RDF Keyword Search by Query Computation

Zongmin Ma, Nanjing University of Aeronautics and Astronautics, Nanjing, China
https://orcid.org/0000-0001-7780-6473
Xiaoqing Lin, Eastern Liaoning University, Dandong, China
Li Yan, Nanjing University of Aeronautics and Astronautics, Nanjing, China
Zhen Zhao, Bohai University, Jinzhou, China

ABSTRACT

Keyword searches based on the keywords-to-SPARQL translation is attracting more attention because of a growing number of excellent SPARQL search engines. Current approaches for keyword search based on the keywords-to-SPARQL translation suffer from returning incomplete answers or wrong answers due to a lack of underlying schema information. To overcome these difficulties, in this article, we propose a new keyword search paradigm by translating keyword queries into SPARQL queries for exploring RDF data. An inter-entity relationship summary with complete schema information is distilled from the RDF data graph for composing SPARQL queries. To avoid potentially wasteful summary graph expansion, we develop a new search prioritization scheme by combining the degree of a vertex with the distance from the original keyword element. Starting from the ordered priority list that is built in advance, we apply the forward path index to faster find the top-k subgraphs, which are relevant to the conjunction of the entering keywords. The experimental results show that our approach is efficient and scalable.

KEYWORDS
Inter-Entity Relationship Summary, Keyword Search, Property Paths, SPARQL, Subgraph Search

INTRODUCTION

RDF (Resource Description Framework) is a standard model for data interchange on the Web. By this model, more and more structured and semi-structured data have been mixed, exposed, and shared across different applications (Klyne, Carroll, and McBride, 2004). As a result, available RDF data rapidly increases. RDF data is a collection of triples, the form (subject, predicate, object). Such a collection of triples can be represented as a directed graph, in which vertices represent subjects or objects, and edges represent predicates that connect subjects and objects. Several important issues of RDF data management are recently reviewed in (Ma, Capretz, Miriam, and Yan, 2016; Wylot, DOI: 10.4018/JDM.2018100101

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
Hauswirth, Philippe, and Sakr, 2018), including RDF data storage techniques, indexing strategies, and query execution mechanisms.

Since SPARQL (Prud’hommeaux and Seaborne, 2013) has been recommended to be the standard query language for RDF data by the World Wide Web Consortium (W3C), there has been a rapid increase in the number of users who want to access RDF data (Yan, Ma, Li, and Cheng, 2017). SPARQL allows the specification of triple and graph patterns to be matched over RDF data graphs (Ma, Jia, Cheng, and AngryK, 2016). And then we can access RDF data correctly and efficiently by SPARQL. But it is still infeasible for non-expert users to master the RDF schema and SPARQL query language. Let us look at an example of SPARQL query for the DBLP dataset shown in Example 1 by the data in Figure 2.

Example 1: Find the published articles and proceedings by author “Coles:Drue” for the DBLP dataset.

```
SELECT ?x ?y
WHERE {?x isIncludedIn ?y.
  ?x type Article_in_Proceedings.
  ?y type Proceedings.
  ?x author "Coles:Drue".}
```

It is shown that, to construct a SPARQL query like Example 1, non-expert users are required not only to master SPARQL but also to know the schema information of RDF data such as “isIncludedIn”, “Article in Proceedings” and “Proceedings”. So, it is not easy for non-expert users to construct SPARQL queries. At this point, keyword search has been a popular tool for exploring RDF data for non-expert users (Izquierdo et al., 2018; García, Izquierdo, Menendez, Dartayre, and Marco, 2017). Users only need to enter keywords and then top-k query results answered can be directly returned to the user. Currently, the number of excellent SPARQL search engines is growing rapidly (Broekstra, Kampman, and Harmelen, 2002; Garrison, Stevens, and Jocuns, 2004; Neumann and Weikum, 2008; Neumann and Weikum, 2010). To make it easy for non-expert users to compose SPARQL queries, in this paper, we concentrate on the approach for translating keyword queries into SPARQL queries. As mentioned in (Gkirtzou, Papastefanatos, and Dalamagas, 2015; Ladwig and Tran, 2010; Tran, Wang, Rudolph, and Cimiano, 2009; Lin, Ma, and Yan, 2018; Wen, Jin, and Yuan, 2018), keyword search based on translation has its advantages. It can provide users with a friendly interface for querying RDF data on the one hand and, on the other hand, we can obtain a better query performance by using the existing SPARQL search engines on the condition of guaranteeing the correctness.

Generally speaking, two categories of keyword searches can be identified according to different query processing ways. The first one aims to find smaller substructures that contain all keywords over the original RDF datasets. The substructures, which can be graphs (Li, Ooi, Feng, Wang, and Zhou, 2008; Sinha, Lu, and Theodoratos, 2010; Han, Zou, Yu, and Zhao, 2017), cliques (Kargar and An, 2011), trees (Chen, Wang, Liu, and Lin, 2009; Fakas, 2011; He, Wang, Yang, and Yu, 2007; Kacholia, Pandit, Chakrabarti, Sudarshan, and Karambelkar, 2005; Hulgeri and Nakhe, 2002; Kasnei, Ramanath, Sozio, Suchanek, and Weikum, 2009; Le, Li, Kementsietsidis, and Duan, 2014; Liu, Yu, Meng, and Chowdhury, 2006; Sun, Chan, and Goenka, 2007; Wang, Zou, Pan, and Zhao, 2012) and other patterns defined over RDF graphs (Dass, Aksoy, Dimitriou, Theodoratos, and Wu, 2016; Li, Yan, and Ma, 2019; Peng, Zou, and Qin, 2017), are computed by a scoring mechanism. A semantic search engine is developed in (Taha and Elmasri, 2009) to answer XML keyword-based queries as well as loosely structured queries using object-oriented techniques. In (Lian, Hoyos, Chebotko, Fu, and Reilly, 2013), k-nearest keyword search in RDF graphs is investigated by utilizing an index for k closest pairs of vertices to prune search space. The second one interprets keyword queries into SPARQL queries (Fu, Gao, and Anyanwu, 2011; Gkirtzou et al., 2015; Ladwig et al., 2010; Tran et al., 2009; Zenz, Zhou, Minack, Siberski, and Nejdli, 2009; Zhou, Wang, Xiong, Wang, and Yu, 2007).
25 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product's webpage:

www.igi-global.com/article/rdf-keyword-search-by-query-computation/227035?camid=4v1


www.igi-global.com/e-resources/library-recommendation/?id=154

Related Content

A Paradigm For Natural Language Explanation Of Database Queries: A Semantic Data Model Approach

www.igi-global.com/article/paradigm-natural-language-explanation-database/51129?camid=4v1a

Information Mediation Using Metamodels: An Approach Using XML and Common Warehouse Metamodel

www.igi-global.com/article/information-mediation-using-metamodels/3375?camid=4v1a
Leveraging Early Aspects in End-to-End Model Driven Development for Non-Functional Properties in Service Oriented Architecture
www.igi-global.com/article/leveraging-early-aspects-end-end/52994?camid=4v1a

Fuzzy Decision Rule Construction Using Fuzzy Decision Trees: Application to E-Learning Database
www.igi-global.com/chapter/fuzzy-decision-rule-construction-using/44385?camid=4v1a