基于模式结构分类的本体映射方法

周 栩,刘 磊,范任宏
(吉林大学计算机科学与技术学院,吉林长春 130012)

摘 要: 本体映射是语义集成的关键技术,本文基于本体的模式结构提出了一种自顶向下的本体映射方法,该方法在考虑四种基本映射关系并假定所有映射关系均为1:1的情况下,首先将本体描述为图,并将本体中的每个概念都定义为树,在此基础上给出了树中叶节点、非叶节点相似度的计算方法,通过概念分类将子树进一步合并,根据分类的结果重新组织图结构,最后给出了一个完整的本体映射模型.当本体的数目大于2个时,采用概念组分离的方法,即相似的本体在同一组中,不相似的本体在不同的组中,分离直至收敛为止.实验结果表明,这种自顶向下逐层分类的本体映射方法在对大规模结构化本体进行映射发现时可以有效地减少不相关概念之间的计算,在效率和准确度上均取得了理想的效果.

关键词: 语义网;本体映射;分类;相似度计算
中图分类号: TP301 **文献标识码:** A **文章编号:** 0372-2112 (2011) 04-0882-05

A Method of Ontology Mapping Based on Classifying Schema Structure

ZHOU Xu, LIU Lei, FAN Ren-hong
(*College of Computer Science and Technology, Jilin University, Changchun, Jilin* 130012, *China*)

Abstract: Ontology mapping plays a key role in semantic integration area, a top-down ontology mapping method is proposed by this paper, it considering four basic mapping and this mapping process is assumed that all cases are 1:1. This approach describes the ontology as a graph and each ontology concept defined as a tree, on the basis of the tree, give the similarity computation method of leaf nodes and non-leaf nodes, merge the concept sub-tree by concept classification, According to the results of classification re-organization ontology chart structure. Finally, give a complete ontology mapping model. If the number of ontology is greater than two, concept of group separation method will be used, that is, a similar ontology in the same group, no similar in a different group, the separation date until convergence. Experimental results show that the layer classification ontology mapping method can reduce non-related concepts among count for large scale structure ontology mapping for the efficiency and accuracy it achieve the desired results.

Key words: semantic web; ontology mapping; classification; similarity computation

1 引言

近年来,语义网的应用日益受到人们的关注[1].在语义网中,信息是以结构化的形式存在的,而本体则描述语义含义.由于本体的多样性,为了在多个本体之间传递、交互信息,就必须实现语义映射.

目前,此领域已有一些基于模式结构的本体映射方法[2],如基于图的技术[3,4]、基于分类的技术[5,6]等,但是这些方法均是将本体中的所有概念进行统一映射,而本体间存在着许多极不相关的概念,对此类不相关概念的映射计算极大降低了映射算法的性能和准确度.本文提出了一种从模式结构出发的映射方法,根据概念的层次关系对概念进行分类,并自顶向下对概念进行分析.本方法在很大程度上避免对不相关概念间的计算,从而减少了映射过程中的计算量,提高了映射算法的性能和准确性.

2 概念定义

此节将介绍本文涉及的两个关键概念,分别是本体和本体映射.

2.1 本体

本体的相关定义很多,本文采用Gruber所提出的本体定义形式[7].

定义1 本体是概念模型的明确的规范说明,可用五元组表示为 $O=(C,I,R,F,A)$[8].

其中,C 代表概念集合,即抽取出来用来描述事物对象

收稿日期:2008-03-27;修回日期:2010-06-20
基金项目:教育部博士学科点专项科研基金(No.20060183044);国家自然科学基金(No.60873044)

的集合. I 表示概念的实例,代表元素,从语义上讲实例表示的就是对象. R 为定义在概念集合上的关系集合. F 为定义在概念集合上的函数集合. A 表示公理集合,用于约束概念、关系、函数的一阶逻辑谓词集合.

2.2 本体映射

定义 2 本体映射就是找到两个本体之间的语义映射关系,映射函数[9]表示为:

$$Map(\{e_{i_1}\},\{e_{i_2}\},O_1,O_2)=f \tag{1}$$

给定两个本体 O_1 和 O_2,从本体 O_1 到 O_2 的映射是指对于本体 O_1 中的每一个实体,在本体 O_2 中找到一个相对应的实体,并指定它们之间的对应关系. 本体 O_1 叫做源本体,本体 O_2 叫做目标本体. 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). f 可以是一种映射类型(如 *equivalentclass*, *subclass*, *superclass*, *sameindividualas*, *nullunionof*, *disjointwith* 等)或者为 *null*. 当 f 为 *null* 时,表示 $\{e_{i_1}\}$ 和 $\{e_{i_2}\}$ 之间没有对应关系.

本文主要考虑 4 种映射关系: *equivalentclass*, *subclass*, *superclass* 和 *null*,并且假设所有的映射都是 1:1 的映射.

3 本体映射方法

本文将本体描述为图[10,11],将其中的每个概念描述为树. 对任一本体 O,使用函数 $Graph(O)$ 将其进行图形化.

定义 3 图的形式化定义如下:

$$\begin{aligned} Graph &: (Tree)^+ \\ Tree &: Root(node)^* \\ Root &: nood \\ nood &: leaf\,|\,(node)^* \end{aligned}$$

本文映射算法的基本原理主要涉及两部分:一是相似度的计算法则,二是概念的分类原则.

3.1 相似度计算

相似度计算分为两种,一种是叶节点的相似度计算,一种是非叶节点的相似度计算. 对比其他方法常用的绝对相似,本文采用相对相似[3]的概念,即对于实体 e_1 与 e_2, e_1 对 e_2 的相似程度与 e_2 对 e_1 的相似程度是独立的,其相似度值在许多情况下也是不同的. 通过相对相似的比较,可以更准确地确定两个概念间的对应关系.

3.1.1 叶节点的相似度计算

对于叶节点的相似度计算,主要是通过 3 条基本的规则[12]来计算叶节点概念间的相似度:

规则Ⅰ 如果两个节点的标签是相似的,那么它们所代表的概念也很有可能是相似的(R_1).

规则Ⅱ 如果两个概念具有相同的指示符,那么这两个概念是相同的(R_2).

规则Ⅲ 如果显示表明两个概念是相同的,那么它们就是相同的(R_3).

相似度函数 $leafsim(e_1,e_2)$ 根据 3 条规则(R_1, R_2, R_3)进行计算,并将分别应用这 3 条规则得到的结果整合起来,最终得到一个相似度值. 整合的方法有很多,本文采用的是曲线方程[12]的方法. 使用曲线方程,不仅对每条规则得到的值都赋予了权值,并且对这些值本身都进行了计算,最终标准化相似度值在[0…1]范围之内,如图 1 所示.

$$leafsim(e_1,e_2)=\sum_{k=1}^{3} w_k\times sig_k(leafsim(e_1,e_2)-0.5) \tag{2}$$

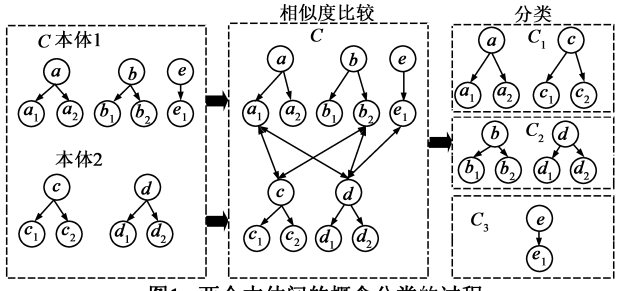

图1 两个本体间的概念分类的过程

定义 4 叶节点的相似度函数定义为:

$leafsim(e_1,e_2)$ 为使用 R_k 得到的结果.

其中 $sig(x)=\dfrac{1}{1+e^{-ax}}$, a 是一个作为斜率的参数.

使用曲线方程的合理性[12]是非常明显的:一个高相似度的值应该被按比例的加重权值,而一个低相似度的值应该相对的被忽略. 而且,使用曲线方程整合结果的一个好处就是便于为相等的实体定义阈值.

当所有叶节点通过相似度函数 $leafsim(e_1,e_2)$ 计算完毕之后,就得到了一个包括所有叶节点间的相似度值的一张表. 接下来的问题就是确定阈值,也就是说,要确定两个叶节点间的相似度值到达何种程度时,才能确定这两个叶节点是相似的. 本文采用的是文献[13]中给出的 *n percent* 方法,这个方法是用最高相似度的值去除一个固定百分比作为阈值.

$$cut-off=\max(leafsim(e_x,e_y)\mid\forall\, e_x\in O_1,e_y\in O_2)(1-p) \tag{3}$$

根据阈值对相似度值表进行筛选,得到叶节点映射矩阵 *leafMatrix*.

3.1.2 非叶节点的相似度计算

对于非叶节点间的相似度计算,本文采用如下思想[13,14]:两个图中的内部节点的相似度是基于叶节点的相似度得到的,也就是说,两个非叶节点的元素是相

似的,如果它们的叶节点集合是高度相似的,即使它们的直接子节点并不完全相似.因此在对两个非叶节点概念进行比较时,首先是采集这两个概念树的叶节点的集合,如果这两个集合彼此都有一定程度的相似,那么我们就可以认为这两个待比较的概念间具有一定的对应关系,反之,我们就可以认为这两个概念间不具备对应关系.

定义 5　非叶节点的相似度函数定义为:

$$sim(e_1,e_2)=\sum_{i\in set(e_1,map(O_1,O_2))}W_i^{e_2}\Big/\sum_{j\in set(e_2)}W_j^{e_2} \quad (4)$$

其中,$set(e_1,map(O_1,O_2))$表示 O_1 中 e_1 的叶节点在 O_2 中对应的叶节点的集合,而 $set(e_2)$表示 O_2 中 e_2 的叶节点的集合.W 则代表这些节点在 O_2 中具有的相对权值.所谓相对权值,是指在对不同非叶节点进行计算的时候,每个叶节点的权值是不同的.对于权值的给定,本文采用如下思想:对于待计算的非叶节点 $e\in 0,l\in 0$ 是 e 的一个叶节点,那么如果 l 离 e 的距离越近,l 对 e 的相对权值就越高,因此 W^e 与 $d(e,l)$成反比.采用此思想是因为在一个层次结构中,一个节点离根节点的距离越近,那么它所代表的信息越抽象,反之越具体.

3.2　概念的分类

分类是将原类中具有相似语义信息的子树进一步合并.节点间的映射操作只在同一类中进行,这样可以避免两个本体中不相关概念间的映射操作,从而提高映射算法的性能.

类的形式化定义如下:

$$\begin{aligned} CL0 &= Graph\\ CL0 &: CL^+\\ CL &: C\mid CL^*\\ C &: (Tree)^+ \end{aligned}$$

对于每一个元类 C,根据相似度函数 sim 依次计算出其中分属于两个本体的树的相似度值,再通过定义阈值来确定两棵树是否应该归为一类.这里采用的是 Constant similarity value 方法,也就是使用一个固定值来作为阈值.

$$b=c,b\text{ 为阈值} \quad (5)$$

当且仅当 $sim(e_1,e_2)<b$ 并且 $sim(e_2,e_1)<b$ 时,e_1 与 e_2 不分在同一类,同时得出 $map(e_1,e_2,O_1,O_2)=null$,$map(e_2,e_1,O_2,O_1)=null$,否则它们将被划分在同一类中.

分类函数 $divide$ 通过相似度函数 sim 以及阈值 b,就可以将图中现有的元类 C 进行进一步分类.

定义 6　分类函数定义如下:

$$\begin{aligned} divide(C,sim,b)&=divide((t_1,\cdots,t_n,tt_1,\cdots,tt_m),sim,b)\\ &=(C_1,\cdots,C_k)=CL \end{aligned} \quad (6)$$

其中 $t_i\in O_1,1\leqslant i\leqslant n$,$tt_j\in O_2,1\leqslant j\leqslant m$,$C_l=(t_l^1,\cdots,t_l^p,tt_l^1,\cdots,tt_l^q),p\geqslant 0,q\geqslant 0,p\times q\neq 0$.

且对任意 $C_l,1\leqslant l\leqslant k$,有 $sim(t_l^{p'},tt_l^{q'})>b$ 或 $sim(tt_l^{q'},t_l^{p'})>b$,对任意 $C_l,C_{l'},1\leqslant l,l'\leqslant k,l\neq l'$,有 $C_l\cap C_{l'}=\varnothing$.

具体分类过程举例如图 1 所示.

$$\begin{aligned} divide(C,sim,b)&=divide((a,b,e,c,d),sim,b)\\ &=(a,c)\mid(b,d)\mid(e)\\ &=C1\quad\mid C2\quad\mid C3 \end{aligned}$$

3.3　调节图结构

在将图重新分类之后,将根据各类中根节点的 sim 函数值以及前面定义的阈值 b,重新组织图结构.组织方法是将一些根节点去除,并将去除根节点的树依据继承关系分解为一组子树.选择被去除的根节点的原则如下:

在同一类中,对于 $e_1\in O_1,e_2\in O_2$,如果

1. $sim(e_1,e_2)<b$ 并且 $sim(e_2,e_1)>b$
   则将节点 e_2 去除,并得出
   $map(e_1,e_2,O_1,O_2)=subclass$,
   $map(e_2,e_1,O_2,O_1)=superclass$
2. $sim(e_1,e_2)>b$ 并且 $sim(e_2,e_1)<b$
   则将节点 e_1 去除,并得出
   $map(e_1,e_2,O_1,O_2)=superclass$,
   $map(e_2,e_1,O_2,O_1)=subclass$
3. $sim(e_1,e_2)>b$ 并且 $sim(e_2,e_1)>b$
   任取一节点去除,并得出
   $map(e_1,e_2,O_1,O_2)=equivalentClass$,
   $map(e_2,e_1,O_2,O_1)=equivalentClass$

4　本体映射模型

4.1　映射过程

基于模式结构的自顶向下的本体映射算法为:

1. $G_1=Graph(O_1);G_2=Graph(O_2);G=G_1\cup G_2$;
   $leafMap=leafMatch(G_1,G_2)$ //using
   $leafsim(e_1,e_2)$
2. For each G_i in G
   For each (e_1,e_2) in G_i
   $similarityMatrix\leftarrow sim(e_1,e_2)$;
3. For each G_i in G
   $divide(C_i,sim,b)$;
4. rebuild(G);
   if (exist $c\in CL$ can be divided)
   goto 2;
5. $result=leafMap+similarityMatrix$;

图 2 给出了本体映射的主要流程,输入是两个本体,映射的任务是建立两个本体间的映射关系,初始时是将两个本体中的所有树归为一类.映射过程主要分为 4 个步骤(其中后 3 个步骤是一个迭代过程):(1)应用相似度函数 $leafsim(e_1,e_2)$计算所有叶节点的相似度,并产生两个本体中叶节点的映射矩阵;(2)应用相似度函数 $sim(e_1,e_2)$计算各类中顶层节点的相对相似

度.这一步将输出部分节点间的映射函数值;(3)根据概念的分类原则将图中各类中的各个树再进行分类,并且产生新的合并图;(4)根据各类中根节点的映射函数值,重新组织图结构.当所有类中均无树可以继续分解时,迭代过程结束.这时将叶节点的映射矩阵和相似度矩阵结合起来,得到最终的本体映射结果.

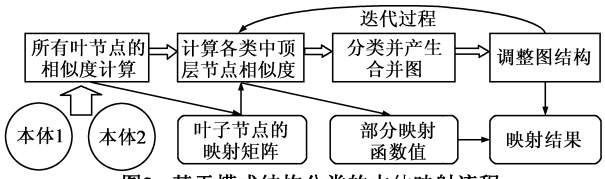

图2 基于模式结构分类的本体映射流程

4.2 例子

本小节将通过一个实例对本体映射过程进行进一步说明.

(1)首先将两个待比较本体按照其自身的继承关系转化为两个图,本文使用文献[2]中的本体用于举例说明,其本体模式结构图如图 3 所示.

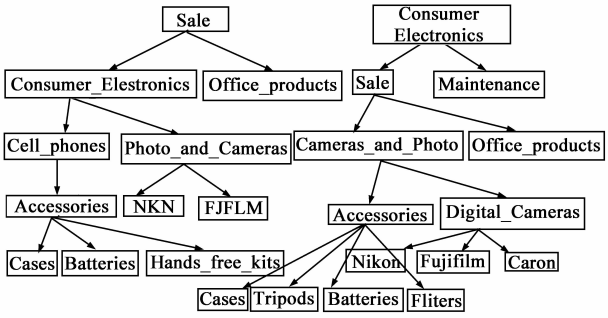

图3 两个本体模式图

使用 *leafsim* 函数对所有叶节点进行计算,所得结果如表 1 所示.

表 1 本体 1 与本体 2 的 *leafsim* 值

本体 1	本体 2	leafsim
Office_Products	Office_Products	0.8
NKN	Nikon	0.48
FJFLM	FujiFilm	0.5
Cases	Cases	0.8
Batteries	Batteries	0.8
…	…	…

取阈值公式中的 p 为 0.5,得出 $cut - off = 0.8 * (1 - 0.5) = 4$,并得到的叶节点映射关系如表 2 和表 3 所示.

表 2 本体 1 叶节点对应表 $leafmap(o_1, o_2)$

本体 1 节点	对应节点
Office_Products	Office_Products
NKN	Nikon
FJFLM	FujiFilm
Cases	Cases
Batteries	Batteries
Hand-Freekis	Null

表 3 本体 2 叶节点对应表 $leafmap(o_2, o_1)$

本体 2 节点	对应节点
Maintenance	Null
Office_Products	Office_Products
Nikon	NKN
FujiFilm	FJFLM
Canon	Null
Cases	Cases
Batteries	Batteries
Tripods	Null
Filters	Null

(2)初始时树 *Sale* 和树 *Consumer Electronics* 在同一类中,使用 *sim* 函数计算得出:

$$sim(Sale, Consumer\_Electronics) = 0.59$$

$$sim(Consumer\_Electronics, Sale) = 0.85$$

(3)取 $b = 0.5$,因此树 *Sale* 和树 *Consumer Electronics* 仍被划分在同一类中.

(4)将根节点 *Consumer Electronics* 去除,重新组织之后产生的如图 4.

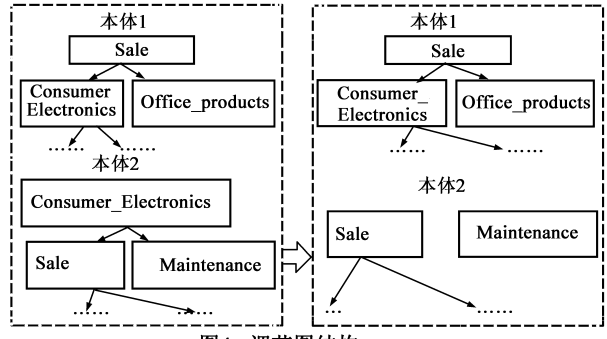

图4 调节图结构

对以上三步进行迭代,当所有类中均无树可以继续分解时,迭代过程结束.这时将叶节点的映射矩阵和 *map* 值结合起来,得到最终的本体映射结果.

5 实验结果分析

本文通过映射几组不同领域的本体,从算法效率及准确性两方面评估本文提出的本体映射方法.

在效率方面,由于本文采用的方法是将待匹配的两个本体逐层进行分类,因而在分类过程中可避免对不相关本体信息进行相似度计算.当对两个分别具有 N 个节点的本体进行比较时,在一般的本体映射方法中,相似度的计算量将会达到 $O(N^2)$.在本文提出的方法中,当两个本体的相似度很高,且涉及概念相对独立时,相似度的计算将会达到 $O(M^2)$,其中 M 为每个本体的叶节点数.可见当本体非常大,涉及概念非常多时,这种方法对提高效率的影响是非常大的.

在准确性方面,由于本文提出的映射方法是基于本体结构进行分析计算的,因此,结果的准确性与待匹配本体的结构设计有很大关系.当两个待比较本体的

结构设计非常清晰、丰富时,映射结果将非常理想.由于现在实际领域的本体大多涉及很多概念,因而在结构上都被抽象出很多层次,在这种情况下,该映射方法将得到很好的发挥.

总之,当两个待比较本体涉及的概念很多,且结构层次分明时,本文提出的本体映射方法将在效率及准确性两方面得到理想结果.

6　总结

本文提出了一种基于模式结构分类的本体映射方法.本方法将两个待比较的本体的映射过程分为对叶节点的映射和非叶节点的映射两个部分,其中非叶节点的映射结果是通过逐层进行分类迭代得到的.在相似度计算方法上,对叶节点的相似度计算是按照多规则整合的方法进行的,对非叶节点的相似度计算是通过加权的集合比例求解进行的,且两种相似度都是相对相似度.最后的映射结果将涉及四种关系:*equivalentClass*, *subclass*, *superclass* 和 *null*.当本体的数目大于 2 个时,可采用概念组分离的方法,即相似的本体在同一组中,不相似的本体在不同的组中,分离至收敛为止.实验证明,本文提出的方法在大规模本体映射发现中得到了理想结果.在下一步的工作中,我们将考虑更加复杂的映射关系,并且进一步提高映射的精度.

参考文献

[1] 金龙飞,刘磊.一种本体演化波及效应分析方法[J].电子学报,2006,34(8):1469 – 1474.
Jin Long-Fei, Liu Lei. A ripple-effect analysis method for ontology evolution[J]. Acta Electronica Sinica, 2006, 34(8): 1469 – 1474. (in Chinese)

[2] Shvaiko P, Euzenat J. A survey of schema-based matching approaches[J]. Springer Journal on Data Semantics IV, 2005, 37(30): 146 – 171.

[3] Melnik S, Garcia M, Rahm E. Similarity flooding: A versatile graph matching algorithm and its application to schema matching[A]. 18th International Conference on Data Engineering (ICDE)[C]. San Jose CA, 2002. 117 – 128.

[4] Euzenat J, Valtchev P. Similarity-based ontology alignment in OWL-lite[A]. Proceedings of the European Conference on Artificial Intelligence (ECAI)[C]. Valencia, Spain, 2004. 333 – 337.

[5] Noy N, Musen M. Anchor-PROMPT: using non-local context for semantic matching[A]. Proceedings of the workshop on Ontologies and Information Sharing at the International joint the conference on Artificial Intelligence (IJCAI)[C]. Seattle, USA, 2001. 63 – 70.

[6] Dieng R, Hug S. Comparison of "personal ontologies" represented through conceptual graphs[A]. Proceedings of the 13th European Conference on Artificial Intelligence (ECAI)[C]. Brighton, UK, 1998. 341 – 345.

[7] Gruber T R. A translation approach to portable ontology specifications[J]. Knowledge Acquisition, 1993, 5(2): 199 – 220.

[8] Silver E A. An overview of heuristic solution methods[J]. Journal of the Operational Research Society, 2004, 55(9): 936 – 956.

[9] Wang Z J, Wang Y L, Zhang S S. Effective large scale ontology mapping[A]. The 1st International Conference on Knowledge Science, Engineering and Management, KSEM[C]. Guilin, China, 2006. 454 – 465.

[10] Shasha D, Wang J T L, Giugno R. Algorithmics and applications of tree and graph Searching[A]. Proceedings of the Symposium on Principles of Database Systems (PODS)[C]. Madison, Wisconsin, 2002. 39 – 52.

[11] Shasha D, Zhang K. Pattern Matching Algorithms[M]. Oxford University Press, 1997.

[12] Ehrig M, Sure Y. Ontology mapping-an integrated approach[A]. Proceedings of the European Semantic Web Symposium (ESWS)[C]. Heraklion, Greece, 2004. 76 – 91.

[13] Do H H, Rahm E. COMA-a system for flexible combination of schema matching approaches[A]. Proceedings of the Very Large Data Bases Conference (VLDB)[C]. Roma, Italy. 2001. 610 – 621.

[14] Madhavan J, Bernstein P, Rahm E. Generic schema matching with Cupid[A]. Proceedings of the Very Large Data Bases Conference (VLDB)[C]. Roma, Italy. 2001. 49 – 58.

作者简介

周　栩　男,1975 年出生于吉林长春.现为吉林大学讲师,博士生,主要研究方向为语义网和本体工程.
E-mail: zhouxu@jlu.edu.cn

刘　磊(通讯作者)　男,1960 出生于吉林长春.现为吉林大学教授、博士生导师,主要研究方向为程序理论、软件工程和语义网技术.
E-mail: liulei@jlu.edu.cn

范任宏　女,1983 出生于吉林长春.现为吉林大学硕士生,主要研究方向为语义网和本体工程.
E-mail: Hr_fan@yahoo.cn