文章编号:1671 - 6833(2015)03 - 0106 - 04

一种基于多种类型匹配器的本体映射方法

张凌宇[1], 马志晟[2], 陈淑鑫[1]

(1. 齐齐哈尔大学 计算中心,黑龙江 齐齐哈尔 161006;2. 齐齐哈尔大学 教务处,黑龙江 齐齐哈尔 161006)

摘 要: 不同本体之间的异构性严重地影响了本体之间的知识共享与重用,为此,提出一种基于多种类型匹配器的本体映射方法 OM-Matchers (Ontology mapping based on multiple matchers). 在建立本体之间映射关系的过程中,OM-Matchers 先使用多个类型的匹配器从本体模型中抽取相应类型的信息;然后这些匹配器为概念对计算相似度值,其中概念对所包含的两个概念来自于不同的本体;最后为待映射的本体模型建立相似度矩阵,并采用迭代策略完成本体映射任务. 为了验证本文所提方法在处理本体映射问题时的可行性与有效性,采用 OAEI 所提供的共享数据集的 benchmarks 子集来测试 OM-Matchers. 实验结果表明:OM-Matchers 可以有效地建立异构本体之间的映射关系.

关键词: 本体;本体映射;匹配器;迭代策略

中图分类号: TG335.58 **文献标志码:** A doi:10.3969/j.issn.1671 - 6833.2015.03.023

0 引言

本体模型[1]作为一种明确的、共享的概念模型,可以为语义 Web[2]领域提供形式化的规范说明. 然而,不同的本体构建者可能采取不同的方法和不同观点,建立可以满足具体应用需求的本体模型,这样必然会造成本体之间的语义冲突和结构异构等问题. 因此,需要使用本体映射方法[3]来解决异构本体之间知识共享、重用以及语义查询等互操作问题.

目前,国内外的很多研究者都在从事于本体映射方法的研究. 采用不同的数据模型和技术方法来完善本体映射方法的性能已经成为语义 Web 领域内的一个热点研究课题. 为此,笔者提出一种基于多种类型匹配器的本体映射方法(OM-Matchers). 为了提高本体映射的精确度,OM-Matchers 采用名称匹配器、内容匹配器、属性匹配器、实例匹配器和结构匹配器,分别计算概念之间不同类型信息之间的相似度. 然后,OM-Matchers 根据不同类型信息占总信息量的比重为这些匹配器分配权值,并为概念对计算一个最终的相似度值. 最后,OM-Matchers 采用迭代的映射算法为相似度大于给定阈值的概念对建立映射关系. 需要说明的是,迭代映射算法将反复地执行相似度计算和映射筛选的步骤,直到算法找不出新的映射关系为止.

1 相关工作

为了提高概念相似度计算的精确度,很多本体映射方法对概念的不同方面进行了相似性的比较. 例如,Cupid[4]在映射过程中对概念的四种信息(名称信息、数据类型信息、约束信息以及元素所在的子结构信息)进行相似性比较. GLUE[5]在计算概念之间不同方面的相似度时,主要比较概念的名称、标识信息和实例之间的差异性. RiMOM[6]在计算概念相似度时,提出了多种决策:基于名称的决策、基于实例的决策和基于描述信息的决策、基于上下文的决策和基于约束的决策. ASMOV[7]使用概念的语言信息、内部结构信息、外部结构信息和个体信息,计算概念之间的相似度. MSBN[8]是一种基于多策略和贝叶斯网络的本体映射方法,它使用概念名称的编辑距离、概念的描述信息和实例特征,计算概念之间的相似度,最后使用本体的结构信息来辅助映射的查找.

收稿日期:2015 - 01 - 24;**修订日期:**2015 - 03 - 10

基金项目:国家自然科学基金资助项目(61204127);中国博士后科学基金面上项目(2012M510898);黑龙江省自然科学基金资助项目(F030503,F201336).

作者简介:张凌宇(1981 -),男,河北省蠡县,齐齐哈尔大学讲师,博士,研究方向为语义 Web、(模糊)本体映射、(模糊)本体集成,E-mail:zhanglingyu00217@126.com.

在一些经典的本体映射框架中,概念相似度计算是一个重要的步骤. 例如: QOM[9]是一种快速本体映射框架,它的核心由相似度计算与合并模块、建立映射模块和迭代控制模块组成. 为了大幅度地提高映射效率,它的相似度计算模块和建立映射模块提供了人工监控机制,使得运行时间复杂度从原来的 $O(n^2)$ 降低为 $O(n \cdot \lg(n))$. MAFRA[10]利用多种相似度计算方法来建立语义桥(Semantic bridge),再配合其他的功能模块形成分布式本体映射框架.

2　多种类型匹配器

匹配器(matcher)是计算概念相似度的基本单元,它可以解析本体文本文件 OWL,并为计算概念相似度抽取可以处理的信息. 笔者总结出以下 5 种匹配器,并给出它们的处理对象、工作原理以及在本体映射过程中的作用.

(1)词法匹配器. 词法匹配器也可细分成:名称匹配器和内容匹配器. 名称匹配器可以计算不同概念名称之间的相似度;内容匹配器可以计算不同概念标签以及描述信息之间的相似度. 下面的两个公式分别给出名称匹配器和内容匹配器的计算方法. 其中:C_1 和 C_2 表示两个概念,S 表示它们最近公共父节点,函数 $W()$ 返回概念内容所包含的词集合,函数 size( ) 返回集合所包含元素的数量.

$$\mathrm{Sim}N(C_1,C_2)=\frac{2\lg P(S)}{\lg P(C_1)+\lg P(C_2)}. \quad (1)$$

$$\mathrm{Sim}C(C_1,C_2)=1-\frac{W(C_1)\cap W(C_2)}{\mathrm{size}(W(C_1))+\mathrm{size}(W(C_2))}. \quad (2)$$

(2)属性匹配器. 属性由定义域和值域构成,它是定义概念内在含义的基本元素. 为了计算概念的属性相似度,属性匹配器首先会分析属性名称的语义. 如果属性名称不是由简单的标记符号组成,属性匹配器将利用语义词典库来计算属性名称之间的相似度,这个计算过程与名称匹配器相似. 如果属性的名称是由简单的助记符号构成,属性匹配器可以具体地分析属性定义域之间的相似度和属性值域之间的相似度,最后计算出属性之间的相似度. 计算公式如公式(3)所示,其中 函数 $D()$ 和 $R()$ 分别返回属性的定义域和值域,如果属性的定义域(值域)相同函数 Sim( ) 返回 1,否则返回 0.

$$\mathrm{Sim}P(P_1,P_2)=\begin{cases}\mathrm{Sim}N(P_1,P_2), \text{if } P_1 \text{ and } P_2 \text{ are found in WordNet;}\\ (\mathrm{Sim}(D(P_1),D(P_2))+\mathrm{Sim}(R(P_1),R(P_2)))/2, \text{otherwise.}\end{cases} \quad (3)$$

(3)实例匹配器. 实例匹配器可以采用概念的联合概率分布(joint distribution)来计算概念之间的相似度,也就是说实例匹配器采用 Jaccard 系数计算概念之间的相似度,如下所示.

$$\mathrm{Sim}I(x,y)=\frac{P(x\cap y)}{P(x\cup y)}=\frac{P(x,y)}{P(x,y)+P(x,\bar{y})+P(\bar{x},y)}. \quad (4)$$

式中:x 和 y 表示两个概念节点;$P(x, y)$ 表示同时属于 x 和 y 的实例占总实例的比例. 分母表示 x 和 y 包含的所有实例占总实例的比例.

(4)结构匹配器. 结构匹配器在计算概念之间相似度时,将找出概念的父类概念和子类概念,然后综合考虑它们之间的相似性. 概念的父类概念和子类概念可以被组成一个概念集合,这个集合可以约束概念的语义范围,因此可被称之为概念的上下文(context)集合. 这样,结构匹配器可以采用 Jaccard 系数来计算概念在结构方面的相似性.

(5)权值分析器. 权值分析器为匹配器所分配的权值将由本体所包含的具体信息而定. 例如,假设待映射的本体属于上层的抽象本体,它们不包含任何的实例信息,那么在映射过程中,权值分析器将实例匹配器的权值设置成 0. 在一般情况下,权值统计出各类信息占总信息量的比例,分析出各类信息的重要程度并为相应的匹配器赋予适当的权值. 例如,在本体模型 benchmark 中(详见第四节),统计出描述概念的名称、内容、属性、实例和结构信息的数目分别为 32,27,65,112,267. 那么,上述 5 种匹配器的权值分别为:0. 06,0. 05,0. 13,0. 22,0. 53.

利用各个匹配器的计算结果和它们相应的权值,可以计算出概念之间的相似度,如下所示:

$$\mathrm{Sim}(C_1,C_2)=\lambda_N\mathrm{Sim}N+\lambda_C\mathrm{Sim}C+\lambda_P\mathrm{Sim}P+\lambda_I\mathrm{Sim}I+\lambda_S\mathrm{Sim}S. \quad (5)$$

3　基于多类型匹配器的本体映射方法

笔者在前面匹配器的基础上提出一种迭代策略的本体映射方法:OM-Matchers (Ontology Mapping based on multi-Matchers). 映射过程如图 1 所示.

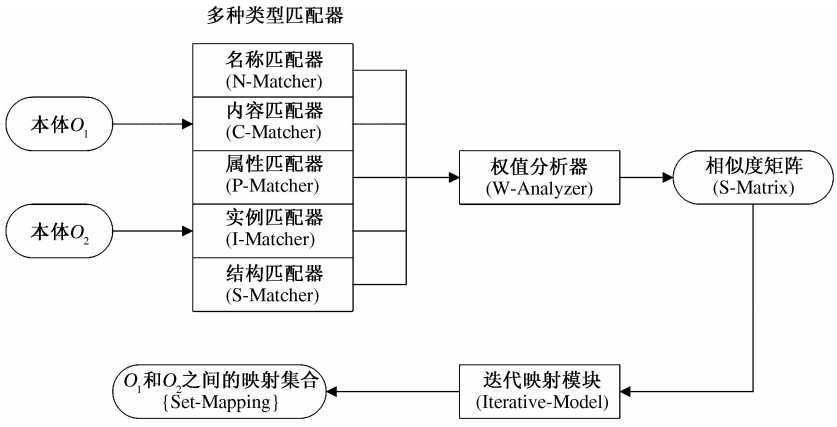

图 1 OM-Matchers 的映射过程

Fig. 1 The mapping process of OM-Matchers

OM-Matchers 以 2 个本体模型 O_1 和 O_2 作为输入;然后将本体内的信息分类,并将不同类型的信息发送给相应的匹配器;然后,匹配器为概念计算多个相似度值;权值分析器再根据各种类型信息在本体映射过程中所起到的作用,为匹配器指定权值;利用匹配器的计算结果和权值,OM-Matchers 为本体 O_1 和 O_2 生成相似度矩阵,其中 O_1 和 O_2 所包含的概念分别用于标识矩阵的行和列;最后,采用迭代的映射策略反复地更新相似度矩阵,当矩阵中的元素大于给定的阈值时,为行标识和列标识所对应的概念建立映射关系,存储于映射结果集合. 经过多次迭代映射过程后,相似度矩阵中的元素将收敛于一个固定值. 这时,映射过程结束,方法 OM-Matchers 返回映射结果集合.

假设,本体 O_1 和 O_2 的概念集合分别是 $\{C_1, C_2, \cdots, C_m\}$ 和 $\{C_1', C_2', \cdots, C_n'\}$,概念之间的关系如图 2 所示. 方法 OM-Matchers 的迭代映射过程可分成以下几步来实现.

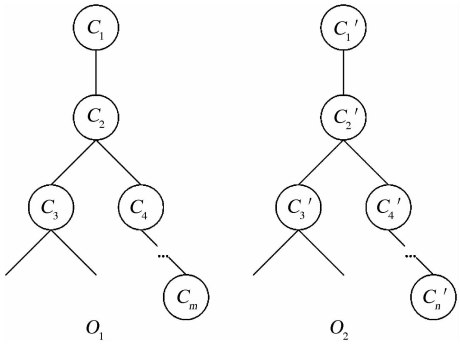

图 2 本体 O_1 和 O_2 的结构

Fig. 2 The structure of ontology O_1 and O_2

(1)生成待映射概念对的堆栈(Stack). 利用本体映射系统预先设定的阈值(t: threshold),从相似度矩阵中筛选出相似度大于 t 的概念对. 如果有多个概念对的相似度值大于阈值 t,还需要使用堆栈来暂时存储概念对. 概念对进栈的顺序由概念对中概念的层次决定.

(2)建立映射关系并生成邻近概念集合{Set-Near}. 位于 Stack 最低层的概念对出栈,建立概念之间映射关系,并将映射的概念对存储于映射关系集合{Set-Mapping}. 例如:概念 A 与 B 之间的映射关系为 $A \leftrightarrow B$. 接下来,分别为概念 A 和 B 查找与它们直接相关的概念. 然后,建立邻近概念集合 $\{\text{Set-Near}\}_A$ 和 $\{\text{Set-Near}\}_B$. 由于概念 A 与 B 之间已经建立映射关系,可以断定概念 A 与 B 是等价的. 基于相似度传播原理可知:概念 A 周围的概念与概念 B 周围的概念也可能存在映射关系. 因此,为已建立映射关系的概念查找出它们的邻近概念集合将有助于接下来的映射过程.

(3)更新相似度矩阵. 使用步骤 2 得到两个邻近概念集合生成概念对. 然后,再根据这些概念对,从相似度矩阵中找出相应的相似度值. 使用下面的公式(6)来修改这些相似度值,从而得到更新后的相似度矩阵. 在公式(6)中,$C_i \in \{\text{Set-Near}\}_A$, $C_j \in \{\text{Set-Near}\}_B$.

$$\text{Sim}(C_i, C_j') = \text{Sim}(C_i, C_j') \cdot (\text{Sim}(A, B)/t). \quad (6)$$

(4)返回步骤 1 或者映射过程结束. 方法 OM-Matchers 将反复地执行步骤 1 到步骤 3. 如果相似度矩阵中的所有数据都收敛,即每次更新相似度矩阵时,所有数据的变化小于给定的阈值($t < 0.000\ 1$),迭代映射过程结束并返回映射集合{Set-Mapping}作为方法 OM-Matchers 的运行结果.

$$\begin{array}{cccc} \text{Sim}(C_1,C_1') & \text{Sim}(C_1,C_2') & \cdots & \text{Sim}(C_1,C_n') \\ \text{Sim}(C_2,C_1') & \text{Sim}(C_2,C_2') & \cdots & \text{Sim}(C_2,C_n') \\ \cdots & \cdots & \cdots & \cdots \\ \text{Sim}(C_m,C_1') & \text{Sim}(C_m,C_2') & \cdots & \text{Sim}(C_m,C_n') \end{array}$$

图 3 O_1 和 O_2 的相似度矩阵图

Fig. 3 The similarity matrix for O_1 and O_2

4　实验分析

在实验过程中,采用信息检索的标准度量方法:查全率(Precision)、查准率(Recall)和 F 参数(F-Measure),来衡量方法 SM-Context 的性能.

为了验证方法 OM-Matchers 的映射性能,使用 OAEI (Ontology Alignment Evaluation Initiative)所提供的数据集 benchmarks 中的部分数据作为测试数据集. 数据集 benchmarks 共包含了 51 个本体,其中本体#101 为参考本体(Reference ontology),包含 32 个概念、65 个属性和 112 个实例. 本体#102 中的信息与参考本体#101 完全不相关,其他本体都是在参考本体的基础上增加、修改或者删除部分语义信息而得到的. 在实验过程中,方法 OM-Matchers 将分别建立参考本体与这些本体之间的映射关系. 映射结果的查全率(R: Recall)、查准率(P: Precision)和 F 系数(F: F-Measure),如表 1 所示.

表 1　OM-Matchers 的查准率、查全率和 F 系数

Tab. 1　The precise, recall and F-Measure of OM-Matchers

参数	#101	#103	#104	#201	…	#266	平均值
P	1.00	1.00	1.00	0.91	…	0.72	0.90
R	1.00	1.00	1.00	0.92	…	0.73	0.91
F	1.00	1.00	1.00	0.91	…	0.73	0.90

图 4 给出了方法 OM-Matchers 和其他几种经典本体映射方法的测试结果. 这些方法都是采用了 OAEI 中的 benchmark 数据集作为实验对象. 实验结果表明:方法 OM-Matchers 可以有效地利用本体所包含多种类型的信息,精确地计算概念之间的相似度. 而且,方法 OM-Matchers 所采用的迭代映射策略可以反复地利用多种类型的匹配器来计算概念之间的相似度,从而提高了映射的查全率、查准率和 F 系数.

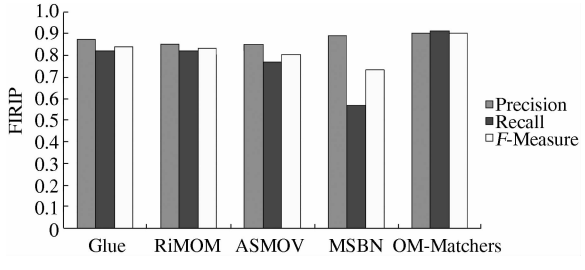

图 4　对比实验结果

Fig. 4　The result of contrast test

5　结论

提出了一种基于多种类型匹配器的本体映射方法(OM-Matchers). 在建立两个本体之间映射关系的过程中,该方法利用 5 种匹配器(名称匹配器、内容匹配器、属性匹配器、实例匹配器、结构匹配器),统计不同类型信息的重要程度(即:权值)并计算出概念之间的相似度. 根据计算结果,OM-Matchers 采用迭代的映射策略,建立本体之间的映射关系. 实验结果表明:多种类型匹配器和迭代策略的使用,可以提高 OM-Matchers 映射的性能参数(查全率、查准率、F 系数).

在接下来的研究工作中,还需要针对不同类型的知识库来设计出更多种类的匹配器. 另外,还需要为 OM-Matchers 设计用户界面,以提高该方法的交互能力. 这些研究工作将提高 OM-Matchers 的综合处理能力.

参考文献:

[1] HAASE P, HORROCKS I, HOVLAND D, et al. Optique system: towards ontology and mapping management in OBDA solutions [C]//Proceedings of the Second International Workshop on Debugging Ontologies and Ontology Mappings-WoDOOM13. Berline: Springer, 2013:21 – 32,.

[2] LANGE C. Ontologies and languages for representing mathematical knowledge on the semantic web [J]. Semantic Web, 2013, 4(2):119 – 158.

[3] SHVAIKO P, EUZENAT J. Ontology matching: state of the art and future challenges [J]. IEEE Transactions on, Knowledge and Data Engineering, 2013, 25(1):158 – 176.

[4] MADHAVAN J, BERNSTEIN P, RAHM E. Generic schema matching with cupid [C]//proceedings of the International Conference on Very Large Databases (VLDB). Berlin: Springer, 2001:49 – 58.

[5] DOAN A, MADHAVAN J, DOMINGOS P, et al. Learning to map between ontologies on the semantic web [C]//Proceedings of the Eleventh International World Wide Web Conference. New York:ACM, 2002: 662 – 673.

[6] LI J, TANG J, LI Y, et al. RiMOM: A dynamic multi-strategy ontology alignment framework [J]. Transaction on Knowledge and Data Engineering, 2009,21 (8):1218 – 1232.

[7] JEAN-MARY Y, KABUKA M. ASMOV Results for OAEI 2007 [C]//Proceedings of International Semantic Web Conference 2007 Ontology Matching Workshop. Busan: Citeseer, 2007:150 – 159.

[8] 张凌宇, 马宗民. 一种基于贝叶斯网络模型及多策略计算的本体映射方法 [J]. 小型微型计算机系统, 2011, 33(11): 2385 – 2391.

(下转第 119 页)