

基于领域本体的概念相似度算法研究

吴雅娟 王 鑫

(东北石油大学计算机与信息技术学院,大庆 163318)

摘 要 随着本体技术的逐渐成熟,如何为本体搭建语义桥梁以实现知识的重用与共享成为新的研究热点。在分析现有相关技术的基础上,提出一种计算不同本体中概念间语义相似度的方法,该方法以基于距离的概念相似度算法为基础,同时对概念结构进行分析将两者结合,从而计算出最终的概念间语义相似度。实验证明该方法有效。该研究工作可以应用于面向 Web 的知识检索领域。

关键词 领域本体 概念相似度 算法

中图法分类号 TP311.52; **文献标志码** A

近年来,本体已经成为语义 Web、人工智能、数据集成、信息检索等研究领域的热门课题。领域本体可以有效地组织领域中的知识,使知识更好地共享、重用。但是在利用本体的同时,如何提高概念相似度计算精度却成了本体应用的一个难题。例如,目前大多数的信息检索方法都是基于关键字进行检索,查准率不高。既然本体描述了数据的语义,则基于本体进行信息检索的检索效率显然要高,本体在信息检索中的应用能够显著地提高检索的精确率和返回率^[1]。在信息检索领域中,概念的语义相似度计算起着重要的作用。因此需研究基于领域本体的计算概念语义相似度的计算方法。

1 相关背景及研究工作

1.1 概念语义相似度

同样的词语在不同的上下文中可能会有不同的语义,即语义多元化。在已经对词语排除歧义的

基础上,对概念的语义进行比较。语义相似度在不同的应用领域中可能会有不同的含义。例如,在信息整合领域中,相似度一般指的是文本与文本能够匹配的程度;而在信息检索领域中,相似度则反映与用户查询在语义上的匹配程度,相似度越高,表明该文本与用户的请求越接近^[2]。

工作背景是信息检索领域。现约定,相似度的取值范围在 0~1 之间。当比较的 2 个概念完全相同的时候,其相似度为 1;反之,当比较的 2 个概念没有任何关联的时候,其相似度为 0;在其他情况下,即比较的两个概念之间有一定的关联的情况下,其相似度在 0 到 1 之间。

1.2 相关研究

对于概念的语义相似度计算,国外许多研究者利用了语义词典 Wordnet 中的同义词集组成的树状层次体系结构^[3],一种方法是考虑两个概念共享信息的程度,基于信息理论定义相似度计算方法;另一种采用了先计算两概念在树中的信息熵或语义距离,然后转化为语义相似度的办法。在国内,相关研究起步相对较晚。具体而言,文献[4,5]中,首先计算两个概念在树中的语义距离,然后转换为 2 个概念间语义相似度;文献[6]对概念实例采用联合分布概率统计的方法,确定概念间语义相似度;

2010年5月24日收到 黑龙江省自然科学基金(F2007—11)资助
第一作者简介:吴雅娟(1966—),女,黑龙江望奎人,硕士,教授,研究方向:人工智能与数据挖掘,计算理论与算法。

*通信作者简介:王 鑫(1984—)女,硕士生,研究方:为人工智能与数据挖掘。

文献[2]运用基于概念实例的相似算法,再结合概念层次树中影响相似度的两个因素,最后得到不同本体间两个概念的相似度;文献[7]提出了概念结构相似度的基本思想和相应公式。

上述实验结果都与人的主观判断的结果相符。但是文献[4,5]是基于一个本体中,内部概念的相似度计算,并没有涉及到多个不同本体间的概念相似度计算;文献[7]对概念实例的范畴划分过于绝对,对于不属于该概念范畴的部分实例,与该概念可能存在一定的相似,这些相似度被忽略了;文献[2]通过计算概念子概念间的相似度而得出概念的总相似度,并未考虑概念之间的父概念及兄弟概念之间的相似性;文献[8]提出了概念结构相似度的基本思想,但给出的公式还不够精细。

针对上述情况,现对文献[2,7]中的公式进行改进,并结合一种新的基于距离的概念相似度算法从而得到计算概念相似度的新算法。

2 基本概念

根据 studder 的定义,本体是共享概念模型的明确形式化规范说明,它提供了一种明确的形式化的领域知识描述手段,同时支持对隐含知识进行推理,在信息集成和知识管理等领域发挥着重要的作用。本体的形式化的定义为 $O = \{C, R, H^C, H^R, A, I\}$,其中 C 是领域概念的集合; R 为概念间的层次关系; H^C 为概念间的其它关系,如 Same As 关系,Part Of 关系,Contains 关系等,它们是概念集笛卡尔积的子集; H^R 定义了关系之间的层次结构; A 是公理的集合,代表永真断言, I 是本体实例的集合。概念是客观世界任何事物的抽象描述。如事物、功能、行为、过程、策略等,语义上它表示对象的集合^[8]。现定义概念为一个四元组: $C = \{i, L, P, I^c\}$,其中 i 为概念的唯一标识符,用 URI 表示, L 为概念的语言词汇, P 为概念所拥有的属性的集合, I^c 为属于该概念的实例的集合。当两个本体元素具有某些共同特征时,则定义它们是相似的。相似的程度用相似度来表示。

(1) $\text{sim}(x, y) \in [0, 1]$ 。相似度的计算值为 $[0, 1]$ 区间中的一个实数。

(2) $\text{sim}(x, y) = 1$ 当且仅当 $x = y$ 。如果两个对象是完全相似的,则相似度为 1。

(3) $\text{sim}(x, y) = 0$ 。如果两个对象没有任何共同特征,那么其相似度为 0。

(4) $\text{sim}(x, y) = \text{sim}(y, x)$ 。相似关系是对称的。

在有的研究中如自然语言处理,常采用距离的概念。一般说来,两个词汇的距离越大,其相似性越小。

3 相似度计算

3.1 基于距离的概念相似度算法

比较同一个本体中两个概念 C_1, C_2 的相似度。定义 C_f 为 C_1 和 C_2 的最近公共父结点。定义 d_{c_1}, d_{c_2} 分别为从 C_1, C_2 到 C_f 的结点数。 w_1, w_2 分别为 C_1, C_2 的权值。 D 为树的最大深度。 $\text{dep}(C_1)$ 表示节点 C_1 在树中的深度即层次。计算概念 C_1, C_2 的相似度:

$$w_1 = \frac{\text{Dep}(C_1)}{\text{Dep}(C_1) + \text{Dep}(C_2)} \quad (1)$$

$$w_2 = 1 - w_1 \quad (w_1, w_2 > 0)$$

$$\text{sim}(C_1, C_2) = 1 - \frac{w_1 d_{c_1} + w_2 d_{c_2}}{2Dw_1w_2} \quad (2)$$

$$\text{sim}(C_1, C_2) \in [0, 1]。$$

以上是比较同一个本体中两个不同概念的相似度,现在再对不同本体间的两个概念进行相似度的比较。

(1) 定义概念 C_1, C_2 分别来自两个本体 O_1, O_2 。定义 R 为 C_2 的所有父节点,子节点,兄弟节点的集合。条件如下:

(i) 若 a 是 b 的父结点。如果存在结点集合 $n_1, n_2, n_3, \dots, n_k | n_i$ 是 n_{i+1} 的直接父结点,即

$$a = n_1, b = n_k。$$

(ii) 若 a 是 b 的子结点,如果存在结点集合 $n_1, n_2, n_3, \dots, n_k | n_i$ 是 n_{i+1} 的直接子结点,即

$$a = n_1, b = n_k。$$

(iii) 如果 a, b 共有直接父结点, 则 a, b 互为直接兄弟结点。

(2) 计算 C_2 到 x 的距离 d :

(i) 定义 S 为 x 的集合; x 为概念 $|x \in \mathbf{R}$ 并且 $x \in O_1$ 。

(ii) d 为从 C_2 到 x 所经历的结点数。

(3) 重复步骤(2)得到最小距离 d 。

(4) 通过式(2)计算 C_1 与 S 中每个 x 的相似度 (C_1 与 x 同属 O_1)。

(5) 通过步骤(3)中的最小距离 d 来定义一个误差值 g 。

(6) 用 $\text{sim}M(C_1, x)$ 来表示在步骤(4)中得到的 C_1 与 x 的最大相似度值。

(7) 最终得到 C_1, C_2 的相似度。

$$\text{sim}(C_1, C_2) = \text{sim}M(C_1, x) - g \quad (3)$$

3.2 概念结构相似度算法

定义结构相似度 $S_{CH}(S_{ci}, S_{cj})$ ^[8]

$$S_{CH}(S_{ci}, S_{cj}) = (\alpha S_{CHf}(S_{ci}, T_{cj}) + \beta S_{CHb}(S_{ci}, T_{cj}) + \lambda S_{CHs}(S_{ci}, T_{cj})) / (\alpha + \beta + \lambda) \quad (4)$$

式(4)中 $S_{CHf}(S_{ci}, T_{cj})$ 表示概念 S_{ci}, T_{cj} 的父概念之间的相似度; $S_{CHb}(S_{ci}, T_{cj})$ 表示概念 S_{ci}, T_{cj} 的兄弟概念集之间的相似度; $S_{CHs}(S_{ci}, T_{cj})$ 表示概念 S_{ci}, T_{cj} 的子概念集之间的相似度。 α, β, λ 表示权重因子, 因考虑到层次结构中父子、兄弟概念对其相似度的影响是不同的, 故在此赋值的权值大小为 $\alpha \geq \beta \geq \lambda \geq 0$ 。

具体如下:

$$S_{CHf}(S_{ci}, T_{cj}) = \text{sim}(S_{cif}, T_{cjf}) \quad (5)$$

式(5)中, S_{cif}, T_{cjf} 分别表示概念 S_{ci}, T_{cj} 的父概念。

$$S_{CHb}(S_{ci}, T_{cj}) =$$

$$\sum_{S_{cibn} \in S_{cib}} \max_{T_{cjb n} \in T_{cjb}} (\text{sim}(S_{cibn}, T_{cjb n})) / 2\text{Wid}(S_{cib}) + \sum_{T_{cjb n} \in T_{cjb}} \max_{S_{cibm} \in S_{cib}} (\text{sim}(T_{cjb n}, S_{cibm})) / 2\text{Wid}(T_{cjb}) \quad (6)$$

式(6)中, $m \in (0, \text{Wid}(S_{cib}))$, $n \in (0, \text{Wid}(T_{cjb}))$; S_{cib}, T_{cjb} 分别表示概念 S_{ci}, T_{cj} 的兄弟概念集; $S_{cibm}, T_{cjb n}$ 表示 S_{cib}, T_{cjb} 中的概念; $\text{Wid}(S_{cib}), \text{Wid}(T_{cjb})$ 分别为 S_{ci}, T_{cj} 所拥有的兄弟概念的个数。

$$S_{CHs}(S_{ci}, T_{cj}) =$$

$$\sum_{S_{ciscg} \in S_{cis}} \max_{T_{cjs h} \in T_{djs}} (\text{sim}(S_{ciscg}, T_{cjs h})) / 2\text{Wid}(S_{cis}) + \sum_{T_{cjs h} \in T_{cjs}} \max_{S_{ciscg} \in S_{cis}} (\text{sim}(T_{cjs h}, S_{ciscg})) / 2\text{Wid}(T_{cjs}) \quad (7)$$

其中, $g \in (0, \text{Wid}(S_{cis}))$, $h \in (0, \text{Wid}(T_{cjs}))$; S_{cis}, T_{cjs} 分别表示概念 S_{ci}, T_{cj} 的子概念集; $S_{ciscg}, T_{cjs h}$ 表示 S_{cis}, T_{cjs} 中的概念; $\text{Wid}(S_{cis}), \text{Wid}(T_{cjs})$ 分别为 S_{ci}, T_{cj} 所拥有的子概念的个数。

3.3 相似度影响因子

本体中概念的层次结构越接近, 其相似度越大, 因而 2 个概念在其相应概念树中所处的层次差越小, 其相似度越高, 针对概念在概念树中所处的层次深度差对概念间语义相似度的影响, 本文引入系数 x 如下:

$$x = t \sqrt{1 - \frac{|\text{Dep}(S_{ci}) - \text{Dep}(T_{cj})|}{\text{Dep}(S_{ci}) + \text{Dep}(T_{cj})}} \quad (8)$$

式(8)中 t 为可调节参数, $\text{Dep}(S_{ci})$ 表示 S_{ci} 在 S 概念树中的层次 (考虑到多重继承的问题, 这里选用从该结点到根结点的最长路径来计算), $\text{Dep}(T_{cj})$ 表示 T_{cj} 在 T 概念树中的层次, 概念树的根结点层次为 0。

同样, 两个概念的兄弟节点个数相对较大, 也即, 它们的父节点分类细致程度较高, 则这对概念的语义会较为接近, 所以一个概念的分类细致程度也应该是计算语义距离时应考虑的一个因素。引入系数 y 如下。

$$y = k \sqrt{1 - \frac{|\text{Wid}(S_{ci}) - \text{Wid}(T_{cj})|}{\text{Wid}(S_{ci}) + \text{Wid}(T_{cj})}} \quad (9)$$

式(9)中 k 为可调节参数。

3.4 总相似度

将式(5)、式(6)和式(1)结合得到式(7)如下

$$\text{Sim}(S_{ci}, T_{cj}) = (\alpha S_{CHf}(S_{ci}, T_{cj}) + \beta S_{CHb}(S_{ci}, T_{cj}) + \lambda S_{CHs}(S_{ci}, T_{cj}) xy) / (\alpha + \beta + \lambda) \quad (10)$$

$\text{Sim}(S_{ci}, T_{cj})$ 即为最终的概念间语义相似度。

4 实验结果

4.1 实验评价准则

本文采用信息检索领域查全率和查准率作为评价映射算法的主要准则,并定义如下:

(1) 概念查全率 (Recall)

$$r = \frac{\text{正确发现的概念对}}{\text{可能存在的概念对}}^{\circ}$$

(2) 概念召回率 (Precision)

$$p = \frac{\text{发现的正确概念对}}{\text{所有发现的概念对}}^{\circ}$$

4.2 实验结果

实验利用本体建模工具 Protégé3.1 创建了油田地质领域的两个不同的本体,分别包括 300 多个概念和关系,对照专家经验,按照本文的算法,取不同的加权系数,相似度计算结果比使用单一的算法计算的结果更接近实际情况。实验结果表明改进后的算法提高了计算精度。

5 结束语

领域本体在知识的共享和重用中起到关键的作用。然而由于各自建立适合自身的本体,使不同本体之间存在个体差异性,本体间也就不可避免地存在着语义冲突,研究者使用本体概念的相似度值判断两个概念间的语义关系。本文针对目前概念

相似度计算所存在的问题,提出了一种新的综合的相似度计算方法。从概念的结构不同层次分别计算概念的相似度,然后加权平均求出综合的概念相似度,提高了概念映射的查全率。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,对权值的设定也可以使用 sigmoid 函数自动选择。另外,在计算概念的相似度时,没有考虑概念名称、属性、实例等的相似度,而属性对概念的影响因素是很重要的,还需做深入的研究。

参 考 文 献

- 1 Uarino N, Masolo C, Gverter. Onto Seek: Content - based Access to the Web IEEE Intelligent Systems, 1999; 14(3): 70—80
- 2 王家琴, 李仁发, 李仲生, 等. 一种基于本体的概念语义相似度方法的研究. 计算机工程, 2007; 33(11): 201—203
- 3 Evaluating Word Net2 based measures of lexical semantic relatedness. Computational Linguistics, 2004; 1(1): 1—49
- 4 徐德智, 郑春卉, Passi K. 基于 SUMO 的概念语义相似度研究. 计算机应用, 2006; 26(1): 180—183
- 5 吴 健, 吴朝晖, 李 莹. 基于本体论和词汇语义相似度的 Web 服务发现. 计算机学报, 2005; 28(4): 595—602
- 6 Doan A H, Madhavan J, Domingos P. Learning to map between Ontologies on the semantic Web. // Proceedings of the 11th International Conference on World Wide Web, New York, USA: ACM Press, 2002: 662—673
- 7 徐德智, 肖文芳, 王怀民. 本体映射过程中的概念相似度计算. 计算机工程与应用, 2007; 43(9): 167—169
- 8 程 勇, 黄 河, 邱莉榕, 等. 一个基于相似度计算的动态多维概念映射算法. 小型微型计算机系统, 2006; 27(6): 975—979

Research on Concept Similarity Based on Domain Ontology

WU Ya-juan, WANG Xin

(School of Computer & Information, Northeast Petroleum University, Daqing 163318, P. R. China)

[Abstract] With the gradual maturing of ontological technology, how to establish the semantic bridge among ontologies is becoming the new hotspot of current researches in order to reuse and share knowledge. Based on analyzing current technologies, a method which based on distance - based algorithm between different concepts and analyses and combine their structure is proposed to calculate the final concept similarity. It has been proved to be feasible by experiments. The research can be applied in the field of knowledge searching.

[Key words] domain ontology concept similarity algorithm