# 一种综合的概念相似度计算方法*)

曹泽文 钱 杰 张维明 邓 苏
(国防科技大学信息系统与管理学院 长沙 410073)

**摘 要** 本体映射可以用来解决本体异构问题,也是本体结盟、本体集成、本体合并、本体翻译等的技术基础。本文针对目前本体映射中概念相似度计算所存在的问题,提出了一种综合的相似度计算方法。首先根据两个概念名称相似性过滤出最相关的概念,减少相似度的计算;然后基于概念实例、基于概念属性、基于概念关系计算概念相似度,并进行综合;最后对其性能进行了简单分析。

**关键词** 本体映射,概念相似度

## A Compositive Approach for Concept Similarity Computation

CAO Ze-Wen QIAN Jie ZHANG Wei-Ming DENG Su
(College of Information System & Management, NUDT, Changsha 410073)

**Abstract** Ontology-Mapping, the base of ontology-alignment, ontology-integration, ontology-merging, ontology-translation, can be used to solve the problem among heterogeneous ontologies. To aim at the current problems of the computation of concept similarity, this paper puts forward a compositive approach. Firstly, the most-related concepts are filtered out according to the similarity between twoconcept names so as to reduce the amount of computation. Secondly, an integrated approach of based-instance computation, based-attribute computation, based-relation computation is designed and is used to compute the concept similarity. Lastly, a simple analyse about the approach is given.

**Keywords** Ontology-mapping, Concept similarity

## 1 引言

随着本体应用领域的增多,如何解决本体间的互操作是一个比较棘手的问题[1]。本体映射能很好地解决本体异构问题,它是发现两个相同领域本体的概念之间的相关性(映射关系)的过程,是本体间概念和关系取得一致性的一个规范说明。本体映射是本体结盟、本体集成、本体合并、本体翻译等的技术基础[2~4],一般分信息本体化、相似性提取、语义映射、映射执行和映射后处理共5步来执行[5]。相似性提取是本体映射的一个重要步骤,它主要是进行相似度的计算,并产生一个相似矩阵。

目前的本体映射存在相似度的计算方法不完善、相似度的计算量过高、概念相似度的计算过于片面等问题。因为本体一般可理解为概念、属性和关系的集合,属性即概念的属性,关系即概念间的关系,所以本体映射主要是集中在概念的相似度计算及相应的映射。在映射过程中,本体映射的核心内容是计算两个概念的相似度,并求出概念的相似矩阵。当其相似度大于某个阈值时,就认为这两个概念之间存在一定的映射关系。

现在最常用的相似度计算方法是根据概念的实例计算相似度[1,6]。根据实例计算相似度是利用一定量的实例在两个概念中出现的联合分布概率来计算两个概念的相似度。对于本体 $O_1$ 中的概念 $A$ 和本体 $O_2$ 中的概念 $B$,$Sim(A,B)$ 表示概念 $A$ 和概念 $B$ 的相似度。该方法对于两个本体的实例集没有交集时就无能为力,而这种情况对于不同部门建立的本体是一种普遍现象。另外,目前计算两个本体 $O_1$ 和 $O_2$ 中概念的相似度时,本体中的每一对概念都被考虑在内。如果本体 $O_1$ 中含有 $m$ 个概念,本体 $O_2$ 中含有 $n$ 个概念,那么就要计算 $m\times n$ 次相似度,也就是每对概念之间的相似度都要计算出来,并形成 $m\times n$ 维的相似矩阵,因此计算量很大。而有的两个概念根本就不相似,计算它们的相似度是没有必要的。因此,计算时应该对概念对的数量进行限制,以减少相似度的计算。第三,目前对于概念相似度的计算,仅仅利用概念自身的语义进行,没有考虑概念的属性和关系对概念的描述作用。对于本体中的每一个概念,概念的属性和关系也是重要的组成部分。在计算概念相似度时,不仅应该考虑概念自身的语义,而且应该考虑概念的属性和关系。

针对以上提出的问题,本文提出一种综合的相似度计算方法。对于本体 $O_1$ 中的一个概念 $A$,我们不是比较本体 $O_2$ 中所有概念,而是根据两个概念名称相似性度量公式过滤出本体 $O_2$ 中最相关的概念,产生一组候选概念集,只对概念 $A$ 与候选概念集中的概念计算相似度。在计算概念相似度时基于概念实例、基于概念属性、基于概念关系分别计算概念相似度,然后进行相似度合并。这样可使概念相似度的计算更加全面,计算结果更加准确。

## 2 综合的概念相似度计算方法

### 2.1 寻找本体中某概念的候选概念集

对于本体 $O_1$ 中的一个概念 $A$,本体 $O_2$ 中一般只有部分概念与它基本相似。我们只选择其中最有可能相似的概念,通过计算概念之间的相似度判断概念之间的相似关系。如何找出这些基本相似的概念?为此,我们根据两个概念名称的相似性,先过滤出本体 $O_2$ 中最相关的概念,产生一组候选概念集。通过对概念对的数量进行限制,可以减少相似度的计

*)国家自然科学基金(60172012)、国防预研基金资助项目(51421020904KG01)。曹泽文 副教授,博士生,主研方向为知识系统、决策支持系统等。

算，提高映射的效率。

我们定义两个概念名称相似性度量公式：

假设本体 $O_1$ 中的概念是 $A$，本体 $O_2$ 中的概念是 $B$，则概念 $A$、$B$ 名称相似性度量公式为 $Sim(Aname, Bname)=\frac{N(Aname\text{与}Bname\text{的最长子串})}{(N(Aname)+N(Bname))}$ (1)

这样，可以计算出本体 $O_2$ 中与本体 $O_1$ 中概念 $A$ 最相似的 $K$ 个概念，即相似度最高的 $K$ 个概念，我们记作 $B_{[1..k]}$，这样可以得到与 $A$ 进行相似性比较的候选概念集为

$CandidateSet(A)=B_{[1..k]}\cup$

所有与 $B_{[1..k]}$ 概念存在关系的概念 $\cup$

所有 $B_{[1..k]}$ 概念的父概念 $\cup$

所有 $B_{[1..k]}$ 概念的子概念 (2)

### 2.2 概念相似度的计算

概念相似度是指概念间自身语义的相似程度，因此概念语义相似度的计算主要考虑概念语义间的相似情况。因为确定候选概念集时已使用了概念名称之间的相似性，此处仅仅根据概念的实例来计算相似度。概念的属性和关系对概念有重要的描述作用，所以计算概念相似度时应该考虑概念的属性和关系对相似度的影响。本文从 3 个方面来计算概念的相似度，即基于实例、基于属性、基于关系分别计算相似度。然后，依据一定的权值把 3 个概念相似度进行合并，生成最终的概念相似度。

#### 2.2.1 基于实例计算概念相似度

在需要映射的两个本体中，可以利用概念的具体实例计算概念相似度。一个概念的实例也是它祖先概念的实例。基于实例计算概念相似度的理论依据是：如果概念所具有的实例全部都相同，那么这两个概念是相同的；如果两个概念具有相同实例的比重是相同的，那么这两个概念是相似的。

用具体实例来计算概念 $A$ 和概念 $B$ 的相似度，记为 $Sim_{instance(A,B)}$，计算公式为

$$Sim_{instance}(A,B)=\frac{P(A\cap B)}{P(A\cup B)}=\frac{P(A,B)}{P(A,B)+P(A,\bar{B})+P(\bar{A},B)} \quad (3)$$

$Sim_{instance}(A,B)\in[0,1]$。最小值为 0，表示两个概念完全无关；最大值为 1，表示两个概念完全相同。

基于实例计算概念相似度牵涉到 3 个概率：$P(A,B)$、$P(A,\bar{B})$、$P(\bar{A},B)$。其中 $P(A,\bar{B})$ 是从一个本体的实例空间中随机选取的一个实例属于 $A$ 但不属于 $B$ 的概率，也可以理解为所有属于 $A$ 但不属于 $B$ 的实例在实例空间中所占的比重。因此，在计算 $P(A,B)$、$P(A,\bar{B})$、$P(\bar{A},B)$ 时要用到概念 $A$ 和概念 $B$ 在各自本体中的实例个数。用 $U_i$ 表示本体 $O_i$ 中的实例集，$N(U_i)$ 表示实例集中的实例个数。用 $N(U_1^{A,B})$ 表示在 $U_i$ 中既属于 $A$ 又属于 $B$ 的实例个数。以 $P(A,B)$ 的计算为例，有以下 6 个步骤：

1)对于本体 $O_1$ 中的实例集 $U_1$，把它分成属于概念 $A$ 的实例集 $U_1^A$ 和不属于概念 $A$ 的实例集 $U_1^{\bar{A}}$。

2)把这两个实例集中的实例分别作为正反样本用机器学习方法来训练对于概念 $A$ 的学习器 $L$。

3)对于本体 $O_2$ 的实例集 $U_2$，把它分成属于概念 $B$ 的实例集 $U_2^B$ 和不属于概念 $B$ 的实例集 $U_2^{\bar{B}}$。

4)使用学习器 $L$ 对实例集 $U_2^B$ 中的实例进行分类，分成两个实例集 $U_2^{A,B}$ 和 $U_2^{\bar{A},B}$。同样，用 $L$ 把实例集 $U_2^{\bar{B}}$ 分成两个实例集 $U^{A,\bar{B}}$ 和 $U_2^{\bar{A},\bar{B}}$。这样可以获得实例集 $U_2^{A,B}$、$U_2^{A,\bar{B}}$、$U_2^{\bar{A},B}$ 和 $U_2^{\bar{A},\bar{B}}$。

5)把本体 $O_1$ 和本体 $O_2$ 的位置调换过来，重复以上各步，最终可以获得实例集 $U_2^{A,B}$、$U_1^{A,B}$ 和 $U_1^{\bar{A},B}$ 和 $U_1^{\bar{A},\bar{B}}$。

6)从各步计算中求得 $N(U_1)$，$N(U_2)$，$N(U_1^{A,B})$ 和 $N(U_2^{A,B})$ 并利用式(4)来计算 $P(A,B)$：

$$P(A,B)=\frac{[N(U_1^{A,B})+N(U_2^{A,B})]}{[N(U_1)+N(U_2)]} \quad (4)$$

采用同样的步骤方法计算 $P(A,\bar{B})$ 和 $P(\bar{A},B)$，计算公式分别为式(5)和式(6)：

$$P(A,\bar{B})=\frac{[N(U_1^{A,\bar{B}})+N(U_2^{A,\bar{B}})]}{[N(U_1)+N(U_2)]} \quad (5)$$

$$P(\bar{A},B)=\frac{[N(U_1^{\bar{A},B})+N(U_2^{\bar{A},B})]}{[N(U_1)+N(U_2)]} \quad (6)$$

然后用式(3)计算概念 $A$ 和概念 $B$ 基于实例的相似度 $Sim_{instance}(A,B)$。

在实例学习过程中，可以根据不同的情况，采用不同的机器学习方法来训练不同的学习器[7]。机器学习的方法有多种，如记忆学习、传授学习、类比学习、归纳学习、解释学习等。每个学习器可以使用其中的一种方法。

#### 2.2.2 基于属性计算概念相似度

概念的属性对概念的描述具有十分重要的作用。在某些本体中，属性也被当成一个概念。基于属性计算概念相似度的理论依据是：如果两个概念的属性都相同，那么这两个概念是相同的；如果两个概念具有相似的属性，那么这两个概念也是相似的。属性有属性名称、属性数据类型、属性实例数据等要素，因此判断两个属性是否相似主要从这 3 个要素的相似度进行考虑。

属性名称、属性类型本身都是字符串，因此可以采用字符串相似度计算方法进行判定。例如用 humming distance 来比较两个字符串。两个字符串 $s$ 和 $t$ 的相似度的计算公式如(7)式所示，其中若 $s[i]=t[i]$，则 $f(i)=0$，否则 $f(i)=1$。

$$Sim(s,t)=1-\frac{(\sum_{i=1}^{\min(|s|,|t|)}f(i))+\|s|-|t\|}{\max(|s|,|t|)} \quad (7)$$

字符串相似度的计算也可以采用其它方法，如 *N-gram distance*、*edit distance* 等，也可以根据子串相似度来确定字符串的相似程度。由于每个概念的实例对该概念的每一个属性都分配了一个相应的值，因此对于其它类型的数据，也可以采用基于实例的方法进行计算。

设概念 $A$ 的属性为 $a_i$，概念 $B$ 的属性为 $b_j$，两个属性间的相似度记为 $ASim(a_i, b_j)$。属性相似度计算公式如下：

$$ASim(a_i,b_j)=W1*Sim(a_{i_{name}},b_{j_{name}})+W2*Sim(a_{i_{datatype}},b_{j_{datatype}})+W3*Sim(a_{i_{instance}},b_{j_{instance}}) \quad (8)$$

其中 $w1$、$w2$、$w3$ 是权重，代表属性名称、类型、数据对属性相似度计算的重要程度，$w1+w2+w3=1$。

设概念 $A$ 和概念 $B$ 之间共计算出 $m$ 个 $ASim(a_i,b_j)$，并设置相应的权值 $w_{attribute}^k$。概念 $A$ 和概念 $B$ 基于属性的相似度计算公式为

$$Sim_{attribute}(A,B)=\frac{\sum_{k=1}^{m}W_{attribute}^{k}ASim(a_i,b_j)}{\sum_{k=1}^{m}W_{attribute}^{k}} \quad (9)$$

另外，由于一个概念可能有多个属性，每个属性对概念的描述程度和作用也各不相同。如果每个属性都考虑，则计算

(下转第 191 页)

析阶段),使之能够适应本体协同进化环境。本文引入了一种本体变化生成图来表示知识工程师的本体改进意图,通过寻找变化生成图之间的冲突来发现知识工程师本体改进意图之间潜在的冲突,并通过对变化生成图进行冲突消解来解决知识工程师之间的冲突,从而使得多个本体改进意图能够共存,最终达到了本体协同进化的目标。在下一步的工作中,我们将以本文所提出的算法为基础,设计并实现一个本体协同进化系统,从而进一步完善本体协同进化的研究。

## 参 考 文 献


1 Fensel D. Ontologies: dynamics networks of meaning. In: Proceedings of the 1st Semantic web working symposium, Stanford, CA, USA, 2001
2 Stojanovic L. Methods and Tools for Ontology Evolution: [PhD thesis]. 2004: University of Karlsruhe, 2004
3 Pierre G, Steen M V. Dynamically selecting optimal distribution strategies on web documents. IEEE Transaction on Computers, 2002, 51(6): 637~651
4 Horrocks I, Patel-Schneider P F, Harmelen F V. From SHIQ and RDF to OWL: The Making of a Web Ontology Language. Journal of Web Semantics, 2003, 1(1)
5 Stojanovic L, et al. User-driven Ontology Evolution Management. In: European Conf Knowledge Eng. and Management (EKAW 2002). Springer-Verlag, 2002
6 Haase P, Stojanovic L. Consistent Evolution of OWL Ontologies. In: Proceedings of the Second European Semantic Web Conference. Heraklion, Greece, 2005
7 Plessers P, Troyer O D. Ontology Change Detection Using a Version Log. In: 4th International Semantic Web Conference, ISWC 2005. Galway, Ireland, 2005
8 Horrocks I, Patel-Schneider P F. Reducing OWL Entailment to Description Logic Satisfiability. Journal of Web Semantics, 2004, 1(4)
9 Yaorui L. Introduction to Artificial Intelligence. Beijing: Tsinghua University Press, 1997
10 Weimin Y. Data Structure. Beijing: Tsinghua University Press, 1993


(上接第 175 页)

量会大大增加。所以在计算属性相似度时,可以先依据机器学习方法计算出属性的信息增益[8],并以此为依据来确定各个属性的优先级。最后,只选取几个信息增益大的属性进行相似度的计算,这样可以减少计算量。

2.2.3 基于关系计算概念相似度

本体中的概念之间都存在一定的关系,概念的关系对概念的描述也具有重要的作用。关系有关系名称、关系类型、关系实例数据等要素,因此判断两个关系是否相似主要从这 3 个要素的相似度进行考虑。

设概念 $A$ 的关系为 $r_i$,概念 $B$ 的关系为 $s_j$,两个关系间的相似度记为 $RSim(r_i, s_j)$。关系相似度计算公式如下:

$$RSim(r_i, s_j) = W1 * Sim(r_{i_{name}}, S_{j_{name}}) + W2 * Sim(r_{i_{type}}, S_{j_{type}}) + W3 * Sim(r_{instance}, S_{j_{instance}}) \quad (10)$$

其中 $w1$、$w2$、$w3$ 是权重,代表关系名称、类型、数据对关系相似度计算的重要程度,$w1+w2+w3=1$。

关系名称、关系类型本身都是字符串,因此可以采用字符串相似度计算方法进行判定。对于关系的实例数据,也可以采用基于实例的方法进行计算。为此把相似度的定义扩展到关系相似度,并用基于实例的方法来计算两个关系基于实例的相似度。例如两个关系 $r_i$ 和 $s_j$,它们之间的相似度为 $RSim(r_i, s_j)$,即 $RSim(r_i, s_j) = P(r_i \cap s_j)/P(r_i \cup s_j)$。其中,$r_i$ 是本体 $O_1$ 中的关系,$r_i \subseteq A_1 \times A_2 \cdots \times A_m$,$A_k(k=1,\cdots,m)$是本体 $O_1$ 中的概念;$s_j$ 是本体 $O_2$ 中的关系,$s_j \subseteq B_1 \times B_2 \cdots \times B_n$,$B_k(k=1,\cdots,n)$是本体 $O_2$ 中的概念。当 $m$ 和 $n$ 都等于 2 时,$r_i$ 和 $s_j$ 都是一个二元关系。概念中的一个关系会连接两个概念中的所有实例。对于关系 $r_i(A_1, B_1)$和 $s_j(A_2, B_2)$,利用它们对应的概念实例进行相似度计算。

设概念 $A$ 和概念 $B$ 之间共计算出 $L$ 个 $RSim(r_i, s_j)$,并设置相应的权值 $w^l_{relation}$。概念 $A$ 和概念 $B$ 基于关系的相似度计算公式为

$$Sim_{relation}(A,B) = \frac{\sum_{k=1}^{l} W^{tu}_{k\,relation} RSim(r_i, S_j)}{\sum_{k=1}^{l} W^{k}_{relation}} \quad (11)$$

2.2.4 合并概念相似度

把基于实例、基于属性、基于关系计算得到的概念相似度进行合并,得到最后的概念相似度 $Sim(A,B)$,计算公式如下:

$$Sim(A,B) = w_{instance} Sim_{instance}(A,B) + w_{attribute} Sim_{attribute}(A,B) + w_{relation} Sim_{relation}(A,B) \quad (12)$$

其中,$w_{instance} + w_{attribute} + w_{relation} = 1$,权值的具体设置根据具体环境由用户确定。

## 3 性能分析

本文提出的综合的概念相似度计算方法通过概念名称的相似性过滤相关本体中某概念的候选概念集,可以大大减少相似度计算的次数。对于每一对基本相似的概念,采用了综合的相似度计算方法,虽然比单纯的基于实例的相似度计算公式计算量更多,但对于概念相似度的计算更能反映概念之间的相似关系。通过选择合适的权值,可以确保概念相似度的计算更全面、更准确。当然,在概念相似度的计算过程中存在大量的权值设定,可能对性能存在一定的影响。对于权值的选取,可以通过神经网络学习技术进行修正[7]。

**结束语** 本体是人和机器、程序间知识交流的语义基础,使用本体的目的是为了知识的共享和重用。然而由于各自建立的局限性和不同本体之间存在个体丰富性,本体间也就不可避免地存在着语义冲突,因而解决不同本体的概念间的语义冲突的本体映射成为本体研究领域的重要课题。本文针对目前本体映射中概念相似度计算所存在的问题,提出了一种综合的相似度计算方法。首先只对最相关的概念对计算相似度,减少相似度的计算;并从 3 个方面对概念进行相似度的计算,确保计算效果更加全面,计算结果更加准确。但是,计算过程中各个权值的设定还只是根据经验来给定,有一定的误差,应该对权值的设定做进一步的研究。另外,属性增益的计算在一定程度上增加了计算量,可以考虑采用其它更有效的机器学习方法来确定属性的优先级。

## 参 考 文 献


1 Doan A H. Learning to Map between Structured Representations of Data: [Ph thesis]. University of Washington, 2002
2 Noy N, Musen M. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In: Proc. AAAI2000. AAAI Press, 2000
3 Noy N F, Musen M A. SMART: Automated support for ontology merging and alignment [J]. In: Twelth Workshop on Knowledge Acquisition, Moeling, and Management, Banff, Canada, 1999
4 Dou D, McDermott D, Qi P. Ontology Translation by Ontology Merging and Automated Reasoning: [Ph thesis]. University of Yale, 2004
5 郑丽萍. 本体映射的研究. [硕士学位论文]. 山东科技大学, 2005
6 Kong C Y, Wang C L, Lau F C M. Ontology Mapping in Pervasive Computing Environment . URL. http://www. csis. hku. hk/~clwang/papers/EUC2004-laurel. pdf
7 蔡自兴, 徐光佑. 人工智能及其应用. 清华大学出版社, 2001
8 范明, 孟晓锋. 数据挖掘概念与技术. 机械工业出版社, 2001