ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.6, June 2007, pp.1400−1418
DOI: 10.1360/jos181400

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

XML 数据的查询技术*

孔令波[1+], 唐世渭[1,2], 杨冬青[1], 王腾蛟[1], 高 军[1]

[1](北京大学 计算机科学技术系,北京 100871)

[2](北京大学 视觉与听觉信息处理国家重点实验室,北京 100871)

Querying Techniques for XML Data

KONG Ling-Bo[1+], TANG Shi-Wei[1,2], YANG Dong-Qing[1], WANG Teng-Jiao[1], GAO Jun[1]

[1](Department of Computer Science and Technology, Peking University, Beijing 100871, China)

[2](National Laboratory on Machine Perception, Peking University, Beijing 100871, China)

+ Corresponding author: Phn: +86-10-62755440, E-mail: lbkong@126.com, http://www.pku.edu.cn/lbkong

Kong LB, Tang SW, Yang DQ, Wang TJ, Gao J. Querying techniques for XML data. *Journal of Software*, 2007,18(6):1400−1418. http://www.jos.org.cn/1000-9825/18/1400.htm

Abstract: XML has become the de facto standard for data representation and exchange for Web applications, such as digital library, Web service, and electronic business. How to retrieve interesting information from the promising XML data is an active research area. Among techniques in this area, the description of query patterns is a crucial section. This paper reviews the actualities of recent researches on this topic. It classifies the query descriptors into two categories, XML Query type and XML IR type (with three subcategories: XML IR/keyword, XML IR/fragment and XML IR/query), and concludes three popular problems: Twig pattern processing, SLCA (smallest lowest common ancestor) problem, and similarity measuring techniques for retrieved XML fragments. It analyzes the virtue and deficiency of related techniques based on their convenience for common users. And hereby it proposes four issues for further XML querying researches: structural keywords and corresponding structural similarity measuring, wiping off the redundancy in XML data processing between XML Query (including XML IR/query) and XML IR/keyword, theoretical discussion of XML Query and its realization, and the management of peculiar XML data.

Key words: XML query; XML IR; XPath; XQuery; XML keyword search; XQuery FT; Twig; structural join; SLCA(smallest lowest common ancestor); dewey encoding; similarity measuring; tree edit distance; VSM; TF*IDF

摘 要: XML 规范已成为当前网络应用(包括数字图书馆、Web 服务以及电子商务)中事实上的数据表达、交换的标准.针对 XML 数据的查询在当前 XML 数据管理研究中占有重要的地位,也是当前 XML 数据处理研究领域的热点方向,相关的研究文献有很多.根据查询模式描述的不同,将当前 XML 查询技术归入两大类:XML Query 方式和

* Supported by the National Natural Science Foundation of China under Grant No.60503037 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2005AA4Z3070 (国家高技术研究发展计划(863)); the Beijing Natural Science Found of China under Grant No.4062018 (北京市自然科学基金)

Received 2006-04-25; Accepted 2007-01-23

XML IR 方式.后者又进而可分以为 3 个子类:XML IR/keyword 方式、XML IR/fragment 和 XML IR/query 方式,并从中挑选出 3 个研究者关注的问题进行了简述,它们是:Twig 查询模式的处理、SLCA(smallest lowest common ancestor)节点的获取以及对所获取的 XML 片段相似性的度量.以方便普通用户使用为准则探讨了相关 XML 查询技术的优、缺点,将如下 4 个问题作为需要进一步关注的研究内容:结构化关键字查询及相应的结构相似性度量方法,如何消除 XML Query 查询处理模式(包含 XML IR/query)和 XML IR/keyword 查询处理模式间数据冗余的问题,XML Query 查询方式的理论探讨及其实现以及针对特定应用的 XML 数据的有效管理.

关键词: XML 查询;XML IR 查询;XPath;XQuery;XML 关键字查询;XQuery FT;Twig 查询模式;结构连接; SLCA 节点;Dewey 编码;相似性度量;树编辑距离;向量空间模型;TF*IDF

中图法分类号: TP311 文献标识码: A

XML(extensible markup language)(最新的规范为 2004 年的 XML1.1),即可扩展的标记语言,是一套定义语义标记的规范,其目标是能够定义计算机和人都方便识别的数据类型.随着网络应用的快速发展,符合 XML 规范的数据(称为 XML 数据)已大量存在于当前的信息社会,尤其是电子商务、Web 服务、数字图书馆等应用理念的进一步发展,使得 XML 类型的数据成为当前主流的数据形式.对 XML 数据的有效管理也随之成为当前数据库领域研究的热点[1].

在规范了 XML 数据格式后,如何方便地从 XML 数据中提取用户感兴趣的信息就成为 XML 数据管理研究的重要主题,内容涉及查询请求模式的描述、查询语句的执行机制以及查询结果的显示.对这类问题的探讨已成为数据库研究领域的一个方向,到目前为止,在各高等级数据库会议上涌现出了数目繁多的研究论文(SIGMOD,VLDB,SIGIR,ICDE,ICDT 等).本文的目的就是尝试对当前与该专题有关的研究成果进行汇总.

本文第 1 节在简单回顾 XML 数据的特点后,根据描述查询请求的不同将当前 XML 数据的查询技术分为两大类:XML Query 方式和 XML IR 方式,并将后者进而分为 3 个子类:XML IR/query,XML IR/fragment 和 XML IR/keyword;之后进一步给出不同类别下查询模式的概括.第 2 节以 Twig 查询模式和 XML 关键字查询(SLCA(extensible markup language)问题)为线索,讲述 XML 查询语句执行的研究状况.第 3 节概括 XML 查询结果显示的问题,主要集中于满足给定关键字的 XML 片段的相似性度量问题.最后是总结和研究展望,根据是否便于普通用户使用的原则,提出舍弃标签信息但保留结构关系的新型查询模式描述的建议以及新型的 XML 片段结构相似性求解方法;并根据最近的研究文献归结出另外两个值得关注的研究方向,即 XML Query 查询方式的理论探讨及其实现,以及针对特定应用的 XML 数据的有效管理.

本文讨论的问题只是 XML 数据管理领域中的一部分,观点也可能存在偏颇之处,但我们希望通过本文的工作,能给数据库研究者,尤其是正在进入相关研究领域的人员一些启发和帮助.需要指出的是,若在一篇文章中将有关 XML 数据查询的研究内容,即从查询模式的分类到查询技术的实现全部覆盖是一件非常困难的事情,而且文献[2]概括的 XML 索引技术已经很好地概括了 XML 查询实现方面的研究内容.所以,本文在叙述上如下安排:内容着重于查询特征的概括,同时兼顾自文献[2]后出现的新的 XML 查询实现技术.例如,将 XML 查询根据其查询模式的不同分作两大类——XML Query 和 XML IR,然后概括叙述两种查询模式的研究点;在阐述研究点的过程中,对新出现的查询实现技术作简要的介绍.

1 XML 查询模式分类

1.1 XML数据、模型及其结构信息

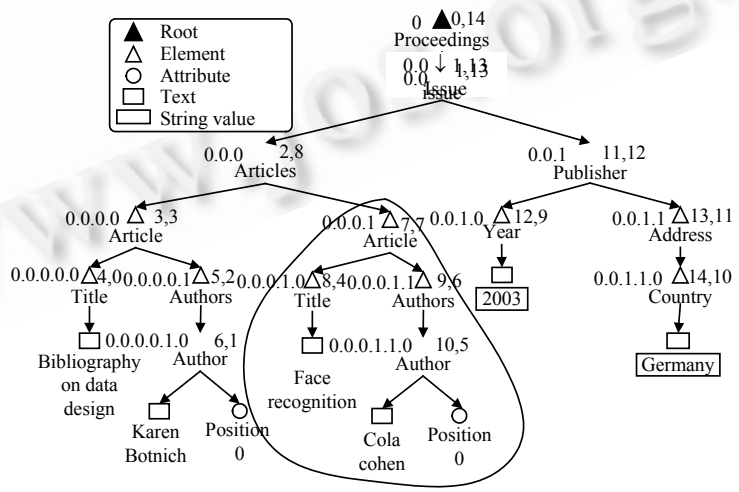
符合 XML 规范的数据称作 XML 数据.这类数据有两个基本特点:一是自描述,XML 数据本身就已经包含了元数据——关于数据本身的信息,表现为不同语义的标记(例如元素、属性等等).在所有标记中,元素标记最为重要.一个元素标记由两个起、止标签构成,起止标签所含的文本就是对应的语义单元;二是半结构化,即不同于传统关系数据库(传统的数据库都有一定的数据模型,可以根据模型来具体描述特定的数据),XML 数据的结构

限制并不严格,表现为语义单元相互嵌套的层次关系.XML 数据的基本形式为 XML 文档,如图 1(a)所示即为一个实际的 XML 文档.在实际处理 XML 数据时,更为常见的是 XML 标签有向图模型,由 XPath 规范描述.通常简化为 XML 标签有向树模型,$G=(V,E,r,A)$,其中的 V 表示 G 中所有节点的集合,E 表示 G 中所有边的集合,r 表示 G 的根节点,A 是所有节点所带标签的集合.图 1(b)即为满足 XPath 树模型定义的 XML 树,图中每个元素节点左侧的数字序列表示该节点的 Dewey 编码,右侧的数对则对应前、后序遍历生成的区间编码.有关两种编码的叙述参见文献[2].

```
〈?xml version="1.0" encoding="UTF-8"?〉

〈proceedings〉
 〈issue〉
  〈articles〉
   〈article〉
    〈title〉 bibliography on data design 〈/title〉
    〈authors〉
     〈author position=0〉 Karen Botnich 〈/author〉
    〈/authors〉
   〈/article〉
   〈article〉
    〈title〉 face recognition 〈/title〉
    〈authors〉
     〈author position=0〉 cola cohen 〈/author〉
    〈/authors〉
   〈/article〉
  〈/articles〉
  〈publisher〉
   〈year〉 2003 〈/year〉
   〈address〉
    〈country〉 Germany 〈/country〉
   〈/address〉
  〈/publisher〉
 〈/issue〉
〈/proceedings〉
```

(a) XML document instance
(a) XML 文档示例



(b) XML tree instance
(b) 对应的 XML 树

Fig.1 XML document instance—Proceedings.xml, and its XML tree

图 1 XML 文档示例——Proceedings.xml 及对应的 XML 树结构示例

XML 数据包含的信息可分为两部分,一是内容(content),由 XML 数据中包含的 text 文本以及属性值组成;另一部分为结构信息(structure),由 XML 数据的标签间的嵌套关系组成(通常不考虑由 IDREF 表示的关联关系),包括直接包含(direct containment)、紧包含(tight containment)和邻近关系(proximity property)等.这些结构关系与 XML 树模型下节点间的结构关系有着直接的对应,可由节点间的结构关系来描述.例如,紧包含关系对应树结构中的父亲-孩子(parent-child)结构关系,包含关系就对应祖先-子孙(ancestor-descendant)结构关系和父亲-孩子结构关系之和,而邻近关系则对应树结构中的兄弟(sibling)关系.由于 XML 标签有向树模型中的相关概念与数据结构中树结构的概念有着明显的对应关系,所以前面的叙述直接引用了相应的概念,没有作更多的叙述.这里介绍的有关 XML 数据的内容和结构信息不仅有助于把握 XML 数据的本质,而且也有助于更深刻地理解后续介绍的查询模式描述的内容:从 XML 数据中提取感兴趣的信息必然深受这两个特点的影响.

1.2 XML查询方式分类

作为日渐广泛采用的数据形式,从中提取有用的信息是一个不可回避的研究内容.为了从自描述的、半结构化的 XML 数据中抽取用户感兴趣的信息,研究人员开发了许多查询描述形式.根据查询请求描述特点的不同,可概括为两大类查询模式:XML Query 查询模式和 XML IR (information retrieval)查询模式.本节就对这一分类及其相关研究内容的概括.

1.2.1 XML Query 查询模式

在从 XML 数据中获取感兴趣信息的处理过程中,有一类查询描述方式占有重要的地位,即 XML Query 查询模式:首先定义精致的查询模式描述语言,用户借助它来描述自己感兴趣的模式,将用户的模式交由实际的 XML 数据处理系统处理,返回与模式相匹配的结果.

1.2.1.1 XML Query 查询模式概述

属于此类的查询语言有很多,包括 Lorel[3],XML-QL[4],XML-GL[5],Quilt[6],Xpath[7],Xquery[8].其共同的特征是采用了正则路径表达式[9−11]的形式,其本质是捕捉 XML 数据单元间的结构关系和内容.XPath 是实现 XML 数据周游的基本语言,是 XSLT,XQuery 的基础.

根据前面的叙述,这类查询语言的缺陷也是明显的:首先,查询语言的使用者必须学习相关的语法机制,这一点本身就增加了普通用户使用它们的难度.而且用户即便是已经掌握了相关的查询描述机制,如果她/他不了解实际要查询的 XML 数据的组织情况(例如,数据中有哪些标签?感兴趣的标签间具有什么样的关系?),她/他同样不能完成查询语句的书写.在此,本文并不想通过概述相关查询语言的语法描述规则来说明这一点,而是采用实际例句的形式帮助读者意识到这一点.需要说明的是,这类查询语言发展到现在已出现了集中的趋势,其代表便就由 W3C 推出的 XPath 和 XQuery 查询语言.

图 2 给出的两个示例分别是 XPath 和 XQuery 的查询语句.图 2(a)为 XPath 查询语句,表示从 XML 文档中提取所有作者名包含 Botnich 并且该作者位于文章首位的所有文章的名称.其中的“//”表示位于其前、后的标签(例如 issue 和 articles)对应的节点应具有祖先-子孙的结构关系;图 2(b)为 XQuery 查询语句,表示查询 proceedings.xml 将所有文章的题目全部列出来,其中除了前述 XPath 中有的“//”以外,FOR 和 RETURN 则是新的内容.从两个实际的查询语句的例子中可以看出,除了需要掌握 XPath 中的技术外,用户还需要进一步理解 FOR,LET,ORDER BY,WHERE 和 RETURN(即 XQuery 中最具代表性的 FLOWR 格式)等知识(实际上 XQuery 还有其他内容).这对于普通用户而言不能不说具有相当的难度.有感于此,并且看到传统信息检索对用户使用友好的特点,促使部分研究者将信息检索特点引入 XML 数据处理的研究,即后面第 1.2.2 节要概括的内容.

如前所述,由于 XPath 查询语言是 XSLT,XQuery 的基础,所以,XPath 描述的查询模式的不同形式也就代表了 XML Query 查询模式的主要内容,主要分为 3 个层次[12]:线性路径表达式、分支路径表达式和路径树.

定义 **1**. 线性路径表达式.递归定义如下:

(1) 如果 *s* 是一个标签,例如 issue,article,那么“/*s*”,“//*s*”都是线性路径表达式;

(2) 如果 *L* 是一个线性路径表达式,*s* 是一个标签,那么“*L*/*s*”,“*L*//*s*”也是线性路径表达式.

当线性路径表达式中不存在“//”时,通常称为简单路径表达式.

定义 **2**. 分支路径表达式.递归定义为:

(1) 如果 L_0,L_1 是线性路径表达式,k 是一个整数,那么,$L_0[L_1]$,$L_0[k]$就是分支路径表达式;

(2) 如果 B 是一个分支路径表达式,L 是一个线性路径表达式,那么,$B[L]$,$B[k]$也都是分支路径表达式.

定义 **3**. 路径树.路径树 $T(N,E)$(N 是节点集合,E 是边的集合)可递归定义如下:

(1) 以一个线性路径表达式为标签的单个节点就是一棵路径树;

(2) $T_1(N_1,E_1)$,$T_2(N_2,E_2)$是两棵路径树,其中 $v_1 \in N_1, v_2 \in N_2$,如果从 v_1 到 v_2 之间有一条边 e,那么,$T(N_1 \cup N_2, E_1 \cup E_2 \cup \{e\})$也是一棵路径树;

(3) $T_1(N_1,E_1)$是一棵路径树,$v_1 \in N_1$,v_2 是一个标签为 k 的节点,e 是从 v_1 到 v_2 之间的一条边,那么,$T(N_1 \cup \{v_2\}, E_1 \cup \{e\})$也是一棵路径树;

(4) 路径树中的节点通常会与如下形式的谓词(predicate)关联:*op value*,其中,*op* 为“=”或者“≠”;

(5) 路径树中的 1 个或多个节点可被设定为输出节点.

其中,路径树的定义包含了研究文章中常见的“小枝(twig)”的概念 [13−17]:专指具有多个叶节点的树结构形式的路径树查询形式.图 2(a)中的 XPath 语句就是 Twig 查询模式的实例.针对 Twig 查询模式的处理是 XML Query 查询处理研究中的主要问题之一.相关工作的叙述参见第 2.1 节.

```
//issue//articles[.//author[text()=Botnich][@position=0]]
//title
```

(a) XPath instance
(a) XPath 查询语句示例

```
LET $titles:=document ('proceedings.xml')
    //articles//title

FOR $title in $titles
RETURN ⟨titles⟩
        ⟨title⟩
          ⟨name⟩ $title ⟨/name⟩
        ⟨/title⟩
      ⟨/titles⟩
```

(b) XQuery instance
(b) XQuery 查询语句示例

Fig.2 XML Query instances

图 2 XML Query 查询语句示例

1.2.1.2 XML Query 查询实现的要点及相关研究概括

通过上述 XML Query 查询模式 3 种层次的描述可以看出,如何快速地确定任意两个元素间的结构关系(containment relationship)是实现此类查询模式的关键.为达到这一目标,相关的研究者提出了种类繁多的索引方法,可概括为两大类:一是节点记录类索引方法;二是结构摘要类索引方法.前者进而可分为 3 个小类:节点序号类索引方法、节点路径类索引方法以及混合类索引方法.有关内容已在文献[2]中作了较为详尽的叙述,所以在本文中对这方面的内容不再复述.文献[18]的阅读也会有助于读者进一步理解此处的内容.

需要说明的是,在浏览针对 XML Query 查询处理的最新研究文献时还发现,尽管新出现的索引技术不能直接被文献[2]所覆盖,但是仍然可以很好地归入文献[2]中的划分类别.例如,在 VLDB 2006 文献[19]提出的 FIX 索引方法,其基本内容为:在认识到对应于 Twig 结构的矩阵的特征值向量可作为表征该 Twig 结构的关键字的基础上,将 XML 文档进行 Twig 分解并根据特征向量构建索引结构,以满足 XPath 的查询.虽然该方法不与文献[2]中提到的任何索引方法相同,但是按照该文的叙述,该方法是可以归入结构摘要类索引结构的.

另外,还有一些最新的研究文献中也与 XML 数据及 XML Query 查询有着紧密的联系,只是这类文章的切入点大多是着眼于获得同时包含其他信息的 XML 结构片段.这类文章包括文献[20−22].文献[20]在 XML 数据处理中引入元数据信息(meta-data,如数据的时新性、完整性以及敏感性),并提出了两个索引结构 FMI(full meta-data index)和 IMI(inheritance meta-data index)以支持对这类信息的管理;文献[21]则考察了 XML 以及 XPath 在语言数据(linguistic data)管理中的适用情况,进而给出了面向语言数据的查询语言——LPath;文献[22]尝试了如何管理 XML 数据中的不确定信息的情况,例如,语义并不明确的 Web 服务的信息,并提出了模糊树模

型(fuzzy tree model)对这类信息进行管理.这类与 XML Query 和 XML IR 处理模式既有紧密联系同时又有所区别的思路,构成了 XML 数据处理未来的研究方向之一.

1.2.2 XML IR 查询模式

如果说类似 XPath 的查询处理是近期 XML 数据处理的主要研究内容之一,那么,在 XML 数据查询中吸收传统信息检索[23]研究中的特点,如方便普通用户使用、返回的结果能够体现与查询模式的相关程度等就成为当下 XML 数据处理新的研究点.本文将针对向这一方向努力的查询方式记作 XML IR 查询.

1.2.2.1 XML IR 查询模式概述

根据吸收信息检索特点方式的不同,XML IR 查询方式可以分为 3 个方面:扩展前述 XML Query 查询语言(如 XQuery)以吸收部分 IR 特点的方式[24−29],本文称为 XML IR/query 方式;另一种方式就是直接将对用户使用友好的关键字查询延伸至 XML 数据[30−34],记为 XML IR/keyword 方式;最后一种就是用户直接使用 XML 片段来描述查询请求的方式[35],记作 XML IR/fragment.图 3(b)即为用 XML 片段描述查询请求的示例.由于这种方式非常直接,而且有关的研究非常少,所以本文在此仅对 XML IR/query 和 XML IR/keyword 两种方式作简单的概括.

```
FOR $i IN document (“proceedings.xml”)//article
WHERE $i//author contains ‘Cohen’
      AND $i//title contains ‘Face’
      AND $i//title contains ‘Recognition’
RETURN ⟨result⟩
         ⟨author⟩ $i//author ⟨/author⟩
          ⟨title⟩ $i//title ⟨/title⟩
       ⟨/result⟩
```

(a) Xquery FT instance
(a) XQuery FT 查询语句

```
⟨article⟩
  ⟨title⟩ Bibliography ⟨/title⟩
  ⟨authors⟩
    ⟨author⟩ Botnich ⟨/auhtor⟩
  ⟨/authors⟩
⟨/article⟩
```

(b) XML IR/fragment instance
(b) XML IR/fragment 查询语句

Fig.3 Query instances of XQuery FT and XML IR/fragment

图 3 XQuery FT 和 XML IR/fragment 查询语句

1.2.2.1.1 XML IR/query 模式

将 IR 的特点吸收到 XML Query 语言中,是 XML IR 处理研究中的一个重要方面.早期的扩展多是针对 IR 特性的某一方面进行的,例如,仅仅支持布尔查询的扩展,或者仅仅支持关键字相似性(keyword similarity)的扩展[36,37]以及针对邻近距离和相关性分级(proximity distance,relevance ranking)的扩展[24,29,30,38,39];而且,所扩展的 XML Query 语言也是各种各样的,例如,2000 年、2001 年的 XIRQL 扩展的便是 XQL 查询语言.近期由于 XQuery 在 XML Query 查询处理中的突出地位,大多数的扩展还是基于 XQuery 进行的.例如,2004 年的 TeXQuery[28]、FleXPath[29]和 2005 年的 GalaTex[27].针对这一方向的研究,W3C 组织于 2005 年专门推出了覆盖全文检索特点的 XQuery 扩展规范——XQuery FullText (XQuery FT),它在扩展的 XQuery 中同时支持了经典全文检索所具有的一些主要特征,例如简单的关键字查询、布尔查询以及关键字-距离谓词(keyword-distance predicates).

在实际系统实现上,诸多原本针对 IR 的搜索引擎也进行了扩展,以支持 XML 数据上的搜索.为支持 IR 的特点,这类引擎所支持的查询请求描述大多采用两类方法:一方面是引进文本和上下文相似性判断的操作来体现一定功能的结果分级机制;另一方面,大多数还是采取了在描述语言中引入 XPath 部分功能的形式,并采取类似 SQL 语句执行的方式,例如 ELIXIR,XXL 以及 XIRQL.

但是,从图 3(a)给出的符合 XQuery FT 规范的查询语句的实例可以看出,对普通的用户来说,想要掌握这类语句仍然具有相当的难度.

1.2.2.1.2 XML IR/keyword 模式

XML IR 查询的另一种重要方式就是直接将传统信息检索中的关键字查询方式应用到 XML 数据中.主要有两种方式:一是直接将纯关键字方式不加修改地挪用到 XML 数据的查询中;二是辅助信息限定关键字所在

节点的范围,例如标签[26]或标签路径信息[40,41].Cohen Face Recognition 就是一个纯关键字的示例;后者的实际例子可以是 Face+Recognition article: Author: Cohen,表示希望获取包含作者信息(Cohen)而且名称包含 Face 和 Recognition 的所有 article 片段.由于后者引入的标签信息使得用户还要了解 XML 数据的实际组织,增加了用户使用的复杂性,而且本质上讲,这种扩展关键字的方式只是增加了过滤关键字节点的作用,实际处理与纯关键字方式并无很大不同,所以,本节仅针对纯关键字形式的处理加以概括.虽然这种直接嫁接保留了原有的优点,但是由于前述 XML 数据的独有特点,使得针对 XML 数据的关键字处理具有了新的特征.例如,如何快速获得所有满足关键字组合语义的最紧致 XML 片段的问题,实际研究中,研究者往往将这个问题转化为 SLCA 问题[32].显然,它是 XML 关键字查询处理的基本问题,有关它的讨论放到第 2.2 节;在获取了所有包含给定关键字的 XML 片段后,另一个重要的问题就是 XML 片段相似程度的计算(similarity measure).由于 XML 片段具有结构信息,使得针对它的相似性计算在除了考虑传统的相似性以外,还需要考察结构的相似性.围绕这一问题的讨论内容较多,本文第 3 节对它进行了讨论.

1.2.2.2 XML IR 查询实现的要点

与传统的信息检索研究不同,针对 XML 数据的 IR 查询处理有着自己的特点,主要表现为两个方面:一是查询结果往往不是整个文档,而是 XML 数据中的片段;二是针对所获 XML 片段的相似性度量必须包含结构相似性的内容.这两个问题分别构成了 XML IR 处理中的两个基本问题,对它们的叙述分别构成了第 2.2 节和第 3 节的内容.

1.3 XML查询模式小结

图 4 显示了到 2006 年 9 月为止有关 XML 数据查询研究的概括图.横轴对应于相关研究提出的时间,纵轴表示对应查询模式描述技术属于前述的 3 种查询模式 XML Query,XML IR/query 和 XML IR/keyword 中的哪一种.

首先我们发现,对应 XML Query 查询模式的研究目前已明显集中于 W3C 组织推出的 XPath 和 XQuery 查询语言.其次,将信息检索中的技术特点吸收到 XML 数据查询中体现了近几年 XML 数据研究的一种趋势;在这一趋势中,扩展已有 XML Query 查询的 XML IR/query 方式处于主流地位,而且受 XML Query 发展的影响,该方向的研究也越来越集中于 W3C 的 XQuery FT[42]查询语言的实现与扩充.至于吸收信息检索技术特点的 XML IR/keyword 查询方式,其模式仍然是以关键字形式为主,并形成了两个研究的焦点:一个是 SLCA 问题;另一个就是所获 XML 片段的相似性计算问题.

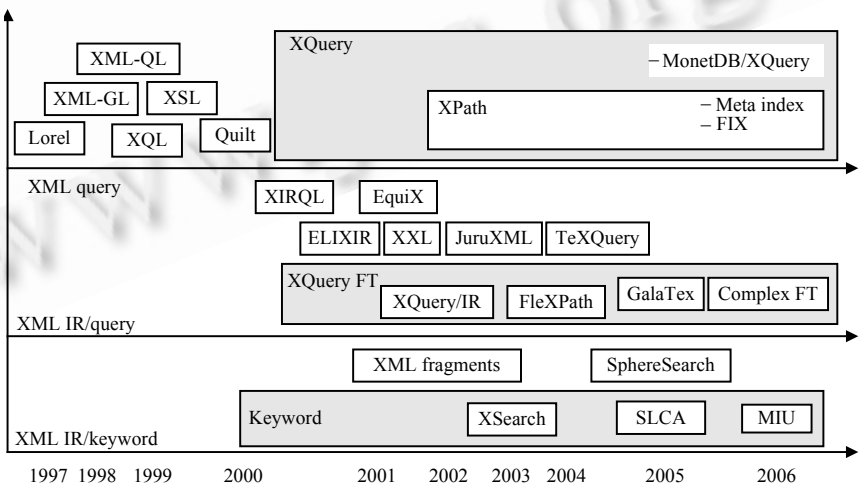


Fig.4 Summary of query patterns for XML data by September, 2006

图 4 XML 查询模式汇总图(截止到 2006 年 9 月)

2 XML 查询处理

在大致了解了前述 3 种查询模式后,从当前研究专题中选取出如下两个问题作为本节的内容,它们是 XML Query 中的 Twig 问题以及 XML IR/keyword 中的 SLCA 问题.之所以选取这两个问题,是因为它们在各自的查询方式中所具有的突出地位,而且通过对这两个问题的讨论,可以帮助读者进一步了解查询处理的实现技术.例如,Twig 查询模式是 XML Query 中较为复杂的查询形式,涵盖了 XML Query 查询方式的所有特点——结构包含、分支等;而 SLCA 对于 XML IR/query 和 XML IR/keyword 的意义可从第 1.2.2 节中得知.至于为什么没有将 XML IR 中的 XML 片段相似性度量的相关研究也放在这一节讲述,原因有二:一是该技术与查询实现相比相对独立;二是 XML 片段相似性度量技术研究本身内容丰富,可以单独作为一节进行讲述,即第 3 节的内容.

2.1 Twig查询模式的处理

2.1.1 问题描述

Twig 是 XML Query 处理中一个重要的查询模式,其主旨就是搜索 XML 树得到满足树状结构的查询模式的结果.文献[12]给出的描述是当前该研究方向的参照,简要叙述如下:

首先是查询匹配问题.

给定具有 n 个节点的 Twig 查询模式 Q 和一个 XML 数据库 G,Q 在 G 中的一个匹配是指在 Q 的节点与 G 的节点之间存在如下的映射关系:(i) 对应于查询节点的目标节点所含的数据必然满足查询节点上的谓词条件; (ii) 目标节点间的结构关系与查询节点组合间的结构关系必须一致,包括父亲-孩子关系和祖先-子孙关系.那么,具有 n 个节点 Q 模式的结果可表示为一个 n 元组的关系$(d_1,\ldots,d_n)$.

进而可定义 Twig 模式匹配问题为:给定 Twig 查询模式 Q 和一个具有某种索引结构的 XML 数据库,所谓 Twig 模式匹配问题就是搜索 XML 数据库得到所有满足 Q 模式的 XML 数据片段,表示为 n 元组的形式.

2.1.2 研究概述

对 Twig 查询模式通行的求解算法大多采取如下步骤:

(1) 将 Twig 查询模式分解为二元结构关系(父亲-孩子,祖先-子孙);

(2) 利用结构连接的算法(即判断两元素满足结构关系的连接算法:Structural join algorithms)从 XML 数据库中寻找所有满足上述结构关系的节点集合;

(3) 将得到的中间结果合并为满足全部结构关系的最后结果.

直观地看,上述 Twig 查询模式求解方式中影响计算性能的因素主要有两个:一个是结构连接算法的效率;另一个就是将 Twig 模式分解的规模.显然,如果需要结构连接计算的数目少,整体的计算则必然会有较好的性能表现.在实际中,针对这一问题的求解除了前述两种思路外,还有一种为吸收流式处理方式的思路,分别叙述如下:

(1) 提高结构连接操作性能

采用这一思路的方式有两种:一是设计新的连接算法,如文献[43]中的 MPMGJN(multi-predicate merge join)算法,其基本思路是,减少传统关系数据库中连接算法中构建全部连接记录之后过滤出所需结果的过程,采取按照所需进行连接的方式;第 2 种方式就是在结构连接中吸收 XML 索引结构信息,达到减少连接记录数目的目的[44,45],常用的 XML 索引形式为节点记录类编码中序号对索引形式的区间编码.这是因为对应节点的数值区间形式具有有效判断节点间结构关系的能力,从而达到筛选节点的目的.有关区间编码这一特点的更详细的叙述参见文献[2]的相关章节.

(2) 增大分解粒度

实际的算法包括文献[46–49].这类方法的基本思路就是采取增大 Twig 模式分解粒度以期达到减少结构判断操作的数目,从而提高整体计算的性能.需要指出的是,虽然文献[47]采取了 FST(有限状态映射器 finite state transducer)的机制,但是其求解 Twig 查询模式的基本方式促使我们将其归为这一类.其计算方式为:首先对 Dewey 编码进行扩展,使其能够蕴含对应节点标签路径的信息,在求解 Twig 查询模式时首先得到对应 Twig 查

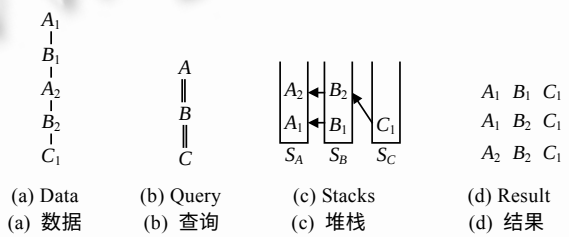
询模式叶节点的所有扩展 Dewey 编码,然后借助于 FST 过滤出符合 Twig 查询模式的节点集合.所以,本质上是将 Twig 查询模式分解为路径之后求解的方式.

沿着这一思路,2006 年,VLDB 中的文献[19]在吸收了 XML 数据处理中结构索引(也称作结构摘要索引形式,参见文献[2]中的叙述)概念的基础上,进一步引入 Twig 特征(twig feature)的技术.

(3) 流式处理

与前述两种方法不同,本节介绍的流式处理方式的基本思路为:首先依照 Twig 查询模式设置相应的堆栈结构,每个堆栈对应 Twig 查询模式中相应的节点,堆栈的排列次序与前序遍历 Twig 查询模式时的节点次序相同;然后,顺序地扫描 XML 数据,将扫描过程中遇到的对应于 Twig 查询模式标签的节点顺序地压入对应的堆栈,当遇到对应于 Twig 查询模式叶节点标签的节点时,回溯经过的相应堆栈,这样,符合 Twig 查询模式结构关系的节点序列即为满足查询请求的结果[12,14,50,51].

该类方法的示意由图 5 给出[12],称为完整路径连接方式(holistic path join,简称 HPJ).图中要注意的是,图 5(c)中返回 $A_1B_2C_1$ 路径结果的处理.当扫描图 5(a)中的各节点遇到 C_1 时,HPJ 即着手根据堆栈中节点的情况构建相应的节点路径.首先看到 S_B 堆栈中最上一层的节点 B_2,查看堆栈 S_A 中与它能够构成路径的节点只有 A_2,所以返回 $A_2B_2C_1$ 路径;进一步考察 S_A 中的 A_1 节点,由于它也与 B_2 相连接,所以,对应的 $A_1B_2C_1$ 也是满足查询的节点路径.类似地,可以得到 $A_1B_1C_1$.此时,S_B 中已没有可用的节点,故可以将 S_C 中的 C_1 节点弹出.由于弹出 C_1 后 S_C 堆栈为空,这就意味着整个处理已结束.



(a) Data (b) Query (c) Stacks (d) Result
(a) 数据 (b) 查询 (c) 堆栈 (d) 结果

Fig.5 Holistic path join processing

图 5 完整路径连接处理示意

根据上面对 HPJ 处理的叙述,显然可以利用 XML 数据编码信息帮助判断各节点的结构关系,这一思路正是后续 3 篇文献[14,50,51]的工作之一;除此之外,它们大多也考虑了更为复杂的 Twig 查询模式的处理,如文献[14]提出了完整 Twig 连接(holistic twig join,简称 HTJ),文献[51]吸收了结构摘要的信息.但是,基本的处理还是延续了文献[12]的概念.

2.2 XML关键字查询中的SLCA问题

2.2.1 问题描述

对于 XML 数据的关键字查询来说,基本操作就是找到包含给定关键字的 XML 片段的根节点:SLCA 节点.

定义 **4**. SLCA 问题:给定一棵标签有向树 $G=(V_G,E_G,r,A)$以及一组关键字 $W=\{k_1,k_2,\ldots,k_k\}$,那么,SLCA 问题就是确定满足给定关键字的最紧致 XML 片段的根节点的问题.所谓最紧致 XML 片段(XML 子树)S 是指 S 具有如下特征:

(1) 全部关键字都出现于 S 的叶节点中;

(2) S 中的任意子树都不可能包含全部关键字.

当前,针对 SLCA 问题的求解方法多与 Dewey 编码有着紧密的联系,主要原因就是因为 Dewey 编码包含了该节点所在节点路径上所有节点的 Dewey 码信息,而这些信息对于建立节点与路径的关联关系,也就是关键字所在节点间的结构关系,有着实际的便利.图 1(b)即为包含 Dewey 码的 SLCA 问题示意:由实线围起来的 XML 片段即为对应于关键字 Cohen Face Recognition 的最紧致 XML 片段,该片段根节点即为 SLCA 节点.从中可以发现,对应 SLCA 节点的 Dewey 码和关键字所在节点的 Dewey 码具有有趣的关联:作为 SLCA 节点的 article 节

点的 Dewey 码为 0.0.0.1,关键字 Face Recognition 所在节点的 Dewey 码为 0.0.0.1.0,另一个关键字 Cohen 所在节点的 Dewey 码为 0.0.0.1.1.0,借助文献[2]中介绍的 Dewey 编码知识可知,SLCA 节点的 Dewey 码正是两个关键字所在节点 Dewey 码的最长公共前缀.

需要指出的是,当一个 XML 文档中包含有多个对应同一关键字的节点时,计算最合适的 SLCA 节点并不能通过求解包含关键字的节点的 Dewey 码的最长公共前缀简单地解决.

2.2.2 研究概述

针对 SLCA 问题,文献[32]归纳出了 3 种算法:Indexed Lookup Eager (ILE)算法、Scan Eager (SE)算法和 Stack 算法[33].鉴于 Dewey 编码具有能够提供路径内部节点信息的能力,所以,这 3 种算法都是基于 Dewey 编码的.文中声称 ILE 算法具有较好的性能,其流程概述如下:

(1) 首先修改 B+-索引树结构.

一是使它能够实现对(keyword,dewey)类型数据的支持,其中的 keyword 表示感兴趣的关键字字符串,dewey 则表示 Dewey 码.记修改后的 B+-树结构为 DBPT(dewey B-plus tree).显然,DBPT 必须支持 Dewey 数据上的运算,包括 Dewey 码的大小、包含、公共前缀(lca)以及求解两 Dewey 码中的处于子孙位置的 Dewey 码的 descendant 运算.

此外,DBPT 还必须实现两个基本的运算,即 lm(dewey, keyword)和 rm(dewey, keyword),二者的目的分别是搜索 DBPT 从所有关键字为给定 keyword 的 Dewey 码集合中得到小于/大于(smaller/greater than)给定 dewey 码的最大/最小 Dewey 码.

(2) 得到对应给定关键字组 $k_1,k_2,...,k_k$ 的 Dewey 码集合 $D_1,D_2,...,D_k$.

D_i 是所有包含关键字 k_i 的 Dewey 码集合,并按照每个集合包含的元素数目的多少(即集合"势"的概念)进行排序,其中,D_1 对应于元素数目最少的 Dewey 码集合.

(3) 那么,基于 Dewey 码集合 $D_1,D_2,...,D_k$ 求解 SLCA 节点 ILE($slca(D_1,...,D_k)$)算法可用如下的公式来表达:

$$slca(D_1,...,D_k)=removeAncestor\left(\bigcup_{v\in D_1} slca(\{v\},D_2,...,D_k)\right),$$

其中的 $slca(\{v\},D_1,\ldots,D_k)$运算由如下公式确定:

$$slca(\{v\},D_2,...,D_k)=slca(slca(\{v\},D_2,...,D_{k-1}),D_k),$$

而从一个 Dewey 码集合 D_i 中寻找与一个 Dewey 码 v 匹配的最佳 SLCA 节点的 Dewey 码的计算,由如下公式确定:

$$slca(\{v\},D_i)=\{descendant(lca(v,lm(v,D_i)),lca(v,rm(v,D_i)))\}.$$

从这 3 个计算公式可以看出,对 D_1 中的任意个 Dewey 码都要在 DBPT 上执行$(k-1)$次 *lm* 和 *rm*;在得到对应 *lm* 和 *rm* 运算的两个 Dewey 码后,ILE 还需要运行两次 *lca* 运算以及一次 *descendant* 运算才能从中计算得到一个准 SLCA 节点 Dewey 码.

进一步考虑实际实现:由于无法确定用户会输入哪些关键字,所以,ILE 必须依据全部的 XML 节点构建 DBPT 结构,如果 XML 数据有 N 个节点,并取 B+树的分叉数为 m,根据数据结构的理论,BPT 的高度不会小于$\lfloor \log_m^N \rfloor$,那么,ILE 算法总共要进行 $(k-1)\times|D_1|\times\lfloor \log_m^N \rfloor$次的 *rm* 和 *lm* 运算.如果考虑到对应关键字的节点数目往往远远小于 XML 数据中的节点数目 N,则以判断出由于 ILE 必须依赖全部 XML 节点求解 SLCA,其计算效率并不高.所以,提高 SLCA 节点的求解效率仍然是需要妥善解决的问题.

3 XML 片段的相似性度量

在获取了满足给定关键字的最紧致 XML 片段后,XML IR/keyword 查询处理的另一个重要步骤就是如何计算 XML 片段间相似程度的问题.由于这个问题的有关研究内容较多,所以专门在小节对该问题进行讲述,尽管它是 XML IR/keyword 查询处理的重要环节.

在第 1.1 节中提到 XML 数据的重要特点之一就是它的结构性,满足给定关键字的 XML 片段同样具有这一

特点,所以,衡量 XML 片段间相似程度的问题就不可能回避其结构信息.现有针对此问题的研究可分为两类:一类基于树结构编辑距离的方法;另一类统信息检索中 TF*IDF 技术深刻影响的近似方法.

3.1 基于树编辑距离的相似性度量

这类方法的核心就是标签有向树结构编辑距离的概念[52,53].

定义 5. 标签有向树结构编辑距离:基于标签有向树结构 $G=(V,E,r,A)$,可定义 3 种编辑操作:更名(rename,即将某一节点的标签修改为另一标签)、删除某节点(delete)和插入一节点操作(insert),并指定 3 种编辑操作的代价分别为 C_R,C_D 和 C_I.对应两个标签有向树 T_1 和 T_2 间的编辑脚本 s,是指在仅仅使用前述 3 种编辑操作的前提下将 T_1 转换为 T_2 的编辑操作的序列,其编辑代价记作 $\gamma(s)$,表示 s 中全部编辑操作代价之和.由于每个编辑操作都对应一个编辑代价,那么,所有将 T_1 转换为 T_2 的编辑脚本 S 中代价最小的那个脚本就是最优编辑脚本(optimal edit script),其编辑代价值就是 T_1 和 T_2 之间的编辑距离.记作 $EDist(T_1,T_2)$,即 $EDist(T_1,T_2)=\min_{s\in S}\{\gamma(s)\}$.

虽然标签有向树结构的编辑距离有着清晰的表述形式,但是其实际计算往往过于复杂,从而研究者更多地关注于吸收传统信息检索 TF*IDF 技术[23]的近似方法.

3.2 相似性度量近似方法

由于本节的近似方法与传统信息检索中的 TF*IDF 技术有着紧密的联系,因此本节首先简要地回顾传统信息检索中 TF*IDF 技术.然后将针对 XML 片段相似性度量的近似方法分为 4 种类型:基于 TF*IDF 的回归方法、结构化的 TF*IDF 方法、MLP(叶节点路径最大值:maximum leaf path)方法和路径包模型(path bag model).

3.2.1 传统的 TF*IDF 技术

传统信息检索处理有一个非常鲜明的特点,那就是查询处理返回的结果总是与用户提供的查询模式最为相关的部分结果.例如 Top-k,k-NN(k nearest neighbor)查询结果显示功能.为了实现这一功能,就需要信息检索处理具有衡量所获结果与给定查询模式的相似程度的能力,即 Ranking 机制或相似性度量机制(similarity measure).传统信息检索领域针对这一问题提出了许多解决办法,详细情况请参见文献[23],其中最具代表性的就是向量空间模型(vector space model,简称 VSM)和 TF*ID 计算方法.

向量空间模型是 TF*IDF 技术的基础,其基本思路是:首先从查询目标文档($d_1,d_2,...,d_N$,N 为文档的数目)中提取出感兴趣的文字单元(term unit),并利用文字单元组成的向量——$(t_1,t_2,...,t_m)$,m 为文字单元的数目——作为近似文档内容的形式,那么,文档(d_j)和关键字(q)都可看作是向量空间中的点,然后即可利用向量空间中点距离的知识来计算文档 d_j 与关键字 q 之间的近似程度.

基于这一思路,TF*IDF 计算方法如下:首先确定对应关键字的统计向量 $\vec{q}=(w_{1,q},w_{2,q},...,w_{m,q})$ 与对应文档 d_j 的统计向量 $\vec{d}_j=(w_{1,j},w_{2,j},...,w_{m,j})$ 表示.其中,统计信息 $w_{i,j}(1\le i\le m)$ 计算如下:首先确定文字单元 t_i 在文档 d_j 出现的频率 $freq_{i,j}$,进而确定该文字单元在所有文档中出现频率的最大值 $\max_l freq_{l,j}$,并记 n_i 表示包含文字单元 t_i 的文档数目,那么,$tf_{i,j}$ 和 idf_i 可根据如下公式计算($tf_{i,q}$ 的计算也类似):

$$tf_{i,j}=\frac{freq_{i,j}}{\max_l freq_{l,j}},\ idf_i=\log\frac{N}{n_i}.$$

文字单元 t_i 在文档 d_j 的权重计算如公式:$w_{i,j}=tf_{i,j}\times idf_i$.

而文字单元 t_i 在关键字模式 q 中的权重信息 $w_{i,q}(1\le i\le m)$ 计算如下(在实际研究中还有其他计算形式):

$$w_{i,q}=(0.5+0.5tf_{i,q})\times idf_i.$$

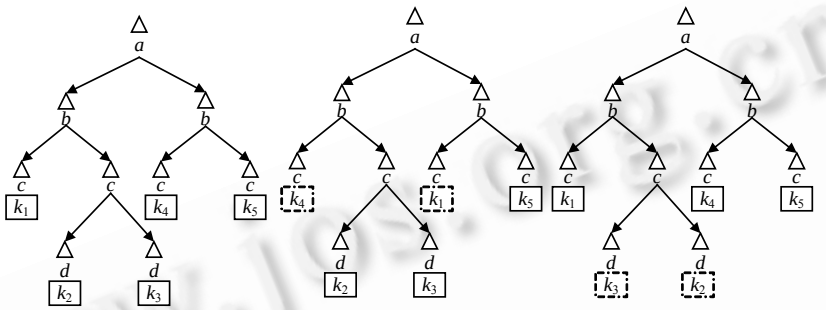
于是,基于向量空间的距离理论,文档 d_j 与关键字模式 q 之间的距离可计算如下:

$$sim(d_j,q)=\frac{\vec{d}_j\cdot\vec{q}}{|\vec{d}_j|\times|\vec{q}|}=\frac{\sum_{i=1}^{m}w_{i,j}\times w_{i,q}}{\sqrt{\sum_{i=1}^{m}w_{i,j}^2}\times\sqrt{\sum_{i=1}^{m}w_{i,q}^2}}.$$

可见,基于 VSM 的 TF*IDF 计算方法具有两个基本的技巧:首先是文字单元,即将查询目标与查询模式都看

作是由文字单元为基本元素构成向量的思路;其次就是利用向量空间的距离理论计算近似的相似程度.XML 片段的相似性判断近似方法大多延续了 TF*IDF 计算的技巧——抽取 XML 数据中的部分信息作为“单元”(例如标签路径[54]、小枝[55]等),然后将查询模式和 XML 片段都映射为“单元”向量空间中的点,然后利用向量空间的距离公式计算 XML 片段的相似性.

为了便于读者直观地了解后续近似方法的讨论,在此给出 3 个 XML 片段,如图 6 所示.其中,图 6(b)和图 6(c)都是通过交换图 6(a)中的某两个节点得到的片段.直观地看,图 6(a)和图 6(c)仍具有相同的结构;而图 6(a)和图 6(b)的结构则不相同.



(a) Source XML fragment (a) XML 源片段 (b) Exchanging k_1 and k_4 (b) 交换 k_1 和 k_4 后的片段 (c) Exchanging k_2 and k_3 (c) 交换 k_2 和 k_3 后的片段

Fig.6 Three XML fragments with same labels

图 6 标签相同的 3 个 XML 片段

3.2.2 基于 TF*IDF 的回归方法[31,33,35,55−57]

本质上讲,这类方法可以看作是一种基于叶节点文本统计特性的多元回归问题,即首先直接利用 TF*IDF 方法计算 XML 片段中叶节点的统计值,之后利用 XML 片段中叶节点到 SLCA 节点间的相对位置信息为参数求解对应于 SLCA 节点的影响因子.所以,这种方法并不能有效地捕捉 XML 片段的结构特征,即这类方法通常不能区分图 6(a)和图 6(b)之间的结构差异,而是会将它们作为同一结构.

3.2.3 结构化的 TF*IDF 方法[55,58−61]

这类方法同样吸收了 TF*IDF 技术中“单元(term)”的概念,只是此时的“单元”变成了具有结构信息的“Twig 单元(twig unit)”.所以,这类方法的核心就是如何分解 XML 数据,以便得到能够体现 XML 数据结构特征的“Twig 单元”向量.实践证明,这种分化的计算是复杂的.文献[55]也承认了这一点,并在提出了较为严格的关于“Twig 单元”分解的计算方法后,将注意力转到了类似路径包模型的近似计算方法上.文献[58−60]中的工作与此类似.这类方法不仅计算复杂,而且对 XML 数据的标签信息以及片段节点的相对位置信息是敏感的,对来自不同 XML 模式的 XML 片段的相似性判断也就无能为力.而从不同来源的 XML 数据中搜寻符合某关键字的信息,是有着实际的意义的.

3.2.4 MLP (maximum leaf path)[62]

这里提出的计算方法与其他近似方法都不相同,它将 XML 片段的相似性度量的计算分成了两个部分.其中一部分为引入 MLP 概念,进而构建 MLP 向量以捕捉 XML 片段的结构特征.所谓 MLP 就是指某节点到以之为根的所有叶节点的最长距离,如图 7 所示.另一部分是利用节点标签集合的覆盖程度来捕捉 XML 片段的语义相似性.但是,基于 MLP 向量的方法在捕捉 XML 片段的结构特征上并不理想.例如,如果将图 7 中的根节点所含的空白叶节点移到其左侧灰节点,则前后的结构是不同的,但 MLP 向量还是相同的.所以,图 6(a)和图 6(b)之间的结构差异也就不能分辨.

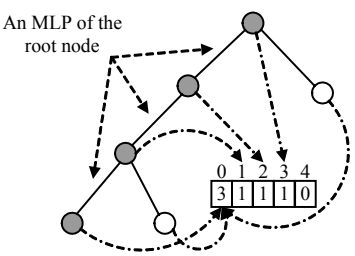


Fig.7 Illustration of MLP concept

图 7 MLP 示意图

3.2.5 路径包模型[54,63]

这类方法的核心思想是,采用 XML 片段中标签路径的分布信息作为 XML 片段结构特征的近似.由于它们没有考虑到节点间的关联结构,而且对节点标签也是敏感的,所以,这类方法就不能区分图 6 中 3 个 XML 片段的结构差异,这是因为它们具有相同的标签路径.为了弥补这一缺陷,文献[63]进一步引入了节点位置信息,以区分不同的分叉.然而,这种扩展也引入了不利的因素,即对节点的相对位置信息敏感,从而会将结构相同的图 6(a)和图 6(c)区分为不同的结构.

3.3 XML片段相似性度量方法小结

作为传统信息检索中重要的概念,查询结果的相似性比较同样在 XML 关键字查询中具有重要的意义.相关研究算法的汇总由表 1 给出.从中可以看出,针对这一问题的研究方法主要分为两类:一类是基于数结构编辑距离的方法;另一类是延续 TF*IDF 概念的近似方法.

由于前者实现的复杂性使得近期的研究更多地集中于近似方法上,可分为 4 种类型:基于 TF*IDF 的回归方法、结构化的 TF*IDF 方法、MLP 方法和路径包模型.但是,当集中考察各方法在捕捉 XML 片段的结构特征上的表现后发现,这些近似方法存在如下不足:即不能很好地分辨 XML 片段结构上的细微差异,而且部分算法也过多地依赖于 XML 数据的标签或节点相对位置信息,从而不能满足 XML 片段来自不同 XML 数据源的情况.

总之,当前,针对 XML 片段相似性度量的研究还有待进一步深入.其中,设计既能够有效地捕捉 XML 片段结构上的细小差异,同时又不必过多地依赖于 XML 片段中标签、节点相对位置信息的新方法,是一个值得关注的研究点.

Table 1 Summarization for similarity measures of XML fragments

表 1 XML 片段相似性度量方法概述

Similarity measure	Reference	Year	Class	Subclass	Description
Tree edit distance	[52,53]	1992, 1997, 2005	Accurate method	Tree edit distance	◦ Three edit operations: -Rename, delete, and insert ◦ Smallest edit cost ◦ Complicated to computing
Regression TF*IDF	[31,33,35] [55−57]	2003~ 2005	Approximate methods	Regression TF*IDF	◦ TF*IDF at leaf nodes ◦ Using hierarchical information as regression parameters ◦ Be insensitive to the structural difference
UBDist	[58]	2002		Structural TF*IDF	◦ Structural term is the kernel ◦ Structural term vector distance as the similarity measure ◦ Be dependent of labels or node positions ◦ Complicated to computing
Structural terms	[61]	2002			
Twig units	[55]	2005			
Binary branch distance	[59]	2005			
pq-Grams	[60]	2005			
Path bag model	[54,63]	2002, 2003		Path bag model	◦ Label path vector is the kernel ◦ Similarity measure is the vector distance ◦ Be dependent of labels or node positions ◦ Be insensitive to the structural difference
MLP	[62]	2004		MLP	◦ Two parts: Structural similarity + Semantic similarity ◦ MLP vector is the kernel for structural similarity measure ◦ Be insensitive to the structural difference

4 总结与研究展望

4.1 XML查询技术小结

XML 数据管理的研究已经受到数据库研究领域的广泛关注.其中,针对 XML 数据的查询技术的研究有着

特殊的意义:首先,数据管理的最终目的还是需要用户使用,而查询方式则是与用户直接相关的第一层机制;其次,查询方式也深切地影响到 XML 数据处理实现的方方面面,例如,不同查询模式的处理就需要不同索引机制.所以,对当前有关 XML 数据查询的研究作汇总也就有着实际的意义.需要说明的是,虽然当前 XML 数据处理方式有两种——流式 XML 数据处理形式和传统 XML 数据处理形式,但在 XML 查询描述形式上,二者有着相同的形式.

在广泛调研的基础上,本文首先将相关的查询方式分为两大类,分别为 XML Query 查询方式和 XML IR 查询方式.其中,后者又可以分为 3 个小类,分别为 XML IR/query 方式、XML IR/keyword 方式以及 XML IR/fragment 方式,并归结出 3 个受到广泛关注的研究问题,分别为 XML Query 中的 Twig 查询模式、XML IR 中的 XQuery FT 的实现以及 XML IR/keyword 中的 SLCA 问题和 XML 片段相似性度量问题.

在本文的书写过程中,我们深切地感受到,尽管前期我们针对 XML 数据的索引技术作了汇总[2],但是要真正理解各种各样的索引技术背后的"所以然",还是需要回到宏观的查询层上面.这也是让我们坚持完成本文,进而希望图帮助相关研究者对相关研究内容有所了解的原因.

4.2 研究展望

除了沿着传统的 XML 数据处理继续深入以外(例如查询优化、索引技术、Twig 处理、XQuery FT 实现[68]等),在通观现有相关研究的基础上,本文认为吸收传统信息检索特点的 XML IR 查询会是值得进一步深入探讨的方向,其中,方便普通用户使用的 XML IR/keyword 查询形式的处理尤其值得研究人员的关注.沿着这个方向进行思考,认为有两个问题值得进一步探讨:结构化关键字查询描述及结构相似性度量和 XML 数据处理系统的数据冗余,分别对应第 4.2.1 节和第 4.2.2 节.此外,根据 2006 年最新的研究文献,与 XML Query(包括 XML IR/query)有关的进一步研究可概括出两个值得关心的方向,分别为 XML Query 查询的理论研究及其实现(对应于第 4.2.3 节)和新应用背景下的 XML 数据的管理(对应于第 4.2.4 节).

4.2.1 结构化关键字查询描述及结构相似性度量

我们知道,结构信息对于 XML 数据而言有着突出的重要地位,但是,现有关键字的扩展方式在这一点上的表现差强人意——只是借助于标签或标签路径信息达到过滤关键字所在节点的目的,而不能提供更多的结构信息.因此,赋予 XML IR/keyword 查询以结构信息也就有着实际的研究价值:不仅有利于用户结构化组织感兴趣的关键字从而赋予查询模式更多的信息,而且也有利于减少查询过程中临时 XML 片段的数目.

这里要注意的是,新的机制虽然能够描述关键字间的结构信息,但却不应当以增加用户使用的难度为代价.例如,前述的关键字扩展形式就需要用户了解 XML 数据的组织信息.具有这两个特点的新型关键字查询模式由于没有标签信息的约束,基于它的查询就可以覆盖模式不同的 XML 数据,这一点是与纯关键字形式的查询能力相同的.

与之紧密相关的另一个问题就是研制结构与内容相分离的相似性度量技术.既然获得的 XML 片段可以来自不同的 XML 模式,那么,现有与标签信息有关的 XML 片段相似性度量方法也就无能为力,而且现有方法对结构差异不敏感的缺陷也需要研究人员提出新的方法.针对这一问题,MLP 方法中将结构相似性和语义相关性度量相分离的方式是一个有益的启发.

4.2.2 XML 数据处理系统的数据冗余

既然是针对 XML 数据查询的调研,过程中也就自然地考察了相关 XML 数据处理系统的实现情况.与前述 XML 数据查询的分类相吻合,有关的系统实现也能够很好地归入相关的类别中,这在一定程度上也说明了本文分类的合理性.在绘制完成查询模式汇总图的图 4 之后,我们意识到,当前 XML 数据处理系统中存在着数据冗余的现象,即支持 XML Query 查询的系统与支持 XML IR/keyword 查询的系统都必须各自维护一套 XML 数据的转换格式,即便查询的目标是同一个 XML 文档.这一点也可以通过当前 XQuery/FT 通行的实现架构得到验证:它们通常采取全文检索和 XQuery 处理松散结合的架构形式,而没有统一的索引结构[27−29].

文献[64]的基本索引结构形式为倒排索引与节点序号类索引混合的形式.但是,该文只是简单地延续了传统关键字查询中得到目标文档所含关键字位置信息的处理,而没有意识到 XML 数据上的关键字查询应以最紧

致 XML 片段的求取为特征的,而根据第 2.2 节以及对 XML 索引机制的了解[2],不论是文中的倒排表还是节点序号类索引,是不能有效地完成这样的计算的.文献[65]提出了将节点序号对索引与结构摘要索引相结合的复合索引结构:在得到 XML 数据的 1-Index 结构摘要后,对结构摘要中的节点赋予唯一的标示,然后将其与扩展节点序号对索引相结合.借助于结构摘要的特性,该索引方式可以支持 XML 数据的结构查询;而借助于基于堆栈的算法[32,33],该索引结构也能够支持包含关键字的查询,但是,由于直接依赖于结构摘要索引结构,该方式就不能避免第 1.2 节中提到的不足,而且根据文献[32]中的结论,基于堆栈的 SLCA 问题求解方案,其性能也并不占优.

4.2.3 XML Query 查询的理论研究及其实现

有感于关系代数对于关系数据管理的重要性,如何在理论上探讨 XML 数据管理,一直以来都是研究者关注的热点之一.由于描述这类数据的 XML 规范已经相对稳定,这类理论上的探讨也就更多地集中于 XML 数据的查询.文献[66,67]是针对这一方向的两个最新的研究成果.

此外,在 XQuery FullText 版本推出来之后,如何有效地实现 XQuery/FT 的查询机制,也就成为数据库研究人员需要着手解决的问题.到目前为止,支持大部分 XQuery/FT 特性的研究也只有文献[27]的 GalaTex 系统.该系统在原有实现 XQuery 查询的 Galax 系统上引入倒排表机制,以支持 XQuery/FT 中的与文本查询有关的特点,其实现的策略可以适用于其他基于 XQuery 查询引擎的 XQuery/FT 实现.尽管如此,XQuery/FT 的复杂性决定了针对它的实现还是存在许多值得研究的问题.例如,如何更有效地实现支持全文本查询相关的算子以及如何设计更为合理的程序框架,包括其索引机制的设计、查询执行流程的优化、目标 XML 片段的相关性计算等等.2006 年,SIGMOD 的 Complex FT[68]就是在这一方向的努力结果.

4.2.4 新应用背景下的 XML 数据的管理

正如第 1.2.1.2 节中提到的,将不同应用领域的数据以 XML 形式进行保存已日渐为研究者所采纳.虽然这类数据的管理可以自然地吸收现有 XML 数据管理研究的成果,但是,不同应用领域的不同特点也使得对这类 XML 数据的管理有了独特的要求.如何开发能够满足这类要求的技术,也就成为未来研究值得关注的又一个方向.

References:

[1] Meng XF, Zhou LX, Wang S. State of the art and trends in database research. Journal of Software, 2004,15(12):1822−1836 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/15/1822.htm

[2] Kong LB, Tang SW, Yang DQ, Wang TJ, Gao J. XML indices. Journal of Software, 2005,16(12):2063−2079 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/16/2063.htm

[3] Abiteboul S, Quass D, McHugh J, Widom J, Wiener J. The Lorel query language for semistructured data. Int'l Journal on Digital Libraries, 1997,1(1):68−88.

[4] Deutsch A, Fernandez M, Florescu D, Levy A, Suciu D. A query language for XML. Computer Networks, 1999,31(11−16): 1155−1169.

[5] Ceri S, Comai S, Damiani E, Fraternali P, Paraboschi S, Tanca L. XML-GL: A graphical language for querying and restructuring XML documents. Computer Networks, 1999, 31(11−16):1171−1187.

[6] Chamberlin D, Robie J, Florescu D. Quilt: An XML query language for heterogeneous data sources. In: Suciu D, Vossen G, eds. Proc. of the Int'l Workshop on the Web and Databases (WebDB 2000). Dallas: Springer-Verlag, 2000. 1−25.

[7] Clark J, DeRose S. XML Path Language (XPath) Version 1.0 W3C Recommendation. World Wide Web Consortium, 1999. http://www.w3.org/TR/xpath

[8] Chamberlin D. XQuery: A query language for XML W3C working draft. Technical Report, WD-xquery-20010215, World Wide Web Consortium, 2001. http://www.w3.org/TR/xquery/

[9] Li QZ, Moon B. Indexing and querying XML data for regular path expressions. In: Apers PMG, Atzeni P, Ceri S, Paraboschi S, Ramamohanarao K, Snodgrass RT, eds. Proc. of the 27th Int'l Conf. on Very Large Data Bases (VLDB). Rome: Morgan Kaufmann Publishers, 2001. 361−370.

[10] Cooper BF, Sample1 N, Franklin MJ, Hjaltason GR, Shadmon M. A fast index for semistructured data. In: Apers PMG, Atzeni P, Ceri S, Paraboschi S, Ramamohanarao K, Snodgrass RT, eds. Proc. of the 27th Int'l Conf. on Very Large Data Bases (VLDB). Rome: Morgan Kaufmann Publishers, 2001. 341−350.

[11] Zhang C. Relational databases for XML indexing [Ph.D. Thesis]. Wisconsin: University of Wisconsin-Madison, 2002.

[12] Bruno N, Koudas N, Srivastava D. Holistic twig joins: Optimal XML pattern matching. In: Franklin MJ, Moon B, Ailamaki A, eds. Proc. of the 2002 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Madison: ACM Press, 2002. 310−321.

[13] Amer-Yahia S, Cho S, Lakshmanan LVS, Srivastava D. Minimization of tree pattern queries. In: Aref WG, ed. Proc. of the 2001 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Santa Barbara: ACM Press, 2001. 497−508.

[14] Jiang HF, Wang W, Lu HJ, Yu JX. Holistic twig joins on Indexed XML documents. In: Freytag JC, Lockemann PC, Abiteboul S, Carey MJ, Selinger PG, Heuer A, eds. Proc. of the 29th Int'l Conf. on Very Large Data Bases (VLDB). Berlin: Morgan Kaufmann Publishers, 2003. 273−284.

[15] Chen ZY, Jagadish HV, Korn F, Koudas N. Counting twig matches in a tree. In: Young DC, ed. Proc. of the 17th Int'l Conf. on Data Engineering (ICDE). Heidelberg: IEEE Computer Society, 2001. 595−604.

[16] Lee DW, Srivastava D. Counting relaxed twig matches in a tree. In: Lee YJ, Li JZ, Whang KY, Lee DH, eds. Proc. of the 9th Int'l Conf. on Database Systems for Advances Applications (DASFAA). LNCS 2973, Springer-Verlag, 2004. 88−99.

[17] Jagadish HV, Al-Khalifa S. TIMBER: A native XML database. The VLDB Journal, 2002,11(4):274−291.

[18] Meng XF, Wang Y, Wang XF. Research on XML query optimization. Journal of Software, 2006,17(10):2069−2086 (in Chinese with English abstract). http://www.jos. org.cn/1000-9825/17/2069.htm

[19] Zhang N, Özsu MT, Ilyas IF, Aboulnaga A. FIX: Feature-Based indexing technique for XML documents. In: Dayal U, Whang KY, Lomet DB, *et al*., eds. Proc. of the 32nd Int'l Conf. on Very Large Data Bases (VLDB). Seoul: ACM Press, 2006. 259−270.

[20] Cho SR, Koudas N, Srivastava D. Meta-Data indexing for XPath location steps. In: Chaudhuri S, Hristidis V, Polyzotis N, eds. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Chicago: ACM Press, 2006. 455−466.

[21] Bird S, Chen Y, Davidson SB, Lee HJ, Zheng YF. Designing and evaluating an XPath dialect for linguistic queries. In: Liu L, Reuter A, Whang KY, *et al*., eds. Proc. of the 22nd Int'l Conf. on Data Engineering (ICDE). Atlanta: IEEE Computer Society, 2006. 52.

[22] Abiteboul S, Senellart P. Querying and updating probabilistic information in XML. In: Ioannidis YE, Scholl MH, eds. Advances in Database Technology, Proc. of the 10th Int'l Conf. on Extending Database Technology (EDBT 2006). Munich: Springer-Verlag, 2006. 1059−1068.

[23] Baeza-Yates R, Ribeiro-Neto B, *et al*. Modern Information Retrieval. Pearson Education Limited, 1999.

[24] Fuhr N, Groβjohann K. XIRQL: A query language for information retrieval in XML documents. In: Croft WB, Harper DJ, Kraft DH, Zobel J, eds. Proc. of the 24th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR). New Orleans: ACM Press, 2001. 172−180.

[25] Barg M, Wong RK. Structural proximity searching for large collections semi-structured data. In: Paques H, Liu L, Grossman D, eds. Proc. of the ACM Conf. on Information and Knowledge Management (CIKM). Atlanta: ACM Press, 2001. 175−182.

[26] Cohen S, Mamou J, Kanza Y, Sagiv Y. XSearch: A semantic search engine for XML. In: Freytag JC, Lockemann PC, Abiteboul S, Carey MJ, Selinger PG, Heuer A, eds. Proc. of the 29th Int'l Conf. on Very Large Data Bases (VLDB). Berlin: Morgan Kaufmann Publishers, 2003. 45−56.

[27] Curtmola E, Amer-Yahia S, Brown P, Fernàndez M. GalaTex: A conformant implementation of the XQuery FullText language. In: Florescu D, Pirahesh H, eds. Proc. of the 2nd Int'l Workshop on XQuery Implementation, Experience, and Perspectives (XIME-P). Baltimore: ACM Press, 2005. 1024−1025.

[28] Amer-Yahia S, Botev C, Shanmugasundaram J. TeXQuery: A FullText search extension to XQuery. In: Feldman SI, Uretsky M, Najork M, Wills CE, eds. Proc. of the 13th Conf. on World Wide Web (WWW). Manhattan: ACM Press, 2004. 583−594.

[29] Amer-Yahia S, Lakshmanan LV, Pandit S. FleXPath: Flexible structure and full-text querying for XML. In: Weikum G, König AC, Deβloch S, eds. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Paris: ACM Press, 2004. 83−94.

[30] Balmin A, Papakonstantinou Y, Hristidis V. A system for keyword proximity search on XML databases. In: Freytag JC, Lockemann PC, Abiteboul S, Carey MJ, Selinger PG, Heuer A, eds. Proc. of the 29th Int'l Conf. on Very Large Data Bases (VLDB). Berlin: Morgan Kaufmann Publishers, 2003. 1069−1072.

[31] Weigel F, Meuss H, Schulz KU, Bry F. Content and structure in indexing and ranking XML. In: Amer-Yahia S, Gravano L, eds. Proc. of the 7th Int'l Workshop on the Web and Databases (WebDB). Maison de la Chimie: ACM Press, 2004. 67−72.

[32] Xu Y, Papakonstantinou Y. Efficient keyword search for smallest LCAs in XML databases. In: Ozcan F, ed. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Baltimore: ACM Press, 2005. 537−538.

[33] Guo L, Shao F, Botev C, Shanmugasundaram J. XRANK: Ranked keyword search over XML documents. In: Halevy AY, Ives ZG, Doan A, eds. Proc. of the 2003 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). San Diego: ACM Press, 2003. 16−27.

[34] Florescu D, Kossmann D, Manolescu I. Integrating keyword search into XML query processing. http://www9.org/w9cdrom/index.html

[35] Carmel D, Maarek YS, Mandelbrod M, Mass Y, Soffer A. Searching XML documents via XML fragments. In: Proc. of the 26th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR). Toronto: ACM Press, 2003. 151−158.

[36] Chinenyanga T, Kushmerick N. Expressive and efficient ranked querying of XML data. In: Mecca G, Siméon J, eds. Proc. of the 4th Int'l Workshop on the Web and Databases (WebDB 2001). Santa Barbara: ACM Press, 2001. 1−6.

[37] Theobald A, Weikum G. The index-based XXL search engine for querying XML data with relevance ranking. In: Jensen CS, Jeffery KG, Pokorný J, eds. Proc. of the 8th Conf. on Extending Database Technology (EDBT). Prague: Springer-Verlag, 2002. 477−495.

[38] Bremer JM, Gertz M. XQuery/IR: Integrating XML document and data retrieval. In: Fernandez MF, Papakonstantinou Y, eds. Proc. of the 5th Int'l Workshop on the Web and Databases (WebDB). Madison: ACM Press, 2002. 1−6.

[39] Hayashi Y, Tomita J, Kikui G. Searching text-rich XML documents with relevance ranking. In: Proc. of the SIGIR Workshop on XML and Information Retrieval. 2000. http://www.haifa.il.ibm.com/sigir00-xml/final-papers/Hayashi/hayashi.html

[40] Schmidt A, Kersten LM, Windhouwer M. Querying XML documents made easy: Nearest concept queries. In: Young DC, ed. Proc. of the 17th Int'l Conf. on Data Engineering (ICDE). Heidelberg: IEEE Computer Society, 2001. 595−604.

[41] Graupmann J, Schenkel R, Weikum G. The SphereSearch engine for unified ranked retrieval of heterogeneous XML and Web documents. In: Böhm K, Jensen CS, Haas LM, *et al*., eds. Proc. of the 31st Int'l Conf. on Very Large Data Bases (VLDB). Trondheim: ACM Press, 2005. 529−540.

[42] XQuery 1.0 and Xpath 2.0 fulltext. W3C Working Draft 1, 2006. http://www.w3.org/TR/xquery-full-text/

[43] Zhang C, Naughton J, DeWitt D, Luo Q, Lohman G. On supporting containment queries in relational database management systems. In: Aref WG, ed. Proc. of the 2001 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Santa Barbara: ACM Press, 2001. 425−436.

[44] Wang J, Meng XF, Wang S. Structural join of XML based on range partitioning. Journal of Software, 2004,15(5):720−729 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/15/720.htm

[45] Wan CX, Liu YS, Xu SH, Liu XP, Lin DH. Indexing XML data based on region coding for efficient processing of structural joins. Chinese Journal of Computers, 2005,28(1):113−127 (in Chinese with English abstract). http://cjc.ict.ac.cn/qwjs/view.asp?id=1746

[46] Wang J, Meng XF, Wang Y, Wang S. Target node aimed path expression processing for XML data. Journal of Software, 2005, 16(5):827−837 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/16/827.htm

[47] Lu JH, Ling TW, Chan CY, Chen T. From region encoding to extended Dewey: On efficient processing of XML twig pattern matching. In: Böhm K, Jensen CS, Haas LM, Kersten ML, Larson P, Ooi BC, eds. Proc. of the 31st Int'l Conf. on Very Large Data Bases (VLDB). Trondheim: ACM Press, 2005. 193−204.

[48] Chen Y, Davidson SB, Zheng YF. BLAS: An efficient XPath processing system. In: Weikum G, König AC, Deßloch S, eds. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Paris: ACM Press, 2004. 47−58.

[49] Wang W, Wang HZ, Lu HJ, Jiang HF, Lin XM, Li JZ. Efficient processing of XML path queries using the disk-based F&B index. In: Böhm K, Jensen CS, Haas LM, *et al*., eds. Proc. of the 31st Int'l Conf. on Very Large Data Bases (VLDB). Trondheim: ACM Press, 2005. 145−156.

[50] Jiang HF, Lu HJ, Wang W. Efficient processing of XML twig queries with OR-predicates. In: Weikum G, König AC, Deßloch S, eds. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Paris: ACM Press, 2004. 59−70.

[51] Chen T, Lu JH, Ling TW. On boosting holism in XML twig pattern matching using structural indexing techniques. In: Ozcan F, ed. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Baltimore: ACM Press, 2005. 455−466.

[52] Shasha D, Zhang K. Approximate tree pattern matching. In: Apostolico A, Galil Z, ed. In: Proc. of the Pattern Matching Algorithms. Oxford University, 1997.

[53] Bille P. A survey on tree edit distance and related problems. Theoretical Computer Science, 2005,337(1-3):217−239.

[54] Joshi S, Agrawal N, Krishnapuram R, Negi S. A bag of paths model for measuring structural similarity in Web documents. In: Getoor L, Senator TE, Domingos P, *et al*., eds. Proc. of the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (SIGKDD). Washington: ACM Press, 2003. 577−582.

[55] Amer-Yahia S, Koudas N, Marian A, Srivastava D, Toman D. Structure and content scoring for XML. In: Böhm K, Jensen CS, Haas LM, Kersten ML, Larson P, Ooi BC, eds. Proc. of the 31st Int'l Conf. on Very Large Data Bases (VLDB). Trondheim: ACM Press, 2005. 361−372.

[56] Arvola P, Junkkari M, Kekäläinen J. Generalized contextualization method for XML information retrieval. In: Herzog O, Schek H, Fuhr N, *et al*., eds. Proc. of the 2005 ACM CIKM Int'l Conf. on Information and Knowledge Management (CIKM). Bremen: ACM Press, 2005. 20−27.

[57] Wolff JE, Flörke H, Cremers AB. Searching and browsing collections of structural information. In: Proc. of the IEEE Advances in Digital Libraries (ADL 2000). Washington: ACM Press, 2000. 141−150.

[58] Guha S, Jagadish HV, Koudas N, Srivastava D, Yu T. Approximate XML joins. In: Franklin MJ, Moon B, Ailamaki A, eds. Proc. of the 2002 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Madison: ACM Press, 2002. 287−298.

[59] Yang R, Kalnis P, Tung AK. Similarity evaluation on tree-structured data. In: Ozcan F, ed. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Baltimore: ACM Press, 2005. 754−765.

[60] Augsten N, Böhlen MH, Gamper J. Approximate matching of hierarchical data using *pq*-grams. In: Böhm K, Jensen CS, Haas LM, Kersten ML, Larson P, Ooi BC, eds. Proc. of the 31st Int'l Conf. on Very Large Data Bases (VLDB). Trondheim: ACM Press, 2005. 301−312.

[61] Schlieder T, Meuss H. Querying and ranking XML documents. Journal of the American Society for Information Science and Technology, 2002,53(6):489−503.

[62] Kailing K, Kriegel H, Schönauer S, Seidl T. Efficient similarity search for hierarchical data in large databases. In: Bertino E, Christodoulakis S, Plexousakis D, *et al*., eds. Advances in Database Technology-EDBT 2004, Proc. of the 9th Int'l Conf. on Extending Database Technology (EDBT). Greece: Springer-Verlag, 2004. 676−693.

[63] Kotsakis E. Structured information retrieval in XML documents. In: Proc. of the 2002 ACM Symp. on Applied Computing (SAC). Madrid: ACM Press, 2002. 663−667.

[64] Wan CX, Liu YS. Efficient supporting XML query and keyword search in relational database systems. In: Meng XF, Su JW, Wang YJ, eds. Advances in Web-Age Information Management, Proc. of the 3rd Int'l Conf. (WAIM). LNCS 2419, Beijing: Springer-Verlag, 2002. 1−12.

[65] Hristidis V, Papakonstantinou Y, Balmin A. Keyword proximity search on XML graphs. In: Dayal U, Ramamritham K, Vijayaraman TM, eds. Proc. of the 19th Int'l Conf. on Data Engineering (ICDE). Bangalore: IEEE Computer Society, 2003. 367−378.

[66] Ré C, Siméon J, Fernández MF. A complete and efficient algebraic compiler for XQuery. In: Liu L, Reuter A, Whang KY, *et al*., eds. Proc. of the 22nd Int'l Conf. on Data Engineering (ICDE). Atlanta: IEEE Computer Society, 2006. 14.

[67] Zhang SH, Dyreson C. Symmetrically exploiting XML. In: Carr L, Roure DD, IyengarA, *et al*., eds. Proc. of the 15th Int'l Conf. on World Wide Web (WWW). Edinburgh: ACM Press, 2006. 103−111.

[68] Amer-Yahia S, Curtmola E, Deutsch A. Flexible and efficient XML search with complex full-text predicates. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Chicago: ACM Press, 2006. 575−586.

附中文参考文献:

[1] 孟小峰,周龙骥,王珊.数据库技术发展趋.软件学报,2004,15(12):1822−1836. http://www.jos.org.cn/1000-9825/15/1822.htm

[2] 孔令波,唐世渭,杨冬青,王腾蛟,高军.XML 数据索引技术.软件学报,2005,16(12):2063−2079. http://www.jos.org.cn/1000-9825/16/2063.htm

[18] 孟小峰,王宇,王小锋.XML 查询优化研究.软件学报,2006,17(10):2069−2086. http://www.jos.org.cn/1000-9825/17/2069.htm

[44] 王静,孟小峰,王珊.基于区域划分的 XML 结构连接.软件学报,2004,15(5):720−729. http://www.jos.org.cn/1000-9825/15/720.htm

[45] 万常选,刘云生,徐升华,刘喜平,林大海.基于区间编码的 XML 索引结构的有效结构连接.计算机学报,2005,28(1):113−127. http://cjc.ict.ac.cn/qwjs/view.asp?id=1746

[46] 王静,孟小峰,王宇,王珊.以目标节点为导向的 XML 路径查询处理.软件学报,2005,16(5):827−837. http://www.jos.org.cn/1000-9825/16/827.htm

孔令波(1974－),男,博士生,山东日照人,主要研究领域为关系数据库实现技术,XML 数据处理技术,数据挖掘.

唐世渭(1939－),男,教授,博士生导师,CCF 高级会员,主要研究领域为数据库,半结构化数据,Web 数据集成,数据挖掘.

杨冬青(1945－),女,教授,博士生导师,CCF 高级会员,主要研究领域为数据库,数据仓库,Web 数据集成,移动数据挖掘.

王腾蛟(1973－),男,博士,副教授,CCF 高级会员,主要研究领域为数据库,数据仓库,Web 数据集成,数据挖掘.

高军(1975－),男,博士,副教授,CCF 高级会员,主要研究领域为数据库,数据仓库,半结构化数据,Web 数据集成,移动数据挖掘.