一种基于虚拟路径的本体匹配算法

黄 涛 崔弘扬 刘清堂 杨宗凯
(国家数字化学习工程技术研究中心 武汉 430079)
(华中师范大学教育信息技术工程研究中心 武汉 430079)

摘 要 本体匹配是实体之间关系的体现,不仅单个概念之间的关系影响本体的匹配关系,概念相邻元素及其语义联系对本体的匹配关系的影响也不容忽视。提出了基于虚拟路径的本体匹配新方法,通过为两个本体元素分别建立由具有语义联系的相邻元素及其联系所构成的虚拟路径,将两个目标元素的虚拟路径中各独立要素分别对应地进行概念语义相似性比较;综合虚拟路径内各独立要素的概念语义相似性,获取两个目标元素虚拟路径的图形语义相似性;根据虚拟路径的图形语义相似性推导两个目标元素之间的映射关系。实验表明,该方法能够有效提高本体匹配的质量和性能。

关键词 虚拟路径,本体匹配,匹配算法,概念语义相似性,图形语义相似性
中图法分类号 TP182 **文献标识码** A

Ontology Matching Approach Based on Virtual Path

HUANG Tao CUI Hong-yang LIU Qing-tang YANG Zong-kai
(National Engineering Research Center for E-learning, Wuhan 430079, China)
(Engineering Research Center for Education Information Technology, Huazhong Normal University, Wuhan 430079, China)

Abstract Ontology Matching is to Measuring the relation between entities of ontology. It relies not only on the relation between concepts, but also between concept neighbors and semantic associations. A new Ontology Matching Approach Based on Virtual path was proposed. Through the definition of virtual path around the two concepts, the approach utilized the semantic similarity of concepts, and computed the combined similarity of two virtual paths of two concepts with graph matching. After computing the similarity of two virtual paths, it decided the matching relation of ontology element. The experiment showed that the approach could increase ontology measuring quality and performance effectively.

Keywords Virtual path, Ontology matching, Matching approach, Semantic similarity of concept, Semantic similarity of graph

1 引言

用户请求与知识服务提供之间存在着相关性的某种客观评价标准。明确地说,就是所有的服务中都有一些客观存在的特征信息,这些特征信息可以和用户的查询需求按照某种算法进行匹配。相关性之所以复杂,是因为信息检索领域体现的是用户人脑内的概念化体系和计算机(系统)内的概念化体系之间的匹配问题。匹配的定义是为了找到两个数据模型或者本体之间的语义联系,并产生一个匹配的结果。

本体作为一种语义和知识层面上的概念共享模型,自提出以来就引起数据整合、P2P 系统、电子商务、语义 Web 服务、社会网络(Social Networks)等应用领域科研人员的广泛关注,并得到了有效的研究与应用,提出了一系列基于本体的领域问题解决方法。本体匹配(Ontology Matching)是发现不同本体之间实体元素(包括本体的类、属性或者个体)匹配关系的关键技术,已被国内外学者普遍认为是解决计算机系统语义异构问题的有效手段之一。

本体匹配中匹配元素是四元组$\langle mID, Ni1, Nj2, R\rangle$,$i=1,\cdots,h$;$j=1,\cdots,k$;其中 mID 是独一无二的标示符,$Ni1$ 表示第一个图形的第 i 个结点,$Nj2$ 表示第二个图形的第 j 个结点,h 是第一个图形结点的数量,k 是第二个图形结点的数量,R 表示其中结点的相似关系。匹配就是通过一定的方法找出这种匹配关系的过程[1]。文献[1]对本体匹配操作的定义是输入两个本体,每个本体都由具体的实体 Entity(包含类 Class、属性 properties、关系 Relationships)等构成,最后的输出结果决定了这些实体之间的关系(等于、包含等关系)。

本体匹配方法可分为基于元素的方法和基于结构的方法[1,2]。基于元素的方法可以理解为一种分析独立元素的方

到稿日期:2009-12-24 返修日期:2010-03-10 本文受国家高技术研究计划(863 计划)(2008AA01Z127),教育部人文社会科学研究项目基金(09YJC870009)资助。

黄 涛(1979—),男,博士,讲师,主要研究方向为知识发现与知识工程、计算机应用技术、远程教育技术等,E-mail:tmht@mail.ccnu.edu.cn;崔弘扬(1986—),女,硕士生,主要研究方向为本体、文本挖掘;刘清堂(1969—),男,博士,教授,博士生导师,主要研究方向为知识工程、远程教育关键技术等;杨宗凯(1963—),男,博士,教授,博士生导师,主要研究方向为教育信息化、计算机网络等。

法，特点是将元素从待比较的本体结构中提取出来后进行独立比较，不考虑本体中其他周围要素对元素的影响，主要是利用外部信息源或者元素的概率信息等来计算单个元素之间的语义相似度或者语义距离。基于结构的方法可以理解为是一种分析元素本体结构的方法，特点是利用元素在本体的图形结构或者逻辑描述中所处的地位，考虑结构中其他元素的影响，以计算单个元素之间的语义相似度。目前的本体匹配方法主要侧重于本体概念本身，以及概念的实例信息来求取本体元素的语义相似性，并没有充分挖掘本体结构中概念的相邻元素及其语义联系。由于本体是概念以及概念关系的体现，因此概念的相邻元素及其语义联系对概念的语义影响是不可忽略的。

本文针对现有技术的不足，提出一种基于虚拟路径的本体匹配方法，克服了现有本体匹配方法忽略概念相邻元素及其语义联系的影响、利用本体语义程度低的缺陷。

2 相关研究工作

目前国内外有很多研究者[14-16]都在研究本体匹配技术。P. Shvaiko 等人对本体匹配操作的定义是输入两个本体，每个本体都由具体的实体 Entity（包含类 Class、属性 properties、关系 Relationships）等构成，最后的输出结果决定了这些实体之间的关系（等于、包含等关系）。

E. Rahm 提出了本体匹配方法分类。将模式/本体匹配方法分为独立匹配方法和混合匹配方法两类。独立匹配方法指一种可以单独进行本体匹配的方法，混合匹配方法中用到了两种以上的独立匹配方法。独立匹配方法包含基于模式的方法和基于实例的方法。基于模式的方法是目前研究比较多的一种方法，它包含元素层次的方法和结构层次的方法。

基于模式的匹配方法主要包括（1）元素层次的匹配方法：此匹配方法是指从实体本身来考虑某个实体之间的匹配，而不考虑实体周围的其他元素或者关系，也就是将元素从元素所处的虚拟中割裂出来单独考虑。（2）结构层次的匹配方法：此匹配方法不仅从实体本身来考虑某个实体之间的匹配，而且将实体所处的结构综合考虑，即将元素和元素的属性、属性值等元素所处的结构中与元素相关的要素进行综合考虑。（3）基于语言学的匹配方法：语言层次的匹配方法从实体的文本名称或者对实体的文本描述来考虑实体之间的匹配关系，适用于本体（模式）元素层次的匹配。目前运用比较广泛的基于语言学的匹配方法主要包括基于关键字词频的统计方法、基于向量空间模型（VSM）的统计方法、基于编辑距离（Edit Distance）的方法。（4）基于约束的匹配方法：此方法从实体的类型或者线索来取得实体之间的匹配关系，既适用于元素层次的匹配，也适用于结构层次的匹配。它根据本体（模式）实体的数据类型、值域、关系类型、实体的势等约束条件来计算相似性（Similarity）。基于约束的方法很少独立使用，一般都是配合其他方法一起使用。

与本体匹配方法分类相对应，本体匹配系统可以分为几大类：基于模式（Schema）的匹配系统、基于实例（Instance）的匹配系统、基于混合模式的匹配系统。

Cupid[3]是德国莱比锡大学（University of Leipzig）的 Erhard Rahm 和美国华盛顿大学（University of Washington）的 Jayant Madhavan 提出的一种模式匹配方法。该方法将匹配问题看成是计算两个模式元素的相似系数（Similarity Coefficient），系数的取值范围在[0，1]之间，然后通过相似系数来推导元素的匹配关系。该方法将模式匹配分成两个步骤，分别计算元素的独立语义性和元素的结构语义性。

COMA[4]（COmbination of MAtching algorithms）是由德国莱比锡大学 Do Hong-Hai 和 Erhard Rahm 提出的一种组合多种匹配方式的混合型模式匹配系统。与 Cupid 采用独立匹配方法不同，COMA 系统采用的是混合匹配方法，通过灵活地组合不同的匹配方式及其结果来推导最终的模式匹配结果。COMA 匹配系统将待比较的模式转化成带有根节点的有向无环图（Rooted Directed Acyclic Graphs），匹配操作将输入两个模式，然后决定两个模式中元素的语义匹配关系。

SF[5]是由斯坦福大学（Stanford Univeristy）的 Sergey Melnik 和德国莱比锡大学的 Erhard Rahm 于 2002 年提出的一种匹配多种数据源的通用结构层次，可以广泛适用于多个应用领域。SF 的基本思想是如果模式结构中两个相邻元素是相似的，那么可以推断这两个元素也是相似的。

Falcon-AO[6,7]（Finding，aligning and learning ontologies，ultimately for capturing knowledge via ontology-driven approaches）是由东南大学瞿裕忠教授和胡伟博士等人开发的基于模式的本体对齐工具，它分别通过语言特性（LMO）和结构特性（GMO）两个途径来研究本体的相似性，这两个方法都是比较本体中元素的匹配关系。

H-Match[8,9]是由意大利米兰大学的 Silvana Castano，Alfio Ferrara 等人提出的面向分布式本体的动态匹配方法，它以两个本体作为输入，并输出两个本体中具有语义相似性的元素对。相似性的分析是通过计算概念的[0，1]之间的语义相近系数（Semantic Affinity）。它是在模式匹配方法 Artemis 的基础上，借鉴基于 WordNet 词义系统的方法来计算概念的语言层次相似性（Linguistic Affinity），然后在语言层次概念相似性的基础上，给出了 4 个层次的结构相似性。

S-Match[10,11]是由意大利特兰托大学的 Fausto Giunchiglia，Pavel Shvaiko 等人提出的模式语义匹配系统，它与 Cupid，COMA 等模式匹配系统采用的方法有很大的区别。Cupid，COMA 等系统将模式匹配分为元素层次匹配和结构层次匹配，并且匹配的结构用[0，1]之间的相似系数表示。而 S-Match 采用概念标签匹配、概念语义匹配两个步骤来推导概念的语义联系，匹配的结果用语义关系符（属于⊑、包含⊒、不相交⊥等）来表示。因此，S-Match 被称为真正意义上的语义匹配系统。

GLUE[12]是比较典型的基于实例的匹配系统，它由美国华盛顿大学的 AnHai Doan，Jayant Madhavan 等人提出，是一种利用机器学习技术来发现本体匹配关系的算法。给定两个本体，针对其中一个本体的任意元素，GLUE 可以在另一本体中找到与之相匹配的元素。GLUE 的另一个关键技术是使用多策略学习（Multiple Learning Strategies），每一个学习策略针对某一类型的本体数据实例或者模式信息。

基于集合的模式匹配系统[13]（Corpus-based schema matching）是由美国华盛顿大学的 Jayant Madhavan 和微软研究院（Microsoft Research）的 Philip A. Bernstein 等人提出的扩展模式匹配系统，它的目标是解决模式匹配系统缺少足够实例信息的缺陷。该方法利用模式所处的外部文本集合的信

息来加强模式匹配的结果。

3 基于虚拟路径的本体匹配算法

本文提出的本体匹配方法思路如图 1 所示。根据元素提出的本体结构，利用本体结构的上下文环境构建元素的虚拟路径，在综合考虑概念元素概念在语言特征层面以及元素在本体结构中的上下文(前驱元素以及后驱元素)的基础上，计算各个独立要素的语义相似性，并加权平均。具体步骤如下。

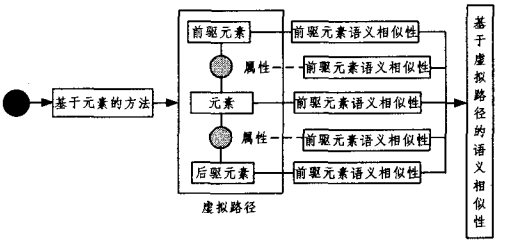

图 1 基于虚拟路径的本体匹配过程

步骤 1 将待比较的两个本体概念定为目标元素，为两个目标元素分别建立由具有语义联系的相邻元素及其联系所构成的虚拟路径；

步骤 2 将两个目标元素的虚拟路径中各独立要素分别对应进行概念语义相似性比较；

步骤 3 综合虚拟路径内各独立要素的概念语义相似性，获取两个目标元素虚拟路径的图形语义相似性；

步骤 4 根据虚拟路径的图形语义相似性推导两个目标元素之间的映射关系。

3.1 虚拟路径的构建

虚拟路径由具有语义联系的相邻元素及其联系构成。相邻元素包括有目标元素的前驱元素和后驱元素；虚拟路径的结构由 5 个独立要素组成，即前驱元素、前驱元素与目标元素之间的属性或者语义关系、目标元素、目标元素与后驱元素之间的属性或者语义关系、后驱元素。

如图 2 所示，b_2 和 b_2'为待比较的两个目标元素。其中 b_2 的前驱元素为 b_1，后驱元素为 b_3，构成的虚拟路径用图 2(a)的树状结构表示；其中 b_2'的前驱元素为 b_1'，后驱元素为 b_3'，构成的虚拟路径用图 2(b)的树状结构表示。

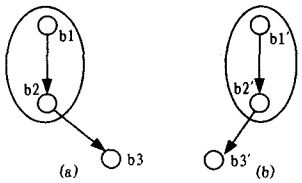

图 2 虚拟路径示例

3.2 概念语义相似性计算

本文提出了一种改进的概念语义相似性比较方法，以便高效准确地获取概念语义相似性比较结果。无论是前驱元素、目标元素或后驱元素，都以元素为单位进行比较，将待比较的元素标记为 $e1$ 和 $e2$。首先采用基于语言的方法，即去除待比较的两个元素 $e1$ 和 $e2$ 的元素名称中的冗余信息；然后用基于字符串的方法判断元素名称是否一致，即对元素名称逐个字符比较。如果一致，即元素名称的语义相似性为 1；如果不相似，再借助外部信息源计算两个元素 $e1$ 和 $e2$ 的概念语义相似性。计算公式如下：

$$\mathrm{Sim}E(e1,e2)=\frac{C_{base}}{C_{base}+sp+num}$$

其中，sp 表示元素 $e1$ 和 $e2$ 之间最短路径的长度，num 表示元素 $e1$ 和 $e2$ 之间最短路径的方向改变次数，C_{base} 表示 4 种关系的语义相似度基数，如果两个元素是相等关系，那么 $C_{base}=4C$；如果两个元素是包含与属于关系，那么 $C_{base}=3C$；如果两个元素是部分与整体关系，那么 $C_{base}=2C$；如果两个元素是不相交关系，那么 $C_{base}=C$；$C=1.0$。其中比较特殊的是相等关系和不相交关系，当两个元素相同时，$C_{base}=4$，$sp=0$，$num=0$，所以 $\mathrm{Sim}E(e1,e2)=1.0$。当两个元素不相交时，$C_{base}=1$，sp 和 num 都远大于 1，所以 $\mathrm{Sim}E(e1,e2)=0$。

如图 2 所示，其中(a)表示的元素 a_1 和 a_2 之间是部分或整体关系，元素 NCA 表示两个元素的最近共同祖先元素。可以得出，$C_{base}=2$，$sp=2$，$num=1$，所以元素 a_1 和 a_2 的概念语义相似度 $\mathrm{Sim}E(a_1,a_2)=2/(2+2+1)=0.4$。图 2(b)表示的元素 a_1'和 a_2'之间关系属于 Strong(包含或属于关系)，因此 $C_{base}=3$，$sp=1$，$num=0$，由此得出元素 a_1'和 a_2'的概念语义相似度 $\mathrm{Sim}E(a_1',a_2')=3/(3+1+0)=0.75$。

当对前驱元素与目标元素之间的属性或者语义关系、目标元素与后驱元素之间的属性或者语义关系进行概念语义相似性比较时，将属性或者语义关系视为一个元素，按上述独立要素比较方法实现即可。图 2 所示的虚拟路径中独立要素均按照独立要素比较方法分别比较后，得到前驱元素的概念语义相似性 $\mathrm{Sim}E(b_1,b_1')$、前驱元素与目标元素之间属性的概念语义相似性 $\mathrm{Sim}E(P(b_1),P(b_1'))$、目标元素的概念语义相似性 $\mathrm{Sim}E(b_2,b_2')$、目标元素与后驱元素之间属性的概念语义相似性 $\mathrm{Sim}E(P(b_2),P(b_2'))$以及后驱元素的概念语义相似性 $\mathrm{Sim}E(b_3,b_3')$。

本文提出的概念语义相似度比较方法是对现有 Hirst & St-Onge[21]语义相似度比较方法的改进。Hirst & St-Onge 语义相似度比较方法中 sp 和 num 都是作为差数出现的，而在本文所提供的方案中作为分母出现。这种方法的优点是可以处理不相交关系中 sp 为无穷大以及相同关系中 $sp=0$ 的特殊情况。另外，本文提供的计算公式是经过标准化处理的，语义相似度值域在[0,1]之间，能够节约计算机系统运行开销。而 Hirst & St-Onge 语义相似度比较方法中是以权重 Weight 的值作为相似度的值，不符合语义相似度在[0,1]区间的特点。

3.3 虚拟路径的图形语义相似性计算

考虑到元素与属性之间具有不可分割的意义，本文提出综合虚拟路径内各独立要素的概念语义相似性的方案为：首先对虚拟路径内各独立要素进行分组，第一组为前驱元素—前驱元素与目标元素之间的属性或者语义关系，第二组为目标元素，第三组为目标元素与后驱元素之间的属性或者语义关系—后驱元素；按分组将各独立要素的概念语义相似性加权综合为两个目标元素虚拟路径的图形语义相似性。然后按组加权综合，其中第一组根据 $\mathrm{Sim}E(b_1,b_1')$和 $\mathrm{Sim}E(P(b_1),P(b_1'))$求得表示该组元素以及元素属性或语义关系的相似关系的组合语义相似性 $\mathrm{Sim}EP(b_2^{pre},b_2'^{pre})$；第三组根据 $\mathrm{Sim}E(P(b_2),P(b_2'))$和 $\mathrm{Sim}E(b_3,b_3')$求得该组的元素属性或语义关系以及元素的相似关系的组合语义相似性 $\mathrm{Sim}PE(b_2^{next},b_2'^{next})$。虚拟路径的图形语义相似性可视为这三组的语义相

似性加权之和，因此元素 b_2 和 b_2' 的虚拟路径的图形语义相似性为

$$SimC(b_2,b_2')=W_p*SimEP(b_2^{pre},b_2'^{pre})+W_e*SimE(b_2,b_2')+W_n*SimPE(b_2^{next},b_2'^{next})$$

其中，$SimEP(b_2^{pre},b'^{pre})=SimE(b_1*P(b_1),b_1'*P(b_1'))$

$=SimE(b_1,b_1')*SimE(P(b_1),P(b_1'))$

$SimPE(b_2^{next},b_2'^{next})=SimE(P(b_2)*b_3,P(b_2')*b_3')=SimE(P(b_2),P(b_2'))*SimE(b_3,b_3')$

W_p 表示前驱元素及其属性的语义相似性分配的权重比例，W_e 表示元素的独立语义相似性分配的权重比例，W_n 表示属性与后驱元素的语义相似性分配的权重比例。这 3 个权重之和为 1，即 $W_p+W_e+W_n=1$。

3.4 匹配关系推导

取得虚拟路径的图形语义相似性比较结果 S 后，即可根据结果推导两个目标元素之间的映射关系。本文采用国际上广泛采用的映射关系分类方法，将映射关系类型分为相等关系、类属关系、部分整体关系、不相交关系；如果语义相似性比较结果 $S=1$，则两个元素为相等关系；如果 $0.8\leqslant S<1$，则两个元素为类属关系；如果 $0.5\leqslant S<0.8$，则两个元素为部分整体关系；如果 $S<0.5$，则两个元素为不相交关系。

4 实验与分析

为了评价基于上下文的元素层次语义匹配算法的有效性，本文从质量(Quality)和性能(Performance)两个方面对基于上下文的元素层次语义匹配算法进行评价[17]。质量评价即评价算法是否完成了功能或者逻辑上的目标，性能评价的主要指标是时间与空间的开销。

4.1 质量评价测试

质量评价测试最著名的标准来源于信息检索领域的查准率(Precision)和查全率(Recall)[18]两个指标，目前本体匹配研究广泛采用的方法(国际本体对齐评价组织 OAEI)[1]是将查准率和查全率应用于本体匹配的算法评价[10,19]。如图 3 所示，如果将手动匹配的结果看成是评价自动匹配结果的标准，也就是将手动匹配看成一种非常理想的情况，它能够发现所有配对的关系，并且正确地测量元素关系，那么手动匹配的结果就可能包含两个部分 A 和 B。A 表示手动匹配结果正确而自动匹配不能发现的结果数量，B 表示手动和自动匹配都能够正确计算的结果数量。而自动匹配也由两个部分的结果 B 和 C 组成，C 表示手动匹配计算错误的结果的数量。由此可以得出参与自动匹配的元素数据集的数量为 $B+C$，而手动匹配结果的数量，即正确匹配结果的数量为 $A+B$。

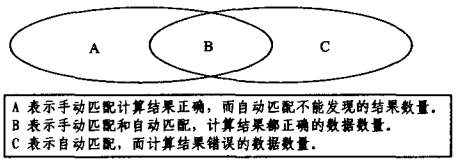

图 3 自动匹配与手动匹配结果的比较

查准率表示自动匹配计算正确结果的数量占所有自动匹配发现结果的数量的比例。

$$Precison=\frac{B}{B+C} \tag{1}$$

查全率表示自动匹配计算正确结果的数量占所有计算正确结果的数量的比例。

$$Recall=\frac{B}{A+B} \tag{2}$$

在理想状况下，Precison=Recall=1。单独使用查准率或查全率都不能准确地评价算法的质量。而且很多情况下，查准率和查全率的结果并不一致，即有可能出现的情况是查准率的结果很大(自动匹配发现的数据很少，自动匹配结果都正确)，而查全率的结果却很小(很多数据都没被自动匹配发现)；或者是查全率的结果很大，而查准率结果很小。因此，有必要将查全率和查准率综合起来评价算法质量。有几种方法综合了查准率和查全率的计算结果[7,10,19,20]，如 OAEI 采用的 F-Measure 和 OverAll。

$$F\text{-}Measure=\frac{Precison*Recall}{(1-w)*Precison+w*Recall} \tag{3}$$

式中，w 表示给查全率和查准率分配的权重，其意义在于它决定了查全率和查准率在综合测量结果 F-Measure 中的重要程度，w 的取值范围是[0,1]。当 $w->0$ 时，表示查全率在 F-Measure 中可以忽略不计，因此 F-Measure 等于查准率的结果。当 $w->1$ 时，表示查准率在 F-Measure 中可以忽略不计，因此 F-Measure 等于查全率的结果。一般来说，设置查准率和查全率具有同等重要性，即 $w=0.5$。这样式(3)可以表示为

$$F\text{-}Measure=2*\frac{Precison*Recall}{Precison+Recall} \tag{4}$$

SF(Similarity Flooding)[5]匹配系统采用 OverAll 表示查全率和查准率的综合测量结果。OverAll 的定义如式(5)所示。和 F-Measure 不同，OverAll 指标用来表示 B 和 C 之间的差异，即自动匹配中正确结果的数量和自动匹配发现但计算结果错误的数量之差。

$$OverAll=\frac{B-C}{A+B}=Recall*(2-\frac{1}{precision}) \tag{5}$$

本文根据本体匹配标准的测试数据集(OAEI 2006 BenchMark[2])和 S-Match[10,11]所采用的数据集[3]来测试基于上下文元素的匹配算法。对于 OAEI 2006 数据集，本文选择属于学习资源本体范畴的参考文献本体(Reference Ontology)＃101－＃104，＃201－＃204，＃302 一共 3 组、9 对数据集作为算法的实验数据。而对 S-Match 的数据集，本文选择其中的＃S7，即大学课程本体作为另一个实验数据。这 4 组数据、10 个本体都是与学习资源相关的本体。匹配测试数据的具体描述如表 1 所列。

表 1 本体匹配实验测试数据集

序号	匹配测试任务
＃101	Reference Ontology Vs Reference Ontology
＃102	Reference Ontology Vs Reference Ontology
＃103	Reference Ontology Vs Reference Ontology(Generalization)
＃104	Reference Ontology Vs Reference Ontology (Language restriction)

1) http://oaei.ontologymatching.org/2006/

2) http://oaei.ontologymatching.org/2006/benchmarks/

3) http://dit.unitn.it/~accord/Experimentaldesign.html

#201	Reference Ontology Vs Reference Ontology (No names)
#202	Reference Ontology Vs Reference Ontology (No names, no comments)
#203	Reference Ontology Vs Reference Ontology (Misspelling)
#204	Reference Ontology Vs Reference Ontology (Naming conventions)
#302	Real ontology: Reference Ontology Vs UMBC Ontology
#S7	Cornell Course Ontology Vs Washington Course Ontology

本文对这4组、一共10个本体进行了查准率、查全率、F-Measure和OverAll 4个指标的测试。图4显示了#101参考文献本体与自身匹配、#201参考文献本体与没有标签的参考文献本体匹配、#302参考文献本体与UMBC(美国马里兰大学)[4]的本体进行匹配以及卡耐尔大学课程本体与华盛顿大学课程本体进行匹配的查准率、查全率、F-Measure和OverAll 4个指标的数据。

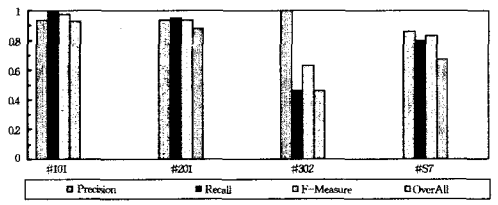

图4 测试数据集的4个指标

数据显示这4组本体匹配都能得到比较高的查全率、查准率。这说明,本文的匹配算法对同一领域本体以及本体规模不太大的情况都能取得比较好的匹配质量。由于查全率和查准率都只能从一个方面评价算法质量,F-Measure能综合查全率和查准率的结果,因此能全面地反映匹配算法的质量。数据集中所有本体对匹配结果的F-Measure如图5所示。结果表明,10个匹配对的F-Measure的平均值比较高,说明算法的综合评价指标还是不错的。

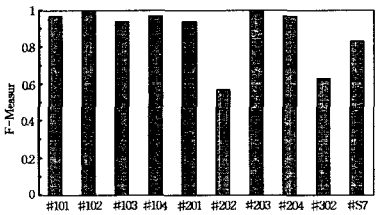

图5 测试数据集的综合测量指标F-Measure

4.2 性能评价测试

元素层次的语义匹配涉及到两个本体中所有的元素以及元素上下文的比较,因此算法的时间开销是一个需要考虑的因素,它受本体的元素数量和所采用算法的影响。性能评价测试的目标是分析算法的时间开销,它反映了匹配算法的计算效率。为了客观地评价算法性能,本文将测试的计算机硬件参数设置如下:Pentium4 2.8GHz,512RAM,Windows XP,并且除了匹配程序外,没有其他任何程序运行。

通过元素层次的匹配算法对10组数据集进行匹配实验,发现算法程序运行的时间开销都在20s以内,因此可以证明匹配算法的时间开销还是比较理想的。其中,影响算法开销时间的主要参数是两个比较本体的元素数量$E1$和$E2$、两个本体中属性的数量$P1$和$P2$。在外部条件相同的情况下,不同元素数量的本体匹配算法的计算时间稍有不同。随着本体元素数量的增大,匹配算法的计算时间会略有增加。

结束语 本文提出一种本体匹配方法,包括以下步骤:将待比较的两个本体概念定为目标元素,为两个目标元素分别建立由具有语义联系的相邻元素及其联系所构成的虚拟路径,即前驱元素—前驱元素与目标元素之间的属性或者语义关系—目标元素—目标元素与后躯元素之间的属性或者语义关系—后驱元素;将两个目标元素的虚拟路径中各独立要素分别对应进行概念语义相似性比较;综合虚拟路径内各独立要素的概念语义相似性,获取两个目标元素虚拟路径的图形语义相似性;根据虚拟路径的图形语义相似性推导两个目标元素之间的映射关系。此本体概念匹配方法侧重于概念的相邻元素及其语义联系,考察范围更为全面。实验结果表明,本方法提高了本体匹配的质量和性能。

[4] http://swoogle.umbc.edu/

参考文献

[1] Shvaiko P, Euzenat J. A Survey of Schema-based Matching Approaches [J]. Journal on Data Semantics, LNCS 3730, 2005: 146-171

[2] Rahm E, Bernstein P A. A Survey of Approaches to Automatic Schema Matching[J]. VLDB Journal, 2001(10): 334-350

[3] Madhavan J, Bernstein P, Rahm E. Generic schema matching using Cupid[C]//Proceedings of 27th International Conf. on Very Large Data Bases (VLDB'01). Rome, Italy, 2001: 49-58

[4] Do H H, Rahm E. COMA: A system for flexible combination of schema matching approaches[C]//Proceedings of 28th International Conference on Very Large Databases (VLDB). Hong Kong, 2002: 610-621

[5] Melnik S, Molina H G, Rahm E. Similarity flooding: A versatile graph matching algorithm[C]//Proceedings of Eighteenth International Conference on Data Engineering. San Jose, California, February 2002: 117-128

[6] Jian Ningsheng, Hu Wei, Cheng Gong, et al. Falcon-AO: Aligning Ontologies with Falcon[C]//K-Cap 2005 Workshop on Integrating Ontologies. Canada, October 2005

[7] Hu W, Jian N S, Qu Y Z, et al. GMO: A Graph Matching for Ontologies[C]//K-Cap 2005 Workshop on Integrating Ontologies. 2005: 43-50

[8] Castano S, Ferrara A, Montanelli S. Matching Ontologies in Open Networked Systems: Techniques and Applications [EB/OL]. http://islab.dico.unimi.it/hmatch/downloads.php? cat_id=2, 2007-9-20

[9] Castano S, Ferrara A, Montanelli S. H-MATCH: an Algorithm for Dynamically Matching Ontologies in Peer-based Systems [EB/OL]. http://islab.dico.unimi.it/hmatch/downloads.php? cat_id=2, 2007-9-20

[10] Shvaiko P. Iterative Schema-based Semantic Matching [D]. taly: International Doctorate School in Information and Communication Technology (ICT), University of Trento, 2006

[11] Giunchiglia F, Yatskevich M, Shvaiko P. Semantic Matching: Algorithms and Implementation [J]. Journal on Data Semantics (JoDS), IX, LNCS 4601, 2007: 1-38

[12] Doan A, Madhavan J, Domingos P, et al. Learning to Map between Ontologies on the Semantic Web[C]//Proceedings of the World Wide Web Conference (WWW). 2002

[13] Madhavan J, Bernstein P, Doan An-Hai, et al. Corpus-based schema matching[C]//Proceedings of the International Conference on Data Engineering (ICDE). 2005: 57-68

[14] Zhong Jiwei, Zhu Haiping, Li Jianming, et al. Conceptual Graph Matching for Semantic Search[C]//Proceedings of the 10th International Conference on Conceptual Structures (ICCS 2002).

LNCS 2393, Borovets, Bulgaria, July 2002: 15-19
[15] 袁洋,李善平. 基于语义 Web 的本体映射方法综述[J]. 计算机科学,2004,31(5):5-8
[16] 曹泽文,钱杰,张维明,等. 一种综合的概念相似度计算方法[J]. 计算机科学,2007,34(03):174-191
[17] Castano S, Ferrara A, Montanelli S. Matching Ontologies in Open Networked Systems: Techniques and Applications [EB/OL]. http://islab.dico.unimi.it/hmatch/downloads.php?cat_id=2,2007-9-20
[18] Joost C, van Rijsbergen K. Information retrieval[EB/OL]. http://www.dcs.gla.ac.uk/Keith/Preface.html,2007-9-21
[19] Hong-Hai D, Sergey M, Erhard R. Comparison of Schema Matching Evaluations [J]. Lecture Notes in Computer Science, 2003 (2593): 221-237
[20] Avesani P, Giunchiglia F, Yatskevich M. A large scale taxonomy mapping evaluation[C]// Proceedings of the International Semantic Web Conference (ISWC). 2005: 67-81
[21] Hirst G, St-Onge D. Lexical Chains as representations of context for the detection and correction of malapropisms[M]. Fellbaum, 1998: 305-332

(上接第 174 页)

足性检测算法,则节点的选择由在约束类型表达式 T_1 下对 e_1 的节点选择和在约束类型表达式 $\cdot T_2$ 下对 e_2 的节点选择共同决定)

• XPath 中的空集决策问题:$E^{\rightarrow}||e_1||_{||T_1||}$

• XPath 中的重叠决策问题:$E^{\rightarrow}||e_1||_{||T_1||} \wedge E^{\rightarrow}||e_2||_{||T_2||}$

• XPath 中的覆盖决策问题:$E^{\rightarrow}||e_1||_{||T_1||} \wedge \wedge_{2\leqslant i\leqslant n} \neg E^{\rightarrow}||e_i||_{||T_i||}$

可满足性检测算法还可以用来对 XML 类型检查中的两类问题进行处理。

• 带注释的 XPath 查询的静态类型检查

$E^{\rightarrow}||e_1||_{||T_1||} \wedge \neg||T_2||$(如果表达式不满足可满足性检测算法,则所有的节点都来自于包含于约束类型表达式 T_2 下的 T_1 相对于对 e_1 的节点选择)

• 类型约束下的 XPath 等价转化

$E^{\rightarrow}||e_1||_{||T_1||} \wedge \neg E^{\rightarrow}||e_2||_{||T_2||}$ 和 $E^{\rightarrow}||e_1||_{||T_1||} \wedge E^{\rightarrow}||e_2||_{||T_2||}$(这个测试可以用于动态地检查那些经过 T_2 修改后的 T_1 以及用同样的方式经过 e_2 修改过的 e_1 得到的节点,在查询过程中经常会遇到当一个输入类型发生改变时相应的 XPath 查询也必须同时改变的这种情况)。

现有的研究中没有出现过应用定位逆向轴进行递归操作的实例,因此我们简单地提供证据来证明我们的方法是有效的。我们进行的扩展性实验[7],所展示的仅仅是具有代表性的示例,包括许多复杂的语言特征,例如递归的前向和后向轴、交集和带字母表的递归形式。实验使用到的 XML 数据的类型表示如表 1 所列,XPath 表达式如表 2 所列(其中"//"是"/desc-or-self::*/"是缩写),表 3 描述了一些决策问题以及相关的实验结果。运行时间以毫秒为单位。

表 1 实验中使用到的数据类型

DTD	Symbols	Binary Type Variable
SMIL 1.0	19	11
XHTML1.0 Strict	77	325

第一个包含实例的 XPath 首先在文献[8]中作为示例来计算,假定树模式的同构技术并不完全。e_8 的示例说明官方的 XHTML DTD 并没有禁止语义上的嵌套。对于一个 XHTML 的示例,我们观察到所需的时间是非常重要的,但是仍与实际相关,尤其对于那些在编译时间执行的静态分析操作。

表 2 实验中使用到的 XPath 表达式

e1: /a[.//b[c/*//d]/b[c//d]/b[c/d]]
e2: /a[.//b[c/*//d]/b[c/d]]
e3: a/b//c/foll-sibling::d/e
e4: a/b//d[prec-sibling::c]/e
e5: a/c/following::d/e
e6: a/b[//c]/following::d/e∩a/d[prec-sibling::video]
e7: *//switch[ancestor::head]//seq//audio[pre-sibling::video]
e8: descendant::a[ancestor::a]
e9: / descendant::*
e10: html/(head|body)
e11: html/head/descendant::*
e12: html/body/descendant::*

表 3 一些决策问题以及相关的实验结果

XPath 决策问题	XML 类型	时间(ms)
$e_1 \subseteq e_2$ 且 $e_2 \not\subseteq e_1$	none	353
$e_3 \subseteq e_4$ 且 $e_4 \subseteq e_3$	none	45
$e_6 \subseteq e_5$ 且 $e_5 \not\subseteq e_6$	none	41
e_7 是可满意的	SMIL1.0	157
e_8 是可满意的	XHTML1.0	2630
$e_9 \subseteq (e_{10} \cup e_{11} \cup e_{12})$	XHTML1.0	2872

结束语 本文的主要工作是提出了一个正确的、完备的涉及正则树类型的决策问题的可满足性检测算法。我们的算法以有限树的子逻辑为基础,算法的证明方法显示了这些逻辑规则和 XPath 决策问题之间的联系。首先,XML 正则树类型和 XPath 片段之间的转化是无环、线性表达式大小的;其次,在一个有限树中,在否定情况下带有一个驻点的操作符的逻辑是闭合的,这就使得我们可以解决关键的 XPath 决策问题。如何将对于 XPath 类型的决策问题的研究扩展到对于数值比较的决策问题的研究将是下一步的研究方向。

参 考 文 献

[1] Huet G P. Functional Pearl the Zipper[J]. J. Functional rogramming, 1997, 7(5): 549-554
[2] Hosoya H, Vouillon J, Pierce B C. Regular expression types for XML[J]. ACM Trans. Program. Lang. Syst., 2005, 27(1): 46-90
[3] Pan G, Sattler U, Vardi M Y. BDD-based decision proceduresfor the modal logic K[J]. Journal of Applied Non-classical Logics, 2006, 16(1/2): 169-208
[4] Benedikt M, Fan W, Geerts F. XPath satisfiability in the presence of DTDs[C]// PODS '05: Proceedings of the twenty-fourth ACM Symposium on Principles of Database Systems. 2005: 25-36
[5] Genevès P, Layaïda N. Deciding XPath containement with MSO [J]. Data & Knowledge Engineering, 2007, 63(1): 108-136
[6] Mø ller A, Schwartzbach M I. The design space of type checkers for XML transformation languages[C]// Proc. Tenth International Conference on Database Theory, ICDT '05, volume 3363 of LNCS. 2005(1): 17-36
[7] Genevès P, Layaïda N, Schmitt A. A satisfiability solver for XML and XPath[J]. Journal of the ACM, 2007, 6(42): 342-351
[8] Miklau G, Suciu D. Containment and equivalence for a fragment of XPath[J]. Journal of the ACM, 2004, 51(1): 2-45