

本体映射中的概念相似度计算

郑 诚, 秦多荣

(安徽大学 计算智能与信号处理教育部重点实验室, 安徽 合肥 230039)

摘 要:本体是概念、属性和关系的集合, 本体映射是解决本体异构的最好方法。文中针对目前本体映射过程中概念相似度计算存在的问题, 提出一种综合的相似度计算方法。先根据本体中两个概念名称的相似性, 选出最相关的概念, 减少相似度的计算, 然后分别基于概念的属性、实例和关系来计算概念相似度, 并进行综合得到概念相似度。在计算属性相似度时, 先通过计算属性的信息增益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, 从而减小计算量。

关键词:本体映射; 概念相似度; 属性相似度

中图分类号: TP181

文献标识码: A

文章编号: 1673-629X(2008)11-0125-03

Computation of Conceptual Similarity in Ontology Mapping

ZHENG Cheng, QIN Duo-rong

(Educational Department Key Laboratory of Intelligent Computing & Signal
Processing, Anhui University, Hefei 230039, China)

Abstract: Ontology is usually viewed as the sets of concepts, attributes and relations. ontology mapping is the best way to solve ontology heterogeneity. To aim at the current problems of the computation of concept similarity, puts forward an improved approach. Firstly, the most related concepts are filtered out according to the similarity between two concept names so as to reduce the similarity computation, and then an integrated approach of based - concept attribute, based - concept instance, based - concept relation is designed to compute the concept similarity. When calculating attribute similarity, determine the priority of various attributes by computing their entropies, then only compute the attribute similarity through several attributes with relative large entropy values to reduce the amount of computation.

Key words: ontology mapping; concept similarity; attribute similarity

0 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]。为了实现异构本体间的互操作, 一般可采用三种方法:

(1) 本体间建立包含关系;

(2) 本体间建立映射关系;

(3) 对本体进行合并, 生成一个完整的公共本体。

在这三种方法中, 本体映射是最有效的解决方法。本体映射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。它是本体结盟、本体集成、本体合并、本体翻译等的技术基础, 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五步来执行。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^[2]。目前一些具体的映射系统和实现方法已经被开发出来, 例如 Glue 系统^[3]、本体代数方法^[4]、MAFRA^[5]、IF-Map 和 OMEN 系统。

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上。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。

目前常用的计算相似度的方法有两种, 一种是根据概念的实例计算相似度, 一种是利用启发规则计算相似度, 但这两种方法都有缺点。首先, 根据概念的实例计算相似度时, 要对两个本体中的每个概念对进行相似度计算, 因此计算量很大。如, 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 $m \times$

收稿日期: 2008-02-02

基金项目: 国家自然科学基金资助项目(60475017); 安徽省高等学校自然科学研究项目(2006kj055B)

作者简介: 郑 诚(1964-), 男, 安徽歙县人, 副教授, 硕士生导师, 从事数据挖掘、机器学习研究。

n 次相似度,并形成 $m \times n$ 维的相似矩阵;其次,在计算相似度时,仅仅利用概念自身的语义进行相似度计算,而没有考虑概念的属性和关系对概念的描述作用;第三,在相似度计算过程中,若考虑概念的属性对概念的影响,由于每个概念都有若干个属性,这样势必会大大增加计算量。所以在本体映射时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。

针对以上问题,文中提出一种综合的相似度计算方法。首先,对于本体 O_1 中的一个概念 A ,不是比较本体 O_2 中所有概念,而是根据两个概念的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。其次,在计算概念相似度时基于概念的属性、实例和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。在计算基于概念属性的相似度时,先通过计算属性的信息增益^[6]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,这样可以减少相似度的计算量。

1 概念相似度的计算

1.1 确定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。为找出这些基本相似的概念,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算,提高映射的效率。假设本体 O_1 中的概念 A 和本体 O_2 中的概念 B ,由于概念 A 和 B 的名称均由字符串表示,所以可以根据字符串的匹配来度量概念 A 、 B 的名称相似性,假设概念 A 、 B 的名称分别为 a 、 b ,则概念 A 、 B 的名称相似性度量公式为:

$$\text{sim}_{\text{name}}(A, B) = 1 - \frac{\left(\sum_{i=1}^{\min(|a|, |b|)} f(i) \right) + ||a| - |b||}{\max(|a|, |b|)} \quad (1)$$

其中,当 $a[i] = b[i]$ 时, $f(i) = 0$, 否则 $f(i) = 1$ 。

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,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:

$\text{CandidateSet}(A) = B_{[1..k]} \cup \text{所有与 } B_{[1..k]} \text{ 概念存在关系的概念} \cup \text{所有 } B_{[1..k]} \text{ 概念的父概念} \cup \text{所有 } B_{[1..k]} \text{ 概念的子概念。}$

1.2 属性相似度计算

在本体中,属性对概念的描述起重要作用,每个属性也是一个概念。属性一般可分为文本(text)、数字(numeric)和日期(date)三类。不同类型的属性用不同的方法计算相似度。对于日期和数字型的属性,则可以根据数据类型匹配表、数据的取值范围或一定的对应关系来进行相似度的确定。文本的类型有多种,如果属性是字符型,则可以根据字符串的匹配方法来获得相似度。由于每个概念有若干个属性,计算每个属性对的相似度必然会加大计算量,因此,先利用决策树算法中的信息增益来确定属性的优先级,然后选择优先级比较高的属性进行计算,从而减少相似度的计算。

1.2.1 计算属性的信息增益

属性的信息增益可以按以下各步来计算:

(1) 按照信息论的定义,设 S 是 s 个数据样本的集合,类标号属性具有 m 类样本的训练数据集, s_i 是类 c_i ($i = 1, \dots, m$) 中的样本数,则一个给定的样本分类所需要的期望信息由公式(2)计算给出:

$$I(s_1, s_2, \dots, s_m) = \sum_{i=1}^m p_i \log_2(p_i) \quad (2)$$

其中 p_i 是任意样本属于 c_i 的概率,并用 $\frac{s_i}{s}$ 估计。

(2) 设属性 A 具有 v 个不同值 $\{a_1, a_2, \dots, a_v\}$, 可以用属性 A 将 S 划分为 v 个子集 $\{S_1, S_2, \dots, S_v\}$; 其中 S_j 包含 S 中在 A 上具有值 a_j 的样本。设 s_{ij} 是子集 S_j 中类 c_i 的样本数。由 A 划分成子集的熵(entropy)或期望信息由公式(3)计算:

$$E(A) = \sum_{j=1}^v \frac{s_{1j} + s_{2j} + \dots + s_{mj}}{s} I(s_{1j}, \dots, s_{mj}) \quad (3)$$

其中项 $\frac{s_{1j} + s_{2j} + \dots + s_{mj}}{s}$ 充当第 j 个子集的权,并且等于子集 S_j (即 A 的值为 a_j) 中的样本个数除以 S 中的样本总数。熵值越小,子集划分的纯度越高。对于给定的子集 S_j ,所需要的期望信息由公式(4)计算给出:

$$I(s_{1j}, s_{2j}, \dots, s_{mj}) = \sum_{i=1}^m p_{ij} \log_2(p_{ij}) \quad (4)$$

其中 $p_{ij} = \frac{s_{ij}}{|S_j|}$ 是 S_j 中的样本属于类 c_i 的概率。

(3) 最后属性 A 的信息增益由公式(5)给出:

$$\text{Gain}(A) = I(s_1, s_2, \dots, s_m) - E(A) \quad (5)$$

也就是说, $\text{Gain}(A)$ 是由于知道属性 A 的值而导致的熵的期望压缩。通过以上各步计算可获得概念每个属性的信息增益,并根据各值的情况设定相应的属性权值 $w_{\text{attribute}}^k$ 。

1.2.2 计算属性的相似度

属性本身也可以被当成一个概念,属性有属性名

称、属性数据类型等要素,因此对这 2 个要素的相似度进行考虑。属性名称本身都是字符串,因此可以采用字符串匹配的方法进行相似度计算;属性数据类型可以根据数据类型匹配^[7]来计算,如表 1 所示。

表 1 数据类型匹配

匹配值	实型	整型	字符型	日期型
实型	1	0.9	0.1	0.7
整型	0.9	1	0.1	0.8
字符型	0.1	0.1	1	0.1
日期型	0.7	0.8	0.1	1

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 $ASim(a_i, b_j)$ 。属性相似度计算由公式(6)给出:

$$ASim(a_i, b_j) = w_1 * Sim(a_{i_name}, b_{j_name}) + w_2 * Sim(a_{i_datatype}, b_{j_datatype}) \quad (6)$$

其中 w_1, w_2 是权重,代表属性名称和数据类型对属性相似度计算的重要程度, $w_1 + w_2 = 1$ 。假设根据信息增益选择出的属性,概念 A 和概念 B 间共计算出 m 个 $ASim(a_i, b_j)$, 并设置相应的权值 $w_{attribute}^k$, 则概念 A, B 的属性相似度由公式(7)计算得出:

$$Sim_{attribute}(A, B) = \frac{\sum_{k=1}^m w_{attribute}^k ASim(a_i, b_j)}{\sum_{k=1}^m w_{attribute}^k} \quad (7)$$

1.3 实例相似度计算

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。概念 A 和概念 B 的实例相似度^[8] 记为 $Sim_{instance}(A, B)$, 并由公式(8) 计算给出:

$$Sim_{instance}(A, B) = \frac{P(A \cap B)}{P(A \cup B)} = \frac{P(A, B)}{P(A, B) + P(\bar{A}, B) + P(A, \bar{B})} \quad (8)$$

$Sim_{instance}(A, B) \in [0, 1]$, 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。

1.4 关系相似度计算

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、关系类型等要素,因此对这 2 个要素的相似度进行考虑,关系名称、关系类型本身都是字符串,因此可以采用字符串匹配进行相似度计算。

设概念 A 的关系为 S_i , 概念 B 的关系为 T_j , 关系 S_i 和 T_j 相似度表示为 $RSim(S_i, T_j)$ 。关系相似度的计

算如公式(9) 所示:

$$RSim(S_i, T_j) = w_1 * Sim(S_{i_name}, T_{j_name}) + w_2 * Sim(S_{i_type}, T_{j_type}) \quad (9)$$

其中 w_1, w_2 是权重,代表关系的名称和类型对关系相似度计算的重要程度, $w_1 + w_2 = 1$ 。设概念 A 和概念 B 之间共计算出 n 个 $RSim(S_i, T_j)$, 并设置相应的权值 $w_{relation}^l$, 则概念 A 和概念 B 的关系相似度由公式(10) 可计算得出,并表示为 $Sim_{relation}(A, B)$:

$$Sim_{relation}(A, B) = \frac{\sum_{l=1}^n w_{relation}^l RSim(S_i, T_j)}{\sum_{l=1}^n w_{relation}^l} \quad (10)$$

1.5 合并概念相似度

把概念 A 和概念 B 的名称相似度、属性相似度、实例相似度和关系相似度按公式(11) 计算可得到总的概念相似度,并表示为 $Sim(A, B)$:

$$Sim(A, B) = w_{name} * Sim_{name}(A, B) + w_{attribute} * Sim_{attribute}(A, B) + w_{instance} * Sim_{instance}(A, B) + w_{relation} * Sim_{relation}(A, B) \quad (11)$$

其中, $w_{name} + w_{attribute} + w_{instance} + w_{relation} = 1$, 权值的具体设置值根据具体环境由用户来定。

2 性能分析

文中提出一种综合的概念相似度计算方法,通过概念名称的相似性过滤出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,分别对它们的属性、实例和关系进行相似度的计算,最后合并概念的名称、属性、实例和关系相似度得到一个较为全面的概念相似度,而且在属性相似度计算时,为减少计算量,先对各属性的优先级进行确定,然后选择出优先级较大的属性进行相似度计算,虽然这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,这种综合的计算方法也存在不足,首先,对属性的优先级进行确定,虽然减少了属性相似度的计算量,但增加了信息增益的计算,所以应该试用其它更有效的方法来对属性的优先级进行确定;另外,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。

3 结束语

本体是人和机器、程序间知识交流的语义基础,然而本体的建立是一个费时费力的过程,这就要求不同

(下转第 137 页)

4.1 安全性攻击实验

对完全随机选块嵌入水印后的图像进行安全性攻击,受到攻击像素点比例 P 取 $(0,1]$,量化步长 Δ 分别为 40,80,120,像素亮度改变量 α 等于 5 和 8 时 A1 和 A2 算法下结果如图 2 所示。其中,横坐标为受到攻击像素点比例,纵坐标为提取出水印的 NC 值。当 NC 大于 0.85 时,可以认为水印基本正确提取。由结果可以看出,原始算法在安全性攻击下,大部分攻击比例下不能够正确提取出水印,而改进的算法即使在攻击强度较大的情形下,也能完全正确提取。算法抗安全性攻击的能力随着量化步长的增大而增强,同时随着 α 的增大而减弱。

4.2 鲁棒性实验

改进的算法在 JPEG 压缩、高斯噪声、椒盐噪声、中值滤波、维纳滤波和剪切等常见图像处理下,提取水印与原始算法对比结果如表 1 所示。可以看到,对于这些攻击类型,除剪切外改进的算法鲁棒性不受影响。因为普通的攻击对像素点亮度的改变有正有负,所以图像整体平均亮度改变量很小,在进行亮度调整后,

表 1 鲁棒性攻击对比实验

攻击类型 \ 步长	A1			A2		
	40	80	120	40	80	120
JPEG(Q=20)	0.567	1	1	0.567	1	1
中值滤波(5×5)	0.8577	0.9321	0.9637	0.8577	0.9321	0.9637
维纳滤波 5×5	0.8389	0.9706	0.9956	0.8389	0.9706	0.9956
椒盐噪声(D=1%)	0.7183	0.8782	0.9644	0.7183	0.8782	0.9644
高斯噪声($\sigma^2=0.002$)	0.6668	0.9316	0.9920	0.6668	0.9316	0.9920
剪切 10%	0.9515	0.9525	0.9534	0.0437	0.9515	0.9646

(上接第 127 页)

本体间可以进行知识的共享和重用。但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。文中针对目前本体映射过程中概念相似度计算的存在的问题,提出一种基于名称、属性、实例和关系的综合相似度计算方法,确保本体映射更加全面而准确。

参考文献:

[1] Doan A H. Learning to Map between Structured Representations of Data[D]. Washington: University of Washington, 2002.

[2] Maedche A, Motik B. Ontologies for Enterprise Knowledge Management[J]. IEEE Intelligent Systems, 2003, 18(2): 26

($L2-L1$) 为一个很小的数,而图像像素亮度值 $S2$ 必须为整数,因此会出现取整误差,调整后的平均亮度值不完全等于攻击前的值。在这种情况下,改进的算法和原始算法实际上是一致的,所以鲁棒性不受影响。而剪切对图像像素亮度改变量非常大,进行亮度调整后,则可能出现很大误差。此外,只要嵌入水印图片较小,还可以对水印信息进行纠错编码,进一步提高算法的鲁棒性。

5 结束语

利用保持嵌入水印前后图像的平均亮度不变以及混沌加密和对图像完全随机分块的方法,增强了基于 DCT 直流系数量化的数字水印的安全性。实验结果证明了算法增强了安全性,且对鲁棒性基本无影响。

参考文献:

[1] 孙圣和,陆哲明,牛夏牧. 数字水印技术及应用[M]. 北京: 科学出版社,2004.

[2] Cox I J, Miller M L, Bloom J A. 数字水印[M]. 王 颖,黄志蓓,译. 北京:电子工业出版社,2003:198-199.

[3] 张 涛,汤光明,孙怡峰. 基于密码技术的数字水印研究[J]. 计算机工程与应用,2003(26):109-111.

[4] 陈 禧. 基于卷积码和软判决的图像水印盲检算法[J]. 电子技术应用,2007(1):142-144.

[5] Chan Chi-Kwong, Cheng L M. Security of Lin's image watermarking system[J]. The Journal of Systems and Software, 2002, 62:211-215.

[6] 杜鹏超,唐通林. 数字水印研究中常用的测试指标(上)[J]. 电子质量(测试技术与自动化卷),2002(11):7-10.

[7] Zheng Liping, Li Guangyao. Design of ontology mapping framework and improvement of similarity computation[J]. Journal of Systems Engineering and Electronics, 2007, 18(3): 641-645.

[8] 黄烟波,张红宇,李建华. 本体映射方法研究[J]. 计算机工程与应用,2005(18):27-33.