计算机科学2004Vol.31№.9A

应用本体解决面向语义的信息集成中的查询处理

赵 宁 李庆忠

(山东大学计算机科学与技术学院 济南250061)

摘 要 面向语义的信息集成中基于Web的查询处理应用本体来解决在多个领域中检索数据可能产生的语义层的异构,对于在统一的用户界面下提交的查询请求,根据语义,从Web上搜索相关领域的信息,最终将结果显示在用户界面上。该过程是通过查询分解、子查询转换、分发子查询和子查询结果收集及语义转换来完成的。同时,考虑到基于Web的信息查询一般要涉及多个分布的数据源,查询的响应时间要依赖于网络的传输速度,一般来讲比较长,还借鉴了缓存系统的思想,在查询处理架构中引入了缓存数据库,有选择地存放最常用的查询信息,以提高整个查询的速度。

关键词 本体,语义,查询,缓存库

Apply Ontology to Solve Query Processing in Semantics-Oriented Information Integration

ZHAO Ning LI Qing-Zhong

(College of Computer Science and Technology, Shandong University, Jinan 250061)

Abstract In order to solve the semantic heterogeneities resulted from the Web-based query processing in the information integration, this paper uses ontology to build a general architecture. The architecture can search related domain information from the Web for a query request submitted by the user through a universal user interface, then show the results on the interface. This processing consists of query decomposition, subquery transformation, subquery delivery and subquery collection and merge and semantic transformation. In addition, information retrieval on the Web generally involves multiple distributed data sources, the response time relys heavily on the internet. So the paper adopts the idea of caching system, adding the caching database to the architecture, which can be used to selectively store the most useful query information, to improve the whole query efficiency.

Keywords Ontology, Semantic, Query, Caching system

1 引言

随着计算机的应用越来越广泛,TCP/IP、HTTP的普及,各部门、各地区之间,都要求能够实现网络互联互通,从而达到信息的交流。但随着发展,信息孤岛问题变得越来越突出。各个部门、各个地区之间,相同的数据由于不同的用户以不同的方式建模,导致不同程度的异构,包括语义层的异构,使得相互联系的部门之间不能交换信息。如何在现有数据源的基础上进行包装,使得具有相同语义的数据有统一的表示形式,从而解决各个数据源之间的语义冲突,达到信息的互连互通,已经成为当前亟待解决的问题。

为解决这个问题,有人提出了一个基于本体知识来检索领域信息的搜索引擎,采用了mediator技术、过滤技术,通过对用户行为建模来实现对某领域内信息的搜索,还有人提出了在层次领域中应用本体进行信息检索,但解决的是某个应用领域内语义层的异构。还有人也提出了利用本体进行面向语义的Web信息集成的思想,但没有给出具体的实现过程,也没有考虑查询的响应时间。由于本体是通过概念间的关系来描述概念的语义,而且本体本身具有很好的概念层次结构和支持逻辑推理,因此本文引入本体建立一个面向语义的Web信息检索框架,对于用户输入的查询进行面向语义的处理,通过Web检索多个相关领域的信息。本文应用本体建立了基于Web的面向语义的查询处理框架OBQA(Ontology-Based Query Architecture)。OBQA提供了一个用户查询平台,来快速有效地处理对于多个异构数据源查询的用户的查询请求,并解决了其中可能产生的语义层异构的问题。在下面的部分中,首先介绍OBQA的总体架构,其次提出本体的形式化定义,并讨论在OBQA对来自于用户的查询请求的处理过程,再次讨论对缓存数据库中存放信息的处理,然后针对某个应用实例讨论OBQA工作的整个过程。最后对工作进行总结并提出了对以后工作的展

赵 宁 硕士,主要从事本体和信息集成的研究。李庆忠 博士,教授,博士生导师,主要从事应用集成与信息集成,数据仓库,数据库技术的研究。

望。

2 面向语义的信息检索的总体架构

在面向语义的信息集成中，首先要对 Web 上的各个领域的信息进行建模，形成一个公共本体，然后本文基于这个公共本体所表达的语义来检索相关领域信息。基于本文所描述的框架下的查询以公共本体中的概念来表达。处理查询的整体架构如图1所示。

OBQA 主要包括本体，以及全局查询生成器，查询分解器，本体管理器，查询转换器，查询分发器，查询结果收集器等组件。这些组件处理用户以公共本体表达的查询，将查询转变成子查询，通过使用公共本体中与本地数据源间的映射将这些子查询以本地数据源中的实体来表达。处理过表示上的不同后，把子查询的结果经过适当的转换合并起来，然后把结果在用户端显示出来。OBQA 能处理对分布的异构的多数据库查询，主要是通过对分布的异构数据库的统一的访问来实现的，即在这些局部数据库系统之上使用一个单独的公共查询语言。下面首先提出本文中本体的定义，然后讨论在本文的架构下对来自于用户查询请求的处理过程。

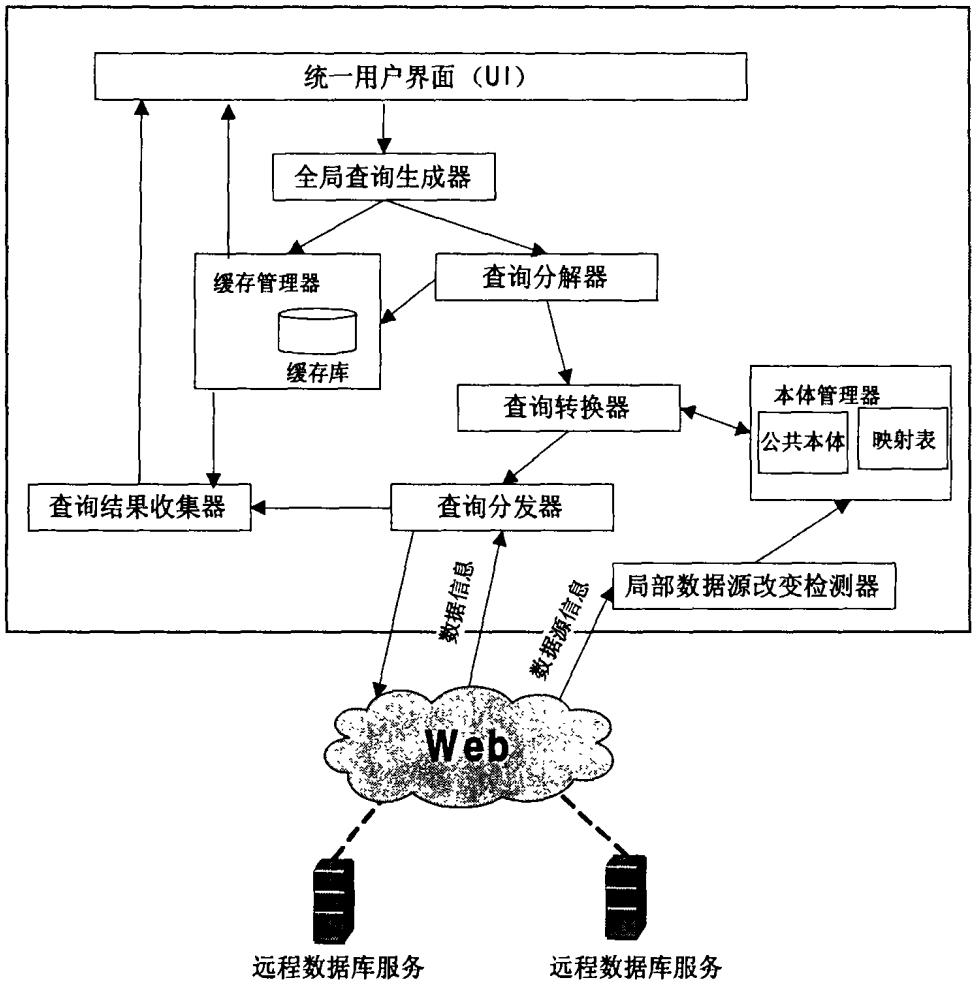

图1 OBQA 的总体框架

2.1 本体

Studer 定义本体为“本体是共享概念模型的明确的形式化规范说明”。概念模型是通过抽象出客观世界中一些现象的相关概念而得到的模型，其表示含义独立于具体的环境状态；明确指所使用的概念及使用这些概念的约束都有明确的定义；形式化指本体是计算机可读的；共享指本体中体现的是共同认可的知识，反映的是相关领域中公认的概念集，它所针对的是团体而不是个体。本文中，本体是一个领域中术语及之间关系的明确的形式化描述，所以必须依靠领域专家的帮助才能建立起相关的领域本体。智能组件可以使用公共本体来标识统一的意义，然后解决一个问题，或推断出一个结果。下面给出本文中本体的形式化的定义。

定义1 本体是一个5元组(C,R,F,A,I)，其中：

·C 是概念或类，指任何事务，如工作描述、功能、行为、策略和推理过程。从语义上讲，它表示的是对象的集合，其定义一般采用框架(frame)结构，包括概念的名称，与其他概念之间的关系的集合，以及用自然语言对概念的描述。

·R 是关系，指在领域中概念之间的交互作用，

形式上定义为 n 维笛卡儿积的子集：$R:C_1\times C_2\times\cdots\times C_n$。在语义上关系对应于对象元组的集合。从语义上讲，关系基本可以分为四种：part-of，表达概念之间部分与整体的关系；subclass-of，表达概念之间的继承关系，类似面向对象中的父类与子类之间的关系，如“教授”就是 subclass-of“教师”；instance-of，表达概念的实例与概念之间的关系，类似面向对象中的对象和类之间的关系；attribute-of，表达某个概念是另一概念的属性。如“姓名”是教师的一个属性。

·F 是函数，一类特殊的关系。该关系的前 n－1 个元素可以唯一决定第 n 个元素。形式化定义为 $F:C_1\times C_2\times\cdots\times C_{n-1}\rightarrow C_n$。如 Teacher-of 就是一个函数，Teacher-of (x,y,z)表示 y 是给 x 上 z 课程的老师。

·A 是公理，代表永真断言，如“教授”属于“教师”的范围。

·I 是实例，代表元素。从语义上讲实例表示的就是对象。

2.2 对于一个来自于用户界面的查询请求的处理过程

OBQA 对于来自用户界面的查询请求按如下步骤进行处理：

1)用户通过统一的用户界面提交一个查询请求，该用户界面将用户请求交给全局查询生成器，将该请求用全局本体中的概念的形式来表达。

2)查询请求先被送往缓存管理器，如果该查询的结果可以从缓存数据库中得到，并且数据可信度超过一个阈值，则缓存管理器就把结果提交给用户界面，整个查询处理过程结束。否则查询请求送至查询分解器。本文中数据可信度是指数据可以被信任的等级。数据可信度随着数据的存放时间逐渐降低，并且降低的程度与数据源有关，存放实时数据如存放与股票交易有关的数据源的数据可信度降得快些，而存放有关某个人家庭住址信息的等非实时数据源的数据可信度降得慢些。数据可信度的阈值也是依赖于特定数据源的。

3)查询分解器根据下面的步骤进行查询分解：

首先分解 from 子句；然后处理投影表；再处理 where 条件。

4)将分解得到的子查询，从缓存库中查找，可以查找到的子查询结果，如果数据可信度超过阈值，则由缓存管理器提交给结果收集器，否则子查询送至查询转换器。

5)查询转换器查询本体管理器中公共本体与局部数据源本体之间的映射表，将子查询中的公共本体的概念或实体转换成对应的局部数据源的概念或实体。这样就把全局子查询转换为局部子查询，并得到映射表中局部子查询对应的远程局部数据库的地址，然后将这些信息送至查询分发器。

6)查询分发器根据得到的局部子查询的地址信息调用相应的远程数据库服务器的 Web 服务通过唤醒该 Web 服务部署的方法去检索信息，将返回的数据信息传送给结果收集器，数据源信息提交给局部数据源改变检测器。局部数据源改变检测器通过查找本体管理器中相应的数据源信息，判断数据源信息是否改变，改变是否影响公共本体，如果需要，局部数据源改变检测器要求领域专家修改公共本体的定义和映射表中相应的映射关系。

7)查询结果收集器对来自于查询分发器的结果进行整理转换，然后把各个子查询结果连同缓存管理器提交的子查询结果进行合并，一方面将结果提交给用户查询界面，一方面提交给缓存管理器，由缓存管理器调用算法3.1决定是否存入缓存。整个查询过程结束。

3 对缓存数据库中存放信息的处理

本文对于减小查询的响应时间所采用的方法基于文[3]所描述的思想，把认为有用的信息按一定的形式放入到框架中的附加信息库中。在学校教学应用系统中，“教师”是一个领域概念，教师的信息可以从教务处和财务处两个数据源中获得，图2为“教师”这个领域概念的本体模型。假设定义了“教授”这样一个信息类——教授的研究方向和教授的工作时是会被经常查询的。这样，这些信息就可以以一定的形式存放到附加的信息库“教授缓存信息库”中。如图3所示。当查询教授研究方向或参加工作时间时，查询处理框架会从附加信息库中直接获取信息而不必去查询 Web 信息源。这样，就可以大大减少查询的响应时间，提高查询效率。OBQA 选择存放到缓存库中的信息时只考虑用户查询的分布。如果用户对某些信息的查询频率超过某个阈值，缓存管理器就创建一个附加信息库，将信息存放到附加信息库中。本文提出了一个算法来分析用户查询，从中分析出有用的、值得存放的信息。

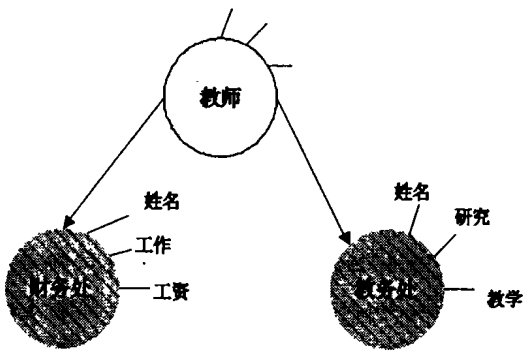

图2 教师概念的本体模型

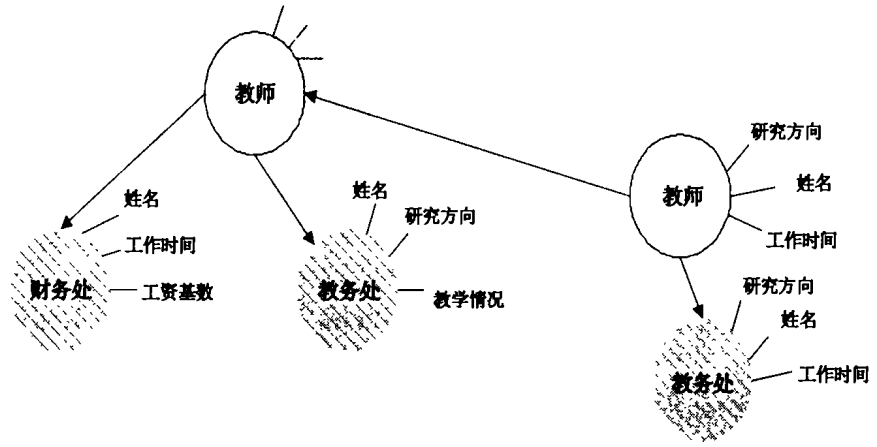

图3 添加附加信息库

3.1 算法 Cluster&Merge()

算法如下：

输入：用户的查询集

输出：要存放到缓存库的信息

(1)对输入的查询集合用 classify()进行分类，决定用户感兴趣的是哪类数据。

(2)对每个数据类用 Cluster()选取用户在该类数据中感兴趣的属性组。

(3)对数据类中的所有(数据源，属性组)，用 Merge()合并可以合并的数据类，以保证数据类更紧凑。

其中 classify 分类算法如下：

输入：查询集

输出：用户感兴趣的数据类的本体

(1)取得当前查询集中的一个查询 Q

(2)根据 Q 的约束条件 p，求 Q 的子查询 Sp

(3)得到 Q 的子查询中用户感兴趣的子查询集合{Spi}

(4)若 Q 是第一个查询，则建立 Sp 的本体，否则更新 Sp 的本体

(5)若{Spi}为空，则更新查询 Q 的数据源中属性 A 的数目，否则对{Spi}中的每个 Spi，更新每个 Spi 中 A 的数目

(6)更新 Sp

其中 cluster 算法如下：

输入：数据类 O

输出：该数据类中用户感兴趣的属性组

算法：对于 O 中的每个数据类中出现具有相似频率的属性组进行分类聚集

其中 Merge 算法如下：

输入：数据类 O 的所有(数据源，属性组)对(S，C)

输出：合并之后的数据类 O′

算法：(S,C)为(s1,A),(s2,A),…(sn,A)，若 s1,s2,…,sn 属于同一个直接超类 S′，并且 s1,s2,…,sn 形成 S′的一个覆盖，并且(s1,A),(s2,A),…(sn,A)对于 A 的查询频率相差在一个阈值范围内，则将 (s1,A),(s2,A),…,(sn,A)合并为一个(数据源，属性组)对(S′,A)。

上面的算法是对用户查询进行分析，决定是否可以存入附加信息库。当用户查询的分布中对于某数据类查询的频率超过查询频率阈值，那么算法就会输出该类。

4 应用实例

本文以数字化校园系统为例，在该系统中，有教务处、财务处两个应用领域，在这两个领域中，都有教师的概念，教师作为领域本体的一部分，在两个领域本体中分别定义如下：

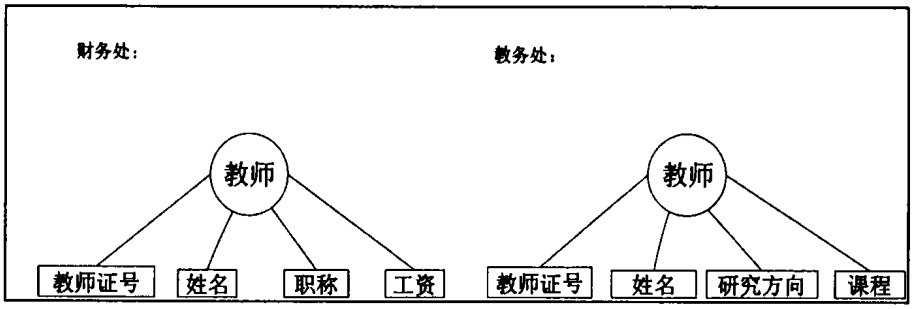

图4 财务处教师的本体模型和教务处教师的本体模型

作为一个应用系统，本文可以集成这两个领域中的教师本体，建立教师的公共本体为：

如果查询为:查询教"面向对象技术"课程的教授的姓名。那么本文可以形成全局查询为:

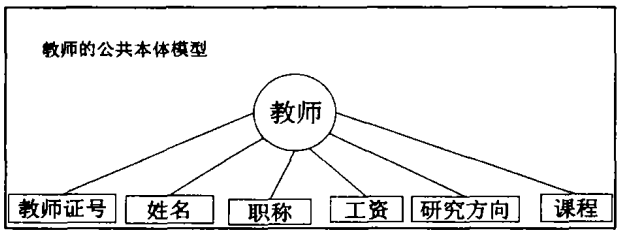

图5 教师的公共本体模型

select 姓名 from 教师 where 课程="面向对象技术"and 职称="教授"。先把该查询发送到缓存管理器,如果缓存中有该查询的结果并且结果符合用户要求,则由缓存管理器直接把结果返回给用户界面,查询过程结束。否则通过查询分解,假设在系统中只有财务处与教务处与教师相关,那么可以分解查询如下:select 姓名 from 教师 where 职称="教授";select 姓名 from 教师 where 课程="面向对象技术",并且这两个子查询查询出的作为主键的教师证号是相同的。通过从公共本体与各领域本体映射表中查找各项的对应项,从而得到两个局部子查询为:select 姓名 from 财务处 where 职称="教授";select 姓名 from 教务处 where 课程="面向对象技术",同时分别得到这两个局部子查询的地址信息:com. finacial. sdu. edu. ,com. dean. sdu. edu,然后查询分发器根据地址去调用相应子查询的数据源在Web服务器上部署的方法,然后得到返回的查询结果。如果结果需要合并转换,再由结果收集器合并结果并根据用户查询进行转换,然后显示给用户。该查询结果还提交给缓存管理器,缓存管理器根据算法3.1决定是否将结果存入缓存库,是否需要合并存入缓存库。到此查询完成。

结束语 本文提出了处理Web查询的一个总体框架,并对缩短查询的响应时间提出了解决办法,但框架中缓存库中存放的信息并不是实时更新的,如何保证缓存库中的信息的正确性,虽然本文提出了可信度限制,但并不能保证信息的实时更新,如何让缓存数据库中存放的信息自动实时更新,并计算维护的成本是以后准备研究的工作。

参 考 文 献

1 Braga R M M, Werner C M L. Using ontologies for Domains Information Retrieval. Computer Science Department Federal University of Rio de Janeiro, 0-7695-0680-1/00 (c)2000 IEEE

2 Cruz I F, Rajendran A. Semantic Data Integration in Hierarchical Domains. University of Illinois at Chicago, 1094-71671031©2003 IEEE

3 Ashish N, Knoblock C A, Shahabi C. Selectively Materializing Data in Mediators by Analyzing User Queries. Integrated Media Systems Center. Information Sciences Institute and Department of Computer Science University of Southern California, 4676 AdmiraltyWay, Marina del Rey, CA 90292

4 邓志鸿,唐世渭,杨冬青. 面向语义集成——本体在Web信息集成中的研究进展. 北京大学计算机科学与技术系. 计算机应用, 2002(1)

5 邓志鸿,唐世渭,张铭,杨冬青,陈捷. Ontology研究综述. 北京大学计算机科学与技术系. 北京大学学报(自然科学版), 2002(5)

(上接第104页)

目在一般情况下要远远少于叶节点的数目,因此可以避免信息单元与查询之间过多的相似度计算量,可以在提交给用户大小适中、语义完全的检索结果信息的同时,加速信息检索的运行效率,提高用户对检索结果的满意程度。

结论 本文对基于文档划分的XML信息检索技术进行了研究,提出了一种用统一的检索技术进行基于关键词查询的XML信息检索方法。由于充分利用了XML文档的结构和语义特点,比较传统的XML信息检索技术,该方法可以在XML文档中自动界定适于检索的信息单元,以减少系统运行的计算开销,提高信息检索速度。在今后的研究中,将对这种方法进行进一步完善和改进,在不影响查询速度的前提下,研究提高其查全率和查准率,进而提高检索的综合性能。

参 考 文 献

1 Berglund A, Boag S, Chamberlin D, et al. XML Path Language (XPath)2.0. W3C Working Draft. World Wide Web Consortium (W3C), May 2003. http://www.w3.org/TR/xpath20/

2 Salton G, Wong A, Yang C S. A vector space model for automatic indexing. Communications of the ACM. New York, USA, 1975

3 Lee J. Analyzig the Effectiveness of Extended Boolean Models in Information Retrieval. In: Proc. of ACM SIGIR'94. Dublin, Ireland, 1994

4 Hatano K, Kinutani H, Yoshikawa M, Uemura S. Information Retrieval System for XML Documents. In: Proc. of the 13th Intl. Conf. on Database and Expert Systems Applications. Aix-En-Province, France, 2002

5 Hayashi Y, Tomita J, et al. Searching text-rich XML documents with relevance ranking. ACM SIGIR 2000 Workshop on XML and Information Retrieval. Athens, Greece, 2000