基于本体映射的多本体查询方法

徐德智,谭 毅
(中南大学 信息科学与工程学院,湖南 长沙 410083)

摘 要:随着语义网的发展,本体已经成为很多领域表达知识的主要手段。许多领域都根据自己的需求建立了本体来描述本领域内的知识。但是目前许多针对本体的语义查询只能对一个本体进行查询。为了实现一个查询能够对多个本体进行访问并且返回适当的查询结果,文中提出了一种利用本体映射实现对多本体的查询方法。其中的映射方法是一种基于语义的多策略结合方式。通过实验发现查询的速度与本体的数量基本呈线性关系且不会因为本体异构程度而增加。

关键词:多本体查询;映射;多策略结合
中图分类号:TP393 **文献标识码**:A **文章编号**:1673-629X(2008)11-0013-05

Multi-Ontology Query Based on Ontology Matching

XU De-zhi, TAN Yi
(College of Information Science and Engineering, Central South University, Changsha 410083, China)

Abstract: With the development of semantic web, ontology has become the main method of expressing knowledge in many areas. According to their own demand, many areas have established ontologies to describe knowledge of their own area. However, current semantic query of ontology can only query from a single ontology and return appropriate results. To realize a query system that can access many ontologies and return appropriate results, proposed a multi-ontology query method based on ontology matching. The ontology matching algorithm is multi-strategy one, combined based on semantics. The result of experiments shows that the time of excecuting query is in line with the number of ontologies, and does not increase with the heterogeneous of ontologies.

Key words: multi-ontology query; mapping; multi-strategy combination

0 引 言

随着语义网的迅速发展,许多领域都开始用本体来对本领域内的知识进行描述。不同领域内的本体结构上都是不同的,而且同一个概念在不同的领域中会被描述成不同的形式,许多相似的概念会出现在不同的领域内。某领域内的用户往往需要其它领域内的知识,这样如何处理好多本体的查询便成为了语义网研究中一个亟待解决的问题。目前,已经提出了一些多本体的查询方法,其中本体集成的方法是目前的主要方法,在结果的查准率方面有突出的表现,但是它对每个本体都需要人工给出一个它和其它本体之间的映射关系。文中利用对原查询的重写,提出了一种基于本体映射的多本体查询方法,并对查询的结果做了较为合理的融合。通过实验发现查询的时间与本体的数量大致呈现正比关系,而且本体的异构性对查询的时间影响不大。

收稿日期:2008-03-05
基金项目:国家自然科学基金重点项目(60433020);湖南省自然科学基金(06JJ50142)
作者简介:徐德智(1963-),男,博士,教授,研究领域为Web计算。

1 相关工作

随着多本体查询的发展,目前出现了两种主要的多本体查询方法,一种是多本体集成的方式,它的主要思想是将多个结构和内容相似的本体集成为一个本体然后在集成后的本体上进行推理查询。在文献[1]中就提出了一种基于集成的方法,但是它并不能自动地实现本体的集成。因为集成本体推理对本体之间的相关信息要求比较高,所以在集成的过程中需要人工的加入本体之间的映射关系。另外一种是分布式查询的方式,它的主要思想是将查询映射成多个查询来对资源进行查询。在文献[2]提出了一种基于SAIL中间件的分布式RDF查询方法。文中针对本体的查询,利用本体映射将原查询映射成多个语义相近的查询,从而实现分布式方式来对多本体进行查询。

2 多本体查询方法的基本思想

为了能够处理相关领域中用本体描述的知识,文

中提出的多本体查询分成三个阶段。例如，一个查询处理器处理一个针对本体 O_a 的查询 Q_a，其中 Q_a 的词汇都是本体 O_a 中的词汇。同时语义网上还有大量的其它本体存在 $\varphi = \{O_1, O_2, \cdots, O_n\}$，这些本体都是查询 Q_a 可以访问的。多本体查询如图 1 所示。这样多本体查询处理器可以按照下面步骤来处理查询问答：

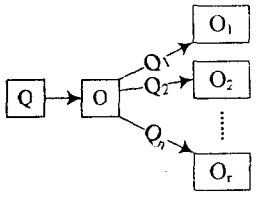

图 1 多本体查询示意图

(1) 资源选择：因为查询推理需要大量的时间，而且不是所有的本体中都有用户需要的数据，所以不能直接对所有的资源进行查询，首先多本体查询处理器必须选择一个本体子集 $\varphi' \subseteq \varphi$。

(2) 查询重写：因为不同领域的本体中表达同义或者相近的概念的时候所用的词汇很有可能是不一样的，所以多本体查询处理器必须重新根据 φ' 中的每一个本体 O_i 的词汇对查询 Q_a 进行重写。

(3) 结果融合：多本体查询处理器必须将每一个子查询的返回结果根据用户的需求进行融合。

以上的三个步骤实际上就是要处理查什么、如何查、结果是什么。其中第一步将利用一种基于多策略的映射方法来实现对本体中匹配实体对的选择，进而实现对本体资源的选择。用户在对本体进行查询的时候根据自己的需要给出一个对查询结果的可信度的期望值 α，其中 $0 \leqslant \alpha \leqslant 1$。本体映射将根据 α 来对本体集 φ 中的本体进行选择。然后根据映射找到匹配的词汇来对查询进行重写。重写后的查询都会有自己相应的可信程度。然后多本体查询处理器将每一个查询分别交给每一个本体查询推理器进行推理查询。最后获得的查询结果将根据其可信程度和用户所关心的程度进行融合，可信程度越高的结果将优先返回给用户。

2.1 资源选择和查询重写

可以利用概率数据逻辑(Probabilistic Datalog)[3]来表示实体之间的映射。例如一个映射可以描述成 $\alpha_{i,j}: T_j(X) \leftarrow T_i(X)$，其中 T_i 是原本体中的词汇，T_j 是目标本体中的词汇，$\alpha_{i,j}$ 是这两个词汇的相似度。同样可以用概率数据逻辑来表示一个查询 Q，可以表示成 $\alpha: Q(X) \leftarrow T_1(X), T_2(X), \cdots, T_n(X)$，其中 $1 \leqslant n$，α 表示查询的可信程度。例如一个查询可以表述为 1.00:Query(x,y) ← Creator(x, "Thinking in Java"), Live(x,y), Nationality (y, "USA")。如果有两个映射 0.9:HaveHouseIn(x,y) ← Live(x,y), 0.8:Auther(x,y) ← Creator(x,y)，那么上面的查询可以映射成另外一个查询 0.72:Query(x,y) ← Author(x, "Thinking in Java"), HaveHouseIn(x,y), Nationality(y, "USA")。其中 $0.72 = 1 \times 0.8 \times 0.9 \times 1$。由此可以得到如下定义：

定义 1：查询映射 $\alpha_b: Q_b(X) \leftarrow \alpha_a: Q_a(X)$。

$\alpha_a: Q_a(X) \leftarrow T_{a1}(X), T_{a2}(X), \cdots, T_{an}(X)$ 是原查询，$\alpha_b: Q_b(X) \leftarrow T_{b1}(X), T_{b2}(X), \cdots, T_{bm}(X)$ 是目标查询，查询 Q_b 的可信程度 $\alpha_b = \alpha_a \times \alpha_{a1,b1} \times \alpha_{a2,b2} \times \cdots \times \alpha_{an,bm}$，其中 $n > 0$，$\alpha_{ai,bi}$ 是原查询中第 i 个词汇到目标查询中第 i 个词汇的相似度。

用户提出查询的时候会被要求给出一个对结果期望的可信度 α，一般情况下 α 是接近 1 的。多本体查询器首先提取查询中的所有词汇，再利用基于语义的多策略方法[4]来找出计算出原本体中这些词汇所表示的实体与其它本体中的哪个或哪些实体是匹配，然后计算到它们的映射关系和相似度。利用这些映射关系可以计算出其查询映射的可信度，若有大于用户期望可信度的则这个本体被选取。重写此查询，利用现有的本体查询推理器推理查询出结果。由此可以得到资源选择和查询重写的算法，其中本体对实体相似度的计算方法将在 2.3 中给出说明。

算法 1：资源选择。

输入：原查询 Q，期望可信度 α，Q 可以访问本体资源集合 $\varphi = \{O_1, O_2, \cdots, O_n\}$。

输出：选择后的本体资源集合 $\varphi' = \{O_1, O_2, \cdots, O_m\}$

Step1：提取查询 Q 中的所有词汇构成词汇集合 $\tau = \{T_1, T_2, \cdots, T_n\}$。

Step2：从 φ 中取出一个本体 O'，对 τ 中的每一个词汇 T_i 所对应的原本体 O 中的实体，计算与其相似度大于 α 的目标本体中的映射的集合 μ_i。若有某个 μ_i 为空则舍弃 O'，继续 Step2；否则进入 Step3。

Step3：从每个映射集合中取出相似度最大的映射，计算这些映射所对应的相似度的积 α'，若 $\alpha' \times \alpha_Q \geqslant \alpha$ 则将 O' 加入 φ'，否则舍弃 O'。其中 α_Q 是原查询的可信度。

Step4：循环 Step2，3 直到 φ 中所有的本体都被判断。

利用资源选择中计算的本体映射的结果可以将原查询映射成可以对目标本体进行查询的目标查询，使其中的词汇都是目标本体中出现的实体。

算法 2：查询重写。

输入：原查询 Q，期望的可行度 α，对应 φ' 中某个本体 O' 的所有被计算的映射集合 μ_i。

输出：对应 O' 的目标查询集合 $\theta = \{Q_1', Q_2',$

…，$Q_n'\}$。

Step1：从每个映射集合 μ_i 中取出一个映射，计算相似度的积 $\alpha' \times \alpha_Q \geqslant \alpha$。

Step2：若 $\alpha' \geqslant \alpha$，则将原查询中的词汇替换成对应的 O' 中对应的实体，将 α' 作为重写后的查询的相似度。

Step3：循环 Step1 直到计算完所有的实体组合。

2.2　结果融合

重写后的查询 Q' 分别被提交给各自对应的本体 O' 进行推理查询，得到的返回结果不能随便返回给用户，因为用户会有自己比较关心的结果。各个领域的本体用户对自己领域内事物的关心程度都不相同，所以他们对自己领域内本体中实例的访问次数也有较大区别。但是因为查询重写是基于本体映射的结果完成的，所以各个针对不同领域的重写后的查询所返回的结果中很有可能有许多相同的结果。如何将这些结果根据用户的需求返回，并且如何处理好相同的结果便成为了结果融合的主要任务。

各个领域内的用户对各自领域内事物的访问次数在很大程度上体现了他们对这些事物关心的程度，所以可以根据返回结果中每个事物被访问的次数得到一个权重。这个权重就表示了此事物在此领域内被用户所关心的程度。

定义：访问权重

$$w^r(a) = \frac{s(a) - \min_{j \in r} s(b)}{\max_{j \in r} s(b) - \min_{j \in r} s(b)}$$

$s(i)$ 代表实例 b 被访问的次数，r 表示查询所返回的结果实例集合。若 $|r| = 1$ 则设定 $w^r(b) = 1$。

由于每个查询都会返回相应的结果，而每个查询所针对的本体又是为不同的领域所设计的，当多个查询返回的结果中有相同的实例时，就需要根据它在不同领域中受用户关心的程度做综合考虑。同时由于每个查询和原查询的相似程度不同，所以返回结果的融合也需要考虑查询相似度的作用。

定义 1：融合评分

$$\hat{s}(a) = h \cdot \frac{\sum_{r \in R} \alpha_r \cdot w^r(a)}{|R|}$$

h 代表本体集 φ 中有 h 个本体的查询返回了实例 a，α_r 表示结果 r 所对应查询的可信度，$R = \{r\}$ 表示每个查询结果集的集合。

对上述融合方法的说明，如果一个实例 a 在对许多领域的查询结果中都出现了则表示它很有可能是各个领域所共同关注的，所以当 h 越大的时候实例应该更优先的返回给用户。$\alpha_r \cdot w^r(a)$ 既考虑了实体 a 在返回结果集 r 中受用户的关注程度，同时也兼顾了结果集 r 与原本体结果集的相似程度 α_r，α_r 越高说明结果 r 越可信。所以最后的结果将根据 $\hat{s}(a)$ 对所有的实例进行排序，$\hat{s}(a)$ 越大的优先返回给用户。

在文献[4]中对各种结果融合策略做了详细的对比。发现其中的 $\sum .s.1$ 融合方法在各个系统中的表现优于其它方法。上述的融合方法是对 $\sum .s.1$ 融合方法的应用和改进。

2.3　实体对相似度的计算

基本思想：利用基于语义的多策略结合方法来发现本体之间匹配的实体对。再利用加权平均各种相似度(实例，注释，名称，结构，属性)的方法计算实体对之间的相似度，并参考多策略中的相关性依据对结果进行修正。

在之前的工作中已经提出了一种基于语义的多策略结合方式，它的主要工作是找到两个本体中语义上匹配的实体对。但是基于语义的多策略结合方式只会对实例和注释的相似度进行计算，所以在时间上的开销要比直接通过计算相似度来确定匹配实体要小。而且经过 OAEI 提供的测试数据所作的对比发现，多策略的方法在不牺牲查准率的前提下对查全率有很大的提高。

基于语义的多策略结合方式映射算法步骤：

1)使用基于 URI 的策略直接获得匹配关系；

2)使用基于实例和注释的策略，相似度很高则确定匹配关系，较高则提供相关性依据；

3)使用基于结构和属性的策略，提供相关性依据；

4)使用基于名称的策略，再参考前面的相关性依据对结果进行修正。

具体的映射算法：

算法 3　基于语义的多策略映射算法。

输入：经过待映射本体 O_1'、O_2'；

输出：确定了匹配关系的实体对 (e_{i1}, e_{i2})；

Step1：URI 相同则两实体匹配，不同则不作处理。

Step2：计算实例相似度 S_{Ins}，$S_{Ins} \geqslant U_{Ins}$ 则两实体匹配；$U_{Ins} > S_{Ins} > L_{Ins}$ 则两实体相关；$S_{Ins} \leqslant L_{Ins}$ 则不作处理(U_{Ins} 和 L_{Ins} 是人为选定的上、下位阈值)。

Step3：计算注释相似度 S_{com}，$S_{com} \geqslant U_{com}$ 则两实体匹配；$U_{com} > S_{com} > L_{com}$ 则两实体相关；$S_{com} \leqslant L_{com}$ 时，则确定两实体不匹配，否则不作处理(U_{Ins} 和 L_{Ins} 是认为选定的上、下位阈值)。

Step4：对于实体的名称集，若名称集中有至少两个元素相同，则两实体匹配；若只有一个元素相同，则两实体同名；将每个实体名称在 Wordnet 中的同义词加入名称集，若有相同元素(无论几个)，则两实体同名。

Step5:若概念的父概念集、子概念或兄弟概念中有相同元素,则两概念相关。特别的,若概念的父概念相同且子概念有相同部分,则两概念匹配。

Step6:若属性的父属性或子属性中有相同元素,则两属性相关。

Step7:若两个属性有相同定义域或值域,则它们相关。

Step8:若两个同名实体是相关的(利用 Step2,3,5,6 的结果),则它们匹配。

在确定了所有的匹配实体对后再对它们进行上述五种相似度的计算,并进行加权平均得到最后的相似度。其中关于概念相似度的计算公式可以参考之前发表的文章[5]。对于属性不对其进行相似度的计算,若匹配则确定为同义的。

因为所有需要计算相似度的实体对都是通过基于语义的多策略方式发现的,说明实体对之间的语义相关程度还是比较高的,实体中的实例很有可能是用户所需要的,所以当计算出来的相似度 $S < \alpha$ 时不能轻易舍弃。此时可以将其相似度提升至 α。

3 系统体系结构及实现

在上述多本体查询思想的基础上,设计出一个多本体查询系统,实现对语义数据库中多本体的查询(见图 2)。由于文中的多本体查询是应用于 SNAX 系统中的,所以将其命名为 SNAX_MOQ(Multi_Ontology Query)。整个框架建立在 Java 1.5 的运行平台上。

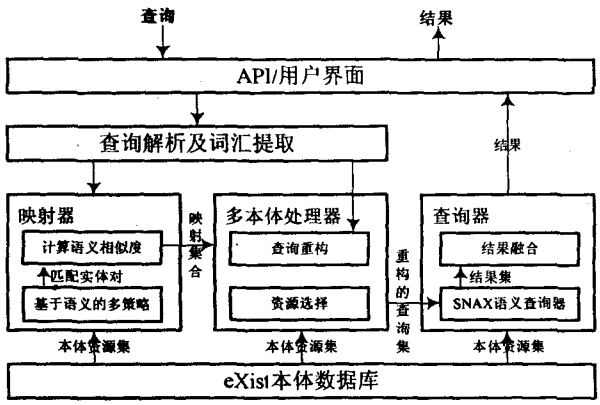

图 2 实验结构框图

eXist 原生数据库是一个开源的 XML 数据库,在 SNAX 系统中对它进行了适当的修改使它能够存储 OWL 语言所描述的本体。将本体文件全部存储在 eXist 本体数据库中。

映射器的主要任务是根据查询语句中的实体发现原本体和目标之间的匹配实体对,并计算出它们之间的相似度。其中基于语义的多策略已经在之前的工作中实现。

多本体处理器是本系统的关键。它负责本体资源的选取,并将原查询重写成适合每个本体的查询。使用 Jena2 开发包来对本体进行解析。

查询器 SNAX 语义查询器是已经实现的针对本体的查询器,它主要是利用 Jena2 中的推理器。结果融合将根据 2.2 中提出的融合方法将融合后的结果返回给用户。

API/用户界面的主要任务是提供一个可视化的界面给用户。用户通过 SPARQL 语言提出查询。

4 实验结果评估

在本节中分别做了两个实验对 SNAX_MOQ 系统进行测试。第一个实验利用 LUBM(Lehigh University Benchmark)[6]评价标准对系统在本体数量方面的表现进行测试。在 LUBM 给出的本体的基础上分别利用不同的实例数据集生成了 6 个同构的本体。在 SNAX_MOQ 系统中对下面三个 SPARQL 查询进行测试。本体文件在 http://www.lehigh.edu/zhp2/2004/0401/univ-bench.owl 可以下载。

```
SELECT? x
  WHERE {? x rdf:type ub:GraduateStudent }
SELECT? x? y
  WHERE {? x rdf:type ub:AssistantProfessor.
         ? y rdf:type ub:Publication
         ? y ub:publicationAuthor? x}
SELECT? x? y? z
  WHERE {? x rdf:type ub:GraduateStudent.
         ? y rdf:type ub:University.
         ? z rdf:type ub:Department.
         ? x ub:memberOf? z.
         ? z ub:subOrganizationOf? y.
? x ub:undergraduateDegreeFrom y }
```

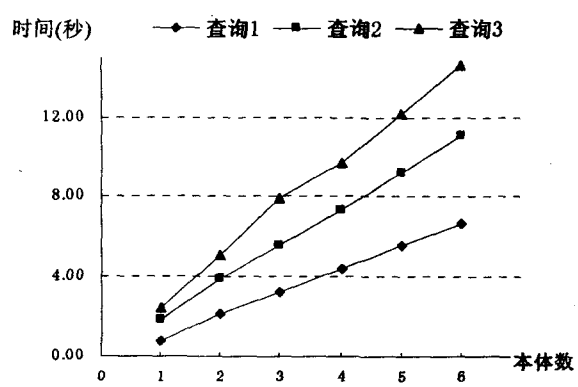

图 3 多本体测试结果图

通过图 3 可以发现,基于本体映射的多本体查询的时间与本体的数量形成线性关系,说明查询的时间

不会因为本体数量的增加而出现急剧恶化的现象。其主要原因是资源选择后很多不符合要求的本体被排除,减少了大量的查询推理时间。

实验二利用 LUBM 提供的本体和 SWRC (Semantic Web for Research Communities)提供的关于大学的本体对上面的 3 个查询做了本体异构方面的测试。其中 LUBM－SWRC 表示 LUBM 是原本体,SWRC 是目标本体,其它依此类推。本体文件可以在 http://swrc.ontoware.org/ ontology 下载。

通过图 4 可以发现,基于本体映射的多本体查询与本体的异构程度基本无关。说明多本体查询的时间与本体异构的程度的关系不大。其主要原因是因为利用基于语义的多策略中对结构的策略做了适当的修改,而且不需要计算出结构上的相似度就能给出实体对的相关性。结合其它策略就能判断实体对是否匹配。而我们只针对这些匹配对进行语义相似度的计算从而减少了异构性的影响。

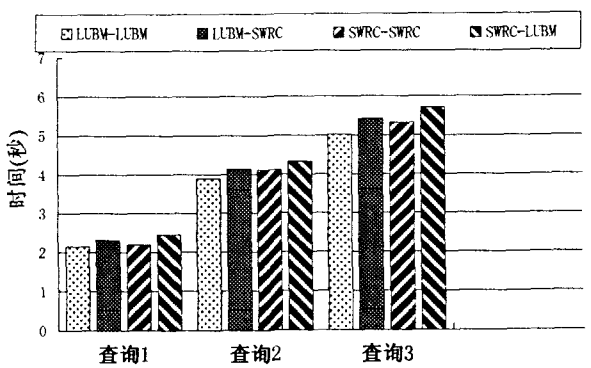

图 4　异构本体测试结果图

因为对单个本体的查询是利用的现有的推理查询器,所以在此没有对推理查询的准确性做测试。

5　结束语

多本体查询主要有两种方式,一种是本体的集成,一种是分布式查询。文中主要是一种分布式的查询方式。利用本体映射来自动地对本体资源进行选择,并且根据映射的结果重写原本体的查询语句,对被选中的本体资源进行查询,从而实现了对多本体的查询。根据映射中的相似度和用户对实例的访问次数对多个查询结果进行适当的融合。通过实验测试发现查询的时间与本体的数量呈线性关系,并且本体的异构程度对查询的时间影响不大。

未来的工作主要是:结合本体集成的方法,适当融合相关程度高的本体,进一步减少本体的数量,从而减少查询的时间。

参考文献:

[1] Haase P, Motik B. A Mapping System for the Integration of OWL－DL Ontologies[C]//Proceedings of the first international ACM workshop on Interoperability of Heterogeneous Information Systems (IHIS'05). Bremen, Germany: [s. n.], 2005:9－16.

[2] Stuckenschmidt H. Towards distributed processing of RDF path queries[J]. Journal of the ACM, 2005, 32(3):112－124.

[3] Fuhr N. Probabilistic Datalog: Implementing Logical Information Retrieval for Advanced Applications[J]. Journal of the American Society for Information Science, 2000, 51(2):95－110.

[4] Renda M E, Straccia U. Web Metasearch: Rank vs. Score Based Rank Aggregation Methods[C]//In 18th Annual ACM Symposium on Applied Computing (SAC'03). Melbourne, Florida, USA: [s. n.], 2003:841－846.

[5] 徐德智,肖文芳,王怀民.本体映射过程中的概念相似度计算[J].计算机工程与应用,2006(9):167－169.

[6] Guo Y, Pan Z, Heflin J. Lubm: A benchmark for owl knowledge base systems[J]. Journal of Web Semantics, 2005, 3(2): 158－182.

(上接第 12 页)

之间的性能差异。而且通过何种指标进行算法的评价,也没有统一的标准。下一阶段工作将重点着手建立基于视频的烟雾特征数据库,以及针对测试库的算法评价标准,以便为在一个开放的、公平的平台上进行相关学术研究打下基础。

参考文献:

[1] 程晓舫,王瑞芳,张维农,等.火灾探测的原理与方法(上)[J].中国安全科学学报,1999,9(1):24－29.

[2] Collins R T, Lipton A J, Kanade T. A System for Video Surveillance and Monitoring[C]//Proc. of American Nuclear Society 8th Int. Topical Meeting on Robotics and Remote Systems. Pittsburgh: [s. n.], 1999.

[3] 陈　莹.大空间图像型火灾探测和自动灭火技术的研究[D].天津:天津大学,2006.

[4] 冈萨雷斯. 数字图像处理[M].北京:电子工业出版社,2003.

[5] Toreyin B U, Dedeoglu Y, ç etin A E. Contour Based Smoke Detection in Video Using Wavelets[C]//14th European Signal Processing Conference EUSIPCO 2006. Florance, Italy: [s. n.], 2006.