International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-6, August, 2019

A Query Search Technique for Semantic Web Based on The Dynamic Distribution of Backjumping Method

Rubin Thottupurathu Jose, Sojan Lal Poulose

Abstract*: *In the resource description language of the semantic web, vagueness of Resource Description Framework (RDF) data is playing an important role. The effective querying of the RDF data is increasing importance in semantic web. In this research, the Dynamic Distribution of BackJumping (DDBJ) algorithm is proposed in the fuzzy graph due to the ability of the algorithm to maintain its autonomy in the data. The fuzzy graph is generated from the triplets in the RDF and the vertices, edges are extracted from pattern matching techniques. The vertices and edges are applied in the DDBJ to suggest the query in the semantic web. To analyze the effectiveness of the proposed DDBJ, the two real datasets are used. The proposed DDBJ method has the f-measure of 59 %, while the state-of-art method such as backtracking has achieved 56 %. The result shows that the DDBJ method has the higher performance than existing method in query processing.

Index Terms*: *Dynamic Distribution of BackJumping, fuzzy graph, pattern matching, Resource Description Framework, and Semantic web.

I. INTRODUCTION

Querying for the required information in the large storage volumes is an important concern on the internet. Relational data are successfully used by the database community and increase the performance of the semantic web in the last 40 years [1, 2]. Developing databases related to the linking and linking tools configuration are major problems in the data analysis [3]. The Resource Description Framework (RDF) archiving system developed from that databases, for examples, support query engines that is based on the standard SPARQL query language [4]. The increasing number of data in the web requires the effective method to handle these semantic data and provide the proper answer for the user queries. RDF is the widely used standards, which is developed by W3C to structure the Semantic Web [5]. The data is retrieved based on the matching points are identified between the overlap of the information in the different ontologies [6].

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
Rubin Thottupurathu Jose*, School of Computer Sciences, M G University, Kottayam, Kerala, India.
Dr Sojan Lal Poulose, Principal, Mar-Baselious Institute of Technology and Science, Kothamangalam, Kerala, India.

The triple values of data are used to plot the graph, provides the relation between the entities and the search method analyze the relation between entities to retrieve the answer. RDF store allows storing triples and SPARQL queries are used to retrieve them [7]. Many data publishers are relied on the benefits of the Semantic web for quick publishing, processing the data by machines and parsing [8]. Many techniques tend to find the relevant answer to the user query based on semantic web. Most of techniques are not answer the query effectively due to the representation of RDF and lack of search method in querying [9 - 10]. In this research, the Dynamic Distribution BackJumping (DDBJ) algorithm is applied in fuzzy graph for query processing. First, the triplets of the RDF are effectively represented by the fuzzy graph. The DDBJ algorithm is applied to the fuzzy for the query processing technique. The DDBJ can order the variables in the fuzzy graph and still able to maintain the autonomy for triplets. The developed technique is evaluated in the dataset and compared with the state-of-art method such as backtracking technique.

The organization of the paper is in the form of literature survey in section II, proposed method described in the III, experimental design is given in section IV, result analysis illustrated in section V. The conclusion of the research is made in section VI.

II. LITERATURE SURVEY

The notable research in the semantic query in searching RDF was analyzed in this section. The various techniques are applied to the semantic web query search to improve its effectiveness. Arenas, et al. [11] presented the faceted search with OWL 2 ontologies in the RDF graph. The faceted search as query paradigm easily identifies the well-defined SPARQL fragments. The computation complexity for answering the query is high in the system. The faceted interface problem is analyzed in this study to extract the meaningful queries in exploratory search. The search method is used to increase the effectiveness of the result. Achichi, et al. [12] provided the solution to the important number of heterogeneities to minimize the user effort in the configuration based on four processes. The first step involves in property filtering, an automatic data cleaning process. The second process involves in profiling instance to denotes each resource in the sub graph considered relevant for comparison task. The vector representation is provided for instances in step three.

The fourth step uses the post-processing of hierarchical clustering and key ranking technique aims to reduce the high similarity, even it is not an identical instance. The optimization technique is required to improve the efficiency of the method.

Zimmermann, et al. [13] structure the annotated language, deductive system and address the query answering problem. The generic method is applied for the hybridization of the multiple annotation process that represent the temporally annotate fuzzy RDF. The problem in the query language that is inspired by SPARQL, including several features of SPARQL with the formal definitions of their semantics. The effective search technique is needed to combine with the annotation technique to improve the relevance of the answer.

Zou, et al. [14] included the gStore system to manage the uniform and scalable manner in the SPARQL queries with aggregate and wildcards in the dynamic RDF datasets. This is the graph- based method and stores the RDF data in the large graph and represents a SPARQL query in the query graph. The query optimization problem is converted into a subgraph matching problem. The index with pruning rules and efficient search algorithm is applied to increase the performance in the system. The query optimization method can be applied to increase efficiency.Li, et al. [15] introduces the practical extension of the RDF model to denote the vagueness in the fuzzy graph. The fuzzy graph is constructed based on the RDF data and analyzed the problem in the graph. The graph pattern surpasses the previous method in terms of expressiveness that has the regular expression to represent the structural condition in the graph in addition to the fuzzy condition on the vertex values. The algorithm is applied to evaluate the graph pattern in the queries based on a revised notion of graph homomorphism. The backtracking technique is not effective in solving the strategic problem and time consumption is high. The limitations in the previous works are mostly inefficient and more time-consuming in query processing. This limitation can be overcome by proposing the search technique to increase performance.

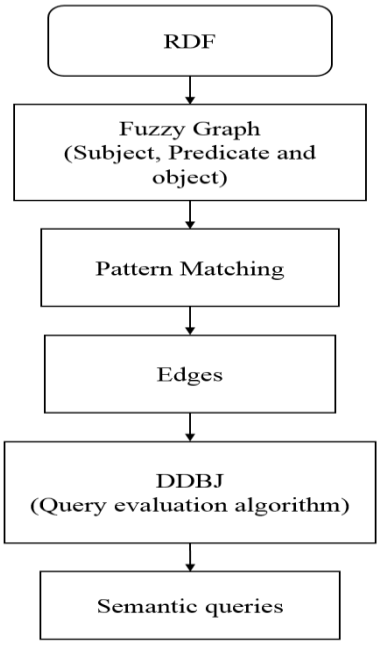

Fig. 1. The block diagram of DDBJ algorithm in fuzzy graph

III. PROPOSED METHOD

The RDF data are represented in the fuzzy graph and pattern matching is applied to the graph. The DDBJ algorithm is used for the query processing that is applied on the vertices and edges. The fuzzy graph consists of the triplet i.e.., subject, predicate and object, consists of vertices and edges. The pattern matching fuzzy graph and DDBJ algorithm are briefly explained in this section. The block diagram of the DDBJ algorithm in fuzzy graph is shown in Fig. 1.

A. Pattern Matching Fuzzy Graph

The RDF triples are distributed in the fuzzy graphs and pattern matchings are applied in the vertices and edges. The pattern matching fuzzy graph effectively represent the RDF data. The RDF graph is constructed and the conditions are followed here is same as in the research [15]. A fuzzy graph pattern $Q=(V_q,E_q,F_v,R_E)$ is matched with a fuzzy RDF data in the graph $G=(V,E,\Sigma,L,\mu,\rho)$ with the satisfaction degree threshold δ_t, present in the injection mapping $\phi: Q \to G$, which consists of total mapping with vertices and edges of Q to vertices and path of G, such that:

- **Matching vertices**

Every vertex $u \in V_q$ has an image vertex $\phi(u) \in V$ by the injective function. For example, if u is consider as constant vertex $(F_v(u) \in U \cup L)$, then its matching vertex $\phi(u)$ associated with a satisfaction value of $(F_v(u) \in VAR)$ and their labels are matched (i.e., $L(u)=L(\phi(u))$); if u is consider as variable vertex, the set of matches is $(\phi(u) \in U \cup L)$, i.e., vertices $\phi(u)$ is increased by all the matches ϕ of $Q in G$ that is each item is related with the satisfaction value $\delta_u = \mu(\phi(u))$.

- **Checking Condition on vertices**

In the fuzzy condition C of vertex $u \in V_q, \phi(u)$ satisfies C with a satisfaction degree of δ_{co} is explained as follows, in the manner of C:

- In case, C is in the manner of "$?x op c$", then $\phi(?x)$ satisfy the condition C with a satisfaction degree of $\delta_{co} = \mu_{op}(\phi(?x), c)$ The membership function is denoted as μ_{op} of fuzzy comparator. The condition is analyzed by the Boolean semantics operator in the fuzzy, this provide 1 if the condition is true, otherwise 0.
- If the condition C is in the manner of "$x?op?y$", the condition C satisfy the $\phi(?x) and \phi(?y)$ with the degree of $\delta_{co} = \mu_{op}(\phi(?x), \phi(?y))$.

- If C is in the manner of "$x\,is\,Tterm$", then $\phi(?x)$ satisfies the condition C to the degree $\delta_{co} = \mu_{Fterm}(\phi(?x))$. Here, μ_{Fterm} represent the fuzzy membership function of the fuzzy term $Fterm$.
- If C is in the manner of $C1 \wedge C2 \, or \, C1 \vee C2$, usual interpretation of the fuzzy operator is applied.

- **Matching Edges**

For each edge $(u_i, u_j) \in E_q$, there exist two vertices $\phi(u_i) \, and \, \phi(u_j) \, of \, V$ such that $\phi(u_i) \, and \, \phi(u_j)$ match vertices $u_1 \, and \, u_2$ respectively, and the path p of the vertex $\phi(u_i) \, to \, \phi(u_j)$ in G fulfills $re(u_i, u_j)$ with the degree of $\delta_{re}(p)$ is defined as follows, according to the form of re (in the following, $R, R1 \, and \, R2$ are regular expressions):

1) re present in the manner of ɜ:. If path is zero, then $\delta\_re(p) = 1 else \, \delta_{re}(p) = 0$.
2) δ_{re} is present in the manner of $e \, with \, e \in U$: If path is an edge e^t from vertex $\phi(u_i) \, to \, \phi(u_j)$, where $e^t = e$, then $\delta_{re}(p) = \rho(\phi(u_i)\phi(u_j)) else \, \delta_{re}(p) = 0$.
3) re is in the manner of $R1.R2$: Assign P to the set of all path pairs $(p1, p2)$, hence p is in the manner of $p1p2$. One has $\delta_{re}(p) = \max P(\min(\delta R1(p1)))$.
4) re is in the manner of $R1|R2$: One has $\delta_{re}(p) = \max(\delta_{R1}(p1), \delta_{R2}(p2))$.
5) re is in the manner of $R^+ : Let \, P$ be set of tuples path $(p1, ..., pn)(n > 0)$ such that p in the manner of $p1...pn$. One has $\delta_{re}(p) = \max_p(\min(\delta_R(p_1), ...\delta_R(p_n)))$.

- **Aggregating satisfaction degree**

The $\delta_Q(G)$ denotes the satisfaction degree for overall query, which is the addition of each elementary matching and conditions. The satisfaction degree is the sum of the elements; different types of data are considering for different application. For example, minimum, is a cautious choice that consider the satisfaction value of the set of triples is the least triple value. An optimistic choice is medium that is the satisfaction value. Note that, the satisfaction degree of $\delta_Q(G)$ is much greater than the δ_t . If no matching is found, then $\delta_Q(G) = 0$, i.e., G does not match.

The graph patter Q is analyzed and the result is provided in the binary relation as $M \subseteq V_q \times V$.

1) for each $u \in V_q$, analyze $v \in V$, hence $(u, v) \in M$;
2) for each edge (u_i, u_j) , path p is present from $v_i \, to \, v_j \, in \, G$, for the (i) the vertex label $L(v_i) \, of \, v_i$ fulfill the predicate condition presented by $F_v(u_i)$; (ii) the path is constructed based on rectangular expression $re(u_i, u_j)$; and (iii) (u_j, v_j) is present in M .

B. Dynamic Distribution Backjumping

The Dynamic Distribution Backjumping algorithm is the complete, distributed form of the backjumping algorithm that process on the graph. The algorithm combines the dynamic backjumping algorithm with the synchronous AFC algorithm, which has computation efficiency [16]. This is combined with variable order and dynamic heuristics value.

a. Procedure

Agent performs the value assignments process into two stages:

- Advanced Forward Phase: The advanced forward phase process in the time of adding a new tuple value in the partial solution.
- Backjumping Phase or backward: This phase process when agent has some problems. The process is performed in backward to find the culprit agent.

The forward phase is forward sequentially: the agent provides the OK signal to the next agent to process and Forward Checking (FC) message to the unassigned agent. If the agents encounter the conflicts in the messages, then the process is backjumped to find the culprit agent and send OK message to next agent, No Good (NG) for an unassigned agent.

The DDBJ method uses the dynamic value and variable reordering in the manner of heuristics for the searching process. Each agent has the potential conflict list in its domain and also the conflicts of the other agents. The agents select the lower counter value to assigns the variables and send the OK message to the agents with a higher value. If the agents have equal counter value, then it follows the chronological order. Initially, each agent's counter values are set as zero.There are 8 types of messages are used in the DDBJ. These are explained as follows.

1) SUCCESS: This message is the stopping message that is process to all agents. This message is send when the agent got solution.
2) FAILURE: This message is a stopping message that send to all the agents when the first agents found that there is no solution for the given problem.
3) ERROR: This message is termination message that is send to all the agents if error is found. The error is usually present when it crosses the limit of time.
4) OK: This message consists of the Current Partial Solution (CPS) that contains tuples i.e. variables and values, with the related TimeStamp's.
5) FC: This message has a copy of the OK message. This message has the copy of OK message. The assigned agent sends this message to the link agent.
6) NG: This message is send when the agent doesn't provide good partial solution (No Good). This is send to the linked agent, which is send to not assigned agent based on AgentView.

7) BT: This is send when the agent doesn't have a good partial solution. This is sent back to culprit agent.
8) PC: This message contains no good solution. This message is sent to possible conflict agent, found when it is conflict.

The DDBJ method process parallels on all agents to find the solution. The appropriate function is performed based on the received message. The OK message is send to first agent to start the process. The pseudo code of the search algorithm of DDBJ is shown below.

```
procedure receiveOK()
 1: if Msg is newer than AgentView then
 2:    update TimeStamps
 3:    if previously determined nogood then
 4:       return
 5:    set AgentView.consistent = true
 6:    updateDomain(MPS)
 7:    if success then
 8:       assignVal()
 9:    else
10:       backJump(previous)
end
procedure receiveFC()
 1: if Msg is newer than AgentView then
 2:    update TimeStamps
 3:    if not previously determined nogood then
 4:       set AgentView.consistent = true
 5:    if AgentView.consistent then
 6:       updateDomain(MPS)
 7:       if not success then
 8:          update PCA
 9:          send NG to unassigned agents; PC to agents in PCA
10:          backJump(culprit)
end
procedure receiveNG()
 1: if AgentView orderly contains Msg then
 2:    restoreDom()
 3:    set AgentView.consistent = false
 4: else if Msg is newer than AgentView then
 5:    set AgentView.consistent = false
 6:    update TimeStamps
 7:    updateDomain(MPS-last)
 8:    if not success then
 9:       update PCA
10:       send NG to unassigned agents; PC to agents in PCA
11:       backJump(culprit)
12: if self is assigned then
13:    reset to unassigned
end
```

IV. EXPERIMENTAL DESIGN

This section describes the dataset used in this method, experimental setup and the metrics used to evaluate the methods. The vertices and edges are varied to measure the performance of the methods.

A. Datasets

The two real data subsets from the IMDB and DBLP-S (small version) are applied to investigate the effectiveness. The IMDB is the movie datasets that have entity relationship graph data. The DBLP graph is a bibliography dataset and used for effectiveness and efficiency analysis. These two datasets are used to investigate the effectiveness and response time of the method.

B. Experimental setup

The DDBJ algorithm and the existing methods are executed on the Linux computer with Intel i5 processor and 8GB of memory. The techniques are evaluated in the same environment with the same data for the comparative purpose. The experiment is evaluated by varying the vertices and edges in the system. The various metrics are evaluated from the proposed and existing method to investigate the effectiveness.

C. Evaluation Metrics

The precision is a description of the system error and recall is the total relevant result provided by the method. The precision and recall are used to evaluate the performance of the proposed method. The f-measure is another metrics used in this method that is based on the measure of precision and recall. The f-measure can be defined as a measure of the test's accuracy. The formula for the f-measure is shown in the Eq. (1). The response time is the crucial metrics in the query processing system. The response time is also measured in this method to investigate the proposed DDBJ algorithm with existing methods.

$$f-measure = \frac{2\times precision\times recall}{\mathrm{Re}call+\mathrm{Pr}ecision} \quad (1)$$

V. EXPERIMENTAL RESULT

The development a large number of data in the semantic web, requires the effective method for query processing. Many existing methods involves in the use of a different method for the effective performance of the query processing. Many techniques lag in the effectiveness and response time. In this research, the DDBJ algorithm is proposed in the fuzzy graph for query processing. The two datasets such as IMDB and DBLP dataset are used to evaluate the effectiveness of the DDBJ method. The different metrics are used to measure the performance of the methods in query processing. The proposed DDBJ method is compared with state-of-art method such as backtracking [15] method and other two existing methods such as NAGA [13], and PQ [14]. This section provides a brief explanation about the DDBJ method and compares with other existing techniques.

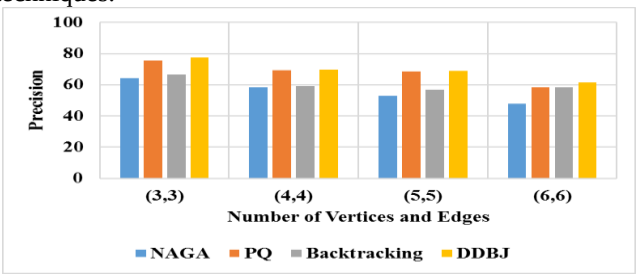

Fig. 2. Precision For The Different Methods

The precision is calculated for the different number of vertices and edges in the IMDB dataset. The 5 queries are performed for the different methods and compared with each other in the Fig. 2. The precision value is high for the DDBJ algorithm compared to the state-of-art method backtracking. The precision value of DDBJ is 61.5 % for the (6,6), while state-of-art method backtracking has the precision value of 58.2 %. The high efficiency of the DDBJ method is due to its capacity to maintain autonomy of the data than the existing method.

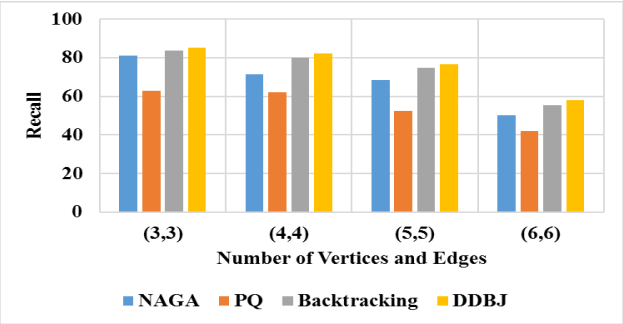

Fig. 3. Recall for different methods

The recall value is measured for the different methods in IMDB datasets and shown in the Fig. 3. The proposed DDBJ method has the higher recall value compared to the other existing method in search queries in RDF. These methods are experimented in the same environment and analyzed its performance. The 5 queries are used to evaluate the efficiency of the methods and recall value is calculated. The DDBJ algorithm has the recall value of 58.2 %, while state-of-art method has the recall value of 55.3 %.

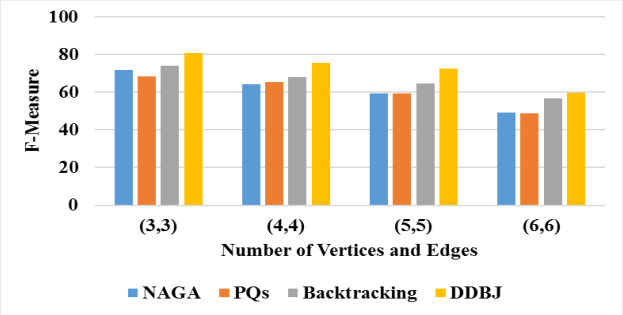

Fig. 4. F-Measure For Different Methods

The queries are executed in the IMDB dataset for different methods and compared in Fig. (4). The DDBJ algorithm has a higher f-measure compared to the other existing methods in IMDB dataset. The 5 different vertices and edges are used to evaluate the performance of the DDBJ method. The DDBJ algorithm has achieved the f-measure of 59 %, while the state-of-art method of backtracking has achieved 56 %.

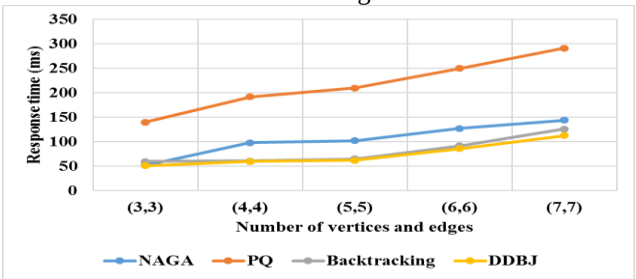

Fig. 5. Response Time For The Query Processing In IMDB Database

The response time of the different methods are measured for the IMDB database and shown in the Fig. (5). The response time is one of the important factor in the query processing method. The proposed DDBJ method has the lower response time compared to the other existing methods in IMDB dataset. The response time is measured for the different number of vertices and edges, as shown in Fig. (5). The response time of the DDBJ is 113 ms for the (6,6) and state-of-art method has the response time of 126 ms for (6,6).

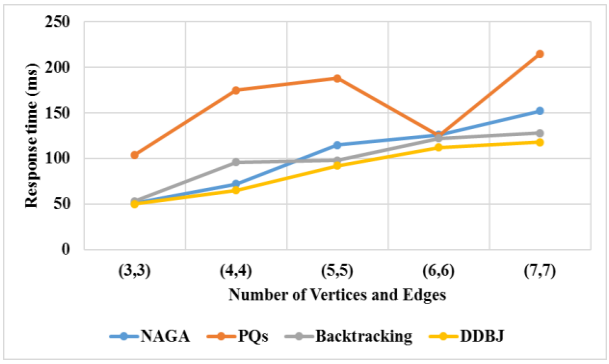

Fig. 6. The Response Time Of Query Processing In DBLP Data

The response time for the different method is measured in the DBLP dataset and shown in the Fig. (6). This shows that the proposed DDBJ method has lower response time compared to other methods in DBLP dataset. The proposed DDBJ algorithm response time increase with the increase of a number of vertices and edges. The proposed DDBJ algorithm has the response time of 118 ms, while state-of-art method has achieved the response time of 112 ms.

Therefore, the proposed DDBJ algorithm in the fuzzy graph has higher f-measure value than state-of-art method in query processing. The response time of the DDBJ method is also low compare to other methods.

VI. CONCLUSION

This paper aims to increase the query efficiency in the RDF by proposing DDBJ method in a fuzzy graph. The fuzzy graph is plotted based on the RDF data and pattern matching is applied in the graph. The vertices and edges are given as input to the DDBJ algorithm and this will search forward and backward in the fuzzy graph. The two real datasets are used to evaluate the performance of the proposed DDBJ method. The proposed DDBJ method compared with state-of-art-method in the two datasets to investigate the effectiveness. This shows that the proposed DDBJ method has a higher efficiency than the existing method. The existing and proposed methods are evaluated in the same environment. The proposed and existing methods are evaluated with different level of vertices and edges. The proposed DDBJ method has the precision value of 61.5 % when compared to the existing method of 58.2 %. The experimental result shows that the proposed technique has a higher efficiency compared to the existing method.

The future work of the proposed DDBJ method is modified to further decrease the response time.

REFERENCES

1. C. M. Medeiros, M. A. Musicante, and U. S. Costa, (2019). LL-based query answering over RDF databases. Journal of Computer Languages, 51, pp. 75-87.
2. S. Angelopoulos, C. Dürr, and T. Lidbetter, (2019). The expanding search ratio of a graph. Discrete Applied Mathematics, 260, pp. 51-65.
3. M. Achichi, Z. Bellahsene, M. B. Ellefi, and K. Todorov, (2019). Linking and disambiguating entities across heterogeneous RDF graphs. Journal of Web Semantics. 55, pp. 108-121.
4. R. Taelman, M. Vander Sande, J. Van Herwegen, E. Mannens, and R. Verborgh, (2019). Triple storage for random-access versioned querying of RDF archives. Journal of Web Semantics, 54, pp. 4-28.
5. X. Hu, D. Dang, Y. Yao, and L. Ye, (2018). Natural language aggregate query over RDF data. Information Sciences, 454, pp. 363-381.
6. X. Xue, and J. Chen, (2019). Using Compact Evolutionary Tabu Search algorithm for matching sensor ontologies. Swarm and Evolutionary Computation. 1(48), pp. 25-30
7. P. Bellini, and P. Nesi, "Performance assessment of rdf graph databases for smart city services," Journal of Visual Languages & Computing, 45, pp. 24-38.
8. J. L. Martinez-Rodriguez, I. Lopez-Arevalo, and A. B. Rios-Alvarado, (2018). OpenIE-based approach for Knowledge Graph construction from text. Expert Systems with Applications, 113, pp. 339-355.
9. H. Chen, A. Trouve, K. J. Murakami, and A. Fukuda. (2017). An intelligent annotation-based image retrieval system based on RDF descriptions. Computers & Electrical Engineering, 58, pp. 537-550.
10. P. Peng, L. Zou, and Z. Qin, (2017). Answering top-K query combined keywords and structural queries on RDF graphs. Information Systems, 67, pp. 19-35.
11. M. Arenas, B. C. Grau, E. Kharlamov, Š. Marciuška, and D. Zheleznyakov. (2016). Faceted search over RDF-based knowledge graphs. Journal of Web Semantics, 37, pp. 55-74.
12. M. Achichi, Z. Bellahsene, M. B. Ellefi, and K. Todorov, (2019). Linking and disambiguating entities across heterogeneous RDF graphs. Journal of Web Semantics. 55, pp.108-121
13. A. Zimmermann, N. Lopes, A. Polleres, and U. Straccia, (2012). A general framework for representing, reasoning and querying with annotated semantic web data. Web Semantics: Science, Services and Agents on the World Wide Web, 11, pp. 72-95.
14. L. Zou, M. T. Özsu, L. Chen, X. Shen, R. Huang, and D. Zhao. (2014). gStore: a graph-based SPARQL query engine. The VLDB Journal—The International Journal on Very Large Data Bases, 23(4), pp. 565-590.
15. G. Li, L. Yan, and Z. Ma. (2019). Pattern match query over fuzzy RDF graph. Knowledge-Based Systems, 165, pp. 460-473.
16. D. Hou, and R. Stouffs. (2019). An algorithmic design grammar embedded with heuristics. Automation in Construction, 102, pp. 308-331.

AUTHORS PROFILE

Rubin T Jose, MCA, M Tech. Scholar in the area of Semantic web and Ontology Engineering, already published 3 Journal Papers and 8 Conference publications in the area. Attended short course in the Protégé tool in Stanford University, USA. He has got a total of 15 years of teaching experience in the field of Computer Science and Engineering.

Dr P. Sojan Lal has more than 30 years of blended experiences, with major international petroleum companies in Middle East and premier educational institutions in India. He has authored 6 technical books, two of them published in Germany and other books published in India. He is an approved research supervisor of Mahatma Gandhi University, Kottayam; University of Petroleum and Energy Studies, Dehradun; and APJ Abdul Kalam Technological University, Kerala, India.

Dr. Sojan Lal has authored 65 National and International Journal and Conference papers and guided 4 PhD scholars. He has the world record for the highest number of publications within shortest period in 2014. He has been listed in "Marquis Who's Who in the World" since 2009, representing the world's most accomplished individuals.