ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.2, February 2009, pp.234−245
doi: 10.3724/SP.J.1001.2009.03271

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

本体辅助的自动化模式匹配技术*

刘 强[1,2+], 赵 迪[1,2], 钟 华[1], 黄 涛[1]

[1](中国科学院 软件研究所 软件工程技术研究开发中心,北京 100190)

[2](中国科学院 研究生院,北京 100049)

Ontology-Aided Automatic Schema Matching

LIU Qiang[1,2+], ZHAO Di[1,2], ZHONG Hua[1], HUANG Tao[1]

[1](Technology Center of Software Engineering, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

[2](Graduate University, The Chinese Academy of Sciences, Beijing 100049, China)

+ Corresponding author: E-mail: liuq@otcaix.iscas.ac.cn

Liu Q, Zhao D, Zhong H, Huang T. Ontology-Aided automatic schema matching. *Journal of Software*, 2009,20(2):234−245. http://www.jos.org.cn/1000-9825/3271.htm

Abstract: This paper introduces an ontology-aided schema matching method in the mapping-based data exchange framework. The decision tree learning and WordNet are used to match attribute names. A data type ontology is constructed to compute semantic distance between attribute data types. Also, domain ontologies can be used to detect 1:*n* semantic matches. The three steps improve the match quality steadily. Experiments of several real applications show encouraging results, yielding high precision and recall measures.

Key words: schema matching; ontology; decision tree learning; database

摘 要: 在基于映射的数据交换系统框架下,提出了一种本体辅助的模式匹配方法.它利用 WordNet 词汇本体和决策树学习相结合的方法进行属性名称匹配,构建数据类型本体计算属性数据类型的语义距离,依赖领域本体发现一对多的语义匹配关系,这 3 个过程逐步提高了匹配质量.建立在实际应用数据上的实验结果表明,该方法具有较高的精确度和召回率.

关键词: 模式匹配;本体;决策树学习;数据库

中图法分类号: TP311 **文献标识码**: A

模式匹配是指把两个模式作为输入,计算模式元素间映射关系的过程[1].模式可以是关系数据库模式、XML schema/DTD 和本体等类型,其元素也相应地包括关系、表、XML Element、类和属性等.一般来说,这个过程是半自动化的,一方面是因为所依赖技术的不精确性,另一方面则是由于语义的复杂性.半自动化的模式匹配技术已被广泛应用于数据集成、数据仓库、电子商务等领域,对建设工业级产品起着重要的作用.

综述文献[2,3]对模式匹配问题进行了较好的总结.根据输入的模式元素特征分类,包括基于模式(schema)的、基于实例(instance)的和基于结构(structure)的 3 种类型.其中,基于模式的匹配技术仅考虑元数据信息,利用

* Supported by the National Natural Science Foundation of China under Grant No.60573126 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2004AA112010 (国家高技术研究发展计划(863))

Received 2007-10-19; Accepted 2008-02-01

字符串分析技术和语言学特征等.基于实例的技术仅考虑数据实例信息,采用信息检索技术和统计分析技术等.而基于结构的技术关注元数据之间的关系,主要使用图匹配技术发掘元数据所处的语义上下文.具体采用何种技术,与应用所决定的可获取元素信息相关.

本文的研究动机源于对数据交换中间件 OnceDI[4]的深入研发,其主要功能是实现关系数据库/XML 文档/平面文件之间的数据交换.与传统的集成模式不同,它并不是采用 GAV/LAV[5]方法生成全局模式,而是直接在参与交换的数据源之间构建模式元素的映射关系,以便进行数据查询和抽取.然而,OnceDI 在诸多方面仍存在不足:仅能自动匹配名称相同的模式元素,但这可能由于同名异义而造成错误,如属性“phone”在不同的关系中语义可能分别是“电话号码”和“电话机”;不能自动构建属性数据类型间的映射关系,如 Oracle 数据库的“number”数据类型和 XML schema 的“decimal”数据类型;无法解决属性间的语义冲突,如不同的度量单位;不具备发现一对多映射关系的能力,如“name=first_name+last_name”.

本文结合对 OnceDI 数据交换中间件的研发实践,对关系数据库的模式匹配问题进行研究.主要考虑把模式元素作为问题的输入,而很少使用数据实例,因为机器学习和统计分析技术作用在数据实例上的效果并不理想[3].与其他技术不同,本文采用本体(ontology)辅助的方法.本体是对共享概念的形式化和显式的表示[6],容易表达概念间的语义关系.本文的贡献在于:采用决策树学习(decision tree learning)和 WordNet 词汇本体[7]相结合的方法计算属性名称匹配;定义属性数据类型本体和属性语义冲突本体解决属性匹配问题;利用领域本体构建属性间一对多的映射关系.实验结果表明,该方法具有较高的精确度(precision)和召回率(recall).

本文首先对所研究的问题以及结果的度量进行形式化的定义,然后逐一描述属性名称匹配、属性数据类型匹配、属性语义冲突消解和一对多映射的计算方法,并给出实验评估及讨论,最后与相关工作进行比较.

1 问题定义

本节首先定义数据交换系统的框架,包括源和目标模式的定义以及映射元素的定义,然后描述映射计算结果的表示形式,最后给出全文可用的举例.

1.1 数据交换系统的框架

定义 1. 数据交换系统 X 是一个三元组$(D,\{S_i\},\{M_i\})$,D 表示目标数据源模式,$\{S_i\}$是 n 个源数据源模式的集合,$\{M_i\}$是 n 个源-目标映射的集合,每个源数据源模式 S_i 对应一个源-目标的映射 M_i.其中,$1\le i\le n$.

定义 2. 源数据源模式 S_i 或目标数据源模式 D 包括:关系的有限集合 R;属性的有限集合 A;属性数据类型的有限集合 T;获取关系属性的函数 $attr:R\to 2^A$;获取属性数据类型的函数 $type:A\to T$.

定义 3. 源-目标的映射 M_i 记作$\theta(\sum S_i)\to\sum D$.其中,$\sum S_i$ 表示源数据源模式 S_i 的元素集;$\sum D$ 表示目标数据源模式 D 的元素集;θ作为映射表达式表示如何由$\sum S_i$ 中的元素生成对应$\sum D$ 中的元素,$\theta\in\{\varnothing,Union,Select,Merge,Split\}$.对于一对一的映射关系,$\theta=\varnothing$,不作任何操作;对于一对多的映射关系,$\theta$可以是其他一种操作.

从以上定义可以看出,每个目标数据源的模式元素对应源数据源模式中的 1 个元素或者多个元素的虚视图,因此,对于数据查询操作来说,仅需按照θ操作进行规则展开,相对于 GAV/LAV 方法的查询重写(query reformulation)更为简单、高效.

1.2 映射的计算

定义 4. 为了发现模式 S 和模式 D 之间的映射关系,共使用 l 种方法来计算任意两个模式元素的相似度,其取值区间为[0,1].S 的模式元素包括 $s_1,s_2,\ldots,s_n$,D 的模式元素包括 $d_1,d_2,\ldots,d_m$,那么,使用第 $k(1\le k\le l)$种方法计算 S 和 D 的相似度矩阵 M_k 见表 1,其中,$1\le i\le n$,$1\le j\le m$.

Table 1 Similarity matrix M_k computed by method k

表 1 第 k 种方法计算的相似性矩阵 M_k

	d_1	d_j	d_m
s_1	$sim(s_1,d_1)$	…	$sim(s_1,d_m)$
s_j	…	$sim(s_i,d_j)$	…
s_n	$sim(s_n,d_1)$	…	$sim(s_n,d_m)$

总相似度的计算方法为 $M_{SD}=\alpha_1M_1+\alpha_2M_2+\ldots+\alpha_kM_k+\ldots+\alpha_lM_l$.其中,$\alpha_1+\alpha_2+\ldots+\alpha_k+\ldots+\alpha_l=1$,$\alpha_k$ 表示所采用方法的权重,其取值由实验确定.当 $sim(s_i,d_j)$大于确定的阈值β时,认为 s_i 和 d_j 具有映射关系.

1.3 模式示例

为了清楚地说明所采用的方法和计算过程,给出一对源和目的数据源的模式示例,如图 1 所示.

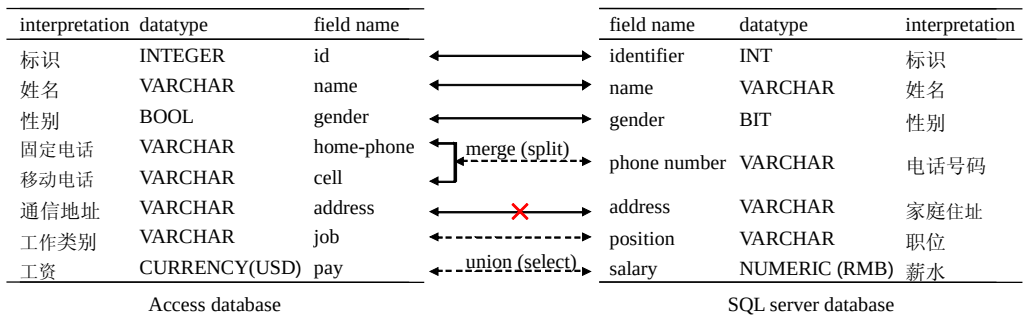


Fig.1 Two schemas of employee and their mapping relationships

图 1 表示雇员信息的两种数据库模式及其映射关系

假设图 1 中的两个数据库模式分别表示一个外资企业和一个中资企业的雇员信息,现由于公司兼并要在两数据库之间进行数据交换.使用本文的方法自动构建模式间的映射信息,如图中箭头所示.实线表示完全匹配的映射,虚线表示具有一定的相似度,“merge(split)”和“union(select)”表示一对多匹配的映射表达式.

对于属性名称匹配,首先对属性名进行规范化处理,如把简写“id”扩充为全称单词“identifier”,把复合词组“home-phone”拆解为独立的单词.然后使用 WordNet 本体与机器学习相结合的方法,计算每对属性名称的相似度.经过该步骤处理,可以初步得出一些映射关系.例如,模式 1 中的“id”、“name”分别与模式 2 中的“identifier”、“name”完全匹配,模式 1 中的“job”与模式 2 中的“position”具有一定的相似度.

在属性名称匹配的基础上开发数据类型本体,进一步辅助计算属性数据类型的相似度,如两个模式中属性“gender”的数据类型“BOOL”与“BIT”等价,模式 1 中属性“pay”的数据类型“CURRENCY”与模式 2 中属性“salary”的数据类型“NUMERIC”具有一定的相似度.然而,由于“pay”与“salary”的度量单位不同,还需要使用语义冲突本体解决数据转换的一致性问题.

属性名称的匹配不能发现一对多的映射关系,进一步借助于轻量级的语义本体,得知“home-phone”和“cell-phone”与“phone number”之间存在“part-of”关系,可以使用“merge”操作构建一对多的映射关系.

从以上步骤可以看出,各种类型的本体在匹配过程中起到了至关重要的作用,不仅提高了映射发现的精确度,而且融合了模式元素的语义,能够辅助发现更多类型的映射关系.第 2 节~第 4 节将详细介绍这些方法.

2 属性名称匹配

名称匹配是模式匹配最重要的手段,已有的研究方法[8−12]包括:使用自然语言处理技术对名称进行规范化处理,找出要匹配的基本名称元素,然后使用语法和语义分析技术计算名称之间的相似度.语法分析主要是进行字符串处理,语义分析主要是利用电子词典或领域本体[13]对词汇的组织方式(synonym,hypernym)计算其语义距离.本节介绍已提出的名称匹配方法,相关工作比较在第 6 节中加以介绍.

2.1 名称规范化处理

我们采用 3 个步骤对名称进行规范化处理:(1) 拆分复合词组.使用符号分析器,把复合词组拆分为独立的单词,如“landscape-in-many-cities”→⟨landscape,in,many,cities⟩.(2) 还原词汇变体.查找 WordNet,还原单词的基本形式,如“cities”→“city”,“id”→“identifier”.(3) 去除修饰词,仅留下名词,如⟨landscape,in,many,city⟩→⟨landscape,city⟩.

当计算复合词组 A 与某个单词 b 的相似度时,需要逐一计算 A 中的单词与 b 的相似度,然后求和.同时,考虑复合词组中不同单词的重要性,为其相似度赋予不同的权重.假设 $a_1,a_2,\ldots,a_k$ 是 A 中按重要性顺序排列的 k 个单词,那么 A 和 b 的相似度计算公式记作:

$$sim(A,b)=\sum_{r=1}^{k}\frac{1}{r^2\cdot N}\times sim(a_r,b),N=\sum_{r=1}^{k}\frac{1}{r^2}.$$

如果要匹配的另一方也是复合词组 $B,b_1,b_2,\ldots,b_l$ 是 B 中按重要性顺序排列的 l 个单词,那么 A 和 B 的相似度计算公式记作:

$$sim(A,B)=\sum_{q=1}^{l}\frac{1}{q^2\cdot M}\times\left[\sum_{r=1}^{k}\frac{1}{r^2\cdot N}\times sim(a_r,b_q)\right],N=\sum_{r=1}^{k}\frac{1}{r^2},M=\sum_{q=1}^{l}\frac{1}{q^2}.$$

2.2 名称匹配

已有的利用词汇本体计算相似度的方法大多仅关心某种语义关系,如同义关系(synonymy)、上下文关系(hypernymy)等,而没有考虑多种语义关系之间的相互影响.例如,一个名词可能由于其多义性而与另一名词出现随机匹配的情况.因此,全面考虑多种语义关系,并使用合适的方法归纳经验,能够有效地提高名称匹配的准确性.

WordNet 是一个自足的词汇数据库,其名词的基本语义关系包括同义关系、上下位关系、整体部分关系(meronymy)、反义关系(antonymy)、近义关系(polysemous)等,可利用的特征多达 20 多种.假设计算 A 和 B 的相似度,为使产生的结果更直观、可验证,本文仅使用最重要的 4 个特征:同义关系($c1$),A 和 B 与其共同上位词的距离和($c2$),A 和 B 共同上位词的个数($c3$),A 和 B 多义词的总和($c4$).

本文的目的是根据这些特征归纳出规则,在确定的规则下计算 A 和 B 的相似度.决策树学习[14]是应用最广的归纳推理算法之一,它是一种逼近离散值函数的方法.对应于本节的问题,目标函数具有离散的输出值:匹配为“YES”,不匹配为“NO”.学习得到的决策树能够被表示为多个 if-then 规则,与神经网络方法相比较,具有很好的可读性和可验证性.从数据库中选取 140 对正例和 140 对反例,即 140 对匹配的(A,B)和 140 对不匹配的(A,B),根据 WordNet 计算(A,B)对的以上特征值,把这些作为算法的输入(使用 C4.5 算法[15]),生成的决策树如图 2 所示.

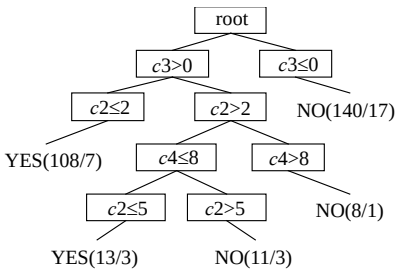


Fig.2 Resulting decision tree trained by 280 (A,B) pairs

图 2 由 280 对(A,B)数据训练生成的决策树

在生成的决策树中,$c3$ 的分类能力最强,被作为树的根节点测试,这意味着,当 A 和 B 不能找到一个共同的上位节点时,它们是没有关系的(当 $c3\le0$ 时,分类出 140 对反例).$c2$ 的分类能力次之,也就是说,当 A 和 B 与共同上位节点的距离和较小时($c2\le2$),相似度较大.如果 A 和 B 的词义较多($c4>8$),它们的匹配程度出现了一定的随机性,因此剔除了部分结果(8 个).然而,即使在 A 和 B 的多义性较小时,决策规则仍然不接受那些($c2>5$)的数据(11 个).值得注意的是,规则 $c1$ 并没有出现在决策树中,并不是因为没有符合条件的数据,而是规则 $c2$ 覆盖了它(A 和 B 与共同上位节点的距离和为 0 等价于同义异名).

在叶子节点中,使用(x/y)给出了训练实例的总个数 x 和不正确分类的实例个数 y,其中 y 的个数由人工检查确定,本文把($x-y/x$)作为 A 和 B 的相似度.表 2 给出了使用图 2 决策树作用在图 1 示例上的相似度矩阵.

Table 2 Similarity matrix computed by WordNet

表 2 利用 WordNet 计算得出的相似度矩阵

	identifier	name	gender	phone number	address	position	salary
id (identifier)	1.00	0.27	0.27	0.29	0.27	0.13	0.27
name	0.27	1.00	0.27	0.78	0.13	0.13	0.27
gender	0.27	0.27	1.00	0.24	0.13	0.13	0.27
home-phone (phone home)	0.24	0.78	0.24	0.69	0.13	0.13	0.27
cell-phone (phone cell)	0.27	0.78	0.24	0.69	0.13	0.13	0.27
address	0.27	0.13	0.13	0.13	1.00	0.94	0.27
job	0.13	0.13	0.13	0.13	0.13	0.94	0.13
pay	0.27	0.27	0.27	0.27	0.27	0.13	1.00

在经过名称的规范化处理后,直接赋予名称相同的(A,B)对相似度为 1.那些在 WordNet 词典中异名同义的词汇相似度也被置为 1(“pay”和“salary”).对于基本单词对,例如(“address”,“position”),由于“position”具有对“address”直接的上位关系(c2=2),根据决策树计算其相似度为(108−7)/108=0.94.又如(“address”, “gender”),由于它们与共同上位词汇“abstraction”的距离之和大于 2(c2>2),且多义度之和为 10(c4>8),计算其相似度为 1−[(8−1)/8]=0.13.对于复合词组的相似度计算,则使用第 2.1 节所述的方法.

根据表 2 所示的结果,取相似度的最大值,可以初步得出一些匹配关系:(“id”,“identifier”),(“name”,“name”),(“gender”,“gender”),(“home-phone”,“name”),(“cell-phone”,“name”),(“address”,“address”),(“job”,“position”),(“pay”,“salary”).可以看出,这样的结果并不理想,主要原因有 3 个方面:(1) 由于语义的多样性,(“home-phone”,“name”)的相似度超过了(“home-phone”,“phone number”)的相似度,在 WordNet 中确实存在着“phone”和“name”具有共同的上位词“language unit”,但这种“phone”的词义(“sound”)不是所期望的.借助于语法特征(“phone”是数字,“name”是字符串)可以解决该类问题.(2) 由于“pay”和“salary”是异名同义词,直接赋予其相似度为 1,然而其属性语义不同,分别由不同的度量单位表示,直接进行数据交换将会发生错误.第 3 节将解决该问题.(3) 属性名称匹配不能发现一对多的映射关系,这个问题将在第 4 节加以解决.

3 属性匹配

属性包括属性名、属性数据类型和属性语义 3 个重要的部分.仅仅属性名称匹配是不够的,还需要考虑其他两个重要的因素.例如,(“birthday”,dateTime)和(“birthday”,number(4,0))由于数据类型的不同造成不完全匹配,(“height”,int,unit(meter))和(“height”,int,unit(decimeter))则因为度量单位的语义不同造成不能直接进行数据交换.本节给出属性数据类型匹配的计算方法和属性语义冲突的消解方法.

3.1 属性数据类型匹配

XML Schema 规范的第 2 部分[16]定义了一个完备的数据类型系统,其主要目的之一是满足数据库系统的数据交换.借助 WordNet 的词汇组织方式,基于该类型系统,可以建立囊括各数据库数据类型的本体 O_D,描述参与交换的各种数据类型间的语义关系,从而计算其相似度.

定义 5. 数据类型本体 O_D 可以表示为一个四元组(D_{XML},D_{DB},$R_{is\text{-}a}$,R_{syn}).其中,D_{XML} 表示 XML 数据类型的集合,D_{DB} 表示数据库数据类型的集合,$R_{is\text{-}a}$ 表示 D_{XML} 数据类型之间的关系,R_{syn} 表示 D_{DB} 中数据类型与 D_{XML} 中数据类型的同义关系.

因为 XML 数据类型系统的构建是在基本数据类型的基础上进行聚集(list,union)或者约束形成的,所以可以用 is-a 表示其间关系.另一方面,由于 XML 数据类型系统作为数据库数据的交换标准是完备的,可以在 D_{XML} 中为 D_{DB} 中的任意一个元素找到同义节点,使用 R_{syn} 表示.图 3 给出了 O_D 的部分片断.

在图 3 中,实线方框表示 D_{XML} 中的一个元素,虚线方框表示 D_{DB} 中的一个元素.例如,SQL Server 数据库和 Access 数据库分别使用“bit”和“bool”表示 boolean 数据类型.“→”表示关系 $R_{is\text{-}a}$,“—”表示关系 R_{syn}.值得注意的是,关系 $R_{is\text{-}a}$ 和 R_{syn} 均具有传递性,R_{syn} 还具有对称性.

基于 O_D,按如下规则计算任意两个数据类型的相似度:(1) 具有关系 R_{syn} 的任意两个数据类型的相似度为

1,如“bit”和“bool”.(2) 具有关系 $R_{is\text{-}a}$ 的任意两个数据类型的相似度为 1,如“integer”和“smallInt”(即使在发生单向转换“integer”→“smallInt”时可能发生错误,但这往往不反映用户的意图,将会对数据库设计加以调整以满足数据交换).(3) 任意其他两个数据类型的相似度为 $1-(DIS_{total}/DIS_{\max})$,$DIS_{total}$ 表示两个数据类型与其共同上位节点的距离之和,$DIS_{\max}$ 表示在 O_D 中任意两个数据类型与其上位节点的最大距离之和.在 O_D 设计完成后,已知 $DIS_{\max}=9$.例如,“money”与“varchar”的相似度为 $1-(4/9)=0.56$.

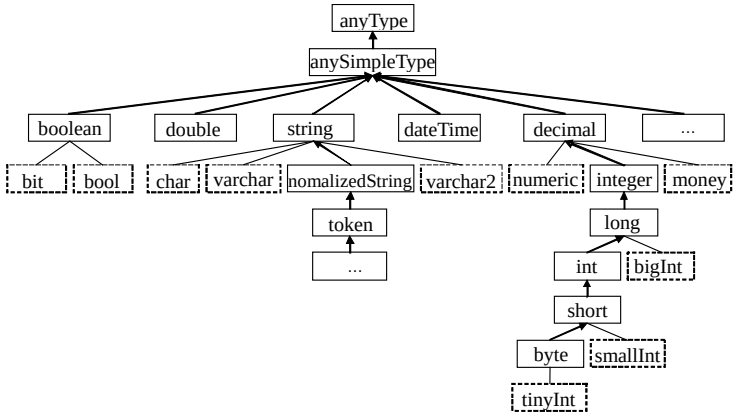


Fig.3 A fragment of O_D

图 3 O_D 片断

如果把属性名称的相似度记作 sim_{DN},把属性数据类型的相似度记作 sim_{DT},那么在属性名称匹配的基础上,考虑属性数据类型的匹配,任意两个属性的相似度为 $sim_{DN}\times sim_{DT}$.然而,属性名称的相似度比属性数据类型的相似度更重要,因此,把 sim_{DT} 修正为 $sim'_{DT}=1(1+10^{-sim_{DT}})$,使其值域由[0,1]缩小至[0.50,0.91]的范围内.

对于图 1 中的模式示例,如果同时计算其属性名称和数据类型的相似度,则结果矩阵见表 3.

Table 3 Similarity matrix computed considering the name and datatype

表 3 同时考虑名称和数据类型计算得出的相似度矩阵

	identifier	name	gender	phone number	address	position	salary
id (identifier)	0.91	0.21	0.21	0.23	0.21	0.10	0.25
name	0.18	0.91	0.21	0.71	0.12	0.12	0.21
gender	0.18	0.21	0.91	0.19	0.10	0.10	0.21
home-phone (phone home)	0.16	0.71	0.19	0.63	0.12	0.12	0.21
cell-phone (phone cell)	0.18	0.71	0.19	0.63	0.12	0.12	0.21
address	0.18	0.12	0.10	0.12	0.91	0.86	0.21
job	0.09	0.12	0.10	0.12	0.12	0.86	0.10
pay	0.25	0.21	0.21	0.21	0.21	0.10	0.91

3.2 属性语义冲突消解

属性的语义冲突包括格式(format)冲突、度量单位(unit)冲突、精度(precision)冲突、默认值(default value)冲突和属性完整性限制(integrity constraint)冲突等类型[17].关系数据库仅能表示数据类型和数据长度等信息,不能表示该类隐含的语义信息.然而,在数据交换过程中,必须有效地消解这些冲突.使用本体可以显式地表示这些冲突属性及其转换关系,以达到冲突消解的目的.图 4 是定义的语义冲突分类本体 SCO 的一个片断.

在图 4 中,“_:H”表示 SCO 本体的顶点.“precision”和“unit”等表示语义冲突的一个子类,它还可以继续细分为更多的子类,如“unit”包含“length”和“weight”等.“meter”和“feet”等是“length”的可实例化子类,由链表结构组织,它们的实例(“meterToKillo”)表示与后继节点的转换函数名称,这些可实例化子类之间可以进行序列转换.

借助于关系数据库到本体的转换方法[18,19],可以对数据库中的该类语义信息进行显式的表示.在进行数据交换时,通过对本体 SCO 中节点的查找匹配,发现语义转换关系以进行冲突消解.语义冲突分类本体 SCO 的形

式化定义和冲突消解算法详见已完成的工作[20].

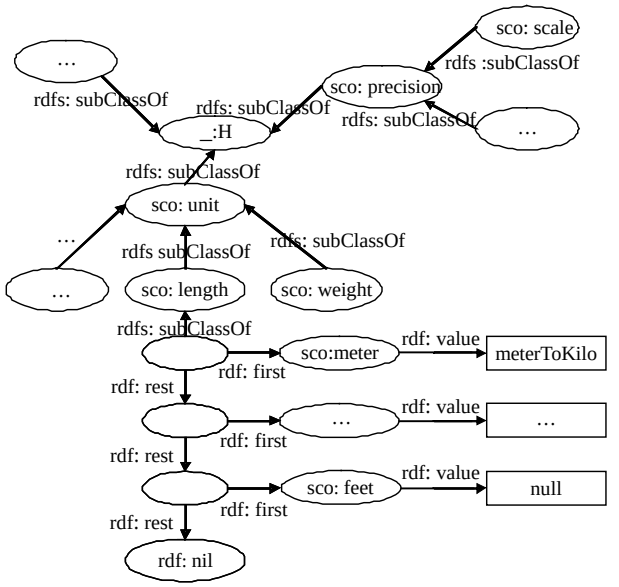


Fig.4 A fragment of SCO ontology

图 4 本体 SCO 的片断

4 一对多映射关系的模式匹配

前述方法仅能发现一对一的映射关系,对于一对多的模式匹配,一般包括语义关系和计算关系两种情况.语义关系如“part-of”和“is-a”等,计算关系的映射如(“listed-price”,“*price*×(1+*tax-rate*)”).可以说,计算关系在一定程度上也隐含着语义.文献[21]使用机器学习方法作用在数据实例上,可以发现数据库模式间一对多的计算关系,但是代价较大,而且不一定能够反映真正的语义关系.

目前,领域本体已被广泛应用于语义集成.即使在领域本体不可用时,数据库设计者也可以方便地借助本体编辑工具,设计一些轻量级的本体以反映实体间的语义关系.图 5 给出了一个轻量级本体示例.

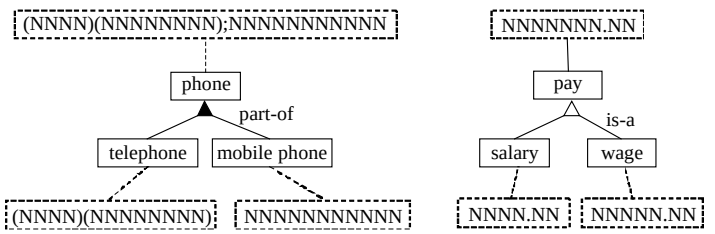


Fig.5 Example of lightweight domain ontology

图 5 轻量级本体示例

图 5 分别表示了具有“part-of”和“is-a”语义关系的两个本体,实线方框表示概念,虚线方框表示该概念实例的规则表达式(pattern).其中,N 表示该位置是一个数字.我们之所以称其为轻量级本体,是因为本体的构建与既有数据库模式无关,它是由领域专家根据现实实体间的语义关系构建的,不需要反映数据库模式中的全部概念.

为了使用轻量级本体辅助发现数据库模式间的语义匹配,首先要为数据库模式中的属性找到本体中对应的概念,如果得到了(“phone”,“phone number”),(“home-phone”,“telephone”),(“cell-phone”,“mobile phone”)之间的映射关系,则可以进一步发现(“phone number”,“home-phone merge cell-phone”)的匹配.显然,可以使用第 2 节所

述的名称匹配方法计算相似度,但对于该示例的效果并不理想.本文使用字符串模式匹配方法解决该问题.

计算字符串间的编辑距离(edit distance)是字符串模式匹配的最有效方法,其基本思想是:分别使用“A”、“S”和“N”替换字符串中的字母、符号和数字,对一个字符串进行插入字符、删除字符和修改字符等操作,使其变为另外一个字符串.这些操作的最小次数称为两字符串间的编辑距离.如,字符串“(010)88641252”转换为“SNNNSNNNNNNNN”,字符串“010-88641252”转换为“NNNSNNNNNNNN”,其编辑距离为 1.

对于数据库模式(包括源和目标)中的任何一个属性 A,根据编辑距离计算它与本体中任意一个概念 B 的相似度.属性 A 包含若干数据实例,逐一地对其进行模式替换,按出现次数的多少顺序选取 n 个模式 a_i,分别计算它们与 B 的模式 b 的编辑距离,记作 $Ed(a_i,b)$.那么,可以使用以下公式计算属性 A 与概念 B 的相似度:

$$sim(A,B)=\sum_{i=1}^{n}\frac{1}{i^2\cdot N}\times\frac{1}{Ed(a_i,b)+1},\ N=\sum_{i=1}^{n}\frac{1}{i^2}.$$

对于每个本体中的概念,选取数据库模式中与之相似度最大的属性,赋予这些属性间的关系为本体概念间的关系.例如,可由“telephone”和“mobile phone”与“phone”的“part-of”关系发现“home-phone”和“cell-phone”与“phone number”之间一对多的映射关系,在进行数据交换时使用“merge(split)”进行操作.

如果把发现的一对多匹配属性间的相似度置为 1,则未发现的属性间相似度设为 N/A.按照第 1.2 节所述的复合相似度计算方法,对表 3 的结果进行修正,相似度为 N/A 的属性不参与复合计算.设 α_1 表示属性匹配(包括名称和数据类型)的权重,α_2 表示一对多关系匹配的权重,α_1=0.7,α_2=0.3(实际值由实验确定),可得表 4 所示的结果.可以看出,“home-phone”和“cell-phone”与“phone number”的相似度已经超过了它们与“name”的相似度.

Table 4 Similarity matrix considering 1:*n* matching method

表 4 融合了一对多匹配方法的相似度矩阵

	identifier	name	gender	phone number	address	position	salary
id (identifier)	0.91	0.21	0.21	0.23	0.21	0.10	0.25
name	0.18	0.91	0.21	0.71	0.12	0.12	0.21
gender	0.18	0.21	0.91	0.19	0.10	0.10	0.21
home-phone (phone home)	0.16	0.71	0.19	0.74	0.12	0.12	0.21
cell-phone (phone cell)	0.18	0.71	0.19	0.74	0.12	0.12	0.21
address	0.18	0.12	0.10	0.12	0.91	0.86	0.21
job	0.09	0.12	0.10	0.12	0.12	0.86	0.10
pay	0.25	0.21	0.21	0.21	0.21	0.10	0.94

根据以上方法,只要可以使用本体表示模式元素相关概念的映射关系,就可以得到模式元素间的映射关系.对于计算关系,同样可以借助于轻量级本体来辅助发现.例如,使用数学表达式二叉树可表示任意的计算关系.

5 实验评估

5.1 实验数据

实验数据的选取应具有较好的可比性,理想的情况是使用标准数据与其他原型系统进行比较.然而,在该研究领域,实验准备比较困难,主要有两方面的原因:(1) 可用原型系统 COMA[10]和 Cupid[12]等主要是一种框架和平台,用于复合多种模式匹配算法,因此与具体的方法不具有可比性;(2) 大家比较认可的测试数据来自于文献[22],但它使用 DTD 表示,而且不包括数据类型和领域本体,应用本文所提出的方法将产生明显的差异,可比性不强.

因此,本文从 OnceDI 的实际应用领域选取数据,包括中央电视台 ChinaEPG 中心服务平台(ChinaEPG)、湖南省防汛(flood prevention)会商系统和烟台市信用信息(credit information)基础数据交换平台[4],目的是验证研究结果的有效性.同时,为了得到一对多的映射关系,构建了部分轻量级本体.表 5 列出了实验数据的特征.

在 3 个系统中,分别选择了一个源数据源和一个目的数据源进行数据交换,表中列出了两个数据源涉及到的数据库类型、表个数、属性个数和可匹配的属性(包括一对多映射)个数.

Table 5 Experiment data

表 5 实验数据

	Number of data source	Database type	Number of tables	Number of attributes	Number of matchable attributes
China EPG	2	Access 2000 vs. oracle 9i	8	54	42
Flood prevention	2	SQL server 2000 vs. oracle 9i	16	86	48
Credit information	2	Sybase 11.5 vs. SQL server 2003	12	72	54

5.2 评价指标

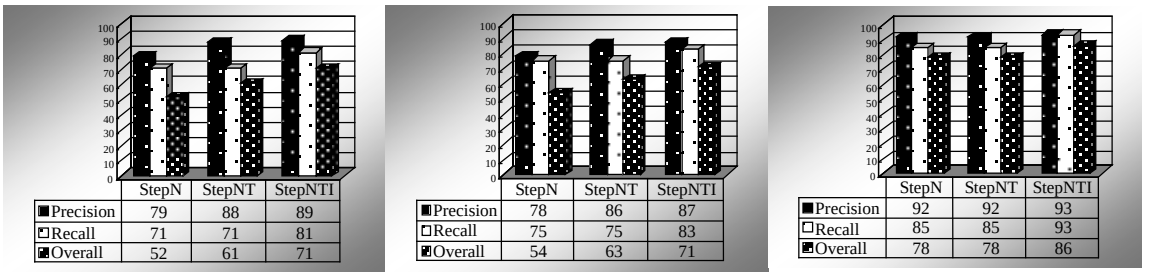
对本文所提出方法的评价指标包括精确度、召回率和全面性能(overall).假设领域专家人工发现的一对一和一对多匹配数为 A,本文方法自动发现的正确匹配数为 C,自动发现的非正确匹配数为 I,未自动发现的正确匹配数为 U,那么 3 个评价指标定义如下:

$$precision=\frac{C}{C+I},recall=\frac{C}{A},overall=1-\frac{I+U}{A}=recall\times\left(2-\frac{1}{precision}\right).$$

精确度和召回率两项指标来源于信息检索领域的研究成果,它们各自并不能正确反映匹配质量.如果返回了很多错误的匹配结果,那么精确度会较低,但是召回率却很高.同样地,如果仅仅返回了少部分正确的匹配结果,则精确度会很高,但是召回率就很低.*overall* 则全面反映了匹配质量.直观地,它综合使用了召回率和精确度指标,反映了为修正自动发现的错误匹配和弥补未发现的匹配所要付出的代价.当 *precison*<0.5 时,*overall* 取负值,这说明人工修补的代价超过了自动匹配的结果.理想情况下,$C=A,I=U=0$,则 $precison=recall=overall=1$.

5.3 实验结果

由于属性数据类型的匹配依赖于属性名称的匹配结果,一对多的模式匹配与前两者独立,所以使用 3 个步骤进行实验:(1) 属性名称匹配 $Step_N$;(2) 属性名称匹配和属性数据类型匹配 $Step_{NT}$;(3) 在属性名称匹配和属性数据类型匹配的基础上进行一对多的模式匹配 $Step_{NTI}$.图 6 给出了 3 种应用数据所产生的实验结果.



ChinaEPG Flood prevention Credit information

Fig.6 Experiment results from three data sources

图 6 3 种数据源对应的实验结果

在 3 种应用数据中,Credit Information 的源和目的数据源具有较高的相似性,接近于数据库复制的应用,而且两种数据库的数据类型具有较高的相似性(Sybase 和 SQL Server),所以获得了较高的精确度、召回率和整体性能.因为 3 个应用均属于数据交换,所以大部分属性可匹配,没有出现整体性能较低的情况.

属性名称匹配解决了大部分的自动匹配问题,分别获得了 79%,78%和 92%的精确度,但是因为没有考虑数据类型的匹配,可能造成了部分不正确的映射,$Step_{NT}$ 进一步修正了该问题,可以看到精确度进一步提升,分别达到了 88%,86%和 92%,但是召回率没有变化(不可能因为数据类型的匹配而发现更多的映射).在 Credit Information 应用中,由于数据类型的高度相似性,数据类型匹配并没有因帮助排除了匹配错误而提高精度.

在 $Step_{NTI}$ 中,由于领域本体的辅助,进一步发现了一对多的映射关系,使得召回率分别提高了 10%,8%和 9%.精确度也因为正确发现的匹配数 C 值的提高而升高.但是,这种提高并不明显(提高了 1%~3%),因为非正确

匹配数 I 值仍然没有变化.

与基于语法的名称匹配方法相比,决策树学习和 WordNet 词汇本体相结合的方法获得了较高的匹配质量(例如 LSD[22]方法获取的匹配正确性为 70%~80%).在此基础上,属性数据类型匹配和一对多的映射发现方法使得整体匹配质量获得了确定性的提高.即使在其他匹配方法的基础上使用 $Step_{NT}$ 和 $Step_{NTI}$,其效果也很明显.

6 相关工作比较

基于文献[2]提出的模式匹配分类框架,列举了一些相关工作与本文进行比较,以分析优点和不足之处.本文同时利用了模式信息和实例信息,包括属性名称、属性数据类型和实例的字符串模式.而没有考虑利用元素间的关系等结构特征和约束.更为重要的是,使用了 WordNet 词汇本体、数据类型本体和领域本体等外部信息,使得在匹配准确性和匹配基数(match cardinality)方面更加完善,可以辅助发现一对一和一对多的映射关系.

6.1 名称匹配

名称匹配主要是利用语法和语义特征计算模式元素的相似度,相关工作包括 Cupid[12],DIKE[23]和 LSD[22]等.Cupid 是一种混合匹配算法,同时利用了语法和结构特征,而且借助于领域本体识别词汇间的同义关系.匹配过程分为 3 个阶段:(1) 在对模式元素进行规范化和分类处理后,使用领域本体中词汇间的同义和上下位关系计算相似性,在这种关系不能获取时,相似度由字符串的共同子串计算得出;(2) 基于模式元素间的结构关系构建树,树节点元素的相似关系由叶子节点的相似度聚集计算得出;(3) 根据节点相似度的加权平均值获取元素的匹配关系.DIKE 主要解决实体-关系模型的元素匹配问题,它并不借助于通用词典,而是基于用户定义的部分词汇间的同义和包含关系,推导出新的同义和包含关系.LSD 主要使用机器学习方法计算元素的相似度,大量地使用实例信息,分为训练和匹配两个阶段.在训练阶段,根据用户提供的部分匹配结果,学习器经过训练,发现一些基于字符串模式或者数字特征的匹配规则.在匹配阶段,利用这些学习到的规则对新的元素计算相似度.

本文的名称匹配方法结合了已有技术的优点,同时也进行了改进.在名称规范化处理阶段,提出了复合词组的相似度计算方法,在实验中得到了较好的效果.在使用通用词典 WordNet 时,不仅考虑了同义关系和上下位关系,而且集成了上位词距离、上位词个数和多义词个数等特征,对语义关系的把握比较全面.在机器学习方法的使用上,决策树学习方法基于领域无关的训练数据,获取了通用的决策树,从而避免了 LSD 方法面对不同应用选择和训练不同数据集的繁琐性.LSD 方法的性能提高建立在训练数据的不断扩充上,这对于数据库中的海量数据集,获取合适的样本比较困难,而且效率较低.

然而,部分已有技术也值得借鉴,如 Cupid 和 DIKE 方法利用结构特征(元素间的相互关系)推导新的匹配关系.一方面,在领域本体或者词典不完全时,这种技术是较好的补充;另一方面,元素的上下文也往往影响着其本身的相似关系,如表 1 示例中 address 间的相似关系仅仅根据名称是匹配的,但是考虑其上下文(一个表示家庭住址,一个表示通信邮箱地址),它们的语义是有差别的.

6.2 属性数据类型匹配

目前,对于属性数据类型匹配的研究较少,一方面是因为它在模式匹配中处于次要的地位,属性数据类型的匹配往往在实际数据集成阶段才予以考虑;另一方面是因为通用模式匹配的研究无法把属性数据类型具体化.研究关系数据库的数据交换,则可以对该问题进行定量的分析研究.

相关工作包括文献[24]和 S-Match 方法[11].文献[24]提出了一个基于本体的包装器/协调器框架,使用全局本体和局部本体解决语法和语义方面的异构问题.对于属性数据类型的匹配,仅作了简单的分析:选择两者中更一般化(generalized)的数据类型作为集成模式中的数据类型;如果二者不能相互转换,选择一个第三方的数据类型,并构建它与二者的映射关系.显然,这种方法过于简单,而且不具有可操作性.S-Match 方法基于 XML Schema 的内置数据类型建立了数据类型本体,该本体表示数据类型间的 is-a 关系,在进行属性匹配时,同时考虑数据类型的关系,但是,该文献的目的是发现元素间的确定性关系(等价、一般化和不相交),而不是计算相似度.

本文方法与以上方法相比,具有明显的优点:数据类型本体囊括了参与集成的所有数据类型,并且构建了各

种数据类型与 XML Schema 基本数据类型的映射关系,这为数据交换过程中的类型转换做好了准备.数据类型匹配的计算是定量的,不是简单的粗粒度的分析,可以很容易地与属性名称匹配方法相结合计算出复合相似度.

6.3 一对多映射关系的模式匹配

该部分的相关工作主要是 iMap[21],它是对 LSD 方法的扩展,可发现如 *listed-price=price*×(1−*discount-rate*)和 address=concat(location,state)等复杂的映射关系.其思想是根据不同的可用信息类型,借助于不同类型的搜索器(如 Text Searcher,Numeric Searcher),使用 Beam Search 方法减小搜索空间(仅搜索 k 个候选元素),对搜索空间内的元素赋予不同的操作符(例如对 Numeric Searcher 使用数学运算符)进行交叉运算,并给出计算结果与待匹配元素的相似度,最后对得出的相似矩阵进行相似度估计,获取最可能的一对多匹配关系.

与本文的方法相比,iMap 方法作用在实例上,并且进行了大量的搜索,时间复杂度较大,而且不能生成一些语义匹配关系,如“part-of”.

6.4 不足与改进

自动模式匹配的质量在很大程度上取决于可利用信息的多样性和完整性.除了本文所用的模式信息(属性名称和属性数据类型)、实例信息(数据库中的数据)以及领域知识(WordNet、领域本体)以外,完整性约束、用户反馈和已有映射关系等信息都具有一定的利用价值.例如,LSD 方法使用约束处理器,消除了一些违反约束的匹配结果,进一步提高了匹配质量.用户反馈也必定会提高匹配的质量,其关键是提供友好的工具方便用户交互,Clio[25]是一个辅助用户半自动化地构建映射的工具集,可使用户有效地参与其中.利用已有映射可以提高匹配效率,也能辅助发现一些新的匹配关系.文献[26]提出了已有映射的复合计算方法,是在该方面的积极探索.

7 结束语

本文从数据交换中间件的研发实践出发,提出了一种本体辅助的自动化模式匹配方法,可以在语法和语义层次上发现关系数据库模式中一对一和一对多的映射关系,从而提高数据交换的效率.首先形式化地定义了基于映射的数据交换系统框架,然后逐一地对属性名称、属性数据类型和一对多映射关系计算其相似度矩阵.在计算过程中,分别使用了 WordNet 词汇本体、数据类型本体和领域本体,采用了语法分析、语义分析和机器学习方法.对实际应用数据的实验结果表明,应用该方法可以获得较高的精确度和召回率.

在与相关工作的比较过程中也发现了一些不足之处,需要在信息利用的多样性方面进一步完善.开发方便用户交互的模式匹配工具,并集成在数据交换中间件中,也是我们今后进一步的工作.

References:

[1] Miller RJ, Ioannidis YE, Ramakrishnan R. Schema equivalence in heterogeneous systems: Bridging theory and practice. Information Systems, 1994,19(1):3−31.

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

[3] Shvaiko P, Euzenat J. A survey of schema-based matching approaches. Journal on Data Semantics IV, 2005,3730:146−171.

[4] Technology Center of Software Engineering, Institute of Software, the Chinese Academy of Sciences. OnceDI. 2006. http://www.once.org.cn

[5] Halevy AY. Answering queries using views: A survey. The VLDB Journal, 2001,10(4):270−294.

[6] Gruber T. A translation approach to portable ontology specifications. Knowledge Acquisition, 1993,5(2):199−220.

[7] Cognitive Science Laboratory, Princeton University. WordNet: A lexical database for the English language. 2006. http://wordnet.princeton.edu/

[8] Melnik S, Garcia-Molina H, Rahm E. Similarity flooding: A versatile graph matching algorithm. In: Proc. of the Int'l Conf. on Data Engineering (ICDE). IEEE Computer Society, 2002. 117−128.

[9] Castano S, de Antonellis V. Global viewing of heterogeneous data sources. IEEE Trans. on Knowledge and Data Engineering, 2001,13(2):277−297.

[10] Do HH, Rahm E. COMA—A system for flexible combination of schema matching approaches. In: Proc. of the 28th Int'l Conf. on

Very Large Data Bases (VLDB 2002). Hong Kong, 2002. 610−621.

[11] Giunchiglia F, Shvaiko P, Yatskevich M. Semantic matching: Algorithms and implementation. Technical Report, DIT-05-014, Trento: University of Trento, 2005.

[12] Madhavan J, Bernstein P, Rahm E. Generic schema matching with Cupid. In: Proc. of the 27th Int'l Conf. on Very Large Data Bases (VLDB 2001). Rome: Morgan Kaufmann Publishers, Inc., 2001. 49−58.

[13] The Gene Ontology Consortium. The gene ontology. 2004. http://www.geneontology.org/

[14] Mitchell TM, Wrote; Zeng HJ, Zhang YK, *et al*., Trans. Machine Learning. Beijing: China Machine Press, 2003 (in Chinese).

[15] Winston P. C4.5 tutorial. 1992. http://www2.cs.uregina.ca/~dbd/cs831/notes/ml/dtrees/c4.5/tutorial.html

[16] Biron PV, Permanente K, Malhotra A. XML schema part 2: Datatypes second edition. 2004. http://www.w3.org/TR/xmlschema-2/

[17] Kim W, Seo JY. Classifying schematic and data heterogeneity in multidatabase systems. IEEE Computer, 1991,24(12):12−18.

[18] Astrova I. Reverse engineering of relational databases to ontologies. In: Davies J, *et al*., eds. Proc. of the ESWS 2004. LNCS 3053, Berlin: Springer-Verlag, 2004. 327−341.

[19] Li M, Du XY, Wang S. A semi-automatic ontology acquisition method for the semantic Web. In: Fan WF, Wu ZH, Yang J, eds. Proc. of the WAIM 2005. LNCS 3739, Berlin: Springer-Verlag, 2005. 209−220.

[20] Liu Q, Huang T, Liu SH, Zhong H. An ontology-based approach for semantic conflict resolution in database integration. Journal of Computer Science and Technology, 2007,22(2):218−227.

[21] Dhamankar R, Lee Y, Doan AH, Halevy A, Domingos P. iMap: Discovering complex semantic matches between database schemas. In: Proc. of the SIGMOD 2004. Paris: ACM Press, 2004. 383−394.

[22] Doan A, Domingos P, Halevy AY. Reconciling schemas of disparate data sources: A machine learning approach. In: Proc. of the SIGMOD 2001. ACM Press, 2001. 509−520.

[23] Palopoli L, Sacca D, Ursino D. Semi-Automatic semantic discovery of properties from database schemas. In: Eaglestone B, Desai BC, Shao JH eds. Proc. of the Int'l Database Engineering and Application Symp. (IDEAS'98). Wales: IEEE Computer Society, 1998. 244−253.

[24] Huma Z, Rehman MJU, Iftikhar N. An ontology-based framework for semi-automatic schema integration. Journal of Computer Science and Technology, 2005,20(6):788−796.

[25] Miller R, Haas L, Hernandez MA. Schema mapping as query discovery. In: Abbadi AE, Brodie ML, Chakravarthy S, Dayal U, Kamel N, Schlageter G, Whang KY, eds. Proc. of the 26th Int'l Conf. on Very Large Data Bases (VLDB 2000). Cairo: Morgan Kaufmann Publishers, Inc., 2000. 77−88.

[26] Bernstein PA, Green TJ, Melnik S, Nash A. Implementing mapping composition. In: Dayal U, Whang KY, Lomet DB, Alonso G, Lohman GM, Kersten ML, Cha SK, Kim YK, eds. Proc. of the 32nd Int'l Conf. on Very Large Data Bases (VLDB 2006). Seoul: VLDB Endowment, 2006. 55−66.

附中文参考文献:

[14] Mitchell TM,著;曾华军,张银奎,等,译.机器学习.北京:机械工业出版社,2003.

刘强(1979－),男,河南卢氏人,博士,主要研究领域为数据库,知识库系统.

赵迪(1982－),男,硕士,主要研究领域为数据集成中间件.

钟华(1971－),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为网络分布式计算,软件工程技术.

黄涛(1965－),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为网络分布式计算,软件工程,方法学.