基于关键字之间结构关系的XML查询结果排序方法

任建华[1] 周 建[2] 孟祥福[1] 魏 珂[1]
(辽宁工程技术大学电子与信息工程学院 葫芦岛 125105)[1] (沈阳炮兵学院通信指挥系 沈阳 111000)[2]

摘 要 非空结果的XML关键字查询中,多个查询关键字之间必然存在联系,这种联系可以通过SLCA(最紧致片段)的结构关系获得。基于SLCA的结构关系,提出了一种推测多个关键字内在联系的XML关键字查询结果排序方法:通过LISA II算法获得SLCA;根据SLCA的结构信息推测出各个关键字之间的内在结构关系,得到所有关键字组成的关系树;然后根据关系树中各关键字对查询结点的严格程度得到对应SLCA的重要程度,据此得到有序的SLCA并输出。该方法利用了XML文档的结构信息对查询结果进行排序。实验结果和分析表明,提出的方法具有较高的准确率,能够较好地满足当前用户的需求和偏好。

关键词 关键字查询,SLCA,小枝查询,结果排序,准确率
中图法分类号 TP311.131 **文献标识码** A

Results Ranking Approach of XML Keyword Search Based on Keyword's Structural Relationships

REN Jian-hua[1] ZHOU Jian[2] MENG Xiang-fu[1] WEI Ke[1]
(School of Electronic and Information Engineering, Liaoning Technical University, Huludao 125105, China)[1]
(Communication Department, Shenyang Artillery Academy, Shenyang 111000, China)[2]

Abstract If the answer of an XML multi-keywords search is not empty, there would be some specific relationships between these keywords and such relationships can be speculated by SLCA (the smallest lowest common ancestor). This paper proposed an XML keywords query results ranking approach based on these relationships: the approach obtains the SLCAs by the LISA II algorithm, leverages the structures of SLCAs to speculate the interior structural relationships of keywords and to obtain the relationship tree. Then, the importance of each SLCA can be estimated by the strict degree of keywords to the query node in the relationship tree. The SLCAs are ranked according to their importance and the ordered SLCAs are treated as the ranked XML keywords query results. The experimental results demonstrate that the approach presented in this paper has the high precision, and can efficiently meet the user's needs as well.

Keywords Keywords search, SLCA, Twig query, Results ranking, Precision

目前,XML查询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结构化查询和XML关键字查询。以小枝查询为代表的XML结构化查询可以得到比较精确的查询结果,但是它对用户有很高的要求,也就是用户既要知道XML的结构模式,还要掌握复杂的语法,这对普通用户来说有着较大的难度。相对而言,XML关键字查询比较灵活,是一种友好便捷的查询方式,用户不需要额外学习复杂的查询语言,也不需要深入了解查询信息的内部底层结构,只需要提供相关内容的关键字就可以实现数据的检索,因此该模式被广泛采用,有着重要的研究意义。

需要指出的是,关键字查询有一个严重的不足,即查询结果的不准确性。根据文献[1]中的调查统计,用户使用关键字搜索时,大概有10%~15%的查询存在错误,40%~51%的查询要经过修改才能获得所需的信息,最典型的问题就是返回的查询结果通常是整篇XML文档。

例如,图1为一个Ford汽车信息的XML文档的树形结构,在此文档中进行关键字为"Ford,¥150000"的查询,用户的实际意图是查找价值为¥150000的福特车的信息,得到了两个SLCA(smallest lowest common ancestor,最近最小公共祖先),即SLCA1://Ford[Focus][¥150000]和SLCA2://Ford/Mazada[Mazada6][¥150000]。很明显,SLCA1符合用户的查询意图,而SLCA2表示的含义是Ford公司所生产的价格为¥150000的Mazada6。换句话说,相对于SLCA1而言,SLCA2是不准确的。本文的排序方法将会解决此类问题,使上例中SLCA1对应的查询结果先于SLCA2返回。

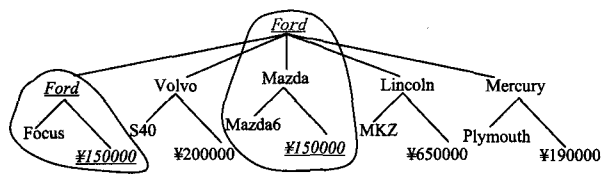

图1 关键字查询得到的SLCA

到稿日期:2012-08-09 返修日期:2012-11-14 本文受国家青年科学基金项目(61003162)资助。

任建华(1973-),男,硕士,副教授,硕士生导师,主要研究方向为数据库理论,E-mail:renjh4665@163.com;周 建(1975-),男,硕士,讲师,主要研究方向为计算机应用、XML数据库;孟祥福(1981-),男,博士,讲师,主要研究方向为Web数据库、XML数据个性化柔性查询技术;魏 珂(1986-),男,硕士,主要研究方向为XML数据库查询技术。

目前，关键字查询结果的准确性已经成为研究的热点，近期已有一些关于关键字查询准确率方面的研究工作[2-6]：文献[2]针对用户使用关键字查询时不能准确表达其真实意图，或者即便是正确表达了查询意图而查询引擎也可能不能准确地返回查询结果的问题，对 XML 关键字查询进行了有效的查询改写，并生成有意义的结果。文献[3]为了推测关键字查询的返回规格，开发了一个 XML 关键字搜索引擎——XSeek，其能够识别数据中的实体和属性，并且能区分查询谓词和返回规格，从而有效地推测查询语义和判定返回结点。文献[4]提出了一种基于有向无环图的索引来有效组织大量基于关键字的查询，以降低查询匹配的代价，并根据数据流的特点，提出了一种基于栈的临时结果缓存方法来过滤大量与查询无关的数据结点。文献[5]设计了一个基于语义相关的 XML 关键字搜索引擎——Ropeway，它利用 XML 数据包含的语义信息，分析了 XML 数据的语义和用户信息需求，推断用户的查询意图，提高了检索的质量。文献[6]提出一种基于 XLCA 的 XML 关键字搜索方法，在 SLCA 的基础上提出了基于 XLCA 的结果集定义，并给出了一种精简的索引结构和相应的 BuP 算法，其能够有效找出结果集。

当前的关于 XML 关键字查询的排序方法一般是通过分析 TD * IDF、LCA(lowest common ancestor)、查询结构化，或者是通过查询松弛来实现计分功能，从而得到 top-k 排序结果[7-10]。文献[7]提出了 XPRAM 方法，即将松弛与序列匹配过程相结合，在匹配过程中使用边泛化、叶子删除、子树提升的松弛方法来获得 top-k 结果。文献[8]提出了一种关键字 top-k 查询方法，该方法通过约束阈值的策略跳过不符合用户偏好的文档，并且基于中间结果在每一个访问过的文档删剪侯选结果。文献[9]提出了一种 IR-style 方式，其主要利用潜在的 XML 数据统计来解决文本数据库和 XML 数据库导致的 3 个问题：(1)用户查询意图的鉴定；(2)关键字歧义问题；(3)计分功能的实现。文献[10]提出了 XML 数据库中支持 top-k 的关键字查询方法，将关键字查询评估降低到关系的合并范围，并引入了关键字 top-k 合并的思想。

本文提出一种基于 SLCA 推测关键字之间结构关系的 XML 关键字查询结果排序方法。首先通过 LISA II 算法求出 SLCA，在 SLCA 的结构中删除所有的非关键字结点，得到每个 SLCA 中所有关键字的结构关系。本文将其定义为关系树，通过关系树可以量化出每个 SLCA 中所有关键字对查询结点的严格程度，根据严格程度推测出各个 SLCA 的重要程度，并据此对所有 SLCA 进行排序，最后输出有序的 SLCA 作为查询结果，从而使得重要的 SLCA 比相对不重要的 SLCA 优先返回。

本文第 1 节介绍关键字查询结果排序方法的相关定义；第 2 节给出关键字查询结果排序解决方案；第 3 节提出 XML 关键字查询结果排序方法的具体实现算法；第 4 节描述实验过程并讨论实验结果；最后总结全文。

1 相关定义

定义 1(SLCA，smallest lowest common ancestor) 即最近最小公共祖先，它包含所有关键字的最小 XML 片段(最紧致片段)。

给定一棵标签有向树 $T=(r,V,E,A)$ 以及一组关键字 $K=(k_1,k_2,\cdots,k_n)$，其中，r 表示数据树的根结点，V 表示树中的所有结点的集合，E 表示所有边的集合，A 代表所有结点标签的集合，SLCA 的结点满足如下条件[11]：

(1)SLCA 或 SLCA 的后裔结点包含所有的关键字；

(2)SLCA 中的任意子树都不可能包含全部关键字。

定义 2(扩展 Dewey 编码) 给定对应 XML 数据的标签有向树 $G=(V,E,r,A)$，G 中任意结点的扩展 Dewey 编码由下列规则确定：

(1)根结点 r 的 Dewey 编码为 0；

(2)在前序遍历 G 的过程中，如果结点 v 是结点 u 的孩子结点，前序结点编码表示为 $pre()$ 函数，那么，结点 v 的 Dewey 码为 $D(u).pre(v)$。其中的 $D(u)$ 表示结点 u 的扩展 Dewey 码[12]。

定义 3(关系树) 在 SLCA 中删掉所有非关键字结点，从而形成包含所有关键字的树形结构。由 SLCA 得到关键字的关系树原则为：

(1)SLCA 中两个关键字相对位置为父子关系，则两个关键字在关系树的关系为父子关系。

(2)SLCA 中两个关键字相对位置为祖先-后代关系，则两个关键字在关系树的关系为祖先-后代关系。

2 关键字查询结果排序方法整体框架

本文提出的基于 SLCA 推测关键字结构关系的关键字查询结果排序方法主要包括 4 个步骤：

(1)SLCA 的获取：SLCA 的获取是 XML 关键字查询方式中最关键也是最基本的问题。

本文通过 LISA II 算法来获得 LSCA。文献[13]中的实验证明，LISA II 算法是 Stack 算法、ILE 算法、SE 算法和 LISA 算法，以及 LISA I 和 LISA II 算法等几种求解 SLCA 的经典算法中性能最好的算法。

(2)求关系树：基于 SLCA 的结构信息可以反映出所有关键字的关系。

本文方法删除 SLCA 中的所有非关键字结点，并将关键字间的位置关系用父子关系和祖先后代关系体现出来，从而得到所有关键字之间的结构关系。

(3)SLCA 重要程度的计算：通过对关系树中关键字之间的父子关系或祖先后代关系进行量化，得到对应的所有 SLCA 的重要程度。

(4)SLCA 的排序：根据每个 SLCA 对应的重要程度，对 SLCA 进行排序，并有序输出 SLCA 作为关键字查询结果。

各步骤的实现方法分别在后文给出。图 2 给出了本文所提方法的整体框架。

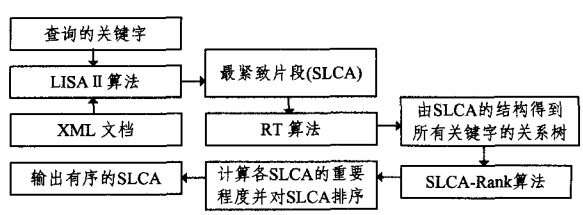

图 2 关键字查询结果排序方法的整体框架

3 基于 SLCA 结构的关键字查询结果排序方法

本节首先介绍 LISAII 算法，然后描述关键字之间关系树

的获取方法，最后介绍 SLCA 的排序方法。

3.1 LISA II 算法

典型的关键字查询一般分两个阶段进行：首先根据关键字找到匹配结点集，然后以包含所有关键字匹配结点的最小子树作为查询结果。

本文通过 LISA II 算法[14]获得 SLCA，LISA 算法基于 SLCA 结点按“层”分布的规律，采取逐层求解 SLCA 结点的思路。LISA II 采用扩展 Dewey 编码，解决了 Dewey 集合交操作时需要扫描待操作的两个结点的 Dewey 码前缀的问题，因此只需考虑两个结点在该层的编码即可。

通过 LISA II 算法可以得到所有包含全部关键字的 SLCA，具体算法如算法 1 所示。

算法 1 LISA II 算法

Input: Dewey code set D_i for each keyword k_i

Output: The SLCA list

1. Transform all Dewey codes into integer sequences, still store them according to keywords, marked as IS_i
2. Compute $maxL_i(IS_i)$ for each IS_i
3. $minmaxL=\min\{maxL_i \mid 1\leqslant i\leqslant k\}$
4. v={}; tmp={};
5. For j=minmaxL−1 to 2
6. While no IS_i is null do {
7. Build integer set $D_i'(1\leqslant i\leqslant k)$ corresponding level j and sort it, D_1' is the smallest set;
8. tmp=D_1';
9. For i=2 to k
10. tmp=Inter(v,D_i');
11. v+=tmp;
12. Delete integer sequences, which have integers contained in v, from each IS_i
13. }
14. Return Dewey codes corresponding to v.

例如，按照 LISA II 算法在图 3 的 XML 文档中进行关键字为“Steven, XML, DTD”的关键字查询，可以得到两个 SLCA（如图 3 中圆圈内所示）。

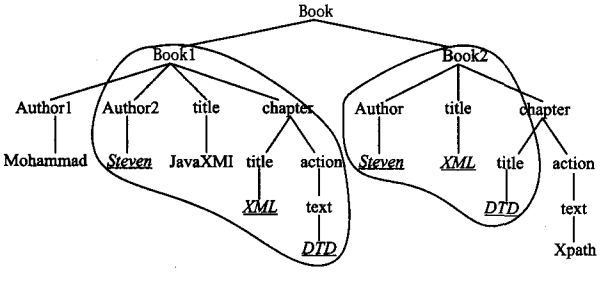

图 3 LISA II 算法得到两个的 SLCA

3.2 所有关键字关系树的获取方法

通过 LISA II 算法已得到所有关键字的 SLCA，根据 SLCA 的结构关系，可以得到其中所有关键字的相对位置关系，形成关系树。

由于 SLCA 完全反映了用户输入关键字在查询对象中的所有结构信息，因此 SLCA 也包括了所有关键字的结构信息。删除 SLCA 中所有的非关键字结点，可以得到结点为所有关键字的树形结构。此结构可以通过父子关系和祖先后代关系的位置关系反映出所有关键字之间的关系。

由于在 LISA II 算法中使用扩展的 Dewey 编码，因此判断任意两个关键字是否是祖先-后代关系，只需判断其中的一个编码是否是另一个编码的前缀即可；对于父子关系，只需在祖先后代关系判断的基础上进一步判断它们之间向量元素的个数是否相差为 1。关系树的具体获取算法如下所示（算法 2）。

算法 2 RT (Relationship Tree) 算法

Input: The SLCA list S

Output: Relationship Trees

1. for each SLCA∈S do
2. read the SLCA;
3. delete the non-keywords nodes;
4. for each k_i
5. create a relationship tree;
6. return relationship trees;

按照 RT 算法，图 3 中的两个 SLCA 可以转换成两个关系树（见图 4）。

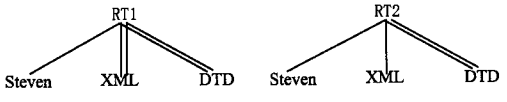

图 4 SLCA 转化的小枝查询

3.3 SLCA 排序方法

在关键字的关系树中，各个关键字结点对查询结点的严格程度直接反映出关系树的重要程度，关键字结点对查询结点 Q 的要求越严格，对应的查询结果应该越符合用户的查询意图。

定义 4（关系树的严格程度） 根据关系树中关键字结点的结构关系，可以得出结构对查询结点 Q 的限制程度，即关键字的严格程度。关键字严格程度的判定主要基于以下直觉：

(1)关系树中各关键字结点对查询结点 Q 的要求越严格，所对应的查询结果就越准确，即要求严格的查询比要求相对不严格的查询更贴近用户的查询意图（因为用户提出的查询要求越具体越严格，说明用户对自己查询意图的描述越准确），此类的关系树对应的 SLCA 也就越重要。

(2)在同一层次上的结点分支越多，对此结点的要求越严格，则在总体上对查询结点的要求也越严格。

(3)父子关系的查询要求要比祖先-后代关系的查询要求严格。

举例来说，图 5 中 3 种常见的关系树类型中(a)和(b)要比(c)查询要求严格，因为(a)和(b)有两个查询条件 k_1 和 k_2 来限制查询结点 Q，(c)只有一个查询条件 k_1/k_2 来限制 Q，其中 k_2 用来限制 k_1；而查询中(a)要比(b)查询要求严格，因为(a)要求 k_1 必须是 Q 的孩子，(b)只要求 k_1 是 Q 的后代即可，一般意义下后代包含孩子。

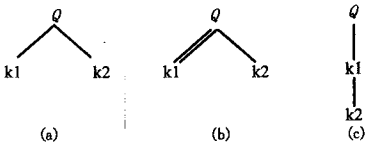

图 5 3 种常见的关系树类型

在关系树中，不同位置的结点对查询结点 Q 的要求是不同的，因此关系树中各结点的重要程度也是不同的。一般来

讲，关系树中查询结点的子结点是最重要的，沿关系树的树形结构中父亲-孩子或祖先-后代的方向，结点的重要程度依次降低。如图 6 所示，结点的重要程度为 $a_1>a_2$，$b_1>b_2>b_3$，$c_1>c_2>c_3$。

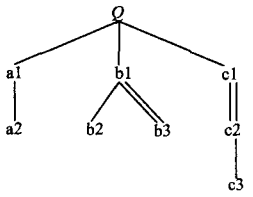

图 6　关系树

设查询结点 Q 的重要程度为 1，则中间结点的重要程度小于 1。设 father=a * childhood，ancestor=b * offspring，其中 a，b 分别为父亲-孩子或祖先-后代重要程度的递减率，则中间结点的重要程度为：

(1)父亲-孩子关系：$p_n=a*p_{n-1}$。其中，p_n 是 p_{n-1} 的孩子。如图 6 中：$a_1=a*1=a$。

(2)祖先-后代关系：$p_n=b*p_{n-1}$。其中，p_n 是 p_{n-1} 的后代。如图 6 中：$c_3=b*(a*1)=b*a$。

按上述方法，可以计算出关系树中每一个结点的重要程度。

定义 5(SLCA 的重要程度)　关系树中所有结点的重要程度之和作为整个关系树的重要程度，即关系树对应的 SLCA 的重要程度。

SLCA 的重要程度可以反映出用户对此 SLCA 的偏好程度，SLCA 的重要程度越高，越符合用户的查询要求，也就越是用户想要早些得到的。因此，此类 SLCA 应尽早返回给用户。本文使用 SLCA 排序(SLCA-Rank)算法，对 SLCA 的重要程度进行计算，并对所有的 SLCA 进行排序，使得越重要的 SLCA 作为查询结果越早返回。SLCA-Rank 算法见算法 3。

算法 3　SLCA-Rank 算法

```
Input: Relationship Trees T
Output: Ranked SLCAs list
1. for each Ti∈T do
2.    read the Ti;
3.    for each Ti
4.       calculate the important degree of each node according to the
         strict degree of each Ti;
5.          if the relationship of the two node is father-childhood
              father=a * childhood
6.          else if the relationship of the two node is ancestor -offspring
              ancestor=b * offspring
7.       add all the node's important degree of the relationship tree;
8. ranking SLCAs according to their important degree
9. return the ranked SLCAs list.
```

4　实验

本节主要介绍实验环境及测试数据，描述实验方法和实验结果并加以分析。

4.1　实验环境和测试数据

本次实验是在 Intel(R) Core(TM) i3 CPU M 380@2.53GHz，2GRAM，500G 硬盘和 Microsoft Windows XP 操作系统下进行的，由 IBM XML data generator[15] 产生测试文档，生成一个 XML 数据集：

数据集 BookDB. xml：从 Amazon 网站[16] 随机获得 50000 条图书记录生成 BookDB. xml 数据集(见图 7)，其最大深度为 6，包含 50000 个 Book 元素。

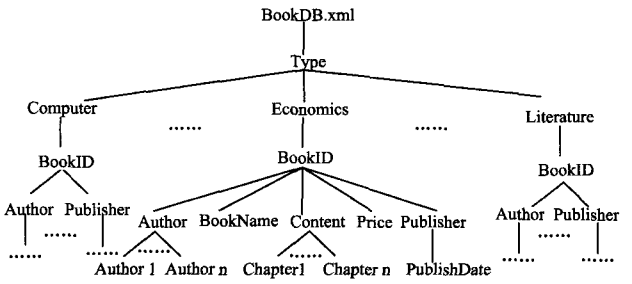

图 7　数据集 BookDB. xml

基于以上数据集，用 Java 语言和 eclipse3.0 编译工具建立了 Web 查询系统。

4.2　查询松弛方法验证

该实验对查询系统进行测试，并对系统的使用者进行用户调查，以验证本文基于 SLCA 推测关键字之间关系的 XML 关键字查询结果排序方法的准确率。准确率(Precision)指查询结果中与用户真实查询意图相关元素的比率[17]。在下面的实验中邀请 10 位同事、学生和朋友作为测试者根据各自需求和偏好，在 BookDB. xml 上分别提出 1 条关键字测试查询(见表 1)。

表 1　数据集 H_i 上的 10 条测试查询

Query ID	查询包含的关键字
Q_1	Joshua, Java, Jsp
Q_2	Frank J. Fabozzi, Market analysis
Q_3	Enterprise, Management, Sun Jian, Zhao Tao
Q_4	Linux, Shell, Sobel
Q_5	The Adventure of Tom Sawyer, Mark Twain
Q_6	Economic, financial, accounting
Q_7	Lippmann, C++, primer
Q_8	Photoshop cs3, computer
Q_9	Guo Taiming, Foxconn, Xu Mingtian
Q_{10}	C#, karliWatson, USA

在进行查询结果的准确率测试时，要求测试者从包含 50000 个 Book 元素的 BookDB. xml 数据集中逐条查找所有与其真实意图匹配的 Book 元素是不现实的。因此针对各条测试查询，取 BookDB. xml 数据集中的 100 条与查询相关的或不相关的适当元素生成对应的小数据集 H_i；然后在本文查询结果的准确率验证阶段，只需对比在数据集 H_i 上利用本文排序方法的查询结果与测试者在数据集 H_i 中标示出的满足用户真实查询意图元素的相同个数，与之相同的结果个数越多，则查全率就越高，查询结果中与户查询意图相同的元素个数越多，则准确率也就越高。

例如，在表 1 中的关键字查询 Q_1 包含 3 个关键字：Joshua，Java 和 Jsp。根据本文方法，在数据集 BookDB. xml 的子集 H_1 上首先通过 LISAII 算法计算出包含 3 个关键字的所有的最紧致片段(SLCA)，使用 RT 算法在各个 SLCA 的树形结构中通过删除非关键字结果得到 3 个关键字之间是祖先-后代关系或是父子关系，生成包含 3 个关键字及其关系的关系树：Q_1 [Joshua]//Java//Jsp，Q_1 [Joshua][Java]//Jsp，Q_1 [Joshua] [//Java]//Jsp(见表 2)。

表 2　测试查询生成的关系树

Query ID	RT 算法生成的包含所有关键字的关系树		
Q_1	Q_1[Joshua] [Java] //Jsp	Q_1[Joshua] [//Java] //Jsp	Q_1[Joshua] //Java//Jsp
Q_2	Q_2[Frank J. Fabozzi Joshua] /Market analysis	Q_2/Market analysis /Frank J. Fabozzi Joshua	——
Q_3	Q_3[Sun Jian] [Zhao Tao] [Management]// Enterprise	Q_3[Sun Jian] [Zhao Tao] [//Management]// Enterprise	Q_3[Sun Jian] [Zhao Tao] //Management// Enterprise
Q_4	Q_4[Sobel] [Linux] //Shell	Q_4[Sobel] [//Linux] //Shell	Q_4[Sobel] //Linux//Shell
Q_5	Q_5[Mark Twain Joshua]/ Adventure of Tom Sawyer	Q_5[Adventure of Tom Sawyer]// Mark Twain Joshua	Q_5/Adventure of Tom Sawyer/ Mark Twain Joshua
Q_6	Q_6[Economic] [financia]/ accounting	Q_6/Economic [financia]/ accounting	Q_6/Economic/ financia/ accounting
Q_7	Q_7[Lippman] [C++]//primer	Q_7[Lippman] [primer]// C++	Q_7[Lippman] primer// C++
Q_8	Q_8/computer/Photoshop cs3	Q_8/Photoshop cs3//computer	——
Q_9	Q_9[Xu Mingtian] [Foxconn]/ Guo Taiming	Q_9[Xu Mingtian] [//Foxconn]//Guo Taiming	Q_9[Xu Mingtian] //Foxconn//Guo Taiming
Q_{10}	Q_{10}[C#] [karliWatson]/ USA	Q_{10}[karliWatson] [USA]//C#	——

在表 2 中得到的关系树中使用 SLCA-Rank 算法对关系树包含的所有关键字的关系进行量化(量化值见表 3)，可以得到 SLCA 对应的所有关键字的重要程度。由于 $0<b<a<1$，$a^2<b$，因此很容易判断两个关系树的重要程度的大小。关系树的重要程度即为 SLCA 的重要程度，因此可以对所有的 SLCA 进行排序，最后输出有序的 SLCA。

表 3　关键字重要程度的量化结果

Query ID	关系树中所有关键字重要程度的量化结果		
Q_1	a+a+a	a+b+b	$a+b+b^2$
Q_2	a+a	$a+a^2$	——
Q_3	a+a+a+b	a+a+b+b	$a+a+b+b^2$
Q_4	a+a+b	a+b+b	$a+b+b^2$
Q_5	a+a	a+b	$a+a^2$
Q_6	a+a+a	$a+a^2+a$	$a+a^2+a^3$
Q_7	a+a+b	a+a+b	a+a+ab
Q_8	$a+a^2$	a+ab	——
Q_9	a+a+a	a+b+b	$a+b+b^2$
Q_{10}	a+a+a	a+a+b	——

本实验的目的是为了验证本文方法的准确性，因此考察在数据集 H_i 中由本文关键字查询方法返回的查询结果与测试者标示出相关结果的相同个数。

将本文算法中得到的前 k 个输出结果与测试者标示出的 k 个相关结果进行对比，得到的比率为准确率，定义为：

$$Precision=\frac{|\text{top-}k\ \text{query resluts}\cap\text{marked resluts}|}{k}$$

为了充分显示本文查询算法准确率的提高，本实验将只经过 LISAII 算法得到的查询结果与本文方法得到的前 10 个查询结果的准确率进行对比，如表 4 所列。

表 4　LISAⅡ算法与本文方法查询结果准确率的对比

Query ID	LISAⅡ算法结果/标示结果	准确率	本文算法结果/标示结果	准确率
Q_1	19/24	80%	24/24	100%
Q_2	15/17	90%	17/17	90%
Q_3	5/5	90%	5/5	90%
Q_4	16/16	60%	16/16	90%
Q_5	4/13	80%	6/13	100%
Q_6	9/13	80%	11/13	100%
Q_7	4/8	60%	7/8	90%
Q_8	11/28	50%	25/28	90%
Q_9	13/15	70%	13/15	80%
Q_{10}	8/19	70%	12/19	100%

表 4 中的实验数据表明，只经过 LISAII 算法得到的查询结果与本文方法得到的前 10 个查询结果的平均准确率分别为 73%和 93%，充分证明了本文方法返回的查询结果的准确率得到了较大提高，能够较好地满足用户偏好。

结束语　本文提出了一种基于 SLCA 推测关键字间关系的 XML 关键字查询结果排序方法。该方法通过 LISA II 算法获得 SLCA，根据 SLCA 得到所有关键字之间的结构关系，通过关系树量化出每个 SLCA 中所有关键字对应关系树的重要程度，从而得到 SLCA 的重要程度，并以此对所有的 SLCA 进行排序，最后输出有序的 SLCA 作为查询结果。实验结果表明，本文方法提高了关键字查询的准确率。算法效率的提升将是下一步研究的重点。

参 考 文 献

[1] Spink A, Jansen B J, Wolfram D, et al. From e-sex to e-commerce: web search changes [J]. IEEE computer, 2002, 35(3): 107-109

[2] 黄静，陆嘉恒，孟小峰. 高效的 XML 关键字查询改写和结果生成技术[J]. 计算机研究与发展，2010，47(5)：841-848

[3] Liu Z, Chen Y. Identifying meaningful return information for XML keyword search [C]// Proceedings of the ACM SIGMOD Conference. 2007：329-340

[4] 周军锋，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(5)：392-397

[5] 郭文琪，温馨，王鹏，等. Ropeway：基于语义相关的 XML 关键字搜索引擎[J]. 计算机研究与发展，2010，47(Suppl.)：470-474

[6] 许建军，汪卫，施伯乐. 一种基于 XLCA 的 XML 关键字搜索方法[J]. 小型微型计算机系统，2008，29(1) 52-56

[7] Li L, Lee M L, Hsu W E, et al. A prüfer based approach to process top-k queries in XML [C]// Proceedings of the DEXA Conference. 2009：348-355

[8] Li J X, Liu C F, et al. Efficient top-k search across heterogeneous XML data sources [C]// Proceedings of the DASFAA Conference, LNCS 4947. 2008：314-329

[9] Bao Z F, Ling T W, Chen B, et al. Effective XML keyword search with relevance oriented ranking [C]// Proceedings of the ICDE Conference. 2009：517-528

[10] 张雷. XML 关键字查询中最紧致片段问题的研究[D]. 济南：山东大学，2009

[11] Sun C, Chan C Y, Goenka A K. Multiway SLCA-based keyword search in XML data [C]// International World Wide Web Conference Committee (IW3C2). 2007：1043-1052

[12] Lu J, Ling T W, Chan C Y, et al. From region encoding to extended Dewey: on efficient processing of XML twig pattern matching [C]// Proceedings of the VLDB Conference. 2005：193-204

(下转第 214 页)

采用随机免疫策略和人工免疫策略且人工免疫率取 $\alpha=0.05$ 时，病毒感染密度随时间的变化过程如图 6 所示。

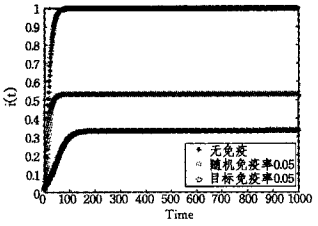

图 6　病毒在采用随机免疫与目标免疫策略时的传播情况

由图 6 可以看出，在相同的人工免疫率下，与采用随机免疫策略相比，采用目标免疫策略时病毒的传播速度更低，被感染节点的数目也少得多。

6.3　结果分析

由于免疫状态不能自然获得，因此可以通过注射疫苗等方式进行提前预防。仿真结果如图 4－图 6 所示，可以看到当进行前期免疫时，是可以避免大规模疾病传播的，感染人数比例迅速降低了，而当采用目标免疫时染病人数比例比之前所有仿真结果都要低。为使疾病在爆发时能得到有效控制，根据仿真结果给出以下几种方案：

方案 1　采取单纯的人工免疫措施，对重点目标（如公众人物、学校师生、医疗从业人员等）进行目标免疫接种，或通过随机免疫大规模进行免疫接种。

方案 2　采取单纯的隔离措施，如学校停课、运输站停止运营等手段限制人口自由流动，对疑似病例进行隔离。

方案 3　采取人工免疫和隔离等多种措施。

结束语　无标度特性是许多现实复杂网络的一个重要特性，本文对一种带有人工免疫作用的 SIRS 类传染病模型在 BA 无标度网络上的传播进行理论分析和仿真研究，从初始感染节点和免疫控制策略两个方面研究了 BA 无标度网络中的病毒传播特性，研究发现网络拓扑结构对病毒的传播有影响，初始感染结点的度越大病毒传播越迅速；引入人工免疫能够有效抑制病毒的传播，而目标免疫策略比随机免疫策略在抑制病毒传播上更为有效。根据这些结论，在疫情发生时，及时收集和发布疫情传播情况，让人们主动切断与感染人群的接触；积极进行人工免疫，提高群体的人工免疫率；限制那些社交广泛的公众群体的频繁流动以及与他人的接触等都是控制疾病传播的有效措施。

参 考 文 献

[1] Poletti P, Ajelli M, Merler S. The effect of risk perception on the 2009 H1N1 pandemic influenza dynamics[J]. PloS one, 2011, 6(2):

[2] 陈端兵，黄晨，尚明生. 复杂网络模型及其在疫情传播和控制中的应用研究[J]. 计算机科学，2011，38(6)

[3] 王亚奇，蒋国平. 复杂网络中考虑不完全免疫的病毒传播研究[J]. 物理学报，2010，59(10)

[4] Kermark M D, Mckendrick A G. Contributions to the mathematical theory of epidemics [J]. Proc Roy Soc, A: Part I, 1927, 115(5): 700-721

[5] Nian F, Wang X. Efficient immunization strategies on complex networks[J]. Journal of Theoretical Biology, 2010, 264(1): 77-83

[6] Kuperman M, Abramson G. Small-world effect in an epidemiological model [J]. Phys Rev Lett, 2001, 86(14): 2909-2912

[7] Stehle J, Voirin N, Barrat A, et al. Simulation of an SEIR infectious disease model on the dynamic contact network of conference attendees[J]. BMC Medicine, 2011, 9: 87

[8] Parshani R, Carmi S, Havlin S. Epidemic Threshold for the Susceptible-Infectious-Susceptible Model on Random Networks [J]. Physical Review Letters, 2010, 104(25): 258701

[9] Costa L F, Oliverira O N, Travieso G, et al. Analyzing and modeling real-world phenomena with complex networks: A survey of applications[J]. Advances in Physics, 2011, 60(3): 329-412

[10] Boccaletti S, Latora V, Moreno Y, et al. Complex networks: Structure and dynamics[J]. Physics Reports, 2006, 424: 175-308

[11] Watts D J, Strongatz S H. Collective Dynamics of Small-World Networks [J]. Nature, 1998, 393(6): 440-442

[12] Barabási A L, Albert R. Emergence of scaling in random networks [J]. Science, 1999, 286(10): 509-512

[13] Castellano C, Fortunato S, Loreto V. Statistical physics of social dynamics[J]. Reviews of Modern Physics, 2009, 81(2): 591-646

[14] Cohen R, Havlin S. Scale-free networks are ultrasmall[J]. Phys. Rev. Lett., 2003, 90(5): 058701

[15] 汪小帆，李翔，陈关荣. 复杂网络理论及其应用[M]. 北京：清华大学出版社，2006：23-40

[16] Fronczak A, Fronczak P, Holyst J A. Mean-field theory for clustering coefficients in Barabási- Albert networks [J]. Phys. Rev. E, 2003, 68: 046126

[17] Dorogovtsev S N, Medes J F F, Samukhin A N. Structure of growing networks with preferential linking [J]. Phys. Rev. Lett., 2000, 85: 4633-4636

[18] 李光正，史定华. 复杂网络上 SIRS 类疾病传播行为分析[J]. 自然科学进展，2006，16(4)：509-512

[19] Moreno Y, Pastor-Satorras R, Vespigani A. Epidemic out breaks in complex heterogeneous networks [J]. European Phys J B, 2002, 26: 521-529

[20] Pastor-Satorras R, Vespigani A. Epidemic dynamic and endemics states in complex networks [J]. Phys Rev E, 2001, 63: 066117

[21] Shi H J, Duan Z S, Chen G R. An SIS model with infective medium on complex networks[J]. Physica A, 2008, 387: 2133-2144

[22] Barthelemy M, Barrat A, Pastor-Satorras R, et al. Velocity and hierarchical spread of epidemic outbreaks in scale-free networks [J]. Physical Review Letters, 2004, 92(17): 178701

（上接第 182 页）

[13] Xu Y, Papakonstantinou Y. Efficient keyword search for smallest LCAs in XML database [C] // Proceedings of the ACM SIGMOD Conference. 2005: 776-787

[14] 孔令波，唐世渭，杨冬青，等. XML 信息检索中最小子树根结点问题的分层算法[J]. 软件学报，2007，18(4)：919-932

[15] IBM Corporation XML data generator [EB/OL]. http://www.alphaworks.ibm.com/tech/xmlgenerator, 2010-08

[16] http://www.amazon.cn/[EB/OL]. 2010-06

[17] Su W, Wang J, Huang Q, et al. Query result ranking over e-commerce web databases [C] // Proceedings of the ACM CIKM Conference. 2006: 575-584