语义 Web 中的本体自动映射

唐 杰 梁邦勇 李涓子 王克宏
(清华大学计算机科学与技术系知识工程组 北京 100084)

摘 要 分布式语义信息集成是语义 Web 面临的六大挑战之一. 本体映射是语义集成的关键. 文章基于贝叶斯决策理论提出最小风险的本体映射模型: RiMOM (Risk Minimization based Ontology Mapping). RiMOM 将映射发现问题转换成风险最小化问题, 提供了一个多策略的本体映射方法. 该方法不仅在 1∶1 的映射上取得了较好的效果, 还实现了 n∶1 映射. 实验表明在几个公开的数据集上, RiMOM 可以取得比同类方法更高的查准率和查全率.

关键词 语义 Web; 本体映射; 最小风险决策; 本体集成
中图法分类号 TP18

Automatic Ontology Mapping in Semantic Web

TANG Jie LIANG Bang-Yong LI Juan-Zi WANG Ke-Hong
(*Knowledge Engineering Group, Department of Computer Science and Technology, Tsinghua University, Beijing* 100084)

Abstract Interoperability over distributed ontologies is one of the six challenges for Semantic Web. Ontology mapping is the key point to each the interoperability. In this paper, based on Bayesian decision theory, the authors propose an approach called RiMOM (Risk Minimization based Ontology Mapping) to automatically discover the mapping between ontologies. RiMOM treats the mapping problem as a decision problem and formalizes mapping discovery as that of risk minimization. Based on multiple strategies, RiMOM deals with not only 1∶1 mapping, but also n∶1 mapping. Experiments on several public data sets show that RiMOM can outperform existing methods in terms of precision and recall.

Keywords semantic Web; ontology mapping; risk minimization decision; ontology interoperability

1 引 言

语义 Web 使计算机能够‘理解’Web 信息, 实现计算机之间的智能交互[1]. 本体是语义 Web 的基础, 它作为一种领域知识概念化和模型化的方法, 可以用来描述计算机处理数据的语义信息. 目前本体已经成为语义 Web 中知识表示的标准.

为实现语义信息共享, 各个领域纷纷定义了相应的本体标准, 例如 Cyc 常识本体库[2]、企业信息本体库[3]以及生物化学本体库[4]等. 然而 Web 本身的分布性使得各个领域甚至同一领域的不同组织必然定义他们自己的本体来描述数据. 这时, 本体自身就是异构的, 实现 Web 信息交互的关键也就变成本体间映射的发现问题.

本体可能存在不同的异构问题: 两个本体中相同‘意义’的元素(元素表示本体中的概念、关系、属性以及实例)可能使用不同的名称; 相同名称也可能

收稿日期: 2004-06-18; 修改稿收到日期: 2006-07-23. 本课题得到国家自然科学基金(90604025)资助. 唐 杰, 男, 1977 年生, 博士, 研究方向包括语义内容标注、本体集成、机器学习、文本挖掘以及信息抽取. E-mail: j-tang02@mails.tsinghua.edu.cn. 梁邦勇, 男, 1978 年生, 博士, 研究方向包括领域数据管理、本体设计、文本处理以及信息检索. 李涓子, 女, 1964 年生, 博士, 副教授, 研究方向包括中文信息处理以及 Web 知识发现和管理. 王克宏, 男, 1941 年生, 教授, 博士生导师, 研究领域包括知识工程、分布式知识处理、网络计算和知识处理.

被用来表示不同‘意义’的元素; 另外相同领域的本体可能定义有不同的分类结构; 不同本体库中的相同实例可能使用不同的表示方法. 本体映射的研究则正是为了解决这些异构问题, 包括发现两个本体的元素之间的对应关系, 统一实例的不同表示形式等.

在分布式数据库及其它数据集成应用中也存在和本体映射类似的问题[5,6]. 其通常的方法是定义一个全局模式描述分布式环境下的所有数据, 这样数据集成问题就转换成本地数据库模式到全局数据库模式的映射问题. 然而基于本体的信息互操作和信息集成问题是一个更加动态的知识共享过程, 这种全局模式方法并不能完全适用.

已有许多关于本体映射的研究, 但到目前为止, 仍然存在几个急需解决的问题: 首先, 目前方法能够处理的映射类型非常有限, 大多数研究都集中于 1∶1 映射[7~14], 然而统计表明现实世界的异构数据大约有 22%~25% 的映射都不仅仅是这种 1∶1 映射[15,16]; 其次本体包含多种类型的信息, 但利用了所有这些可用信息的系统还比较少. 总的来说, 本体映射还需要一个更好的理论模型, 以综合利用所有的本体信息进行映射发现, 同时支持多种类型的映射发现.

本文将本体映射问题形式化为风险决策问题, 将最优映射的发现问题转换成风险最小化问题. 基于贝叶斯决策理论, 提出风险最小化的本体自动映射模型 RiMOM(Risk Minimization based Ontology Mapping)[17,18]. RiMOM 的处理流程主要包括候选映射选择、多策略的映射发现、多策略映射结果的合并以及映射发现机制. 映射发现过程迭代运行直到不能再发现新的映射为止. 在每一次迭代过程中, RiMOM 都支持用户交互以优化发现的映射. 基于本文方法, 我们实现了原型系统.

本文从不同的数据源收集了大量异构本体作为映射的实验数据, 包括来自于 5 个数据源的 28 个本体. 实验结果表明, RiMOM 明显优于基于单一信息的基线方法. 我们将 RiMOM 和同类方法 GLUE[15] 进行了比较, 在两个映射任务上, RiMOM 优于 GLUE; 在另外两个映射任务上, 两种方法的映射精度基本相当. 我们还和 2004 年度本体映射国际竞赛(EON 2004) 中的四种方法(即 Karlsruhe2, Umontreal, Fujitsu 及 Stanford) 进行了比较, 实验表明 RiMOM 明显优于 Karlsruhe2 和 Umontreal; 和 Fujitsu 及 Stanford 的映射精度基本相当.

本文第 2 节介绍相关术语, 分析本体异构性存在的问题, 并给出本体映射的形式化描述; 第 3 节介绍本文提出的模型 RiMOM, 概述其总的映射过程, 并分别阐述各个映射策略; 第 4 节描述映射发现算法, 并对算法复杂度进行分析; 第 5 节给出实验结果和实验分析; 第 6 节介绍相关研究; 最后总结全文.

2　术　语

本节介绍相关的定义和术语, 分析异构本体存在的问题, 并给出本体映射的形式化定义.

2.1　本　体

关于本体的形式化定义很多, 本文采用文献[19,20]提出的定义形式. 本体主要包括概念(concepts)、关系(relations)、实例(instances) 以及公理(axioms), 可表示为

$$O=\{C,\ R,\ I,\ A^O\} \tag{1}$$

其中, C 表示概念集合, c 表示概念($c\in C$); R 表示关系集合, r 表示关系($r\in R$); 符号 i 和 I($i\in I$) 分别表示实例和实例集合; A^O 表示公理的集合.

概念表示特定领域中的一组或一类实体或者‘事物’. 每个概念可以由属性分别描述其不同方面的特点.

关系描述了概念与概念之间或者属性与属性之间的关系. 关系可以分为两类: 分类关系(taxonomies) 和连接关系(associative relationships). 分类关系表示概念与概念之间的父类、子类等上下位的层次关系; 连接关系表示除了上下位层次关系以外的其它关系. 概念可以定义为层次状的分类体系, 在分类体系中各个概念通过分类关系联系在一起. 关系也可以组织成一个层次状的分类体系; 同样关系也可以用属性描述其不同方面的特点(例如: 关系的势(cardinality) 以及关系是否具有传递性(transitive) 等属性).

实例是概念所表示的‘事物’. 严格地说, 实例不应该包括在本体的定义中, 因为它是领域概念化的结果. 本体和它所表示的实例统称为知识库. 但是, 有时区分概念和实例是非常困难的, 不同应用领域往往具有不同的定义. 这也是语义 Web 研究中的一个公开问题[20].

公理用来表示概念或者实例的约束. 从某种意义上来说, 属性和关系可以看作公理的一种. 此外, 公理还包含更一般的规则, 例如: 一个课程应该至少有一位任课教师.

每一个概念、关系、属性或者实例都由本体专家定义其名称,通常表示为一个或者多个单词(实例也可以通过自动标注得到).本体工程师还可以为它们建立描述信息(例如:概念描述),描述信息通常是一段自然语言的文本或者多个关键词的集合.

为便于描述,本文用实体元素 $e(e_i \in C \cup R)$ 统一表示概念、关系以及属性.

2.2 本体异构性

为实现分布式本体集成,首先要解决两个异构问题:元数据异构和实例异构[21,22].元数据异构是指本体元数据定义的异构问题,包括结构冲突和名称冲突.相同本体可能包含不同的语义结构,这就是结构冲突.另外不同的名字可以用来表示'意义'相同的概念,相同名字在不同本体中又可以用于表示不同的概念,这就是名称冲突.

图1给出了存在着这两类元数据异构问题的本体.这是两个异构的课程本体.在本体 A 中,概念'faculty'具有和本体 B 中概念'Academic Staff'相同的意义,这就是命名冲突中的同义异名问题.同时也存在同名异义的情况,例如:在本体 A 中,'courses'包含了属性:'name'、'location'和'time',而在本体 B 中,'courses'只表示了'name'.另一方面,两个本体也包含结构冲突.例如本体 A 中的'student'有两个子类,而本体 B 中的'student'有三个子类.

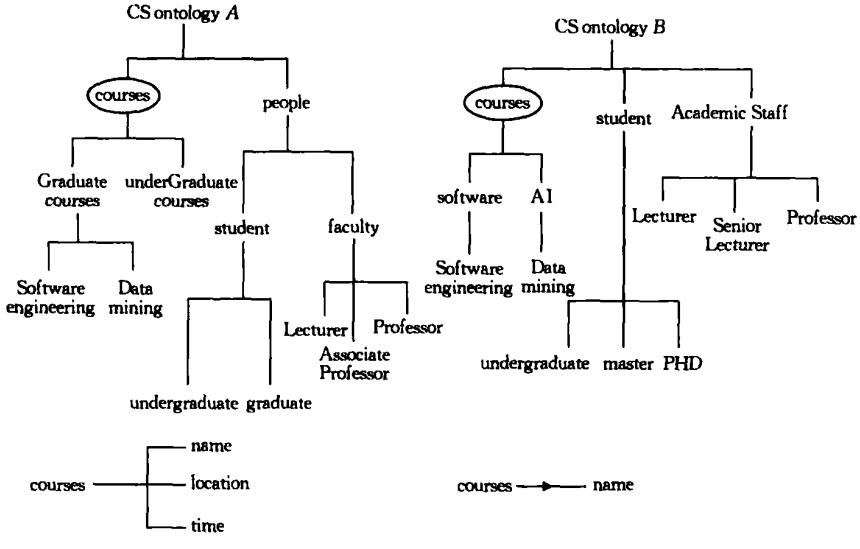


图1 两个课程的本体

实例异构主要考虑实例的表示问题,意义相同的实例可能有不同的表示形式,即实例冲突.例如:同一日期可以表示成"2004/2/27",也可以表示"Feb, 27, 2004",人名可以表示成"Jackson Michael"或者"Michael, Jackson",等等.实例异构问题使得在语义集成之前进行实例规范化成为必需.例如:Wiesman 等人就提出基于规则的归一化方法来解决实例冲突问题[23].本文将着重讨论元数据异构问题,而不讨论实例异构问题.

2.3 本体映射

本体映射算法以两个本体作为输入,然后为这两个本体中的各个元素(概念、属性或者关系)建立相应的语义关系[16].

本体映射领域的研究较多,各种定义、表达甚至研究目的都差别较大(例如:本体集成、本体合成等),这里首先给出本文关于本体映射的定义.我们将本体映射定义为一个有向的映射关系.对于 O_1 到 O_2 的映射,我们称本体 O_1 为源本体,O_2 为目标本体;称本体映射的发现过程为(本体)映射发现或者(本体)映射预测.

将本体映射函数的形式化定义为

$$Map(\{e_{i_1}\}, \{e_{i_2}\}, O_1, O_2) = f \qquad (2)$$

这里 $e_{i_1} \in O_1$, $e_{i_2} \in O_2$ 且 $\{e_{i_1}\} \xrightarrow{Map} \{e_{i_2}\}$. $\{e_{i_1}\}$ 和 $\{e_{i_2}\}$ 都表示元素集合(元素表示本体中的概念、属性及关系).元素集合可以包含一个元素、多个元素或者为空.当目标元素集合为空时表示本体 O_2 中没有源元素集合 $\{e_{i_1}\}$ 的映射对象.在不引起歧义的情况下将映射函数简写为 $Map(\{e_{i_1}\}) = \{e_{i_2}\}$.另外本文将本体 O_1 到 O_2 的所有映射表示为 $Map(O_1, O_2)$.

本体之间的映射存在六种主要的类型,包括 1∶1, 1∶n, n∶1, 1∶*null*, *null*∶1 及 n∶m.表1列出了这些映射类型.

表 1　两个本体间的映射类型

类型	O_1	O_2	映射表达式
1∶ 1	faculty	academic staff	O_1. faculty= O_2. academic staff
1∶ n	name	first name, last name	O_1. name= O_2. first name+ O_2. last name
n∶ 1	cost, tax ratio	price	O_1. cost× (1+ O_1. tax ratio)= O_2. price
1∶ *null*	AI		
null∶ 1		AI	
n∶ m	title, name	book, author	O_1. title+ O_1. name= O_2. book+ O_2. author

在这六种映射类型中, 已有方法主要研究 1∶ 1 的映射. 本文将重点研究 1∶ 1, n∶ 1, 1∶ *null* 和 *null*∶ 1 映射. n∶ m 映射的发现是一个非常复杂的问题, 本文将不作为重点考虑. 本文将 1∶ n 映射看作是 n∶ 1 映射的反类型, 我们通过双向的映射发现过程来实现 1∶ n 映射的发现, 即结合 O_1 到 O_2 的映射和 O_2 到 O_1 的映射来发现 1∶ n 映射. 这样在单独研究 O_1 到 O_2 映射的时候, 只考虑 1∶ 1, n∶ 1, *null*∶ 1和 1∶ *null* 四种映射类型.

对于两个本体, 当发现映射关系 $Map(\{e_{i_1}\})=\{e_{i_2}\}$时, 我们称之为"元素集合$\{e_{i_1}\}$映射到元素集合$\{e_{i_2}\}$". 对于每个可能的映射对$(\{e_{i_1}\},\{e_{i_2}\})$, 称其为候选映射. 在此假设每个元素最多只能存在于一个映射关系中.

3　本体映射模型

本节首先简单介绍贝叶斯决策理论, 然后介绍 RiMOM 如何基于贝叶斯决策理论实现映射发现, 接着介绍 RiMOM 的多策略映射发现过程, 最后详细阐述每个映射发现策略.

3.1　贝叶斯决策理论

贝叶斯决策理论为不确定性推理提供了坚实的理论基础[24]. 贝叶斯理论可以描述为: 设观察数据为随机分布的样本集合 X, 其中每个样本表示为 x. 令 $y\in Y$ 表示一个类别. 每个样本都可能被分类到某个类别中, 概率 $p(y|x)$ 表示样本 x 属于类别 y 的条件概率. 令 $A=\{a_1, a_2, \cdots, a_n\}$表示一组可能的决策行为(不同具体应用对决策行为的定义可能各不相同). 根据贝叶斯决策理论, 每一个决策行为 a_i 都有一个损失函数$L(a_i,x,y)$(例如: 损失函数可以定义为将样本 x 分到类别 y 的损失).

给定 Y 和A, 对样本 x 采取决策行为 a_i的风险定义为

$$R(a_i|x)=\int_y L(a_i,x,y)p(y|x)\mathrm{d}y \qquad (3)$$

贝叶斯决策问题的求解是要发现对每个样本的最优决策行为 a_i(即风险最小的行为), 进而发现整个决策问题的全局最优行为$\{a_i\}$. 每个样本的最优行为定义为

$$a^*=\arg_a \min R(a|x) \qquad (4)$$

分类可以看作贝叶斯决策理论的一个特殊应用. 这时决策行为 A 和分类类别 Y 表示的意义一致, 行为即表示将样本 x 分类到类别 y. 例如, 在贝叶斯分类中, 发现风险最小化的决策行为 a^* 等价于将样本分类到后验概率最大的类别中.

3.2　基于风险最小化的本体映射模型: RiMOM

基于贝叶斯决策理论, 本文将本体映射看作最优决策的发现问题, 提出基于风险最小化的本体映射模型 RiMOM(Risk Minimization based Ontology Mapping).

在本体映射问题中, 观察数据为本体 O_1 和 O_2 中的所有元素. 本体 O_1 中的元素$\{e_{i_1}\}$看作样本集合 X, 本体 O_2 中的元素$\{e_{i_2}\}$看作分类类别 Y. 每个元素 e_{i_1} 可以分类到某个类别 e_{i_2} 中, 即表示存在元素 e_{i_1} 到 e_{i_2} 的映射. 这里使用 $p(e_{i_2}|e_{i_1})$ 表示元素 e_{i_1} 映射到 e_{i_2} 的后验概率. 决策行为定义为所有可能的映射, 即所有的候选映射. 因此, 映射最优化问题就转换成决策行为最优化问题, 即搜索风险最小的决策行为.

令 $L(a_i, e_x, e_y, O_1, O_2)$ 表示损失函数, 对于 O_1 中的元素 e_x, 决策的贝叶斯风险定义为

$$R(a_i|e_x, O_1, O_2)=\int_{e_y} L(a_i, e_x, e_y, O_1, O_2)\,p(e_y|e_x, O_1, O_2)\,\mathrm{d}(e_y) \qquad (5)$$

其中, $e_x\in O_1$, $e_y\in O_2$, e_y 是e_x 在 O_2 中的候选映射. 后验概率公式 $p(e_y|e_x, O_1, O_2)$ 中包含了 O_1 和 O_2, 这表示在计算映射风险的时候不仅要考虑元素 e_x 和 e_y 本身的信息(局部信息), 还需要考虑本体 O_1 和 O_2 中的全局信息(如: 分类结构).

定义损失函数为

$$L(a_i, e_x, e_y, O_1, O_2)=\delta(e_x, e_y) \qquad (6)$$

其中, $\delta(e_x, e_y)$ 表示当元素 e_x 映射到e_y 的时候 $\delta(e_x, e_y)=-1$, 否则 $\delta(e_x, e_y)=1$. 从损失函数可以看出, 当后验概率 $p(e_y|e_x, O_1, O_2)$ 的值越大, e_x 映射到e_y

的风险就越小; 反之, 风险就越大.

最后, 基于贝叶斯决策理论, 将本体 O_1 映射到 O_2 的风险定义为

$$R=\int_{e_x} R(a_i \mid e_x, O_1, O_2)\,\mathrm{d}(e_x), \quad e_x \in O_1 \qquad (7)$$

求解风险最小化的本体映射定义为

$$R^* = \{a^*\} = \arg\min_{\{a_i\}} R(\{a_i\} \mid O_1, O_2) \qquad (8)$$

其意义表示发现具有风险最小化的决策行为集合.

3.3 映射发现过程

方程式(5) ~ (8)给出了利用决策理论求解本体映射的通用公式. 具体实现该公式的方法很多. 本节以后部分给出 RiMOM 的实现方法.

在映射发现过程中, 本体中的各种信息都可以用于映射发现, 例如: 实例、元素名称、元素描述、分类结构以及约束. 根据这些信息, 本文提出多决策的映射方法, 每个决策都可以独立地进行映射发现, 然后将多策略的映射结果进行合并, 最后通过映射发现算法选择最优决策.

图 2 给出 RiMOM 的本体映射流程. 输入是两个异构本体, RiMOM 的任务是建立源本体到目标本体的映射关系. 映射过程是一个迭代的过程, 每次迭代包括五个主要的步骤:

(1) 用户交互过程(可选过程). RiMOM 支持一个可选的用户交互过程, 通过用户交互, 用户可以在自动映射之前预先指定一个或多个映射关系, 也可以在映射自动发现之后纠正RiMOM发现的错误映射, 或者创建遗漏的映射关系. 用户的交互动作将传递影响其它相关联元素的映射, 从而对整个本体的映射产生影响, 以达到提高映射精度的目的.

(2) 多策略映射. 这是每次迭代中最重要的一个步骤, 在该步骤中, 每个策略分别计算候选映射的预测值(预测值归一化到[0, 1]). 这样使用 k 个策略, 在本体 O_1 和 O_2 分别包含 m 和 n 个元素的情况下, 多策略映射过程运行后得到一个 $k \times m \times n$ 的预测值结果, 这些预测值是下一步多策略合并的基础.

(3) 多策略合并. 将多策略方法的映射结果进行合并, 得到候选映射的综合预测值.

(4) 映射发现. 映射发现基于各个独立策略的映射预测值和合并后的综合预测值进行, 并考虑本体的约束和上下文关系选择映射关系. 已有的映射发现机制包括使用阈值的选择策略、最大预测值的选择策略[13]、松弛标注(relaxation labeling)的选择策略[15]以及结构相似度和内容相似度相结合的选择策略.

(5) 映射迭代. 整个映射过程是一个迭代的过程. 如果映射过程中有用户交互存在, 即用户修改过错误的映射或者创建了新的映射, 则算法重新评估已得到映射是否‘ 最优’ . 每次迭代过程都包括两个子阶段: 概念映射发现和属性映射发现. 迭代一直进行, 直到不能再发现新的映射为止.

最后, 算法输出映射表, 表中每一项对应一个映射关系. 每一项包含两个元素集合: 源本体 O_1 中的元素集合$\{e_{i_1}\}$和目标本体 O_2 中的元素集合$\{e_{i_2}\}$.

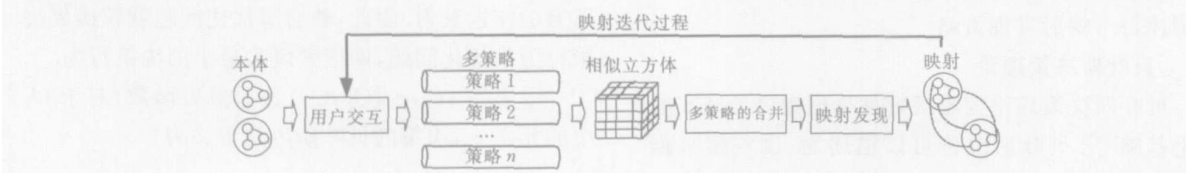


图 2 RiMOM 中的映射过程

3.4 RiMOM 中的映射策略

利用本体的各种信息: 元素名称、元素描述、实例、分类结构和约束, 本文分别设计相应的决策方法. 以下介绍各策略的具体实现方法.

(1) 基于标注实例的决策

使用元素名称发现映射关系可能是映射发现中最直接也是最基本的方法. 传统方法有的使用向量空间模型 VSM(Vector Space Model)计算名称相似性, 从而将问题转换成信息检索问题[25]; Bouquest 等人提出利用编辑距离(edit distance) 计算元素名称的相似度, 用以发现映射关系[22]; Doan 等人利用机器学习方法将问题转换成文本分类问题[15]. 然而, 这些方法还存在一些问题: 信息检索方法通常都会返回一些不可预料的结果(信息检索仅在单词完全或者部分相似的情况下有效) ; 编辑距离的方法将两个字串的相似度定义为将一个字串转换为另一个字串的最小操作次数(操作包括插入、删除和替换), 这种相似方法忽略了一个问题: 两个字串可能在拼写上完全不同, 但其意义却可能很相似; 基于机器学习的文本分类方法在长文本内容上效果较好, 在短文本内容上效果往往较差, 而元素的名称通常只包含一个或者几个单词.

本文结合概念词典和统计方法提出一种基于元素名称相似的映射决策方法, 两个词的相似度定义为

$$sim(w_1, w_2) = (sim_d(w_1, w_2) + sim_s(w_1, w_2))/2 \quad (9)$$

其中, $sim_d(w_1, w_2)$ 表示两个词 w_1 和 w_2 的概念相似度, 我们使用一个通用的语义词典 Wordnet 计算两个词的概念相似度; $sim_s(w_1, w_2)$ 表示利用统计方法得到的两个词的统计相似度. 下面介绍这两种相似度的实现方法.

首先介绍基于语义词典 Wordnet 的概念相似度计算方法.

Wordnet 是一个义类词典, 其中每个节点 s 表示一个词义, 节点中保存了多个同义词或者短语, 每个单词或短语又可以存在于多个语义节点中(即表明该单词有多个词义). 基于义类词典的概念相似度的基本思想是: 两个单词通过上位关系(hypernym)联接的距离越近, 它们的相似度越大; 反之, 它们的相似度越小. 如果它们在一个节点上, 即 $s_1 = s_2$, 则 $sim_d(w_1, w_2) = 1$, 如果它们在有限上位层次中没有共同的父节点(实验中取 10 层), 则 $sim_d(w_1, w_2) = 0$. Lin 等人根据 Wordnet 定义了两个词义的相似度[26]:

$$sim_d(s_1, s_2) = \frac{2 \times \log p(s)}{\log p(s_1) + \log p(s_2)} \quad (10)$$

其中 $p(s) = count(s)/total$ 表示在 Wordnet 中词义节点 s 及其子节点所包含的单词个数在整个词典中所占的比例, $total$ 是 Wordnet 的单词总数. 另外 $w_1 \in s_1$, $w_2 \in s_2$, 表示单词 w_1 和 w_2 分别位于节点 s_1 和 s_2 中. 节点 s 是 s_1 和 s_2 的公共祖先节点.

令 $s(w_1) = \{s_{1i} | i = 1, 2, \cdots, m\}$ 和 $s(w_2) = \{s_{2i} | i = 1, 2, \cdots, n\}$ 分别表示单词 w_1 和 w_2 的所有词义, 则两个单词的相似度定义为它们之间词义相似度的最大值:

$$sim_d(w_1, w_2) = \max_{s_{1i} \in s(w_1), s_{2j} \in s(w_2)}(sim_d(s_{1i}, s_{2j})) \quad (11)$$

以下介绍两个单词的统计相似度的实现方法.

Pantel 和 Lin 等人利用统计学习方法建立了统计相似词典[26]. 词典中, 两个单词的相似性定义为两个单词的上下文的相似度, 每个词可以有多组意义相近的单词(每一组代表一个词义), 每两个相似单词之间具有一定的相似值. 两个词的统计相似性 $sim_s(w_1, w_2)$ 通过直接查词典获得.

计算两个元素名称的相似性之前, 需要将两个名称进行分词, 如: 概念名称“Earth-and-Atmospheric-Sciences”经过分词可得到单词{Earth, and, Atmospheric, Sciences}. 然后计算两个元素名称所包含单词的相似矩阵. 矩阵中的值表示两个元素名称中某两个单词的相似值. 形式化的描述为: 输入两个元素名称 $name_1$ 和 $name_2$, 通过预处理得到两个单词集合$\{w_{1i} | i = 1, 2, \cdots\}$ 和$\{w_{2j} | j = 1, 2, \cdots\}$(以下简写为$\{w_{1i}\}$和$\{w_{2j}\}$); 然后对于 $name_1$ 中的每个单词 w_{1i}, 从$\{w_{2j}\}$中选择单词相似度 $sim(w_{1i}, w_{2j})$ 最大的作为单词 w_{1i} 和名称 $name_2$ 的相似度, 即 $sim(w_{1i}, name_2)$. 最后, 定义名称相似度为

$$sim(name_1, name_2) = \sum_{i=1,2,\cdots,n} sim(w_{1i}, name_2)/n \quad (12)$$

其中 n 是名称 $name_1$ 中单词的个数.

和传统方法相比, 我们采用的这种结合概念词典和统计技术计算元素名称相似度的方法不仅在元素名称完全或者部分相同的情况下有效, 而且在名称完全不同但存在一定语义联系的情况下也非常有效.

(2) 基于实例的决策

基于实例的决策方法利用文本分类技术实现本体映射, 其输入是两个本体的所有元素及其对应的实例.

在本体库中, 每个元素都有多个实例. 每个实例通常包含实例名称以及一组相关联的属性值. 本文将实例名称及其关联的属性值都看作该实例的文本内容, 另外还将和实例关联的文档也看作实例的文本内容, 例如在 Doan 的 Course 数据集中, 概念“Chinese”的实例“CHIN-101-102-Elementary-Standard-Chinese”的文本内容可以包括它所关联的网页. 利用每个实例的文本内容, 可以为每个实例建立一个‘文本’, 从而为每个元素建立了一个‘文本集’(每个元素可能有多个实例), 本体映射的问题也就可以转换成文本分类问题.

基于实例的决策方法使用实例对应的‘文本’中出现的单词及其频率信息来发现元素之间的映射关系. 对于给定的两个本体 O_1 和 O_2, 令它们分别包含的元素集合为$\{e_{i_1}\}$和$\{e_{i_2}\}$, $I_{i_1} = \{i_{1_k}\}$ 为元素 e_{i_1} 在本体 O_1 中的实例集合, $I_{i_2} = \{i_{2_k}\}$ 为元素 e_{i_2} 在本体 O_2 中的实例集合. 基于实例的决策方法将元素$\{e_{i_2}\}$看作分类类别, 将本体 O_2 中的实例看作训练样本, 将本体 O_1 中的实例看作测试样本. 映射的发现通过预测测试样本的类别来实现. 每个实例的文本内容

经过预处理(包括分词、停用词过滤、词干提取)生成单词的集合. 令 $i_{i_1 k}=\{w\}$ 表示经过预处理的实例,其中 w 表示一个单词,$\{w\}$ 表示单词集合.

实现文本分类的方法很多,我们使用简单贝叶斯 Naive Bayes(NB)分类器[27]. NB 从训练文本中学习分类模型,然后在分类的时候,给定测试样本(即实例 I_{i_1}),NB 通过选择最大后验概率 $\arg\max_{e_{i_2}} p(e_{i_2}\mid I_{i_1})$ 预测实例的分类类别(分类类别对应 O_2 中的元素 e_{i_2}). 后验概率 $p(e_{i_2}\mid I_{i_1})$ 定义为

$$p(e_{i_2}\mid I_{i_1})=p(I_{i_1}\mid e_{i_2})p(e_{i_2})/p(I_{i_1}) \quad (13)$$

其中,$p(I_{i_1})$ 是归一化常数,可以忽略,$p(e_{i_2})$ 是元素 e_{i_2} 的实例数目在所有实例中所占的比例. 为计算 $p(I_{i_1}\mid e_{i_2})$,我们假设对于给定类别 e_{i_2},其实例中出现的单词之间的分布相互独立,则有 $p(I_{i_1}\mid e_{i_2})=\prod_{w\in I_{i_1}} p(w\mid e_{i_1})$. 将其代入上式,可得

$$p(e_{i_2}\mid I_{i_1})=\prod_{w\in I_{i_1}} p(w\mid e_{i_2})p(e_{i_2}) \quad (14)$$

其中 $p(w\mid e_{i_2})=n(w,e_{i_2})/n(e_{i_2})$. $n(e_{i_2})$ 是 e_{i_2} 的所有实例中出现的单词总数,$n(w,e_{i_2})$ 是单词 w 在 e_{i_2} 的实例中出现的次数.

对于 e_{i_1} 的所有可能的候选映射,基于实例的决策方法计算该映射的预测值 $p(e_{i_2}\mid I_{i_1})$,然后将预测值最大的候选映射作为 e_{i_1} 的映射.

实验表明当实例的‘文本’较长的情况,基于实例的决策方法效果很好;当实例的‘文本’较短,或者单词分布过于稀疏的时候效果较差.

(3) 基于元素描述信息的决策

元素描述信息是指以自然语言形式存在的对元素的文本描述信息,它也是映射发现的一种重要信息.

本文使用文本分类技术实现基于元素描述的本体映射策略. 每个元素的描述信息可以看作一个‘文本’,这样基于元素描述的映射发现可以转换成文本分类问题. 针对基于元素描述的决策方法的具体实现,我们使用简单贝叶斯 Naive Bayes(NB)分类器. 将目标本体中的元素描述看作训练文本来建立分类器,然后将源本体中的元素描述看作测试样本用以预测元素之间的映射关系. 具体原理与基于实例的决策方法相似.

(4) 结构上下文决策

结构上下文决策利用元素在本体中的上下文信息进行映射发现. 结构上下文决策的思想主要基于:如果两个给定元素有相同/相似的上下文结构,则这两个元素可能存在映射关系. 例如:两个元素的子类和父类都分别存在映射关系,那么这两个元素也往往存在映射关系.

概念的结构上下文信息包括它的父类、子类、属性和关系等;属性的结构上下文包括它所属于的概念、属性的父类、子类、兄弟属性和关系约束等. 两个元素的上下文相似性定义为它们对应的上下文元素之间的相似度之和,而上下文元素之间的相似度则通过其它决策方法计算得到. 目前实现的程序中仅使用直接关联的元素作为上下文元素.

(5) 基于约束的决策

在本体中,约束主要用来定义数据类型和关系,例如值域、唯一性、可选性以及关系类型等,这些信息都为映射发现提供了有用信息.

本文利用约束定义启发式规则优化映射发现的结果. 下面是一些启发式规则的例子:

数据类型为日期的元素只能映射到数据类型为日期的元素上(可信度为 1.0).

数据类型为 float 的元素可以映射到数据类型为 string 的元素上(可信度为 0.6).

对于两个概念的属性映射,如果属性的势(cardinality)各不相同,则这两个概念可能存在映射关系(可信度为 0.7). 同理,对于“minCardinality”和“maxCardinality”本文也定义了类似的规则.

属性关系个数相同的两个概念可能有映射关系(可信度为 0.3).

以上的每个规则都对应一个可信度(如为 1.0,0.7)用于扩展传统的布尔规则(即满足或者不满足),可信度的值通过手工设定. 根据本体的约束和领域知识,本文一共定义了 12 个类似的规则.

(6) 多决策的合成

子决策的预测结果需要进行合并,已有的合并方法很多,其中最流行的主要有两种:hybrid 和 composite 方法[7,8]. Hybrid 方法是将多个算法合并到一个算法中,而 composite 方法是对多个算法的结果进行合并. 本文使用 composite 方法,通过如下公式合并多个策略的映射结果:

$$Map(e_{i_1},e_{i_2})=\sum_{k=1,2,\cdots,n} w_k\sigma(Map_k(e_{i_1},e_{i_2}))/\sum_{k=1,2,\cdots,n} w_k \quad (15)$$

其中 w_k 是各策略的权重,σ 是 sigmoid 函数. sigmoid 函数是一个平滑函数($[0,1]\rightarrow[0,1]$),它使得合并结果偏向于预测值高的策略. 函数 σ 定义为

$$\sigma(x)=1/(1+e^{-5(x-\alpha)}) \quad (16)$$

其中 x 是策略的预测值,α 是 sigmoid 函数中心点.

本文实验设为 0.5.

3.5　映射风险的计算

3.4 节介绍了多策略的本体映射方法, 并介绍了利用 composite 方法进行多策略的合并. 每个候选映射的合并结果都是[0, 1] 之间的实数值, 我们利用合并后的预测值定义元素 e_{i_1} 的映射风险为

$$R(a_j \mid e_{i_1}, O_1, O_2) = \int_{e_{i_2}} \delta(e_{i_1}, e_{i_2}) Map(e_{i_1}, e_{i_2}) \mathrm{d}(e_{i_2}) \tag{17}$$

其中, e_{i_2} 是 e_{i_1} 的候选映射对象, 决策行为 a_j 表示将 e_{i_1} 映射到 e_{i_2}. 直观上看, 求解式(17) 最小化的时候倾向合并预测值大的候选映射, 因为合并预测值越大, 其对应决策行为 a_j 的贝叶斯风险值就越小, 即 $Map(e_{i_1}, e_{i_2})$ (这时 $\delta(e_{i_1}, e_{i_2}) = -1$).

本体 O_1 到 O_2 的映射风险为

$$R = \int_{e_{i_1}} R(a_j \mid e_{i_1}, O_1, O_2) \mathrm{d}(e_{i_1}) \tag{18}$$

其中, $e_{i_1} \in O_1$. 在求解本体 O_1 到 O_2 的映射风险最小化的时候, 不仅需要考虑单个元素 e_{i_1} 的映射风险最小化问题, 还需要考虑其它元素的映射风险问题. 这是因为: 不同映射的生成会对其它元素的映射选择产生不同的影响; 在第 2 节我们假设每个元素最多只能存在于一个映射关系中, 因此当确定 e_{i_1} 到 e_{i_2} 的 1∶1 映射后, 也表明 e_{i_1} 不能再映射到其它元素上, 同时也表明不能再有其它元素映射到 e_{i_2} 上. 求解式(18) 的搜索策略如下: 首先通过式(17) 为本体 O_1 中的每个元素选择风险最小的三个候选映射; 然后将这三个候选映射带入到式(18) 中, 搜索使得映射风险 R 最小的映射结果.

4　映射发现算法

本节介绍 RiMOM 的映射发现算法, 主要介绍本体映射中的预处理和映射发现算法, 然后对算法复杂度进行分析. 在本文的原型系统中, 映射结果通过 XML 语言表示(4.2 节将给一个映射结果的例子). 映射表示的详细介绍可见文献[28, 29].

4.1　预处理

在映射发现之前, 概念、属性以及实例都需要进行预处理. 预处理包括分词、停用词删除、词干抽取、词性标注、命名实体识别以及归一化处理等. 本文使用 GATE[30] 进行预处理.

相同实例可能有不同的表达方式, 即前面介绍的表示冲突. 在自然语言处理领域, Sproat 等人研究了自然语言文本处理的单词归一化问题[31], 他们对非标准单词进行了分类, 并应用多元语言模型、决策树、带权重的有限状态机等方法来实现归一化. 但在本体中, 实例的文本内容不一定表示为自然语言, 可能仅仅是多个单词的集合, 因此基于多元语言模型的方法不一定能取得很好的效果.

本文通过定义有指导的规则实现实例的归一化. 首先使用 GATE 识别命名实体(包括时间、日期、年、百分数、货币、人名等), 然后对这些命名实体进行归一化. 例如: 日期转换成统一格式"年-月-日", 这样"2004-3-1" 和"March 1, 2004" 都统一转换成"2004 March 1"; 对于人名, 统一转换成"名+ 姓" 的格式, 这样"Jackson Michael" 和"Michael, Jackson"统一转换成"Jackson Michael"; 同时也使用规则合并人名"J. Michael" 和"Jackson Michael". 对于其它类型的命名实体, 我们也定义了相应的归一化规则.

就目前而言, 本文提出的这种基于规则的归一化方法是可行的, 定义的规则也比较有效. 通过对实验收集的 28 个本体进行统计发现超过 85.5% 的实例表示冲突问题都来自于时间、日期、年、百分数、货币以及人名等命名实体.

预处理阶段的另一个任务是要对元素名称进行预处理(包括分词和缩写词扩展等), 这种预处理可以提高元素名称相似度计算的精度. 例如: 对于概念名称"company_information", 通过分词可以得到{company, information}. 对于关系"hasEmployee", 通过分词得到{has, Empoyee}. 另外本文也通过用户预定义词典进行缩写词扩展, 如将"CS" 扩展为{computer, science}.

4.2　映射发现

映射发现过程主要包括四个子过程: 基于各个子策略的元素映射、映射合并、映射发现以及映射优化. 整个过程首先通过各个映射子策略分别独立地进行概念映射和属性映射的预测; 然后通过合并方法合并各子策略的预测结果; 接着利用映射发现算法发现各种不同类型的映射关系(包括 1∶1, n∶1, 1∶*null* 和 *null*∶1 映射); 最后对发现的映射进行优化.

映射子策略以及合并算法已经在第 3 节中给予了详细介绍, 这一节将主要介绍映射发现算法和优化算法.

(1) 映射发现算法

在映射发现过程中, RiMOM 计算每一个候选

映射的贝叶斯损失,然后在所有可能的映射空间中搜索风险最小化的映射结果. 映射发现的算法参见图 3. 函数 *preprocess*( )是预处理过程(见 4.1 节); 函数 *NamePrediction*( ), *InstancebasedPrediction*( ), *DescriptionPrediction* ( ), *TaxonomyContextPrediction*( )和 *ConstraintDecision*( )分别对应 6.3.4 节中介绍的五个映射子决策. *DecisionCombination*( )是决策合并函数. 对于每个概念, *PreConceptDecision*( )首先输出风险最小的几个候选映射(前 3 个), 然后对于所有概念, *ConceptMappingDecision*( )在 *PreConceptDecision*( )输出的候选映射中选择最优化的概念映射. *PruneConceptMapping*( )利用约束规则和领域知识对得到的映射进行剪枝, 同时发现可能的 1∶ *null* 映射. 对每个概念映射,算法根据同样的原理搜索属性映射. 最后进行映射优化,算法结合概念映射和属性映射, 利用领域知识对得到的映射进行剪枝, 输出映射结果.

```
输入:onto1, onto2
输出:Mapping table
算法:
//预处理
preprocess(onto1);
preprocess(onto2);
//概念映射发现
foreach(concept_i in onto1)
  foreach(concept_j in onto2)
    //compute all sub-decisions
    NamePrediction(concept_i,concept_j);
    InstancebasedPrediction(concept_i,concept_j);
    DescriptionPrediction(concept_i,concept_j);
    TaxonomyContextPrediction(concept_i,concept_j);
    ConstraintDecision(concept_i,concept_j);
    DecisionCombination(concept_i);
  PreConceptDecision();
ConceptMappingDecision();
//概念映射剪枝
PruneConceptMapping();

//属性映射发现
foreach(property_i in onto1)
  foreach(property_j in onto2)
    //compute all sub-decisions
    NamePrediction(property_i,property_j);
    InstancebasedPrediction(property_i,property_j);
    DescriptionPrediction(property_i,property_j);
    TaxonomyContextPrediction(property_i,property_j);
    ConstraintDecision(property_i,property_j);
    DecisionCombination(property_i);
  PrePropertyDecision();
PropertyMappingDecision();
//属性映射剪枝
PrunePropertyMapping();
//映射优化
MappingRefinement();
```

图 3 映射发现算法

本体映射还需要考虑其它的映射类型,例如: 由于概念、属性和实例并不一定互相独立,因此映射还包括概念到实例、实例到概念、属性到概念等映射. 但因为针对目前实验数据的统计发现, 这些类型的映射还很少,因此本文研究将主要考虑概念到概念和属性到属性的映射.

1∶1 映射是最简单也是最常用的映射类型. 1∶1映射的发现可以通过为 O_1 中的每个元素选择风险最小的映射来实现,然后全局考虑所有的元素, 发现任务则是选择所有元素的最优化映射. 以下介绍 *n*∶1, 1∶*null* 和 *null*∶1 的映射发现.

①*n*∶1 映射

当 O_1 中有多个元素映射到 O_2 中的同一元素时,这时就可能存在一个 *n*∶1 映射. *n*∶1 映射的发现包括两个阶段: 映射发现以及映射表示发现. 映射发现的任务是发现是否存在多个源元素映射到一个目标元素上; 映射表示发现过程的任务是搜索映射的表示函数(即源元素如何组合才能映射到目标元素上). 例如: 如果源元素是 firstname 和 lastname, 目标元素是 personname, 那么这个表示函数可以是 concat(firstname, lastname) (也可以写作 firstname+lastname).

在为源本体中的每个元素都选择了风险最小的几个候选映射后, RiMOM 搜索整个映射空间, 看是否存在多个源元素映射到一个目标元素的情况. 如果存在, RiMOM 触发一个组合过程, 该过程搜索表示函数.

以下用一个实例来解释映射表示函数的搜索过程. 例如: 当 RiMOM 发现概念 address, zipcode, telephone 同时映射到概念 contract_infomation(ci) 上的时候, 则触发一个名为 *Expression_Finding* 的过程, 该过程定义为

$$F(f(e_{\text{address}}), f(e_{\text{zipcode}}), f(e_{\text{telephone}})) = f(e_{\text{ci}}) \tag{19}$$

其中 $f(e)$ 是 e 的函数, 它表示在组合过程中可能存在 e 的函数表达式, 例如: left(*e*, *length*), lowercase(*e*) 等. $F(.)$ 是输入参数的组合函数. 目前实现的算法中仅考虑当 e 为字符串的情况. 函数 f 包括 6 个基本的字串处理函数: left, right, mid, lowercase, uppercase 以及 capitalize. 函数 F 主要指的是字符串连接等函数.

图 4 给出一个 *n*∶1 映射的输出结果. 该映射通过基于实例的决策方法发现得到, 其中源元素包括概念 address, zipcode 和 telephone; 目标元素为概念 contract_infomation; 每个概念都有一个唯一标志号(如: # addr). 映射表达式为“upcase(# addr)

+ # zip+ # tele= # ci", 其意义是 address 的大写形式加上 zipcode 和 telephone 映射到目标元素 contract_information 上. 每一个映射都有预测值(如: score= "0.5931"), 预测值最大的三个映射作为候选映射参与最终的映射发现.

```
〈mappings strategy="instance based decisioin"〉
 〈conceptmapping score = "0.5931" mappingtype = "equivalence"〉
  〈source〉
   〈concept id="#addr"〉address〈/concept〉
   〈concept id="#zip"〉zipcode〈/concept〉
   〈concept id="#tele"〉telephone〈/concept〉
  〈/source〉
  〈target〉
   〈concept id="#ci"〉contract_infomation〈/concept〉
  〈/target〉
  〈expression〉upcase(#addr)+#zip+#tele=#ci〈/expression〉
  〈candidate score="0.0541" type="equivalence"〉
   〈source〉
    〈concept id="#addr"〉address〈/concept〉
    〈concept id="#zip"〉zipcode〈/concept〉
   〈/source〉
   〈target〉
    〈concept id="#ci"〉contract_infomation〈/concept〉
   〈/target〉
  〈expression〉#addr+#zip=#ci〈/expression〉
  〈/candidate〉
  …
 〈/conceptmapping〉
 …
〈/mappings〉
```

图 4　一个 n: 1 映射的输出

②1: *null* 映射

1: *null* 映射是一个比较特别的映射, 我们通过启发式规则进行 1: *null* 映射的发现, 表 2 列出一些规则的实例.

③*null*: 1 映射

当没有任何元素映射到 e_{i_2} 的时候, 称对于元素 e_{i_2} 有 *null*: 1 映射.

(2) 映射优化

本文利用规则对发现的映射进行优化. 映射优化主要包括: 删除那些映射预测值很高但' 不合理' 的映射关系; 同时提高那些预测值较低但' 合理' 的映射关系的预测值. 我们通过以下几种情况来解释映射优化过程.

例 1. 当概念 e_{i_1} 映射到概念 e_{i_2} 时, 如果概念 e_{i_1} 的父概念 $e_{i_1}^p$ 和子概念 $e_{i_1}^s$ 都映射到 e_{i_2} 的父概念 $e_{i_2}^p$ 上, 则这三个映射存在矛盾, 其中至少存在一个映射错误, 这时定义概念 $e_{i_1}^s$ 到 $e_{i_2}^p$ 的映射为错误映射.

例 2. 对于概念 e_{i_1}, 如果它的父概念 $e_{i_1}^p$ 和子概念 $e_{i_1}^s$ 分别映射到和 e_{i_2} 的父概念 $e_{i_2}^p$ 和子概念 $e_{i_2}^s$ 上, 但 e_{i_1} 的风险值最小的映射对象并不是 e_{i_2} (即 e_{i_1} 存在映射风险更小的目标元素 e_{k_2}), 这时如果到元素 e_{k_2} 和到 e_{i_2} 的映射风险之差小于给定阈值, 则将 e_{i_1} 的' 最优' 映射从概念 e_{k_2} 改为 e_{i_2}.

例 3. 属性映射结果可以用来优化概念映射. 对于每一个概念映射(如: e_{i_1} 到 e_{i_2}), RiMOM 都验证它们的属性映射. 这个想法的基本观点就是: 对于映射概念, 如果它们的属性之间不存在映射关系, 则对这样的概念映射进行惩罚. RiMOM 统计每个概念映射中的属性映射比例, 然后将这个概率乘以概念映射的预测值, 将乘积的结果作为新的预测值, 然后重新对所有候选映射进行排序, 选择最' 优' 映射.

例 4. 概念映射的结果也可以用来优化属性映射. 对于每个属性映射来说, RiMOM 都检查它们的"domain" 和"range" 对应元素的映射情况. 当两个属性的"domain" 都是概念的时候(通常情况下是这样), RiMOM 检查这两个概念之间是否存在映射. RiMOM 同时检查属性的"range" 的类型是否相同(例如: 是否都是数据类型 data type 或者是否都是对象类型 object type). 对于数据类型, RiMOM 检查它们的数据类型是否相同(如: string, integer 等); 对于对象类型, RiMOM 检查其对象是否都是概念, 如果是, 则进一步检查这些概念是否存在映射. 然后, RiMOM 计算一个惩罚因子, 用该因子乘以属性映射的预测值, 将结果作为新的预测值. 最后重新对所有候选映射排序, 选择最' 优' 映射.

表 2　部分 1: ***null*** 映射发现的启发式规则

规则分类	规则实例
Threshold	对于语义元素 e_{i_1}, 如果它的所有候选映射的预测值都小于给定阈值 μ, 则判定语义元素 e_{i_1} 存在 1: *null* 映射. 实验取 μ 为 0.2
Taxonomy	对于语义元素 e_{i_1}, 如果其父类元素 $e_{i_1}^p$ 和子类元素 $e_{i_1}^s$ 都有映射到本体 O_2 的目标元素 $e_{i_2}^p$ 和 $e_{i_2}^s$, 但 e_{i_1} 本身没有到 O_2 的映射关系, 并且 O_2 中元素 $e_{i_2}^p$ 和 $e_{i_2}^s$ 之间不存在语义元素(即 $e_{i_2}^p$ 直接是 $e_{i_2}^s$ 的父类), 则判定语义元素 e_{i_1} 存在 1: *null* 映射. 如图 5(a) 中的实例, 对于概念"car", 它的父类"transport" 和子类"cab" 和"police car" 分别映射到 O_2 中的概念"vehicle" 和"prowl car", 但 O_2 中没有定义父概念"vehicle" 和子概念"prowl car" 之间的概念, 因此判定概念"car" 存在 1: *null* 映射
	对于语义元素 e_{i_1}, 设其在 O_2 中有映射对象 e_{i_2}, 如果 e_{i_1} 的子概念比 e_{i_2} 的子概念数目多, 则 e_{i_1} 的子概念中可能存在 1: *null* 映射. 如图 5(b) 中的实例, 概念"Asian languages" 映射到概念"Asian studies", 然而概念"Asian languages" 有四个子概念, 概念"Asian studies" 却只有三个子概念, 则判定"Asian languages" 的某个子概念可能存在 1: *null* 映射. 我们判定风险值最低且没有超过阈值的元素(这里即元素"CHIN") 存在 1: *null* 映射

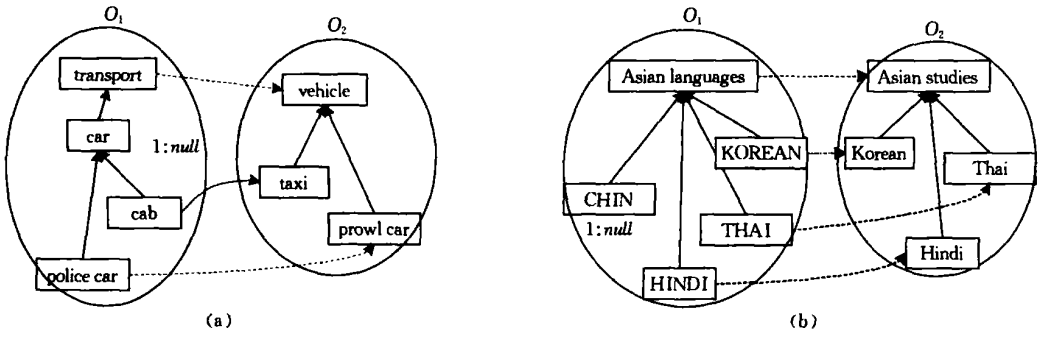


图 5 1: *null* 映射的实例

4.3 算法复杂度分析

算法复杂度是评估本体映射的一个重要指标,但目前在这方面的研究还不多[32].映射的复杂度主要取决于本体的规模和所使用策略的复杂度.本节就文中介绍的各个映射策略分别分析其算法复杂度,然后评估 RiMOM 总的算法复杂度. RiMOM 用 Java 语言实现,实验测试在一台 CPU 为 2.5GHz,内存为 1.5GB 的 PⅣ个人电脑上进行.

设本体 O_1 有 n 个元素,O_2 有 m 个元素.在本体子树上从根节点到当前叶子节点的搜索复杂度为 $O(\log(n))$.搜索本体中所有元素的复杂性为 $O(n)$.

基于实例的决策和基于元素描述的决策都使用贝叶斯分类方法,该分类器的时间开销和文本内容所包含单词的稀疏程度以及文本内容的长度有关,但对每对元素都要计算其映射的后验概率,因此可以认为其复杂度为 $O(mn)$.

对于命名决策来说,使用 Wordnet 计算每两个单词相似度的平均时间是 0.023s.根据对 Doan 提供的测试集(详见 5.1 节)进行统计显示,每个元素名称的平均长度为 1.97≈2,因此两个元素名称相似度的平均计算时间是 0.079s.对于 O_1 和 O_2 的所有元素,RiMOM 计算它们的相似度矩阵,因此总的复杂度为 $O(0.079mn)$.

结构上下文决策检查 O_1 中每个元素和 O_2 中每个元素上下文的相似度,算法复杂度为 $O(mn)$.

基于约束的决策复杂度和生成的候选映射个数成正比,而生成的候选映射个数又和本体 O_1 中的元素个数 n 成正比,因此其复杂度为 $O(n)$.

多策略的合成是合并所有策略生成的相似预测结果($k\times m\times n$),k 为常数,因此该算法的复杂度为 $O(kmn)$.

最后算法是一个迭代的过程,设迭代次数为 r.

综合上面各个策略的时间开销以及可能的迭代次数 r,最后 RiMOM 总的算法复杂度 c 为:

$$c = (O(mn) + O(mn) + O(0.079mn) + O(mn) + O(n) + O(kmn)) \times r \quad (20)$$

当两个本体的元素个数比较接近,并且 $n \gg 1$ 的时候,有

$$c = (O(n^2) + O(n^2) + O(0.079n^2) + O(n^2) + O(n) + O(kn^2)) \times r = O(n^2) \quad (21)$$

算法的实际时间开销和具体的本体有关,不同本体上的时间开销可能相差很大.总的来说,目前 RiMOM 的算法复杂性还比较高,一个主要原因就是每个策略首先都将所有可能的映射作为候选映射进行评估($O(n^2)$).Ehrig 等人提出快速本体映射 QOM 的思想,其主要观点是首先对候选映射的预测值进行排序,选择最可能的候选进行评估,这样可将算法复杂度降到 $O(n\times\log(n))$[32].如何降低 RiMOM 的算法复杂性也是本文今后需要开展的工作之一.

5 实验和评估

5.1 实验设计

(1) 实验数据

本文在五个测试集上做了实验.

Course Catalog ontology Ⅰ.该数据集中的本体分别描述了康奈尔大学和华盛顿大学的课程信息,两个本体的概念和属性的名称定义比较相似.

Company Profile.该数据集中的本体分别描述了 Yahoo.com 和 The Standard.com 公司的商务信息.

Employee Ontology.该数据集中的本体分别描述了某公司的雇员信息.

Sales Ontology.该数据集中的本体分别描述了销售信息.

EON.此数据集包含 19 个本体,这些本体都描述了书籍参考目录的信息.

前两个数据集,即 Course Catalog Ⅰ和 Compa-

ny Profiles 是 Doan 等人设计的测试集(http://anhai.cs.uiuc.edu/archive/summary.type.html); EON 数据集用于 2004 年度国际本体映射竞赛的评估(http://co4.inrialpes.fr/align/Contest/); 本文还用真实数据建立了两个数据集: Employee Ontology 和 Sales Ontology, 每个数据集都定义有两个异构本体.

除了 EON 外, 其它四个数据集都分别包含有两个异构本体, 因此映射任务是发现两个本体相互之间的映射关系. 在 2004 年度国际本体映射竞赛中, 一共使用了 26 个本体进行评估, 其中一个本体是目标本体(也称为参考本体 Reference Ontology), 映射任务是为其它 25 个本体建立到目标本体的映射关系. 最终只评测了 19 个源本体的映射结果. 另外还有一个本体, 本文在实验中碰到了文件解析错误, 因此最终我们在 EON 测试集中保留了 18 个源本体和一个目标本体.

Course Catalog Ⅰ中两个本体的元素名称定义比较相似(包括拼写相似和意义相似), 而 Company Profiles 中两个本体的元素名称定义差别较大. 这两个数据集用来测试基于元素名称的映射策略的效果. Employee Ontology 中两个本体的相似实例较少, 而 Sales Ontology 中两个本体有一定比例的相似实例. 这两个数据集用来测试基于实例的映射策略的性能. EON 包含有 19 个本体和 18 个映射任务, 可以用来测试不同情况下的映射性能, 详见 http://km.aifb.uni-karlsruhe.de/ws/eon2004/.

我们手工建立了 Employee Ontology 和 Sale Ontology 中的映射关系, 用它们作为评测标准. Course Catalog Ⅰ, Company Profiles 以及 EON 数据集中都包含有人工建立的'标准'映射关系.

表 3 给出对这五个测试集的统计数据. Course Catalog Ⅰ, Company Profiles 以及 EON 都只定义了 1∶1 映射, 因此对这三个数据集的评估将只考虑 1∶1 映射.

表 3　测试集的统计数据　(单位: 个)

数据集	本体	概念	属性	映射	实例
Course Catalog Ⅰ	Cornell	34	0	34	1526
	Washington	39	0	37	1912
Company Profiles	Standard.com	333	0	236	13634
	Yahoo.com	115	0	104	9504
Employee Ontology	Ontology 1	51	218	47	5000
	Ontology 2	45	186	45	5000
Sales Ontology	Ontology 1	44	126	44	3000
	Ontology 2	59	163	52	3000

(续　表)

数据集	本体	概念	属性	映射	实例
EON	Reference Onto	33	59	–	76
	101	33	61	91	111
	103	33	61	91	111
	104	33	61	91	111
	201	34	62	91	111
	202	34	62	91	111
	204	33	61	91	111
	205	34	61	91	111
	221	34	61	91	111
	222	29	61	91	111
	223	68	61	91	111
	224	33	59	91	0
	225	33	61	91	111
	228	33	0	33	55
	230	25	54	75	83
	301	15	40	61	0
	302	15	31	48	0
	303	54	72	49	0
	304	39	49	76	0

(2) 评估方法

实验结果使用查准率和查全率进行评估, 这些评估方法定义如下:

$$Precision:\ P = A/(A+B),$$

$$Recall:\ R = A/(A+C).$$

其中 A, B, C 和 D 表示相应样本的个数, 其不同含义见表 4.

表 4　映射目标和映射结果的交叉表

	Is Target	Is Not Target
Found	A	B
Non Found	C	D

同理, 对于本体映射, 表中"Found"和"Non Found"分别表示算法自动识别到的和未识别到的映射结果. "Is Target"和"Is Not Target"分别表示手工标注和手工未标注的映射结果. 因此 A 表示算法识别得到的正确映射结果, B 表示算法识别得到的错误映射结果, C 表示算法没有识别到的正确映射结果, D 表示算法未识别的错误映射结果.

查准率 Precision 和查全率 Recall 都通过将人工标注结果和算法自动识别结果进行比较得到.

在和其它方法进行比较的时候, 我们也使用 Sign Test 进行评估. 在给定数据集上对两个方法进行比较的时候, Sign Test 表示在测试数据上方法 1 较方法 2 的优势是否具有统计意义上的差异(反之亦然). Sign Test 的值越大(如> 0.5)表示方法 1 明显比方法 2 差, 值越小(如< 0.01)表示方法 1 明显优于方法 2(即这种优势具有统计意义)[33]. 我们将对本文提出的方法和基线方法以及其它方法都进行

Sign Test, 以便验证本文方法相对于其它方法是否具有统计意义上的优势.

对于 1∶ n 和 n∶ 1 映射的评测则进一步评测发现映射和手工映射之间能够匹配上的比例.

(3) 实验设计

实验用基于元素名称的策略和基于实例的策略作为基线方法, 用于验证 RiMOM 的映射性能. 同时实验还评估用户交互的效果, 用户交互表现为映射初始点的设置, 即在映射发现之前, 用户手工指定几个初始映射关系(2~ 5 个). 因此, 在每个数据集上, 都进行四个映射方法的评估:

基于元素名称决策(name based decision): 使用元素名称发现映射关系;

基于实例决策(instance based decision): 使用本体实例发现映射关系;

RiMOM: 使用本文提出的方法发现映射关系;

带用户交互的 RiMOM(with initial points): 在本文方法中加入用户交互.

每个实验都分别评估 1∶ 1 映射、n∶ 1 映射和总体映射(Overall)的结果.

5.2 实验结果

(1) 实验

为了评估本文提出的方法和用户交互的效果, 本文在五个数据集上分别作了实验. 为简化描述, 这里用 Cornell 和 Wash 分别表示 Course Catalog Ⅰ数据集中的 Cornell 本体和 Washington 本体; 用 Standard 和 Yahoo 分别表示 Company Profiles 数据集中的 Standard.com 本体和 Yahoo.com 本体; 用 E1 和 E2 分别表示 Employee Ontology 数据集中的本体 1 和本体 2; 用 Sale1 和 Sale2 分别表示 Sales Ontology 中的本体 1 和本体 2; Ref 表示 EON 数据集中的目标/ 参考本体; 101~ 304 表示 EON 中的其它源本体. 表 5~ 7 分别给出在这五个数据集上, 基于元素名称决策、基于实例决策和 RiMOM 的实验结果(Prec. 和 Rec. 分别表示查准率和查全率); 表 8 给出带用户交互的 RiMOM 在前四个数据集上的实验结果, 本文没有给出带用户交互的 RiMOM 在 EON 上的实验结果, 因为: (1) 在 EON 的多个映射任务中, RiMOM 已经能够取得很好的效果; (2) 相对其它的数据集来说 EON 的本体包含的元素个数相对较少.

表 5 基于元素名称决策的实验结果(%)

数据集	映射	1∶ 1		n∶ 1		overall	
		Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
Course Catalog Ⅰ	Cornell to Wash	85.29	85.29	—	—	85.29	85.29
	Wash to Cornell	79.49	83.78	—	—	79.49	83.78
Company Profiles	Standard to Yahoo	64.00	72.40	—	—	64.00	72.40
	Yahoo to Standard	67.38	73.26	—	—	67.38	73.26
Employee Ontology	E1 to E2	85.60	78.00	50.50	57.00	69.49	64.30
	E2 to E1	76.83	83.89	47.30	62.56	66.57	72.78
Sales Ontology	Sale1 to Sale2	76.30	70.50	58.30	59.00	68.50	62.50
	Sale2 to Sale1	81.88	76.17	63.20	71.12	79.44	75.07
EON	101 to Ref	97.00	100.00	—	—	97.00	100.00
	103 to Ref	97.00	100.00	—	—	97.00	100.00
	104 to Ref	97.00	100.00	—	—	97.00	100.00
	201 to Ref	2.00	2.00	—	—	2.00	2.00
	202 to Ref	2.00	2.00	—	—	2.00	2.00
	204 to Ref	93.00	96.00	—	—	93.00	96.00
	205 to Ref	45.00	46.00	—	—	45.00	46.00
	221 to Ref	97.00	100.00	—	—	97.00	100.00
	222 to Ref	91.00	95.00	—	—	91.00	95.00
	223 to Ref	93.00	96.00	—	—	93.00	96.00
	224 to Ref	97.00	100.00	—	—	97.00	100.00
	225 to Ref	97.00	100.00	—	—	97.00	100.00
	228 to Ref	100.00	100.00	—	—	100.00	100.00
	230 to Ref	79.00	99.00	—	—	79.00	99.00
	301 to Ref	52.00	80.00	—	—	52.00	80.00
	302 to Ref	34.00	67.00	—	—	34.00	67.00
	303 to Ref	40.00	79.00	—	—	40.00	79.00
	304 to Ref	77.00	95.00	—	—	77.00	95.00

表 6　基于实例决策的实验结果(%)

数据集	映射	1∶1		n∶1		overall	
		Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
Course Catalog Ⅰ	Cornell to Wash	75.00	61.76	—	—	75.00	61.76
	Wash to Cornell	93.55	78.38	—	—	93.55	78.38
Company Profiles	Standard to Yahoo	80.00	87.50	—	—	80.00	87.50
	Yahoo to Standard	71.40	88.90	—	—	71.40	88.90
Employee Ontology	E1 to E2	55.00	43.50	40.50	66.50	52.50	50.00
	E2 to E1	64.50	56.38	54.68	63.49	61.27	59.64
Sales Ontology	Sale1 to Sale2	88.50	79.00	78.50	65.00	84.50	74.80
	Sale2 to Sale1	84.76	73.32	81.09	70.5	82.49	71.24
EON	101 to Ref	97.00	100.00	—	—	97.00	100.00
	103 to Ref	97.00	100.00	—	—	97.00	100.00
	104 to Ref	97.00	100.00	—	—	97.00	100.00
	201 to Ref	90.00	93.00	—	—	90.00	93.00
	202 to Ref	46.00	43.00	—	—	46.00	43.00
	204 to Ref	95.00	98.00	—	—	95.00	98.00
	205 to Ref	70.00	68.00	—	—	70.00	68.00
	221 to Ref	97.00	100.00	—	—	97.00	100.00
	222 to Ref	90.00	93.00	—	—	90.00	93.00
	223 to Ref	95.00	98.00	—	—	95.00	98.00
	224 to Ref	84.00	87.00	—	—	84.00	87.00
	225 to Ref	96.00	99.00	—	—	96.00	99.00
	228 to Ref	91.00	91.00	—	—	91.00	91.00
	230 to Ref	78.00	97.00	—	—	78.00	97.00
	301 to Ref	36.00	54.00	—	—	36.00	54.00
	302 to Ref	28.00	46.00	—	—	28.00	46.00
	303 to Ref	30.00	50.00	—	—	30.00	50.00
	304 to Ref	58.00	70.00	—	—	58.00	70.00

表 7　**RiMOM** 的实验结果(%)

数据集	映射	1∶1		n∶1		overall	
		Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
Course Catalog Ⅰ	Cornell to Wash	91.18	91.18	—	—	91.18	91.18
	Wash to Cornell	88.89	86.49	—	—	88.89	86.49
Company Profiles	Standard to Yahoo	81.00	89.30	—	—	81.00	89.30
	Yahoo to Standard	73.12	89.74	—	—	73.12	89.74
Employee Ontology	E1 to E2	86.56	84.00	71.66	90.50	82.61	85.89
	E2 to E1	78.38	84.43	63.21	67.39	73.00	78.59
Sales Ontology	Sale1 to Sale2	94.00	91.50	88.60	93.00	91.60	92.00
	Sale2 to Sale1	89.52	86.46	73.63	71.17	86.37	83.44
EON	101 to Ref	97.00	100.00	—	—	97.00	100.00
	103 to Ref	97.00	100.00	—	—	97.00	100.00
	104 to Ref	97.00	100.00	—	—	97.00	100.00
	201 to Ref	88.00	90.00	—	—	88.00	90.00
	202 to Ref	41.00	41.00	—	—	41.00	41.00
	204 to Ref	94.00	98.00	—	—	94.00	98.00
	205 to Ref	62.00	64.00	—	—	62.00	64.00
	221 to Ref	97.00	100.00	—	—	97.00	100.00
	222 to Ref	91.00	95.00	—	—	91.00	95.00
	223 to Ref	93.00	96.00	—	—	93.00	96.00
	224 to Ref	96.00	99.00	—	—	96.00	99.00
	225 to Ref	97.00	100.00	—	—	97.00	100.00
	228 to Ref	100.00	100.00	—	—	100.00	100.00
	230 to Ref	76.00	95.00	—	—	76.00	95.00
	301 to Ref	92.00	77.00	—	—	92.00	77.00
	302 to Ref	79.00	54.00	—	—	79.00	54.00
	303 to Ref	78.00	75.00	—	—	78.00	75.00
	304 to Ref	96.00	95.00	—	—	96.00	95.00

表 8 带用户交互的 RiMOM 的实验结果(3 个映射初始点)(%)

数据集	映射	1∶1		n∶1		overall	
		Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
Course Catalog Ⅰ	Cornell to Wash	94.12	94.12	—	—	94.12	94.12
	Wash to Cornell	94.74	97.30	—	—	94.74	97.3
Company Profiles	Standard to Yahoo	83.50	90.50	—	—	83.50	90.50
	Yahoo to Standard	73.46	90.38	—	—	73.46	90.38
Employee Ontology	E1 to E2	88.50	86.30	76.50	84.00	85.00	85.40
	E2 to E1	81.48	85.51	67.82	64.90	77.16	79.20
Sales Ontology	Sale1 to Sale2	95.80	94.80	92.50	91.00	94.30	93.09
	Sale2 to Sale1	91.06	88.24	80.36	78.92	88.48	85.72

图 6 给出了这四种方法在这五个数据集上实验结果的比较,其中图 6(a)给出这四种方法在前四个数据集上的实验结果,图 6(b)给出基于元素名称决策、基于实例决策及 RiMOM 在 EON 数据集上的实验结果.

图 6(a)中的柱状图从左到右分别表示基于元素名称决策、基于实例决策、RiMOM 以及带用户交互的 RiMOM 的查准率.图 6(b)中的柱状图从左到右分别表示基于元素名称决策、基于实例决策及 RiMOM 的查准率.

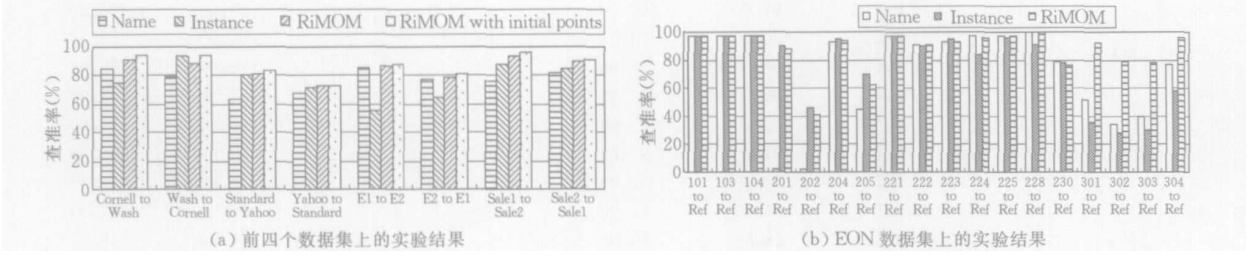


图 6 实验结果比较

从实验结果可见 RiMOM 在几乎所有的映射任务中都能取得较好的效果.在大多数映射任务中,RiMOM 都要好于两个基线方法.对所有的实验结果,本文都做了统计检验(Sign Test),p 值远远小于 0.01,表明从统计意义上来说 RiMOM 也明显好于基线方法.

(2) 实验分析

以下对前四个数据集上的映射结果和 EON 上的映射结果分别进行分析.

①较好的映射效果.在 Course Catalog Ⅰ, Company Profile, Employee Ontology 和 Sale Ontology 的映射中,查准率介于 73%~91.6% 的范围;查全率介于 83.44%~92%.表明本文提出的映射方法是有效的.在 EON 的大多数映射任务中,RiMOM 也能取得较好的效果,平均查准率和查全率为 87.28% 和 87.72%.

②本体实例的作用.在 Course Catalog Ⅰ, Company Profile 和 Sale Ontology 的映射任务中,基于实例的策略要好于基于元素名称的策略(除了从 Cornell 到 Wash 的映射,基于实例的策略都超过基于元素名称的策略,从 3%~16% 不等).另一方面,可以发现当两个本体的相似实例较少的时候,基于实例的策略效果较差,例如:Employee Ontology 中的两个本体相似实例比较少,基于实例决策的映射查准率只有 52.5%,而 Sales Ontology 中的两个本体有较多相似实例,基于实例决策的映射查准率则可以达到 91.6%.在 EON 上,基于实例的决策也优于基于元素名称的决策(查准率平均提高 6.59%,查全率平均提高 2.06%).

③相对于基线方法的提高.相对于基于元素名称的决策来说,RiMOM 明显提高了映射效果,查准率平均提高 15.60%(从 6.91%~33.72%),查全率平均提高 19.49%(从 3.23%~47.2%).相对于基于实例的决策来说,映射性能的提高也很明显,除了 Wash 到 Cornell 的映射,查准率平均提高 15.02%(从 1.25%~57.35%),查全率平均提高 26.79%(从 0.94%~47.64%).基于元素名称决策和基于实例决策存在的最大问题是:这两种方法都过于依赖本体中的某一类信息,这使得它们对数据很敏感,在不同特点的数据上取得的映射结果往往差别很大.例如:当两个本体的元素名称比较相似的时候,基于元素名称的策略能够达到 85.29% 的查准率(例如:Cornell 到 Wash 的映射);然而当元素名称不相似的时候,查准率下降到 64.0%(例如:Stand-

ard 到 Yahoo 的映射); 基于实例的策略同样存在这样的问题. RiMOM 则尽可能利用本体的所有信息进行映射发现, 从而避免了这个问题. 在 EON 数据集上, RiMOM 同样超过基于元素名称的决策(查准率平均提高 14.25%, 查全率平均提高 6.19%)和基于实例的决策(查准率平均提高 21.78%, 查全率平均提高 8.37%).

④用户交互的有效性. 本体是语义 Web 发展的基础, 本体映射的好坏将直接影响到语义 Web 中的语义交互. 有目标的用户交互可以有效地提高映射的精度. 用户交互的方式很多: 用户反馈、用户指定约束以及映射初始点的设定. 本文采用映射初始点设定的方式. 带初始点的 RiMOM 较不带初始点的 RiMOM 分别提高查准率 3.56%, 查全率 2.74%.

⑤实验结果错误分析.

对于 1∶1 映射, 主要存在三种类型的错误: 超过 36% 的错误是因为源元素错误的映射到目标元素的父元素上. 大约 17.65% 的错误是由于源元素和目标元素的名称定义完全不同并且元素的相同实例也很少(这种映射很难用 RiMOM 目前的方法发现). 另外, 11.26% 的错误由于映射优化中的启发式规则导致.

对于 n∶1 映射, 大约 33% 的错误由于漏掉了一个或者两个源元素. 大约 25% 的错误由于映射结果中包含多余的源元素, 18% 的错误来自于映射表达式的错误. 这也说明 RiMOM 在复杂映射的映射表达式发现上还需要进一步提高, 可以借鉴 iMap[35] 使用的字串函数映射和数值函数映射的发现方法进行改进.

⑥算法复杂度. RiMOM 的算法复杂度为 $O(n^2)$, 它的实际时间开销因本体定义的不同而不同. 表 9 列出 RiMOM 在前四个数据集上各个策略的时间开销(表中没有列出 EON 上映射发现的时间开销, 因为EON中各个本体的元素个数较少, 时间开销较小). 数据表明系统目前的时间开销还很大, 尤其是当本体规模比较大的时候, 如 Company 本体. 从另一方面来看, 对于任意两个本体的集成来说, 它们的映射发现通常只需要进行一次就行了, 对实时性的要求并不是很高, 因此在一定程度上也是可以满足需要的.

表 9　算法时间开销

数据集	命名策略	命名描述	基于实例的决策	结构上下文
Course Catalog Ⅰ	3min32s	1min19s	1min35s	18s
Company Profiles	1h41min	52min40s	1h6min	15min
Employee Ontology	6min15s	4min37s	6min	37s
Sales Ontology	7min42s	4min52s	6min48s	43s

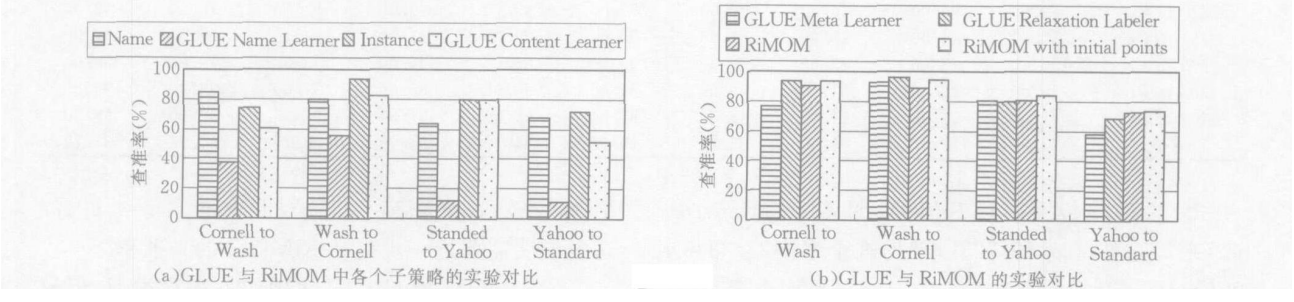
5.3　与已有方法比较

(1) 与 GLUE 比较

本文将 RiMOM 与已有的同类方法进行了比较, 首先同 GLUE 进行了比较, 其实验结果来自文献[15], 比较时使用了相同的数据集和相同的评测标准.

输入两个本体, GLUE 为源本体中的每一个概念发现它在目标本体中最相似的映射概念. GLUE 有两个基本的映射发现方法(称作 Base Learner): Content Learner 和 Name Learner, 这两个方法可以看作分别对应 RiMOM 中基于实例的策略和基于元素名称的策略. 在各个 Base Learner 进行映射发现后, GLUE 使用一个称为 Meta Learner 的方法对 Base Learner 的映射结果进行合并. 最后, 基于领域约束和启发式知识, GLUE 通过 Relaxation Labeler 的方法搜索最优映射.

图 7 给出 GLUE 和 RiMOM 的实验比较, 其中图 7(a) 分别比较了 GLUE 和 RiMOM 的各个子策略, 即基于元素名称的策略(Name) vs. Name Learner 以及基于实例的策略(Instance) vs. Content Learner; 图 7(b) 比较了 Meta Learner, Relaxation Labeler, RiMOM 以及带初始点的 RiMOM (RiMOM with initial points) 四个方法.



注: (Name 表示 RiMOM 中基于元素名称的决策, Instance 表示 RiMOM 中基于实例的决策, GLUE Name Learner 表示 GLUE 中使用元素名称的方法, GLUE Content Learner 表示 GLUE 中使用实例的方法, GLUE Meta Learner, GLUE Relaxation Labeler 表示 GLUE 中映射发现的两种策略)

图 7　GLUE 和 RiMOM 的实验比较

从实验结果的比较可以看出, RiMOM 中基于元素名称的策略要明显好于 GLUE 的 Name Learner; 基于实例的策略也明显好于 GLUE 的 Content Learner; 多数情况下, RiMOM 和带初始点的 RiMOM 优于 Meta Learner; 同 Relaxation Labeler 相比, RiMOM 在 Company Ontology 数据集上的映射结果更好, 在 Course Ontology Ⅰ数据集上的映射结果和 Relaxation Labeler 相当.

(2) 与 EON2004 的实验结果比较

本文还将 RiMOM 和 2004 年度本体映射国际竞赛(EON2004) 的结果进行了比较(包括 Karlsruhe2, Umontreal, Fujitsu 及 Stanford 四种方法, 实验结果来自 http://co4.inrialpes.fr/align/Contest/results/).

表 10 给出 RiMOM 和 EON2004 的实验比较, 表中给出查准率和查全率的结果, 符号"n/a"表示该实验没有输出结果.

从表 10 可以看出, RiMOM 的实验结果明显好于 Karlsruhe2 和 Umontreal; 和 Fujitsu 及 Stanford 的结果基本相当.

(3) 实验分析

①通过对比 GLUE 和 RiMOM 的各个子策略可见, RiMOM 的子策略, 即基于元素名称的策略和基于实例的策略, 明显优于 GLUE 中相应的子策略, 即 Name Learner 和 Content Learner(基于元素名称的策略超过 Name Learner 41.95% ~ 512.55% 不等; 基于实例的策略超过 Content Learner 大约 15%). 前者的主要原因是基于元素名称的策略结合使用了语义词典和统计方法, 而 Name Learner 使用分类器计算元素名称的相似度. 分类器通常在短文本内容上效果较差. 基于实例决策的优势来自于实例归一化等技术的应用.

②在两个映射任务中 RiMOM 优于 GLUE 的 Meta Learner 映射发现策略(Cornell 到 Wash 超过 18.42%, Yahoo 到 Standard 超过 23.39%). 在 Wash 到 Cornell 的映射上, RiMOM 的映射精度比 Meta Learner 低 4.42%. RiMOM 的优势来自于每个子策略的优势. 另一方面, 实验结果也同时显示 RiMOM 仅仅和 Relaxation Labeler 的映射结果相当, 这是因为 Relaxation Labeler 充分利用领域约束和启发式知识, 提高了映射精度; 同时也意味着 RiMOM 中的合并策略和启发式规则还不足以有效, 这也是本文的下一步工作之一.

表 10 RiMOM 和 EON2004 的实验比较(%)

算法映射	Karlsruhe2		Umontreal		Fujitsu		Stanford		RiMOM	
	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
101 to Ref	n/a	n/a	59.00	97.00	99.00	100.00	99.00	100.00	97.00	100.00
103 to Ref	n/a	n/a	55.00	90.00	99.00	100.00	99.00	100.00	97.00	100.00
104 to Ref	n/a	n/a	56.00	91.00	99.00	100.00	99.00	100.00	97.00	100.00
201 to Ref	43.00	51.00	44.00	71.00	98.00	92.00	100.00	11.00	88.00	90.00
202 to Ref	n/a	n/a	38.00	63.00	95.00	42.00	100.00	11.00	41.00	41.00
204 to Ref	62.00	100.00	55.00	90.00	95.00	91.00	99.00	100.00	94.00	98.00
205 to Ref	47.00	60.00	49.00	80.00	79.00	63.00	95.00	43.00	62.00	64.00
221 to Ref	n/a	n/a	61.00	100.00	98.00	88.00	99.00	100.00	97.00	100.00
222 to Ref	n/a	n/a	55.00	90.00	99.00	92.00	98.00	95.00	91.00	95.00
223 to Ref	59.00	96.00	59.00	97.00	95.00	87.00	95.00	96.00	93.00	96.00
224 to Ref	97.00	97.00	97.00	100.00	99.00	100.00	99.00	100.00	96.00	99.00
225 to Ref	n/a	n/a	59.00	97.00	99.00	100.00	99.00	100.00	97.00	100.00
228 to Ref	n/a	n/a	38.00	100.00	91.00	97.00	100.00	100.00	100.00	100.00
230 to Ref	60.00	95.00	46.00	92.00	97.00	95.00	99.00	93.00	76.00	95.00
301 to Ref	85.00	36.00	49.00	61.00	89.00	66.00	93.00	44.00	92.00	77.00
302 to Ref	100.00	23.00	23.00	50.00	39.00	60.00	94.00	65.00	79.00	54.00
303 to Ref	85.00	73.00	31.00	50.00	51.00	50.00	85.00	81.00	78.00	75.00
304 to Ref	91.00	92.00	44.00	62.00	85.00	92.00	97.00	97.00	96.00	95.00
平均	72.90	72.30	51.00	82.28	89.22	84.17	91.78	79.78	87.28	87.72

③在 EON 数据集上, RiMOM 明显优于 Karlsruhe2(平均查准率高 19.72%, 平均查全率高21.33%) 和 Umontreal(平均查准率高71.13%, 平均查全率高 6.12%). 和 Fujitsu 相比, RiMOM 平均查全率高 4.22%, 但查准率低 2.18%; 相比于 Stanford, RiMOM 平均查全率高 9.96%, 但查准率低 4.90%. 这也说明需要进一步提高 RiMOM 的查准率.

④RiMOM 和 GLUE 使用的方法很类似, 但分别又有不同的侧重点. 首先, RiMOM 针对不同的信息制定了不同的映射策略, 而 GLUE 对不同信息都

使用相同的映射方法(基于分类学习的方法); 其次 GLUE 使用 Relaxation Labeling 进行基于约束的优化, 而 RiMOM 使用风险最小化进行最优映射的发现; 另外, 在映射过程中, 两种方法分别使用了不同的映射策略, 例如: 对于元素名称, RiMOM 使用了基于语义词典和统计相结合的方法, 而 GLUE 使用了基于文本分类的方法; 对于实例, RiMOM 通过归一化等技术进行预处理, 而 GLUE 没有这种预处理过程; 此外, GLUE 没有提供用户交互的功能, 而实验表明用户交互能有效地提高映射精度; 最后, GLUE 的策略合并方法 Relaxation Labeler 的表现要好于 RiMOM 中的策略合并方法.

6　相关工作

本体映射研究的综述请见文献[35~ 39]等. 根据研究重点的不同, 可将和本体映射相关的研究分为 Schema 映射、基于上层本体的映射、基于相似度的映射、基于机器学习的映射、基于组合方法的映射以及其它映射研究.

6 1　Schema 映射

在数据库领域, 关于 Schema 映射的研究很多, 例如: 文献[7~ 10, 21, 40], 综述请见文献[16]. 其主要方法是通过定义全局模式来描述所有的分布数据, 这样数据集成问题就变为分布数据库模式到全局数据库模式的映射问题. 然而基于本体的信息互操作和语义集成问题是一个更加动态的知识共享过程, 这种全局模式的方法显得有些不太适合. 这里举例介绍部分相关研究工作和系统.

目前 Schema 映射已经有了相对比较成熟的研究, 但本体映射不同于 Schema 映射[28, 36]. 首先, Schema 没有为数据提供清晰的语义, 而本体为数据表示提供了清晰的形式化表示, 在本体映射中可以充分利用这些形式化的语义信息; 其次 Schema 的目的不是专门用来共享和可重用的, 然而本体最基本的任务之一就是为了实现可重用和共享; 再次, 本体开发需要在一个越来越分布的环境下完成; 最后 Schema 映射需要考虑每一个数据变化(例如: 添加新的类)对映射结果的影响, 而本体中知识表示的原语更加丰富、复杂, 包括反函数、cardinality 约束、传递属性、类型检查约束等, 这些丰富的原语既为本体映射提供了有用的信息, 也可能给本体映射带来新的困难. 以上这些差异使得 Schema 映射的方法不能简单直接地用到本体映射中.

尽管 Schema 映射和本体映射存在非常大的差异, 许多面向 Schema 映射的基本方法和技术还是可以被借鉴到本体映射的研究中. 实际上, 目前许多 Schema 映射系统都在进行扩展以支持本体映射.

6 2　基于上层本体的映射方法

目前, 许多研究组织开始着手研究‘通用’的上层本体(common top-level ontology), 用这些本体描述最基本的概念, 如事件(event)、时间(time)、空间(space)、事物(thing)、人物(human)、处理流程(process)等, 其中一些通用本体还逐渐成为领域标准, 例如 SUMO[41] 和 DOLCE[42]. 上层本体的目的就是提供一个通用的词汇集作为领域本体定义的基础. 基于上层本体建立映射的基本思想是: 首先定义通用上层本体, 然后不同的领域本体分别基于这些上层本体建立, 这样不同领域本体之间的映射问题就可以利用它们和上层本体之间的关系实现. 这种方法的前提是所有领域本体必须基于上层本体建立, 这严重地限制了该方法的通用性.

6 3　基于相似度计算的本体映射

基于相似度计算的本体映射计算两个实体元素之间的相似度, 映射的发现问题可以看作搜索相似度最大的两个实体元素的问题. 计算相似度的方法很多, 例如: 利用 Minkowski 距离公式.

基于相似度的本体映射方法的基本思想是: 本体 O_1 到 O_2 的映射发现问题, 可以分解为 O_1 中的每一个元素 x 搜索其在 O_2 中最相似的映射对象 x'. 这种方法获得的结果是局部最优解. 一些研究者对该方法进行扩展, 试图使其支持全局最优的映射发现.

例如, Melnik 等人提出了 Similarity Flooding 的本体映射算法[12]. 该算法是一个通用的图匹配算法, 首先将两个本体转换成有向图, 其中点表示概念, 在计算两个点之间相似度的时候, 同时考虑图中相邻节点之间的相似度(相邻节点是通过分类关系或者连接关系所关联的概念). 算法是一个迭代过程, 首先计算两个点之间的初始相似度, 然后在每次迭代中都考虑相邻节点之间的相似度. 类似方法的其它研究还包括文献[43, 44]等.

基于相似度的映射方法在多数情况下都只能发现局部最优的映射结果, 对其进行扩展以发现全局最优映射需要面向特定应用建立映射规则, 这在一定程度上限制了该方法的普遍应用.

6.4 基于机器学习的映射

基于机器学习的映射方法将映射问题转换成分类问题,为某个概念选择最优映射的问题就转换成对其进行分类的问题.分类学习的方法通常利用一个本体中的信息学习分类模型,然后利用另一个本体中的信息预测其每个元素可能的映射对象.

基于机器学习的本体映射通常利用已有的机器学习方法,如使用支持向量机(Support Vector Machines, SVMs)[45]、形式概念分析(Formal Concept Analysis, FCA)[46,47]、贝叶斯学习(Bayes Learning, BL)[48]以及神经网络(Neural Networks, NN)[49]等.

Doan 等人提出通过学习实例的联合概率分布自动发现映射关系,并开发了原型系统 GLUE[15].GLUE 的基本思想是:两个概念相同的实例越多,那么它们之间映射的可能性就越大.

利用其它机器学习算法发现本体映射的系统还包括 APFEL[50], CAIMAN[51], OMEN[52,53] 等.

以上这些方法都没有充分利用本体中的所有可用信息,如元素名称、本体约束及本体结构上下文等信息;此外,支持 n∶1 等映射类型的系统还很少.

6.5 组合映射

利用上面介绍的方法,学术界开发了许多本体映射的原型系统和应用工具,其中很多系统集成了各种不同的映射方法以试图提高映射精度,少数系统还提供了用户交互的功能.例如:Anchor-PROMPT[13]和 Chimaera[11]等.

6.6 其它本体映射研究

还有许多关于本体映射的研究.基于语义推理的映射方法的主要思想是利用本体的约束或逻辑来验证映射结果,主要方法包括命题可满足性(Propositional Satisfiability, SAT)、形式可满足性技术(Modal SAT Techniques)以及基于描述逻辑(Description Logic)的方法.例如,SAT 将本体映射问题转换成一系列的命题公式,将映射发现问题转换成命题公式合法性的验证问题[54].基于语义推理的映射方法往往需要和其它方法结合使用,即首先由其它映射方法指定一些候选映射,然后利用本体的约束和逻辑对候选映射进行验证,以提高映射精度.

除此之外,还有一些关于复杂映射的研究,例如,Multi-matcher System[55]和 iMap[34]等.这些研究主要针对复杂字串函数和数值函数映射发现.关于映射效率的研究包括 QOM[32]等,其基本思想是想通过改进映射算法,提高映射效率,同时使得映射精度的损失尽可能小.

7 总 结

本章主要研究本体映射问题.基于贝叶斯决策理论,本文提出了风险最小化的本体映射方法 RiMOM.RiMOM 同时支持多种映射类型的发现,包括 n∶1, 1∶*null*, *null*∶1 和 1∶1 映射;综合利用了本体中的各种信息,实现了多策略的映射方法;RiMOM 还支持用户交互.实验表明 RiMOM 可以有效地提高映射精度.在与同类方法 GLUE 的比较中,RiMOM 能够取得更高的映射精度.在和 2004 年度本体映射国际竞赛 EON 的比较中,RiMOM 明显优于 Karlsruhe2 和 Umontreal 的方法,和 Fujitsu 和 Stanford 的方法相当.

致 谢 感谢香港科技大学的陆宏钧老师,他曾提出了许多非常有价值的建议.感谢审稿人的宝贵意见,让我们发现了论文和研究中的很多问题!

参 考 文 献

1 Berners-Lee T., Fischetti M., Dertouzos M.L.. Weaving the web: The original design and ultimate destiny of the World Wide Web, Harper, San Francisco, USA, 1999

2 Lenat D.B.. Cyc: A large-Scale investment in knowledge infrastructure. Communications of the ACM, 1995, 38(11): 32~38

3 Uschold M., King M., Moralee S., Zorgios Y.. The enterprise ontology. The Knowledge Engineering Review, Special Issue on Putting Ontologies to Use, 1998, 13(1): 31~89

4 Miled Z.B., Webster Y.W., Li N., Bukhres O., Nayar A.K., Martin J., Oppelt R.. BAO, a biological and chemical ontology for information integration. Online Journal of Bioinformatics, 2002, 1: 60~73

5 Hammer J., Garcia-Molina H., Ireland K., Papakonstantinou Y., Ullman J., Widom J.. Information translation, mediation, and mosaic-based browsing in the TSIMMIS system. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, San Jose, California, 1995, 483

6 Levy A.Y., Rajaraman A., Ordille J.J.. Querying heterogeneous information sources using source descriptions. In: Proceedings of the VLDB'96, Bombay, India, 1996, 251~262

7 Do H., Rahm E.. Coma: A system for flexible combination of schema matching approaches. In: Proceedings of the 28th International Conference on Very Large Data Bases, Hong Kong, 2002, 610~621

8 Doan A.H., Domingos P., Halevy A.. Reconciling schemas of disparate data sources: A machine-learning approach. In:

Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, 2001

9 Kang J., Naughton J.. On schema matching with opaque column names and data values. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, Santa Barbara, CA, USA, 2003, 205~216

10 Madhavan J., Bernstein P., Rahm E.. Generic schema matching using Cupid. In: Proceedings of the 27th International Conference on Very Large Data Bases, Rome, Italy, 2001, 48~58

11 McGuinness D., Fikes R., Rice J., Wilder S.. An environment for merging and testing large ontologies. In: Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning, Colorado, USA, 2000, 483~493

12 Melnik S., Molina-Garcia H., Rahm E.. Similarity flooding: A versatile graph matching algorithm. In: Proceedings of the 18th International Conference on Data Engineering, San Jose, California, USA, 2002, 117~128

13 Noy N.F., Musen M.A.. PROMPT: Algorithm and tool for automated ontology merging and alignment. In: Proceedings of the 2000 National Conference on Artificial Intelligence, Austin, Texas, 2000, 450~455

14 Palopoli L., Terracina G., Ursino D.. The system DIKE: Towards the semi-automatic synthesis of cooperative information systems and data warehouses. In: Proceedings of the 2000 ADBIS-DASFAA Symposium on Advances in Databases and Information Systems, Prague, Czech Republic, 2000, 108~117

15 Doan A., Madhavan J., Domingos P., Halevy A.. Learning to map between ontologies on the semantic web. In: Proceedings of the 11th World Wide Web Conference, 2002, 662~673

16 Rahm E., Bernstein P.A.. A survey of approaches to automatic schema matching. The VLDB Journal, 2001, 10(4): 334~350

17 Tang J., Liang B., Li J.. Multiple strategies detection in ontology mapping. In: Proceedings of the 14th International World Wide Web Conference (WWW'2005), Chiba, Japan, 2005, 992~993

18 Tang J., Li J., Liang B., Huang X., Li Y., Wang K.. Using Bayesian decision for ontology mapping. Journal of Web Semantics, 2006(Accepted)

19 Dean M., Schreiber G., Bechhofer S., van Harmelen F., Hendler J., Horrocks I., McGuinness D.L., Patel-Schneider P.F., Andrea Stein L.. OWL web ontology language reference. W3C recommendation, 2004

20 Ting K.M., Witten I.H.. Issues in stacked generalization. Journal of Artificial Intelligence Research, 1999, 10: 271~289

21 Kim W., Seo J.. Classifying schematic and data heterogeneity in multi-database systems. IEEE Computer, 1991, 24(12): 12~18

22 Bouquet P., Euzenat J., Franconi E., Serafini L., Stamou G., Tessaris S.. Specification of a common framework for characterizing alignment. Knowledge Web Deliverable 2.2.1v2, University of Karlsruhe, 2004

23 Wiesman F., Roos N., Vogt P.. Automatic ontology mapping for agent communication. Technical Report, 2001

24 Berger J.. Statistical Decision Theory and Bayesian Analysis. New York: Springer-Verlag, 1985

25 Madhavan J., Bernstein P., Chen K., Halevy A., Shenoy P.. Corpus based schema matching. In: Proceedings of the IJCAI'2003 Workshop on Information Integration on the Web (IIWeb'2003), Acapulco, Mexico, 2003

26 Pantel P., Lin D.. Discovering word senses from text. In: Proceedings of the 2002 ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, 2002, 613~619

27 Mitchell T.M.. Machine Learning. McGraw Hill, Columbus, USA, 1997, 154~199

28 Maedche A., Moltik B., Silva N., Volz R.. MAFRA-An ontology MApping FRAmework in the context of the semantic web. In: Proceedings of the EKAW 2002, Siguenza, Spain, 2002, 235~250

29 Silva N., Rocha J.. Semantic web complex ontology mapping. In: Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence, Halifax, Canada, 2003, 82~100

30 Cunningham H., Maynard D., Bontcheva K., Tablan V.. GATE: A framework and graphical development environment for robust NLP tools and applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 2002, 168~175

31 Sproat R., Black A., Chen S., Kumar S., Ostendorf M., Richards C.. Normalization of non-standard words. In: Proceedings of the WS'99 Final Report, Baltimore, Maryland, USA, 1999, 12~48

32 Ehrig M., Staab S.. QOM — Quick ontology mapping. In: Proceedings of the 4th International Semantic Web Conference, Hiroshima, Japan, 2004, 683~697

33 Gillick L., Cox S.. Some statistical issues in the comparison of speech recognition algorithms. International Conference on Acoustics Speech and Signal Processing, 1989, 1: 532~535

34 Dhamankar R., Lee Y., Doan A.H., Halevy A., Domingos P.. iMAP: Discovering complex semantic matches between database schemas. In: Proceedings of the SIGMOD'2004, Paris, France, 2004, 383~394

35 Euzenat J.. State of the art on ontology alignment. Knowledge Web Deliverable 2.2.3, University of Karlsruhe, 2004

36 Wache H., Voegele T., Visser U., Stuckenschmidt H., Schuster G., Neumann H., Huebner S.. Ontology-based integration of information — A survey of existing approaches. In: Proceedings of IJCAI'2001 Workshop on Ontologies and Information Sharing, Seattle, Washington, USA, 2001, 108~117

37 Kalfoglou Y., Schorlemmer M.. Ontology mapping: The state of the art. The Knowledge Engineering Review, 2003, 18(1):

1~ 31

38 Kalfoglou Y., Hu B., Reynolds D., Shadbolt N.. Semantic integration technologies. In: Proceedings of the 6th Month Deliverable, University of Southampton and HP Labs, ECS e-Prints Report # 10842, 2005

39 Noy N. F.. Semantic integration: A survey of ontology based approaches. SIGMOD Record, 2004, 33(4): 65~ 70

40 Melnik S., Rahm E., Bernstein P.. Rondo: A programming platform for model management. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, CA, US, 2003, 193~ 204

41 Niles I., Pease A.. Towards a standard upper ontology. In: Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS' 2001), Ogunquit, Maine, 2001, 2~ 9

42 Gangemi A., Guarino N., Masolo C., Oltramari A.. Sweetening wordnet with DOLCE. AI Magazine, 2003, 24(3): 13~ 24

43 Euzenat J., Valtchev P.. Similarity-based ontology alignment in OWL-lite. In: Proceedings of the 15th ECAI, Valencia (ES), 2004, 333~ 337

44 Hovy E.. Combining and standardizing largescale, practical ontologies for machine translation and other uses. In: Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC), Granada, Spain, 1998, 535~ 542

45 Zhang D., Lee W. S.. Web taxonomy integration using support vector machines. In: Proceedings of the World Wide Web Conference (WWW' 2004), New York, USA, 472~ 481

46 Ganter B., Stumme G., Wille R.. Formal Concept Analysis, Foundations and Applications. Springer, 2005

47 Stumme G., Madche A.. FCA-Merge: Bottom-up merging of ontologies. In: Proceedings of 7th International Conference on Artificial Intelligence (IJCAI' 2001), Seattle, WA, 2001, 225~ 230

48 Berlin J., Motro A.. Database schema matching using machine learning with feature selection. In: Proceedings of the Conference on Advanced Information System Engineering (CAiSE' 2002), Toronto, Ontario, Canada, 2002, 452~ 466

49 Li W. S., Clifton C.. Semantic integration in heterogeneous databases using neural networks. In: Proceedings of the 20th VLDB, Santiago (CH), 1994, 1~ 12

50 Ehrig M., Staab S., Sure Y.. Bootstrapping ontology alignment methods with APFEL. In: Proceedings of the 2nd International Semantic Web Conference (ISWC' 2005), Galway, Ireland, 2005, 186~ 200

51 Lacher M., Groh G.. Facilitating the exchange of explicit knowledge through ontology mappings. In: Proceedings of the 14th International FLAIRS conference, Key West, FL, USA, 2001, 305~ 309

52 Mitra P., Noy N., Jaiswal A.. Ontology mapping discovery with uncertainty. In: Proceedings of the 2nd International Semantic Web Conference (ISWC' 2005), Galway, Ireland, 2005, 537~ 547

53 Su X.. A text categorization perspective for ontology mapping. Technical Report, 2002

54 Giunchiglia F., Shvaiko P.. Semantic matching. The Knowledge Engineering Review, 2004, 18(3): 265~ 280

55 Xu L., Embley D.. Using domain ontologies to discover direct and indirect matches for schema elements. In: Proceedings of the Semantic Integration Workshop at ISWC' 2003, Sanibel Island, 2003, 1~ 6

TANG Jie, born in 1977, Ph. D.. His main research interests include semantic Web annotation, ontology interoperability, machine learning, text mining, and information extraction.

LIANG Bang-Yong, born in 1978, Ph. D.. His research interests include domain data management, ontology development, text processing, and information retrieval.

LI Juan-Zi, born in 1964, Ph. D., associate professor. Her main research interests include Chinese information processing and knowledge discovery and management on the Web.

WANG Ke-Hong, born in 1941, professor, Ph. D. supervisor. His research interests include knowledge engineering and distributed knowledge processing.

Background

This work is supported by the National Natural Science Foundation of China under Grant No. 90604025, entitled Research of Domain Oriented Key Technology of Semantic Web and its Applications. The work addresses issues of semantic annotation, ontology mapping, and semantic search. In particular, the authors are focused on investigating the issues in domain oriented applications. Currently, they have made some progress in the three topics, and have a dozen of related papers published at international conferences and journals. They also developed a prototype system for the work. The work introduced in the paper is concerned with the problem of ontology mapping.