文章编号:1007-2985(2011)03-0036-03

一种改进的 SLCA 语义查询算法*

曾琳玲

(吉首大学张家界学院,湖南 张家界 427000)

摘 要:在 XML 关键字查询处理中,寻找最小最低公共祖先 SLCA(Smallest Lowest Common Ancestor)是一个重要问题.分析了基于 SLCA 语义查询处理算法的特点,在关键字和 Dewey 编码的倒排索引的基础上,利用栈结构实现了 SLCA 算法.

关键词:XML 关键字查询;SLCA;栈;算法

中图分类号:TP311 **文献标志码**:A

XML 已经成为 Web 上数据交换的标准,Web 上海量数据以 XML 的形式进行保存和传输,如何高效准确查询 XML 数据是目前的一个研究热点问题. XQuery[1] 和 XPath[2] 是 W3C 推出的标准 XML 查询语言,它们能够进行复杂精确的查询,但是这 2 种查询语言都是基于路径的查询,要求用户详细了解 XML 文档的数据模式信息并掌握复杂的查询语法,相对于关系模型的结构化查询语言 SQL 来说要复杂很多,普通用户掌握这种查询语言难度非常大,影响了 XQuery 和 XPath 的推广与应用.而基于关键字的 XML 文档查询只需要用户输入查询的关键字不需要了解模式和路径信息就能够完成查询操作,所以基于关键字的 XML 查询具有非常大的应用空间.

与传统的信息检索中的关键字查询不同,针对 XML 文档的关键字查询目标不是整个 XML 文档,而是用户感兴趣的满足关键字条件的 XML 文档片段,所以根据查询关键字确定文档中返回的片段是 XML 关键字查询的基本问题.针对这个问题,目前主要有基于树模型的 XML 关键字查询[3]、基于图模型的 XML 关键字查询[4]和基于 XML 数据流的关键字查询[5]这 3 种类型.其中基于树模型的 XML 关键字查询是基于 Dewey 编码[6]的应用最广泛的查询方式之一.笔者介绍了树模型关键字查询的基本概念,分析了存在的 SLCA 查询处理的特点,提出了改进的 SLCA 算法.

1 基本概念

定义 1 XML 树模型.一个 XML 文档是一个 4 元组 $D(r,V,E)$,r 表示树的根节点,V 表示节点集合,E 表示边的集合.如图 1 所示,树中的节点表示 XML 文档中的属性、元素或者文本值,有向边表示节点之间的关系.图 1 采用 Dewey 编码来唯一标识每个节点.

Dewey 编码.根节点的编码为 1,给定一个节点 v 的 k 个子节点 $u_1,u_2,\cdots,u_k$,u_i 的 Dewey 编码是 $L(u_i)=L(v).i$.对于 Dewey 编码,对于任意 2 个节点 u 和 v,当且仅当 $L(u)$ 是 $L(v)$ 的前缀,则 u 是 v 的祖先节点;当且仅当 $L(u)$ 是 $L(v)$ 的前缀且 $L(u)$ 比 $L(v)$ 多且只多 1 个“.”连接符,则 u 是 v 的父节点.

定义 2 关键字匹配集合.对于给定的 XML 文档 D 和关键字 k,用符号 KMS(k) 表示 D 中所有匹配关键字 k 的节点集合,KMS(k) $=\{v \mid v\in V, k=\text{tag}(v)$ 或者 $k=\text{val}(v)\}$.其中 tag(v) 表示非叶节点标签,val(v) 表示叶节点的值.如图 1 所示,KMS(XML) $=\{1.1.2.1.1,1.2.2.1.1.1\}$,KMS(Java) $=\{1.2.2.3.1.1\}$,KMS(papers) $=\{1.1.2,1.2.2\}$.

定义 3 最低公共祖先 LCA(Lowest Common Ancestor).给定 m 个节点 $n_1,n_2,\cdots,n_m$,节点 v 是它们的最低公共祖先,当且仅当:(ⅰ)v 是 $n_i(1\leqslant i\leqslant m)$ 的祖先节点;(ⅱ)不存在节点 u,u 是 v 的后代,并且 u 是 $n_i(1\leqslant i\leqslant m)$ 的祖先节点,记作 $v=\text{LCA}(n_1,n_2,\cdots,n_m)$.文献[1]认为基于 LCA 语义的关键字查询片段是一个较好结果.如图 1 所示,LCA(Mike(1.1.1.1),XML(1.1.2.1.1.1),DB(1.1.2.3.1.1)) = author(1.1).

定义 4 最小最低公共祖先 SLCA(Smallest Lowest Common Ancestor).给定节点集合 $S_1,S_2,S_3,\cdots,S_n$,如果存在 $v_1\in S_1,v_2\in S_2,\cdots,v_n\in S_n$,使得 $v=\text{LCA}(v_1,v_2,\cdots,v_n)$,那么记作 $v\in\text{LCA}(S_1,S_2,S_3,\cdots,S_n)$.满足如下条件则成为 SLCA

* 收稿日期:2011-03-02

作者简介:曾琳玲(1983-),女(土家族),湖南吉首人,吉首大学张家界学院教师,主要从事计算机科学与技术研究.

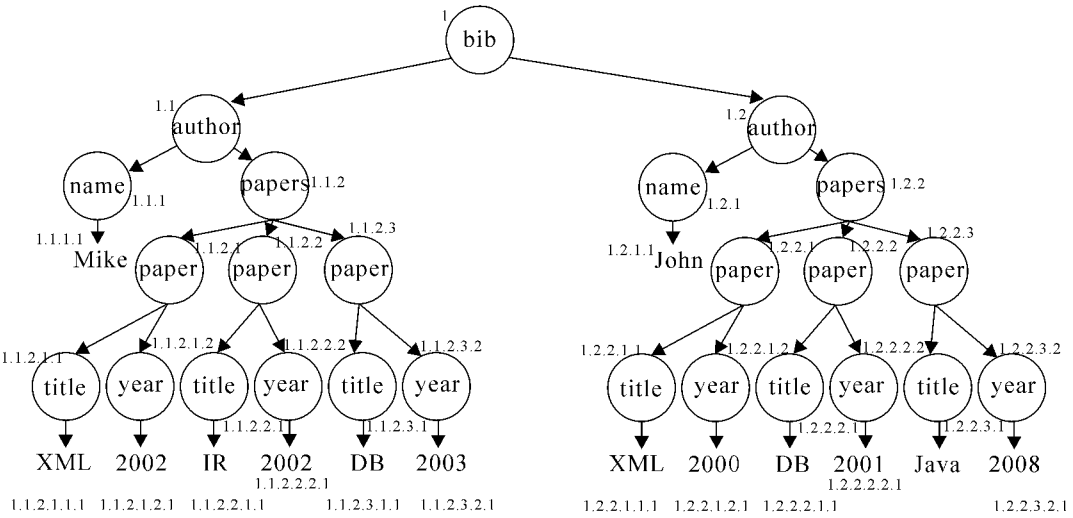

图 1 带有 Dewey 编码的 XML 文档

节点:若一个节点 $v \in \mathrm{LCA}(S_1, S_2, S_3, \cdots, S_n)$**,则对任意的** $u \in \mathrm{LCA}(S_1, S_2, S_3, \cdots, S_n)$**,**$v$**不是**$u$**的祖先节点,那么节点**$v$**称为** $S_1, S_2, S_3, \cdots, S_n$ **的 SLCA 节点,记作** $v \in \mathrm{SLCA}(S_1, S_2, S_3, \cdots, S_n)$.

SLCA **的基本思想是,给定关键字查询** $Q=\{k_1, k_2, \cdots, k_m\}$**,如果节点** $v \in \mathrm{SLCA}(\mathrm{KMS}(k_1), \mathrm{KMS}(k_2), \cdots, \mathrm{KMS}(k_m))$**,那么**$v$**是查询**$Q$**的一个解. 也即是将** $\mathrm{LCA}(\mathrm{KMS}(k_1), \mathrm{KMS}(k_2), \cdots, \mathrm{KMS}(k_m))$ **中存在祖先-后代关系的祖先节点删除的结果. 如图 1 所示,对于给定的关键字** KMS(Mike) **和** KMS(XML) **的查询,**$\mathrm{LCA}\{\mathrm{KMS}(\mathrm{Mike}), \mathrm{KMS}(\mathrm{XML})\} = \mathrm{author}(1.1)$ **或者** $\mathrm{LCA}\{\mathrm{KMS}(\mathrm{Mike}), \mathrm{KMS}(\mathrm{XML})\} = \mathrm{bib}(1)$**,但是** bib(1) **是** author(1.1) **的祖先节点,所以** $\mathrm{SLCA}\{\mathrm{KMS}(\mathrm{Mike}), \mathrm{KMS}(\mathrm{XML})\} = \mathrm{author}(1.1)$.

2　改进 SLCA 算法

2.1 改进 SLCA 算法的思想

根据定义 3,求出 LCA **可以根据** Dewey **编码的特点计算最长公共前缀表示的节点就是这些节点的** LCA. **在一个** XML **文档上给定关键字查询** $Q=\{k_1, k_2, \cdots, k_m\}$**,对于每个查询关键字** $k_i (1 \leqslant i \leqslant m)$**,从倒排索引中找到对应的关键字匹配集合** $\mathrm{KMS}(k_i)$**,对所有节点组合**$\{v_1, v_2, \cdots, v_m\} (v_i \in \mathrm{KMS}(k_i))$**,计算** $\mathrm{LCA}(v_1, v_2, \cdots, v_m)$.

对于计算 SLCA**,可以采用基于栈的进行处理. 栈的设计如下:栈中每个元素由**(id,keywords)**对组成. 假设一个从栈低到栈顶元素** en **的** id **分别为** $\mathrm{id}_1, \mathrm{id}_2, \cdots, \mathrm{id}_m$**,那么元素** en **代表了** Dewey **编码为** $\mathrm{id}_1, \mathrm{id}_2, \cdots, \mathrm{id}_m$ **的节点.** Keyword **是长度为** k **的布尔值数组,**k **为查询关键字的个数. 栈中元素的** keywords$[i]=$T **表示以该元素表示的节点的子树包含关键字** k_i**. 例如图** 2-a **中的栈顶元素表示了节点** 1.1.2.1.1.1**,这个节点包含了关键字** XML.

	XML	2000
1	T	F
1	F	F
1	F	F
2	F	F
1	F	F
1	F	F

a 节点1.1.2.1.1.1

	XML	2000
1	F	T
2	F	F
2	F	F
2	T	F
1	F	F
1	F	F

b 节点1.1.2.2.2.1

	XML	2000
1	T	F
1	F	F
1	F	F
2	F	F
2	F	F
1	F	F

c 节点1.2.2.1.1.1

	XML	2000
1	F	T
2	F	F
1	T	F
2	F	F
2	F	F
1	F	F

d 节点1.2.2.1.2.1

图 2　执行 $Q=\{$**XML,2 000**$\}$**的栈状态转换**

基于以上栈结构,给出 SLCA **算法的思想:每次取出编码最小节点与栈顶元素求** LCA**,将栈中不是** LCA **的元素出栈,并将它包含关键字的情况记录在其父节点中. 对于每个出栈的元素,如果它包含了所有关键字,那么它就是** SLCA **节点,将它返回,并将栈中所有元素包含关键字情况置为** F. **如果它不包含所有关键字,那么更新它包含关键字的情况,将不是** LCA **部分入栈.**

2.2 改进的 SLCA 算法

算法 1　基于栈的 SLCA **语义查询处理算法.**

```
Input:查询关键字 Q={k1,k2,…,km},XML 文档 D;
Output:SLCA 节点集合;
将栈 S 置为空;//S 是一个全局的栈结构
While (关键字匹配集合中还没有访问到的节点) Do
{ v:=编码最小的节点;//v 包含关键字 ki,1≤i≤m
  p=LCA(S,v);//计算 S 栈顶元素和 v 的编码
  最长公共前缀
While(栈 S 的长度大于 p 的长度) Do
  { e:=栈顶元素出栈;
    If(e 中 keywords 数组中值全为 True) Then
    { 返回 e 为 SLCA;
      将 S 中所有元素 keywords 数组的值置
      为 False;}
    Else
    { For(j:=1→m) Do
      If(e.keywords[j]==True) Then
        栈顶元素 keywords[j]:=True;}
  }
  For(p<j≤v.length) Do
  将 v[j]至 v[v.length]入栈;
  栈顶元素 keywords[i]:=True;
  从栈顶开始扫描栈,找到 keywords[i]都是
  True 的节点,返回 SLCA 节点;
}
检查栈中元素,返回 SLCA 节点;
```

对于图 1 中的查询 $Q=\{XML,2\ 000\}$,首先从倒排表中取得每个关键字对应的匹配集合,"XML"和"2 000"对应的节点分别是{1.1.2.1.1.1,1.2.2.1.1.1}和{1.1.2.2.2.2.1,1.2.2.1.2.1},将最小节点1.1.2.1.1.1放入栈中,如图 2-b 所示.将次小节点 1.1.2.2.2.1 与栈中元素计算 LCA,得到 1.1.2,将栈中不是 1.1.2 的元素出栈,并将包含关键字 XML 的情况传递到元素 1.1.2.将 1.1.2.2.2.1 中不是 1.1.2 的部分入栈,结果如图 2-b 所示.图 2 中每个子图都代表了处理完它的下面表示的节点后的结果.最后可以得到 2 个 SLCA,分别是 papers(1.1.2)和 paper(1.2.2.1).

3 结语

介绍了基于树模型的 XML 文档关键字检索的有关概念,根据 Dewey 编码计算节点结构关系的特点,结合 SLCA 算法的基本思想,建立了(keywords,Dewey 编码)的倒排索引结构可以求出 KMS(k_i),然后利用(id,keywords)的栈结构高效计算出节点集合的 SLCA.

参考文献:

[1] World Wide Web Consortium. XQuery 1.0:An XML Query Language (Second Edition) [EB/OL]. http://www.w3.org/TR/2010/REC-xquery-20101214/,2010-12-14.

[2] World Wide Web Consortium. XML Path Language (XPath) 2.0 (Second Edition) [EB/OL]. http://www.w3.org/TR/xpath20/,2010-12-14.

[3] HRISTIDIS V,KOUDAS N,PAPAKONSTANINOU Y,et al. Keyword Proximity Search in XML Trees [J]. IEEE Transactions on Knowledge and Data Engineering,2006,18(4):525-539.

[4] CONG Y,JAGADISH H V. Querying Complex Structured Database [C]//Proceedings of the 33rd International Conference on Very Large Data Bases. Vienna,2007:1 010-1 021.

[5] 周军峰,孟小峰,张 新,等. XML 数据流上基于关键字的多查询处理 [J]. 计算机研究与发展,2007,44(增刊):392-397.

[6] TATARINOV S,VIGLAS D,BEYER K J,et al. Storing and Querying Ordered XML Using a Relational Database System [C]//Proc. of the ACM SIGMOD 2002. Los Alamitos,CA:IEEE Computer Society,2002:204-215.

Improved Algorithm on SLCA Querying

ZENG Lin-ling

(Zhangjiajie College of Jishou University,Zhangjiajie 427000,Hunan China)

Abstract:It is a important problem to find SLCA(Smallest Lowest Common Ancestor)on information retrieves on XML keywords. Through analyzing the semantics characteristics of LSCA,an algorithm on SLCA is implemented by stack on basis of inverted index on keywords and Dewey labeling scheme.

Key words:information retrieves on XML keywords;SLCA;stack;algorithm

(责任编辑 向阳洁)