一种基于语义的RDF近似查询方法

周文健 马睿哲

(东北大学信息科学与工程学院 沈阳 110819)

摘 要 针对返回结果为空或甚少的情况,提出RDF查询松弛和同源词替换相结合的方法:通过RDFS蕴含规则对初始查询进行松弛,选取合适的松弛查询进行同源词替换得到更多的查询结果。为了返回与初始查询在语义上相近的结果,提出面向RDF的语义距离概念,即通过语义距离的计算选取与初始查询在语义上相近的结果。在上述查询策略的基础上,给出基于语义的RDF近似查询处理的算法,通过实验验证了所提方法的可行性,并与现有的RDF查询方法进行了比较。实验结果表明,所提方法在查准率以及查全率方面均具有一定的优越性。

关键词 语义Web,RDF,近似查询,松弛,语义距离

中图法分类号 TP391 **文献标识码** A

RDF Approximate Query Approach Based on Semantics

ZHOU Wen-jian MA Rui-zhe

(College of Information Science and Engineering, Northeastern University, Shenyang 110819, China)

Abstract To handle the problem of empty or few answers returned from RDF in response to a user query and the problem of synonyms in RDF queries, an approximate query approach by combining query relaxation and replacement with kinship words was proposed. RDF entailment to triple patterns is applied on the original query to relax it and then appropriate queries are chosen for replacement of kinship words so that more results can be obtained. The notion of semantic distance was introduced. The results that are semantically close to the original query can be determined. An approxiamate query algorithm based on semantics was hereby developed. The proposed approach was verified with experiments and it has good performances in its recall and precision.

Keywords Semantic Web, RDF, Approximate query, Relaxation, Semantic distance

1 引言

作为语义Web[1]的重要组成部分,RDF(Resource Description Framework)[2]是一个通用的元数据模型标准。近年来,RDF被广泛应用于Web应用领域,大规模RDF查询已经成为RDF管理的重要内容[3]。对于RDF的查询,一方面,随着RDF规模和复杂性的增加,要求普通用户了解其结构和内容已不现实,此时即使用户查询意图明确,仍有可能获得过少甚至空查询结果;另一方面,不同的用户对同一事物可能有不同的描述,如一义多词(同义词)以及外文词形变化等。RDF本体的近似查询是解决上述问题的一种有效方法。文献[4]提出一种逻辑松弛方法,该方法通过RDFS蕴含产生更普遍的查询来查询潜在的相关结果,但该文只给出了相应的框架,没有给出具体的实现,也没有考虑一义多词的问题。文献[7]在处理RDF查询时,提出了一种基于相似度的查询松弛方法,但相似度的计算只考虑了本体层次结构的影响,没有考虑RDF本体数据的影响,也没有考虑一义多词的问题。

针对RDF查询返回结果为空或少量以及一义多词的情况,本文提出查询松弛和同源词替换相结合的方法,即通过RDFS蕴含规则对初始查询进行松弛,选取合适的松弛查询进行同源词替换,以得到更多的查询结果。为了返回与初始查询在语义上相近的结果,提出语义距离的概念,即通过语义距离的计算选取与初始查询在语义上相近的结果。在此基础上给出基于语义的RDF近似查询算法,通过实验验证了所提方法的可行性,并与现有的RDF查询方法进行了比较。

2 基本概念

RDF是用于描述Web资源的通用框架。在RDF模型中,一个含有属性和相应属性值的事物通过三元组进行描述,三元组包括主体(Subject)、谓词(Predicate)和客体(Object)。主体为所要描述的资源,谓词为资源的属性,客体为该属性所对应的值。一个RDF三元组表示为:$t(s,p,o)\in(I\cup B)\times I\times(I\cup B\cup L)$,其中$I$是一个IRIs(国际化资源标志符)集,$B$是一个空白节点集,$L$是文字集,$s$称为主体,$p$称为谓词,$o$称为客体。RDF三元组集合构成了RDF本体。

RDF模型是三元组的集合,其中每个三元组都可以用节点-边-节点的连接来表示,一系列三元组构成了RDF图。在RDF图中,节点是主体和客体,边的方向由主体指向客体。RDF三元组通过谓词说明了事物间的某种联系,RDF图的含义由图中众多三元组共同进行说明。

RDF图模式可定义为$G=(t_1,t_2,\cdots,t_i,\cdots,t_m)$,其中$t_i\in T$,$T$是三元组模式集合。基于RDF图模式,一个用户查询

到稿日期:2012-11-26 返修日期:2013-02-13

周文健 男,硕士生,主要研究方向为本体与语义Web;马睿哲 女,主要研究方向为智能数据处理,E-mail:ruizhema.neu@gmail.com。

定义为 $Q=\{q_1,q_2,\cdots,q_i,\cdots,q_n\}$，其中 $q_i\in T$。

RDFS(RDF Schema)是对 RDF 的一种补充，它定义了类和属性，可以用这些类和属性来描述其它的类和属性，从而增强 RDF 对资源的描述能力。

3 RDF 近似查询策略与方法

RDF 近似查询的基本思想是，对于 RDF 上的一个初始查询 Q，首先根据 RDFS 蕴含规则，得到松弛查询集合 $W(Q)=\{Q_1,Q_2,\cdots Q_k,\cdots,Q_t\}$。其中，$Q_k=\{q_{k1},q_{k2},\cdots,q_{kn},\cdots,q_{kn}\}$ 为 Q 的一个松弛查询，Q 中的三元组 q_m 经松弛后得到三元组集合 $W(q_m)=\{q_{1m},q_{2m},\cdots,q_{km},\cdots,q_{tm}\}$。对于 $W(Q)$ 各个松弛查询中属性值为原子值的三元组集合，通过 WordNet 将其初始的原子属性值替换为相应的同源词，生成替换查询集合 $W(Q_1)$。与 Q 的相似程度越高的 $W(Q)$，其对语义距离的贡献越小，属性权重也越小，该查询应优先进行同源词替换。为此需要使用语义距离评估 Q 与 $W(Q_1)$ 中相应各个查询之间的语义距离，将与 Q 语义距离小的查询优先纳入查询范围。最后，通过查询重写实现对 Q 的近似处理，合取所有松弛基本查询条件，最终形成松弛查询 Q。

定义 1(松弛三元组模型) 对于给定的三元组模型 t，通过应用简单松弛和本体松弛规则中的一种或多种得到 t'，就称 t' 为 t 的一个松弛模型，记为 $t<t'$。

定义 2(松弛查询) 对于一个给定的查询 $Q(t_1,t_2,\cdots,t_n)$，如果至少存在一个序列 (t_i,t_i') 满足 $t_i<t_i'$，则称 $Q'(t_1',t_2',\cdots,t_n')$ 为 Q 的松弛[4]，记为 $Q<Q'$，称 Q' 为松弛查询。

定义 3(替换查询) 对于一个给定的查询 $Q(t_1,t_2,\cdots,t_n)$，其经过同源词替换得到 $Q'(t_1',t_2',\cdots,t_n')$，则称 $Q'(t_1',t_2',\cdots,t_n')$ 为 Q 的替换查询。

3.1 基于 RDFS 的 RDF 查询松弛方法

利用 RDFS 蕴含规则，可实现初始查询的松弛。文献[4]提出两种类型的查询松弛模型：三元组模型上的简单松弛和三元组模型上的本体松弛。

(1)三元组模型上的简单松弛

两个三元组模型 $t_1(a,b,c)$ 和 $t_2(d,e,f)$，若存在一个从 t_1 的元素到 t_2 的元素的函数 u(元素包括 $IRIs$、文字和保留变量)，使得 $(u(a),u(b),u(c))=(d,e,f)$，则 $t_1(a,b,c)$ 与 $t_2(d,e,f)$ 存在映射关系，即 $u:t_1(a,b,c)\rightarrow t_2(d,e,f)$。当 t_1 到 t_2 的映射和 t_2 到 t_1 的映射同时存在时，称这两个三元组同构。形式上，如果存在一个从 t_1 到 t_2 的映射，则将其表示为 $t_1\prec t_2$，并把 t_2 称为 t_1 的简单松弛。

(2)三元组模型上的本体松弛

两个 RDF 图 G_1 和 G_2，如果可以用图 1 两组规则(A)和(B)中的任一规则，则可以由 G_1 得到 G_2，记作 $G_1\Rightarrow_{RDFs}G_2$。图 1 中，sc 和 sp 分别表示 rdfs:subClassOf 和 rdfs:subPropertyOf。

Group A(Subproperty)

(1) $\dfrac{(a,sp,b)(b,sp,c)}{(a,sp,c)}$　　(2) $\dfrac{(a,sp,b)(x,a,y)}{(x,b,y)}$

Group B(Subproperty)

(3) $\dfrac{(a,sc,b)(b,sc,c)}{(a,sc,c)}$　　(4) $\dfrac{(a,sc,b)(x,type,a)}{(x,type,b)}$

图 1 RDFS 推理规则

设 onto 为 RDF 本体，closure(onto)为 onto 的闭包，设 t_1 和 t_2 是两个三元组模型，并且 $t_1\notin$ closure(onto)，$t_2\notin$ closure(onto)。当 $t_1\cup$onto$\Rightarrow_{RDFs}t_2$ 时，称 t_2 是 t_1 的一个本体松弛，记为 $t_1\lhd t_2$。本体松弛包含以下的类型条件和属性松弛：

(1)如果 $(b,sc,c)\in$ closure(onto)成立，三元组模型 $(a,type,b)$ 可松弛为 $(a,type,c)$。

(2)如果 $(p_1,sp,p)\in$ closure(onto)成立，三元组模型 (a,p,b) 可松弛为 (a,p_1,b)。

3.2 松弛查询的选取方法

根据松弛查询模型相对于初始查询模型的权重对松弛查询进行排序，优先执行权重小的松弛查询。一个松弛查询相对于初始查询的权重由它们之间的语义相似度所决定。语义相似度的计算主要考虑 RDFS 所体现的本体层次结构，其值由 RDF 图上节点之间的语义相似度以及 RDF 图上弧之间的语义相似度两个因素所决定[8,9]。这里 RDF 本体包含有类的集合 IC 及属性集 IP。

(1)RDF 图节点之间的语义相似度

对于 RDF 图上的两个节点 c_1 和 c_2，它们的最小公共祖先(记作 LCA)表示的是与节点 c_1 和 c_2 距离最近的共同父类节点。若有一节点是 c_1 和 c_2 的 LCA，则该节点为 c_1 和 c_2 的父类，且该节点在 c_1 和 c_2 的所有父类节点中与 c_1 和 c_2 节点距离最近。距离相等的两个概念的语义相似度随着它们所在的层次深度总和的增加而增加，随着它们之间层次差的增加而减少。考虑上述因素的影响，RDF 图上两个节点之间的语义相似度定义为：

$$\mathrm{sim}(c_1,c_2)=\frac{2\times depth(LCA(c_1,c_2))}{depth(c_1)+depth(c_2)}$$

其中，$depth(c)$ 表示 RDF 图中节点 c 的深度。

(2)RDF 图上弧之间的语义相似度

设三元组中属性 $P_1\in$ IP 经本体松弛得到属性 $P_2\in$ IP，则 P_2 是 P_1 的超属性，P_1 是 P_2 的子属性。RDF 图上的弧通过 RDFS 蕴含规则进行松弛，两个弧 p_1 和 p_2 之间的语义相似度定义为：

$$\mathrm{sim}(p_1,p_2)=\frac{2\times depth(LCA(p_1,p_2))}{depth(p_1)+depth(p_2)}$$

$LCA(p_1,p_2)$ 是 p_1 和 p_2 的最小公共祖先，是指与 p_1 和 p_2 距离最近的且为 p_1 和 p_2 的超属性的弧，即该弧表示的是 p_1 和 p_2 的超属性，且该弧在 p_1 和 p_2 的所有超属性弧中与 p_1 和 p_2 距离最近。

给定一个三元组 $q(s,p,o)$ 和它的一个松弛三元组 $q'(s',p',o')$，它们之间的相似度定义为：

$$\mathrm{sim}(q,q')=\mathrm{sim}(s,s')+\mathrm{sim}(p,p')+\mathrm{sim}(o,o')$$

设 $Q(q_1,q_2,\cdots,q_i,\cdots,q_n)$ 是 RDF 上的一个初始查询，其中 $q_i\in T$，T 是三元组模式集。设 $Q'(q_1',q_2',\cdots,q_i',\cdots,q_n')$ 为 Q 经过 RDFS 蕴含规则松弛得到的一个松弛查询，其中 $q_i'\in T$ 且 q_i' 为三元组 q_i 松弛后的三元组。Q 和 Q' 之间的语义相似度定义为：

$$\mathrm{sim}(Q,Q')=\prod_{i=1}^{n}\mathrm{sim}(q_i,q_i')$$

Q' 相对于 Q 的权重定义为：

$$weight(Q',Q)=\frac{1-\mathrm{sim}(Q',Q)}{1+\mathrm{sim}(Q',Q)}$$

可以看出，$\mathrm{sim}(Q,Q')$ 值越大，$weight(Q',Q)$ 就越小。设 Q_1 和 Q_2 是 Q 的两个松弛查询，若 $\mathrm{sim}(Q,Q_1)>\mathrm{sim}(Q,Q_2)$，则 $weight(Q_1,Q)<weight(Q_2,Q)$，$Q_1$ 对语义距离的贡献要小于 Q_2。对于一个初始查询经过 RDFS 蕴含规则松弛得到的全部松弛查询，根据其权重对它们进行排序，之后依次选取最小的权重进行下一步操作。

3.3 同源词替换方法

自然语言中的一个词通常存在同义词或语义上相似的词，这些词称为同源词。在进行 RDF 查询时如果考虑进行同源词替换，则可以更有效地解决空查询或少量查询结果的问题。同源词可通过 WordNet 获取。给定一个词 w，通过 WordNet 可以得与 w 语义相关的 5 类词：w 本身、w 的变形、w 的同义词、w 的邻接下位词和邻接上位词，这 5 类词被称作 w 的同源词。设 $t=(tw_1,tw_2,\cdots,tw_n)$ 是 RDF 本体三元组上属性的一个取值，对于任意一个单词 $tw_i\in t(1\leqslant i\leqslant n)$，通过 WordNet 可获得其同源词集合，表示为

$$E(t)=\begin{Bmatrix} tw_{11},tw_{12},\cdots,tw_{1p_1} \\ tw_{21},tw_{22},\cdots,tw_{2p_2} \\ \cdots\cdots \\ tw_{n1},tw_{n2},\cdots,tw_{np_n} \end{Bmatrix}$$

定义 f 为同源词 tw_{ij} 的频率，即 tw_{ij} 在 $E(t)$ 中出现的次数。t 的同源词集合可以表示为 $K(t)=\bigcup_{i=1,2,\cdots,n}K(tw_i)$，如果 $w\in K(t)$，则称 w 是 t 的同源词。

为防止因大量同源词存在产生大量的替换查询而影响系统的效率，采取以下的处理过程：

(1)去除 t 中不能被 WordNet 所识别的符号或字符串组，得到一个集合 $t_0=\{tw_1,tw_2,\cdots,tw_n\}$。

(2)对 t_0 中的每一个词 tw_i，使用 WordNet 的 API 函数获得其同源词集合 $k(tw_i)$，t_0 的同源词集合为 $K(t_0)=\bigcup_{i=1,2,\cdots,n}K(tw_i)$。

(3)对于每一个属性值是字符串的三元组，由前面两步得到该三元组字符串属性值的同源词集合，进而得到整个 RDF 本体的同源词集合 $\tau(R)=\bigcup_{i=1,2,\cdots,n}K(t_i)$($m$ 为属性值为字符串的三元组数目)。

(4)对 $\tau(R)$ 中的所有单词建立索引结构，包括：一个哈希表(hash table)、一个队列(wn-list)以及 m 个队列(db-list)，如图 2 所示。哈希表中 hash 函数把字符串变为一个 bucket，每个 bucket 中包含一个指针，指向 wn-list 中的一个节点。wn-list 用来储存 $\tau(R)$ 中的同源词，每个节点包含 $\tau(R)$ 的一个单词。每个节点指向其相应的 db-list。db-list 中的每个节点表示一个同源词，对于 wn-list 中的任意一个单词 w_i，如果 w_i 在 $K(t_s)$ 中出现，那么 w_i 就是字符串 t_s 的同源词，就将 t_s 加入到相应的 db-list 中。

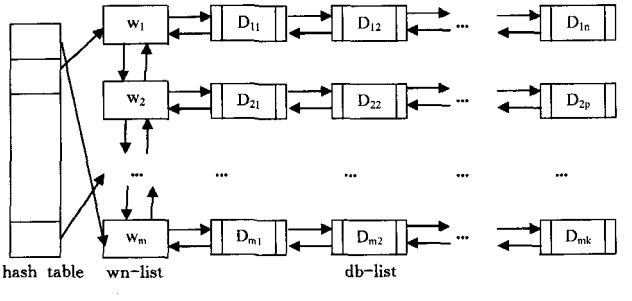

图 2 索引结构

3.4 查询的语义距离

经松弛和同源词替换后，RDF 上的初始查询 Q 可能会产生大量的替换查询，它们需要按照其 Q 的满足程度进行排序。借鉴文献[12]中单词语义距离的计算方法，将其用于 RDF 语义距离的计算。给定 RDF 上的一个查询 $q=(q_1,q_2,q_3,\cdots,q_i,\cdots,q_{k-1},q_k)$，$q_i\in T$，$T$ 是个三元组模式集，q_i 中含有 z_i 个词，$qw_i=(q_{i1},q_{i2},q_{i3},\cdots,q_{in})$ 是 q 上一个三元组 q_i 属性上的值，包含 n 个词。$t(t_1,t_2,t_3,\cdots,t_i,\cdots,t_m)$ 是 RDF 本体上一个三元组的属性值，包含 m 个词。定义 4 种类型的语义距离：

(1)$d(w_1,w_2)$，w_1 和 w_2 是两个词，当 $w_2=w_1$，$d(w_1,w_2)=d_0$；当 w_2 是 w_1 的变形，$d(w_1,w_2)=d_1$；当 w_2 是 w_1 的同义词，$d(w_1,w_2)=d_2$；当 w_2 是 w_1 的邻接下位词，$d(w_1,w_2)=d_3$；当 w_2 是 w_1 的邻接上位词，$d(w_1,w_2)=d_4$；否则，$d(w_1,w_2)=d_5$。其中，$d_0<d_1<d_2<d_3<d_4<1\leqslant d_5$，$d_i(i=0,1,2,\cdots,5)$的值可以基于训练获取。

(2)$d(w,t)$，w 是 t 的一个同源词，$w\in K(t)$，且有 $d(w,t)=\alpha(n)\min(d(w,tw_i),w\in K(tw_i))$，其中，$\alpha(n)$是一个单调递增函数，用来调整 $d(w,t)$，$d(w,t)$会随着 n 的增大而增大。对于任一个 $n<L(L$ 是一个大值常量)，满足 $1\leqslant\alpha(n)\leqslant 2$，并且 $d(w,t)$也介于 0 和 1 之间。

(3)$d(q,t)$，表示 q 和 t 之间的语义距离，qw_i 与 t 之间的语义距离定义为：$d(qw_i,t)=\prod_{q_{is}\in K(t)}(\varphi(f)d(q_{is},t))$，其中，$q_{is}$ 是 t 的一个同源词，f 是 q_{is} 在 $E(t)$ 中出现的频率，$\varphi(f)$是一个单调递减函数，用来调节 $d(qw_i,t)$的值，且满足 $0<\varphi(f)\leqslant 1$。在本文中，$\varphi(f)=1/\varphi(f)=1/\lambda(f)$，其中 $\lambda(f)=1+\log_{10}(1+\log_{10}(f))$。

(4)$d(q,T)$，其中 qw 与 t 之间的语义距离为：$d(qw,t)=\min\{d(qw_i,t)\mid i=1,2,\cdots,k\}$，若 q_i 的属性值不为原子值(字符串)，且其经过 RDFS 蕴含规则松弛后得到的三元组 q_{i1}，则 q_i 和 q_{i1} 之间的语义距离定义为：$d(q_i,q_{i1})=weight(q_i,q_{i1})$。若 q_i 的属性值为原子值(字符串)v_i，且其经过 RDFS 蕴含规则松弛后得到的三元组 q_{i1}，q_i 的属性值用 v_{i1} 来替换 v_i，则 q_i 和 q_{i1} 这两个三元组属性值之间的语义距离定义为：$d(q_i,q_{i1})=d(v_i,v_{i1})\times weight(q_i,q_{i1})$。

依照上述语义距离的定义，初始查询 Q 和其经过松弛和替换后的查询 Q' 之间的语义距离为：

$$D(Q,Q')=\prod_{q_i\in Q}d(q_i,q_{i1})$$

可以看出，Q 和 Q' 之间的语义距离取决于它们相对应的三元组模型之间的语义距离，三元组模型之间的语义距离越小，则 Q 和 Q' 之间的语义距离也越小。

4 查询松弛算法

对于初始查询 Q 首先给定一个阈值，限定替换查询与初始查询之间的语义距离，之后对 Q 进行松弛并选择权重最小的松弛查询，并对该查询进行同源词替换。如果替换查询与初始查询的语义距离在阈值之内，则认为该替换查询符合用户查询要求，可将其放入查询结果集。此时，若查询结果集的结果数达到 K 个，则中止查询，否则继续从松弛查询中选择权重第二小的松弛查询进行替换，直至查询结果集中的结果数达到 K 个为止。具体的算法实现如下所示：

SDRQ 算法

输入：Q($t_1(0),t_2(0),\cdots,t_n(0)$)，Smin，K′；

输出：结果集 Result；

1. 初始化 Result、Pset、Qset、Rset 和 K；
2. 松弛初始查询 Q，得到松弛查询集合 Q′；
3. Add(Q′，Qset)；
4. While(|Qset|>0)do
5. Qc=SelectMinWeight(Qset)；
6. 通过 wordnet 或是 w-index 得到 Qc 中将要进行替换的字符串 m 的同源词集合 K(m)；

```
7.  对该查询进行同源词替换,得到替换查询集合 Qti;
8.  Add(Qti,Pset);
9.  Delete(Qc,QSet);
10.  If(K< K' and |Pset|>0)
11.    计算 Pset 中的查询与初始查询的语义距离;
12.    将与初始查询之间的语义距离小于给定的阈值 Smin的替换查
       询加入 RSet 中;
13.    While(|Rset|>0 and K< K')do
14.      Qt=SelectMinDistance(Q)
15.      执行查询 Qt,并将查询结果存入结果集 Result 中;
16.      K=K+1;
17.      Delete(Qt,RSet);
18.    END While
19. END While
20. Return Result
```

其中 $Q(t_1(0),t_2(0),\cdots,t_n(0))$ 为初始查询,Result 为查询结果集,QSet 为初始查询松弛后的查询集,PSet 为松弛查询被替换后的查询集,Rset 为与初始查询之间的语义距离小于给定阈值的替换查询集,K' 表示所设定的结果数限定值,K 表示所查询到的结果数。SDRQ 算法的时间复杂度为 $O(kp)$,其中 k 为选取的松弛模型个数,p 为待替换的同源词个数。SDRQ 算法的最大时间复杂度为 $O(pm^{2n}|G|^n)$,其中 G 是 RDF 图。

5 实验结果分析

本文使用 Java 基于 Eclipse3.2 工具开发了一个原型系统。实验数据采用目前被广泛采用的本体基准 Lehigh University Benchmark(LUBM)[10],用 LUBM 数据生成器 UBA 生成包含 6000k 个三元组的本体实例。数据存储管理及算法实现使用 Jena SDB[11],后台数据库为 Mysql5.0.18。运行环境:Windows XP,P4 1.86G CPU,1G RAM,160G 硬盘。实验选取表 1 所列的 4 个初始查询模型,它们的返回结果均为空。

表 1 查询实例

名称	查询模型实例
Q_1	(?x,type,AssistantProfessor)(?x,ProceedingEditorof,?y)(y,type,http://www.Department0.Universtiy0.edu:Proceedings)(?z,isTaughtBy,?x)(?z,name,"mathematics")(y,type,Proceedings)
Q_2	(?x,teacherOf,?z)(?x,type,FullProfessor)(?x,worksFor,http://www.Department0.Universtiy0.edu)(?z,type,course)(?z,name,"mathematics")
Q_3	(?y,publicationAuthor,?x)(?y,type,ConferencePaper)(?x,worksFor,http://www.Department0.University0.edu # ResearchGroup1)(?x,type,Lecturer)(?z,isTaughtBy,?x)(?z,name,"mathematics")
Q_4	(?x,type,TeachingAssistant)(?x,teachingAssistantOf,http://www.Department0.University0.edu # Course3)(?x,mastersDegreeFrom,http://www.Department0.University0.edu)(?z,isTaughtBy,?x)(?z,name,"mathematics")

实验对 4 个初始查询进行松弛和同源词替换,在查询响应时间、查全率和查准率 3 个方面对本文的方法进行评估。在测试查询响应时间时,为方便与文献[5,7]中的方法进行比较,查询中止条件 Top-K 中的 K 取 50,在测试查全率时,查询中止条件采用指定阈值的方法。Q_1、Q_2、Q_3 和 Q_4 是本文方法(记作 SQDR)得出的结果,Q_1'、Q_2'、Q_3' 和 Q_4' 以及 Q_1''、Q_2''、Q_3'' 和 Q_4'' 分别是文献[5,7]中的方法(分别记作 SQR1 和 SQR2)。

(1)查询响应时间测试

查询响应时间是指得到 $K=50$ 个结果时所用的时间。图 3 给出阈值分别为 0.05、0.10、0.15、0.20、0.25、0.30、0.35、0.40、0.45、0.50 时 Q_1、Q_2、Q_3 和 Q_4 的查询响应时间。可以看出,对于同一个查询来说,阈值设定得越小,查询响应时间则越大。图 4 给出了 3 种不同查询方法的查询响应时间。Q_1、Q_2、Q_3 和 Q_4 和 Q_{t1}、Q_{t2}、Q_{t3} 和 Q_{t4} 是本文所使用的方法(阈值为 0.15),其中前者直接使用 WordNet 得到同源词集合(记作 SQDR),后者使用 w-index 产生同源词集合(记作 SQDR1)。可以看出,本文的方法在查询响应时间上没有优势,因为该方法首先需要对初始查询进行松弛,而后选取合适的松弛查询进行同源词替换并计算语义距离。但可以看到,本文使用 w-index 代替 WordNet 得到同源词集合,可以大幅度减少时间消耗,提高系统效率。

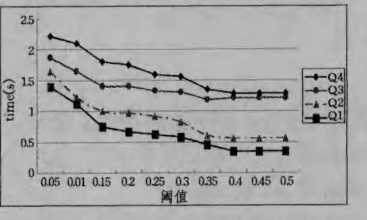

图 3 不同阈值下的查询响应时间

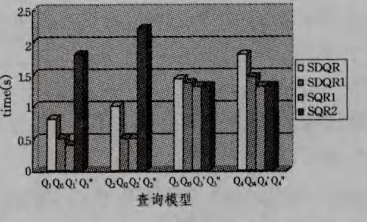

图 4 查询响应时间

(2)查全率和查准率测试

对于每个查询实例,生成一个相应的数据集 D_i。D_i 中包含了 30 条与该查询相关的三元组,是通过上面给出的松弛策略分别获得的前 10 条相关查询结果、去掉重复三元组并适当添加随机选择的三元组组合而成,之后再由用户选定其中 10 个与初始查询最为相关的三元组。查全率 Recall 是查询结果中相关三元组数与数据集中相关三元组总数之比,查准率 Precision 用来对查询结果的准确率进行测试。对每一个查询实例返回前 10 个三元组,通过测试这 10 个三元组与用户选定的 10 个最相关的三元组的重叠程度进行查准率测试。查全率和查准率定义如下。

$$\text{Recall}=\frac{\text{查询结果中包含的数据集中的三元组数目}}{\text{数据集中的三元组数目}}$$

$$\text{Precision}=\frac{|\text{最相关的 10 个三元组}\cap\text{本文方法返回的 10 个三元组}|}{10}$$

图 5 给出了 3 种查询方法在查全率上的对比,图 6 给出了 3 种查询策略在查准率上的对比,可以看出本文方法的查全率和查准率均高于另外两种方法。

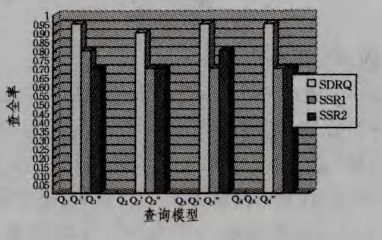

图 5 查全率

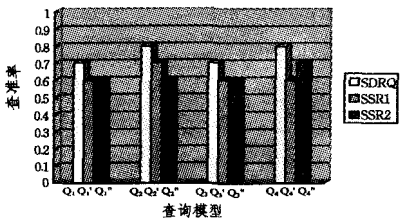

图 6 查准率

综上可以看出，本文提出的方法在权重分配和语义距离评估方面是合理的。一方面，通过对查询及 RDF 的预处理，可以在很大程度上减少由于查询松弛及同源词替换导致的时间性能上的不足，另一方面，在查准率以及查全率上，本文的方法均有良好的表现。

结束语 为了解决 RDF 查询返回结果为空或少量的问题，本文提出基于语义的 RDF 近似查询的处理方法。首先，初始查询通过 RDFS 蕴含规则进行松弛，之后利用权重对松弛查询进行语义选择并进行同源词替换，最后利用语义距离选取与初始查询在语义上相近的结果。在此基础上，给出基于语义的 RDF 近似查询处理的算法。实验结果表明，本文提出的方法能够为用户提高更多更为准确的查询结果，并且有较好的查询响应时间。目前 RDF 查询(包括近似查询)方法主要关注的是 RDF 显式表示的信息，基于推理机制的 RDF 查询能够抽取出 RDF 非显式表示、但能由 RDF 显式表示信息推导出的信息。未来我们将为本文提出的近似查询方法提供推理机制。

参 考 文 献

[1] Berners-Lee T, Handler J, Lassila O. The Semantic Web[M]. Scientific American, 2001, 184: 34-43

[2] Miller E, Swick R, Brickley D. Resource Description Framework RDF[C]// Recommendation, W3C. 2004

[3] Clark K G. RDF Data Access Use Cases and Requirements [C]// W3C Working Draft. March 2005

[4] Hurtado C A, Poulovassilis A, Wood P T. A relaxed approach to RDF querying[C]// Proceedings of the 5th International Semantic Web Conference. LNCS, 2006, 4273: 314-328

[5] Hurtado C A, Poulovassilis A, Wood P T. Query relaxation in RDF[J]. Journal of Data Semantics, 2008, 10: 31-61

[6] Poulovassilis A, Wood P T. Combining approximation and relaxation in semantic web path queries[C]// Proceedings of the 2010 International Semantic Web Conference. LNCS, 2010, 6496: 631-646

[7] Huang H, Liu C F, Zhou X F. Computing relaxed answers on RDF databases[C]// Proceedings of the 9th International Conference on Web Information Systems Engineering. LNCS, 2008, 5175: 163-175

[8] Andreasen T, Bulskov H, Knappe R. From ontology over similarity to query evaluation [C] // Proceedings of the 2nd CoLogNETElsNET Symposium-Questions and Answers: Theoretical and Applied Perspectives. 2003: 39-50

[9] Maedche A, Staab S. Measuring similarity between ontologies [C] // Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. LNCS, 2002, 2473: 251-263

[10] Guo Y, Pan Z, Hein J. An evaluation of knowledge base systems for large OWL datasets[C]// Proceedings of the 3rd International Semantic Web Conference. LNCS, 2004, 3298: 274-288

[11] Jena S D B[OL]. http://jena. hpl. hp. com/wiki/SDB

[12] Zhu L, Ma Q, Liu C N. Semantic-distance based evaluation of ranking queries over relational databases[J]. Journal of Intelligent Information Systems, 2010, 31: 415-445

(上接第 151 页)

[13] Boneh D, Crescenzo G, Ostrovsky R, et al. Public key encryption with keyword search[C]// Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques (Eurocrypt'04). 2004: 506-522

[14] Abdalla M, Bellare M, Catalano D, et al. Searchable Encryption Revisited: Consistency Properties, Ration to Anonymous IBE, and Extensions[C]// Proceedings of International Cryptology Conference (CRYPTO'05). LNCS 3621, 2005: 205-222

[15] Hwang Y H, Lee P J. Public Key Encryption with Conjunctive Keyword Search and Its Extension to a Multi-User System[C]// Proceedings of International Conference on Pairing-Based Cryptography(Pairing'07). LNCS 4575, 2007: 2-22

[16] Yang Y J, Bao F, Ding X H, et al. Multiuser private queries over encrypted databases[J]. Journal of Applied Cryptography, 2009, 1(4): 309-319

[17] Yang Y J, Lu H B, Weng J. Multi-user private keyword search for cloud computing[C]// Proceedings of Third IEEE International Conference on Cloud Computing Technology and Science (CloudCom'11). 2011: 264-271

[18] DARPA Information Science and Technology Study Group. Privacy with security [R]. Technical report. http://www. cs. berkeley. edu/-tygar/papers/ISAT-final- briefing. pdf, 2002-12

[19] Boneh D, Lynn B, Shacham H. Short signatures from the Weil pairing[J]. Journal of Cryptology, 2004, 17(4): 297-319

[20] Zhu R, Yang G M, Wong D. An efficient identity-based key exchange protocol with KGS forward secrecy for low-power devices[C] // Proceedings of Internet and Network Economics First International workshop (WINE'05). LNCS 3828, 2005: 500-509

(上接第 155 页)

[15] Shamus Software Ltd., Miracl library [OL]. http://www. shamus. ie/index. php? page=home

[16] Ren K, Lou W, Zeng K, et al. On broadcast authentication in wireless sensornetworks [J]. IEEE Trans. on Wireless Commun., 2007, 6(11): 4136-4144

[17] Boneh D, Franklin M. Identity-based encryption from the Weil pairing[A] // CRYPTO 2001, 2001[C]. New York: Springer-Verlag, 2001: 213-229

[18] Barreto P, Kim H, Bynn B, et al. Efficient algorithms for pairing-based cryptosystems [A] // CRYPTO 2002, 2002 [C]. New York: Springer-Verlag, 2002: 354-368

[19] Bao F, Deng R H, Zhu H. Variations of Diffie-Hellman problem [A] // Proc. ICS, 2003[C]. New York: Springer-Verlag, 2003: 301-312