doi:10.3969/j.issn.1005-3026.2016.07.006

模糊 XML 关键字查询方法

李 婷,马宗民
(东北大学 计算机科学与工程学院,辽宁 沈阳 110819)

摘 要:在实际应用中数据经常存在不确定性和模糊性,因而对模糊 XML 数据的关键字查询处理成为一种非专业用户的需求.针对模糊 XML 数据的关键字查询方法进行研究,对模糊 XML 的关键字查询语义进行分析,提出一种新的模糊 XML 文档的编码方法 CDewey,该编码方法能够有效地对节点类型进行区分.在此基础上,提出关键字查询算法 FIndex Loop,该算法能够准确求解输入关键字的 SLCA 结果及结果的可能性值,最后通过实验表明此查询方法的有效性.

关 键 词:模糊 XML;关键字;索引;查询;可能性
中图分类号:TP 311 **文献标志码**:A **文章编号**:1005-3026(2016)07-0937-05

Keyword Querying of Fuzzy XML

LI Ting, MA Zong-min
(School of Computer Science & Engineering, Northeastern University, Shenyang 110819, China. Corresponding author: LI Ting, E-mail: kitehyabc@163.com)

Abstract: In the practical application, there often exists uncertainty and ambiguity in the data. Keyword query processing over fuzzy XML data becomes a requirement for non professional users. Aiming at making a research on the method of keyword querying over fuzzy XML data, the semantics of keyword querying over fuzzy XML was analyzed, and a new coding method CDewey for the fuzzy XML document was proposed. Types of nodes could be effectively distinguished by this coding method. On the bases, a keyword query algorithm FIndex Loop was proposed, this algorithm can get the SLCA results of keywords inputted and values of possibilities of the results accurately. Finally, experimental results showed the effectiveness of the query method.

Key words: fuzzy XML; keyword; index; query; possibility

随着网络技术的快速发展,XML 已经成为 Web 中数据表示和转换的标准,XML 查询技术也就成为信息检索领域中一个重要的研究课题.XML 关键字查询是一种用户友好的查询方式,用户只需要提交一个或者几个关键字信息即可得到相关的检索结果.基于 XML 数据模型,研究者们提出了许多关键字查询语义和方法,例如 SLCA,XRank 等[1-2],其中 SLCA(最小最低公共祖先)语义得到了学术界的广泛关注.

不精确性和不确定性广泛存在于许多 Web 应用中,例如信息整合、信息提取和 Web 数据过滤等,而 XML 数据模型的柔性特点可以很好地表达不精确和不确定数据[3].考虑到现实世界事物中的不确定性特征,许多研究者对概率 XML 数据进行了研究,设计和分析了概率 XML 数据模型[4].基于概率 XML 数据模型,一些研究者们提出了概率 XML 数据上的关键字查询方法,文献[5]提出一个更加广泛的概率 XML 数据模型 PrXML$^{\{exp,ind,mux\}}$ 来表达数据间的概率分布关系,给出了得到 SLCA 节点的相关算法,通过使用子树的关键字分布概率表 kdptabs 来计算点积、笛卡尔积等运算,并得出相关的概率值计算结果.

基于模糊 XML 数据模型,现有的研究成果主要集中在模糊 XML 结构查询方面[6-8].文献

收稿日期:2015-04-15
基金项目:国家自然科学基金资助项目(61370075).
作者简介:李 婷(1988-),女,山东潍坊人,东北大学博士研究生;马宗民(1965-),男,山东金乡人,东北大学教授,博士生导师.

[6]基于模糊 XML 数据模型[3]提出了能处理包含复合谓词 AND,NOT,OR 的 LTwig 算法,该算法可以仅对与查询相关的数据流进行一次扫描得出相应小枝查询的结果. 文献[7]提出了一种针对模糊 XML 文档的动态编码方案,设计了在动态环境中进行小枝查询匹配的算法 DQTwig. 而对于模糊 XML 数据的关键字查询方法的研究目前未见有报道,仅对模糊 XML 数据的结构查询研究很难满足非专业用户的查询需求. 如何处理在模糊 XML 环境下的关键字查询成为亟需解决的问题,为此本文将重点研究如何针对模糊 XML 数据进行关键字查询.

1 模糊 XML 数据模型

传统 XML 文档一般被表示为一个有向树结构,即 $T=(V,E)$,其中,V 表示节点的集合,E 表示边的集合. 对于一棵 XML 结构树 t,$V(t)$ 表示树中所有节点的集合,$E(t)$ 表示树中所有边的集合,可知,$E(t)\in V(t)\times V(t)$. 对于任意 $v\in V$ 有一个唯一的标签 Label(v). 若一棵 XML 树 T' 是树 T 的子树,则满足以下条件:$V(T')\subseteq V(T)$,$E(T')\subseteq E(T)$.

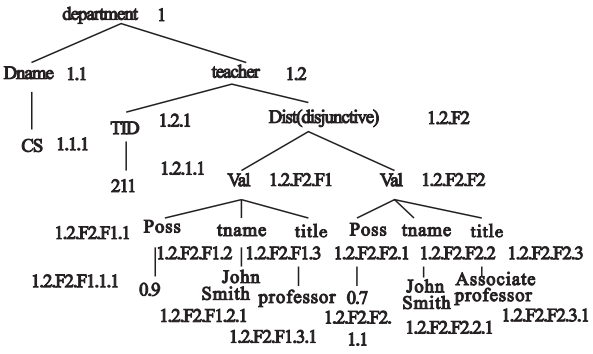

图 1 模糊 XML 树形结构

Fig. 1 The tree structure of fuzzy XML

对于表示模糊数据信息的模糊 XML 结构模型,同样被定义为一个树形结构 T_{fuzzy},通过引入模糊集和可能性分布理论来表示和处理模糊信息[3]. 在 XML 中采用两种模糊类型,一种是将元素节点与隶属度结合来表达元素的模糊度,另一种是将可能性分布与元素节点的属性值相结合来表达属性的各个可能值的模糊度. 对于可能性分布,存在两种类型,分别为析取(disjunctive)可能性分布和合取(conjunctive)可能性分布. 为表达元素的隶属度,在模型中引入一个可能性属性,用 Poss 表示,它在[0,1]之间取值. 可能性属性 Poss 与一个模糊结构体 Val 一起来表示在 XML 文档中给定元素的可能性大小. 如图 1 中的模糊 XML 树形结构和图 2 中的模糊 XML 文档所示(图 1 是图 2 中部分 XML 文档的树形结构). 图 2 中第 2 行 < Val Poss = “0.8” > 表示部门 department 为 Computer Science 的可能性为 0.8. 而对于隶属度值为 1 的元素节点,它的隶属度的表达 < Val Poss = “1” > 和 </Val > 将被忽略. 为了表达元素属性值的可能性分布,在 XML 模型中采用一个模糊结构体 Dist 来表示属性的各个值的可能性分布,一个 Dist 元素可以有多个 Val 元素作为孩子节点,每个 Val 节点关联一种可能性,Dist 需对可能性分布类型(析取或者合取)加以说明. 例如在图 2 中,第 5 ~ 18 行是一个析取结构体 Dist 关联于元素 teacher 信息的可能性分布. 它包含两个 Val 节点关联教师 John Smith 的两种可能性信息及其隶属度值. 一种可能性信息为他的职称是教授,可能性值为 0.9 ;另一种可能性信息为他的职称是副教授,可能性值为 0.7.

2 关键字查询语义与编码

2.1 模糊 XML 关键字查询语义

传统 XML 文档的关键字查询语义大多基于 SLCA [1] 语义进行研究. 考虑对图 1 中的 XML 文档片段进行 SLCA 关键字语义查询,若输入关键字 {John Smith,Associate professor},得到的 SLCA 结果是 Val 节点,此节点是模糊节点,不是一般的节点,不能作为查询结果返回. 可知对模糊 XML 文档运用传统的关键字查询语义会导致错误.

另外由于模糊 XML 模型中引入了模糊集和可能性分布理论,体现出元素的存在可能性(隶属度)大小,以及属性值的可能性分布及其隶属度大小.

因而关键字查询的 SLCA 结果的可能性大小成为一项重要的指标,如何确定和计算查询结果的可能性值(整体可能性)λ 便成为模糊 XML 文档上的关键字查询需要研究的重要问题.

定义 1 给定一组关键字集合 $\{m_1,m_2,\cdots,m_n\}$ 和一个模糊 XML 文档 T,对 T 进行 SLCA 语义的关键字查询,即得出包含全部关键字 $\{m_1,m_2,\cdots,m_n\}$ 的 SLCA 结果 R_k 及其可能性值 $\{(R_1,\lambda_1),(R_2,\lambda_2),\cdots,(R_k,\lambda_k)\}$,并且 SLCA 结果节点 $R_i(1\leqslant i\leqslant k)$ 为一般节点.

对于一个 SLCA 节点的可能性值,记为

$P^{w}_{slca}(v)$. 由于模糊 XML 模型的特点，$P^{w}_{slca}(v)$ 可以通过式(1)得出：

$$P^{w}_{slca}(v) = P(\text{path}_{r\to v}) \times P^{L}_{slca}(v). \quad (1)$$

式中，$P(\text{path}_{r\to v})$ 中，r 是根节点，整个式子表示从根节点 r 到节点 v 路径中，节点 v 的存在可能性大小. 假设从根节点 r 到节点 v 路径上的隶属度分布为 $\{1, 0.8, 0.5, 0.6\}$，则 $P(\text{path}_{r\to v}) = 1 \times 0.8 \times 0.5 \times 0.6 = 0.24$. $P^{L}_{slca}(v)$ 表示在以节点 v 为根节点的子树结构 $T_{sub}(v)$ 中，节点 v 的本地可能性大小. 假设以节点 v 为根节点的子树结构 $T_{sub}(v)$ 中，节点 v 到包含关键字的节点的隶属度分布为 $\{0.9, 0.6, 0.8\}$，那么 $P^{L}_{slca}(v) = 0.9 \times 0.6 \times 0.8 = 0.432$. 故 $P^{w}_{slca}(v) = 0.24 \times 0.432 = 0.104$.

```
1.  <course CName =" Artifical Intelligence ">
2.    <Val Poss =" 0.8 ">
3.      <department DName =" Computer Science ">
4.        <teacher TID =" 211 ">
5.          <Dist type =" disjunctive ">
6.            <Val Poss =" 0.9 ">
7.              <tname > John Smith </tname >
8.              <title > Professor </title >
9.              <salary > 7000 </salary >
10.             <tel > 024 - 8368001 </tel >
11.           </Val >
12.           <Val Poss =" 0.7 ">
13.             <tname > John Smith </tname >
14.             <title > Associate Professor </title >
15.             <salary > 5000 </salary >
16.             <tel > 024 - 8368001 </tel >
17.           </Val >
18.         </Dist >
19.       </teacher >
20.     </department >
21.   </Val >
22. </course >
```

图 2　部分模糊 XML 文档片段

Fig. 2　Partial fuzzy XML document fragment

2.2　节点编码

普通 XML 关键字查询处理普遍采用 Dewey 编码，对于模糊 XML 文档，Dewey 编码不能将模糊节点和一般节点进行相应的区分，难以断定参与查询的节点的类型，为了更好地对模糊文档进行关键字查询，对 Dewey 编码进行相应的扩展以更好地适应模糊 XML 关键字查询. 首先对模糊 XML 文档进行一般的 Dewey 编码，当遇到模糊节点 Dist，Val 时，采取在其节点编码的最后组成部分前添加字符‘ F ”来表示此节点是模糊节点. 相应的编码实例如图 1 所示. 将这种编码方式定义为 CDewey.

3　模糊 XML 关键字查询

3.1　索引

为了在模糊 XML 文档上面进行关键字查询，构建了 3 个索引，一个是记录包含各个关键字 $\{m_1, m_2, \cdots, m_n\}$ 的节点列表 $\{M_1, M_2, \cdots, M_n\}$. 模糊 XML 文档中与节点 s 相关的隶属度分布采用两个索引记录，一个是记录以节点 s 为根节点的子树中包含关键字的节点的隶属度分布，记为 L_L. 图 3 是简化的模糊 XML 文档树形结构，其中模糊节点用矩形表示(Dist 和 Val 的组合合并为一个模糊节点)，一般节点用椭圆表示，将元素以及属性值的隶属度大小标记在连接该节点与其父节点的边 E 上，未标记隶属度的边表示连接其两端的节点中孩子节点相对父节点的隶属度值默认为 1. 假设关键字为 B_2, E_3，其 SLCA 结果节点是 C_1，对于 L_L 将记录 $\{D(C_1): 0.8, 1, 0.9, 1\}$，$D(C_1)$ 是 C_1 的 CDewey 编码，即包含关键字的节点到节点 C_1 路径上的隶属度分布. 此索引可用来计算节点 C_1 的本地可能性值. 用另一列表索引 L_E 来记录节点 s 到根节点的路径上的隶属度分布，例如节点 C_1，它到根节点的隶属度值为 $\{0.7, 1\}$，故将 $\{D(C_1): 0.7, 1\}$ 存放在索引 L_E 中. 索引 L_E 可用来计算节点 s 的存在可能性值.

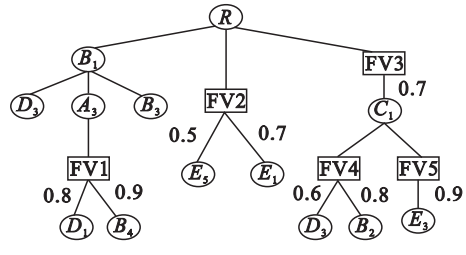

图 3　简化的模糊 XML 树形结构

Fig. 3　A simplified tree structure of fuzzy XML

3.2　关键字查询算法

模糊 XML 关键字查询不仅需要找出符合查询要求的一般 SLCA 结果节点，而且还需要计算其相应的可能性值. 为此，提出算法 FIndex Loop (见算法 1)，首先按照传统 SLCA 语义对输入的关键字进行初步搜索产生结果集 S_q，采用文献[1]中提出的基于 get_slca( ) 的算法得到结果集

S_q,对于 S_q 中的节点 s,如果它是模糊节点,它的父节点 $P(s)$ 是一般节点,将 $P(s)$ 及其相关隶属度分布记录到列表 L_E 中,$P(s)$ 作为结果节点输出. 如果它的父节点 $P(s)$ 仍然为模糊节点,向上寻找 s 的祖先节点,当 Ancestor(s) 为一般节点时,将 Ancestor(s) 及其相关隶属度分布记录到列表 L_E 中,并作为结果节点输出. 对于 S_q 中的一般节点,直接作为结果节点输出,执行 Compute Possibility(s) 功能(见算法 2),最后返回 SLCA 结果及其相应的可能性值.

算法 1 FIndex Loop

输入:关键字集合 $\{m_1, m_2, \cdots, m_n\}$,一个已编码的模糊 XML 数据树 T.

输出:关键字查询的 SLCA 结果及其可能性值 $\{(R_1, \lambda_1), (R_2, \lambda_2), \cdots, (R_k, \lambda_k)\}$.

1) 下载和访问关键字节点列表 $M = \{M_i\}$,$1 \leqslant i \leqslant n$,创建和更新列表 $L_E\{u:\sigma\}$;

2) 求解包含所有关键字 $\{m_1, m_2, \cdots, m_n\}$ 的 SLCA 结果,即 $S_q = \text{slca}(M_1, M_2, \cdots, M_n)$;

3) for 每个候选结果节点 $s \in S_q$;

4) if s 是模糊节点,find $P(s)$;//$P(s)$ 是 s 的父节点;

5) 当 $P(s)$ 是一般节点时,将节点 $P(s)$ 的记录更新到列表 $L_E\{P(s):\sigma\}$;

6) 当 $P(s)$ 是模糊节点时,寻找 s 的祖先节点 Ancestor(s);

7) 当 Ancestor(s) 是一般节点时,将节点 Ancestor(s) 的记录更新到列表 $L_E\{\text{Ancestor}(s):\sigma\}$;

8) else s 是一般节点,将 s 的记录更新到列表 $L_E\{s:\sigma\}$;

9) 执行 Compute Possibility(s) 功能;

10) Return (R_i, λ_i). //返回 SLCA 结果及其可能性值.

算法 2 是计算 SLCA 结果的可能性值的一个功能模块. 首先访问关键字节点列表 M,取列表 M 中的最小编码的节点 u,初始化一个栈 stack,把 u 的编码压入栈 stack 中,取 M 中剩余节点最小编码的节点 u',计算节点 u 和 u' 的最长公共前缀 LCP(longest common prefix),如果 LCP 的长度小于栈列 stack 中节点 u 编码长度,可知节点 u' 与节点 u 不存在祖先后代关系,取结果集 R_i 中的最小编码节点 R,得出节点 u,u' 与 R 的 LCP,如果 LCP(u, u', R) $\neq$ CDewey(R),则说明 u 和 u' 不是节点 R 的后代节点,将栈 stack 中除去 LCP(stack, u') 的剩余部分弹出;如果 LCP(u, u', R) = CDewey(R),则表明 u 和 u' 是节点 R 的后代节点,记录节点 u 和 u' 在列表 $L_E\{u:\sigma\}$ 的索引条目,将栈 stack 中除去 LCP(stack, u') 的剩余部分弹出;当节点 u 和 u' 存在祖先后代关系时,如果 LCP(stack, R) = CDewey(R),那么 u 和 u' 是节点 R 的后代节点,记录节点 u 和 u' 在列表 $L_E\{u:\sigma\}$ 的索引条目;访问节点 u' 后,将 $D(u')$ 中除去 LCP(stack, u') 的剩余部分压入栈并继续处理列表 M 中的下个节点. 当结果节点 R 的子树中的所有包含关键字的节点及其在列表 $L_E\{u:\sigma\}$ 的索引条目找到时,可以得到 $L_L\{R:\xi\}$. 重复过程 4) ~ 10),依次得出结果集 $\{R_1, R_2, \cdots, R_k\}$ 中 R_i $(1 \leqslant i \leqslant k)$ 的索引列表 $L_L\{R_i:\xi\}$. 访问相关索引分别求得结果节点的本地可能性值 $P^L_{\text{slca}}(R_i)$ 和存在可能性值 $P(\text{path}_{r \to R_i})$,计算结果节点 R_i 的可能性值,最后输出所有结果及其可能性值.

算法 2 Compute Possibility(s)

输入:SLCA 结果集 $\{R_1, R_2, \cdots, R_k\}$.

输出:SLCA 结果及其可能性值 $\{(R_1, \lambda_1), (R_2, \lambda_2), \cdots, (R_k, \lambda_k)\}$.

1) 访问关键字节点列表 $M = \{M_i\}$,$1 \leqslant i \leqslant n$,SLCA 结果集 $\{R_1, R_2, \cdots, R_k\}$,创建和更新列表 $L_E\{R_i:\sigma\}$;

2) 从列表 M 的最小 CDewey 编码节点 u 开始访问;

3) 初始化一个栈 stack,将 CDewey(u) 设为初始值;

4) 取列表 M 中下一个最小编码节点,u' = GetNextNode(M);

5) 计算节点 u 和 u' 的最长公共前缀 f = LCP(stack, u');

6) 当 stack. size > LCP. length 时,If LCP(u, u', R) $\neq$ CDewey (R),pop entry();// R 是 SLCA 结果集中的最小编码节点;

7) If LCP(u, u', R) = CDewey(R),记录节点 u,u' 在列表 L_E 中的条目,and pop entry();

8) 当 stack. size = LCP. length 时,If LCP(stack, R) = CDewey(R),记录节点 u,u' 在列表 L_E 中的条目;

9) for($f < j \leqslant u'$. length) stack. push($u[j]$);

10) 当得到节点 R 的子树中的所有的关键字节点在列表 L_E 的记录时,创建列表 $L_L\{R:\xi\}$.

11) 重复过程 4) ~ 10),直到得到所有的结果节点 R_i $(1 \leqslant i \leqslant k)$ 的列表 $L_L\{R_i:\xi\}$;

12) 访问列表 $L_L\{R_i:\xi\}$,计算节点 R_i 的本地

可能性 $P^{L}_{slca}(R_i)$；

13）访问列表 $L_E\{R_i:\sigma\}$，计算节点 R_i 的存在可能性 $P(\text{path}_{r\to R_i})$；

14）$\lambda = P^{L}_{slca}(R_i) \times P(\text{path}_{r\to R_i})$；//计算节点 R_i 的可能性值；

15）Return (R_i, λ_i).

4　实验与分析

对每个关键字查询构建相应的结构查询语句，采用 LTwig 算法[6] 求解出输入关键字查询的相应的结构查询结果. LTwig 是一个能够有效求解在模糊 XML 数据上面的小枝查询结果的算法，将结构查询得出的结果作为准确结果，对比 FIndex Loop 算法的查全率和查准率.

4.1　数据集和实验环境设置

实验数据集采用合成的 XML 数据集 XMark，随机生成大小 30 MB 的 XMark 数据集，并采用文献[8]中使用的随机模糊信息产生方法将原始的 XMark 数据集转换为模糊 XML 数据集，记为 FXM. 所有实验测试均在 Inter(R) Core (TM) i3CPU @ 2.13 GHz 处理器，2 GB RAM 和内置 320 GB 硬盘的 Windows 7 系统下进行. 查询关键字见表 1.

表 1　关键字查询示例

Table 1　Keyword query examples

查询 ID	关键字
Q_1	open auction, check, location
Q_2	person 6, phone
Q_3	buyer, item, address
Q_4	person 9, credit, name, price
Q_5	United States, close auction
Q_6	person 34, person 5, address
Q_7	ship, open auction
Q_8	America, phone, seller
Q_9	person 5, person 23, address
Q_{10}	buyer, credit, Africa

4.2　查全率和查准率

对于一个关键字查询，将其相应构建的结构查询通过 LTwig 算法得出的结果作为准确结果，记为 AR(accurate result)，通过 FIndex Loop 算法得出的结果作为近似结果，记为 PR(approximate result)，则查全率 Recall = $|AR \cap PR| / |AR|$，查准率 Precision = $|AR \cap PR| / |PR|$. 在模糊数据集 FXM 上面对 $Q_1 \sim Q_{10}$ 的关键字查询进行查全率和查准率测试，如图 4 所示. 实验结果表明 FIndex Loop 具有较高的查全率和查准率.

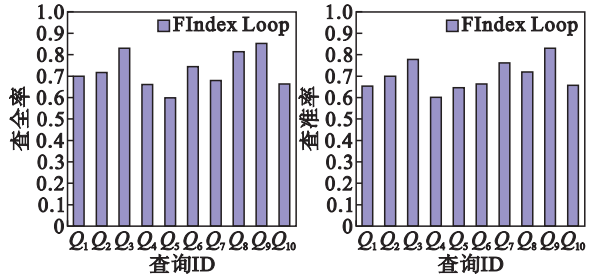

图 4　FIndex Loop 算法的查全率和查准率

Fig. 4　Recall and precision of FIndex Loop algorithm

5　结　　语

本文针对模糊 XML 数据模型下的关键字查询方法进行了研究. 首先研究并提出了模糊 XML 的关键字查询语义，设计了一种新型编码方法 CDewey. 在此基础上，提出了模糊 XML 关键字查询算法 FIndex Loop，实现对模糊 XML 文档的关键字语义查询及其 SLCA 结果可能性值的计算. 最后通过实验验证所提方法的有效性. 未来工作将集中在对相关算法的优化，以及对查询结果的排序处理等领域.

参考文献：

[1] Xu Y, Papakonstantinou Y. Efficient keyword search for smallest LCAs in XML databases [C]//Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data. Baltimore: ACM, 2005: 527 – 538.

[2] Guo L, Shao F, Botev C, et al. XRANK: ranked keyword search over XML documents [C]//Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data. San Diego: ACM, 2003: 16 – 27.

[3] Ma Z M, Yan L. Fuzzy XML data modeling with the UML and relational data models [J]. *Data & Knowledge Engineering*, 2007, 63: 972 – 996.

[4] Nierman A, Jagadish H V. ProTDB: probabilistic data in XML [C]// Proceedings of the 28th International Conference on Very Large Data Bases. Hong Kong: VLDB Endowment, 2002: 646 – 657.

[5] Zhang C J, Chang L, Sha C F, et al. Keywords filtering over probabilistic XML data [M]// Web Technologies and Applications. Berlin: Springer-Verlag, 2012: 183 – 194.

[6] Liu J, Ma Z M, Ma R Z. Efficient processing of twig query with compound predicates in fuzzy XML [J]. *Fuzzy Sets and Systems*, 2013, 229: 33 – 53.

[7] Liu J, Ma Z M, Qu Q L. Dynamically querying possibilistic XML data [J]. *Information Sciences*, 2014, 261: 70 – 88.

[8] Liu J, Ma Z M, Yan L. Efficient processing of twig pattern matching in fuzzy XML [C]// Proceedings of the 18th ACM Conference on Information and Knowledge Management. Hong Kong: ACM, 2009: 117 – 126.