基于二分图的RDF关键词扩展查询方法

郑志蕴 王振涛 张行进 王振飞
(郑州大学信息工程学院 郑州 450001)

摘 要 使用图表示RDF数据可以保持数据间的关联信息和语义信息,越来越多的关键词查询方法基于图结构实现RDF数据的查询处理。将二分图与RDF数据图相结合,定义RDF二分图模型,并提出一种基于二分图的RDF关键词扩展查询方法KERBG。该方法将文本信息封装在二分图顶点标签上,以支持对关系的查询;利用关键词同义词扩展技术对查询关键词进行语义扩展,有效解决同一对象的描述用词的多样性问题,进而提高查准率;利用RDF二分图的反对称邻接矩阵及其幂矩阵构造包含关键顶点的查询结果子图,实现关键词查询处理,并降低查询响应时间。实验结果表明,在查准率和查询响应时间方面,提出的KERBG方法优于当前主流方法。

关键词 RDF,二分图,关键词查询,反对称邻接矩阵,同义词扩展
中图法分类号 TP391.3 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.11.053

Keyword Expansion Query Approach over RDF Data Based on Bipartite Graph

ZHENG Zhi-yun WANG Zhen-tao ZHANG Xing-jin WANG Zhen-fei
(School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China)

Abstract Using graph to express RDF data can both retain data correlation information and semantic information. To date, more and more keyword query methods based on graph structure have realized RDF data query processing. In this paper, an approach named RDF keyword expansion query approach based on bipartite graph was proposed. This approach enables keyword-based query over RDF data. RDF data is modeled as a RDF bipartite graph, in which all text information is encapsulated by nodes labels. Based on the keyword synonym expansion technology, the approach realizes the semantic extension of query keywords, effectively solves the problem of delivering the same object description words and also improves the query precision. Through RDF bipartite graph of the anti-symmetric adjacency matrix and its power matrix, the approach structures the subgraphs of query results consisting of key vertices, realizes the keyword query processing and then reduces the query response time. The experimental results show that when comparing query precision and query response time, KERBG method proposed in this paper is better than the current mainstream methods.

Keywords RDF, Bipartite graph, Keyword search, Anti-symmetric adjacency matrix, Synonym expansion

1 引言

RDF(Resource Description Framework)是一个用于描述Web资源的框架[1],提供了定义描述资源的词汇表时必须遵循的规则,已经成为语义数据描述的标准。它允许用户使用自己的词汇描述资源,具有领域无关的特性,因此在越来越多的领域得到应用。RDF用主语(subject)、谓词(predicate)、宾语(object)的三元组形式来描述Web上的资源。据W3C的SWEO(Semantic Web Education and Outreach)研究小组统计,截止2012年3月,互联网上的RDF数据集中的三元组数量已经达到520亿[2]。

通常RDF数据被表示为一个带标签的有向图,图中的结点对应三元组中的主语和宾语,谓词则为边。使用图表示RDF数据既能保持数据间的关联信息又不丧失语义信息[3],因此,RDF数据的查询处理常被转换为子图匹配问题,即在RDF数据图上定位包含关键词的斯坦纳树(Steiner Tree)[4]。然而,由于RDF数据图中包含很多文本信息,结点之间关联多,图规模巨大,导致基于图的关键词查询响应速度较慢。另外,由于人们对现实生活中相同对象的描述用词存在着多样性,两个人使用同样的关键词描述同一对象的概率小于20%[5],导致基于关键词的查询返回无关结果,降低了关键词查询的查准率。因此,如何在RDF数据图上执行高效的关键词查询成为一个重要问题。

本文提出一种基于二分图的RDF关键词扩展查询方法(Keyword Expansion Query Approach over RDF Data Based on Bipartite Graph, KERBG)。该方法将RDF数据建模为RDF二分图,支持对实体和关系进行查询。使用WordNet[6]词典对查询关键词进行同义词扩展,提高关键词查询的查准率。

到稿日期:2015-11-11 返修日期:2016-03-07 本文受河南省国际科技合作项目(144300510007),郑州市科技攻关计划项目(141PPTGG368)资助。

郑志蕴(1962—),女,博士,教授,主要研究方向为分布式计算、语义网;王振涛(1989—),男,硕士生,主要研究方向为语义网、大数据处理;张行进(1973—),男,博士,副教授,主要研究方向为语义网、大数据处理;王振飞(1973—),男,博士,副教授,主要研究方向为云计算、语义网,E-mail:iezfwang@zzu.edu.cn(通信作者)。

依据 RDF 二分图的反对称邻接矩阵及其幂矩阵，实现关键词查询的快速响应，降低关键词查询的查询响应时间。通过相关性评测函数对查询结果子图进行相关性排序，输出 top-k 个查询结果子图。

SPARQL[3]（Simple Protocol and RDF Query Language）是 W3C 提出的 RDF 数据查询标准语言，也是目前被广泛采用的一种 RDF 上的查询语言。当前，大多数 RDF 系统都支持 SPARQL 查询。SPARQL 共有 4 种查询方式，分别为 SELECT，CONSTRUCT，DESCRIBE 和 ASK。目前最常用的是 SELECT 查询方式，它与 SQL 的语法相似，用来返回满足条件的数据。对于普通用户，这种形式化 RDF 数据查询方法的语法规则显得较为复杂，使用起来较为不便。本文提出的关键词查询方式虽然不支持形式化查询方法，但恰好与该查询方式形成互补，使得普通用户能够更方便地检索和重用 RDF 数据。

本文第 2 节介绍相关研究工作；第 3 节给出 RDF 二分图模型及相关定义和定理；第 4 节描述基于二分图的 RDF 关键词扩展查询方法；第 5 节给出实验结果和分析；最后对本文工作进行总结。

2 相关工作

目前，基于图的 RDF 数据关键词查询方法主要分为两类：1）通过构建 RDF 数据为 RDF 图模型[7-14]，将 RDF 图上的关键词查询问题转化为大图上包含关键词的子图匹配问题，并建立相关索引，以此来降低关键词查询的响应时间；2）通过构建 RDF 数据为二分图模型[15,16]，从而支持用户将关系作为关键词进行查询。

EASE[9]是第一类中具有代表性的方法，该方法是由清华大学和新加坡国立大学于 2008 年共同开发的适用于非结构、半结构和结构化 RDF 数据的关键词查询方法。在 EASE 方法的查询过程中，首先将 RDF 数据建模为 RDF 图，求出 RDF 图的邻接矩阵，将 RDF 图上的关键词查询问题转化为定位包含关键词的斯坦纳图问题，然后通过邻接矩阵的 N 次方找到 top-k 个包含关键词的斯坦纳图作为查询结果子图，最后以查询结果子图的路径长度和其倒数为其评分，将 top-k 个查询结果子图进行排序输出。该方法的时间复杂度为 $O(n^2)$（n 为 RDF 图中顶点的个数）。然而，随着 RDF 数据规模的增大，其对应的 RDF 图的顶点数 n 会变得很大，查询的响应时间会随着 n 的变大而增加，极大地影响了关键词查询的效率。另外，用 RDF 图表示 RDF 数据，不支持用户对 RDF 数据中的关系进行查询。

KREAG[16]是第二类中具有代表性的方法，该方法是基于实体三元组关联图的 RDF 数据关键词查询方法。该方法将 RDF 数据建模为顶点带标签的实体三元组关联图，文本信息全部封装在关联图顶点标签上，不仅支持用户对关系进行查询，而且将 RDF 数据中实体间的关联关系转化为关联图中顶点间的通路，有效保持了 RDF 数据的语义性。该方法将关键词查询问题转化为关联图上有向斯坦纳树的查找问题，在保证近似比为查询关键词个数的前提下，利用近似算法实现查询快速响应，并通过合理的评分方式衡量查询结果的相关性，支持 top-k 查询，算法的时间复杂度为 $O(mn)$（m 为查询关键词的个数，n 为实体三元组关联图中顶点的个数）。然而，该查询方法需要提前建立关键词倒排索引和最短路径索引，索引时间和空间消耗太大。虽然通过降低索引步长可以减少索引建立的时间和空间开销，但是索引步长变短会降低查准率。

查询扩展[17-19]是公认的能够有效提高查准率的技术之一，基本思想是利用与查询关键词相关的词语对查询进行修正，找到更准确的相关文档，提高查准率。各个领域的学者提出了各种不同的查询扩展方法，主要包括基于用户查询日志、基于关联规则和基于本体（或领域本体）的查询扩展方法。在 RDF 数据的关键词查询过程中，运用适当的查询扩展技术可以提高查准率。

通过对上述方法的分析研究，综合其优点，提出了基于二分图的 RDF 关键词扩展查询方法 KERBG，目的是提高关键词查询的查准率，减少查询响应时间。

3 RDF 二分图模型

本节首先给出 RDF 二分图模型的定义，理论上说明该模型不仅可以保持 RDF 数据实体间的关联关系，而且支持对关系进行查询。然后给出反对称邻接矩阵的定义及其性质，推导出反对称邻接矩阵的幂矩阵的相关性质，进而得出定理，证明通过 RDF 二分图的反对称邻接矩阵及其幂矩阵能够构造出包含查询关键词的查询结果子图。

3.1 模型定义

RDF 数据是指以 RDF 三元组形式组织起来的语义信息，可以通过有向图模型表示。本文用 (s,p,o) 描述一个 RDF 三元组，并简记为 t，用 $s(t)$，$p(t)$ 和 $o(t)$ 分别表示三元组 t 中的主体、谓词和客体。同理，所有 RDF 三元组的集合记为 T，$s(T)$，$p(T)$ 和 $o(T)$ 分别表示所有三元组中的主体集合、谓词集合和客体集合。依据 RDF 推荐标准文档[1]，给出 RDF 有向图的定义。

定义 1（RDF 有向图） 设 $G=\langle V,E\rangle$ 表示一个带标签的 RDF 有向图，其中 $V=\{v\mid v\in s(T)\cup o(T)\}$，由 RDF 三元组中主体和客体组成的顶点的集合 $E=\{\langle s(t),o(t)\rangle\mid t\in T\}$ 描述主体和客体之间关系的谓词组成的有向边的集合，边的方向由主体顶点指向客体顶点。

本文称 RDF 三元组中的主体和客体为实体，称图 G 中的顶点为实体顶点，并将所有实体的集合记为 $entity(T)$。图 G 中边上的标签存储三元组中谓词的文本信息。如果存在从实体 e_1 到 e_k 的实体关系序列 $e_1p_1e_2p_2\cdots p_{k-1}e_k$，则称 e_1 关联 e_k。图 1 是一个真实的 RDF 数据片段的 RDF 有向图表示，图中边上的标签存储了部分 URI 信息。

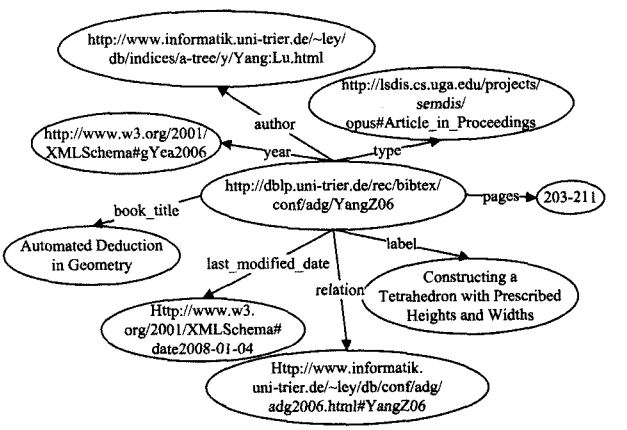

图 1 RDF 图示例

将 RDF 数据转化为 RDF 图后，RDF 数据上的关键词查询问题就转换成了 RDF 图上子图匹配大图的问题，可以将图论中的遍历算法运用到关键词查询上，以提高关键词查询效率。然而，当基于 RDF 图进行关键词查询时，RDF 图的边上的关联信息不能作为查询对象，无法处理将关系作为关键词进行查询的情况。为了解决该问题，且不破坏实体之间的关联关系，本文提出 RDF 数据的二分图模型，模型定义如下。

定义 2(RDF 二分图) 设 $B(G)=\langle V_E \cup V_P, E\rangle$ 表示一个 RDF 二分图，其中 G 是一个 RDF 有向图，V_E 和 V_P 分别是它的实体顶点集合和谓词顶点集合，且有 $V_E \cap V_P=\varnothing$，$V_E=\{v_e \mid e\in entity(T)\}$，$V_P=\{v_p \mid p\in p(T)\}$，$E$ 是 B 中实体顶点和谓词顶点之间的边的集合，且有 $E=\{\langle v_e, v_p\rangle \mid (v_e\in V_E \wedge v_p\in V_P \wedge (\exists t\in T(s(t)=e \wedge p(t)=p)))\} \cup \{\langle v_p, v_e\rangle \mid (v_e\in V_E \wedge v_p\in V_P \wedge (\exists t\in T(p(t)=p \wedge o(t)=e)))\}$。

在该 RDF 二分图模型中，RDF 三元组中的谓词单独分开作为一类顶点，称为谓词顶点；RDF 三元组中主体和客体作为另一类顶点，称为实体顶点。每个 RDF 三元组被分割为两个二元组$\langle s,p\rangle$和$\langle p,o\rangle$，这种划分使得能够针对谓词上的关系进行查询。另外，RDF 二分图模型通过实体顶点到谓词顶点再到实体顶点的通路保持实体之间的关联。图 1 对应的 RDF 二分图如图 2 所示。

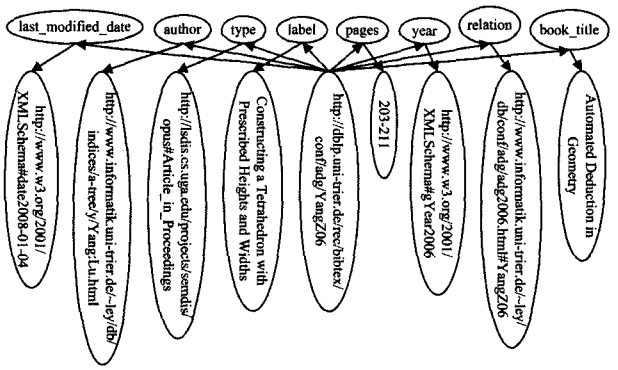

图 2 RDF 二分图示例

3.2 相关定义和定理

对于传统的 RDF 图模型，RDF 二分图模型的优势不仅在于支持对属性或关系的查询，而且它的邻接矩阵具有很多特殊性质。在 RDF 查询过程中，通过运用这些性质，能够降低矩阵运算的时间复杂度，从而提高查询速率。下面将从理论上去阐述它的这些性质及相关定义和定理。

在 RDF 二分图模型的基础上，根据图论中邻接矩阵的相关定义，给出 RDF 二分图的邻接矩阵定义。

定义 3(RDF 二分图的邻接矩阵) 设 RDF 二分图$B(G)=\langle V_E \cup V_P, E\rangle$，$V_E=\{v_1, v_2, \cdots, v_{|V_E|}\}$，$V_P=\{v_{|V_E|+1}, v_2, \cdots, v_{|V_E|+|V_P|}\}$，令 $a_{i,j}^{(1)}$ 为顶点 v_i 邻接到 v_j 边的条数，并且 $a_{i,j}^{(1)}$ 的值只能是 0 或 1，称$(a_{i,j}^{(1)})_{n\times n}(n=|V_E|+|V_P|)$为 B 的邻接矩阵，记作 $A(B)$，或简记为 A。

根据矩阵理论知识可知，反对称矩阵作为一种特殊的矩阵，具有特殊的性质和广泛的用途。下面给出反对称矩阵的定义及其性质。

定义 4(反对称矩阵) 设 O 是一个 n 阶方阵，如果 $O^{\mathrm{T}}=-O$(其中 O^{T} 为 O 的转置矩阵)，则称 O 为反对称矩阵。

性质 1 设 A 是任意一个 n 阶方阵，则 $A-A^{\mathrm{T}}$(其中 A^{T} 为 A 的转置矩阵)为反对称矩阵。

证明：因为$(A-A^{\mathrm{T}})^{\mathrm{T}}=A^{\mathrm{T}}-(A^{\mathrm{T}})^{\mathrm{T}}=A^{\mathrm{T}}-A=-(A-A^{\mathrm{T}})$，所以 $A-A^{\mathrm{T}}$ 为反对称矩阵。

由反对称矩阵的定义和性质，结合 RDF 二分图的邻接矩阵的定义，下面给出 RDF 二分图的反对称邻接矩阵的定义及其性质，其中 r 表示矩阵的幂次。

定义 5(RDF 二分图的反对称邻接矩阵) 设 RDF 二分图 $B(G)=\langle V_E \cup V_P, E\rangle$，$V_E=\{v_1, v_2, \cdots, v_{|V_E|}\}$，$V_P=\{v_{|V_E|+1}, v_2, \cdots, v_{|V_E|+|V_P|}\}$，$A=(a_{i,j}^{(1)})_{n\times n}$ 为图 B 的邻接矩阵，令 $o_{i,j}^{(1)}=a_{i,j}^{(1)}-a_{j,i}^{(1)}$，称$(o_{i,j}^{(1)})_{n\times n}(n=|V_E|+|V_P|)$为 B 的反对称邻接矩阵，记作 $O(B)$，或简记为 O。

性质 2 已知 RDF 二分图 B，其反对称邻接矩阵 O 为 n 阶方阵，且有如下形式：

$$\left[\begin{array}{ccc:ccc} 0 & \cdots & 0 & o_{0,s} & \cdots & o_{0,n-1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & o_{s-1,s} & \cdots & o_{s-1,n-1} \\ \hdashline -o_{0,s} & \cdots & -o_{s-1,s} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -o_{0,n-1} & \cdots & -o_{s-1,n-1} & 0 & \cdots & 0 \end{array}\right]$$

可简记为：$\begin{bmatrix} C & D \\ -D & E \end{bmatrix}$，其中 C 为 $|V_P|$ 阶零矩阵，E 为 $|V_E|$ 阶零矩阵。

性质 3 已知 O 为 B 的反对称邻接矩阵，则当 r 为奇数时，O^r 为反对称矩阵；当 r 为偶数时，O^r 为对称矩阵。

证明：因为 O 为反对称邻接矩阵，依据定义 4 可知 $O^{\mathrm{T}}=-O$，则有$(O^r)^{\mathrm{T}}=(O^{\mathrm{T}})^r=(-O)^r$，当 r 为奇数时，$(O^r)^{\mathrm{T}}=-O^r$，所以 O^r 为反对称矩阵，当 r 为偶数时，$(O^r)^{\mathrm{T}}=O^r$，O^r 为对称矩阵。

根据反对称邻接矩阵的性质，运用矩阵的幂运算规则，可以推导出如下结论。

推论 1 反对称邻接矩阵 O 的幂矩阵具备以下性质：当 r 为奇数时，幂矩阵 O^r 形如$\begin{bmatrix} 0 & (-1)^{(r-1)/2}P^r \\ (-1)^{(r+1)/2}P^r & 0 \end{bmatrix}$；当 r 为偶数时，幂矩阵 O^r 形如$\begin{bmatrix} (-1)^{r/2}P^r & 0 \\ 0 & (-1)^{r/2}P^r \end{bmatrix}$。

下面给出定理，说明依据 RDF 二分图的反对称邻接矩阵的幂矩阵，能够得到包含关键词的关系序列，即查询结果子图。

定理 1 设 O 为 RDF 二分图 B 的反对称邻接矩阵，B 的顶点集 $V=\{v_1, v_2, \cdots, v_n\}$，则 O 的 r 次幂 $O^r(r\geqslant 1)$ 中元素 $o_{i,j}^{(r)}$ 的绝对值为 B 中 v_i 和 v_j 之间步长为 r 的关系序列的个数。

证明：(1)当 $r=1$ 时，因为 O 为 RDF 二分图 B 的反对称邻接矩阵，根据定义 3 和定义 5 可知 $o_{i,j}^{(1)}=a_{i,j}^{(1)}-a_{j,i}^{(1)}$，且当 $a_{i,j}^{(1)}=a_{j,i}^{(1)}=0$ 时，$o_{i,j}^{(1)}=0$，顶点 v_i 邻接到顶点 v_j 边的条数为 0；当 $a_{i,j}^{(1)}=1, a_{j,i}^{(1)}=0$ 时，$o_{i,j}^{(1)}=1$，顶点 v_i 邻接到顶点 v_j 边的条数为 1；当 $a_{i,j}^{(1)}=0, a_{j,i}^{(1)}=1$ 时，$o_{i,j}^{(1)}=-1$，顶点 v_j 邻接到顶点 v_i 边的条数为 1。综上可知，$o_{i,j}^{(1)}$ 的绝对值为 B 中 v_i 和 v_j 之间步长为 1 的关系序列的个数。

(2)假设当 $r=k$ 时，$o_{i,j}^{(k)}$ 的绝对值为 B 中 v_i 和 v_j 之间步长为 k 的关系序列的个数，可知 $o_{i,j}^{(k)}=a_{i,j}^{(k)}-a_{j,i}^{(k)}$。当 $r=k+1$ 时，有 $o_{i,j}^{(k+1)}=a_{i,j}^{(k+1)}-a_{j,i}^{(k+1)}=\sum_{f=0}^{n-1}(a_{i,f}^{(k)}\cdot a_{f,j}^{(1)}-a_{j,f}^{(1)}\cdot a_{f,i}^{(k)})$，且

当 f 满足 $a_{f,j}^{(1)}=a_{j,f}^{(1)}=0$ 时，v_i 和 v_f 之间无论是否存在步长为 k 的关系序列，v_i 和 v_j 之间都不存在步长为 $k+1$ 的关系序列，即 $o_{i,j}^{(k+1)}=0$；当 f 满足 $a_{f,j}^{(1)}=1, a_{j,f}^{(1)}=0$ 时，v_i 和 v_j 之间步长为 $k+1$ 的关系序列的个数等于 v_i 和 v_f 之间步长为 k 的关系序列的个数，即 $o_{i,j}^{(k+1)}=\sum_{f\in\alpha}a_{i,f}^{(k)}$（$\alpha$ 为满足条件的 f 的取值空间）；当 f 满足 $a_{f,j}^{(1)}=0, a_{j,f}^{(1)}=1$ 时，v_i 和 v_j 之间步长为 $k+1$ 的关系序列的个数等于 v_f 和 v_i 之间步长为 k 的关系序列的个数，即 $o_{i,j}^{(k+1)}=-\sum_{f\in\beta}a_{f,i}^{(k)}$（$\beta$ 为满足条件的 f 的取值空间）。综上可知，$o_{i,j}^{(k+1)}$ 的绝对值为 B 中 v_i 和 v_j 之间步长为 $k+1$ 的关系序列的个数。

综合(1)、(2)，对任意的正整数 $r\geqslant 1$，O 的 r 次幂矩阵 O^r 中的元素 $o_{i,j}^{(r)}$ 的绝对值为 B 中 v_i 和 v_j 之间步长为 r 的关系序列的个数。

定义 6（关键顶点） 设 RDF 二分图 $B(G)=\langle V_E\cup V_P, E\rangle$，关键词集 $W=\{w_1,w_2,\cdots,w_m\}$，对应的同义词集簇 $S=\{s_1,s_2,\cdots,s_m\}$，其中 w_i 为单个关键词，s_i 为关键词 w_i 的同义词集。若 w_i 或 s_i 中的同义词命中 $B(G)$ 中的某个顶点，则称该顶点为关键顶点。

定义 7（查询结果子图评分函数） 设关键词集 $W=\{w_1,w_2,\cdots,w_m\}$ 的查询结果子图为 QRS，其包含的关键词个数为 x，定义其评分函数为：

$$score(QRS,W)=\frac{x}{size(QRS)} \tag{1}$$

其中，$size(QRS)$ 为查询结果子图中顶点的个数。

该评分函数使得包含关键词越多、顶点越少的查询结果子图的评分越高。利用此函数，可以评判查询结果子图的优劣，进而确定查询结果的 top-k。

4 基于二分图的 RDF 关键词扩展查询算法

本节首先给出 RDF 二分图构造算法的描述，并说明依据 RDF 二分图的邻接矩阵和反对称邻接矩阵的定义可以求出 RDF 二分图的反对称邻接矩阵。然后分 3 个部分（即关键词扩展、关键词匹配和查询结果子图构造）描述了关键词扩展查询算法，并给出关键词扩展查询算法的完整伪代码描述。

4.1 RDF 二分图构造算法

将 RDF 数据转化为 RDF 二分图，不仅保持了实体之间的关联关系，而且支持对实体和关系进行关键词查询。本文通过 RDF 二分图构造算法将 RDF 数据转化为 RDF 二分图，下面是对 RDF 二分图构造算法的具体描述。

第一步　定义二维数组 *edgeArray* 和一维数组 *vertexArray*，其中 *edgeArray* 存储 RDF 二分图的边结构，*vertexArray* 存储 RDF 二分图的顶点。

第二步　将每个三元组 t 分割为两个二元组 $\langle s,p\rangle$ 和 $\langle p,o\rangle$，并将其保存到 *edgeArray* 数组中，每行存储一个二元组。

第三步　遍历 *edgeArray* 数组，将所有不同的顶点存储在数组 *vertexArray* 中，且谓词顶点在前，实体顶点在后。

第四步　依据数组 *vertexArray* 和 *edgeArray* 构造出 RDF 二分图。

第五步　依据 RDF 二分图的邻接矩阵和反对称邻接矩阵的定义，求出 RDF 二分图的反对称邻接矩阵。

结合上面对 RDF 二分图构造算法的具体描述，下面给出 RDF 二分图构造算法的伪代码描述。

算法 1　RDF 二分图构造算法

```
输入：RDF 三元组 t
输出：RDF 二分图及其反对称邻接矩阵 O
1. Begin
2.     初始化 edgeArray 和 vertexArray 数组
3.     while(t∈T&t!=null)
4.       for i=0 to 2|T|-1
5.         edgeArray[i][0]=s(t)
6.         edgeArray[i++][1]=o(t)
7.         edgeArray[i][0]=o(t)
8.         edgeArray[i][1]=p(t)
9.   for j=0 to i
10.       if(edgeArray[j][0]∈o(T))
11.         vertexArray[j]=edgeArray[j][0]
12.       if (edgeArray[j][0]∈s(T))
13.         vertexArray[i-j]=edgeArray[j][0]
14.       if(edgeArray[j][1]∈p(T))
15.         vertexArray[i-j]=edgeArray[j][1]
16. 输出 B(G)及其反对称邻接矩阵 O
17. End
```

图 2 所示的 RDF 二分图的反对称邻接矩阵如图 3 所示。

$$\left[\begin{array}{cccccccc|cccccccc}
0&0&0&0&0&0&0&0&1&0&0&0&-1&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&-1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&-1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&-1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&-1&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&-1&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0\\
\hline
-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0
\end{array}\right]$$

图 3　反对称邻接矩阵示例

4.2 关键词扩展查询算法

关键词扩展查询算法包括关键词扩展、关键词匹配和查询结果子图构造 3 个部分。在关键词扩展过程中，首先对查询 O 进行关键词提取，得到关键词集合 $W=\{w_1,w_2,\cdots,w_m\}$，然后使用 WordNet 词典对查询关键词进行同义词扩展，得到关键词的同义词集簇 $S=\{s_1,s_2,\cdots,s_m\}$。

在关键词匹配过程中，最主要的任务是寻找关键顶点，即在 RDF 二分图中寻找包含关键词或其同义词的顶点。下面是对关键词匹配算法的具体描述。

第一步　定义一维数组 *vertexIndex*，存储关键顶点编号。

第二步　用关键词集合 W 中的关键词 w_i（$i\in\{1,2,\cdots,m\}$）匹配 RDF 二分图的所有顶点，将被命中的顶点的编号存储在数组 *vertext* 中。

第三步　用同义词集 s_i（$i\in\{1,2,\cdots,m\}$）中的同义词匹配 RDF 二分图的所有顶点，将被命中的顶点的编号存储在数组 *vertex* 中。

关键词匹配完成之后，根据关键词的个数 m，求出前 $m-1$ 次方的反对称邻接矩阵的幂矩阵作为构造查询结果子图的初始幂矩阵集合。在计算反对称邻接矩阵的幂矩阵时，本节

利用第三节中的反对称邻接矩阵的定义及性质，使得每次矩阵运算的时间复杂度为 $O(|V_P|(n-|V_P|))$ 或 $O(|V_E|(n-|V_E|))$，其中 $|V_P|$ 是 RDF 二分图中的谓词顶点数，$|V_E|$ 是 RDF 二分图中的实体顶点数，$|V_E|+|V_P|=n$。与一般的矩阵乘法运算相比，矩阵运算速率大为提高。下面是对查询结果子图构造算法的具体描述。

第一步　从矩阵 $O^{m-i+j}(i=1,j=0)$ 开始，遍历关键顶点所在的行和列。若矩阵在该位置的值非零，则把此时的行列号分别记为 *row* 和 *col*，并生成一个候选结果子图，该子图以 *row* 行对应的关键顶点为初始顶点，记为 *StartPoint*，以 *col* 列对应的关键顶点为末端顶点，记为 *StopPoint*，最后跳转至第二步；若扫描所有的关键顶点所在的行和列后并没有构造出满足条件的查询结果子图，则求出更高次的幂矩阵 $O^{m-i+j}(i=1,j=j+1)$，重新开始构造查询结果子图。

第二步　执行 $i=i+1$，遍历矩阵 O^{m-i+j} 中 *col* 对应的列，找到第一个值不为零的行，将该行对应的顶点存储在 *LinkedVertext* 中作为候选结果子图的备选顶点，并将该顶点的行编号存储在数组 $Row[i-2+j]$ 中，然后根据以下条件判断是否将 *LinkedVertex* 顶点加入候选结果子图中：

(1)候选结果子图是否已经包含 *LinkedVertex* 顶点。

(2)O^{i-1+j} 中 *row* 行 $Row[i-2+j]$ 列元素的值是否等于 0。

若 $m-i+j=1$，则跳转至第三步；若 O^{i-1+j} 中 *row* 行 $Row[i-2+j]$ 列元素的值等于 0 或候选结果子图已包含 *LinkedVertex* 顶点，则从 *LinkedVertex* 顶点的下一行开始遍历 *col* 列，找到下一个备选顶点，判断是否满足以上条件；若遍历完所有行都没有找到满足以上条件的 *LinkedVertext* 顶点，则删除该候选结果子图，继续在矩阵 O^{m-i+j} 中扫描关键顶点所在的行和列，生成新的候选结果子图，重复上面的操作；若候选结果子图中不包含 *LinkedVertex* 顶点，并且 O^{i-1+j} 中 *row* 行 $Row[i-2+j]$ 列元素的值非零，则将 *LinkedVertex* 顶点加入到候选结果子图中，并把 *row* 行 $Row[i-2+j]$ 列元素的值存储在数组 *EdgeDirection* 中(当 $i=2$ 时，边的方向由该值的正负确定；当 $i>2$ 时，边的方向由数组 *EdgeDirection* 中最后两个元素的积的正负确定)，然后执行下一个循环。

第三步　若候选结果子图已包含 *LinkedVertex* 顶点或 O^{i-1+j} 中 *row* 行 $Row[i-2+j]$ 列元素的值等于 0，则从 *LinkedVertex* 顶点的下一行开始遍历 *col* 列，找到下一个备选顶点，判断是否满足以上条件；若遍历完所有行都没有找到满足以上条件的 *LinkedVertex* 顶点，则删除该候选结果子图，继续在矩阵 O 中扫描关键顶点所在的行和列，生成新的候选结果子图，重复上面的操作；若候选结果子图中不包含 *LinkedVertex* 顶点，并且 O^{i-1+j} 中 *row* 行 $Row[i-2+j]$ 列元素的值非 0，则将 *LinkedVertex* 顶点加入到候选结果子图中，并把 *row* 行 $Row[i-2+j]$ 列元素的值存储在数组 *EdgeDirection* 中(当 $i=2$ 时，边的方向由该值的正负确定；当 $i>2$ 时，边的方向由数组 *EdgeDirection* 中最后两个元素的积的正负确定)，然后把 O 中 *row* 行 *col* 列元素的值存储在数组 *EdgeDirection* 中，边的方向由该值的正负确定，若候选结果子图包含所有关键顶点，则输出该候选结果子图，并结束扫描。

结合上面对关键词扩展、关键词匹配和查询结果子图构造 3 个部分的具体描述，下面给出关键词扩展查询算法的完整伪代码描述。

算法 2　关键词扩展查询算法

```
输入:Q,B(G)及其反对称邻接矩阵 O
输出:查询结果子图 QRS
1. Begin
2.   提取 Q 中的关键词,得到关键词集 W
3.   对 W 进行同义词扩展,得到同义词集簇 S
4.   for h=1 to m
5.     if(w_h 命中顶点 v)
6.     将顶点 v 的编号存储在 vertexIndex 中
7.   if(s_h 中的同义词命中顶点 v)
8.     将顶点 v 的编号存储在 vertexIndex 中
9.   if(O^{m-1}[vertexIndex[i],vertexIndex[j]]!=0)
10.    St=vertexIndex[i],Sp=vertexIndex[j]
11. for l=m-3 to 0
12. if((l+1)%2==0)
13.        temp=|V_E|
14. else temp=|V_P|
15. for k=temp to |V|
16.   if(O^{l+1}[k,Sp]!=0 & O^{m-3-l}[St,k]!=0 & k∉N)
17.   NodeArray[m-2-l]=k
18.   if(g==0&l!=0 & O^{m-3-l}[St,k]>0)
19.        EdgeDirection[g++]=1
20.   else EdgeDirection[g++]=-1
21.   if (g!=0 & l!=0)
22.     if(O^{m-3-l}[St,k] * O^{m-4-l}[St,N[m-3-l]]>=1)
23.          EdgeDirection[g++]=1
24.     else EdgeDirection[g++]=-1
25.   if(g!=0 & l==0)
26.     if(O^{m-3-l}[St,k] * O^{m-4-l}[St,N[m-3-l]]>=1)
27.          EdgeDirection[g++]=1
28.     else EdgeDirection[g++]=-1
29.     if(O[k,Sp]>0)
30.          EdgeDirection[g]=1
31.     else EdgeDirection[g]=-1
32.   输出 top-k 个 QRS
33. End
```

在算法 2 中，第 12－14 行根据 l 的奇偶，变换 *temp* 的取值，减少了寻找备选顶点的时间，第 16 行的判断条件决定备选顶点是否加入候选结果子图，第 18－31 行确定候选结果子图边的方向。该算法的时间复杂度主要由反对称矩阵的幂运算和查询结果子图构造运算这两部分贡献。在本节前面提到一次反对称矩阵乘法运算的时间复杂度为 $O(|V_P|(n-|V_P|))$ 或 $O(|V_E|(n-|V_E|))$，为方便起见，这里令 $V_{min}=\min\{|V_E|,|V_P|\}$，且有 $0<V_{min}<\frac{n}{2}$，则一次反对称矩阵乘法运算的时间复杂度记为 $O(|V_{min}|(n-|V_{min}|))$，解一元二次方程 $|V_{min}|(n-|V_{min}|)-\frac{n}{4}=0$ 可知，当 $\frac{n-\sqrt{n^2-n}}{2}<|V_{min}|<\frac{n+\sqrt{n^2-n}}{2}$ 时，$|V_{min}|(n-|V_{min}|)<\frac{n}{4}$，显然又可知，当 $n\to+\infty$，$\frac{n-\sqrt{n^2-n}}{2}\to 0$，$\frac{n+\sqrt{n^2-n}}{2}\to n$，因此，对于 m 个关键词的查询，反对称矩阵的幂运算的时间复杂度为 $O((m-1)|V_{min}|(n-|V_{min}|))<O((m-1)n/4)$。而查询结果子图构造运算的时间复杂度为 $O((m-1)n/2)$，因此，该算法的时间复

杂度为 $O((m-1)n/2)$。文献[9]中的 EASE 方法的算法时间复杂度为 $O(n^2)$，文献[15]中的 KREAG 方法的算法时间复杂度为 $O(mn)$，相比之下本算法在执行时间方面有较大优势。

5 实验与结果分析

为了验证提出的基于 RDF 二分图的关键词扩展查询方法的性能，实验使用 Java 语言、语义网开发工具 JENA 以及 MySQL 数据库实现了关键词查询，并将本文提出的方法 KERBG 与第 2 节提到的两类基于图的 RDF 数据关键词查询方法中具有代表性的 KREAG、BLINKS 和 EASE 方法进行对比，对比它们之间的查询响应时间和查询效果。

5.1 实验环境及实验数据

实验环境配置如表 1 所列。

表 1 实验环境

组件	详细描述
JDK 版本	1.7.0_03
MyEclipse	9.0
OS	Windows 7 sp1 32 位
CPU	Intel i3-2130 3.40GHz
内存	4G
网络	局域网

实验采用真实的数据集 swetodblp[20]，数据主题是计算机科学领域发表文章的信息。该数据中共包含 681636 个三元组，将其转化为 RDF 二分图用时 569s，存储占用 53.6MB，边数和顶点数分别为 1026375 和 373219。

本文在数据集上测试了两组共 10 个查询(见表 2)，分别包含 3－5 个查询关键词。第一组 Q1－Q5 是不包含关系的测试查询。第二组 Q6－Q10 是包含关系的测试查询。分别对查询响应时间及查询结果的相关性和正确性进行评价，评价原则和结果的评测指标参见 5.2 节和 5.3 节。

表 2 查询示例

查询	关键词
Q1	ChoiK04 ASP-DAC 2004
Q2	Book_Chapter Amelia Rafiul
Q3	Blakeley95 OQL[C++] 2002-01-03 Jos
Q4	Erwig98 springer 1548 AMAST
Q5	43 Database Object Daniel ACFHK95
Q6	ChoiK04 book_title 2004
Q7	Book_Chapter writer Rafiul
Q8	Jurgen Object label author
Q9	Publisher springer 1548 AMAST
Q10	43 pages writer Daniel ACFHK95

5.2 查询响应时间的分析比较

图 4 示出了 KERBG，KREAG，BLINKS 和 EASE 方法对表 1 中 10 个示例的查询响应时间，4 种方法均设置为返回前 5 个查询结果。其中，KREAG 方法的索引步长设置为 8。

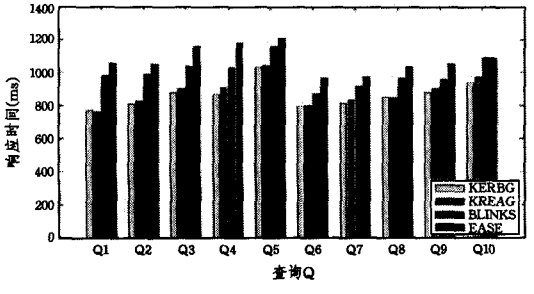

图 4 不同查询的响应时间

从图 4 可以看出，EASE 方法响应时间最长，原因是寻找包含关键词的斯坦纳图消耗时间长。另外，BLINKS 和 EASE 方法的第二组查询的响应时间普遍少于第一组查询的响应时间，原因是不支持对关系的查询，导致第二组查询命中的关键词数目减少，进而响应时间偏少。通过计算图中不同方法对查询示例的平均响应时间可知，KERBG 方法较 KREAG 方法其响应时间降低了 1.9%，而较 BLINKS 和 EASE 方法响其应时间分别降低了 13.6%和 19.8%。

图 5 中的折线图表示 KERBG、KREAG、BLINKS 和 EASE 方法对不同的 k 值，返回 top-k 个查询结果的查询响应时间，该查询响应时间均为对表 1 中 10 个示例的平均查询响应时间。

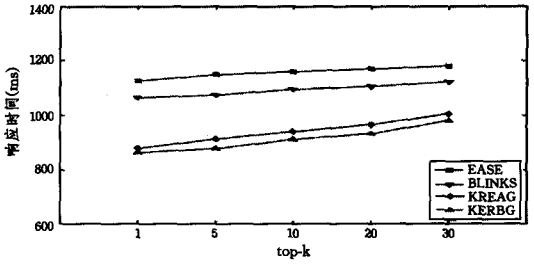

图 5 不同 k 值的响应时间

从图 5 可以看出，随着 k 值的增大，4 种方法的查询响应时间都随之增大。其中，KERBG 和 KREAG 方法的查询响应时间的增长速率几乎一致，BLINKS 和 EASE 方法的查询响应时间增长较为缓慢。但从总体来看，在不同 k 值下，KERBG 方法的查询响应时间仍然最少。

5.3 查询效果的分析比较

本文从查询准确率和平均排序倒数两个方面评价不同方法的查询效果。采用 *Precision@k* 和 *AP@k* 来衡量查询准确率。其中，*Precision@k* 是前 k 个查询结果的查询准确率，*AP@k* 是前 k 个查询结果的加权准确率平均值，以衡量前 k 个查询结果中正确结果的排序情况。采用 *MRR*(Mean Reciprocal Rank)来衡量平均排序倒数，*MRR* 关心的是第一个最相关结果的排序位置。

Precision@k 的计算方式如式(2)所示：

$$Precision@k=\frac{r_k}{k} \tag{2}$$

其中，r_k 表示前 k 个查询结果中正确的结果数。

AP@k 的计算方式如式(3)所示：

$$AP@k=\frac{1}{r_k}\sum_{rank_i\leqslant k}\frac{i}{rank_i} \tag{3}$$

其中，i 表示前 k 个结果中第 i 个正确结果，$rank_i$ 表示第 i 个正确结果的排序。

RR(Reciprocal Rank)的计算方式是第一个最相关结果排序位置的倒数，如果没有返回正确结果，则记为 0。*MRR* 是对测试查询的 *RR* 值求平均。测试中取 $k=5$。

图 6 给出了 4 种方法的 *Precision@k* 值。可以观察，对于查询示例 Q7 和 Q10，KERBG 和 KREAG 方法的查准率相差很大，原因是这两个查询示例包含的实验数据中没有关键词 writer，导致 KREAG 方法的查询结果可能不是最好的，而 KERBG 方法通过关键词扩展技术能够找到 writer 的同义词 author，进而得到最佳查询结果。对于第二组查询，BLINKS

和 EASE 方法由于不能处理包含关系的查询，查准率普遍较低。通过计算图中不同方法对查询示例的平均查准率可知，KERBG 方法较 KREAG 方法其查准率提高了 8.3%，而较 BLINKS 和 EASE 方法查其准率分别提高了 22.6% 和 24.1%。

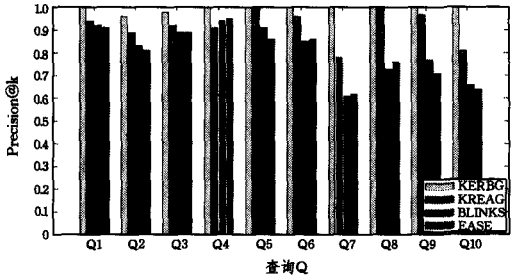

图 6 *Precision@k* 的比较

图 7 给出了 4 种方法的 *AP@k* 值。可以观察，第一组查询的 *AP@k* 值与 *Precision@k* 值相同。对于第二组查询，BLINKS 和 EASE 方法的 *AP@k* 值比对应的 *Precision@k* 值更低，原因是这两种方法不支持对关系的查询，导致查询结果子图中包含的关键词的数目减少，影响查询结果子图评分函数对结果的排序，进而得到更低的 *AP@k* 值。

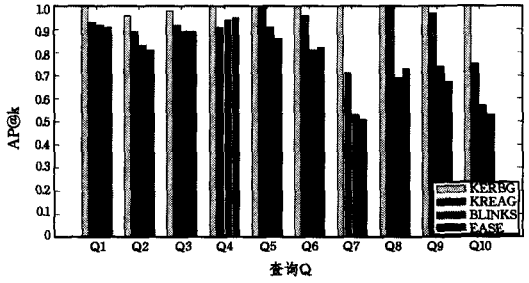

图 7 *AP@k* 的比较

通过表 3 给出的排序倒数指标值可以计算 KERBG 方法的 *MRR* 值为 1，KREAG 方法的 *MRR* 值为 0.98，BLINKS 方法的 *MRR* 值为 0.56，EASE 方法的 *MRR* 值为 0.48。显然 KERBG 和 KREAG 方法基本都能将最相关的查询结果中排序第一的结果返回。

表 3 排序倒数的比较

	排序倒数指标值									
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
KERBG	1	1	1	1	1	1	1	1	1	1
KREAG	1	1	1	1	1/2	1	1	1	1	1
BLINKS	1	1/2	1	1	1/2	1/3	1/4	1/2	1/3	1/5
EASE	1	1/2	1/2	1	1/3	1/2	1/5	1/3	1/4	1/5

结束语 使用图表示 RDF 数据既能保持数据间的关联信息又不丧失语义信息，越来越多的关键词查询方法基于图结构去实现 RDF 数据的查询处理。本文提出的一种基于二分图的 RDF 关键词扩展查询方法(KERBG)以 RDF 二分图为模型，封装文本信息在 RDF 二分图顶点标签上，有效地支持了对 RDF 数据中的实体和关系的查询；利用关键词同义词扩展技术，对查询关键词进行语义扩展，有效地解决了两个人使用同样的关键词描述同一对象的低概率问题，进而提高了查准率；利用 RDF 二分图的反对称邻接矩阵及其幂矩阵构造包含关键顶点的查询结果子图，实现了关键词查询的快速响应，降低了查询响应时间。通过与 KREAG，BLINKS 和 EASE 方法的对比实验表明，该方法在查准率和查询响应时间方面都有很大的改善。在查准率方面，该方法较 KREAG 方法提高了 8.3%，较 BLINKS 和 EASE 方法分别提高了 22.6%和 24.1%；在查询响应时间方面，该方法较 KREAG 方法降低了 1.9%，较 BLINKS 和 EASE 方法分别降低了 13.6%和19.8%。

当用户输入较短的查询而不能完整地表达查询意图时，如何通过关键词扩展技术组成新的、更能准确表达用户查询意图的查询词序列将是我们未来的一个研究工作。另外，当面对海量的 RDF 数据时，如何利用现有的分布式计算框架去并行查询 RDF 二分图将是我们未来的另一个研究工作。

参 考 文 献

[1] W3C:Resource description framework (RDF)[OL]. http://www.w3.org/RDF

[2] Linking open data [EB/OL]. http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpen Data

[3] Du Fang, Chen Yue-guo, Du Xiao-yong. Survey of RDF query processing techniques[J]. Journal of Software, 2013, 24(6): 1222-1242(in Chinese)
杜方，陈跃国，杜小勇. RDF 数据查询处理技术综述[J]. 软件学报，2013,24(6):1222-1242

[4] Marek C, Guy K, Zeev N. Steiner Forest Orientation Problems [J]. SIAM Journal on Discrete Mathematics, 2013, 27(3): 1503-1513

[5] Fischer S, Itoh M, TInagaki M. Screening prototype features in terms of intuitive use: design considerations and proof of concept [J]. Interacting with Computers, 2015, 27(3): 256-270

[6] A M G. WordNet: a lexical database for English [J]. Communications of the ACM, 1995, 38(11): 39-41

[7] Wu Hong-han, Qu Yu-zhong, Li Hui-ying. Searching semantic Web documents based on RDF sentences [J]. Journal of Computer Research and Development, 2010, 47(2): 255-263(in Chinese)
吴鸿汉，瞿裕忠，李慧颖. 基于 RDF 句子的语义网文档搜索[J]. 计算机研究与发展，2010,47(2):255-263

[8] He Hao, Wang Hai-xun, Yang Jun. BLINKS: Ranked keyword searches on graphs [C]// Proc. of the SIGMOD. Beijing, China, 2007: 305-316

[9] Li Guo-liang, Ooi B, Feng Jian-hua, et al. EASE: An effective 3-in-1 keyword search method for unstructured, semi-structured and structured data [C]// Proc. of the International Conference on Management of Data/Principles of Database Systems. Vancouver, BC, Canada, 2008: 903-914

[10] Elbassuoni S, Blanco R. Keyword search over RDF graphs[C]// Proc. of the 20th ACM International Conference on Information and Knowledge Management. ACM, 2011: 237-242

[11] Klara W, Fabian K, Wojciech I, et al. PEST: Fast approximate keyword search in semantic data using eigenvector-based term propagation [J]. Information Systems, 2012, 37(4): 372-390

[12] Xiang L, Eugenio H D, Artem C, et al. k-nearest keyword search in RDF graphs [J]. Journal of Web Semantics, 2013, 22(8): 40-56

[13] Lian X, Chen L, Huang Z. Keyword Search over Probabilistic RDF Graphs[J]. IEEE Trans on Knowledge and Data Engineering, 2014, 27(5): 1246-1260

[14] Zheng Zhi-yun, Liu Bo, Li Lun, et al. Research of Keyword Search Model over RDF Data Graph [J]. Computer Science, 2015, 42(7): 234-239(in Chinese)
郑志蕴, 刘博, 李伦, 等. 基于关键词的 RDF 数据图查询模型研究[J]. 计算机科学, 2015, 42(7): 234-239

[15] Haye J, Gutiërrez C. Bipartite graphs as intermediate model for RDF [C]// Proc. of the 3rd International Semantic Web Conference, Lecture Notes in Computer Science. 2004: 47-61

[16] Li Hui-ying, Qu Yu-zhong. KREAG: Keyword query approach over RDF data based on entity-triple association graph [J]. Chinese Journal of Computers, 2011, 34(5): 825-835(in Chinese)
李慧颖, 瞿裕忠. KREAG: 基于实体三元组关联图的 RDF 数据关键词查询方法[J]. 计算机学报, 2011, 34(5): 825-835)

[17] Jia Shu-fang, Li Lei. Chinese query expansion based on user log clustering [C]// Proc. of IEEE International Conference on Network Infrastructure and Digital Content. 2009: 446-451

[18] Liu C H, Qi R H, Liu Q. Query expansion terms based on positive and negative association rules [C]// Proc. of International Conference on Information Science and Technology. IEEE, 2013: 802-808

[19] Pal D, Mitra M, Datta K. Improving query expansion using WordNet [J]. Journal of the Association for Information Science and Technology, 2014, 65(12): 2469-2478

[20] Semantic Web Technology Evaluation Ontology[OL]. http://lsdis.cs.uga.edu/Projects/SemDis/Swetodblp

(上接第 256 页)

[11] Khan S A, Nadeem A. Automated Test Data Generation for Coupling Based Integration Testing of Object Oriented Programs Using Particle Swarm Optimization (PSO)[C]// Proceedings of the 7th International Conference on Genetic and Evolutionary Computing. Prague, Czech Republic, Springer, 2013: 115-124

[12] Ahmed M A, Hermadi I. GA-based Multiple Paths Test Data Generator [J]. Computers & Operations Research, 2008, 35(10): 3107-3124

[13] Jiang S J, Yi D D, Ju X L, et al. An Approach for Test Data Generation Using Program Slicing and Particle Swarm Optimization [J]. Neural Computing and Applications, 2014, 25(7/8): 2047-2055

[14] Pritanka C, Inderveer C, Ajay R. A Novel Strategy for Automatic Test Data Generation Using Soft Computing Technique[J]. Frontiers of Computer Science, 2015, 9(3): 346-363

[15] Fraser G, Arcuri A. Whole Test Suite Generation [J]. IEEE Transactions on Software Engineering, 2013, 39(2): 276-291

[16] Fraser G, Arcuri A. EvoSuite: Automatic Test Suite Generation for Objected-Oriented Software[C]// Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering. Szeged, Hungray, 2011: 416-419

[17] Arcuri A, Fraser G. On Parameter Tuning in Search Based Software Engineering [C]// Proceedings of the 3rd International Symposium on Search Based Software Engineering (SSBSE). Szeged, Hungary, 2011: 33-47

[18] Fraser G, Arcuri A. The Seed is Strong-Seeding Strategies in Search-Based Software Testing[C]// Proceedings of the 5th International Conference on Software Testing, Verification and Validation (ICST). Montreal, Canada, 2012: 121-130

[19] Mohammad A, Leonardo B. Search-based Software Test Data Generation for String Data Using Program-Specific Search Operators[J]. Software Testing Verification and Reliability, 2006, 16(3): 175-203

[20] Zhao R L. Searc-Based Automatic Path Test Generation for Character String Data [J]. Journal of Computer-Aided Design & Computer Graphics, 2008, 20(5): 671-677(in Chinese)
赵瑞莲. 基于搜索的面向路径字符串测试数据自动生成方法[J]. 计算机辅助设计与图形学学报, 2008, 20(5): 671-677

[21] Phil M, Muzammil S, Mark S. Search-Based Test Input Generation for String Data Types Using the Results of Web Queries [C]// Proceedings of the 5th International Conference on Software Testing, Verification & Validation. Washington DC, USA, IEEE, 2012: 141-150

[22] Zhang X D. Automatic String Test Data Generation Based on Tabu Search [D]. Beijing: Beijing University of Chemical Technology, 2013(in Chinese)
张晓迪. 基于禁忌搜索的字符串型测试数据自动生成[D]. 北京: 北京化工大学, 2013

[23] Kim S H, Kim K, Cho H G. A New String Search Tree with Multiple Alignment[C]// Proceedings of the 12th International Conference on Computer and Information Technology. Chengdu, China, 2012: 456-463

[24] Sheeva A, Phil M, Mark S. Evolving Readable String Test Inputs Using a Natural Language Model to Reduce Human Oracle Cost[C]// Proceedings of the 6th International Conference on Software Testing, Verification and Validation. Luembourg, 2013: 352-361

[25] Mao C Y, Yu X X, Xue Y Z. Algorithm Design and Empirical Analysis for Particle Swarm Optimization-Based Test Data Generation [J]. Journal of Computer Research and Development, 2014, 51(4): 824-837(in Chinese)
毛澄映, 喻新欣, 薛云志. 基于粒子群优化的测试数据生成及其实证分析[J]. 计算机研究与发展, 2014, 51(4): 824-837

[26] Whitley D. The GENITOR Algorithm and Selective Pressure: Why Rank Based Allocation of Reproductive Trials is Best[C]// Proceedings of the 3rd International Conference on Genetic Algorithms, Fairfax, VA. San Francisco, USA, Morgan Kaufmann, 1989: 116-121

[27] Huang J C. Program Instrumentation and Software Testing[J]. Computer Journal, 1978, 11(4): 25-32

[28] Gonzalo N. A Guided Tour to Approximate String Matching [J]. ACM Computing Surveys, 2001, 33(1): 31-88