Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 5.258

IJCSMC, Vol. 5, Issue. 9, September 2016, pg.121 – 125

A SPARQL QUERY GENERATION MECHANISM ON RDF

Ms. Rashmi Dudhani[1], Mrs. Clara Kanmani[2]

[1]PG Student, Department of Computer Science and Engineering, New Horizon College of Engineering, Bangalore, Karnataka, India
[2]Senior Assistant Professor, Department of Computer Science and Engineering, New Horizon College of Engineering, Bangalore, Karnataka, India
[1] rashmidudhni@gmail.com [2] claracalton.joseph@gmail.com

Abstract— Generally most people use keyword queries to fulfil their information needs on the Web. Semantic web makes a transformation to a format which is machine readable. RDF is the common format used in the Semantic Web to store data. SPARQL queries are able to retrieve more relevant results than general keyword routing on RDF data. RDF federations are capable of connecting multiple different kinds of data sources. Therefore an approach is introduced which can convert the keyword queries to SPARQL queries on a federation of different sources. Emerging semantic web technologies would be lots of information which is more structured and can be available which can logically queries for accurate search, but there is much complexity involved in current semantic query languages like RQL, SPARQL, RDQL. There's a critical gap between the semantic search and end users .To reduce this complexity a hybrid approach to retrieve information from semantic datasets using traditional keyword based queries by transforming them to semantic query for SPARQL. In this approach traditional IR techniques can be applied to discover URIs of respective keyword query terms. Using this URIs one can formulate SPARQL queries to semantically retrieve queried information by combining keyword based search with semantic search. To achieve this functionality it is necessary to develop a distributed knowledge base which maps set of keywords to URIs from different datasets. The algorithm to populate this knowledge base is a challenging work given variety and ambiguity in current datasets. Our objective is to develop a prototype system which given a keyword based formal logic query, outputs SPARQL query using above suggested knowledge base. Which in order can be executed against datasets.

Keywords— Semantic web, RDF, SPARQL, Path Indexing, RDF Federation.

I. INTRODUCTION

Now days World Wide Web (WWW) has become essential to everyone for retrieving relevant information efficiently. Semantic Web was introduced as a method of storing web data in a machines understandable way. Resource Description Framework2 (RDF) was presented as the standard format for storing and exchanging data. RDF preserves the interconnections among data elements and use graph structure for storage. These relationships are essential when recognizing the relevancy of data for a user query. Recently many researchers and academic institutes have taken the initiative of exposing their data to the Web in RDF format. SPARQL3 is the query language for RDF data.

© 2016, IJCSMC All Rights Reserved

Structured queries are capable of retrieving more relevant results efficiently and accurately compared to keyword queries. However composing structured queries require expertise knowledge which is lacking among general users. SPARQL is capable of retrieving more relevant results from web data. Therefore transforming user friendly keyword queries to SPARQL allows general users to retrieve highly relevant results without having knowledge about underlying complexities.

The process of translating keyword queries to SPARQL can be decomposed to following steps. 1) Mapping user keywords to data elements. 2) Identifying sub-graphs which can connect mapped data elements. 3) Generating queries based on the relationships in the sub-graphs. Most of the available approaches exploit graph traversal in real time for identifying suitable sub-graphs. Only limited set of functions are carried out as pre-processing. Most of these approaches provide approximate results because traversing RDF graphs with millions or billions of data is very expensive and highly time consuming. Hence there is a requirement to seek for approaches which can reduce graph traversal in query generating time.

There are many contributors of the WWW who provides information on related topics individually. For an example, DBLP and ACM contain academic publication data individually. None of them have entire publication data. In contrast, Google Scholar connects sources such as DBLP and ACM to provide more complete set of results for the public. Therefore it is more useful for retrieving publication details.

Most RDF sources are also maintained as individual dumps. More accurate and complete results can be easily obtained by combining those together. RDF federations have been presented as a solution for this problem. Yet federations only accept SPARQL queries. This research focuses on transforming keyword queries to SPARQL on a RDF federation in order to allow general users to access Semantic Web and fulfil their information needs

Searching Techniques:

There are various searching techniques available, and to mention a few are as follows:

- 2012In this study, an approach enabling keyword-based semantic information query over RDF data is proposed. The approach sets up a keyword-inverted index and a relation index based on the graph and searches the connecting nodes to provide an answer for keyword query. Moreover, the approach uses an improved scoring function based on textual relevancy and relation popularity and supports top-k queries. Here by considering the radius graph as the basic unit, obtain -radius graphs containing all input keywords as candidate answers, and define a subgraph (tree) of candidate answer as the answer. An algorithm for subgraph exploration is proposed. For rapid searching,a keyword-inverted index and a relation index is being used. The time and space complexity to set up this index scheme is acceptable. The results show that our approach achieves high accuracy. Experimental results show that the proposed approach can achieve good query performance

- Recent approaches for RDF databases try to improve the quality of results by introducing an explicit top-k "interpretation" phase in which queries are translated into an ordered list of "most likely intended" structured (SPARQL) queries before query execution. However, even these recent techniques only address keyword query ambiguity in a limited fashion by identifying fine-grained semantic units or segments of a query. This enables some reduction in the space of interpretations, pruning away incorrect interpretations, but the reduction in interpretation space is not as aggressive as it could be. In this paper, we propose a "deep segmentation" technique for keyword queries issued against an RDF database. This approach achieves a more aggressive pruning of irrelevant interpretations from the space of interpretations considered and therefore produces better quality query interpretations even in the presence of significant query ambiguity. We present results for a comprehensive human-based evaluation that is based on a metric that we introduce called degree of ambiguity (DOTA) that has not been considered by previous efforts. The experimental results show that our approach outperforms existing techniques in terms of quality even when queries are very ambiguous.

- It utilizes a set of predefined basic graph pattern templates for generating adequate interpretations of user queries. This is achieved by obtaining ranked lists of candidate resource identifiers for the supplied keywords and then injecting these identifiers into suitable positions in the graph pattern templates. The main advantages of our approach are that it is completely agnostic of the underlying knowledge base and ontology schema, that it scales to large knowledge bases and is simple to use. By evaluating 17 possible valid graph pattern templates by measuring their precision and recall on 53 queries

© 2016, IJCSMC All Rights Reserved

against DBpedia. Our results show that 8 of these basic graph pattern templates return results with a precision above 70%. Our approach is implemented as a Web search interface and performs sufficiently fast to return instant answers to the user even with large knowledge bases..

- In this paper it refers the method to retrieve the web data in structured format using MashQL approach with the help of SPARQL language which allows user to query, navigate, and mash-up a data source(s) without knowing schema and how the data is stored into the database. Assuming web data as input in RDF format used to represent the metadata of the Web applications in structured manner. By using SPARQL language which is the recent recommendation by the W3C.By using this approach we mentioned how user having less technical knowledge is able to retrieve the data in structured format using drop down operations and interactive methods. Here it is mentioned that how retrieval results will be faster by providing keyword search and then use MashQL approach and also mentioned an aggregate function (e.g. sum, avg, max etc) in MashQL so user can perform a calculation on a set of values and return a single value

II. LITERATURE SURVEY

Even though there is no existing approach for transforming keyword queries to SPARQL on RDF federations, research have taken place in addressing each step required for keyword to SPARQL transformation on RDF federations. They are as follows; resolving vocabulary level heterogeneity, mapping user keywords to data source elements and identifying suitable sub-graphs connecting keyword elements.

- L. G. V. Hristidis and Y. Papakonstantinou, "Efficient ir-style keyword search over relational databases,".
  Applications in which plain text coexists with structured data are pervasive. Commercial relational database management systems (RDBMSs) generally provide querying capabilities for text attributes that incorporate state-of-the-art information retrieval (IR) relevance ranking strategies, but this search functionality requires that queries specify the exact column or columns against which a given list of keywords is to be matched.

- W. M. F. Liu, C. T. Yu, and A Chowdhury, "Effective keyword search in relational databases,"
  With the amount of available text data in relational databases growing rapidly, the need for ordinary users to search such information is dramatically increasing. Even though the major RDBMSs have provided full-text search capabilities, they still require users to have knowledge of the database schemas and use a structured query language to search information. This search model is complicated for most ordinary users. Inspired by the big success of information retrieval (IR) style keyword search on the web, keyword search in relational databases has recently emerged as a new research topic. The differences between text databases and relational databases result in three new challenges: (1) Answers needed by users are not limited to individual tuples,. (2) A single score for each answer (i.e. a tuple tree) is needed to estimate its relevance to a given query. (3) Relational databases have much richer structures than text databases. In this paper,a novel IR ranking strategy for effective keyword search.. Experimental results show that our strategy is significantly better than existing strategies. Our approach can be used both at the application level and be incorporated into a RDBMS to support keyword-based search in relational databases.

- B. Kimelfeld and Y. Sagiv, "Finding and approximating top-k answers in keyword proximity search,
  Various approaches for keyword proximity search have been implemented in relational databases, XML and the Web. Yet, in all of them, an answer is a Q-fragment, namely, a subtree T of the given data graph G, such that T contains all the keywords of the query Q and has no proper subtree with this property. The rank of an answer is inversely proportional to its weight. Three problems are of interest: finding an optimal (i.e., top-ranked) answer, computing the top-k answers and enumerating all the answers in ranked order.. First, enumerating in a (θ+1)-approximate order. Second, computing a (θ+1)-approximation of the top-k answers. As a corollary, this paper gives the first efficient algorithms, under data complexity, for enumerating all the answers in ranked order and for computing the top-k answers. It also gives the first efficient algorithms, under query-and-data complexity, for enumerating in a provably approximate order and for computing an approximation of the top-k answers.

- H. He, H. Wang, 1. Yang, and P.S. Yu, "BLINKS : Ranked Keyword Searches on Graphs,"
  Query processing over graph-structured data is enjoying a growing number of applications. A top-k keyword search query on a graph finds the top k answers according to some ranking criteria, where each answer is a substructure of the graph

© 2016, IJCSMC All Rights Reserved

containing all query keywords. Current techniques for supporting such queries on general graphs suffer from several drawbacks, e.g., poor worst-case performance, not taking full advantage of indexes, and high memory requirements. To address these problems, we propose BLINKS, a bi-level indexing and query processing scheme for top-k keyword search on graphs. BLINKS follows a search strategy with provable performance bounds, while additionally exploiting a bi-level index for pruning and accelerating the search. To reduce the index space, BLINKS partitions a data graph into blocks: The bi-level index stores summary information at the block level to initiate and guide search among blocks, and more detailed information for each block to accelerate search within blocks. Our experiments show that BLINKS offers orders-of-magnitude performance improvement over existing approaches.

- S. C. S. S. R. D. V. Kacholia, S. Pandit, and H. Karambelkar, "Bidirectional expansion for keyword search on graph databases
  Relational, XML and HTML data can be represented as graphs with entities as nodes and relationships as edges. Text is associated with nodes and possibly edges. Keyword search on such graphs has received much attention lately. A central problem in this scenario is to efficiently extract from the data graph a small number of the "best" answer trees. A Backward Expanding search, starting at nodes matching keywords and working up toward confluent roots, is commonly used for predominantly text-driven queries. But it can perform poorly if some keywords match many nodes, or some node has very large degree. In this paper we propose a new search algorithm, Bidirectional Search, which improves on Backward Expanding search by allowing forward search from potential roots towards leaves. To exploit this flexibility, we devise a novel search frontier prioritization technique based on spreading activation. We present a performance study on real data, establishing that Bidirectional Search significantly outperforms Backward Expanding search.

- P.Cappellari, RDe Virgilio, AMaccioni, and M.Roantree, "A Path-Oriented RDF Index for Keyword Search Query Processing,"
  Most of the recent approaches to keyword search employ graph structured representation of data. Answers to queries are generally sub-structures of the graph, containing one or more keywords. While finding the nodes matching keywords is relatively easy, determining the connections between such nodes is a complex problem requiring on-the-fly time consuming graph exploration. Current techniques suffer from poorly performing worst case scenario or from indexing schemes that provide little support to the discovery of connections between nodes.
  In this paper, we present an indexing scheme for RDF that exposes the structural characteristics of the graph, its paths and the information on the reachability of nodes. This knowledge is exploited to expedite the retrieval of the substructures representing the query results. In addition, the index is organized to facilitate maintenance operations as the dataset evolves. Experimental results demonstrates the feasibility of our index that significantly improves the query execution performance.

III. CONCLUSION

By presenting a keyword query to SPARQL query conversion mechanism approach on RDF data source federations by utilizing a keyword index along with vocabulary level heterogeneity resolution to identify matching elements for user keywords.A Path Index based approach was used for identify suitable sub-graphs for connecting keyword elements. This approach has totally eliminated real time graph traversal for query generation and showed a significant performance gain over existing query generation approach and were able achieve a promising level of results for the quality evaluation of this approach and also showed that federations are capable of giving more complete results for a user query than just querying from a single data source and came to a conclusion that Semantic Web related keyword query approaches can give more relevant results for user queries than traditional keyword matching.

REFERENCES

[1] L. G. V. Hristidis and Y. Papakonstantinou, "Efficient ir-style keyword search over relational databases," in VLDB, pp. 850-861, 2003.
[2] V. H. H. H wang and Y. Papak on stant in ou, "Object rank: a system for authority based search on databases," in SIGMOD Conference, pp. 796-798, 2006.

© 2016, IJCSMC All Rights Reserved

[3] W. M. F. Liu, C. T. Yu, and A Chowdhury, "Effective keyword search in relational databases," in SlGMOD Conference, pp. 563-574, 2006.
[4] B. Kimelfeld and Y. Sagiv, "Finding and approximating top-k answers in keyword proximity search," in PODS, pp. 173-182, 2006.
[5] H. He, H. Wang, 1. Yang, and P.S. Yu, "BLINKS : Ranked Keyword Searches on Graphs," in Proceedings of the 2007 SlGMOD lnternational Conference on Management of Data, ACM, 2007.
[6] S. C. S. S. R. D. V. Kacholia, S. Pandit, and H. Karambelkar, "Bidirectional expansion for keyword search on graph databases," in VLDB, pp. 505-516, 2005.
[7] P.Cappellari, RDe Virgilio, AMaccioni, and M.Roantree, "A Path-Oriented RDF Index for Keyword Search Query Processing," in DEXA, pp. 366-380,2011.
[8] S. R. T. Tran, H. Wang, and P. Cimiano, 'Top-k exploration of query candidates for efficient keyword search on graph-shaped (rdf) data," in lCDE,lEEE, 2009.
[9] C. N. S. C. G. Bhalotia, A Hulgeri, and S. Sudarshan, "Keyword searching and browsing in databases using banks," in lCDE, pp. 431-440,2002.
[10] T.Tran and L.Zhang, "Keyword Query Routing," lEEE Transactions on Knowledge and Data Engineering, vol. I, no. 2, 2013.

© 2016, IJCSMC All Rights Reserved