Tikrit Journal of Pure Science (2023) 28(5):193-200
Doi: https://doi.org/10.25130/tjps.v28i5.1582

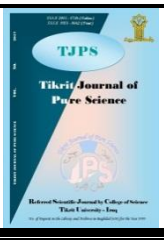
TJPS

IASJ

المجلات الاكاديمية العراقية

Tikrit Journal of Pure Science
ISSN: 1813 – 1662 (Print) --- E-ISSN: 2415 – 1726 (Online)

Journal Homepage: http://tjps.tu.edu.iq/index.php/j



Resource Description Framework Query Languages: Comparative Study

Ali Mustafa Ali Alshaykha[1], Saadi Hamad Thalij[2], Orhan Nooruldeen[3]
[1] College of Engineering / Shirqat, Tikrit University
[2]Department of Computer Science, Collage of Computer & Math, Tikrit University
[3]Department of Computer Technology Engineering, College of Information Technology, Imam Ja'afer Al-Sadiq University

Keywords: extended markup language (XML), uniform resource identifier (URI), Resource Description Framework RDF.

ARTICLEINFO.

Article history:

-Received:	**15 Dec. 2023**
-Received in revised form:	**20 Jan 2023**
-Accepted:	**22 Jan. 2023**
-Final Proofreading:	**24 Oct. 2023**
-Available online:	**25 Oct. 2023**

Corresponding Author*:
Ali Mustafa Ali Alshaykha
eng.ali78@tu.edu.iq

ABSTRACT
Over the period of time, the data on the web grows by leaps and bounds that becomes unmanageable to store and retrieve. As a result, the concept of semantic web comes into existence. Nowadays, various components of semantic web stack become mature such as Resource Description Framework (RDF). Due to the higher adoption rate of RDF in both industry and academia, it becomes crucial to design such systems that can store and retrieve RDF formatted data. Since last decade, various query languages for RDF data retrieval were proposed mainly based on either graph query language or on relational algebra. However, none of them become mature enough that can overcome majority of challenges. Therefore, there is an utmost need for the comparative study of available languages in literature. In this paper, a comparative study based on various parameters was conducted that provides a better insight about the literature.

Tikrit Journal of Pure Science (2023) 28(5):193-200
Doi: https://doi.org/10.25130/tjps.v28i5.1582

TJPS

الملخص

على مدار الوقت، تنمو البيانات على الشبكة العنكبوتية بسرعة فائقة ويصبح من الصعب ادارتها وتخزينها واستدعائها. ونتيجة لذلك ظهر مفهوم الويب الدلالي وفي الوقت الحالي أصبحت المكونات المختلفة لمكدس الويب الدلالي ناضجة ومتطورة مثل أطار وصف الموارد، ونظرا لارتفاع نسبة تبني إطار وصف البيانات كل مجالات الصناعة والاوساط الاكاديمية، يصبح من الاهمية تصميم مثل هذه الانظمة التي يمكنها تخزين واسترجاع البيانات بتنسيق إطار وصف البيانات.

منذ العقد الماضي، تم اقتراح لغات استعلام مختلفة لغرض استرجاع البيانات والخاصة بإطار وصف البيانات بناءا على لغة استعلام الرسم البياني أو الجبر العلائقي.

ومع ذلك، لم يصبح أي منهم ناضجا بما يكفي للتغلب على غالبية التحديات. لذلك، هنالك حاجة ماسة لدراسة المقارنة بين اللغات المتاحة لغرض تسهيل الاختيار للغة المتاحة واستخدامها في المجال المحدد حسب نتائج امكانياتها المدروسة.

وفي هذا البحث، تم أجراء دراسة مقارنة تستند على معايير مختلفة توفر رؤية أفضل حول المراجع وبذلك يكون من السهل اعتماد اللغة البرمجية المناسبة اعتمادا على المعايير التي طبقت في دراسة المقارنة.

1. Introduction

RDF framework is the integration of two utmost technologies of W3C viz. extended markup language (XML) and uniform resource identifier (URI). XML is used to store RDF statements containing a "triple" in the form of <URI, property name, property value>. The first part represents a "subject" followed by the "predicate" and the last represents an "object". RDF was designed for storing data in a distributed environment. Although, due to its representation as a binary relation (i.e., a R b) almost every field can admire its benefit. Consequently, there is an utmost requirement of a query language that can effectively retrieve RDF data and fulfils user's need [1].

Since last decade, various query languages for RDF data retrieval were proposed mainly based on either graph query language or on relational algebra. However, "Simple Protocol and RDF Query Language" (SPARQL) becomes the standard query language for RDF triple retrieval. The model of RDF databases is very similar to graph databases that might be the reason behind generally citing RDF by various researchers as one of the key utilizations of diagram databases. The basic concept is a triple (subject, predicate, object), drawn from a space of uniform asset identifiers (URI's). Hence, the \"predicate\" a bit much originate from a limited arrangement of letters in order, and may furthermore assume the job of a subject or an item in another triple. For instance, {(subject1, predicate1, object1), (subject1, predicate1, object2), (subject1, predicate2, object1)} is a valid set of RDF triples, but in graph databases, it is impossible to have such edges. Hence, there were various issues found related to query evaluation over graphs [1]. Due to this reason, there is utmost requirement to compare various available RDF query languages in different parameters. So that, the pros and cons of every technique should be known in order to enhance the retrieval mechanism while reducing the challenges of retrieval [2].

Organization In section 2, a review of different RDF query language techniques was discussed. Section 3 describes about the evaluation framework for the comparison along with providing an insight about different characteristics of a query language. In section 4, discussion about different query languages that were selected for the comparison along with providing a comparative study of selected languages on the basis of evaluated framework. Finally, conclude the paper with main highlights [2].

Natural language questions/answers about RDF (Resource Description Framework) data have attracted widespread attention. Although several studies can handle a small

number of aggregated queries, these studies have many limitations (e.g., interactive information, controlled questions, or query templates). Until now, there hasn't been a natural language query mechanism that can handle general aggregated queries over RDF data. Therefore, we propose a framework called NLAQ (Natural Language Aggregate Query). First, we propose a novel algorithm to automatically understand the intent of a user query that mainly contains semantic relationships and aggregations. Second, to build a better bridge between the query intent and the RDF data, we propose an extended paraphrase dictionary ED to get more candidate assignments for semantic relations, and we introduce a predicate type neighboring set PT to include inappropriate candidate assignment combinations in the semantics to filter out relationships and basic diagram patterns. Third, we design an appropriate translation plan for each aggregated category, and effectively discriminate whether an aggregated item is numeric, which will greatly affect the aggregated result. Finally, we conduct extensive experiments with real datasets (QALD benchmark and DBpedia). The experimental results show that our solution is effective [3].

RDF Query Language

RDF databases contain triples increases in which the center component not really a piece of a fixed arrangement of marks. Officially, in the event that U is a space of uniform asset identifiers (URI's), at that point an RDF triple (subject, predicate, object) is a subset of U × U × U.

Graph Based Query Language

For defining graph-based query language, first of all we have to explain graph databases. So, A diagram database is only a limited edge-named chart in which every hub has an information esteem joined. Officially, let N be a set of nodes, A is a finite alphabet and D a set of data values. Then a graph database over A is a triple G = (V, E, A), where V is a limited arrangement of nodes, E is a lot of named edges, and A: V → D is a capacity allotting an information incentive to every hub. Each edge is a triple (u, a, v), whose translation is an a-named edge from u to v [4].

Regular path queries (RPQ): Common navigational dialects for diagram databases utilize customary way questions as the fundamental structure obstruct that discover hubs reachable by way whose name has a place with an ordinary language. An RPQ is an expression Subject $\xrightarrow{L}$ Object, where L is a regular language. Given a graph database G = (V, E) over some alphabet, it defines pairs of nodes (u, v) such that there is a path from u to v, which is a subset of L.

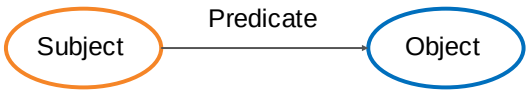


Figure 1: Regular Path

Nested regular expressions (NRE's): These expressions over a limited letters in order, broaden conventional ordinary articulations with the settling administrator and inverses [4].

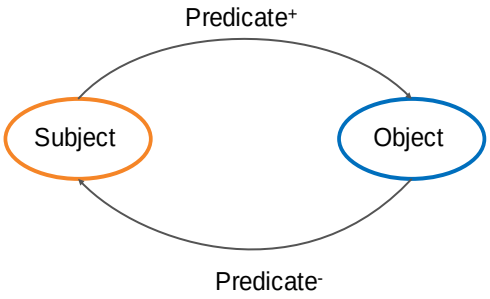


Figure 2: Inverse Navigation

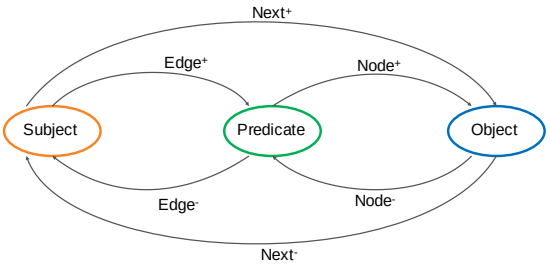


Figure 3: Nested & Inverse Navigation

Relational Algebra Based Query Language

The tasks of the typical social variable-based math are determination, projection, association, distinction, and Cartesian product. A query language that can evaluate a query by performing these operations is based on relational algebra such as SQL [6].

Evaluation Framework

This section introduces the evaluation framework adopted for the qualitative analysis of a set of well-known RDF Query languages. The overall aim of this study is to find an effective RDF language that can fulfils the user's need along with fulfilling the characteristics of an efficient query language. Some of the utmost characteristics of a query language are closure, reachability, complexity, semantic and expressiveness.

Closure

A critical property of a question language is the conclusion, for example inquiries should return objects of a similar kind as their input. In other words, queries should return objects in triple format. Therefore, closed languages are compositional and their operators can be applied to results of queries [6].

Reachability

Another key property of any query language is reachability: retrieval engine should explore all the possible options desired for the input query. i.e., no such data left by the retrieval engine that can be a part of the output [6].

Simplicity

Simplicity describes the knowledge independence of the user about the specific query language. i.e., whether a user can easily adopt the querying pattern or a training about a query language is required [6].

Semantic Schema Support

Language should support a semantic that clearly defines the retrieval of RDF resources and allows the user for its usage in the querying environment [4].

Active Schema Catalog

Users must be able to access the database's structure or a catalog using the same query language that they use to access the database's data [7].

Expressiveness

Expressiveness indicates how powerful queries can be formulated in a given language. Typically, a language should at least provide the means offered by relational algebra, i.e., be relationally complete. Usually, expressiveness is restricted to maintain other properties such as safety and to allow an efficient execution of queries [7].

Analysis of RDF Query Languages

This section describes about the steps taken for evaluation purpose. Firstly, RDF query languages were selected on criteria of retrieval paradigms, standards and their proposed dates. Then, the comparative study on the basis of above characteristics was performed.

RDF Query Languages to be Evaluated

SPARQL

SPARQL is an RDF query language abbreviated as "Simple Protocol and RDF Query Language". It allows retrieval of triples from an RDF database (or triple store). Also, its query pattern resembles the Structured Query Language (SQL). SPARQL becomes a standard query language in 2008 by the recommendation of W3C [4].

nSPARQL

nSPARQL was defined on top of SPARQL; it gives navigational system by utilizing settled ordinary articulations. These are basically customary way questions with XPath-roused hub tests. The evaluation of those uses essentially a graph encoding of RDF [5].

SPARQL2NL

SPARQL2NL was also designed on top of the standard SPARQL. Apart from other query languages, it did not enhance anything technically. Although it provides a better understanding of a query by converting RDF Query into natural language. Consequently, user enables to enter more appropriate query and hence result of a query enhances [6].

LITEQ

LITEQ is designed on top of "Node Path Query Language" (NPQL) i.e., a regular path query language. It provides a better programming interface to the programmer by providing the available schema information from the RDF data source [7].

CPSPARQL

CPSPARQL was defined on top of PSPARQL; it is also a graph-based query language. Enhancing the previously designed PSPARQL by overcoming the challenge of insufficient capability of answering PSPARQL queries modulo RDFS by using constrained regular expression [8].

TRIAL*

Trial* is a query language based on relational algebra and abbreviated as "Triple Algebra". It uses composition, selection, union and a conditional triple join while excluding the projection from the language operations. Its recursive triple algebra enhances the reachability along with ensuring the closure under triples [9].

Comparison of RDF Query Languages

This subsection presents the consequences of the examination dependent on the recently received system and did on the arrangement of preselected RDF Query languages. To facilitate the analysis, a synopsis of main outcomes of this study is reported in Table 1.

In the following each language is presented, in light of criteria included in the adopted framework, for providing the better assessment of the selected RDF query languages.

SPARQL is an RDF query language that represents an RDF triples into an edge-labeled directed graph. For retrieval, it can process the triples using basic graph pattern and search the matching graph pattern similar to the queried graph pattern. The number of matches found in the dataset was returned as the resulted graph. Consequently, it is closed under graph but not under the triple. Apparently, it uses the RPQs containing predicate as a labeled edge

with a restriction of association with a finite set of alphabets. This limits the reachability of the language as the triple not necessarily fulfils this constraint. i.e., in a triple store, predicate of a one triple might be a subject of another triple. As a result, a low reachability was found in SPARQL. Furthermore, SPARQL is a standard query language that was designed to facilitate the user or specifically the query programmer. Therefore, the query pattern is very much similar to structured query language (SQL) without the support of an RDF schema. Hence, novice has to suffer a bit while using SPARQL. Philosophically, knowledge about the context is the essential constrain to express something. Similarly, the reachability is the required constrained for expressiveness. Therefore, As SPARQL have a low reachability, complex queries cannot be expressed in SPARQL.

nSPARQL is another RDF query language that also represents a RDF triples into an edge-labeled directed graph. nSPARQL is a variant of SPARQL that provides a navigation support using the NREs. For retrieval, it can process the triples using NREs and search the matching graph pattern similar to the queried graph pattern. The number of matches found in the dataset was returned as the resulted graph. Consequently, it is closed under graph but not under the triple. Apparently, it uses the NREs containing inverse and navigational operators to provide higher reachability to SPARQL query. This approach enhances the reachability of the language but still leave some patterns to reach, as discussed by Leonid [8]. As a result, a medium reachability was found in nSPARQL. Furthermore, nSPARQL is similar to SPARQL in every other perspective. Therefore, the query pattern is similar to SPARQL. Hence, the similar issue has to face by novice while using nSPARQL. However, a navigation pattern enhances the reachability pattern in nSPARQL for that reason expressiveness of nSPARQL also enhances.

SPARQL2NL is another variant of SPARQL. It is a RDF query language that represents a RDF triples into an edge-labeled directed graph. Besides other features, SPARQL2NL focuses to enhance the user understanding about the query so that, user can provide an appropriate query. For retrieval, it can process the triples using RPQs and search the matching graph pattern similar to the queried graph pattern. The numbers of match found in the dataset were returned as the resulted graph. Consequently, it is closed under graph but not under the triple. Apparently, it uses the RPQs containing predicate as a labeled edge with a restriction of association with a finite set of alphabets similarly like a SPARQL. Hence, a low reachability was also found in SPARQL2NL. Furthermore, the query pattern is different to SPARQL, it transforms the SPARQL query into natural language statements. Hence, the better understanding and ease of query writing provided to novice while using SPARQL2NL. However, power of expressive queries remains same as in SPARQL.

LITEQ is a graph-based query language that represents a RDF triples into an edge-labeled directed graph. LITEQ was designed on top of node path query language (NPQL) that provides an intuitive syntax with operators for the

navigation and exploration of RDF Graphs. For retrieval, it can process the triples using navigational operators of NPQL and search the matching graph pattern similar to the queried graph pattern. The number of matches found in the dataset were returned as the resulted graph. Consequently, it is also closed under graph rather than the triples. Apparently, it uses the navigational operators of NPQL to provide higher reachability to RDF query. As a result, a medium reachability was found in LITEQ. Furthermore, LITEQ also supports RDF schema that defines a retrieval of RDF resources so that, the retrieved resources can be used at runtime. Also, LITEQ provides a static typing mechanism by retrieving the schema from the available RDF resources, as a result programmer need not to manually re-create type structure.

CPSPARQL is another RDF query language that also represents an RDF triple into an edge-labeled directed graph. CPSPARQL is a variant of nSPARQL that was designed on top of PSPARQL for providing a navigation support using the constrained regular expressions (CREs). For retrieval, it can process the triples using CREs and search the matching the restricted graph pattern similar to the queried graph pattern. The number of matches found in the dataset were returned as the resulted graph. Consequently, it is also closed under graph but not under the triple. Apparently, it uses the CREs to provide higher reachability than a SPARQL query. As a result, a medium reachability was found in CPSPARQL. Furthermore, CPSPARQL is similar to SPARQL in terms of querying pattern. Therefore, the similar issue has to face by novice while using CPSPARQL. However, a restriction over CPSPARQL i.e., cpSPARQL [7] can express all nSPARQL queries for that reason expressiveness of CPSPARQL is similar to nSPARQL.

TRIAL* is an RDF query language based on relational algebra. For retrieval, it can process the triples using various operations of relational algebra like composition, join, difference, union, intersection etc. It excludes the projection operator from the language while using conditional cartesian product so that the outcome should be a triple. The number of match found, after performing the conditional cartesian product in the dataset; were returned as the resulted triples. Consequently, it is closed under triples. Apparently, it uses the relational algebra that gives higher coverage of the data. As a result, a higher reachability was found in TRIAL*. Furthermore, TRIAL* also uses query pattern that are similar to structured query language (SQL). However, conditional join pattern or more specifically triple algebra requires in depth knowledge about the language as well as about algebraic expressions as a result novice have to suffer a bit while using TRIAL*. Whereas, the same triple algebra enhances the expressive power of the language [9].

	Closure	Reachability	Simplicity	Semantic Schema Support	Active Schema Catalog	Expressiveness
SPARQL	Closed over Graph	Low	Medium	No	No	Low
nSPARQL	Closed over Graph	Medium	Medium	Yes	No	Medium
SPARQL2NL	Closed over Graph	Low	High	No	No	Low
LITEQ	Closed over Graph	Medium	Medium	Yes	Yes	Low
CPSPARQL	Closed over Graph	Medium	Medium	Yes	No	High
TRIAL*	**Closed over Triples**	**High**	Medium	No	No	High

Table 1: Synopsis of Comparative Study

CONCLUSIONS

In this paper we reviewed various techniques used to retrieve RDF triples. While different techniques have their own merits, graph database approaches were usually used in retrieval engines. Whereas, graph database approaches have a restricted reachability and therefore loss of information is easily predicted. We compare various available query languages on six characteristics and found that only TRIAL* can fulfill the closure over triples while providing the highest reachability along with high expressive power. This comparison shows that relational algebraic technique is better than graph-based technique for RDF triples.

References

[1] Wood, Peter T., (2012). Query languages for graph databases. (ACM) Sigmod Record, **41(1)**:50–60.

[2] Renzo Angles, Claudio Gutierrez, (2015). RDF Query Languages Need Support for Graph Properties. Whitepaper.

[3] Xin Hu, Depeng Dang, Yingting Yao, Luting Ye, (2018). Natural Language Aggregate Query over RDF Data. Information Sciences, DOI: 10.1016/j.ins.2018.04.042.

[4] Prud,Hommeaux, Eric, and Andy Seaborne(2008). "SPARQL query language for RDF." W3C recommendation 15.

[5] Pérez, J., Arenas, M., & Gutierrez, C. (2008). nSPARQL: A navigational language for RDF. Springer Berlin Heidelberg: pp. 66-81.

[6] Ngonga Ngomo, Axel-Cyrille, et al (2013). Sorry, i don't speak SPARQL: translating SPARQL queries into natural language. Proceedings of the 22nd international conference on World Wide Web. International World Wide Web Conferences Steering Committee: 977-988.

[7] Scheglmann, Stefan, Martin Leinbereger, and Steffen Staab, (2015)."LITEQ: Language Integrated Types, Extensions and Queries for RDF Graphs. ACM 978-1-4503-1871-6/13/01.

[8] Alkhateeb, Faisal, and Jerome Euzenat. (2014) Constrained regular expressions for answering RDF-path queries modulo RDFS. International journal of web information systems: 24-50.

[9] Libkin, Leonid, Juan Reutter, and Domagoj Vrgoč (2013). Trial for RDF: adapting graph query languages for RDF data. Proceedings of the 32nd symposium on Principles of database systems. ACM.