重要方法。针对 XML 数据建立有效索引结构，能够快速且直接有效地改善查询检索时的效率，降低查询所花费的时间或花费的成本。

Lore[10]是 Stanford 大学的研究者提出的能有效存储和查询 XML 数据的系统，在系统中为了提高查询效率设计了

四种不同的索引结构，分别为 value index、text index、link index 和 path index。value index 是针对 PCDATA 为原子对象建立的，利用 PCDATA 作为搜索键值，建立对应的 B$^+$-tree 结构。text index 则是针对 PCDATA 为 text 对象建立的，利用 inverted list 结构，对 PCDATA 中较为重要的关键字作索引，并传回所对应的标签名称。link index 则是为了解决在 OEM 图中不支持逆向指针问题，利用 hashing 的方法来记录所有标签名称的父标签。最后，path index 则是记录了部分重要的路径结构及其在 OEM 图的查询结果的对应关系。用户在查询时，系统动态地选择所需要的索引结构，来完成查询。

Lore 只能针对简单路径进行查询计算，用户使用正则路径表达式进行查询时，系统选择索引通常需要同时使用多个索引结构，当正则路径表达式相当复杂时，将会增加查询的负担。文[8,11]基于上述索引结构的思想提出了一种针对正则路径表达式查询来设计索引机制 DataGuid，利用有限状态自动机的技术来解决正则路径表达式查询的问题，减少了使用正则路径表达式必须大量地检查文件中所有可能的路径，达到加快查询的目的。文[12]针对 DataGuide 只能支持单个正则路径表达式查询的局限性进行了改进，提出了一种索引机制 T-index。此索引机制通过所定义的路径模板(path template)，利用等价关系将 XML 数据包含的对象分成若干组，再通过构建有限状态自动机来实现，即通过定义路径模板来实现对复杂的查询建立索引。

还有一些 XML 查询索引技术，如文[14]提出利用 bit-vector 的思想建立索引 EBIM 技术，以及文[15]中描述针对大量数据建立索引结构 Index Fabric 等，但它们都不完全支持基于正则路径表达式的查询方式。

3.2 基于视图的查询重写技术

查询重写是数据库研究的一个基本问题，它和查询优化，数据仓库，信息集成，语义缓存等问题紧密相关[13]。基于视图的查询重写技术其目的是充分利用视图中的信息来对查询进行优化，提高查询的效率。下面给出基于视图的查询重写的定义：

定义 2 给定数据库 D 和在数据库上定义的视图集合 $V=\{V_1,V_2,\cdots,V_n\}$，对于数据库 D 的查询 q，如果存在查询 q'，其中 q' 至少查询了 V 中的一个视图，且 q' 的查询结果和 q 在数据库 D 中的查询结果一致，则 q' 是 q 的查询重写。

目前存在着大量的关于查询重写技术的研究，文[16,17]研究传统数据库的查询重写问题。对于半结构化数据和 XML 数据，文[18]中对基于 TSL 查询语言的重写问题进行研究，提出了通过完全遍历候选空间的方法来得到查询重写的方案。TSL 虽然具有结构的兼容性并且可以由多个路径表达式组成，但是它不支持正则路径表达式。本文着重讨论对于支持正则路径表达式的 XML 查询如何进行重写的问题。

3.2.1 完全查询重写技术

由于正则路径表达式具有极强的表达和重构能力，XML 查询及其视图都可以采用正则路径表达式来进行描述，XML 查询重写问题即转化为如何利用给定正则路径表达式 $\varepsilon=\{E_1,E_2,\cdots,E_n\}$ 来重写正则路径表达式 E_0[25]。Calvanese 在文[19]中针对上述问题提出一种优化的 2-EXPSPACE 的算法来计算查询重写，具体过程如算法 2 所示。另外，Calvanese 在文[20]中对正则路径表达式的语义进行了扩充，增加了 inverse 操作符，并采用了 two-way 的有限状态自动机技术来实现。

在算法 2 中，利用了自动机的转换来实现查询重写，最后将自动机 $R_{S,E}$ 转化为相应的正则路径表达式即为 E_0 的重写正则表达式。下面通过例子具体说明：给定查询 $E_0=$video·film*+video·teleplay·name 和一组视图 $\varepsilon=\{$video, film, teleplay·name$\}$，其中，re$(e_1)=$video，re$(e_2)=$film，re$(e_3)=$teleplay·name。图 3 为根据算法 2 所构建的自动机 A_d，A' 和 $\overline{A'}$。在构建自动机 A' 的过程中，A' 与 A_d 具有相同的状态集，A' 的初使状态为 A_d 的初使状态，A' 的最终状态为除了 A_d 的最终状态以外的其它所有状态。关于转移条件，A' 具有一条从 s_i 到 s_j 的转移 $e\in\sum_\varepsilon$，当且仅当 A_d 中存在一条从 s_i 到 s_j 的转移 $w\in L(re(e))$，其中 $w\in\sum$。由此我们可以得到 $L(R_{S,E})=e_1\cdot e_2^*+e_1\cdot e_3$。

算法 2 正则路径表达式完全查询重写

输入：正则路径表达式查询 E_0 和视图 $=\{E_1,E_2,\cdots,E_n\}$

输出：自动机 $R_{S,E}$

Procedure Query-TotalRewriting(E_0,ε)

(1) 建一个确定有限状态自动机 $A_d=(\sum,S,s_0,\delta,F)$ 使得 $L(A_d)=L(E_0)$；

(2) 构建自动机 $A'=(\sum_\varepsilon,S,s_0,\delta',S-F)$ 其中 $s_j\in\delta'(s_i,e)$ 当且仅当 $\exists w\in L(re(e))$ 即 $s_j\in\delta^*(s_i,w)$；

(3) Return $R_{S,E}=\overline{A'}$

3.2.2 局部查询重写技术

假设存在查询 Q，$re(Q)=R_1\cdots R_{100}$；以及视图 V_1 和 V_2，其中 $re(V_1)=R_1\cdots R_{49}$，$re(V_2)=R_{51}\cdots R_{100}$。若使用 Calvanes 算法，利用视图对查询 Q 进行完全查询重写，重写后得到查询 Q' 的值为空。但根据实际应用的情况，如果把查询 Q 重写为 $Q'=V_1R_{50}V_2$ 还是十分有用的。因此，Grahne 在文[21]中，提出了一种适合局部查询重写技术。具体过程如算法 3 所示。

算法 3 正则路径表达式局部查询重写

输入：正则路径表达式查询 E_0 和视图 $\varepsilon=\{E_1,E_2,\cdots,E_n\}$

输出：查询重写结果 E'_0

Procedure Query-PartialRewriting(E_0,ε)

(1) 计算查询 E_0 的补 E_0^c，即构建一个确定有限状态自动机 $A_d=(\Delta,S,s_0,\delta,F)$ 使得 $L(A_d)=L(E_0)$，$E_0^c=\overline{A_d}$；

(2) 构建局部查询重写的有限转换器 T，并计算转换结果 $T(E_0^c)$；

(3) 计算 $T(E_0^c)$ 的补 $(T(E_0^c))^c$，即 $(T(E_0^c))^c=\overline{T(E_0^c)}$；

(4) 将 $(T(E_0^c))^c$ 与 $M=((\Delta\cup\Omega)^*(E_1\cup\cdots E_n)(\Delta\cup\Omega)^*)^c$ 做并集运算，计算结果 E'_0。(E'_0 在字符集 Δ 上，ε 则在字符集 Ω 上)；

(5) return $E0'_0$。

这里，有限转换器 T 可表示为六元组，即 $T=(S,I,O,\delta,s,F)$，其中 S 为 T 状态的有限集合；I 为输入字符的有限集合；O 为输出字符的有限集合；s 为初使状态；F 为终止状态集；δ 为状态转移与输出结果关系的函数 $\delta\in S\times I^*\times S\times O^*$。假定对于有限状态自动机 $A_d=(\Delta,S,s_0,\delta,F)$，构建有限转换器 $T=(\Delta,S\cup\{s'_0\},\Gamma,\delta',s'_0,\{s'_0\})$，其中 $\Gamma=\Delta\cup\{\Psi\}$ (Ψ 为不在 Δ 上的字符)，即首先在自动机 A 的基础上，增加

一个新的状态 s'_0，并将其同时作为初使状态与终止状态；然后根据文[21]中给出的五组状态转移与输出结果的关系函数 δ' 分别进行转换，最后得到有限转换器 T。

在查询时，如果使用 $ans(Q,DB)$ 来表示 Q 在数据库 DB 上的查询，重写后的查询 Q' 则可表示为 $ans(Q',DB\cup V)$，其中 V 表示视图集。计算 $ans(Q',DB\cup V)$ 的过程实际上为将 Q' 在 DB 上查询的结果和 Q' 在 V 上查询结果进行笛卡尔积运算，即 $ans(Q',DB\cup V)(Q'\times DB)\cup(Q'\times V)$。因为视图集中每个视图的结果信息是已知的，即 Q' 在 V 上查询的结果可以直接得到，这样就可以减少从数据库 DB 中查询次数，从而达到提高查询效率的目的。在计算 Q' 查询结果时，存在两种极端的情况，一种可以完全查询重写，只需要对视图集 V 进行查询就可以得到查询结果，即 $ans(Q',DB\cup V)=ans(Q',V)$，此时执行效率最高；另一种 Q' 不包含视图集 V 中的信息，即 $ans(Q',DB\cup V)=ans(Q',DB)$，其执行效率相当于传统的查询算法。

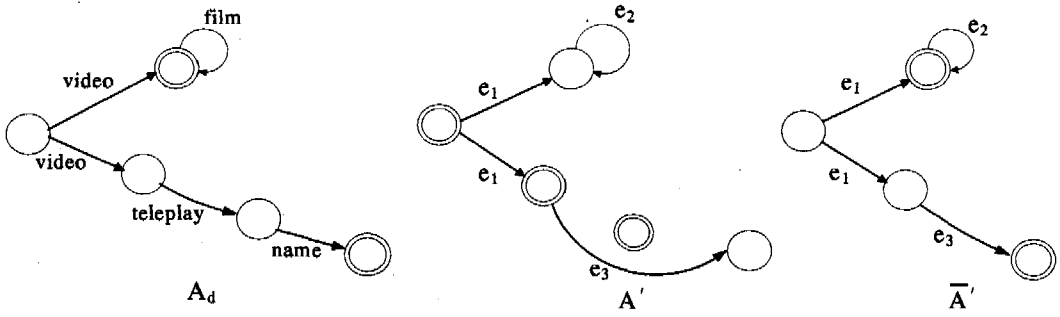

图 3　正则路径表达式查询重写

3.3　查询裁减技术

查询裁剪技术是一种利用数据存储的结构信息或相关的裁剪逻辑规则对查询进行裁剪的技术，从而实现降低查询的复杂度或者减少查询的空间，提高查询的效率。

文[22]提出了一种利用图模式来对基于正则路径表达式的 XML 查询进行裁剪的技术。通过分析图模式中包含的图的结构信息，建立一个 graph schemas。然后对查询 Q 裁剪，即利用 schemas S 的相关信息对 Q 进行重写操作，产生一个新的查询 Q'。

算法 4　正则路径表达式查询裁减

输入：正则路径表达式查询 Q 和 schemas S

输出：裁剪后的查询 Q'

Procedure Query-Pruning(Q,S)

(1) 构建一个非确定有限状态自动机 A，使得 $L(A)=L(Q)$；

(2) 根据 S 和 A，生成一个包含 $N\times P$ 个状态的自动机 $S\times A$，其中 $(s_1,a_1),(s_2,a_2),\cdots,(s_n,a_p)$ 表示状态；

(3) 从初始状态 (s_1,a_1) 到终止状态 (s_n,a_p)，检查每一条转移条件 $(s,a)\xrightarrow{P\wedge Q}(s',a')$，如对于 A 存在转移 $a\xrightarrow{Q}a'$ 但 S 中不存在 $s\xrightarrow{P}s'$，则 $P\wedge Q=$false；

(4) 对自动机 $S\times A$ 进行裁减处理，即对自动机中标签为 *false* 的边进行删除，得到新的自动机 $S\cap A$；

(5) $L(Q')=L(S\cap A)$；

(6) Return Q'。

在查询计算时，由于在重写 Q 的过程中，根据 schemas S 去除对查询结果无影响部分，裁剪后的查询 Q' 则仅需要对数据图的局部进行遍历，从而加快查询速度。算法 4 描述了根据 schemas S 对 Q 进行裁减的过程，值得注意的是，利用上述方法所得查询 Q' 与原查询 Q 在逻辑上是不相等的，但对于符合 schemas S 的 XML 数据库而言，Q 与 Q' 相等。

3.4　其他优化技术

除了上述的应用的较为广泛的优化技术外，S. Park 在文[23]中提出了一种基于识别标志的查询优化方法。此方法将 XML 所对应的 DOM 树的节点进行编码，并运用于查询过程，作为查询过滤的条件。在查询计算时，先通过编码的对比来判断是否需要对当前节点下的子节点继续访问，由此来实现减少不必要的查询过程，达到查询优化的目的。

另外，将 XML 查询转化为传统的数据库查询，直接使用较为完善的传统数据库查询优化技术来实现，也是解决 XML 查询优化问题的一种方法。如采用文[24]中将 XML 转换为关系数据数据库，则可直接使用传统数据库技术中的 B+-tree 或 Hash table 等方法进行优化处理。使用此方法，需要解决好 XML 数据库与传统数据库以及基于正则路径表达式的 XML 查询语言与 SQL 语法之间转换机制问题。

总结与展望　在半结构化数据模式下的 XML 查询技术研究领域中，支持正则路径表达式的查询技术或查询语言，被认为较有研究价值的 XML 查询方式。基于正则路径表达式的 XML 查询计算实际上就是在 XML 数据图上进行查询得到的结果满足此正则路径表达式的一组对象的集合。目前，主要采用将正则路径表达式转化相应的非确定有限状态自动机的方法进行查询计算，并利用索引技术、基于视图的查询重写以及查询裁剪等技术进行查询优化处理，提高查询效率。基于正则路径表达式的 XML 查询及其相关技术的目前正处于研究阶段，仍存在着一些问题需要进一步的探讨和完善，主要有以下几个方面：

(1)定义一种具有更强查询表达能力的，支持任意多个正则路径表达式联合进行查询的查询方式，如 $q(x)$：$y_0r_0z_0,\cdots,y_{n-1}r_{n-1}z_{n-1}$，其中 $\{y_0,z_0,\cdots y_{n-1},z_{n-1}\}$ 表示节点变量，$\{r_1,\cdots,r_{n-1}\}$ 表示正则路径表达式，提高对 XML 数据的查询能力。

(2) 随着 XML 文件的逐渐增加，用户进行查询时不再是对单一的文件进行查询，运用目前针对单个 XML 文件的查询优化技术，将会极大地降低查询过程效率，需要提出适合多份 XML 文件查询的相关技术方法。

(3) 在因特网上存在着海量的半结构化数据，在信息集成中也产生了大量的半结构化视图，有效视图的快速检索技术将会进一步提高查询优化技术的效率。

参 考 文 献

1　Schoning H. Tamino: A DBMS designed for XML. In: Proc. of

ICDE Conference, 2001
2 Abiteboul S. Querying Semistructured data. In: Proceedings of the International Conference on Database Theory, 1997
3 Suciu D. Semi-structured data and XML. In: Proceedings of International Conference on Foundations of Data Organization, 1998
4 聂培尧,李战怀,等. 基于 OEM 的 XML 半结构数据的模式描述方法. 计算机工程与设计, 2003,24(1)
5 Abiteboul S, Quass D, McHugh J, et al. The Lorel Query Language for Semistructured Data. International Journal on Digital Libraries, 1997,11(1)
6 Clark J. XML Path Language. http://www.w3.org/TR/xpath
7 Deutsch A, Fernandez M, Suciu D. Storing Semistructed Data with STORED. In: Proceedings of the ACM SIGMOD International Conference on the Management of Data, 1999
8 Nestorov S, Ullman J, Wiener J, et al. Representative Objects: Concise Representations of Semistructured Hierarchical Data. In: IEEE International Conference on Data Engineering, 1997
9 Abiteboul S, Buneman P, Suciu D. Data on the Web: From Relations to Semistructured Data and Xml. Morgan Kaufmann, San Francisco, 1999
10 Mchugh J, Widom J, Abiteboul S, et al. Indexing Semistructured Data: [Technical Report]. Stanford University Computer Science Department, 1998
11 Goldman R, Widom J. DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases. In: Proc. of the 23rd VLDB Conference, 1997
12 Milo T, Suciu D. Index Structures for Path Expressions. In: Proc. of the 7th International Conference on Database Theory, 1999
13 高军,唐世渭,等. 半结构化数据查询重写. 计算机研究与发展, 2002, 39(2)
14 Tseng V, Lin W. A New Method for Indexing XML Documents. In: Proc. of the 12th Workshop on Object-Oriented Technology and Applications, 2001
15 Cooper F, Sample N, Franklin M, et al. A Fast Index for Semistructured Data. In VLDB, September 2001
16 Grahne G, Mendelzon A O. Tableau Techniques for Querying Information Sources Through Global Schemas. The MIT Press, 1999
17 Pottinger R, Levy A. A Scalable Algorithm for Answering Queries Using Views. In: Proc. of VLDB Journal, 2001
18 Papakonstantinou Y, Vassalos A. Query Rewriting Using Semistructured Views. In: Proceedings of the ACM SIGMOD International Conference on the Management of Data, 1999
19 Calvanese D, De Giacomo G, Lenzerini M, et al. Rewriting of Regular Expressions and Regular Path Queries. In: Proc. of PODS'99, 1999
20 Calvanese D, De Giacomo G, Lenzerini M, et al. Containment of Conjunctive Regular Path Queries with Inverse. In: Proc. of KR 2000, 2000
21 Grahne G, Thomo A. Algebraic Rewritings for Optimizing Regular Path Queries. Theoretical Computer Science, 2003,11(6)
22 Fernandez M, Suciu D. Optimizing Regular Path Expressions Using Graph Schemas. In: IEEE International Conference on Data Engineering, 1998
23 Park S, Kim H J. SigDAQ: An Enhanced XML Query Optimization Technique, Journal of systems and software, 2002,61(2)
24 Florescu D, Kossmann D. Storing and Querying XML Data Using an RDBMS. IEEE Data Engineering Bulletin, 1999
25 Calvanese D, De Giacomo G, Lenzerini M, et al. View-based Query Processing and Constraint Satisfaction. In: Proc. of LICS 2000, 2000

(上接第 140 页)

从实验中可以看出如下几种选取查询词的规则:

1)比较普通的关键字已经可以带来很好的搜索结果:这是由此算法的特殊性所决定的。因为算法中的 Meta-Search 部分主要起一个初步筛选的过程,得到较多相关结果为此部分的主要目的,所以普通的关键字更能得到多的结果。

2)多种关键字的组合可以得到较好的结果:这是因为可以更加详细的描述所要查询的相关网站,所以可以得到更加多的相关网站数量。

3)整个算法对查询词的依赖性不是特别强烈。只要选择比较具有普遍领域知识的查询词,都可以得到比较好的结果。

4)在实际应用中,由于每个网站只检查一次,结果中重复的网站不会对算法造成影响,因此可以采用多种关键字的组合来进行多次的搜索,网站的检查列表采用每组关键字检索结果的并集,会得到比较好的结果。

4.4 实验总结

从整个实验可以看出,算法的结果还是比较好的。搜索"网上书店"关键字得到的 627 个网站中发现 233 个相关的提供网上购书服务的网站,准确率达到 95.28%,而平均抓取代价只有 31 个网页,总共访问了 19716 个网页。这个抓取代价是非常小的,因为一般情况下一个普通的购书网站的网页量就可以达到几万甚至几十万的数量级。用普通的 Crawler 在普通的 PC 机上一小时即可抓取所需要的网页,完成整个实验。

表 3

站点名	Alex 排名	相关网站数目	准确率(%)
www.hao123.com	64	22	91.667
www.265.com	91	65	84.415
www.mscncn.com	55065	86	68.254
我的算法	NA	222	95.27

而相对应的导航型站点,由于是人工整理,其结果的数量以及准确率都不高,以下是对一些导航型站点的数据。表 3 是一些比较数据(以下实验的实验日期是 2006 年 3 月 1 日,Alex 的排名和网站的情况都以当日的情况为准)。

从中可以看出,导航型站点提供的相关网站由于是人工整理的原因,其数量和准确率都不是很好。而本文的算法无论是在准确率还是在数量上都达到了较好的效果。

总结 本文介绍了一种抓取领域相关型网站的小耗费算法,主要阐述了此算法的具体过程以及利用此算法进行的实验。从实验中可以看出,算法在非常小的代价下取得了很好的结果,发展了传统 Focused Crawler 的算法,无论是相关网站发现的数量或是质量上均大大超过了现在的人工收集方法。

未来进一步的工作是算法如何在不同领域中自动生成领域相关的网页评价规则和查询词,而不是由人为设计。这将是一项长期而有意义的工作。

参 考 文 献

1 Chakrabarti S, van den Berg M, Dom B. Focused crawling: a new approach to topic-specific Webresource discovery. In: Proc. of the 8th International World Wide Web Conference, Toronto, Canada, 1990
2 周立柱,林玲. 聚焦爬虫技术研究综述. 计算机应用,2005(9)
3 Qin Jialun, Zhou Yilu, Chau M. Building Domain-Specific Web Collections for Scientific Digital Libraries: A Meta-Search Enhanced Focused Crawling Method. In: JCDL'04, 2004
4 Bergmark D, Lagoze C, Sbityakov A. Focused Crawls, Tunneling, and Digital Libraries. In: Proc. of the 6th European Conference on Digital Libraries, Rome, Italy, 2002
5 Kumar R, Raghavn P, Rajagopalan S, et al. Extracting Large-Scale Knowledge Bases from the Web. In: Proc. of the 25th International Conference on Very Large Data Bases Conference, Edinburgh, Scotland, UK, 1999
6 Kumar R, Raghavan P, Rajagopalan S, et al. Trawlling the Web for Emerging Cyber-Communities. In: Proc. of 8th International World Wide Web Conference, Toronto, Canada. Machine Learning Techniques, Proc. AAAI-99 Spring Symposium on Intelligent Agents in Cyberspace, 1999
7 Toyoda M, Kitsuregawa M. Creating a Web Community Chart for Navigating Related Communities, In: Proc. 8th WWW Conference, 1999
8 Cho J, Garciamolina H, Page L. Efficient crawling through URL ordering [A]. In: Proceedings of the Seventh International Conference onWorldWideWeb [C], April 1998
9 Ehr I, Maedche A. Ontology2focused crawling of Web docu ments [A]. In: Proceedings of the 2003 ACM symposium on App lied omputing [C], March 2003
10 韩近强,赵静,杨冬青,唐世渭. 基于领域知识的网页筛选系统. 见:第 19 界全国数据库学术会议论文集,计算机科学,2002(8)