ISSN 1673-9418 CODEN JKYTA8
Journal of Frontiers of Computer Science and Technology
1673-9418/2007/01(01)-0059-20

E-mail:fcst@nci.ac.cn
http://www.ceaj.org
Tel:+86-10-51616056

基于本体的关系数据库语义检索*

王 珊[1,2],张 俊[1,2,3+],彭朝晖[1,2],战 疆[1,2],杜小勇[1,2]
WANG Shan[1,2],ZHANG Jun[1,2,3+],PENG Zhao-hui[1,2],ZHAN Jiang[1,2],DU Xiao-yong[1,2]

1.中国人民大学 信息学院,北京 100872
2.教育部数据工程与知识工程重点实验室,北京 100872
3.大连海事大学 计算机科学与技术学院,辽宁 大连 116026
1.Information School,Renmin University of China,Beijing 100872,China
2.Key Laboratory of Data Engineering and Knowledge Engineering,Ministry of Education,Beijing 100872,China
3.Computer Science and Technology College,Dalian Maritime University,Dalian,Liaoning 116026,China
+Corresponding author:E-mail:zhangjun11@ruc.edu.cn

WANG Shan,ZHANG Jun,PENG Zhao-hui,et al.Ontology-based semantic search over relational databases.Journal of Frontiers of Computer Science and Technology,2007,1(1):59-78.

Abstract: Taking the economics ontology as an example,the authors study the principles and methods of SemSORD in detail and present a novel SemSORD prototype called Si-SEEKER based on keyword search over relational databases.In the end,the emerging research on SemSORD is proposed.
Key words: relational database;semantic search;keyword search;ontology

摘 要:以经济学领域本体为例,首先研究 SemSORD 基本原理和方法,然后提出基于关系数据库关键词检索(Keyword Search over Relational Databases,KSORD)技术实现的关系数据库语义检索模型,并实现相应的原型系统 Si-SEEKER,最后提出该领域的研究挑战和技术发展趋势。
关键词:关系数据库;语义检索;关键词检索;本体
文献标识码:A **中图分类号**:TP301

1 引言

关系数据库通常是通过结构化查询语言 SQL 来访问的,这种方式是确定性的精确查询,即用户需要构造准确的 SQL 查询语句, 其查询条件是精确比较的,查询结果也是精确的。另外,这种查询方式还要求用户知道并理解关系数据库模式, 懂得如何书写 SQL 查询,一般适合专业用户使用。随着 Internet 和 Web 技术普及和应用, 普通用户已经习惯使用搜索引擎(如 Google,Baidu 等)查找信息,而 Web 上大量的数据实际上也是存放在关系数据库中, 无法被搜索引擎搜索到。因此,近年来,利用关键词查询来访问关系数据库的方式得到广泛关注和研究[1]。但是,

* the National Natural Science Foundation of China under Grant No.60496325,60473069(国家自然科学基金).
Received 2007-02,Accepted 2007-04.

这些访问方式都只采用语法匹配(Syntactic Matching),而没有利用数据之间的语义关系(如同义 Synonymy,同名异义 Homonymy,上下位 Hyponymy,转喻 Metonymy,反义 Antonymy 等)进行语义匹配(Semantic Matching),导致它们的查全率(Recall Rate)和查准率(Precision Rate)往往都不太令人满意[2,3]。例如同义词(Synonym)会降低查询结果的查全率[2],使得该查出来的结果没有查出来,而同名异义词(Homonym)会降低查询结果的查准率[2],使得不该查出来的结果也查出来了。

随着本体(Ontology)和语义网(Semantic Web)技术的不断发展和应用,基于本体的关系数据库语义检索(Ontology-based Semantic Search over Relational Databases,SemSORD)也越来越成为研究热点。所谓本体,通俗地讲,是用来描述某个领域甚至更广范围内的概念以及概念之间的关系[4],是概念和概念之间关系的集合。目前,本体已经被广泛应用于语义网、知识工程(Knowledge Engineering)、信息检索(Information Retrieval,IR)以及信息集成(Information Integration)等方面[5]。所谓语义检索(Semantic Search),是利用数据语义进行语义匹配,通常指对语义 Web 上文档进行的检索[6,7],这些语义 Web 文档是用语义 Web 语言描述的可供 Web 用户访问的在线文档,而语义检索的关键是利用本体标注文档,并计算文档之间以及查询与文档之间的语义相似性(Semantic Similarity)[8-12],实现基于本体的查询(Ontology-based Query,OBQ)[13]。本体在信息检索领域也得到广泛关注和研究[14,15]。基于本体的语义检索可以有效提高查询结果的查全率,同时也能改善查准率。而本体最初用到数据库领域,主要是用来实现基于语义的信息集成[16,17],即把数据库中的模式信息(如表名、表的属性名)都映射到一个给定的本体上,从而实现基于本体的用户查询与各自数据库查询的转换,最终实现多个数据库的语义集成和查询。文章主要分析和研究单个关系数据库上基于本体的语义检索问题。

SemSORD 不同于 IR 或语义 Web 上的语义检索。这是因为 IR 领域所处理的文本数据库(Text Database)中的文档一般比较大,而且没有结构;语义 Web 文档通常是比较大的所谓半结构化的 XML 文档;而关系数据库比文本数据库具有更丰富的数据结构,也不同于 XML 文档的结构。另外,关系数据库设计的规范化导致一个查询结果通常分布在多个元组上。因此,IR 或语义 Web 领域的一些检索策略,并不能充分利用关系数据库的特点[18]。

意大利 Napoli"Federico II"大学的 Piero Bonatti 和美国 Maryland 大学合作提出基于本体扩展的关系(Ontology-Extended Relation,OER)模型以及相应的关系代数[19],主要为了实现异构数据库在语义级上的集成。美国 Maryland 大学的 Octavian Udrea 还提出了一种约束概率本体(Constrained Probabilistic Ontology,CPO)扩展的关系模型和关系代数[20],以实现关系数据库语义检索,提高数据库查询的查全率。Oracle 公司研究了关系数据库上存储 OWL 格式本体的方法,以及实现若干自定义本体操作函数来扩展 SQL 语言以提高关系数据库语义检索能力[21]。IBM 公司研究了利用 SPARQL[22]语言在关系数据库上执行语义查询的方法[23]。德国 Humboldt-Universit¨at zu Berlin 大学的 Chokri Ben Necib 等人提出了基于本体的查询扩展方法来实现关系数据库语义检索[24]。第 3 章将对这些工作进行分类、分析和研究。

近年来,中国人民大学数据工程与知识工程教育部重点实验室研究了关系数据库关键词检索(Keyword Search over Relational Databases,KSORD)技术[1,25],以及领域本体的构建和管理[26,27]、本体学习[28]、本体标注[29]技术,并建立了经济学领域本体,实现一个经济学语义 Web 示范平台[30]。该平台把经济学领域本体、经济学资源以及相应的标注数据都存储在关系数据库中。这个语义 Web 示范平台的一个核心技术就是基于本体的关系数据库语义检索(SemSORD)。从 KSORD 技术出发,研究了一个扩展关系数据库关键词检索系统 SEEKER[25]的 SemSORD 系统原型 Si-SEEKER[26]。

首先探讨 SemSORD 的一般概念,如关系数据库所包含的语义,语义检索的基础以及语义检索的一般过程;第 3 章详细介绍目前在单个关系数据库上实现语义检索的几种主要方法;第 4 章介绍基于 KSORD 技术实现的 SemSORD 系统原型 Si-SEEKER;第 5 章分析并提出了 SemSORD 研究挑战和发展趋势;最后,对全文进行总结。

Authors

rowid	AuthorId	Name
a1	aid1	厉以宁
a2	aid2	龙永图

Cites

rowid	CitingId	CitedId
c1	pid1	pid2
c2	pid3	pid6
c3	pid6	pid8

Papers

rowid	PaperId	Title	Keywords
p1	pid1	论当代中国的银行危机	银行危机
p2	pid2	中国入世后的信用体系建立	信用体系
p3	pid3	经济风暴产生的原因初探	经济风暴
p4	pid4	国际收支体系平衡	收支体系
p5	pid5	世界货币政策比较	货币政策
p6	pid6	论资本主义经济恐慌	经济恐慌
p7	pid7	论资本主义生产过剩危机	生产过剩
p8	pid8	生产不足与生产过剩	生产不足

Writes

rowid	AuthorId	PaperId
w1	aid1	pid1
w2	aid1	pid3
w3	aid1	pid4
w4	aid1	pid5
w5	aid2	pid2
w6	aid2	pid3
w7	aid2	pid4
w8	aid2	pid7

图 1　EPDB 数据示例

Fig.1　Part of EPDB Data

2　关系数据库语义检索

以经济学领域论文数据库(Economics Paper Database,EPDB)为例(如图 1),来描述关系数据库语义检索。假设 EPDB 有 4 个表:Authors(Authorid,name)记录作者信息,Papers(Paperid,Title,Keywords)记录经济学领域历年发表的论文信息;Cites(CitingId,CitedId)记录论文之间的引用信息;Writes(AuthodId,PaperId)记录论文的作者信息,其中带下划线的属性或属性组合为各表的主键,Cites 表的外键 CitingId 和 CitedId,以及 Writes 表的外键 PaperId 都引用 Papers 表中的主键 PaperId,而 Writes 表的外键 AuthorId 引用 Authors 表的主键 AuthorId。注意:图 1 中除了 Authors 中人名是真实的外,其他信息都是虚构的;Papers 表的 Keywords 属性通常包含多个关键词,这里为了描述方便,假设只包含一个关键词列。图 2 是描述图 1 数据语义联系的数据图(结点表示图 1 中的元组,有向边表示元组之间外键引用关系,边方向是从主键指向外键)。

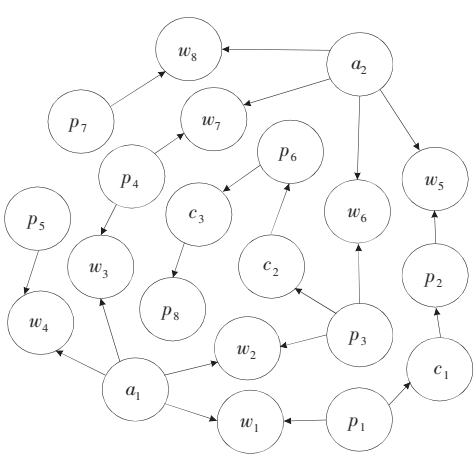


图 2　EPDB 数据图示例

Fig.2　Part of EPDB data graph

2.1　关系数据库的语义

关系数据库由两部分数据组成,一部分是元数据(Meta-data),也称模式(Schema),如表名、表的属性名、数据类型名、主键和外键等完整性约束名;另一部分是真正的数据,从数据粒度上又可以分为元组和属性值。关系数据库的元数据描述了其存储的数据之间的简单语义关系(图 2 描述了图 1 数据的主外键引用关系)。但是,关系模型只有很有限的语义描述能力,并不能完全描述关系数据库数据之间丰富的语义关系。例如图 1 中 Papers 中的元组 p_3、p_6、p_7、p_8 都是关于"经济危机"的论文,然而 Title 属性值中并没有显示地包含"经济危机"这几个字,关系模型就不能表示这些元组数据之间这种内在的语义联系,而本体则可以进一步描述关系数据库的语义。图 3 是利用经济学领域本体(Economics Ontology)标注图 2 数据的语义关系图,虚线左边(图 3(a))是经济学领域本体示例,虚线右边(图 3(b))是经济学领域数据资源示例(实际上是图 2 表示的内容)。从图 3 中可知,p_3、p_6、p_7、p_8 都通过"经济危机"这个概念联系在一起了,其他的元组也跟相应的概念联系在一起,并且这些概念与概念之间也是紧密联系的,看似毫无关系的元组数据通过本体揭示了它们内在紧密的语义联系。这里数据标注粒度是元组(实际上主要标注的是 Papers 表 Title 属性值),如果标注粒度为属性值,则需要在相应数据图中增加属性值节点。这里把关系数据库的语义分为两级,一级是元数据级的语义(Meta-data Level Semantics,MLS),另外一级是数据级的语义(Data Level Semantics,DLS)。利用不同级别的语义可以实现不同的语义检索,如 IBM Almaden 研究中心研究具有大量元数据(例如具

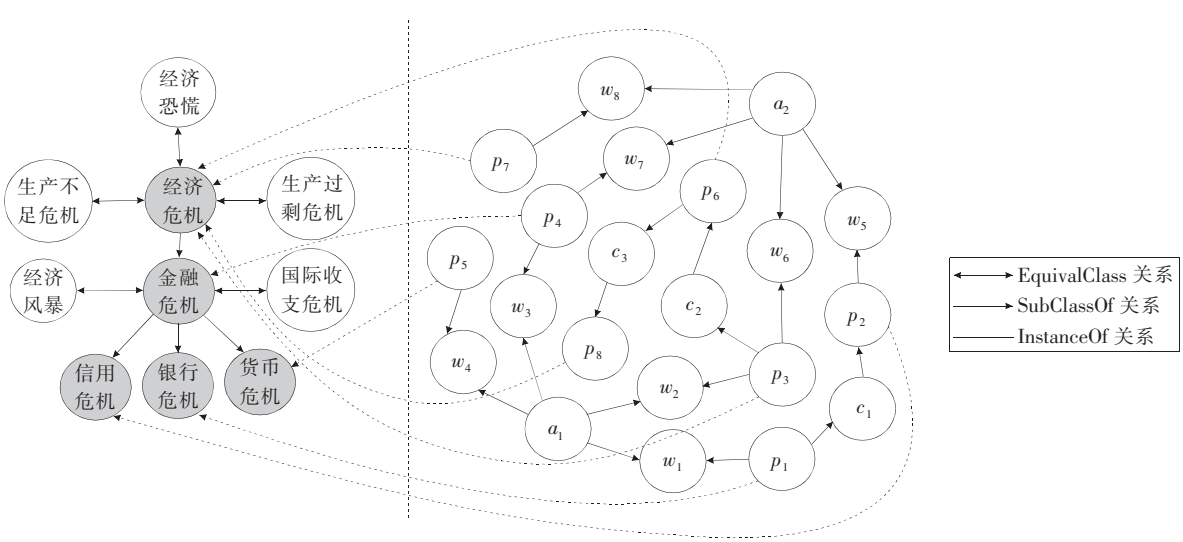
有 500 个表,每个表具有 50 以上的属性)的数据库中实现基于模式语义的数据库检索[32],这是 MLS 级的数据库语义检索，而文章更关注 DLS 级的数据库语义检索。实际上,DLS 级的数据库语义检索,并不是不需要 MLS,也要利用 MLS,甚至要扩充 MLS。

本体可以用来描述关系数据库的两级语义,可以从元数据或数据中抽取、识别出概念,然后映射到已有本体中的相应概念，或者基于本体学习技术[4],构建一个新的本体；也可以直接使用已有本体中的概念来描述元数据和数据，这两种方法都称为本体标注。根据关系数据库的语义级别,这些概念又可以分为:关系概念(Relation Concept,即从元数据的表名中识别出来的概念)、属性概念(Attribute Concept,即从元数据的属性名中识别出来的概念)和值概念(Value Concept,即从数据中识别出来的概念)[24]。

虽然一个关系数据库一般存储某个特定领域的数据，但是光有这个特定领域的本体并不能完全描述这个数据库的语义，需要多个本体来描述数据库语义。一般来说,某个特定领域的数据库除了需要该特定领域的本体外，其关系概念和属性概念可能也需要一个单独的本体来描述，而值概念根据属性概念的不同,也可能需要不同的本体来描述。例如图 1 中的经济学领域论文数据库,图 3(a)中的经济学领域本体可以描述 Papers 论文数据，但是不能描述元数据的语义,如作者和论文之间的写作关系、论文与论文之间的引用关系等，这需要一个专门描述论文写作和引用关系的本体。图 1 中的 Authors 省略了作者的国籍属性，经济学领域本体也不能描述这个属性值的数据语义,需要用关于国籍的领域本体来描述。

2.2 关系数据库语义检索的基础

本体是实现关系数据库语义检索的基础。最初,实现语义检索主要依赖叙词表(Thesaurus)或者是分类表(Taxonomy),例如 Oracle 数据库管理系统的文本检索功能支持基于叙词表的语义检索，但随着本体和语义 Web 技术快速发展和应用,在语义 Web 和信息检索领域,本体已成为语义检索的基础。因为本体可以更为形式化地描述数据之间丰富的语义关系,而且支持推理功能,本体也逐渐成为关系数据库语义检索的研究热点和基础。本体可以用类(Classes)、关系(Relation)、函数(Functions)、公理(Axioms)和实例(Instances)来建模[5]。本体依照领域依赖程度,可以分为顶级本体(Top-level ontology,描述最普通的概念及概念之间的关系)、领域(Domain ontology,描述特定领域如经济学的概念及概念之间的关系)、任务(Task ontology,描述特定任务或行为中的概念



(a)经济学领域本体示例 (b)经济学领域数据资源示例

(a)Part of economics ontology (b)Part of EPDB data

图 3 经济学领域本体及标注数据示例

Fig.3 Part of economics ontology and annotated EPDB data

及概念之间的关系)和应用(Application ontology,描述依赖于特定领域和任务的概念及概念之间的关系)等四类[5,33]。

本体合并(Ontology Merging)、集成(Ontology Integeration)是实现关系数据库语义检索的基础技术。关系数据库语义主要通过多个领域本体来描述,也可以辅以集成的顶级本体、构建的任务本体和应用本体来描述。这些本体可以是已经存在的,也可以是通过学习关系数据库的元数据和数据而构建的,甚至可以随着数据库数据的变化而进化,这是本体构建和学习问题[4]。采用多个本体描述数据库语义,必然就涉及到本体合并或本体集成问题。

本体标注也是关系数据库语义检索的基础技术。简单地讲,本体标注是把本体和数据联系在一起[29,34],以便建立基于本体的语义索引[35]。对于已经存储大量数据的关系数据库,手工标注数据工作量巨大,耗时耗力,但是完全自动化的标注技术还不是很成熟。关系数据库数据具有自身特点(如每个元组或属性值数据比较小),需要研究如何标注关系数据库数据,以便建立有效的语义索引。目前,自动化标注和人工标注相结合的半自动化标注是比较有效的标注方法。

后续的研究内容不涉及本体构建、学习、标注等具体技术细节,而假设本体已经存在,关系数据库数据已经标注。

2.3　关系数据库语义检索的过程

图4描述了基于本体的关系数据库语义检索的一般过程。右边本体标注器(Onto Annotator)执行数据标注、语义索引建立的功能。数据标注一般是预处理生成的,标注过程中可能需要领域专家的辅助。如果数据库数据有变化,也可以通过触发器自动触发Onto Annotator进行标注。

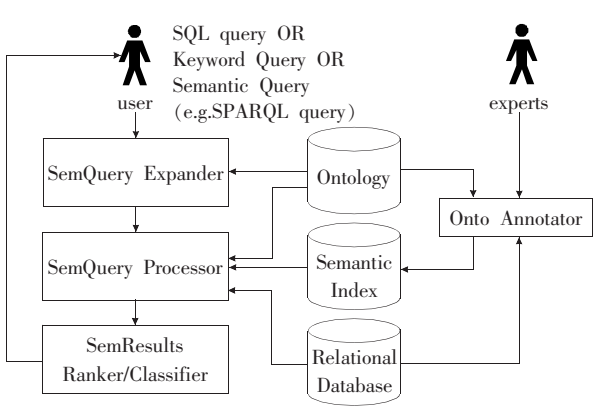


图4　SemSORD的通用框架

Fig.4　General architecture of SemSORD

当通过数据标注建立了语义索引后,用户就可以提交查询了。根据不同的实现方法,用户可以提交SQL查询、关键词查询,甚至是SPARQL形式的语义查询;用户查询也可能需要语义查询扩展器(SemQuery Expander)进行查询扩展,然后提交语义查询处理器(SemQuery Processor)执行;最后,查询结果进行语义查询结果排序器(SemResults Ranker)或者分类器(SemResults Classifier)处理后呈现给用户。根据关系数据库语义检索的实现机制,可以把现有基于本体的关系数据库语义检索方法分为:查询扩展方法、关系模型和关系代数扩展、SQL扩展、语义映射以及基于KSORD技术等5种方法。后续第3节和第4节对这些方法进行了详细分析和研究。

3　基于本体的语义检索方法

3.1　查询扩展方法

查询扩展(Query Expansion)是信息检索领域研究的重要问题[36]。随着本体的应用和发展,基于本体的查询扩展在信息检索领域也得到越来越多的关注和研究[37-39]。

2003年Chokri Ben Necib等人提出基于本体的数据库查询扩展方法,可以把一个SQL查询扩展成多个查询加以执行[24],这里称为OntoQE(Ontology-based Query Expansion)方法。他们选择域关系演算(Domain Relational Calculus,DRC)来表示用户查询:$Q=\{s|\Psi(s)\}$,其中s是元组变量,$\Psi(s)$是建立在原子变量和操作集合上的公式:$u\in R$,$v\theta w$,和$u\theta t$,R是关系名,u、v和w是元组变量,t是常量,θ是算术比较运算符(=,<,等等),变量可以绑定,也可是自由变量,并且公式和变量还可以使用逻辑运算符($\vee$、$\wedge$和$\neg$)递归定义。

OnotQE方法分为三步:预处理(Preprocessing)、执行(Execution)和后处理(Post-processing)三个阶段。

(1)预处理阶段:从附带到该数据库上的本体中抽取新的关键词(Term),然后把用户查询转换成另外一个查询。这个过程中要用到能够完成如下功能的一组转换规则:

①使用 t 的同义词(Synonym)改变查询的选择条件,从而扩展查询。本体必须有基于同义词和上下位(Hyponym)关系推理的功能;

②使用语义上等价的查询条件替换原来的查询条件。

(2)执行阶段:执行转换后的查询,检查查询结果,如果查询结果为空,则执行后处理阶段。

(3)后处理阶段:按照如下三步产生扩展查询:

①使用本体关系的上一级概念(即父概念)来修改查询条件;

②执行查询,如果结果集为空,重复上一步;

③如果找到需要的查询结果或者没有可以替换的概念,停止执行,否则重复第一步。

OntoQE 方法也只适合于那些属性值为本体中相应概念值的属性,没有利用标注数据和语义索引。OntoQE 采用的查询扩展方法也比较简单,只是利用了同义词、父概念等来扩展查询,特别是简单利用父概念来扩展查询时,查询结果的查全率可能大大提高了,但是查准率可能很低。

3.2 关系模型和关系代数扩展方法

2003 年意大利 Napoli "Federico II"大学的 Piero Bonatti 和美国 Maryland 大学合作提出基于本体扩展的关系(Ontology-Extended Relation,OER)模型以及相应的关系代数[19]。但是该方法主要是为了实现异构关系数据库在语义级别上的集成,并且该方法中每个关系只附带一个本体,而通过关系数据库语义分析知道,每个关系的不同属性可能需要不同的本体来描述其语义。2004 年美国 Maryland 大学的 Edward Hung 等人提出了基于本体扩展 XML 数据库 TAX 代数的方法[3],并实现了 TOSS 原型系统。2005 年美国 Maryland 大学的 Octavian Udrea 提出了一种约束概率本体(Constrained Probabilistic Ontology,CPO)扩展关系模型和关系代数[20]的方法,以实现关系数据库语义检索,从而提高数据库查询的查全率,并且实现了一个原型系统 PARQ(Probabilistic Answers to Relational Queries)。PARQ 方法中,一个关系的每个属性都可以附带一个 CPO,从而可以很好地描述关系数据库语义。上述三种方法的共同之处就是都需要用到本体合并、集成技术。由于篇幅有限,主要介绍 PARQ 方法。

图 5 是 PARQ 定义的约束概率本体示例,该本体描述天文学星体分类和星体数据的语义关系。星体数据库模式为 *Stars*(*luminosity*,*mass*,*density*,*size*,*name*,*star_type*),其中前 4 个属性类型均为 real 类型,*name* 属性为字符类型,*star_type* 属性为 starType 类型,并且该属性附带图 5 所示的 CPO。图 5 中的有向边及其标签数字的含义是:如果边 $u\rightarrow v$ 标有数字 p,则类 v 中任意对象事实上在类 u 中的概率为 p。用户可以执行如下典型查询:

(1)Select all *stars* that are *small*.如果某个实体显示地标注为 *small*,则应该返回给用户;任何标注为 *dwarf*、*neutron* 等的实体,根据类和子类(class-sub-

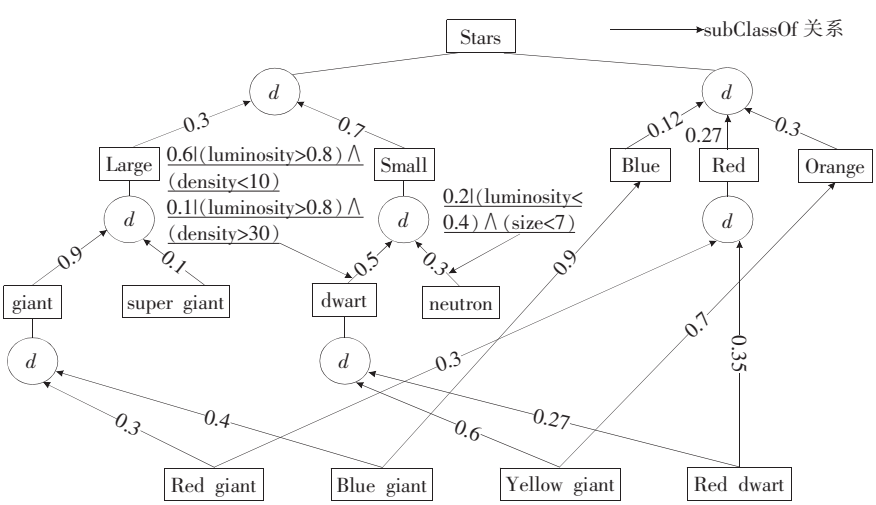


图 5 PARQ 约束概率本体示例:星体数据及其本体

Fig.5 An example of CPO in PARQ:star data and its ontology

classs)关系,也应该返回给用户;考虑某个实体 e 标注为 *star*,则该实体有 70%的概率为 *small*,所以如果系统定义的阈值(根据不同的数据集和 CPO 通过实验确定)$p_{thr} \leqslant 0.7$,则该实体也应该返回给用户。

(2)Join 例子.假设有两个表包含星体数据:R_1={*name*,*density*,*star_type*}和 R_2={*mass*,*location*,*star_type*},需要在 *star_type* 属性上把 R_1 和 R_2 连接起来,并且假设 R_1 某个元组的 *start_type* 值为 *red*,R_2 中某个元组 *start_type* 值为 *crimson*,它们在语义上是一样的,普通关系代数中的 join 操作不能连接这两个元组,而使用 CPO 增强的关系,就可以按照条件 R_1.*star_type*=*red* and R_2.*star_type*=*crimson* 连接成一个新元组。

(3)Select all *stars* that are *dwarf* with a 50% probability or more.所有 *star_type* 为 *dwarf*,*yellow_dwarf*,*red_dwarf* 的实体都返回给用户;如果 *star_type* 为 *small*,则根据要根据实体的 *luminosity* 和 *density* 值计算其约束概率,如果为 0.6 或 0.5,则返回给用户,否则不返回给用户。

文献[20]中给出了 CPO 增强关系模型的形式化定义,如下:

CPO 增强的关系(CPO enhanced relation)定义:一个 CPO enhanced relation 是一个三元组:(T,R,f),其中 T 是一组 CPO,R 是具有模式(A_1,…,A_n)的关系,f:{A_1,…,A_n}→T∪{⊥}是一个映射函数,如下:

(1)$\forall S \in T, \exists j(f(A_j)=S)$,直观的解释就是,$T$ 中的任何 CPO 都至少附带到一个属性上。

(2)$\forall S \in T$,假设 $f(A_i)=S$,则 $dom(A_i) \subseteq C_s$ 必须成立,其中 C_s 是约束概率本体 S 的概念集合。直观解释就是,任何 CPO 对于其附带的关系和属性来说都是定义良好的(Well defined)。

例如,假设 S 标记图 5 中的 CPO,则({S},*Stars*,f)是一个 CPO enhanced relation,其中 f(*star_type*)=S,而当 $A \in Stars$,$A \neq star\_type$,$f(A)=\perp$,并且 S 对于关系 *Stars* 和属性 *star_type* 来说是定义良好的。

根据 CPO enhanced relation 的定义,PARQ 定义了扩展的关系代数操作:

(1)简单选择条件(Simple condition)。假设 C 是 CPO 的类(或概念)集合,CPO 附带到关系 R 的属性{A_1,…,A_m}上。则 A_i $\underline{in}$ T 是简单条件,其中 $T \subseteq dom(A_i) \subseteq C (1 \leqslant i \leqslant m)$。

(2)概率条件(Probability condition)。(δ,p)是一个概率条件,其中 δ 是一个简单条件,p 是[0,1]区间的一个有理数(Rational Number)。

(3)选择条件(Selection condition)。选择条件是任何简单条件或概率条件使用逻辑操作符 ∨、∧ 和 ¬ 的组合。例如(*star_type* $\underline{in}$ {*dwarf*,*giant*},0.5)∧¬(*star_type* $\underline{in}$ {*blue*})是一个选择条件。

(4)选择操作(Select)。假设有一个 *CPO enhanced relation*:(S,R,f),其中 S=(C,⇒,*me*,ψ),R=(A_1,…,A_m),则定义 CPO 增强的选择操作如下:

①简单选择(Simple selection)。$\sigma_{A_i \underline{in} T}(R)$={$t \in R | t.A_i \in T \vee \exists c \in T$ s.t. $t.A_i \Rightarrow^{*} c$}。

②概率选择(Probability selection)。$\sigma_{(A_i \underline{in} T,p)}(R)$={$t \in R | (t.A_i \in T) \vee (\exists c \in T$ s.t. $t.A_i \Rightarrow^{*} c) \vee (\exists c' \in T$ s.t. $c' \rightarrow_p t.A_i$ and $t.A_i$ meets every constraint in $\Gamma(c' \rightarrow_p t.A_i)$)}。

③复合选择(Composite selection)。$\Delta=\Delta_1$ $\underline{op}$ Δ_2,$\sigma_{\Delta}(R)=\sigma_{\Delta_1}(R)$ $\underline{op}'$ $\sigma_{\Delta_2}(R)$,其中 $\underline{op}$ 为逻辑操作 ∨、∧、¬,则 $\underline{op}'$ 为相应的集合操作∪、∩、-。

(5)投影操作(Projection)与笛卡尔积操作(Cartesian Product)。与普通的关系代数操作一样。

(6)连接操作(Join)。考虑两个 CPO 增强的关系(T_1,R_1,f_1)和(T_2,R_2,f_2),并且假设它们的属性重命名之后没有公共属性,$\xi(A_1,A_2)$是连接 R_1 的属性 A_1 和 R_2 的属性 A_2 的连接条件,则(T_1,R_1,f_1)和(T_2,R_2,f_2)在一个有限互操作约束集(Interoperability constraints)I 下是可以连接的(CPO-Join compatible),当且仅当 $f_1(A_1)$ 和 $f_2(A_2)$ 这两个 CPO 在 I 和 $\xi(A_1,A_2)$ 下是可以合并的(mergeable)。据此,可以定义 Join 操作。假设(T_1,R_1,f_1)和(T_2,R_2,f_2)在 I 和 $\xi(A_1,A_2)$ 下是可以连接的,并且假设 S 是 $f_1(A_1)$ 和 $f_2(A_2)$ 合并的一个证据(witness),则 Join 之后的结果为(T,R,f):

①$R=R_1 \rhd\lhd_{\xi(A_1,A_2)} R_2$。

②$T=(T_1 \cup T_2 \cup \{S\})-\{S_1,S_2\}$,其中 $S_1=f_1(A_1)$,$S_2=f_2(A_2)$。

③f 为两个函数 f_1、f_2 的简单合并。

(7)合并操作(Union)。假设两个 CPO 增强的关系 (T_1,R_1,f_1) 和 (T_2,R_2,f_2) 具有相同的模式,对每个属性 A_i,让 $S_i^1=f_1(A_i)$,$S_i^2=f_2(A_i)$,且 I_i 是一组互操作约束,则如果 S_i^1,S_i^2 在 I_i 下有一个证据 S_i,则 (T_1,R_1,f_1) 和 (T_2,R_2,f_2) 在 I_i 下是可以合并的(CPO-union compatible)。因此,定义 $(T_1,R_1,f_1)\cup(T_2,R_2,f_2)=(T,R,f)$,其中:

①$R=R_1 \cup R_2$,∪是普通的关系代数合并操作;

②$T=\{S_i|A_i$ 是 R_i 的一个属性$\}$;

③$f(A_i)=S_i$。

(8)交操作(Intersection)与差操作(difference)。执行这两个操作的关系也必须是可以合并的,它们的操作类似上面定义的合并操作,只不过其中的结果关系 $R=R_1 \cap R_2$ 或 $R=R_1-R_2$。

至此,已经介绍了 PARQ 系统约束概率本体,以及扩展的关系模型和关系代数。PARQ 方法还提出了具体的本体合并算法、互操作约束推理算法、以及自动分配和计算 CPO 概率的方法,由于篇幅所限,包括上述扩展关系模型和关系代数中没有说明的符号和定义,不再详细阐述,具体内容请参见参考文献[20]。

PARQ 提出了基于约束概率本体扩展关系模型和关系代数的方法,以实现关系数据库语义检索,可以有效提高数据库查询结果的查全率。但该方法也有明显的不足,附带 CPO 的关系属性取值只能来自 CPO 中的概念,因此只能适用于简单字符类型的属性。PARQ 也没有办法保证该属性的取值全部来自 CPO 中的概念,但 Oracle 公司提出基于本体的引用约束[40]正好可以保证这一点。另外一个不足是,虽然提高了数据库查询结果的查全率,但查准率有较大的下降。文献[20]显示,当本体概念数达到 478 个,系统概率阈值 p_{thr} 设置为 0.6 时,查全率达到 80%以上,此时查准率仅仅为 50%左右,而不采用 PARQ 方法的数据库查询全率和查准率分别为 52.7%和 100%。因此,如何保证提高数据库查全率的同时,查准率不降低太多,是 PARQ 方法面临的一个挑战。

3.3 SQL 扩展方法

2004 年 Oracle 公司提出在关系数据库中存储 OWL 格式本体,然后扩展 SQL 以提供有关本体的操作,从而实现关系数据库语义检索的方法[21]。该方法是基于本体扩展 SQL 的方法,这里称为 OntoSQL(Ontology-based SQL)方法。具体来说,OntoSQL 定义了 6 个系统表(如图 6):本体表 *Ontologies*、概念表 *Terms*、属性表 *Properties*、属性值表 *PropertyValues*、约束表 *Restrictions* 和概念之间的联系表 *Relationships*,每个表中带下划线的属性为该表的主键,而表之间的外键通过图 6 中的有向边表示。从该模式可以看出,系统可以存储和管理多个本体。

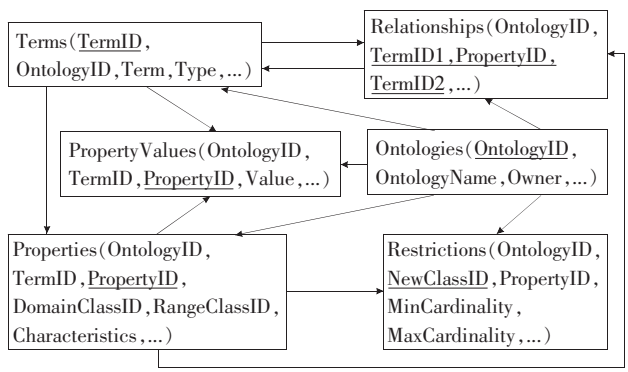


图 6 OntoSQL 存储 Ontology 的模式图

Fig.6 Schema graph for OntoSQL to Store Ontology

然后,OntoSQL 方法实现一组 SQL 操作符(实际上是一组自定义函数)以提供有关本体的操作。这些操作符是:

(1)ONT_RELATED(Term1,RelType,Term2,OntologyName) RETURNS INTEGER:基本语义匹配操作,如果两个概念(Term1 和 Term2)在本体(OntologyName)中具有语义关系(RelType),则该函数返回 1;否则返回零。说明:Term1 和 Term2 只能是简单概念,不能是复合概念;而 RelType 可以是一个由逻辑运算符(AND,NOT 和 OR)构成的复杂关系类型表达式。

(2)ONT_EXPAND(Term1,RelType,Term2,OntologyName) RETURNS ONT_TermRelTableType:语义匹配操作,返回匹配的概念集合。这里 ONT_TermRelTableType 是"CREATE TYPE ONT_TermRelType AS OBJECT(Term1Name VARCHAR(32),PropertyName VARCHAR(32),Term2Nmae VARCHAR(32),Term

Distance NUMBER,TermPath VARCHAR(2000))"和"CREATE TYPE ONT_TermRelTableType AS TABLE OF ONT_TermRelType"两个 SQL 语句创建的自定义类型。

(3)ONT_DISTANCE(NUMBER) RETURNS NUMBER:与 ONT_RELATED 连用,返回与要匹配概念具有最小距离(Smallest Distance)或者是最短路径(Shortest Path)的概念。

(4)ONT_PATH(NUMBER)RETURNS VARCHAR:与 ONT_RELATED 连用,返回与要匹配概念具有最小距离(Smallest Distance)或者是最短路径(Shortest Path)的路径字符串。

ONT_RELATED 和 ONT_EXPAND 操作符都要计算"*RelType*"关系的传递闭包(Transitive Closure),这是通过 Oracle 提供的传递闭包查询"SELECT... FROM...START WITH<condition>CONNECT BY<condition>"来实现的。实际上,包含上述两个本体操作符的 SQL 查询,都会通过查询重写而转换为相应的传递闭包查询。如果本体中含有大量的概念,则传递闭包的计算是非常耗时的。为此,OntoSQL 方法还专门创建一个传递闭包表(*Transitive Closure Table*(*RootTerm*,*RelType*,*Term*,*Distance*,*Path*)),以存储预先计算的本体中所有概念(*RootTerm*)在所有关系(*RelType*)上的传递闭包(*Term*),同时也把相应的距离(*Distance*)和路径(*Path*)记录下来。另外为了加快本体匹配操作,OntoSQL 定义一种 ONT_INDEXTYPE 类型的索引,可以在具有本体概念的列上创建一个索引。

例如,从 EPDB(如图 1)中检索关于"经济危机"的论文,假设 *Papers* 表的 *Keywords* 属性最多只包含一个关键词,则可以通过 SQL 查询 "SELECT *p.Title* FROM *Papers p* WHERE ONT_RELATED (*p.Keywords*,'*subClassOf* ','经济危机','*Economics Ontology*')= 1" 来检索;Papers 表中 5 个元组会检索出来,与"经济危机"直接相关的元组 p_3、p_6、p_7、p_8,以及与"银行危机"相关的 p_1。

OntoSQL 方法充分利用 Oracle 数据库管理系统提供的传递闭包查询、自定义函数和自定义类型功能扩充 SQL,以提供本体操作符,实现关系数据库管理系统提供语义检索的基本方法,并提供基于概念对称属性(Symmetry)和传递属性(Transitivity)的简单推理功能。但是,OntoSQL 目前不支持本体合并,也不支持用户自定义的推理规则。OntoSQL 方法中存储的多个本体也并没有指明与哪个表的哪个属性相关联,这种关联需要用户在构造 SQL 查询时自己把握,即需要用户准确掌握本体与数据之间的语义联系。OntoSQL 另外一个更主要的问题是,目前只适用于存储有单个本体概念的简单字符类型属性(也称为关键词列,Keyword Column)进行语义匹配,没有内嵌概念抽取功能,也没有利用本体标注数据来实现语义匹配。因此,上述查询关于"经济危机"的论文例子,如果 ONT_RELATED 中的匹配属性由 *p.Keywords* 换成 *p.Title*,就会一个元组也查不出来,而实际上按照图 3 标注数据,*Papers* 表中有 5 个元组是相关结果。

即使 OntoSQL 方法应用到具有单个本体概念的关键词列上,又如何保证该属性值是某个本体中的概念呢?Oracle 公司 2006 年提出了基于本体的引用约束(Ontology-based Referential Constraint,ORC)[40],ORC 可以为关键词列建立引用约束,其属性值只能来自其引用的本体中的概念。如 EPDB(图 1)中 *Papers* 表的 *Keywords* 属性,为了保证其值都来自经济学领域本体,就可以为该列创建一个 ORC:ALTER TABLE *Papers* ADD CONSTRAINT *constr_keywords_ORC* FOREIGN KEY(*Keywords*) REFERENCES(SELET *Term1Name* FROM TABLE(*ONT_EXPAND*(NULL,'*subClassOf*','*RootConcept*','*Economics Ontology*'))),其中 *RootConcept* 假设为 *Economics Ontology* 的根概念,按照实际需要,*RootConcept* 也可以用该本体中其他概念替换。目前 ORC 也只能应用到具有单个概念值的关键词列上,Oracle 还将进一步研究 ORC 如何应用到具有多个概念值的关键词列上,如 *Papers* 表 *Keywords* 属性具有多个概念的情况。

3.4 语义映射方法

IBM T.J.Watson 研究中心采用 SPARQL 和 Minerva 研发了一个原型系统以支持关系数据库语义检索[23],该系统的体系结构如图 7。从图 7 可以看出,关系数据(Relational Database Tables)通过关系到语义映射器(Relational to Semantic Mapper)转换成

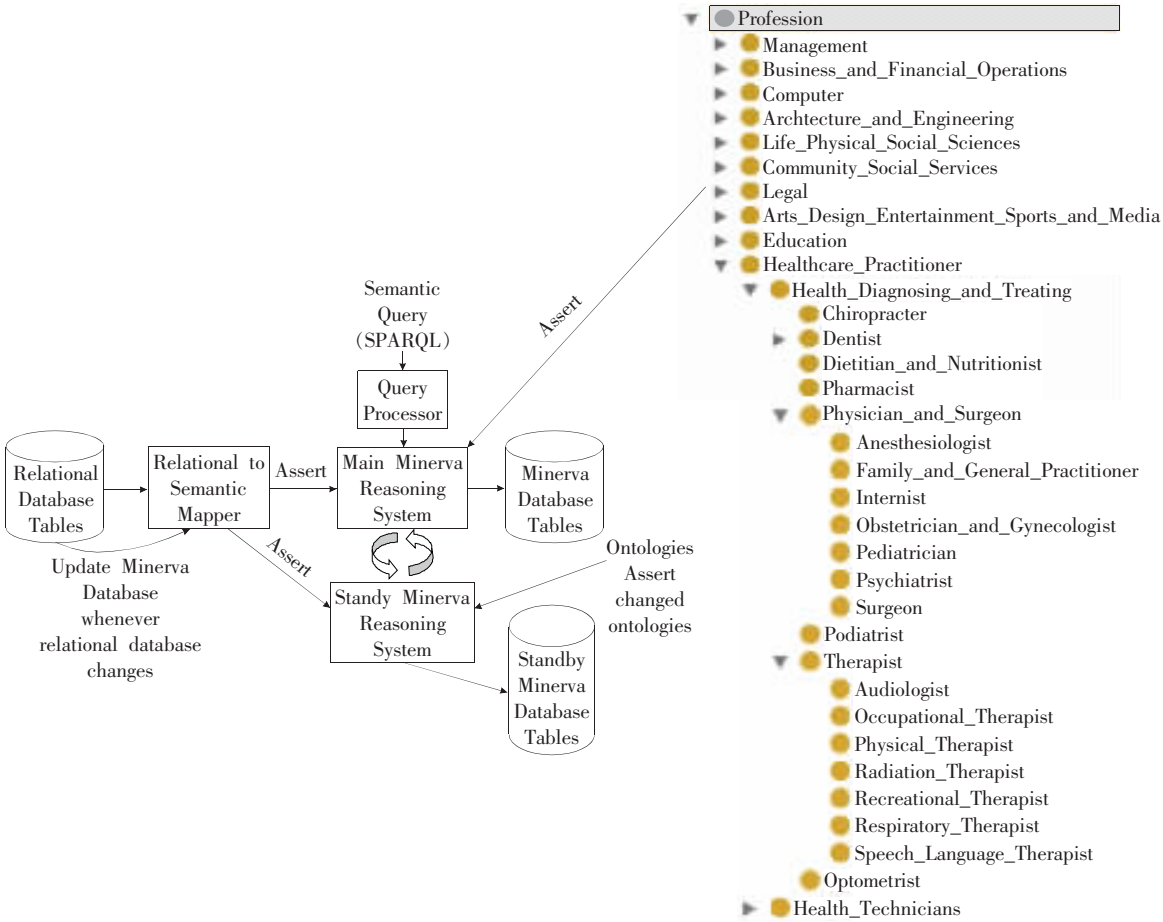


图 7 IBM T.J. Watson 研究的 OntoRSM 体系框架

Fig.7 Architecture of OntoRSM developed by IBM T.J.Watson

Minerva 推理系统(Main Minerva Reasoning System)的断言(Assert)存储在 Minerva 数据库表(Minerva Database Tables)中,用户就可以提交 SPARQL 形式的语义查询(Semantic Query)到查询处理机器(Query Processor),然后提交 Minerva 推理系统加以执行,最后返回用户语义查询结果。这里关系到语义映射器是关系数据库和 Minverva 推理系统之间的桥梁,因此,称这种实现关系数据库语义检索的方法为 OntoRSM(Ontology-based Relational to Semantic Mapping)方法。SPARQL 是语义 Web 上的查询语言[22],而 Minerva 是 IBM 研发的基于 RDBMS 存储的可伸缩的高效推理机。

Minerva 推理系统的一个关键特性是预先计算所有的推论(Inferences)并将它们和由关系到语义映射器转换而来的基本断言一起存储在后台关系数据库中,查询时就不用进行描述逻辑推理(Description-logic Reasoning),从而可以大大提高语义查询效率。Minerva 可以高效地处理描述逻辑中的 ABox 断言(描述实例的事实)的变化,但是不能高效地处理 TBox(描述概念的事实)的变化,因为对于大型本体来说,这种变化导致需要较长的时间重新预计算所有推论。因此,OntoRSM 体系结构设置两个 Minerva 推理系统,一个主推理系统(Primary Minerva),一个备用推理系统(Standby Minerva)。系统启动时,相关本体装载到主推理系统中,语义映射器把关系数据库中的元组转换成 Minerva 可以处理的描述逻辑断言,当关系数据变化时,关系数据库中相应的触发器可以自动通知语义映射器修改 Minerva 系统。当 TBox 变化时,系统启动备用推理系统,并装载本体和数据库,预先计算所有推论,然后和主推理系统切换。

OntoRSM 定义了三种查询结果:直接结果(Direct Results),即直接从关系数据库查询到的结果;推

理结果(Inferred Results),即利用关系数据和本体中的领域知识推理得出的结果;相关结果(Related Results),即利用关系数据、相似概念以及本体中的实例(individuals)推理得出的结果,包括那些并不严格匹配用户查询,但是与真正的结果在语义上相似的结果。

查询处理器可以利用概念层次属性、传递属性等各种信息重复执行查询扩展算法(Query Generalization Algorithm)以扩展原始用户查询,然后执行这些查询,直到产生的查询结果达到预先定义的结果数为止。

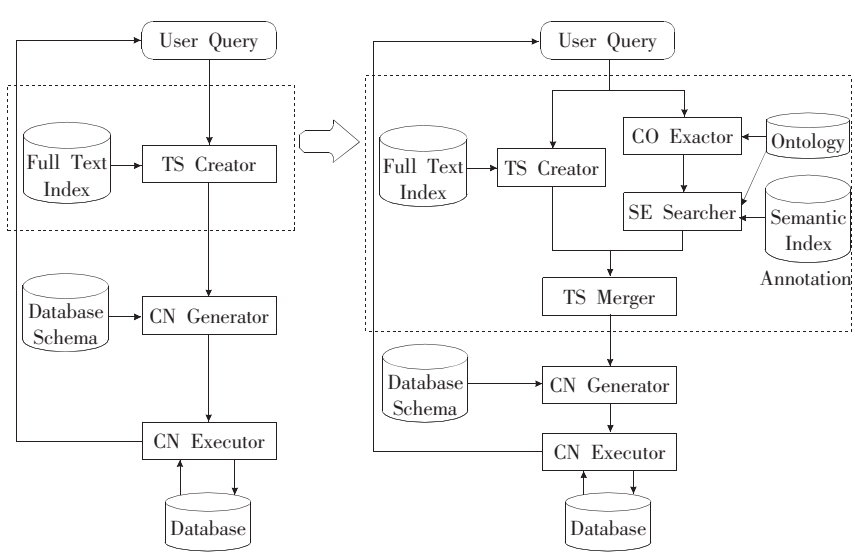
OntoRSM 方法的优点是可以在关系数据库上执行 SPARQL 形式语义查询,直接利用语义 Web 领域的语义检索技术以增强关系数据库语义检索能力。但是,SPARQL 查询语言与关系数据库的查询语言 SQL 并不匹配,因此会增加用户使用系统的难度。另外,OntoRSM 在规模并不大的本体(如 400 个概念、120 属性、500 个实例)上测试,还需要研究更好的 Minerva 数据库索引来减少查询响应时间,也需要研究更好的策略来生成相关结果(Related Results),由此可见,OntoRSM 方法的查询效率还有待提高。

4　基于本体和 KSORD 的关系数据库语义检索方法

近年来,作者研究了如何在关系数据库上实现关键词检索的技术(即 KSORD 技术),并实现了一个基于数据库模式图(Schema Graph,Gs)的 KSORD 原型系统 SEEKER[25]。KSORD 使得普通用户不用了解数据库模式,也不用懂得如何书写 SQL 查询,就可以像使用搜索引擎搜索 Web 一样来搜索关系数据库数据[1]。KSORD 技术一般是基于关系数据库管理系统(Relational Database Management System,RDBMS)提供的全文索引(Full-text Index)技术实现的,其查询过程也只是关键词的语法匹配而不是语义匹配,导致查询结果质量也往往不太令人满意。因此,提出一种语义候选网络(Semantic Candidate Network,SemCN)方法,利用本体来扩展 KSORD 技术以实现关系数据库语义检索。利用 SemCN 方法扩展 SEEKER,从而实现了一个 SemSORD 原型系统 Si-SEEKER(Semantic Improved SEEKER)[31]。下面介绍 SemCN 方法。

4.1　SemCN 方法概述

SemCN 方法是在基于模式图的 KSORD 技术基础上提出来的。首先,简要介绍基于模式图的 KSORD 原型系统 SEEKER。数据库模式图 G_s 是一个有向图,其结点表示数据库模式中的一个表(Table,或称关系 Relation),有向边主要表示表与表之间的外键引用关系。SEEKER 体系结构如图 8(a)所示。当用户提交关键词查询时,元组集生成器(TS Creator)利用数据库



(a)SEEKER　　(b)Si-SEEKER

图 8　SEEKER 与 Si-SEEKER 体系框架

Fig.8　Architecture of SEEKER and Si-SEEKER

全文索引生成元组集(Tuple Set,TS),每个 TS 就是针对数据库模式中具有文本属性的表生成包含关键词的元组的非空临时表,每个临时表继承其源表的主外键关系,并且扩展 G_s 生成元组集图(Tuple Set Graph,G_{ts})。然后,候选网络产生器(CN Generator)宽度优先遍历 G_{ts} 生成候选网络(Candidate Network,CN)[41,42]。最后,CN 执行器(CN Executor)采用某种 Top-k 查询算法执行 CN,生成最终查询结果返回给用户。最终查询结果是一棵棵元组连接树(Join Tree of Tuple,JTT)。所谓候选网络,是元组集连接树(Join Tree of Tuple Set,JTTS),也可以看作是要用来产生查询结果的 JOIN 表达式。由于全文索引是关键词的语法索引(Syntactic Index),而不是语义索引(Semantic Index),而 SEEKER 中的 CN 是通过全文索引产生元组集而生成的,因此称为语法 CN(Syntactic CN,SynCN)。

SEEKER 产生的 SynCN 缺少语义,导致查询结果质量往往不能令用户满意,所以提出了基于本体的 SemCN 方法来扩展 SEEKER 以提供关系数据库语义检索能力,并开发出 SemSORD 原型系统 Si-SEEKER,其体系结构如图 8(b)。从该图可以看出,一方面,用户提交的查询,仍然通过 TS Creator 生成包含关键词的元组集(Keyword-based Tuple Set,TS_k);另外一方面,概念抽取器(CO Extractor)利用本体将用户查询转换成概念查询,然后语义检索器(SE Searcher)利用本体和语义索引执行概念查询以生成语义元组集(Semantic Tuple Set,TS_s),元组集合并器(TS Merger)合并 TS_k 和 TS_s 以形成混合元组集(Composite Tuple Set,TS_c),这些合并之后的 TS_c 也是具有语义的元组集。由此,CN 产生器产生的 CN 也是具有语义的 CN,因此称为语义 CN(SemCN)。最后,CN 执行器仍然采用某种 top-k 查询算法执行 SemCN,生成具有语义的查询结果返回给用户。

概念抽取器在自然语言处理领域有很多研究[43],可以利用 Stanford Parser 生成一个简单的概念抽取器,将用户查询转换成本体中的概念查询。概念抽取器不是文章研究的重点内容。语义索引是通过本体标注技术建立的本体与数据库数据之间的联系,也是预先建立的类似全文索引的倒排索引(Inverted Index),也称为语义倒排索引(Semantic Inverted Index)。关系数据库中每个表的每个文本属性都通过手工标注、自动标注或者手工标注与自动标注相结合的半自动化标注技术标注成概念向量,然后存储在语义倒排索引中,形成所谓的语义索引。当数据库数据变化时,可以通过触发器自动触发自动标注器建立新增数据的语义索引,或者删除语义索引。当本体有变化时,需要重新预处理所有关系数据库数据,更新语义索引,如果本体包含有大量概念,数据库中也包含大量元组,则这种更新非常耗时。所幸本体也像数据库模式信息一样,一般是比较稳定的,不会经常变化。语义检索器和元组集合并器将在后续小节中介绍。

SemCN 的基本思想是:利用本体标注关系数据库数据,将每一个表中的每一个文本属性都表示成加权的概念向量,以建立语义索引;将用户关键词查询转换成概念查询;利用扩展向量空间模型(Generalized Vector Space Model,GVSM)和本体之间的层次结构计算概念查询和语义索引之间的语义相似性[8],将超过系统设定的语义相似性阈值 ε_{ssim} 的元组选择出来,生成语义元组集;提交后续 KSORD 模块执行,以生成语义查询结果。下面具体介绍 SemCN 的语义查询处理过程。

4.2 语义查询处理

概念抽取器把用户提交的关键词查询转换成本体中的概念查询,但是本体本身的不完善性,以及用户表达查询要求的不准确性,都可能导致用户提交的关键词查询不能转换成本体中的概念查询,因此,在本体指导下让用户形成更准确的语义查询,也是未来可以研究的内容。这里假设用户提交的关键词查询(Keyword Query,Q_k)总可以转换成某个概念查询,先给出几个相关定义。

假设关系 R 有 m 个文本属性和 n 个元组,a_i $(1\leqslant i\leqslant m)$代表 R 的文本属性,r 为 R 中的元组($r\in R$),并假设有一个本体 O。

定义 1 关键词查询(Keyword Query)记为 Q_k。一个关键词查询由一组关键词组成,记为 $Q_k(k_1,k_2,\cdots,k_l)$,k_j $(1\leqslant j\leqslant l)$为关键词,每个关键词都有一个权值 W_{qj}。假设 Q_k 为 OR 语义查询。

定义 2 概念查询(Concept Query)记为 Q_c。一个概念查询由相应的关键词查询 Q_k 转换而来，并由一组概念组成,记为 $Q_c(c_1,c_2,\cdots,c_{l_1}),c_j(1\leqslant j\leqslant l_1)$ 为本体 O 中的概念，每个概念 c_j 都有一个权值 W_{cj}。假设 Q_c 为概念之间的 OR 语义查询。定义概念查询向量如下：

$$\boldsymbol{Q}_c=\sum_{j=1}^{l_1}W_{cj}\boldsymbol{c}_j \tag{1}$$

概念查询向量中的概念权值 W_{cj} 可以通过概念在查询中出现的位置、概念在本体中的重要程度等因素来自动确定。

定义 3 概念文本属性(Concept Text Attribute)记为 a_i。关系 R 的文本属性 a_i 通过本体标注为本体 O 中的一组概念,记为 $a_i(c_1,c_2,\cdots,c_{l_2}),c_j(1\leqslant j\leqslant l_2)$ 为本体 O 中的概念,每个概念 c_j 具有一个权值 W_{aj}。定义概念文本属性向量如下：

$$\boldsymbol{a}_i=\sum_{j=1}^{l_2}W_{aj}\boldsymbol{c}_j \tag{2}$$

概念文本属性向量中的概念权值 W_{aj} 可以通过概念在数据中的出现频率、概念在本体 O 中的相对重要程度等信息自动确定，也可以在标注时由人工确定。

SemCN 方法的一个核心问题是要计算概念文本属性和概念查询的语义相似性,记为 $Sim_s(a_i,Q_c)$,通过 cosine 函数来计算,如下：

$$Sim_s(a_i,Q_c)=\frac{\boldsymbol{a}_i\cdot\boldsymbol{Q}_c}{\sqrt{\boldsymbol{a}_i\cdot\boldsymbol{a}_i}\sqrt{\boldsymbol{Q}_c\cdot\boldsymbol{Q}_c}}=\frac{\sum_{k=1}^{l_2}\sum_{j=1}^{l_1}W_{a_ik}W_{cj}\boldsymbol{c}_k\cdot\boldsymbol{c}_j}{\sqrt{\sum_{k=1}^{l_2}\sum_{j=1}^{l_2}W_{a_ik}W_{aj}\boldsymbol{c}_k\cdot\boldsymbol{c}_j}\sqrt{\sum_{k=1}^{l_1}\sum_{j=1}^{l_1}W_{ck}W_{cj}\boldsymbol{c}_k\cdot\boldsymbol{c}_j}} \tag{3}$$

则元组 r 与概念查询之间的语义相似性 $Sim_s(r,Q_c)$ 定义为：

$$Sim_s(r,Q_c)=\frac{\sum_{i=1}^{m}Sim_s(a_i,Q_c)}{m} \tag{4}$$

语义检索器使用公式(3)和(4)来计算具有文本属性的关系中每个元组与概念查询之间的语义相似性,只有那些相似性分数大于系统设定的阈值 ε_{ssim} 的元组才被选择出来,从而生成语义元组集 TS_s。ε_{ssim} 是通过实验确定的,是一个经验值,并且与关系数据库相关，不同的关系数据库数据可能需要设置不同的语义相似性阈值。

从公式(3)可以看出,需要计算 $\boldsymbol{c}_k\cdot\boldsymbol{c}_j$ 的值,在经典向量空间模型中,如果 c_k 和 c_j 的字符串不相同,就假设它们是正交的,其内积为零。但是,实际上,在本体 O 的层次结构上,c_k 和 c_j 其实是有关联的,是有语义相似性的,所以需要扩展经典的向量空间模型,以便可以计算本体 O 上任意两个概念之间的语义相似性。计算概念之间的语义相似性,有许多方法[8-12],这里主要利用本体的层次结构来计算本体中概念之间的语义相似性。而本体可能有许多种层次结构(如父子关系 subClassOf，部分关系 partOf，相关关系 relatedTo 等),其中父子关系(subClassOf)是在查询处理中最重要的层次结构[17]。先介绍几个相关定义(定义 4、5 来自[3]、6 和 7 来自[8])：

定义 4 层次(Hierarchy)记为 $H(S,\leqslant)$。假设 $(S,\leqslant)$ 是一个偏序集,$H(S,\leqslant)$ 是一个哈斯图(Hasse diagram),即一个有向无环图,其结点集为 S,并且有一个最小边集 E,当且仅当 $\exists(u\in S\wedge v\in S\wedge u\leqslant v)$,$u\rightarrow_p v$(表示在这个哈斯图中从 u 到 v 存在一条路径)。

定义 5 本体(Ontology)记为 $O(C,R,H)$,其中 C 是概念集$\{c_1,c_2,\cdots,c_t\}$,R 是本体上概念与概念之间的关系集合$(r_1,r_2,\cdots,r_l)$,H 是层次集合 $H(C,r_i)$ $(r_i\in R)$,每个层次 $H(C,r_i)$ 在 C 中有一个最抽象的概念称为根(Root)。

定义 6 概念深度(Concept Depth)记为 $depth_o^r(c)$。本体 $O(C,R,H)$ 上某个概念 c 在层次 $H(C,r)$ 上的深度定义为从该层次的根到概念 c 的路径上的边数。由于只考虑本体的父子关系(subClassOf)层次结构,所以本体表示 $O(C,R,H)$ 也可以简化为 $O(C)$,而 $depth_o^r(c)$ 也可以简化为 $depth_o(c)$。

定义 7 最近公共祖先(Lowest Common Ancestor)记为 $LCA_o(c_1,c_2)$。给定本体 $O(C)$ 上的两个概念 c_1 和 c_2,$LCA_o(c_1,c_2)=\{c'|depth_o(c')=\max(depth_o(c_1'),depth_o(c_2'),\cdots,depth_o(c_k'))$,$c_i'(1\leqslant i\leqslant k)$ 为 c_1 和 c_2

的公共祖先}。直观的解释就是,$LCA_o(c_1,c_2)$定义为c_1和c_2的公共祖先中具有最大深度的概念。LCA总是定义良好的(well defined),因为任意两个概念至少有一个公共祖先即根概念,并且在任意两个概念的所有公共祖先中,不可能有两个概念的深度相同。

因此,对于本体$O(C)$上的任意两个概念c_1和c_2可以定义其内积(Dot Product)[8]为:

$$\boldsymbol{c}_1\cdot\boldsymbol{c}_2=\frac{2\times depth_o(LCA_o(c_1,c_2))}{depth_o(c_1)+depth_o(c_2)} \tag{5}$$

例如:图3中的经济学领域本体示例,假设根概念为'经济危机',LCA_o('信用危机','货币危机')='金融危机'。$depth_o$('信用危机')=$depth_o$('货币危机')=2,$depth_o$('金融危机')=1,按照公式(5)可以计算出概念'信用危机'和'货币危机'的语义相似性为0.5。

因此通过这种方法就可以计算概念查询与关系数据库元组之间的语义相似性。

4.3 KSORD与SemSORD相结合

由于本体和关系数据库数据标注的不完善性,也可能导致某个关键词查询Q_k不能转换成概念查询Q_c;或者即使转换成概念查询,查询出来的语义结果也可能为空,或者比较少。需要把KSORD与SemSORD结合起来,以提供更好的检索。借鉴信息检索领域把关键词检索与基于本体的语义检索相结合的一些思想方法[14,15],来融合关系数据库上的关键词检索与语义检索技术。

按照图8(b)所示,元组集合并器(TS Merger)执行合并语义检索产生的元组集TS_s与关键词检索产生的元组集TS_k的任务。这里需要解决两个问题:如何合并TS_s和TS_k?如何融合TS_s元组语义相似性分数与TS_k元组关键词相似性分数?

首先介绍关键词检索的打分机制。如图8(b),元组集生成器利用关系数据库全文索引产生元组集TS_k,每个元组与关键词查询Q_k的关键词相似性分数$Sim_k(r,Q_k)$计算如下:

$$Sim_k(r,Q_k)=\frac{\sum_{i=1}^{m}Sim_k(a_i,Q_k)}{m} \tag{6}$$

其中$Sim_k(a_i,Q_k)$表示元组的文本属性a_i与关键词查询Q_k的关键词相似性分数,是由RDBMS的IR引擎全文索引检索得出的。根据不同的RDBMS,$Sim_k(a_i,Q_k)$取值范围可能不同,如Oracle产生的分数取值范围在0到100之间,而PostgreSQL产生的分数则在0到1之间。由公式(6)可知,$Sim_k(r,Q_k)$的取值范围与$Sim_k(a_i,Q_k)$相同。

然后分析元组集合并器如何合并TS_s和TS_k。对于每个具有文本属性的关系R,都可能产生TS_s与TS_k,因此元组集合并器需要对每个R分别执行合并任务以产生混合元组集TS_c,而$TS_c=\{r|(r\in TS_s\vee r\in TS_k)\wedge(Sim_c(r,Q)=composite(Sim_s(r,Q_c),Sim_k(r,Q_k)))\}$。

那么,如何计算$Sim_c(r,Q)$呢?由于语义相似性打分机制与关键词相似性打分机制的不同,这两种相似性分数的取值范围可能不同,语义相似性分数的取值范围在0到1之间,而关键词相似性分数的取值范围随RDBMS的不同而不同。因此采用最大最小值规范化方法来规范关键词性相似性分数(如公式(7))。对于Oracle数据库系统,$\mathrm{Min}(Sim_k(r,Q_k))$等于0,$\mathrm{Max}(Sim_k(r,Q_k))$等于100。

$$Sim_k(r,Q_k)=\frac{Sim_k(r,Q_k)-\mathrm{Min}(Sim_k(r,Q_k))}{\mathrm{Max}(Sim_k(r,Q_k))-Sim_k(r,Q_k)} \tag{7}$$

规范化$Sim_k(r,Q_k)$后,采用公式(8)来计算$Sim_c(r,Q)$[14]

$$Sim_c(r,Q)=t\times Sim_s(r,Q_c)+(1-t)\times Sim_k(r,Q_k) \tag{8}$$

其中t是一个实验经验值,可能对不同的数据集需要选用不同的设置值。在Si-SEEKER的实验中,采用ACM Computing Classification System(1998)作为领域本体,SigmodRecord部分数据作为测试数据,得出t如下:

$$t=\begin{cases}0.6 & \text{if } Sim_s(r,Q_c)\neq 0 \text{ and } Sim_k(r,Q_k)\neq 0\\ 1 & \text{else if } Sim_k(r,Q_k)=0\\ 0.3 & \text{else } Sim_s(r,Q_c)=0\end{cases} \tag{9}$$

4.4 SemCN方法小结

目前有许多方法实现KSORD[1]。ObjectRank[44]通过激励传播模型(Activation Spread Model)来实现关键词检索和结果排序,可以查询出不包含查询关键词的结果,实现一定程度上的语义检索能力。目前还没有直接利用本体来扩展KSORD以实现关系数据

库语义检索的研究，这里提出 SemCN 方法通过扩展 SEEKER,以实现关系数据库语义检索,也只是初步研究尝试。初步实验表明,Si-SEEKER 可以有效提高查询结果的查全率和查准率，但查询执行效率还有待进一步提高。因为概念查询和语义索引之间的语义相似性计算量非常大，这里采取一些预处理的方法,比如预先计算任意两个概念之间的 LCA,以及每个概念的深度,来提高查询处理的效率。未来还需要研究如何实现高效的语义索引，进一步提高关系数据库语义检索的查询效率。

4.5 SemCN 与其他 SemSORD 实现方法比较

表 1 从查询语言、数据模型以及查询结果质量和查询效率等 11 个方面对 SemCN 方法与第 3 章介绍的四种实现 SemSORD 方法进行综合比较。其中,查询扩展的“强”表示该方法主要是采用直接查询扩展方法实现的,“弱” 表示该方法并不主要是通过直接查询扩展实现的，比如 PARQ 方法主要是通过关系模型和关系代数的直接扩展,OntoSQL 是通过直接扩展 SQL 查询语言，这两种方法都是让用户具有更强的语义查询表达能力，从而实现数据库语义检索的目的;“中”表示该方法具有直接查询扩展功能,也具有其他的查询处理能力，比如 OntoRSM 在直接结果和推理结果为空的情况下，就会采用相似概念来直接扩展查询,以获取相关结果,而 SemCN 把用户直接转换成概念查询，然后计算概念查询与语义索引之间的语义相似性,这也实现了一定程度的查询扩展。

从表 1 可以看出,除了 PARQ 和 SemCN 两种方法对语义查询的查全率和查准率做了一些实验验证外，其他方法都没有很明确地讨论分析语义检索的查全率和查准率。实际上,SemSORD 肯定能够为用户查询找到更有意义的查询结果,但是,还没有专门针对关系数据库语义检索结果评价的测试基准。另外,对各种方法来说，如何提高语义检索的查询效率也是未解决的问题,需要进一步研究。

从表 1 也能看出,SemCN 不同于其他各种实现方法,仍然提供用户关键词查询的方式,可以适应各种字符类型的列,提供 Top-k 查询。从 KSORD 技术出发，进一步研究基于本体的关系数据库语义检索技术,具有很好的发展前景。当然,这种方法也有许多问题亟待研究解决,比如高效的语义索引建立、查询结果质量保证,以及查询效率等。

5 研究内容和发展趋势

第 3 章和第 4 章详细分析了目前实现 SemSORD 的各种方法,第 4.5 节比较了各种方法。基于这些研究,分析并提出 SemSORD 需要进一步加强研究的内

表 1 实现 SemSORD 各种方法的比较

Table 1 SemSORD approaches comparison

特点	OntoQE	PARQ	OntoSQL	OntoRSM	SemCN
研究者	德国 Humboldt-Universitätzu Berlin 大学 Chokri Ben Necib 等	美国 Maryland 大学 Octavian Udrea 等	Oracle 公司 Souripriya Das 等	IBM T.J Watson 研究中心 Anand Ranganathan 等	中国人民大学 张俊等
查询语言	SQL	SQL	SQL	SPARQL	Keyword Query
查询扩展	强	弱	弱	中	中
数据模型	没有变化	扩展关系模型和代数	增加了 ORC	增加 OWL 模型	建立语义索引
本体标注	不利用	不利用	不利用	利用	利用
适用情况	关键词列	关键词列	关键词列	任意字符列	任意字符列
推理功能	简单	简单	简单	很强	简单
Top-k 查询	无	无	无	有	有
结果排序	无	无	无	有	有
查全率	有效提高	有效提高	没有说明	没有说明	有效提高
查准率	没有说明	降低很多	没有说明	没有说明	降低较小
查询效率	没有说明	最多 478 个概念上测试,在 4.5 s 内可以获得查询结果。	最多 25 762 个概念上测试,效率与结果数成线性关系。	400 个概念上测试,效率可能不是很高	有待提高

容和趋势。这些研究内容和趋势并不是很全面,只是选择了几个重点方面进行阐述,力图起到抛砖引玉的作用。下面主要从基于KSORD技术的语义检索、语义相似性计算、性能优化、语义查询形成、结果排序分类和评价等方面介绍。

5.1 基于数据图的关系数据库语义检索

数据图是表示关系数据库数据的一个重要模型,基于数据图可以实现比较灵活高效的关系数据库关键词检索算法[45]。SemCN是基于模式图的关系数据库语义检索方法,并不能直接利用到基于数据图的KSORD系统上来,因此,还需要研究基于数据图的关系数据库语义检索方法,包括基于数据图的语义检索算法、语义相似性打分机制等。基于数据图的关系数据库语义检索方法可以与SemCN方法对比分析,从而进一步改进基于KSORD技术的关系数据库语义检索方法,提高语义检索结果的质量。

5.2 语义相似性计算

目前各种实现SemSORD方法都需要计算概念之间的语义相似性,以便进行查询扩展。尤其是SemCN方法,主要依赖概念查询和语义索引之间语义相似性计算。语义相似性计算也是本体学习、本体合并与集成的重要研究内容。目前基于本体的语义相似性研究比较活跃。文献[8]提出了基于本体层次结构计算语义相似性的多种方法,文献[10]提出了使用信息量(Information Content)计算概念之间语义相似性的方法,而文献[11]和[12]分别提出使用WordNet和基因本体这两种具体本体的概念语义相似性计算方法。文献[9]提出了不同本体中概念语义相似性计算方法,为本体集成提供了理论基础。文章指出本体集成是关系数据库语义检索的一个基础技术。如何选择、改进或者重新设计适合关系数据库语义检索特点的语义相似性计算方法,是值得研究的重要问题。

5.3 性能优化

表1表明,如何提高查询效率,仍是关系数据库语义检索亟待解决的一个关键问题,尤其是大规模本体支持下的关系数据库语义检索的查询效率问题。Oracle公司实现的OntoSQL方法预先计算并存储概念及概念之间的关系的传递闭包,并且实现基于本体的索引来提高查询效率[21]。IBM T.J.Watson研究中心的OntoRSM方法,还需要研究更好的索引结构,以支持Minerva推理系统高效的语义查询处理。SemCN方法计算概念之间的语义相似性需要消耗大量的计算时间。理论上,某个本体的任意两个概念都是相关的,都具有一定的语义相似性(而在关键词检索中一般假设关键词之间是相互独立),当本体规模较大,包含的概念比较多时,语义相似性计算耗费大量的运行时间。因此,如何在大规模本体支持下,建立高效的语义索引,快速计算概念之间的语义相似性,提供更强的推理能力,从而提高系统查询效率,是性能优化的重要研究问题。

5.4 语义查询形成

为了对关系数据库进行语义检索,SQL扩展方法[21]需要用户构建SQL查询,而语义映射方法[23]需要用户构建SPARQL语义查询,这两种查询方法都可以精确的表达用户的查询需求,但是也需要花很多时间去学习使用,一般只适于专业用户使用,而不适于大多数普通用户使用。SemCN方法提供关键词查询对关系数据库进行语义检索,简单的关键词查询虽然灵活方便、易于使用,但不能精确地表达用户查询需求,有时用户也不清楚应该用什么样的关键词来构造查询。本体不但可以用来支持用户查询扩展(Query expansion),也可以用来支持用户查询形成(Query formulation)过程[46]。文献[47]也研究了基于知识的生物数据库查询系统。用户如何利用本体来构造有效的关键词查询,甚至是个性化的关键词查询,还需要进行更多的研究。

5.5 语义查询结果排序和分类

在数据库系统中,查询结果的排序只是可选的因素,而在信息检索系统中,查询结果的排序处于核心地位[36],系统给每个结果赋予一个相关性分数,分数越大说明该结果和本次查询越相关,越排在前面。基于本体的数据库语义检索应该将结果按照相关性顺序展现给用户,考虑到语义检索的特点,结果排序面临一些新的挑战。例如,由于本体不完善以及数据库中数据的语义缺失等原因,有必要集成数据库上语义检索与关键词检索两种方式,这样在对结果打分时,既要考虑语义相似性分数,又要考虑关键词匹配相似性分数,需要设计合适的策略将二者融合起来。

与结果排序相关的一个问题是结果的分类,因为检索系统往往产生大量的结果,导致用户无法快

速找到自己所需要的结果。采用一定的分类或聚类方法,将结果组织成若干个不同语义的类别,能够帮助用户掌握结果的整体概要, 提高用户浏览结果的效率。文[48,49]针对 KSORD 结果提出一种按照结构和内容聚类的方法,而对于本体支持下的语义查询,如何挖掘更多的语义信息,从而改进分类的效果,很值得研究。

5.6 语义查询评价标准

信息检索领域很早就建立了多个参考文档集来评价各种信息检索系统的性能,例如 TREC 文档集[50],而 KSORD 系统目前还缺乏公认的评价标准和标准数据集, 在此基础上的语义查询同样缺乏统一的评价标准, 从而无法对不同的语义检索系统的性能做出有说服力的评价。构建语义查询的结果评价体系,包括查全率、查准率、执行时间和空间效率、支持并发访问的能力等评价指标, 是一个很有意义的基础性工作,也是一个开创性的工作。

6 结论

随着本体和语义 Web 技术的快速发展和应用,基于本体的关系数据库语义检索也越来越成为研究热点。利用本体来描述关系数据库语义,实现关系数据库语义检索方法, 可以有效提高关系数据库查询的查全率和查准率。文章分析了基于本体的关系数据库语义、语义检索的基础以及一般过程,然后重点分析和研究了目前单个关系数据库上实现基于本体的语义检索的几种主流方法, 如查询扩展、SQL 扩展、关系模型和关系代数扩展,语义映射以及基于关键词检索技术的语义检索方法。不但介绍了这些方法的基本原理,也分析了各种方法的优缺点,并且从多个角度对这些方法进行了综合比较。最后提出了该领域的研究挑战和技术发展趋势。在未来工作中,将从关系数据库关键词检索技术出发, 结合经济学领域示范语义平台的建设, 进一步研究和开发关系数据库语义检索技术。

References:

[1] Wang S,Zhang K.Searching databases with keywords[J]. Journal of Computer Science and Technology,2005,20(1): 55-62.

[2] Hyvönen E,Styrman A,Saarela S.Ontology-based image retrieval[C]//Towards the semantic web and web services. Proceedings of XML Finland 2002 Conference.Helsinki, Finland,2002:15-27.

[3] Hung E,Deng Y,Subrahmanian V S.TOSS:an extension of TAX with ontologies and similarity queries[C]//SIGMOD, 2004:719-730.

[4] Du X,LI M,Wang S.A survey on ontology learning research[J].Journal of Software,2006,17(9):1837-1847.

[5] Deng Z,Tang S,Zhang M,et al.Overview of ontology[J]. Journal(Natural Sciences) of Peking University(ASJNS), 2002,38(5):730-738.

[6] Guha R,McCool R,Miller E.Semantic search[C]//WWW2003 Proc of the 12th International Conference on World Wide Web,Budapest,Hungary,2003:700-709.

[7] Li Ding,Finin T,Joshi A,et al.Swoogle:a semantic Web search and metadata engine[C]//CIKM'04,2004

[8] Ganesan P,Garcia-Molina H,Widom J.Exploiting hierarchical domain structure to compute similarity[J].ACM Trans Inf Syst,2003,21(1):64-93.

[9] Rodríguez M A,Egenhofer M J.Determining semantic similarity among entity classes from different ontologies[J].IEEE Transactions on Knowledge and Data Engineering,2003, 15(2):442-456.

[10] Resnik P.Using information content to evaluate semantic similarity in a taxonomy[C]//International Joint Conference for Artificial Intelligence(IJCAI),1995:448-453.

[11] Yang Dong-qiang,Powers D M W.Measuring semantic similarity in the taxonomy of WordNet[C]//Proceedings of the Twenty-eighth Australasian Conference on Computer Science,Newcastle,Australia,2005,38:315-322.

[12] Couto F M,Silva M J,Coutinho P.Semantic similarity over the gene ontology:family correlation and selecting disjunctive ancestors[C]//CIKM,2005:343-344.

[13] Andreasen T,Bulskov H,Knappe R.On ontology-based querying[C]//18th International Joint Conference on Artificial Intelligence,Ontologies and Distributed Systems(IJCAI), 2003:53-59.

[14] Vallet D,Fernández M,Castells P.An ontology-based information retrieval model[C]//ESWC,2005:455-470.

[15] Varga P,Mészáros T,Dezsényi C,et al.An ontology-based information retrieval system.IEA/AIE,2003:359-368.

[16] Mena E,Illarramendi A,Kashyap V,et al.Observer:an approach for query processing in global information sys-

tems based on interoperation across pre-existing ontologies[J].International Journal on Distributed and Parallel Databases(DAPD),2000,8(2):223-272.

[17] Kohler J,Philippi S,Lange M.SEMEDA:ontology based semantic integration of biological databases[J].Bioninformatics,2003,19(18):2420-2427.

[18] Liu F,Yu C,Meng W,et al.Effective keyword search in relational databases[C]//SIGMOD'06,2006.

[19] Bonatti P A,Deng Y,Subrahmanian V.An ontology-extended relational algebra[C]//Proceedings of the IEEE International Conference on Information Reuse and Integration(IEEE IRI),2003:192-199.

[20] Udrea O,Deng Y,Hung E,et al.Probabilistic ontologies and relational databases[C]//the Proceedings of the OTM Confederated International Conferences(CoopIS,DOA,and ODBASE)2005,Agia Napa,Cyprus,2005.

[21] Das S,Chong E I,Eadon G,et al.Supporting ontology-based semantic matching in RDBMS[C]//VLDB,2004:1054-1065.

[22] Prudhommeaux E,Seaborne A.SPARQL query language for RDF.Working draft,W3C,2005.

[23] Ranganathan A,Liu Z.Information retrieval from relational databases using semantic queries[C]//Conference on Information and Knowledge Management(CIKM'06),Arlington,Virginia,USA,2006:820-821.

[24] Necib C B,Freytag J C.Ontology based query processing in database management systems.CoopIS/DOA/ODBASE,2003:839-857.

[25] Wen J,Wang S.SEEKER:keyword-based information retrieval over relational data-bases[J].Journal of Software,2005,16(7):1270-1281.

[26] Li M,Du X Y,Wang S.Maintaining materialized relations incrementally to improve performance of ontology query[C]//Proc of International Workshop on XML,Web,and Internet Contents Technologies(XWICT),2006:90-96.

[27] Li M,Du X Y,Wang S.Selection of materialized relations in ontology repository management system[C]//Lecture Notes in Artificial Intelligence 4092(KSEM2006),2006:241-251.

[28] Hu He,Du Xiao-yong.An ontology learning model in grid information services[C]//Proceedings of 1st International Conference on Innovation Computing,Information and Control Beijing,China,.IEEE Press,2006,3:398-401.

[29] Hu He,Du Xiao-yong.ConAnnotator:ontology-aided collaborative annotation system[C]//Proc of the 10th International Conference on CSCW in Design(CSCWD 06) Nanjing,China.IEEE Press,2006:850-855.

[30] Du Xiao-yong,Hu He,Li Man,et al.An economic semantic Web platform[C]//Proc of the 5th International Conference on Grid and Cooperative Computing(GCC2006)Workshops Changsha,China,IEEE Press,2006.

[31] Zhang J,Peng Z,Wang S,et al.Si-SEEKER:ontology-based semantic search over databases[C]//The First International Conference on Knowledge Science,Engineering and Management(KSEM 2006).Guilin,China,2006:599-611.

[32] Shah G,Mahmood T S.Searching databases for sematically-related schemas[C]//SIGIR'04,Sheffield,South Yorkshire,UK,2004:504-505.

[33] Guarino N.Semantic matching:formal ontological distinctions for information organization,extraction,and integration[M]//Pazienza M T.Information extraction:a multidisciplinary approach to an emerging information technology.[S.l.]:Springer Verlag,1997:139-170.

[34] óscar Corcho.Ontology based document annotation:trends and open research problems[J].International Journal of Metadata,Semantics and Ontologies(IJMSO),2006,1(1):47-57.

[35] Kang B.A novel approach to semantic indexing based on concept[C]//Proceedings of the 41st Annual Meeting on Association for Computational Linguistics,2003:44-49.

[36] Baeza-Yates R,Ribeiro-Neto B.Modern Information Retrieval[M].[S.l.]:ACM Press,1999.

[37] Wollersheim D,Rahayu W.Ontology based query expansion framework for use in medical information systems[J].International Journal of Web Information Systems,2005,1(2):101-115.

[38] Fiaidhi J,Mohammed S,Jaam J,et al.A standard framework for personalization via ontology-based query expansion[J].Pakistan Journal of Information and Technology,2003,2(2):96-103.

[39] Royo J A,Mena E,Bernad J,et al.Searching the Web:from keywords to semantic queries[C]//Third International Conference on Information Technology and Applications(ICITA 2005),Sydney,Australia,2005:244-249.

[40] Chong E I,Das S,Eadon G,et al.Supporting keyword columns with ontology-based referential constraints in DBMS[C]//Proceedings of the 22nd International Conference on Data Engineering(ICDE'06),2006.

[41] Hristidis V,Gravano L,Papakonstantinou Y.Efficient IR-

Style Keyword Search over Relational Databases[C]//VLDB, 2003:850-861.

[42] Agrawal S,Chaudhuri S,Das G.DBXplorer:a system for keyword search over relational databases[C]//ICDE,2002:5-16.

[43] Bennett N,He Q,Chang C,et al.Concept extraction in the interspace prototype[R].Dept of Computer Science,University of Illinois at Urbana-Champaign,1999.

[44] Balmin A,Hristidis V,Papakonstantinou Y.ObjectRank:authority-based keyword search in databases[C]//VLDB,2004:564-575.

[45] Kacholia V,Pandit S,Chakrabarti S,et al.Sudarshan,Rushi Desai,Hrishikesh Karambelkar:bidirectional expansion for keyword search on graph databases[C]//VLDB,2005:505-516.

[46] García E,Miguel-Ángel Sicilia.User interface tactics in ontology-based information seeking[J].Psychnology,e-journal,2003,1(3):243-256.

[47] Bresciani P,Fontana P.A knowledge-based query system for biological databases[C]//Proceedings of the 5th International Conference on Flexible Query Answering Systems,2002:86-99.

[48] Peng Z,Zhang J,Wang S,et al.TreeCluster:clustering results of keyword search over databases[C]//The 7th International Conference on Web-Age Information Management (WAIM 2006),Hong Kong,China,2006:385-396.

[49] Wang S,Peng Z,Zhang J,et al.NUITS:a novel user interface for efficient keyword search over databases[C]//The 32nd International Conference on Very Large Data Bases (VLDB 2006),Seoul,Korea,2006:1143-1146.

[50] Voorhees E M,Harman D K.Overview of the 6th text retrieval conference(TREC-6)[C]//Proceedings of the 6th Text REtrieval Conference,1997.

附中文参考文献:

[4] 杜小勇,李曼,王珊.本体学习研究综述[J].软件学报,2006,17(9):1837-1847.

[5] 邓志鸿,唐世渭,张铭,等.Ontology 研究综述[J].北京大学学报:自然科学版,2002,38(5):730-738.

[25] 文继军,王珊.SEEKER:基于关键词的关系数据库信息检索[J].软件学报,2005,16(7):1270-1281.

王珊(1944-),女,江苏无锡人,教授,博士生导师,1968 年北京大学物理学毕业,1981 年于中国人民大学获计算机科学专业工学硕士学位,目前是中国人民大学信息学院教授、博士生导师,主要研究领域为高性能数据库、知识工程、数据仓库和网格数据管理。

Wang Shan was born in 1944.She graduated from Peking University in 1968 and received M.S.degree in Computer Science from Renmin University of China in 1981.She is currently a full professor and Ph.D supervisor in School of Information,RUC.Her current research interests include high performance databases and knowledge systems,data warehousing technology,and grid data management.She has extensive publications in these areas.

张俊(1971-),男,湖北崇阳人,博士生,1993 年于吉林工业大学获计算机软件专业工学学士学位,1996 于大连理工大学获计算机应用技术专业工学硕士学位,目前为中国人民大学计算机应用技术专业博士生,同时也是大连海事大学计算机学院副教授,主要研究领域为数据库和信息检索、数据库和知识库系统。

Zhang Jun was born in 1971.He received his B.S.degree from Jilin University of Technology in 1993,M.S.degree from Dalian University of Technology in 1996,and now is a Ph.D candidate at Renmin University of China,all in Computer Science.He has also been with the computer science and technology college at Dalian Maritime University since 1996,where he is now an associate professor.His research focuses on database and information retrieval,database and knowledge system.

彭朝晖(1978-),男,山东聊城人,博士生,1999 年和 2003 年于山东大学分别获得计算机专业工学学士学位和工学硕士学位,目前是中国人民大学计算机应用技术专业博士生,主要研究领域为数据库和信息检索、数据库和知识库系统。

Peng Zhao-hui was born in 1978.He received his B.S.degree in 1999,M.S.degree from Shandong University in 2003,and now is a Ph.D candidate at Renmin University of China,all in Computer Science.His research focuses on database and information retrieval,database and knowledge system.

战疆(1970-),男,山东济南人,博士生,1992 年于山东大学获计算机及应用专业理学学士学位,2000 年于首都经贸大学获管理信息系统专业经济学硕士学位,目前为中国人民大学计算机应用技术专业博士生,主要研究领域为数据库和信息检索、数据库和知识库系统。

Zhan Jiang was born in 1970.He received his B.S.degree in Computer Science from Shandong University in 1992,M.S.degree in Information Management System from Capital University of Economics and Business in 2000,and now is a Ph.D candidate in Computer Science at Renmin University of China.His research focuses on database and information retrieval,database and knowledge system.

杜小勇(1963-),男,浙江开化人,博士,教授,博士生导师,1983 年于杭州大学获得学士学位,1988 于中国人民大学获得硕士学位,1997 年于日本名古屋工业大学获得博士学位,1997 年至 1999 年曾是名古屋工业大学助理教授,目前是中国人民大学教授,博士生导师,CCF 高级会员,主要研究领域为智能信息检索,高性能数据库,知识工程。

DU Xiao-yong was born in 1963.He received his B.S.degree from Hangzhou University in 1983,M.S.degree from Renmin University of China in 1988,Ph.D degree from Nagoya Institute of Technology,Japan,in 1997. He was an assistant professor in Nagoya Institute of Technology from 1997 to 1999.He is currently a professor and Ph.D supervisor in school of Information,Renmin University of China.He is a member of the CCF. His current research interests include high performance databases,intelligent information retrieval,and semantic web.