ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.10, October 2009, pp.2667−2678
doi: 10.3724/SP.J.1001.2009.03487

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

基于函数依赖的结构匹配方法*

李国徽, 杜小坤+, 胡方晓, 杨 兵, 唐向红

(华中科技大学 计算机科学与技术学院,湖北 武汉 430074)

Structure Matching Method Based on Functional Dependencies

LI Guo-Hui, DU Xiao-Kun+, HU Fang-Xiao, YANG Bing, TANG Xiang-Hong

(Department of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China)

+ Corresponding author: E-mail: hustdxkun@163.com, http://www.hust.edu.cn

Li GH, Du XK, Hu FX, Yang B, Tang XH. Structure matching method based on functional dependencies. *Journal of Software*, 2009,20(10):2667−2678. http://www.jos.org.cn/1000-9825/3487.htm

Abstract: Schema matching is a basic problem in many database application domains, such as data integration, E-business, data warehousing and semantic query processing. Recently it has become a research hotspot, and most of the achievements are about the use of element's own information. Research about element's own information is very mature at present. As an important piece of information in a schema, structure information can be useful information for schema matching, but the research of structure information is far behind that of element's own information. This paper divides the similarity between two elements into linguistic similarity and structural similarity, and gets the structural similarity by a new statistic method, and then gets the matching probability by integrating the linguistic similarity and structural similarity. At last, the paper gets the mapping between schema elements according to the matching probability. Extensive simulation experiments are conducted and the results show that this algorithm is better than other algorithms in various performance metrics.

Key words: schema matching; functional dependency; structure match; matching probability

摘 要: 模式匹配是模式集成、数据仓库、电子商务以及语义查询等领域中的一个基础问题,近来已经成为研究的热点,并取得了丰硕的成果.这些成果主要利用元素(典型的为关系模式中的属性)自身的信息来挖掘元素语义,目前,这方面的研究已经相当成熟.结构信息作为模式中一种重要的信息,能够为提高模式匹配的精确性提供有用的支持,但是目前关于如何利用结构信息提高模式匹配的精确性的研究还很少.将模式元素之间的相似度分为语义相似度(根据元素自身信息得到的相似度)和结构相似度(根据元素之间的关联关系得到的相似度),并采用新的统计方法计算元素间的结构相似度,然后再综合考虑语义相似度得到元素间的相似概率;最后根据相似概率得到模式元素间的映射关系(模式元素之间的对应关系).实验结果表明,该算法在查准率、查全率及全面性等方面都优于已有的其他算法.

关键词: 模式匹配;函数依赖;结构匹配;匹配概率

* Supported by the National Natural Science Foundation of China under Grant No.60873030 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2007AA01Z309 (国家高技术研究发展计划(863)); the National Defense Pre-Research Foundation of China under Grant No.9140A15040208JW0501 (国防预研基金)

Received 2008-03-06; Revised 2008-07-08; Accepted 2008-10-07; Published online 2009-05-07

中图法分类号: TP391 文献标识码: A

模式匹配是模式之间的一个二元操作,它以源模式和目标模式为输入,以两个模式的元素(在关系型数据库中对应于一个关系的属性)之间的映射关系作为匹配结果输出.作为信息集成领域的一个基本操作,模式匹配在越来越多的应用领域中发挥着重要的作用.如模式集成、数据仓库、电子商务、语义 Web、P2P 数据库等领域.目前的模式匹配工作大多是由操作人员手工匹配的,这要求操作人员很好地熟悉数据库的模式结构以及每个模式元素的语义,这是一个枯燥、费时且容易出错的过程.随着软件技术的不断发展,数据库模式逐渐增大,现在的数据库中有数百个表、数千个属性都是比较常见的,它们可能由不同的设计人员设计,这就使得了解数据库中每个模式元素的语义变得越来越困难,有时甚至是一个不太可能完成的任务.因此,需要一种自动的模式匹配方式来代替费时、费力且容易出错的人工匹配方式.目前,针对自动和半自动的模式匹配问题的研究已有许多成果[1−10].Bernstein 和 Rahm 对这些成果进行了总结和分类[11],按照获取元素语义时利用信息类型的不同而采用不同的分类标准.一般有模式与实例、元素与结构等几种不同的分类方法,这些分类也隐含地列举了模式匹配过程中可利用的信息:元素自身信息(如元素名、数据类型等)、元素对应的数据信息及元素间相互关联的结构信息等等.

元素自身信息能够最直观地反映元素的语义,对元素自身信息的挖掘是模式匹配研究中应该首先考虑的问题.目前,大多数的研究成果都是针对元素间语义相似度的,对它的研究已经相当成熟.在对元素间相互关联的结构信息的研究中,对结构相似度的计算也以语义相似度为基础.目前对其研究较少,且对结构相似度的研究还存在以下几点不足:(1) 缺少一种能够准确地表达模式结构信息的数据结构;(2) 对与元素相关联信息未按重要程度进行区分.此外,在得到元素间的相似度(语义相似度、结构相似度)之后,当根据相似度值直接选取模式元素间的映射关系时我们会发现,在多个元素匹配对的相似度值相互接近时,单纯地根据相似度值选取的映射关系不够精确.

本文提出了一种新方法,将元素的相似度分为语义相似度和结构相似度.首先根据元素自身信息计算元素之间的语义相似度,并对目标模式中的每个元素选择其候选匹配集,然后对每个候选匹配对中的两个元素的关联元素集(模式结构中与之关联的所有元素组成的集合)间的相互匹配关系进行统计计算而得到这两个元素的结构相似度,最后将语义相似度、结构相似度及相似概率有机地结合起来,为映射关系的选择提供了一个更有效的标准.

本文的主要创新点在于:(1) 将函数依赖概念引入到模式匹配过程中,利用函数依赖关系表达模式结构信息;(2) 提出了一种新的结构相似度定义及传递推导方法;(3) 利用相似概率,使映射的选择有了一个更为有效的标准.

本文第 1 节介绍相关的研究成果并引出新的方法.第 2 节介绍新方法的结构相似度的计算及传递推导算法.第 3 节给出新方法的映射生成过程.第 4 节对本文提出的方法进行实验评价.第 5 节是总结与展望.

1 相关研究介绍

模式匹配可分为相似度计算、映射生成两阶段.相似度计算是利用所有能够反映模式元素语义的信息来计算元素间相似度的过程.我们把计算元素间不同类型信息相似度的方法称为匹配器,常用的匹配器有如下几种:

(1) 名称匹配器:尽管元素名称作为元素语义最直观的表示,但要获取元素名称所表达的准确语义却不是一件容易的事情,因为元素名称中经常包含简写(例如用 *StuID* 表示学号 *Student ID*)、特殊字符等影响我们准确识别元素语义的因素,一般采用 *N*-grams[5]算法来挖掘元素语义,能够有效地排除错误拼写、特殊字符、简写等因素对元素语义挖掘的干扰,达到准确识别语义的目的,实验表明具有良好的效果.常用的算法还有 TF/IDF[12]和 Naïve Bayes[13]等.

(2) 实例匹配器:实例是元素对应的数据信息,相同的元素对应的数据信息应该相似[3].若发现某一元素对应的实例在另一元素对应的实例中大量出现,则可基本判断两个元素相似.实例匹配法即是根据这一原理来计

算元素之间的相似度.

(3) 数据类型匹配器:根据模式元素的数据类型之间的兼容性表示元素之间的相似度[5],数据类型种类是有限的,我们可以用一张表列举出不同数据类型之间的相似度,在匹配时直接对照该表得出两个数据类型之间的相似度值.

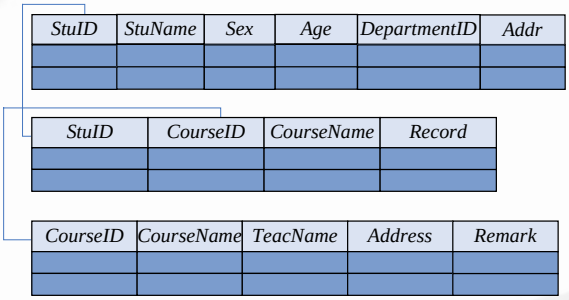
单一的匹配器只能计算元素间确定的某一类型信息的相似度,所以通常将不同类型的信息结合起来计算得到元素间更精确的语义相似度.常用的有如下两种结合策略:

(1) 将各种不同类型的元素自身信息综合处理的混合匹配法;

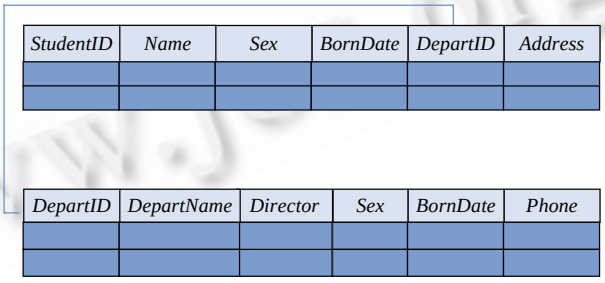
(2) 各种不同的匹配器单独执行,将其结果综合的合成匹配法.

混合匹配法在一遍扫描过程中综合考虑元素各种不同类型的自身信息得到元素间语义相似度,这样可以节省计算时间,但利用的信息类型必须预先设定好,不可根据实际情况调整;合成匹配法将不同的匹配器分别执行后得到的结果综合得到最终的相似度,这样可以根据用户的选择动态调整利用的信息类型,但需要多遍扫描数据库,时间复杂度比较高.这两种结合策略各有利弊,但是出于模式匹配算法通用性的考虑,合成匹配法占有一定的优势,其中比较常用的合成匹配器有 COMA(combination of schema matching approaches)[5]等.COMA 方法提供了一种可扩展的框架,可以方便地将以往提出的各种匹配器融合起来,这些匹配器分别根据模式中信息的不同类别来计算模式元素的相似度,最后将这些不同的匹配器生成的相似度合成为一个综合相似度,根据综合相似度生成匹配信息.混合匹配法中也有一种较常用的匹配算法 SEMINT(semantic integrator)[14],该方法将神经网络与模式匹配相结合,利用神经网络将模式元素间不同种类信息的相似度进行合成,并生成最终映射结果.

单纯地利用元素自身信息(如元素名、数据类型等)或与元素对应的数据信息很难全面反映模式元素间的相似程度,容易被一些特殊的情形所误导,如图 1 所示,源模式中 *course* 表属性 *Address* 与目标模式中 *student* 表属性 *Address* 从自身信息分析被认为匹配,但实际则不然.究其原因,在前面对相似度计算的介绍中,对模式信息的利用大都局限于模式元素自身信息和与模式元素对应的数据信息,未能有效地利用模式的结构信息.



(a) Source schema *S*: Table *student*, Table *stu-course*, Table *course*
(a) 源模式 *S*:依次为 *student* 表、*stu-course* 表、*course* 表



(b) Target schema *T*: Table *student*, Table *Department*
(b) 目标模式 *T*:依次为 *student* 表、*Department* 表

Fig.1 Source schema *S* and target schema *T*

图 1 源模式 *S* 和目标模式 *T*

针对这一缺点,目前已有一些成果对其做出了改善[4,7,8],这些算法以 CUPID[8]和 SF[7]为代表,下面分别对这两种方法作简要介绍.

(1) CUPID:该方法计算语义相似度时先将所有元素分类,对同一类别中的元素计算它们之间的语义相似度,并以此为基础统计得到元素间的结构相似度,然后利用相关元素结构相似度相互影响的原理调整得到最终的结构相似度,最后将语义相似度和结构相似度综合后,选取相似度值大于某一阈值的元素对作为匹配结果.算法有效地利用了模式的结构信息,但在生成映射过程中对综合相似度(语义相似度和结构相似度的综合)采用了简单的取阈值来选择的方法,不同的阈值对结果的影响很大,于是阈值的选取又成为一个难题.

(2) SF(similarity flooding):该方法以模式元素间的语义相似度为基础,根据模式的结构,采用相似度传递算法,使得相关元素的语义相似度相互影响,直至其相似度趋于稳定(即传递前后相似度变化值小于ω)或传递次数超过设定值 N,最后根据元素间相似度选择得到模式映射.该算法有效地利用了模式的结构信息,但模式结构图包含了模式的所有信息(包括元素、元素的数据类型等信息).这些信息在图中都以节点的形式表示,使图的结构很复杂,许多无效信息(如数据类型等)也参与匹配,从而使得算法的时间复杂度很高,且在传递过程中削弱了有效信息匹配对之间的相互影响.

由以上分析可知,在研究结构相似度时主要解决以下两个问题:

(1) 如何表示元素间的关联关系,哪些元素间存在关联;

(2) 如何根据与元素关联的其他元素来计算结构相似度.

表 1 分别对 CUPID,SF 算法的元素关联关系及结构相似度两方面进行比较.

Table 1 Contrast between two structure similarity algorithms

表 1 两种结构相似度算法对比

	Structure	The degree of similarity
Cupid	Tree	$ssim(s,t)=\frac{\lvert\{x \mid x\in leaves(s)\wedge\exists y\in leaves(t), stronglink(x,y)\}\cup\{x \mid x\in leaves(t)\wedge\exists y\in leaves(s), stronglink(x,y)\}\rvert}{\lvert leaves(s)\cup leaves(t)\rvert}$
SF	Graph	$\sigma^{i+1}(x,y)=\sigma^{i}(x,y)+\sum_{(a_u,p,x)\in A,(b_u,p,y)\in B}\sigma^{i}(a_u,b_u)\cdot\omega((a_u,b_u),(x,y))+\sum_{(x,p,a_v)\in A,(y,p,b_w)\in B}\sigma^{i}(a_v,b_v)\cdot\omega((a_v,b_v),(x,y))$

映射生成是根据相似度计算过程中得到的元素对间的相似度选择元素间映射关系的过程,目前的算法都是直接根据相似度进行选择,主要有如下两种策略:

(1) 为目标模式的每个元素选取相似度最高的候选匹配作为实际匹配;

(2) 根据稳定婚姻法[10]或者最大流通量法综合考虑元素间相似度值的相互影响,为目标模式中的每个元素选择一个全局最优的候选匹配.

策略(2)选择精确度较高,所以使用比较广泛.但由于直接使用元素间实际计算出的相似度进行选择,所以在全局考虑时存在不同元素对间的相似度值不具有可比性这样一个缺陷,从而影响了匹配的精确度.

本文所描述的方法利用关系数据库中的函数依赖关系来体现模式元素之间的关联,虽然是从关系模式中引出,但仍然不失其一般性,可方便地扩展至 XML,E-R 图等其他形式的模式结构,并采用图结构作为中间结构,图结构中只存在模式元素节点,有效地降低了算法的时间复杂度.同时,在利用统计方法计算每个元素的结构相似度时,因为一个父元素的相似度比一个子元素的相似度对元素相似度的影响要大,所以我们把结构相似度分为父结构相似度和子结构相似度分别加以计算,以提高算法的精确度.最后,在映射选取时我们引入相似概率这一概念,为映射关系的选取提供了一个更有效的标准,提高了映射选取的精确度.算法 1 是本文方法的算法描述.

算法 1. 结构相似度算法.

输入:源模式 S、目标模式 T 及源模式中每一元素的候选匹配集合 $CAND$;

输出:源模式 S 和目标模式 T 之间的映射关系 $M(S,T)$.

FD_BASED(S,T,$CAND$)

(1) 生成源模式 S 和目标模式 T 的模式图 $G(S)$和 $G(T)$;

(2) 对 *CAND* 中的每一个候选匹配(*s*,*t*)计算父结构相似度 *asim*(*s*,*t*)和子结构相似度 *csim*(*s*,*t*);

(3) 根据模式图 *G*(*S*)和 *G*(*T*),对 *asim*(*s*,*t*)和 *csim*(*s*,*t*)传递调整;

(4) 根据调整后的 *asim*(*s*,*t*),*csim*(*s*,*t*)以及 *lsim*(*s*,*t*)生成相似概率 *Psim*(*s*,*t*);

(5) 根据相似概率选择模式间映射关系 *M*.

2 结构相似度计算

2.1 取得每个模式元素的候选匹配集

首先,我们根据模式元素的语义相似度得到每个元素的候选匹配集合,在候选匹配集的基础上讨论结构相似度.前面已经简要地介绍了语义相似度匹配算法,这里需要说明的是:对于候选匹配集合选取的标准,常用的有如下 3 种:

(1) Max*N*:选取相似度最高的 *N* 个匹配项作为候选匹配.

(2) MaxDelta:选取与相似度最大值间差值小于 *d* 或者最大值的*α*%的匹配项作为候选匹配.

(3) Threshold:选取相似度大于固定阈值(threshold)的匹配项作为候选匹配.

单一的选择标准都存在缺点,例如:Max*N* 和 MaxDelta 返回的值可能相似度都很低,而 Threshold 返回的值可能非常少或者非常多.据此,可将多条标准结合起来考虑,比如 Max*N* 和 Threshold 或者是 MaxDelta 和 Threshold,根据算法的特点,我们将 MaxDelta 和 Threshold 这两个标准结合起来,生成目标模式中任一元素 *t* 的所有候选匹配,这里用 *CAND*(*t*)来表示目标模式中元素 *t* 的所有候选匹配的集合.

以图 1 给出的模式为例,采用以上标准,目标模式 *T* 中每个元素得到如表 2 所示的候选匹配集合.

Table 2 All candidates match of target schema *T*

表 2 目标模式 *T* 中所有元素的候选匹配列表

Object elments	Candidate match
student.StudentID	*student.StuID*; *student.StuName*; *stu-course.StuID*
student.Name	*student.StuName*; *stu-course.CourseName*; *course.CourseName*; *course.TeacName*;
student.Sex	*student.Sex*
student.BornDate	*student.Age*;
student.DepartID	*Student.DepartmentID*;
student.Address	*Student.Addr*; *course.Address*;
department.departID	*Student.DepartmentID*;
department.DepartName	*Student.DepartmentID*; *stu-course.CourseName*; *course.CourseName*; *courser.TeacName*;
department.Sex	*student.Sex*;
department.BornDate	*student.Age*;

2.2 生成函数依赖并建立依赖图

函数依赖是关系数据库领域中重要的理论,它表示模式中元素间的依赖关系.我们利用这种依赖关系来表示模式的结构信息,根据函数依赖的 Armstrong 公理系统[15]能够从模式中推导出所有元素间的函数依赖关系组成的函数依赖集 *FD*,我们采用如下 3 条规则生成函数依赖集 *FD*:

(1) 表结构中除主键外的元素 *s* 函数依赖于该表的主键 *k*,即添加 *k*→*s* 到函数依赖集 *FD*;

(2) 存在外键关系的两个元素 *m*,*n* 为同一元素,在这两个元素相关的函数依赖中将二者修改一致;

(3) 当主键为多个属性的属性组时,在 *FD* 中添加属性组中的单个元素函数依赖于属性组的关系.

我们以源模式 *S* 为例来解释规则的使用:根据规则(1)得到 *student.StuID*→*student.StuName*,*student.StuID*→*student.Sex* 等,根据规则(2)及(*stu-course.StuID*,*stu-course.CourseID*)→*stu-course.Record* 得到(*student.StuID*,*course.CourseID*)→*stu-course.Record* 等,然后又根据规则(3)得到(*stu-course.StuID*,*stu-course.CourseID*)→*stu-course.StuID* 等.

根据规则(1)~规则(3)分别生成两个模式的函数依赖集 F_s 和 F_t;然后根据依赖集对源模式和目标模式建立函数依赖图(如图 2 所示).

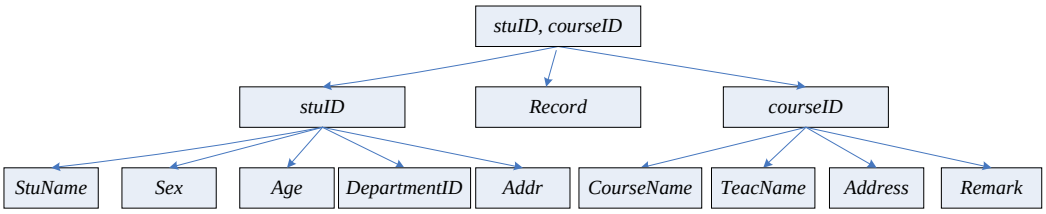


Fig.2 Functional dependency graph of source schema S

图 2 源模式 S 的函数依赖图

定义 1. 函数依赖图 $G=(V,E)$是一个定义在节点集 V 上的有向图,节点集 V 为在函数依赖集 FD 中出现的所有元素及元素组合所组成的集合,集合 E 中的一条有向边为函数依赖集 FD 中表示的一个函数依赖关系.

定义 2. 对函数依赖集 FD 中任意一个函数依赖关系 $k\rightarrow s$,我们将元素 k 称为元素 s 的父元素,元素 s 称为元素 k 的子元素.

一个元素的结构信息由与该元素相关联的其他元素构成,所以,两个元素间的结构相似度即为与这两个元素相关联的其他元素间的相似程度.并且,由于元素之间的关联方式不同(父元素、子元素),不同的关联方式对应的结构信息的类型不同,所以对结构相似程度的影响也不同.这里,将一个元素关联的其他元素分为函数决定元素闭包和函数依赖元素闭包两部分,它们分别对应该元素的子结构相似度和父结构相似度.下面给出相关定义:

定义 3. 元素 s 关于函数依赖集 F_s 的函数决定元素闭包 $s_{F_s}^+$ 是能够由 F_s 根据 Armstrong 公理系统推理规则 2(增广律)、推理规则 3(传递律)推导出的所有完全函数依赖于 s 的元素组成的集合.

定义 4. 元素 s 关于函数依赖集 F_s 的函数依赖元素闭包 $s_{F_s}^-$ 是能够由 F_s 根据 Armstrong 公理系统推理规则 2(增广律)、推理规则 3(传递律)推导出的所有完全函数决定 s 的元素和元素集合组成的集合.

根据定义 3 和定义 4 可生成模式 S 和 T 中每个元素的函数决定元素闭包和函数依赖元素闭包.对于任意候选匹配对,我们根据函数决定元素闭包计算其子结构相似度,根据函数依赖元素闭包计算其父结构相似度.为了下文描述方便,我们给出如下两个定义:

定义 5. 对模式中任一元素 e,我们称 e 的函数决定元素闭包或者函数依赖闭包中的任一元素为 e 的一个结构信息元,函数决定闭包中结构信息元的个数称为决定结构信息数,函数依赖闭包中结构信息元的个数称为依赖结构信息数.

定义 6. 对候选匹配对(s,t),若 $s(t)$的函数决定元素闭包中的任一元素 $s_1(t_1)$在 $t(s)$的函数决定元素闭包中存在候选匹配,则称 $s_1(t_1)$为(s,t)的决定结构匹配元;若 $s(t)$的函数依赖元素闭包中的任一元素 $s_1(t_1)$在 $t(s)$的函数依赖元素闭包中存在候选匹配,则称 $s_1(t_1)$为(s,t)的依赖结构匹配元.

首先,我们以决定元素闭包和父结构相似度为例来介绍结构相似度的算法.我们可以直观地发现:候选匹配对(s,t)的子结构相似度与其决定结构匹配元的个数 n 和元素 s,t 的决定结构信息数之和 m 之间的比值δ有关.δ越大,即所有决定结构信息元中匹配的信息元所占比例越高,候选匹配对(s,t)的子结构相似度就越高.同时,根据δ的计算过程我们可以发现,由于我们根据语义相似度选择候选匹配,而语义相似度的计算又是基于启发式方法,所以候选匹配中不可避免地存在一定程度的误配(候选匹配的两个元素实际上并不匹配).若决定结构信息数很少,即 m 很小,n 由于误配的原因在一定误差范围内变化时对$\delta=n/m$ 的影响很大,即δ值不稳定.据此,我们把 s,t 的决定结构信息数之和 m 称为候选匹配对(s,t)结构稳定因子.m 越大,δ的误差范围越小,同时,根据δ计算的子结构相似度越能真实地反映候选匹配对(s,t)间的结构相似程度.

对于候选匹配对(s,t),首先分别生成模式 S 中元素 s 关于函数依赖集 F_s 的函数决定元素闭包 $s_{F_s}^+$ 和模式 T 中元素 t 关于函数依赖集 F_t 的函数决定元素闭包 $t_{F_t}^+$;然后我们根据公式(1)计算$\delta(s,t)$.

$$\delta(s,t)=\frac{\left|\{x\mid x\in s_F^+\wedge\exists y\in t_F^+, y\in CAND(x)\}\cup\{x\mid x\in t_F^+\wedge\exists y\in s_F^+, y\in CAND(x)\}\right|}{\left|s_F^+\cup t_F^+\right|} \tag{1}$$

得到$\delta(s,t)$后,根据前面对$\delta(s,t)$、稳定因子 m 和子结构相似度 $csim(s,t)$之间关系的定性分析,我们用公式(2)表示三者之间的定量关系.

$$csim(s,t)=(\delta(s,t))^2\times\left(\frac{m}{m+\alpha}\right) \tag{2}$$

公式(2)中,参数α取值越小,稳定因子 m 对子结构相似度的影响减小;反之,稳定因子 m 对子结构相似度的影响增大.我们以源模式中的元素 *StuID* 与其候选匹配 *StudentID* 为例,取α=1,计算其子结构相似度 $csim(StuID, StudentID)=((5+5)/(5+5))^2\times(10/(10+1))=0.909$.

对于依赖元素闭包和父结构相似度,候选匹配对(s,t)的父结构相似度 $asim(s,t)$与元素 s,t 的依赖结构信息数之和 m 以及δ(依赖结构匹配元的个数 n 和 m 之间的比值)之间的关系与其子结构相似度的相关分析相同.对于候选匹配对(s,t),首先生成模式 S 中元素 s 关于函数依赖集 F_s 的函数决定元素闭包 $s_{F_s}^+$ 和模式 T 中元素 t 关于函数依赖集 F_t 的函数决定元素闭包 $t_{F_t}^+$;然后,我们根据公式(3)计算$\delta(s,t)$.这里需要注意的是,当 x 为多元素集合时,只有当 x 中的每个元素 a 都满足($\exists b\in t_F^-, b\in CAND(a)$)时,x 才满足($\exists y\in t_F^-, y\in CAND(x)$).

$$\delta(s,t)=\frac{\left|\{x\mid x\in s_F^-\wedge\exists y\in t_F^-, y\in CAND(x)\}\cup\{x\mid x\in t_F^-\wedge\exists y\in s_F^-, y\in CAND(x)\}\right|}{\left|s_F^-\cup t_F^-\right|} \tag{3}$$

得到$\delta(s,t)$后,我们采用与子结构相似度相同的公式(4)来计算父结构相似度:

$$asim(s,t)=(\delta(s,t))^2\times\left(\frac{m}{m+\alpha}\right) \tag{4}$$

公式(4)中的参数α与公式(2)中的意义相同.以源模式中元素 *StuName* 与其候选匹配 *Name* 为例,取α=1,计算其父结构相似度为 $asim(Stuname,Name)=((1+1)/(1+1))^2\times(2/(2+1))=0.667$.

2.3 结构相似度传递算法

在第 2.2 节中,我们对结构相似度的定义是建立在与该元素关联元素的候选匹配集的基础上,但是每个模式元素的子元素的结构相似度对其自身结构相似度是有影响的,子元素结构相似度越高,就越能提高元素自身的结构相似度,也就是说,结构相似度不仅仅与同该元素相关的元素的语义相似度有关,还与同该元素相关的元素的结构相似度有关.据此,我们采用递归调整算法对结构相似度进行调整优化.

在介绍递归调整算法之前,我们首先介绍元素集合的父结构相似度:对于两个元素集合 X,Y,我们定义二部图 $G(V,E)$如下:$V=X\cup Y, E=\langle(x,y)|x\in X\wedge y\in Y\wedge y\in CAND(x)\rangle$,$E$ 中每条边的权值为所关联两个元素的父结构相似度.这里,我们用二部图的最大流通量表示这两个元素集合的父结构相似度(见公式(5)),对公式(5)的右边采用匈牙利算法[16]计算二部图的最大流通量:

$$\text{asim}(X,Y)=\max\left(\sum_{x\in X}^{x}\sum_{y\in Y}^{y}asim(x,y)\right) \tag{5}$$

同理,在对元素集合的子结构相似度的定义中,我们采用相同的方法定义二部图 $G(V,E)$,但不同的是,E 中每条边的权值为所关联的两个元素的子结构相似度.这里同样用二部图的最大流通量表示这两个元素集合的子结构相似度(见公式(6)),对公式(6)的右边采用匈牙利算法[16]计算二部图的最大流通量:

$$csim(X,Y)=\max\left(\sum_{x\in X}^{x}\sum_{y\in Y}^{y}csim(x,y)y\right) \tag{6}$$

首先,根据模式依赖图对元素父结构相似度进行传递调整,父结构相似度传递算法如算法 2 所示.

算法 2. 父结构相似度传递算法.

(1) 从目标模式图中将不依赖于任何元素的元素添加到队列 Q;

(2) 从队列 Q 中移出元素 t;

(3) 对 t 的每一个候选匹配 $s, asim(s,t)=\alpha\times asim(s,t)+\beta\times asim(parents(s),parents(t))$;

(4) 将元素 t 的子元素都移入队列 Q;

(5) 若队列不为空,返回第(2)步.

算法广度优先遍历目标模式中的所有节点,对每个节点元素 t 与其每一个候选匹配项 s,其父结构相似度为 $asim(s,t)$,算法第(3)步根据两个元素的父元素集合的结构相似度调整它们之间的父结构相似度,调整后的父结构相似度 $asim(s,t)=\alpha\times asim(s,t)+\beta\times asim(parents(s),parents(t))$,其中,$parents(s)$表示在源模式树中节点 s 的父元素的集合,$parents(t)$表示在目标模式树中节点 t 的父元素的集合.$asim(parents(s),parents(t))$是 $parents(s)$和 $parents(t)$这两个元素集合的父结构相似度,α,β表示传递系数,且满足$\alpha+\beta=1$.每一遍传递完成后,我们对父结构相似度作如下处理:选取所有元素对中父结构相似度的最大值 M,以 M 为标准,对所有元素对的父结构相似度作修正,取其值 X 与 M 的比值 X/M 作为其修正值.因为经过传递后的父结构相似度值经常会超过 1,经过多次传递后,其值会变得很大,经过这样的修正后,可使每次传递前父结构相似度保持在[0,1]内.重复执行该传递算法,经过有限次执行后,若每一对候选匹配元素的父结构相似度都趋于稳定(传递前后差值小于ω)或者执行次数大于 N,则父结构相似度传递过程结束.

子结构相似度的传递算法与父结构相似度的传递算法基本一致,但存在如下两点不同:

(1) 采用广度优先遍历的逆序遍历目标模式中的元素;

(2) 在进行相似度传递时,子结构相似度采用如下公式调整:

$$asim(s,t)=\alpha\times asim(s,t)+\beta\times asim(children(s),children(t)).$$

重复执行子结构相似度传递算法,经过有限次执行后,若每一对候选匹配元素的子结构相似度都趋于稳定(传递前后差值小于ω)或者执行次数大于 N,则子结构相似度传递过程结束.

经过父元素相似度传递和子元素相似度传递后,元素间的相似程度已由调整后的父结构相似度值及子结构相似度值充分地反映出来.这时,目标模式中的每一个元素及其候选匹配间都存在语义、子结构、父结构 3 种不同类型的相似度,它们能从不同的角度反映元素之间的相似性.下面我们讨论如何根据这 3 个相似度确定元素之间的映射关系.

3 映射关系确定

映射关系的确定是模式匹配过程中的一个重要步骤.SF[7]介绍了一种解决模式映射问题的方法:稳定婚姻法.其核心思想是:选择这样一些匹配对,使映射的相似度之和最大,但不存在这样的两个匹配对$(x,y),(m,n)$,x 与 n 的相似度大于 x 与 y 的相似度,同时,y 与 m 的相似度大于 y 与 x 的相似度.但是,这种方法仍然存在不同元素对的相似度值不能直接比较的问题,因为相似度的具体数值并不具有实际的意义,一个元素与另一个元素之间相似的程度不仅与这两个元素之间的相似度有关,还与这个元素和其他候选匹配的相似度有关.例如:$lsim(student.Addr,student.Address)=0.2$;$lsim(student.DepartmentID,student.DepartID)=0.483$;$lsim(student.DepartmentID,Department.DepartName)=0.319$.因为在 $student.Address$ 的所有候选匹配中仅存在 $student.Addr$ 一个候选匹配,而 $student.DepartmentID$ 有两个候选匹配,所以,$student.Address$ 和 $student.Addr$ 成为匹配的可能性会高于 $student.DeartmentID$ 和 $Department.DepartmentName$ 成为匹配的可能性,即使 $lsim(student.Address,student.Addr)=0.2<lsim(student.DepartmentID,DepartmentDepartmentName)=0.319$.据此我们给出了相似概率的概念,相似概率表示每个元素与其候选匹配能够相互匹配的概率值.对于源模式中的任意元素来说,所有候选匹配的相似度值的总和越高,表明该元素在目标模式中存在实际匹配的概率就越高;反之则概率越低.对同一个元素的不同候选匹配来说,相似度值越高,匹配的概率就越高;反之概率就越低.由此,我们给出相似概率的定义如下:

对于源模式中任一元素 x 与其候选匹配集 $CAND(x)$,x 在目标模式中存在匹配的概率可由公式(7)计算得出:

$$P(x)=\frac{\sum_{y\in CAND(x)}^{y} sim(x,y)}{\sum_{y\in CAND(x)}^{y} sim(x,y)+d} \tag{7}$$

式中,d 为参数,在相同情况下,d 越大,x 存在候选匹配的概率越小,即算法越悲观;d 越小,x 存在候选匹配的可能性

越大,算法越乐观.

x 与候选匹配集 $CAND(x)$中元素 z 的相似概率可由公式(8)计算得出:

$$P(x,z)=\frac{sim(x,z)}{\sum_{y\in CAND(x)} sim(x,y)+d} \tag{8}$$

相似概率表示的是元素间相互匹配的概率,比相似度更直观.利用相似概率,不同的元素对之间可以通过相似概率直接进行比较,能够很直观地看出哪个元素对中的两个元素互相匹配的概率更大.第 2 节我们得到源模式中每个元素与其候选匹配之间的语义、子结构、父结构 3 种相似度,这里分别根据这 3 种相似度,利用公式(6)计算出语义相似概率、子结构相似概率、父结构相似概率,分别用 $Plism(x,y)$,$Pcsim(x,y)$,$Pasim(x,y)$表示.

还是承接本节开始的例子,在 d=0.5 的情况下,我们得到 $Plism(student.Addr,student.Address)$=0.286,$Plism(student.DepartmentID,student.DepartID)$=0.371,$Plism(student.DepartmentID,Department.DepartName)$=0.245,由此可以得出这样的推断:*student.Addr* 与 *student.Address* 的语义相似概率比 *student.DepartmentID* 与 *Department.DepartName* 的语义相似概率要高,尽管前者的语义相似度绝对值比后者低.由此可见,将所有元素与其候选匹配之间的相似程度用相似概率来表示后,不同元素对的相似概率可以直接进行比较,得到最优的映射结果.在映射生成的过程中,需要对目标模式中所有元素及其候选匹配之间的相似度进行统计比较.这时,用 3 种不同的相似概率来统计会使过程复杂,精确度降低,所以我们对这 3 种相似度进行合并,采用加权平均的方法合并得到每个元素对的相似概率 $Psim(x,y)$.见公式(9):

$$Psim(x,y)=\alpha\times Plsim(x,y)+\beta\times Pcsim(x,y)+\gamma\times Pasim(x,y),(\alpha+\beta+\gamma=1) \tag{9}$$

根据公式(7)得到元素对间的相似概率后,我们在已有的映射选取方法中用相似概率代替相似度来选取模式元素间的映射关系.下面我们用实验对算法的效率进行评价.

4 算法实验评价

本节我们将该方法与其他模式匹配中通用的方法进行实验对比.为了验证利用结构信息能够有效地提高模式匹配的精确度,我们选取常用的未利用结构信息的 COMA,SEMINT 方法与本方法进行比较.另外,为了验证本文方法中利用结构信息的方式能够更有效地提高模式匹配的精确度,我们又选取了比较常用的利用结构信息辅助匹配的 CUPID,SF 方法来进行比较.实验结果我们选取模式匹配方法的研究中最常用、最能够反映模式匹配方法性能的查准率、查全率和全面性这 3 项指标来进行对比[5−7].

(1) 查准率(precision):匹配结果中正确匹配结果占所有匹配结果的比率:

$$Precision=T/P=T/(T+F).$$

(2) 查全率(recall):匹配结果中正确匹配结果占实际匹配结果的比率:

$$Recall=T/R.$$

(3) 全面性(overall):通过使用匹配算法所节省的工作量占总的匹配工作量的比率:

$$Overall=precision\times\left(2-\frac{1}{Recall}\right)=\frac{T-F}{R},$$

其中,T 为匹配算法返回的正确匹配结果,P 为匹配算法返回的所有匹配结果,F 为匹配算法返回的错误匹配结果,R 为所有正确的匹配结果.

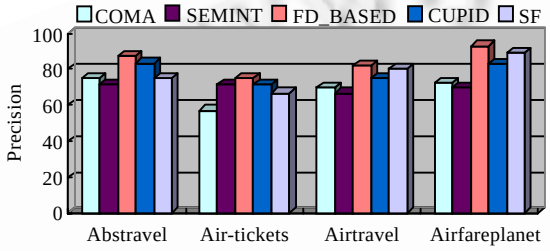
算法测试过程中采用的模式结构取自 http://metaquerier.cs.uiuc.edu/repository/.该模式库搜集的模式结构都来源于实际使用的模式,具有极强的代表性,在模式匹配的研究中经常被使用.库中共列举了 5 种数据源(TEL-8 Query Interfaces,BAMM Extracted Query Schemas,ICQ Query Interfaces,IWRandom,OntoBuilder).本实验采用数据源 ICQ Query Interfaces 中的模式结构信息.我们从 ICQ Query Interfaces 数据源中的 *Airfare* 域选取模式 *aa* 为目标模式,该域中其他模式为源模式,并分别用不同的匹配算法进行模式匹配(源模式分别为 Abstravel,Air-tickets,Airtravel,Airfareplanet).测试中将模式及其对应数据导入 MySql5.0 数据库中,使用 ODBC 连接数据库

以获取各种模式信息.由于实验结果并不涉及算法的时间特性,所以测试环境的硬件及系统软件配置对实验结果无任何影响,这里不作介绍.

实验结果如图 3 所示.由图 3(a)查准率对比图中不难看出,在两种未利用结构信息的方法 COMA 和 SEMINT 与利用了结构信息的 3 种算法 FD_BASED,CUPID 和 SF 的对比中,前者的查准率显然低于后者.由于查准率是正确匹配结果占所有匹配结果的比率,所以利用结构信息能够提高匹配结果中正确匹配的比率.从本算法与另外两种利用了结构信息的算法 CUPID 和 SF 的对比中我们也可以看出:本方法可以更有效地利用结构信息提高匹配查准率,也就是提高匹配结果中正确匹配所占的比率.

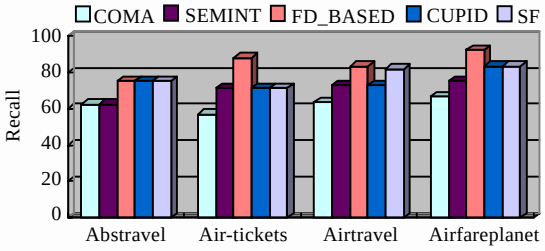
由于查准律表示结果中正确匹配所占的比率,较高的查准率只能说明匹配结果的正确性越高.但模式匹配的任务不仅要求结果都是正确的,还要求尽可能地找出所有的正确结果,查全率就是对该性能进行描述的指标,表示查询结果中的正确匹配结果占实际匹配结果的比率.图 3(b)是几种匹配方法的查全率指标的对比,图形显示,前两种未利用结构信息的方法(COMA,SEMINT)在查全率指标上明显低于后面 3 种利用了结构信息的方法(FD_BASED,CUPID,SF),也就是说,结构信息的利用能够有效地提高模式匹配算法的查准率.在后面 3 种算法的对比中,本算法在查全率指标上也优于其他两种(CUPID,SF)利用了结构信息的算法,说明本方法可以更有效地利用结构信息,找出更多的正确匹配结果.

图 3(a)和图 3(b)分别对查准率和查全率这两项指标进行了对比,图 3(c)是对全面性指标的对比.全面性指标是指为了得到最终的正确匹配结果,使用该匹配算法在工作量上的节省率.从图 3 中前面两种未利用结构信息的方法(COMA,SEMINT)与后面 3 种利用了结构信息方法(FD_BASED,CUPID,SF)的数据对比可以看出,利用结构信息能够有效地提高算法的全面性指标.另外,从本算法与其他两种利用了结构信息的算法(CUPID,SF)的对比可以看出,本算法在全面性指标上也优于其他两种算法.



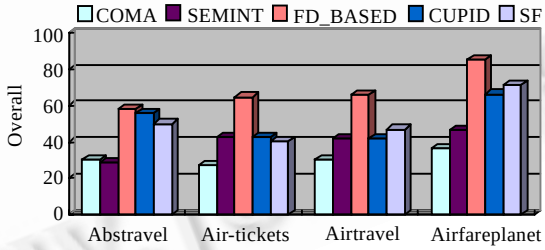
(a) Contrast graph of the precision of several match methods

(a) 各种匹配方法针对不同源模式的查准率对比图



(b) Contrast graph of the recall of several match methods

(b) 各种匹配方法针对不同源模式的查全率对比图



(c) Contrast graph of the overall of several match methods

(c) 各种匹配方法针对不同源模式的全面性对比图

Fig.3 Contrast of several match methods

图 3 几种不同匹配方法指标对比

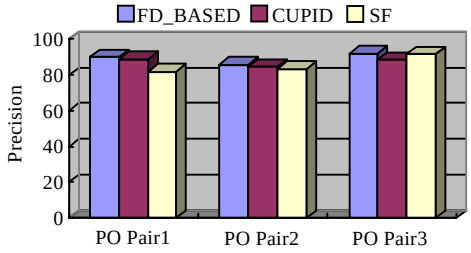
以上是对本文方法与模式匹配中常用的典型方法针对关系数据库进行对比的实验.由于本文方法可以方便地扩展至 XML,ER 图等其他类型模式,所以我们针对 XML 数据源进行实验来测试算法的可扩展性能.实验中使用的 XML 数据源采用文献[17]中提供的 3 对 PO 和 PurchaseOrder 模式,分别用 PO Pair1,PO Pair2 和 PO Pair3 表示,表 3 列出了 3 对模式的属性数对比情况.

Table 3 Contrast of the number of attributes in PO Pair1, PO Pair2 and PO Pair3

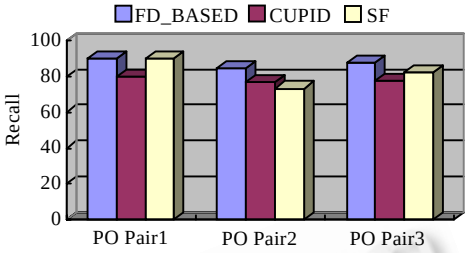
表 3 PO Pair1,PO Pair2 和 PO Pair3 中各模式属性数对比

	PO Pair1	PO Pair2	PO Pair3
PO	10	13	40
PurchaseOrder	9	15	43

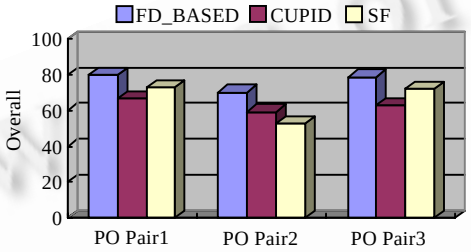
图 4 给出了本文方法与利用结构信息的 SF 方法和 CUPID 方法的实验对比情况.



(a) Precision contrast of three algorithms for PO Pair1, PO Pair2 and PO Pair3
(a) 3 种结构匹配方法针对 PO Pair1,PO Pair2 和 PO Pair3 的查准率对比图



(b) Recall contrast of three algorithms for PO Pair1, PO Pair2 and PO Pair3
(b) 3 种结构匹配方法针对 PO Pair1,PO Pair2 和 PO Pair3 的查全率对比图



(c) Overall contrast of three algorithms for PO Pair1, PO Pair2 and PO Pair3
(c) 3 种结构匹配方法针对 PO Pair1,PO Pair2 和 PO Pair3 的全面性对比图

Fig.4 Contrast of CUPID, SF and FD_BASED for XML schema

图 4 CUPID,SF 和 FD_BASED 针对 XML 模式匹配的指标对比

从图 4(a)的查准率对比图我们可以看出,FD_BASED 方法针对 XML 模式进行匹配的查准率指标略高于 CUPID 和 SF 方法,并且随着属性数目的增加,查准率未发生明显变化.同时,根据图 4(b)和图 4(c)对查全率和全面性指标的对比,我们可以发现与图 4(a)相同的规律.据此可知,FD_BASED 方法具有良好的可扩展性能,扩展至 XML 模式结构后仍然能够保持较高的查准率、查全率和全面性指标,并且在各项指标上都优于其他同类算法.

综上可以看出,利用模式结构信息能够提高模式匹配算法的各项指标,而本方法更加有效地利用了模式的结构信息,提高了算法的精确度,并且在方便地扩展至其他形式模式结构的同时,还能维持较好的匹配性能.

5 总结与展望

本文提出了一种利用模式元素间的函数依赖关系建立模式结构、使用元素相似概率来度量元素间相互匹配程度的新方法,并从理论分析和实验结果两个方面论证了这些改进能够在一定程度上提高模式匹配的精确度.目前,模式匹配方法中所使用的模式结构信息的种类还比较单一(比如在关系数据库系统中主要是主键与外键关系),信息量还比较少,其在匹配中的主要作用表现在:当同一元素的两个候选匹配自身信息的各方面都比较接近时,利用结构信息来辅助选择.今后我们可以通过其他一些方式获取更丰富的模式结构信息,使其在模式匹配的初期就能参与匹配,并在模式匹配过程中发挥更大的作用,提高匹配的精确度.

References:

[1] Zhao HM. Semantic matching across heterogeneous data sources. Communications of the ACM, 2007,50(1):45−50.

[2] Bohannon P, Elnahrawy E, Fan WF, Flaster M. Putting context into schema matching. In: Proc. of the VLDB. 2006. http://www.vldb.org/dblp/db/conf/vldb/vldb2006.html

[3] Bilke A, Naumann F. Schema matching using duplicates. In: Proc. of the 21st Int'l Conf. on Data Engineering (ICDE). 2005. http://www.informatik.uni-trier.de/~ley/db/conf/icde/icde2005.html

[4] Madhavan J, Bernstein PA, Doan AH, Halevy A. Corpus-Based schema matching. In: Proc. of the 21st Int'l Conf. on Data Engineering (ICDE). 2005. http://www.informatik.uni-trier.de/~ley/db/conf/icde/icde2005.html

[5] Do HH, Rahm E. COMA—A system for flexible combination of schema matching approaches. In: Proc. of the VLDB. 2002. http://www.vldb.org/dblp/db/conf/vldb/vldb2002.html

[6] Aumueller D, Do HH, Massmann S, Rahm E. Schema and ontology matching with COMA++. In: Proc. of the SIGMOD. 2005. http://www.informatik.uni-trier.de/~ley/db/conf/sigmod/sigmod2005.html

[7] Melnik S, Garcia-Molina H, Rahm E. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In: Proc. of the ICDE. 2002. http://www.informatik.uni-trier.de/~ley/db/conf/icde/icde2002.html

[8] Madhavan J, Bernstein PA, Rahm E. Generic schema matching with CUPID. In: Proc. of the VLDB. 2001. http://www.vldb.org/dblp/db/conf/vldb/vldb2001.html

[9] Xu L, Embley DW. Discovering direct and indirect matches for schema elements. In: Proc. of the 8th Int'l Conf. on Database Systems for Advanced Applications (DASFAA 2003). 2003. http://www.informatik.uni-trier.de/~ley/db/conf/dasfaa/dasfaa2003.html

[10] Gusfield D, Irving R. The Stable Marriage Problem: Structure and Algorithms. Cambridge: MIT Press, 1989.

[11] Rahm E, Bernstein PA. A survey of approaches to automatic schema matching. VLDB Journal, 2001,10(4):334–350.

[12] Salton G. The SMART Retrieval System—Experiments in Automatic Document Retrieval. Englewood Cliffs, 1971.

[13] Domingos P, Pazzani M. On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning, 1997,29(2-3):103–130.

[14] Li WS, Clifton C. SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks. Data & Knowledge Engineering, 2000,33(1):49–84.

[15] Sa SX, Wang S. An Introduction to Database System. 3rd ed., Beijing: Higher Education Press, 2000. 183–187 (in Chinese).

[16] Lu ZN, Zhang HS. Foundation of Operations Research. 2nd ed., Hefei: University of Science and Technology of China Press, 2006. 117–123 (in Chinese).

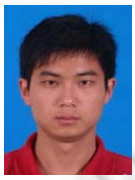
[17] Tansalarak N, Claypool KT. QMatch-Using paths to match XML schemas. Data & Knowledge Engineering, 2007,60(2):260–282.

附中文参考文献:

[15] 萨师煊,王珊.数据库系统概论.第 3 版,北京:高等教育出版社,2000.183–187.

[16] 路正南,张怀胜.运筹学基础教程.第 2 版,合肥:中国科学技术大学出版社,2006.117–123.

李国徽(1973－),男,湖南衡阳人,博士,教授,博士生导师,CCF 高级会员,主要研究领域为现代数据库技术,传感器网络,实时计算.



杜小坤(1980－),男,博士,主要研究领域为模式匹配.

胡方晓(1983－),男,博士,主要研究领域为嵌入式实时系统的容错,节能调度,现代数据库事务处理技术.

杨兵(1975－),男,博士,主要研究领域为移动实时数据库,传感器网络.

唐向红(1979－),男,博士,主要研究领域为现代数据库技术.