第35卷 第23期 计 算 机 工 程 2009年12月
Vol.35 No.23 Computer Engineering December 2009
·软件技术与数据库· 文章编号：1000—3428(2009)23—0062—04 文献标识码：A 中图分类号：TP311.52

基于可信度的多策略本体映射

曾凡秩
（湖南工程职业技术学院信息工程系，长沙 410114）

摘 要：针对本体映射中各种策略不能依据本体间的差异进行不同映射处理以及多策略结合时权值分配不合理的问题，提出一种本体映射方法。该方法对各种映射策略根据当前映射任务的特点进行自适应处理，在映射过程中对不同策略的可信度做自适应计算，依据可信度进行多策略结合，得到最终的映射结果。实验结果表明，该方法在保证通用性和稳定性的同时，能提高映射的查准率。
关键词：本体映射；多策略；自适应；可信度

Ontology Mapping for Multi-strategies Based on Credibility

ZENG Fan-zhi
(Department of Information Engineering, Hunan Engineering Polytechnic, Changsha 410114)

【Abstract】 Aiming at the problems of many strategies can not do different operations on ontology mapping according to the differences existed in ontology and they are distributed the irrational weights when integrating each strategy, this paper proposes a new ontology mapping method. In this method, different mapping strategies do different process according to the character of mapping task, and the credibility of different strategies is computed in the process of mapping, the different strategies are assembled by the credibility, get the final mapping results. Experimental results show that this method can improve the recall and precision of ontology mapping while maintaining currency and stability.
【Key words】 ontology mapping; multi-strategies; self-adapting; credibility

1 概述

语义 Web 的发展导致本体数量激增，然而由于本体的创建者不同、使用的建模方法不同，不同的领域专家开发出来的本体必然存在着差别。本体映射的目的就是要找到本体之间的语义联系，以便知识共享和重用。

目前，大多数本体映射系统都综合利用多策略来发现映射，但各种映射策略都不能根据映射任务的不同而对算法进行自适应调整，在对多种映射策略进行结合时也不能充分利用本体的语义信息，导致了在某一映射任务中表现很好的方法可能在另一映射任务中表现得并不理想，并使各种策略的价值有所降低，影响了映射的查全率和查准率，不能满足大多数映射任务和语义检索的需求。

针对上述缺陷，本文提出一种面向多策略的自适应本体映射方法，该方法在传统方法的基础上对各种策略进行自适应改进，在映射过程中依据各策略在当前映射任务中的表现对该策略的可信度进行计算，通过可信度结合各种映射策略，得到最终的映射结果，并通过实验验证该方法的有效性。

2 相关工作

随着本体映射思想的提出，对于映射策略[1]的研究是映射技术中的关键部分和研究热点。由于本体构建者的习惯、水平、拥有的资源、针对的应用领域的不同，某种映射策略所需要的信息可能恰恰是某些本体所缺乏的，因此导致了映射策略存在着固有缺陷， 例如，传统的名称映射策略是一种基于字符串处理的方法(如编辑距离、相同字符个数)，但名称相同的 2 个概念可能是同名异义，而当两者的名称完全不同时，也可能表示相同的语义；当待映射本体的结构差异程度很大时，通过自底向上的遍历方式来发现映射(如 cupid[2])，得出的映射结果很多是不正确的。

由于实际的待映射本体常常包含多种信息，将几种策略结合使用可以更充分的利用本体信息，因此往往会比采用单一策略产生更好的结果。清华大学设计的 RiMOM[3]系统是一种基于风险最小化的本体映射模型，它对多个映射策略进行了整合，但该系统在面对结构特征差异较大的本体时，由于不能进行自适应处理，而是都假设使用不同策略所获得的相似度值可以累加，对其得到的相似度赋以相应权值，这实际上是把多个结果进行数值上的结合，而并没有真正在语义层面综合考虑各种策略对映射结果的重要性，因此，待映射本体对缺失了某一方面的特征或特征发生变化时(随着本体的进化这是很可能的)，容易因为权重过配或分配不足而得出错误的或遗漏掉映射关系，从而影响映射的质量。

3 面向多策略的自适应本体映射

为了叙述方便，本文先给出本体的定义：

定义 1 本体：$O=(C, P, R, I, T)$，其中，C 表示概念的集合；P 表示属性的集合；R 表示关系的集合；I 表示实例集合；T 表示公理集合。概念、属性通称为实体。假设 A 和 B 分别表示源本体 O_1 和目标本体 O_2 中的某实体。

3.1 基于名称的策略

本体中相似的实体对其名称通常也存在相似性，传统的基于编辑距离或字串的方法来进行名称映射的准确度都比较

作者简介：曾凡秩(1969－)，男，讲师、硕士，主研方向：本体映射
收稿日期：2009-04-04 **E-mail**：zfz01048@163.com

低，不适合使用。本文基于 WordNet 来进行实体间的名称相似度计算，其核心思想是：如果 2 个实体的 URI 一致或 2 个实体互为同义词则相似度为 1；否则，通过计算 2 个实体的同义词集在 WordNet 中的路径距离来计算相似度，取最大值。

定义 2 在 wordNet 层次图中，实体 *A, B* 之间的语义距离 *Dist*(*A, B*)为连接它们的最短路径上 *n* 条边的权值的总和，即

$$Dist(A,B)=\sum_{i=1}^{n} weight_i \tag{1}$$

其中，$Weight_i$ 是连接 *A, B* 的最短路径上第 *i* 条边的权值。考虑到自顶向下，实体的分类是由大到小，大类间的相似度肯定要小于小类间的，所以处于不同深度的实体的边对其赋予不同的权值，当实体由抽象逐渐变得具体，连接它们的边对语义距离计算的影响将逐渐减小。即边的权值可表示为

$$Weight(E)=\frac{1}{2^{Dep(E)}} \tag{2}$$

其中，*Dep*(*E*)表示边 *E* 在 WordNet 层次树中的深度。对于根节点来说，*Dep*(*E*)为 0。

另外，提出 2 个关键因子：边的强度和边的密度。一般地，一个父节点对某一子节点相对于其他子节点越重要，即边的强度越大，则该父子节点相连的边的权值越大；随着边的密度增加，边的权值越大。

假设某一条边 *E* 的子节点为 *C*，父节点为 *F*，边的强度可以表示为

$$edge_{\text{important}}(E)=\left|IC(C)-IC(F)\right| \tag{3}$$

其中，*IC*(*C*)和 *IC*(*F*)分别表示节点 *C* 和节点 *F* 包含的信息量，信息量的计算参考文献[4]。

边的密度可以表示为

$$edge_{\text{density}}(E)=\frac{1}{Wid(F)} \tag{4}$$

其中，*Wid*(*F*)表示由节点 *F* 引出的边的数目。

根据上述分析，则任意一条边 *E* 的权值经过修正后可表示为

$$Weight(E)=\frac{1}{2^{Dep(E)}}\times edge_{\text{important}}(E)\times edge_{\text{density}}(E) \tag{5}$$

式(5)保证了随着实体的边在 WordNet 层次结构中所处深度、强度和密度的增加，其权值会减小。

对于同义词集中的词汇，其名称相似度由同义词在 WordNet 层次树中的语义距离确定。分 2 种情况考虑：如果待比较实体的同义词存在着公共上位词(cuw)，则两者的距离由它们分别与最近公共上位词的 *Dist* 值之和确定；如果不存在公共上位词，则由两者的最短路径距离确定。其语义距离度量公式为

$$S(A,B)=\begin{cases} Dist(A,cuw(A,B))+Dist(B,cuw(A,B)) & A,B\text{有公共上位词} \\ Dist(A,B) & \text{其他} \end{cases} \tag{6}$$

根据式(6)转化得名称相似度计算公式为

$$Sim_{name}(A,B)=\begin{cases} 1 & A\text{和}B\text{的URI相同，}A\text{和}B\text{互为同位词} \\ \max(1-S(Synset_A,Synset_B)) & \text{其他} \end{cases} \tag{7}$$

其中，$Synset_A$, $Synset_B$ 分别表示实体 *A, B* 的同义词集合，由于 WordNet 是依据实体之间的语义组成的同义词典，因此该方法不仅在实体名称完全或部分相同的情况下有效，而且在实体名称完全不同但存在一定语义关联的情况下也有效。

3.2 基于结构的策略

本体的结构蕴含着丰富的语义信息，在本体映射中起到非常重要的作用。本文针对传统方法存在的缺陷，考虑了本体结构特征的差异，将结构级映射分为 2 种情况：自顶向下映射和自底向上映射。一般来说，自顶向下映射算法花费的代价较小，因为一开始所要比较的对象比较少，以后的比较也只要用到前面的比较结果，而自底向上映射由于利用的是本体中最终描述的原子数据，所以更有可能得到好的映射结果。但是对于具体的应用环境，须做不同的处理以保证映射的质量。本文给出 2 个本体部分节点的树形层次结构图，如图 1 和图 2 所示。

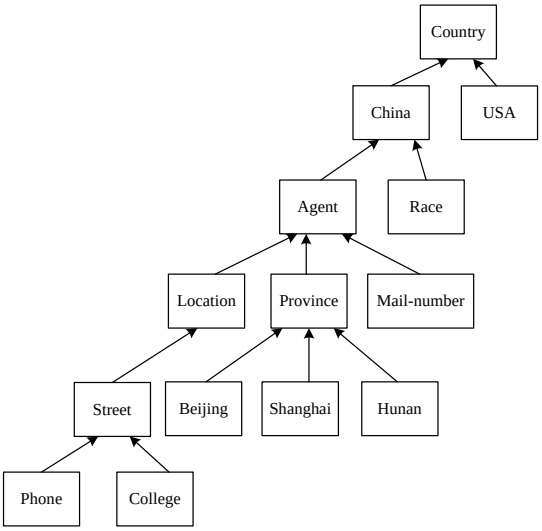

图 1　源本体(片段)

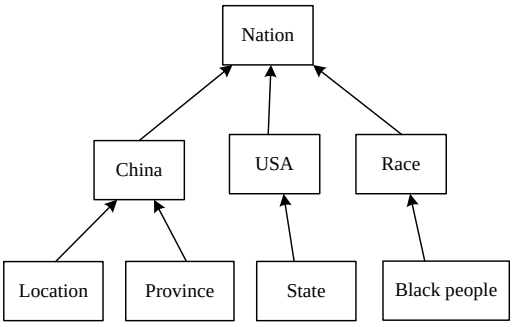

图 2　目标本体(片段)

对于图 1 和图 2 所示的源本体和目标本体，如做自底向上的映射，先计算 2 个本体叶子节点的相似度，再不断向上迭代计算，但是由于两者的结构存在较大的差异(概念个数相差很大、本体树高度不一致、叶子节点数目不同)，它们的叶子节点并没有多大的相似性，映射结果显得意义不大。相反，此时如进行自顶向下的映射，效果会更好。

依据上述分析，结构级映射时，遍历方式的选择需谨慎考虑，如果遍历方式不当，则有可能出现计算复杂度增加但得出的映射结果却偏低的情况。针对这个问题，本文提出利用本体异构度来评价不同本体间结构特征的差异程度，当本体异构时，通过层次遍历，做自顶向下映射；反之，通过后序遍历，做自底向上映射。

定义 3 本体异构度定义为

$$isomerous(O_1,O_2)=1-\frac{Entity(O_1,O_2)+Depth(O_1,O_2)+Leaf(O_1,O_2)}{3} \tag{8}$$

其中，*Entity*(O_1, O_2)指本体中实体个数的比值(总取较小的一个为分子)；*Depth*(O_1, O_2)指本体树高度的比值；*Leaf*(O_1, O_2)指叶子数目的比值。用 1 减去 3 个因子和的平均是因为：即

使高度(或者叶子数目)不一样，但是如果实体的个数差不多，则结构也可能是近似的。当 *isomerous*>*p*(规定值)时，则认为源本体和目标本体异构。*p* 值由领域专家或实验确定，本文中取值 0.7。结构级映射的算法描述如下：

算法 1 结构级映射算法

```
Structure-Match(SourceTree S,TargetTree T){
sim_structure(s_i,t_j)=datatype_compatibility(s_i,t_j); //初始化操作
If (isomerous<P){//做自底向上映射
Down_Top Matching(S,T){ //做自底向上映射
//按后序遍历来唯一列举本体树的元素
ArrayList  s'= post_order(S);  //s'={s0,s1,…,sn}
ArrayList  t'= post_order(T);  //t'={t0,t1,…,tm}
sim(S,T)=W_structure* sim_structure(s_i,t_j)+( 1-W_structure)*sim_name(s_i,t_j)
if(sim>P1)  增加叶子集的相似度；
    if(sim>P2)  降低叶子集的相似度；}}
Else{
Top_Down Matching(S,T){ //做自顶向下映射
//按层次遍历的顺序来唯一列举本体树的元素
ArrayList  s'= layer_order(S);  //s'={s0,s1,…,sn}
ArrayList  t'= layer_order(T);  //t'={t0,t1,…,tm}
sim(S,T)=W_structure* sim_structure(s_i,t_j)+( 1-W_structure)*sim_name(s_i,t_j)
if(sim>p3)  迭代计算直到收敛于固定点；}}
  }
```

3.3 基于实例的策略

传统的方法利用 Jaccard[5]相似度来进行实例级相似度计算，这种方法具有比较直观的意义：当 2 个实体没有公共的实例时，其相似值为 0。如果所有的实例均相同，其相似值为 1。但这种方法没有考虑实例个数的差异程度，当实体间的实例数目分布不均衡时，得出的映射结果容易失真。本文在传统的方法上进行改进，提出 2 个关键因子：丰富度和差异度。

定义 4 丰富度定义为

$$richness = \min\left\{1-\frac{1}{(sum_A+a)},1-\frac{1}{(sum_B+a)}\right\} \tag{9}$$

其中，sum_A, sum_B 分别表示 *A, B* 的实例集；*richness* 值随实例数目的增大而增大，但随实例数目的增大，*richness* 的增长应该放缓(因为 2 个实例比一个实例要可信得多，但 50 个实例与 40 个实例则没有明显区别)；*a* 的作用是令 *richness* 值在实例数目为 1 时不会过小。2 个实体拥有的实例数目越丰富时，则基于实例的策略得出的结果越可靠。

定义 5 差异度定义为

$$difference=1-\frac{sum_A}{sum_B} \tag{10}$$

总取较小的一个为分子。差异度反应 2 个实体实例丰富程度的差异，差异越大，*difference* 值越大。当 *difference* 值较大时，即算$|sum_A \cap sum_B|=|sum_A|$，即 *A* 的实例完全被映射，最终计算出的实例相似度都可能永远达不到阈值。为了避免这种情况，*difference* 值较大时，分母用 $2\times\min(|sum_A|, |sum_B|)$ 来取代$|sum_A \cup sum_B|$。基于实例的实体相似度计算公式为

$$Sim_{instance}(A,B)=\begin{cases} richness\times JaccardSim(A,B) & difference \leqslant E \\ richness\times\dfrac{|sum_A \cap sum_B|}{2\times\min(|sum_A|,|sum_B|)} & difference > E \end{cases} \tag{11}$$

3.4 基于属性的策略

待比较实体对有相同的属性时，则这 2 个实体可能是相似的。依据这种思想，本文通过判断 2 个实体对应的属性级相似度来发现映射。但由于一个实体可能会有多个属性，每个属性对实体的描述程度和作用是不同的，因此，在使用属性计算相似度时，本文通过计算属性包含的信息量来确定属性的优先级，只对优先级高的属性进行相似度的计算，以减少计算量。有些属性(如 String, Int 等)在本体中出现的次数太多，则认为没有比较的价值，对于这些属性，直接去除掉。另外，考虑到本体中实体的属性通常分为对象属性和数据类型属性，本文分别做出映射处理。这样做的好处是在保证映射质量的同时提高了时间效率。属性级映射的算法描述如下：

算法 2 属性级映射算法

输入 经过优先级排序的待比较实体的属性集

输出 实体的映射关系

Step1 迭代抽取对象属性 OP 和数据类型属性 DP 并分别存储；

Step2 当待比较实体对的属性属于 OP， 如果它们的定义域和值域、属性的子属性或父属性相同，则匹配；

Step3 当待比较实体对的属性属于 DP，如果它们的值域相同、取值范围相同或约束一致，则匹配；

Step4 迭代执行 Step2 和 Step3，直到没有新的映射关系加入。

3.5 多策略的结合

本体映射的一个关键问题是怎样最优地结合各个策略，以充分利用本体的语义信息。映射结果的质量取决于各映射策略是否适合于映射任务的特点，例如，待映射本体间拥有的实例数目越多，则基于实例策略的映射结果的精确度就越高；待映射本体对的结构信息越丰富，则基于结构策略的映射结果的精确度就越高。

在实际应用中，对于不同的匹配器(或多策略)，本文选择不同的本体特征来进行可信度预测，对于名称匹配器，长的标签常带有更多的信息，因此，2 个待匹配元素的标签长度是一个特征；同样，标签中的词数也可以作为一个特征。路径名字匹配器的情况与名称匹配器类似，但还须考虑路径长度这个特征；对于结构匹配器，一个元素节点的叶子数目，以及其子树的深度，都将影响该匹配器的有效性，故为其 2 个特征；另一个重要的特征是初始相似度的可信度，因为毫无疑问这会影响该匹配器的效果。对于大多数匹配器来说，其输出的相似度是一个有用的特征，因为一个匹配器的不同的输出值常具有不同的可靠性。

基于以上分析，本文依据本体的特征在映射过程中动态地对各策略的可信度进行预测，通过预测得到的可信度利用 Sigmoid 函数[5]对多策略进行结合，由于 Sigmoid 函数法将较高的相似度赋予高权值比例，将较低的相似度赋予低权重值，突出了主要语义成分，因此，在大多数情况下能够得到合理的映射结果。

进行可信度预测的依据是：对于任意待映射实体对，通过某策略计算得到的相似度与通过机器学习方法得到的估计值的差异程度越小，表明该策略越可信。

定义 6 可信度定义为

$$Credibility_t = e^{-\frac{1}{N}\sum_{i=1}^{N}(sim_i^t-sim_i^t)^2} \tag{12}$$

其中，*N* 表示映射任务中待映射本体的实体对的数目；sim_i^t 表示不同的实体对依据不同策略计算出的相似度(*t* 表示不同的

策略，i 表示不同实体对)；而 sim_i^t 的值则通过机器学习方法来进行评估(本文使用在线学习技术，限于篇幅不再详述)，对于匹配的元素为 1，否则为 0。

上述公式计算得到的可信度代表了各个策略对于最终映射结果的重要程度，即各策略的权重。最后，本文选择 Sigmoid 函数来进行多策略的结合：

$$Sim(e_{O_1}, e_{O_2}) = \frac{\sum_{t=1,2,3,4} Credibility_t \sigma(sim_t(e_{O_1}, e_{O_2}))}{\sum_{t=1,2,3,4} Credibility_t} \tag{13}$$

其中，$\sigma(x) = \frac{1}{1+e^{-5(x-0.5)}}$，$sim_t(e_{O_1}, e_{O_2})$ 表示不同策略得出的相似度，t 的取值表示不同的策略。

4 实验结果及分析

本文利用 OAEI2007 的标准测试数据集进行实验，在 6 组测试数据中，只有第 1 组的测试方式为公开并且提供了相应的标准测试结果。在对系统进行实验测试时，为了方便进行统一地度量和与其他系统的比较，本文也只利用 OAEI 提供标准结果的第 1 组数据 benchmarks 作为实验的测试数据。此数据集中，包含了 51 个本体，其中一个本体 # 101 为参考本体(Reference Ontology)，其他的本体皆为某种(些)特征缺失后的变换形式，称为其变体。另外，本文采用信息检索领域的查全率和查准率作为评价的标准，定义如下：

查准率(Precision)：

$$p = |R \cap A| / |A| \tag{14}$$

查全率(Recall)：

$$r = |R \cap A| / |R| \tag{15}$$

其中，A 表示算法识别得到的正确映射结果；R 表示参考映射结果。将本文方法得出的映射结果取名为 Map，将其与目前国内外典型的映射系统得出的结果进行比较，对比结果如表 1 和表 2 所示。其中，# 1××~3××表示标准测试数据集 benchmarks 中第 1 组的本体编号。

表 1　Map 与其他测试系统的查全率比较

System	Falcon	Rimom	OntoDNA	ASMOV	Map
1××	1.00	1.00	1.00	1.00	1.00
2××	0.87	0.80	0.76	0.87	0.84
3××	0.77	0.86	0.78	0.86	0.78
Total	0.88	0.89	0.85	0.91	0.87

表 2　Map 与其他测试系统的查准率比较

System	Falcon	Rimom	OntoDNA	ASMOV	Map
1××	1.00	1.00	0.94	1.00	1.00
2××	0.93	0.94	0.75	0.94	0.95
3××	0.82	0.90	0.90	0.75	0.93
Total	0.92	0.95	0.86	0.90	0.96

从实验结果来看，Map 的查准率最高，查全率比 Falcon[6]，Rimom[3]和 ASMOV[7]等优秀映射系统稍低，但高于 OntoDNA[8]。说明本文提出的方法在保证了查全率的同时，较为明显地提高了查准率，即提高了映射结果的质量，对于各组不同的测试数据，Map 的性能也比较稳定。这是由于在映射过程中充分利用了本体的语义信息来进行相似度计算，并针对本体结构、实例数目的差异进行了自适应处理，从而使得映射算法更精确地反映了实体之间的语义关系；在对多策略进行结合时，通过动态地预测各个策略的可信度，避免了传统方法中由于人为干预或简单加权造成的结果偏差，从而使得映射算法更具通用性。此外，通过仔细对比各组数据的特点和不同映射算法的实验结果，我们发现对部分本体使用传统映射算法会出现以下情况：可参考的信息增加时，查全率和查准率同时降低。

由于 Map 并未对每种映射策略赋以固定的权值，而是根据本体的特征进行动态的预测，因此，当某种信息增加时，只会提供更丰富的语义信息，不会发生同样的问题。从整体实验结果均衡来看，该方法是有效的，达到了预期的目的。

5 结束语

本文针对现有多策略本体映射存在的缺陷，提出了改进的映射方法，对单个策略进行了优化，并通过分析待映射本体对的特征，在多策略结合时，利用可信度对各个策略做重要性评判，充分利用了各策略对映射结果的价值，使某些特征信息的缺乏不会对整个系统产生大的影响。

实验结果表明，本文方法的性能比传统的本体映射方法有所改进。下一步研究工作包括以下 2 个方面：

(1)将应用领域的知识如规则，约束等信息融入相似度计算中；

(2)利用推理技术对映射结果进行一致性检测，修正并优化。

参考文献

[1] Giunchiglia F, Yatskevich M, Shvaiko P. Semantic Matching: Algorithms and Implementation[J]. Journal on Data Semantics, 2007, (9): 1-38.

[2] Madhavan J, Bernstein P A. Generic Schema Matching with Cupid[C]//Proc. of VLDB'01. Rome, Italy: Morgan Kaufmann Publishers, 2001.

[3] Li Yi, Zhong Qian, Li Juanzi, et al. Result of Ontology Alignment with RiMOM at OAEI2007[C]//Proc. of the ISWC'07. Bexco, Korea: [s. n.], 2007.

[4] Wen Dunwei, Fan Xiaohu. Information Theory Based Heuristic Ontology Matching Framework[J]. Computer Technology and Development, 2007, 17(10): 43-46.

[5] Cheng Yong, Huang He, Qiu Lirong. A Similarity-based Dynamic Multi-dimension Concept Mapping Algorithm[J]. Journal of Chinese Computer Systems, 2006, 26(5): 975-979.

[6] Hu Wei, Zhao Yuanyuan, Li Dan, et al. Falcon-AO: Results for OAEI 2007[C]//Proc. of the ISWC'07. Bexco, Korea: [s. n.], 2007.

[7] Yves R, Jean-Mary1, Kabuka M R. ASMOV Results for OAEI 2007[C]//Proc. of the ISWC'07. Bexco, Korea: [s. n.], 2007.

[8] Kiu C C, Lee C S. OntoDNA: Ontology Alignment Results for OAEI 2007[C]//Proc. of the ISWC'2007. Bexco, Korea: [s. n.], 2007.

编辑　金胡考