改进的多策略的概念相似度计算方法①

孙海真, 谢颖华

(东华大学 信息科学与技术学院, 上海 201620)

摘 要: 本体映射是解决本体异构的有效手段, 而概念相似度计算是本体映射的关键环节. 针对目前本体映射中概念相似度计算存在的问题, 提出一种改进的多策略的概念相似度计算方法. 首先根据两个概念的名称相似度进行初始映射判断, 然后基于概念的属性、结构、实例计算概念相似度, 并选取适当的权值进行加权综合. 最后采用 OAEI 提供的标准数据测试集 benchmark 进行实验. 实验结果表明, 该方法在保证映射效率和通用性的同时, 提高了映射结果的查全率和查准率.

关键词: 本体; 本体映射; WordNet; 概念相似度; 多策略

Improved Multi-strategy Concept Similarity Calculation Method

SUN Hai-Zhen, XIE Ying-Hua

(Department of Information Science and Technology, Donghua University, Shanghai 201620, China)

Abstract: Ontology mapping is the key technology to solve the heterogeneous problem among ontology, and concept similarity computation is a crucial step in ontology mapping process. In view of the problems existed in ontology mapping, an improved multi-strategy ontology mapping model is proposed. The most related concepts are filtered based on similarity of concept names. Then we calculate concept similarity by combining property-based similarity, structure-based similarity and instance-based similarity. Finally, we use the test set called benchmark provided by OAEI to test the performance of our mapping algorithm. The experimental results show that this method improves the recall and precision of ontology mapping while maintaining mapping efficiency and currency.

Key words: ontology; ontology mapping; WordNet; concept similarity; multi-strategy

近年来, 随着语义网[1]的发展, 本体在信息集成、信息检索及数字图书馆等领域得到广泛应用, 导致本体数量迅猛增长. 然而由于领域专家们基于各自不同的应用需求, 采用不同的组织方式、逻辑结构来描述本体, 造成了本体异构. 本体映射的目的是在异构本体之间建立交互规则, 使得异构本体在语义级别上能够沟通和互操作. 由于本体映射主要描述两个异构本体概念之间对应关系, 因此两个概念间的相似度计算是本体映射的核心. 目前大部分本体映射方法都涉及到具体的概念, 根据这些方法基于概念的不同特征, 我们大致将其划分为四类[2]. 第一类是概念名称的相似度计算, 如果两个概念名称的表示形式是相同的或相近的, 那么其代表的含义存在一定的相似关系. 应用该思想的方法有编辑距离(Edit Distance), 知网(HowNet), WordNet 等. 第二类是概念实例的相似度计算, 如果两个概念相同的实例比重较大, 则这两个概念可能是相似的. 利用这种思想的技术有贝叶斯分类技术、支持向量机等. 第三类是概念结构的相似度计算, 这种方法的主要思想是概念节点的父节点、子节点、兄弟节点的相似度越高, 则当前节点越相似. 第四类是概念属性的相似度计算, 主要基于下述考虑: 如果两个概念的属性相似度超过一定的阈值, 则这两个概念是相似的.

目前本体映射算法多数是通过对不同概念相似度计算方法的结果直接加权得到映射概念对[3]. 因此存在各映射策略权重分配不完善的问题, 并且每种概念

① 收稿时间:2014-10-24;收到修改稿时间:2014-11-28

相似度计算方法都有其特定的优势和劣势, 单纯通过结合不足以充分利用概念的各个特征来考察它们之间的语义关系. 这种结合方式不仅没有充分利用各个方法的映射结果集, 而且加大了计算量, 严重影响了映射效率. 针对上面本体映射中存在的问题, 本文提出了一种改进的多策略的概念相似度计算方法: 1)提出一种综合的本体概念相似度计算方法-NPSI 模型(N 代表名称(Name), P 代表属性(Property), S 代表结构(Structure), I 代表实例(Individuals)); 2)对概念名称相似度计算加以改进, 使得映射结果更加准确.

1 概念相似度计算的本体映射方法

1.1 改进的多策略的本体映射框架

首先, 我们从本体 O1 和本体 O2 中提取待映射概念对, 计算基于概念名称的相似度, 将满足一定阈值条件的映射结果输出, 完成两个本体概念间的初始映射发现. 随后对于不满足映射条件的概念, 计算基于属性、结构及实例的概念相似度, 并选择合适的权值加权, 更新概念相似度矩阵, 输出最终的映射结果. 图 1 是改进的多策略的本体映射框架.

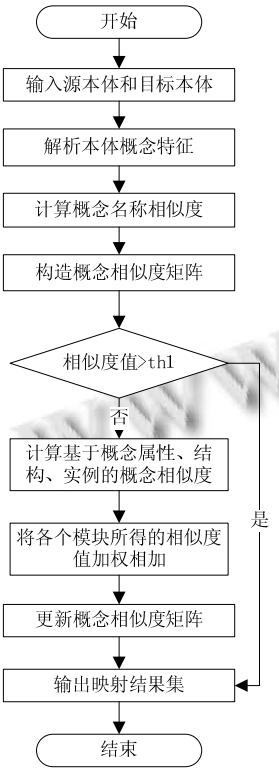

图 1　改进的多策略的本体映射框架

1.2 改进的基于名称的概念相似度计算

本文概念名称相似度的计算结合编辑距离、WordNet 语义词典和类的注释信息相似度这三个方面, 得到概念名称相似度的匹配公式, 如下所示:

$$\begin{aligned} Sim_{name}(c_1,c_2) = {} & w_1 Sim_{Edit}(c_1,c_2) \\ & + w_2 Sim_{wordNet}(c_1,c_2) \\ & + w_3 Sim_{comment}(c_1,c_2) \end{aligned} \tag{1}$$

其中, $Sim_{name}(c_1,c_2)$ 表示概念名称相似度, $Sim_{edit}(c_1,c_2)$ 表示用编辑距离算法得到的概念 c_1 和 c_2 的相似度, $Sim_{WordNet}(c_1,c_2)$ 表示利用 WordNet 语义词典得到的概念 c_1 和 c_2 的相似度, $Sim_{comment}(c_1,c_2)$ 表示考虑类的注释信息得到的概念 c_1 和 c_2 的相似度. w_1、w_2、w_3 分别代表三种相似度的权重, 且 $w_1+w_2+w_3=1$.

这部分权值的计算引入 sigmoid 函数. 本文采用的 sigmoid 函数形式见式(2):

$$f(x)=\frac{1}{1+e^{-5(x-0.5)}} \tag{2}$$

其中, $0\le x\le 1$, x 表示各个策略得到的相似度值, f(x) 为各个策略对应的初始权值, 分别设为 f_1,f_2,f_3. 则有 $w_1=\frac{f_1}{f_1+f_2+f_3}, w_2=\frac{f_2}{f_1+f_2+f_3}, w_3=\frac{f_3}{f_1+f_2+f_3}$.

1)基于编辑距离的相似度计算

本文基于语法的相似度计算主要采用编辑距离[4](Edit distance, 又称 Levenshtein distance)的方法. 编辑距离指两个字符串之间, 由一个转化成另一个所需进行的最少编辑操作次数. 常见的编辑操作包括删除、插入、替换等. 其计算公式为:

$$Sim_{edit}(c_1,c_2)=\max\left(0,1-\frac{ed(c_1,c_2)}{\min(|c_1|,|c_2|)}\right) \tag{3}$$

其中 c_1,c_2 分别表示待比较的源字符串和目标字符串, $ed(c_1,c_2)$ 表示字符串 c_1,c_2 的编辑距离. $|c_1|,|c_2|$ 分别表示字符串 c_1,c_2 中字符的个数.

2)基于 WordNet 的相似度计算

本文计算两个概念名称的语义相似度采用 WordNet 语义词典. 基于 WordNet 的语义相似度计算的原理是: 如果两个单词通过上位关系(hypernym)连接的距离越近, 那么它们的相似度越大; 反之, 它们的相似度越小. 如果它们在有限的上位层次中没有公共的父结点, 则 $Sim_d(w_1,w_2)=0$. Lin[5]等人在计算两个概念的相似度时提出了利用概率的方法, 其公式如下所示:

$$Sim_{Lin}=\frac{2\times\log(p(LCA(c_1,c_2)))}{\log(p(c_1))+\log(p(c_2))} \tag{4}$$

p(c) 表示概念节点 c 的所有子节点个数与 WordNet 所有子节点总个数的比. $p(LCA(c_1,c_2))$ 为本体中概念 c_1,c_2 的最低公共父节点的所有子节点与 WordNet 中所有节点总个数的比. 当概念 $LCA(c_1,c_2)$ 为根节点时, $p(LCA(c_1,c_2))$ 为 1, 其信息量为 0; 而当节点越深时, $p(LCA(c_1,c_2))$ 相对越小, 包含的信息量就越多, 相似度越大. 因此可用概念的信息量作为度量概念相似度的因素.

3) 基于注释的相似度计算

为了充分利用本体概念的描述信息, 本文引入了概念注释的相似度计算. 若待映射本体中两个概念的注释信息相同, 则认为两个概念是相似的, 记 $Sim_{comment}(c_1,c_2)=1$; 若待映射本体中两个类的注释信息不同, 则认为两个概念不相似, 记 $Sim_{comment}(c_1,c_2)=0$.

1.3 基于属性的概念相似度计算

在本体中, 概念的属性可以分为两类: 一类是数据类型属性(Datatype Property), 一类是对象类型属性(Object type Property). 基于属性计算概念相似度的基本思路是分别计算数据类型属性和对象类型属性的相似度, 然后为两种属性的相似度设定权值, 得到最终的基于属性的概念相似度[6].

其中对于单个属性的名称相似度计算本文从语法和语义两方面考虑, 结合公式(3)和公式(4)得出如下综合的属性名称相似度计算公式:

$$Sim_{name}(p_1,p_2)=w_1Sim_{Edit}(p_1,p_2)+w_2Sim_{wordNet}(p_1,p_2) \tag{5}$$

其中, $Sim_{name}(p_1,p_2)$ 表示属性名称相似度, $Sim_{edit}(p_1,p_2)$ 表示用编辑距离算法得到的属性 p_1 和 p_2 的相似度, $Sim_{WordNet}(p_1,p_2)$ 表示利用 WordNet 计算得到属性 p_1 和 p_2 的相似度. w_1、w_2 分别代表两种相似度计算方法的权重, $w_1=\frac{f_1}{f_1+f_2}, w_2=\frac{f_2}{f_1+f_2}$, f_1,f_2 为采用公式(2)得到的初始权值.

1) 对象类型属性相似度计算

设概念 A 的对象类型属性集合为 $PO_A=\{a_1,a_2,\cdots,a_m\}$, 概念 B 的对象类型属性集合为 $PO_A=\{b_1,b_2,\cdots,b_n\}$, 其中 m,n 分别是概念 A 和 B 的对象类型属性个数.

Step1: 采用公式(5)计算出概念 A 的对象类型属性 a_i 和概念 B 的对象类型属性 b_j 的相似度, 记作 SO_{ij}, 得到相似度矩阵:

$$SO=\begin{bmatrix} SO_{11} & SO_{12} & \cdots & SO_{1n} \\ SO_{21} & SO_{22} & \cdots & SO_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ SO_{m1} & SO_{m2} & \cdots & SO_{mn} \end{bmatrix}$$

Step2: 遍历矩阵 SO 取得相似度最大的 SO_{ij}, 将 SO_{ij} 所在的行和列删除, 在余下的矩阵中继续重复执行直到矩阵为空, 得到相似度最大的序列为 $p_1,p_2,\cdots,p_k(k=\min(m,n))$;

Step3: 最终得到的对象类型属性相似度为:

$$SO(A,B)=\frac{1}{k}\sum_{i=1}^{k}p_i \tag{6}$$

2) 数据类型属性相似度计算

Step1: 将本体 O_1 中概念 A 的数据类型属性按数据类型分类, 这样概念 A 的数据类型属性被分为若干个属性集合; 同理, 将本体 O_2 中概念 B 的数据类型属性按数据类型分类;

Step2: 数据类型属性名称的相似度按式(5)进行计算, 构造概念 A 和概念 B 的属性名称相似度矩阵;

Step3: 遍历属性相似度矩阵, 取得最大的相似度值, 将其所在的行和列删除, 在余下的矩阵中重复执行直到矩阵为空, 得到相似度序列记为 $p_1,p_2,\cdots,p_k(k=\min(m,n))$.

Step4: 按数据类型计算属性相似度的平均值;

Step5: 计算所有数据类型属性的名称相似度的平均值, 记为 SD(A,B);

3) 将概念的数据类型属性和对象类型属性相似度进行加权相加, 得到最终两个概念的属性相似度计算公式:

$$Sim_{property}(A,B)=w_1SD(A,B)+w_2SO(A,B) \tag{7}$$

w_1,w_2 代表两种类型的属性对整个属性相似度的权值, 且 $w_1+w_2=1$, 本文令 $w_1=w_2=0.5$.

1.4 基于结构的概念相似度计算

本文中概念结构相似度计算考虑概念的直接父概念、直接子概念和兄弟概念. 参考文献[7], 给出基于结构的启发规则:

1) 如果两个概念相似, 那么它们的子概念在一定程度上也相似;

2) 如果两个概念的子概念相似, 那么这两个概念也相似;

3) 如果两个概念具有相似的兄弟概念, 则这两个概念也相似;

4) 如果概念对中的概念 A 和概念 B 都有多个子

概念, 其中概念 A 有子概念 $\{A_1, A_2, \cdots A_n\}$, 概念 B 有子概念 $\{B_1, B_2, \cdots B_n\}$, 则对概念 A 中子概念与概念 B 中的子概念分别计算相似度, 采用加权相加的方法计算出概念 A 和概念 B 的子概念相似度;

5) 如果概念对中的概念 A 和概念 B 都有多个兄弟概念, 则采用与 4)类似的方法进行处理.

结构相似度的具体公式如下所示:

$$\begin{aligned} Sim_{structure}(c_1,c_2) &= w_1 Sim_{parent}(c_1,c_2) \\ &+ w_2 Sim_{child}(c_1,c_2) \\ &+ w_3 Sim_{brother}(c_1,c_2) \end{aligned} \tag{8}$$

其中, $Sim_{parent}(c_1,c_2)$ 表示概念 c_1, c_2 父节点的相似度, $Sim_{child}(c_1,c_2)$ 表示概念 c_1, c_2 子节点的相似度, $Sim_{brother}(c_1,c_2)$ 表示概念 c_1, c_2 兄弟节点的相似度, w_1, w_2, w_3 为权重因子. 由于在概念的层次结构中, 父、子、兄弟节点对概念相似度的影响不同, 而父节点占有绝对的权重, 因此预定 $w_1 \geq w_2 \geq w_3$.

1.5 基于实例的概念相似度计算

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相同, 那么这两个概念是相似的. 本文利用 Jaccard 公式[8]计算概念 A 和概念 B 的实例相似度, 记为 $Sim_{instance}$, 计算公式为:

$$\begin{aligned} Sim_{instance}(A,B) &= \frac{P(A \cap B)}{P(A \cup B)} \\ &= \frac{P(A,B)}{P(A,B) + P(A,\overline{B}) + P(\overline{A},B)} \end{aligned} \tag{9}$$

其中, P(A,B) 表示待映射本体的两个概念实例集中某实例既属于概念 A 的实例, 也属于概念 B 的实例的概率; $P(\overline{A},B)$ 表示实例集中某实例属于概念 B 的实例却不属于概念 A 的实例的概率; $P(A,\overline{B})$ 表示实例集中某实例属于概念 A 的实例但不属于概念 B 的实例的概率.

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 $N(U_i^{A,B})$ 表示在 U_i 中既属于概念 A 又属于概念 B 的实例的个数. 本文以 P(A,B) 的计算为例进行说明, 具体步骤如下:

a. 将本体 O_1 中的实例集 U_1 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 $U_1^{\overline{A}}$;

b. 分别将 U_1^A 和 $U_1^{\overline{A}}$ 中的实例作为正反样本, 使用机器学习的方法来训练对于概念 A 的学习器 L;

c. 将本体 O_2 中的实例集 U_2 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 $U_2^{\overline{B}}$;

d. 对实例集 U_2^B 中的实例使用学习器 L 进行分类, 得到 $U_2^{A,B}$和$U_2^{\overline{A},B}$ 两个实例集. 类似地, 对实例集 $U_2^{\overline{B}}$ 中的实例使用机器学习器 L 进行分类, 得到 $U_2^{A,\overline{B}}$和$U_2^{\overline{A},\overline{B}}$ 两个实例集. 由此得到 $U_2^{A,B}, U_2^{\overline{A},B}$, $U_2^{A,\overline{B}}$和$U_2^{\overline{A},\overline{B}}$ 四个实例集;

e. 将本体 O_1 和 O_2 的位置进行调换, 重复以上各步, 可以得到实例集 $U_1^{A,B}, U_1^{\overline{A},B}, U_1^{A,\overline{B}}$和$U_1^{\overline{A},\overline{B}}$;

f. 从各步计算中分别求得 $N(U_1), N(U_2), N(U_1^{A,B})$和$N(U_2^{A,B})$, 由下面公式计算一个实例在本体中既属于概念 A 又属于概念 B 的可能性:

$$P(A,B) = \frac{[N(U_1^{A,B}) + N(U_2^{A,B})]}{[N(U_1) + N(U_2)]} \tag{10}$$

用同样的方法计算得到 $P(A,\overline{B})$ 和 $P(\overline{A},B)$, 计算公式分别为:

$$P(A,\overline{B}) = \frac{[N(U_1^{A,\overline{B}}) + N(U_2^{A,\overline{B}})]}{[N(U_1) + N(U_2)]} \tag{11}$$

$$P(\overline{A},B) = \frac{[N(U_1^{\overline{A},B}) + N(U_2^{\overline{A},B})]}{[N(U_1) + N(U_2)]} \tag{12}$$

因此, 可以根据待映射本体中概念 A 和概念 B 所拥有的具体实例来计算 P(A,B)、$P(A,\overline{B})$ 和 $P(\overline{A},B)$ 的值, 从而得出概念间实例的相似度.

1.6 概念相似度结合策略

1) 基于名称的概念相似度计算方法考虑概念名称的自身信息, 对于两个概念名称完全不能匹配的情况并不一定能够保证它们之间不存在映射关系, 因此这种方法只能筛选出部分映射对. 基于名称的计算方法如公式(1)所示.

2) 基于属性和上下文结构的概念相似度计算方法则主要考虑了本体自身的特征信息, 扩展了计算数据的来源; 由于概念的实例拥有丰富的描述信息, 因此基于实例的概念相似度计算在一定程度上可以较准确地发现两个概念是否存在映射关系, 特别是对于实例丰富的概念可以达到很好的映射效果. 所以为了进一步筛选映射概念对, 得到更准确的映射结果, 本文将这三种方法计算所得结果加权平均, 具体如公式如下:

$$\begin{aligned} Sim(c_1,c_2) &= w_1 \times Sim_{property}(c_1,c_2) \\ &+ w_2 \times Sim_{structure}(c_1,c_2) \\ &+ w_3 \times Sim_{instance}(c_1,c_2) \end{aligned} \tag{13}$$

其中 $w_1 + w_2 + w_3 = 1$, 根据事先设定的阈值 th2, 若有 $Sim(c_1,c_2) > th2$, 则概念 c_1, c_2 存在映射关系, 将其输出.

2 实验与分析

2.1 实验数据

本文采用国际本体映射组织 OAEI 2007 提供的标准测试集 benchmarks 作为测试数据集. 此数据集共包含 51 个本体, 共分为三组: #101~104、#201~266、#301~304. 第一个本体#101 为参考本体(Reference Ontology), 其他本体大部分为#101 参考本体的某种特征缺失或变换. 其中#103、#104 是这两个本体是#101 的 OWL-Lite 版本, #102 是与#101 完全无关的本体. #201~266 这组本体是#101 参考本体的某个或某些特征, 如名称、注释信息、结构、实例、约束等替换或删除后得到的供测试用的变体, 目的是测试在缺失某些本体特征信息的情况下系统的健壮性. #301~304 这组本体是与参考本体相似度较大的现实本体, 可以体现出 benchmarks 数据集的实用性, 从一定程度上检测本文提出的算法是否适用于现实本体.

本实验采用 17 个具有代表性的不同类型的本体来测试本文提出的本体映射算法的有效性. 这 17 个本体的特点及统计数据如表 1 所示:

表 1 测试集统计数据

OntID	数据集描述	概念	属性	实例	映射
101	Reference alignment	36	72	56	36
103	Language generalization	36	72	56	36
104	Language restriction	36	72	56	36
201	No name	36	72	56	36
203	No comment	36	72	56	36
204	Naming conventions	36	72	56	36
205	Synonyms	36	71	56	36
206	Translation	36	71	56	36
208	Upper/lower case letters	36	71	56	36
222	Flatenned hierachy	32	73	56	32
223	Expanded hierachy	69	72	56	36
230	Flattened classes	28	60	47	28
232	No instance&specialistion	36	72	1	36
301	Real: BibTeX/MIT	15	40	0	15
302	Real: BibTeX/UMBC	13	30	0	13
303	Real:Karlsruhe	56	72	0	36

2.2 评估方法

本文采用 OAEI 提供的标准测试集 Benchmark 进行映射结果测试, 对映射结果用查准率 Precision、查全率 Recall 来度量[9]. 查准率和查全率是信息检索领域的重要评价标准, 查全率是衡量映射系统检索出相关信息的能力, 查准率是衡量映射系统拒绝非相关信息的能力.

2.3 实验结果分析

下表给出了采用本文提出算法的实验结果, 其中, OntID 用于表示 OAEI 标准测试数据集 benchmarks 中的本体的编号, rec.表示查全率, pre.表示查准率. 同时为了验证本文提出算法 NPSI 的有效性, 将其与 2007 年参加 OAEI 国际本体映射测试比赛的部分算法在查准率和查全率方面进行了比较. SEMA[10]是综合使用 WordNet 语义词典、基于属性、实例等方法得到映射关系的算法; Falcon[11]是综合了基于语言学和基于图结构等方法进行映射的算法. 表 2 给出了这三种算法查准率和查全率的映射结果.

表 2 NPSI 与其他本体映射算法的数据比较

OntID	SEMA		Faclon		NPSI	
	rec.	pre.	rec.	pre.	rec.	pre.
101	1.00	1.00	1.00	1.00	1.00	1.00
103	1.00	1.00	1.00	1.00	1.00	1.00
104	1.00	1.00	1.00	1.00	1.00	1.00
201	0.98	0.92	0.95	1.00	0.97	1.00
203	1.00	1.00	1.00	1.00	1.00	1.00
204	0.96	0.95	0.98	0.98	1.00	1.00
205	0.96	0.93	0.98	1.00	1.00	0.95
206	0.97	0.94	0.93	1.00	0.97	1.00
208	0.80	0.89	1.00	1.00	0.89	1.00
222	1.00	0.96	1.00	1.00	1.00	1.00
223	0.98	0.97	1.00	1.00	1.00	1.00
230	1.00	0.75	1.00	0.94	1.00	1.00
232	1.00	1.00	0.99	1.00	1.00	1.00
301	0.75	0.70	0.82	0.91	0.93	1.00
302	0.60	0.62	0.58	0.90	0.77	0.91
303	0.80	0.55	0.76	0.77	0.76	0.79
304	0.93	0.77	0.93	0.96	0.94	0.97

从实验结果可以看出, 本文提出的基于概念相似度计算的本体映射方法 NPSI 能够比较准确地发现测试数据中的映射关系. 对于大部分数据集NPSI都能较好地完成本体映射的任务. 针对 1xx 和 2xx 两组数据的测试表现, 本文提出的算法在保证查全率的同时提

高了映射结果的查准率; 对于现实中的本体 3xx, 本文提出的算法对映射结果的查准率和查全率都有明显的提高.

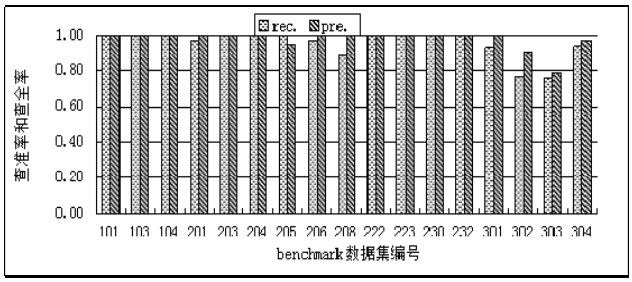

图 2　改进算法的查准率和查全率

图 2 给出了柱形图的方式直观地表示映射结果的查准率和查全率. 图 3 是本文提出的算法 NPSI 与 SEMA 和 Falcon 在查全率上的比较. 图 4 是本文提出的算法 NPSI 与 SEMA 和 Falcon 在查准率上的比较.

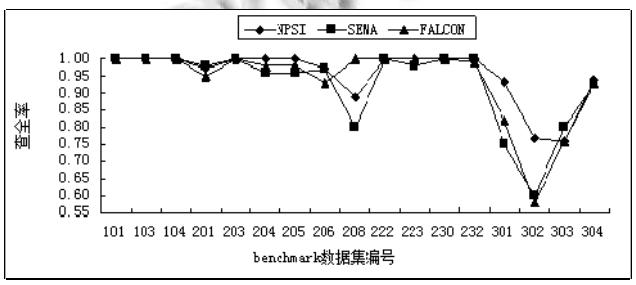

图 3　NPSI 与其他映射方法查全率的比较

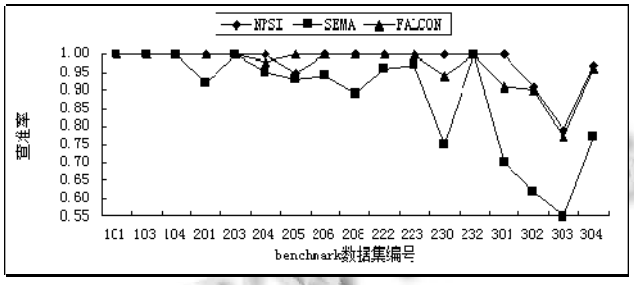

图 4　NPSI 与其他映射方法查准率的比较

3　结语

本文提出了一种改进的多策略概念相似度计算方法, 基于概念名称的相似度计算在原有方法的基础上引入了基于注释的相似度计算, 采用动态加权的方法对参与映射的概念对进行初步约减; 之后对于不满足阈值条件的概念对分别计算其基于属性、结构及实例的概念相似度, 并选取合适的权值进行相似度的合并, 这样减少了综合方法的计算量, 提高了映射效率. 最后采用 OAEI 提供的标准数据测试集 benchmark 进行实验, 同时为验证本文提出算法的有效性, 将本文算法的映射结果与参加 OAEI 2007 竞赛的部分映射系统对比分析. 实验结果表明, 本文提出的算法有效的提高了映射结果的查全率和查准率.

参考文献

1 Berners-Lee T, Hendler J, Lassila O. The semantic web. Scientific American, 2001, 284(5): 28–37.

2 王兵,邢永康.本体概念的语义相似度研究.世界科技研究与发展,2013,35(1):34–37.

3 张忠平,田淑霞,刘洪强.一种综合的本体相似度计算方法. 计算机科学,2009,35(12):142–145.

4 Li W, Zhao Y, Shen N. Concept similarity calculation in ontology mapping. Sixth International Conference on Fuzzy Systems and Knowledge Discovery. IEEE. 2009, 2. 214–218.

5 Lin D. An information-theoretic definition of similarity. ICML. 1998, 98. 296–304.

6 左秀然.基于概念相似度的本体映射系统研究[学位论文]. 武汉:武汉理工大学,2008.

7 李荣,杨冬,刘磊.基于本体的概念相似度计算方法研究.计算机研究与发展,2011,3(11):312–317.

8 姚晓明.高效的基于多策略本体映射方法的研究[学位论文].杭州:浙江大学,2013.

9 Shunmughavel V, Jaganathan P. Semantic enrichment in ontology mapping using concept similarity computing. Fourth International Conference on Advanced Computing (ICoAC). IEEE. 2012. 1–8.

10 Spiliopoulos V, Valarakos AG, Vouros GA, et al. SEMA: Results for the ontology alignment contest. OAEI 2007. OM. 2007.

11 简宁胜.一个本体匹配工具的设计与实现[学位论文].南京: 东南大学,2006.