MULTI- INFERENCE APPROACH FOR EFFICIENT DISTRIBUTED REASONING OF LARGE-SCALE RDF DATA

Saikishore Yagala
P.G Student, Computer Science and Engineering
Institute of Aeronautical Engineering,
Hyderabad, India

Myneni Madhu Bala
Professor, Computer Science and Engineering
Institute of Aeronautical Engineering,
Hyderabad, India

Abstract

In current days, it is emerging to focus on fast growth and a large volume of semantic RDF data for efficient inference-based reasoning. Traditional methods include PTIF (property transfer inference forest), CTIF (class transfer inference forest), DRTF (domain/range transfer forest) determines inference-based reasoning results over to extensive RDF data. It is good enough for performing semantic web query processing; however, these methods are sensitive due to its inference structures in reasoning. Multi-level and multiple inference structures (MLMIS) are processed within the framework of MapReduce for addressing the sensitiveness problem for ample RDF datasets reasoning. Dbpedia benchmarked RDF datasets are used in the experimental study and evaluate the response time of various queries under different Hadoop scripts for demonstrating the effectiveness of PTIF, CTIF, DRTF.

Keywords: Ontologies, RDF, Semantic Data, Distributed Reasoning, MapReduce.

1. INTRODUCTION

Resource description framework (RDF) is a standard notation for representing ontologies that can describe semantic web knowledge in the form of triples. Each triple provides a unit of information in RDF for semantic web data. Triple shows the relationship between subject and object based on the mentioned predicate. Knowledge of semantic web data is mined from inferences of triples hierarchy and used in popular applications, like e-marketplace [1], web services [2], optimal computing [3], advanced database applications [4], health care applications [5]. Nowadays, ontologies can be used as a formal specification for representing the concepts and relationships. Thus, RDF triples play a Vital role in knowledge mining. Triples of RDF consists of the following information: subject, predicate, and object; it is a graphical language that is used to make a data interchange on the Web. RDF extends the linking structure of the Web to use URIs to name the relationship between things as well as the two ends of the link (this is usually referred to as a "triple"). Using this simple model it allows structured and semi-structured data to be mixed, exposed, and shared across different applications [w3school]. This linking structure forms a directed, labeled graph, where the edges represent the named link between two resources, represented by the graph nodes. This graph view is the easiest possible mental model for RDF and is often used in easy-to-understand visual explanations. Semantic Web contains a large number of triples in RDF format.

Inference predicates are applied for discovering reasoning based on subjects and objects, which is useful for processing typical queries over large scale data, and this process is known as inference reasoning. The semantic web is a large scale RDF data, in which deriving inferences may pose a challenging issue of computational time. Several baseline approaches include PTIF, CTIF, and DRTF solves the problem of Computational time. This paper shows the solution for the said problem by reasoning data using both multi-level and multiple hierarchy inference structures (MLMIS). This MLMIS effectively performs inference reasoning with all levels of predicates in RDF. It is experimented using different mappers in Hadoop, i.e., PIG and HIVE mappers, and these mappers are used for demonstrating efficient mapper for performing distributed query reasoning for benchmarked DBpedia datasets. The schematic diagram of the proposed work is shown in Fig. 1.

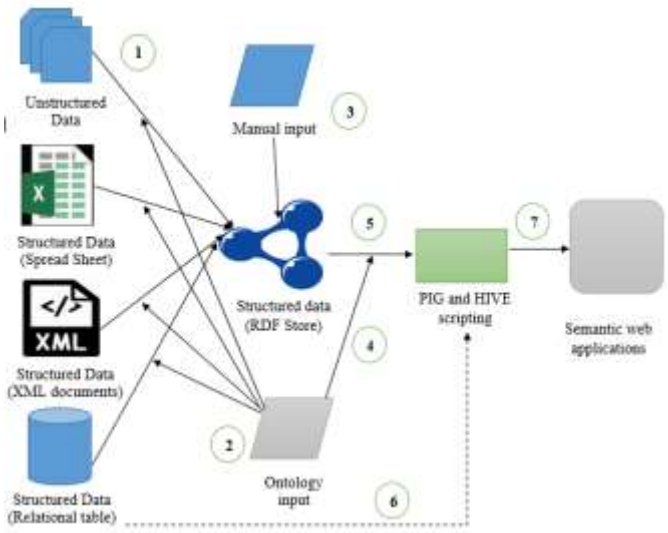

Fig 1: Schematic Diagram for Semantic Query Processing

It shows the procedure of effective query processing for semantic web data. Suppose the data is in an unstructured text document, then it is stored or managed in a spreadsheet or any other structured formats like XML or RDF. Given data is taken from either stored web documents or manually and then it is pre-processed in RDF format in terms of subject, predicate, and objects. Use Hadoop mappers, either PIG or HIVE, to process of distributed inference reasoning through writing suitable queries. Key steps of semantic query processing are described as follows:

1. structured or unstructured data is pre-processed based on conceptual ontology schema and converted into RDF form
2. Analyze subjects and objects based on predicates entered manually by the user
3. Write suitable mappers for supporting of distributed query processing using either PIG or HIVE scripts
4. The efficiency of mappers are evaluated for semantic web applications

The rest of the paper is organized as follows: Section 2 discusses related work, Section 3 overviews distributed inference reasoning, Section 4 exhibits the experimental study, and Section 5 presents the conclusion and future scope of the paper.

2. Related Work

Semantic web data is commonly represented in either RDF or RDF graph formats. RDF based semantic data is shown as directed graphs, and the key ability of RDF is to merge the data sources without having any schema definitions [1]. It is possible to merge the combination of unstructured and semi-structured data across data websites, such cases users can easily crawl the data for application usage. This approach follows the property of reusability for existing information, and it is not necessary for the creation of a new one [2-3].

Three key notations are used in RDF graphs; they described for the following purposes: circles or ellipses are used for representing either resources or subjects, rectangles for literals, and an arrow from subject to object is used for representing predicate [4].

In web data applications, triple stores are playing a vital role, and triples processing is very complex for those applications. As a result of this clearly reflect the strength and weakness while executing in real-time applications. According to Mohamed Morsey et. al.[5], creation of SPARQL benchmark procedure, which they have been applied to the DBpedia knowledge base [6]. In earlier research comparison of relational and triple stores, performance measurement against a relational database which had been converted to RDF [7].

The following example shows the statement for a precise understanding of the RDF statement.

Statement: “The Authors of https:// iare.ac.in// is Dr. LVN” this example statement describes the subject is https://iare.ac.in//, the predicate is “Authors,” an object is “Dr. LVN.” In RDF, the predicate is assigned with a Uniform Resource Identifier (URL), and the data can be viewed as simple statements such as subject-predicate-object. As per the sources of W3C, billions of triples are found on the web, and this data is increased rapidly. Thus, it needs to effective and scalable data RDF data processing system for handling the huge amount of data. Present RDF query languages are DQL, N3QL, RDFQ, and RDQ, etc. [8]. It requires high computational time when querying with RDF data. SPARQL [9] is a well-known RDF query language which is described by W3C;

however, computational requirements on SPARQL is not efficient on huge datasets. It is an emerging requirement to develop more optimal solutions for running queries over huge RDF datasets. Semantic web connects the web documents of the globe, and it makes as publicly available and machine-understandable. THE illustrative RDF sample graph is shown in Fig. 1 for an understanding of various RDF subject of 'cygri,'; three predicates 'type, name, and based_near' and their respective objects.

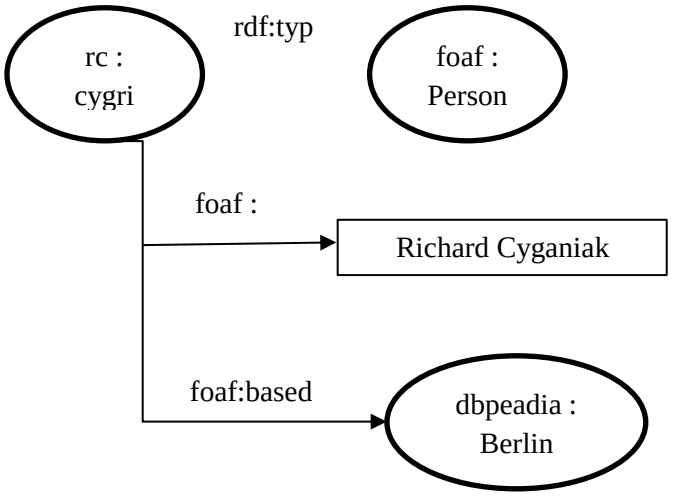

Fig 1 Sample RDF Graph

Semantic web data follows the graph model (RDF), and 'foaf' describes the people profiles and their relationships. Social media provides RDF based descriptions, and it can show such data is more significant than traditional models of social networks. Computation overhead is more expensive for the reasoning of such large RDF based social data. It is addressed by distributed processing, which effectively runs the parallel processing in the Hadoop environment. Scalability and computation time values are optimized by the proposed technique than traditional methods. Key contributions of the paper are organized as follows:

1. RDF based social data is extracted, and features are analyzed with RDF graphs
2. Ontologies based knowledge is mined from the reasoning of social semantic web data
3. MapReduce task framework is established for supporting distributed processing
4. Computational time is reduced by proposed distributed reasoning of RDF based social semantic web data
5. Analyze the scalability and computation time in the experimental study and shows these values are more optimal in proposed distributed reasoning

2. OVERVIEW OF BACKGROUND STUDY

Semantic web data is commonly represented in either RDF or RDF graph formats. RDF based semantic data is shown as directed graphs, and the key ability of RDF is to merge the data sources without having any schema definitions [10]. It is possible to merge the combination of unstructured and semi-structured data across data websites, such cases users can easily crawl the data for application usage. This approach follows the property of reusability for existing information, and it is not necessary for the creation of a new one [11–12]. Three key notations are used in RDF graphs; they described for the following purposes: circles or ellipses are used for representing either resources or subjects, rectangles for literals, and an arrow from subject to object is used for representing predicate [13]. Fig. 2 shows all these notations in the sample RDF graph, and respective RDF statements are shown in Fig.3.

```
<?xml version="1.0"?> <rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax
ns#" xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description>
      <dc:creator>Karl Mustermann</dc:creator>
      <dc:title>Algebra</dc:title>
      <dc:subject>mathematics</dc:subject>
     <dc:language>EN</dc:language>
     <dc:description>An introduction to
algebra</dc:description>
</rdf:Description>
```

Fig. 2: Sample RDF with Three Notations

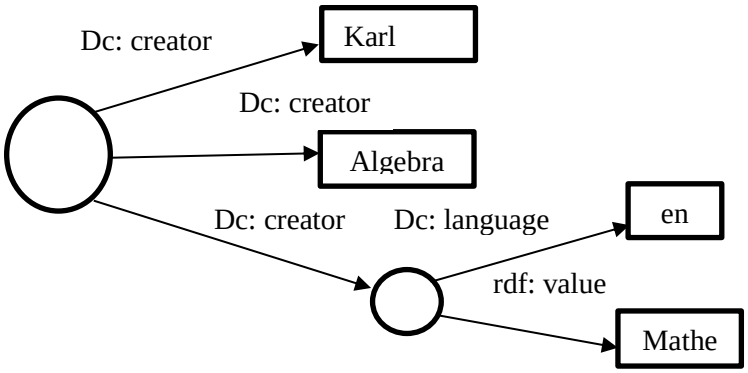

Fig.3: Sample RDF data

SPARQL [14] follows the SPARQL protocol, and it can be used for maintaining RDF queries and RDF query language. It gets the information from RDF, and it is recommended by W3C. SPARQL also contains the data in a triple format and some complex SPARQL having disjunctions and conjunctions and other collective patterns [15,22]. Many tools are available [16] for the construction of SPARQL queries and translate these queries to other query languages like SQL. The SPARQL queries are made to run on NoSQL databases like Cassandra, and MongoDB, etc. An illustrative example of a query of SPARQL is shown as follows:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE {
    ?person a foaf:Person.
?person foaf:name ?name
    }
```

This query is defined more specifically to return the set of all personal names, and it consists of two triples, one for triples of type person and other is find the persons who have person name associated with them[17]. Here, PREFIX is used for representing namespace foaf: and it can be used in query statement instead of using the entire path 'http://xmlns.com/foaf/0.1/'. SELECT is used for getting the information in the database and keep in a tabular format [18]. The construct will be helpful for finding the information from source repositories, and that is to be converted into basic RDF formats. DESCRIBE query find the RDF graph from SPARQL endpoints. Large RDF data handling is one of a key issue in query processing, in such large data cases, Apache Cassandra [19] is succeeding greatly[20]. It is developed by Facebook, and it can be proven as excellent in performance. Concepts of master and slave are ignored in Cassandra, and all the nodes act as peers, and nodes are communicated using GOSIP protocol [21]. Initially set the seed nodes as Cassandra nodes in a cluster setup and send the information in a format of peer-to-peer across the Cassandra cluster setup, and Cassandra acts as decentralized so that there is no cause of a single point of failure.

There is proper continuation work is happen among peers in a Cassandra cluster setup even failure is occurring at a single node. It keeps replication of data among peers and it is customizable. Thus, Cassandra highly satisfies the fault tolerant and it also supports to MapReduce [22] for performing online analytics. Cassandra maintains its own language, "Cassandra Query Language (CQL)" [23]. CQL supports relational database operations such as create, insert, delete, describe, and alter, etc. Client applications with Cassandra may be written in various languages such as Java and Python. The smallest unit in Cassandra is referred to as a column, and it is very similar to a table column. A collective column value is referred to as a row [24]. Each row maintains a primary key for maintaining unique records. Cassandra stores data in a fashion of column-family data model, and it showed in Fig. 4 and data model having columns, rows, column families, and key-space. Data sharing on nodes impose an optimal MapReduce task, and it is proposed in the following section for achieving good scalable reasonable results for large RDF datasets [25][26].

2. PROPOSED ARCHITECTURE

Hadoop based MapReduce is a programming model that can effectively deal with large RDF data with data sharing in a cluster setup of several nodes, hence it is known as a massive parallel programming paradigm. This paradigm model consists of Map ( ) and Reduce ( ) functions, whereas mapping ( ) as to perform any filtering function, but it maps the relevant data.

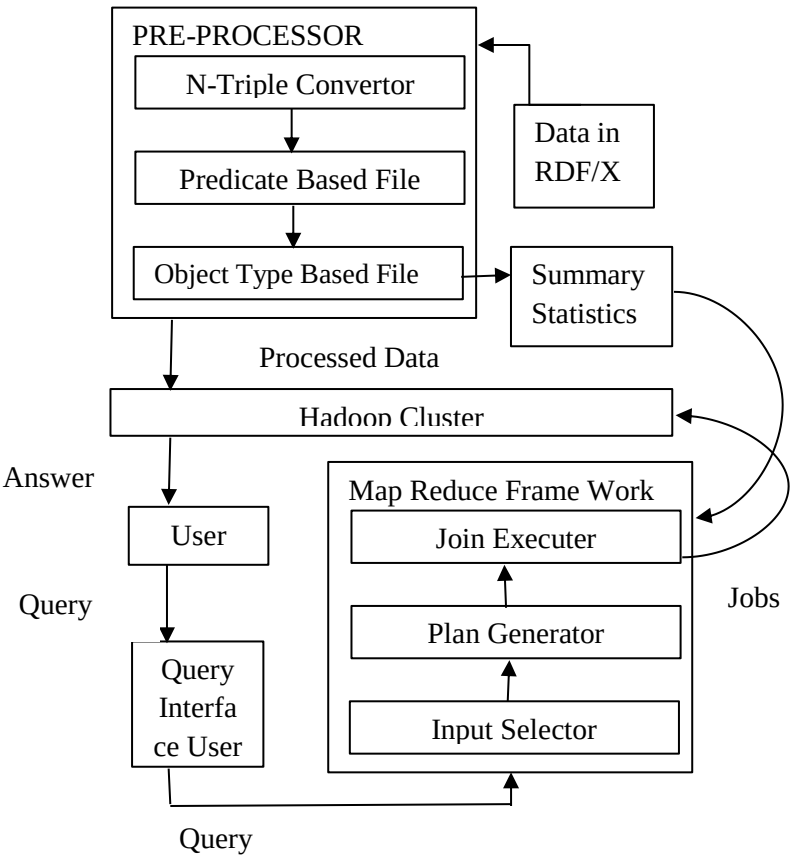

Fig. 5 Proposed System Architecture

The processing input of the map ( ) phase is the list of key-value pairs, and the output of the map ( ) phase may shuffle and sort steps so that output will be a list of keys and values. These values may be the input of the Reducer ( ) function.

Fig. 5 shows the proposed architecture for efficient semantic web data distributed processing, and it is a more scalable architecture model for large RDF datasets.

The proposed system architecture depicts three key components of data generation and data pre-processing. Data generators may generate the data in RDF / XML formats. This data is converted into triples using a triple converter or Jena framework. All triples may show the data in subject, predicate, and object formats, in which predicate based file splitter takes the predicates data and partitioned into predicate file. Object type based file splitter may split the predicates data based on the type of object data. The input of RDF triple is in one line file, and whole data is placed in a file. Query processing of large RDF dataset requires high computational time. Large RDF data may be split into smaller files in the Hadoop component for performing distributed reasoning. This proposed Hadoop based framework process the enormous data of RDF effectively with distributed highly tolerate system.

5. EXPERIMENTAL STUDY

Lehigh university benchmark (LUBM) is particularly important for three key cases: 1) extensional query support over ontologies 2) large-scale data 3) moderate size ontologies tests. The following notation is mainly used for the generation of the LUBM dataset.

LUBM (N, S): the dataset contains N universities and generated using a seed value of S.

LUBM is the benchmarked and most suited for evaluation of queries over equivalent RDF datasets. The data characteristics of LUBM are described in Table 1.

Three different LUBM datasets are generated using the LUBM UBA data generator.

Table 1: LUBM Test Data Characteristics

LUBM Data/ Number of Instances	**LUBM (5,0)**	**LUBM (10,0)**	**LUBM (15,0)**
Number of Classes	43	43	43
	25	25	25
Number of Data-type Properties	7	7	7
Number of Class Instances	129533	263427	556572
Number of Property Instances	516116	1052895	2224750
Number of Triples	646128	1316993	2782419
Data Size	44.8MB	124.6MB	259.2MB

LUBM (5, 0) denotes the data generated for five universities beginning with index 0; LUBM (10, 0)

denotes the data generated for 10 universities beginning with index 0; LUBM (15, 0) denotes the data generated for 15 universities beginning with index 0. A total number of triples, classes, property instances, and other related descriptions are shown in Table 1. The LUBM provides 14 benchmark queries for testing knowledge-based systems. Two performance parameters, such as data retrieval time and query processing time, are used for evaluation of the proposed distributed reasoning system using MapReduce. Data retrieving time is the time for parsing an RDF document saving of triples in an RDF store. During the evaluation, it is observed that LUBM(5,0) takes 372 seconds for 93 RDF files, LUBM(10,0) takes 854 seconds for 189 RDF files, LUBM(15,0) takes 1708 seconds for 402 RDF files. It is noted that either data loading or retrieval is scalable even size of the RDF dataset increases. Another performance parameter, the query processing time, is the total time for mapping the query to graph-store and fetching time of the RDF store. Queries are executed on a single node machine with a 4-node cluster. In the empirical analysis, we have used two different query systems, Cassandra API to query the data, and other is Cassandra with MapReduce (supports distributed reasoning of queries) and observed that data retrieval time is greatly reduced when using MapReduce model; query responsiveness is much faster in MapReduce model due to its parallelism processing in query exaction on shared data of the multi-nodes system.

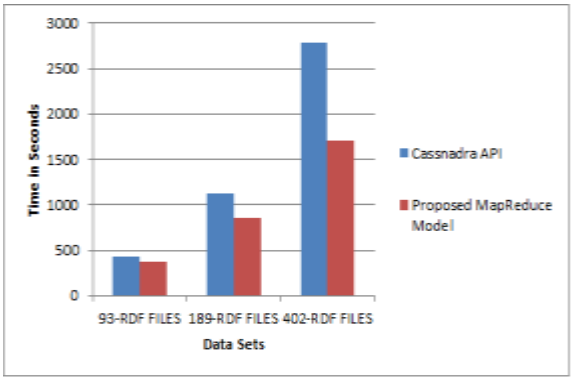

Fig6: Data Retrieval Time

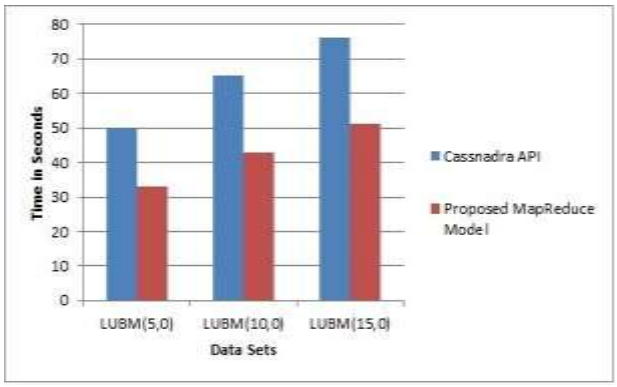

Fig7: Query Processing Time

The performance analysis is visualized in Fig.6 and Fig. 7 for depicting a comparison study between traditional and proposed distributed reasoning models for LUBM datasets.

6. CONCLUSION

This paper presents the proposed distribute reasoning model for supporting effective distribution data storage for large RDF datasets. It processes the queries using the SPARQL query system for handling huge RDF data and perform reasoning operations from Graph-store and retrieves data from the Cassandra storage system. The proposed system achieves the query responsiveness in a faster way than the Cassandra model, and these results are presented in the experimental study for demonstrating the efficiency of the proposed system with respect to data retrieval time and query processing time.

7. References

[1]. M. S. Marshall *et al.*, "Emerging practices for mapping and linking life sciences data using RDF—A case series," *J. Web Semantics*, vol. 14, pp. 2–13, Jul. 2012.

[2]. V. R. L. Shen, "Correctness in hierarchical knowledge-based requirements," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 30, no. 4, pp. 625–631, Aug. 2000.

[3]. K. Rajendra Prasad, C. Raghavendra, Padakandla Vyshnav, "Intelligent System for Visualized Data Analytics A Review," International Journal of Pure and Applied Mathematics, Volume 116 No. 21 2017, 217-224.

[4]. J. Cheng, C. Liu, M. C. Zhou, Q. Zeng, and A. Ylä-Jääski, "Automatic Composition of Semantic Web services based on fuzzy predicate Petri nets," *IEEE Trans. Autom. Sci. Eng.*, Nov. 2013, to be published.

[5]. D. Kourtesis, J. M. Alvarez-Rodriguez, and I. Paraskakis, "Semantic-based QoS management in cloud systems: Current status and future challenges, "*Future Gener. Comput. Syst.*, vol. 32, pp. 307–323, Mar. 2014.

[6]. M. Nagy and M. Vargas-Vera, "Multiagent ontology mapping framework for the Semantic Web," *IEEE Trans. Syst., Man, Cybern. A Syst., Humans*, vol. 41, no. 4, pp. 693–704, Jul. 2011.

[7]. J. Urbani, S. Kotoulas, J. Maassen, F. V. Harmelen, and H. Bal, "WebPIE: A web-scale parallel inference engine using MapReduce," *J. Web Semantics*, vol. 10, pp. 59–75, Jan. 2012.

[8]. J. Urbani, S. Kotoulas, E. Oren, and F. Harmelen, "Scalable distributed reasoning using MapReduce," in *Proc. 8th Int. Semantic Web Conf.*, Chantilly, VA, USA, Oct. 2009, pp. 634–649.

[9]. J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[10]. C. Anagnostopoulos and S. Hadjiefthymiades, "Advanced inference in situation-aware computing," *IEEE Trans. Syst., Man, Cybern. A Syst., Humans*, vol. 39, no. 5, pp. 1108–1115, Sep. 2009.

[11]. H. Paulheim and C. Bizer, "Type inference on noisy RDF data," in *Proc. ISWC*, Sydney, NSW, Australia, 2013, pp. 510–525.

[12]. G. Antoniou and A. Bikakis, "DR-Prolog: A system for defeasible reasoning with rules and ontologies on the Semantic Web," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 2, pp. 233–245, Feb. 2007.

[13]. V. Milea, F. Frasincar, and U. Kaymak, "tOWL: A temporal web ontology language," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 1, pp. 268–281, Feb. 2012.

[14]. D. Lopez, J. M. Sempere, and P. García, "Inference of reversible tree languages," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 4, pp. 1658–1665, Aug. 2004.

[15]. Schlicht and H. Stuckenschmidt, "MapResolve," in *Proc. 5th Int. Conf. RR*, Galway, Ireland, Aug. 2011, pp. 294–299.

[16]. C. Grau, C. Halaschek-Wiener, and Y. Kazakov, "History matters: Incremental ontology reasoning using modules," in *Proc. ISWC/ASWC*, Busan, Korea, 2007, pp. 183–196.

[17]. P. Kiran Kumar, C. Raghavendra, Dr. S. Sivasubramanyan, "Exploring Multi-Scale Mathematical Morphology for Dark Image Enhancement," *International Journal of Pharmacy and Technology, Dec-2016, Vol. 8, Issue No.4, 23590-23597.*

[18]. M. J. Ibá˜nez, J. Fabra, P. Álvarez, and J. Ezpeleta, "Model checking analysis of semantically annotated business processes," *IEEE Trans. Syst., Man, Cybern. A Syst., Humans*, vol. 42, no. 4, pp. 854–867, Jul. 2012.

[19]. *Linking Open Data on the Semantic Web* [Online]. Available: http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/DataSets/Statistics

[20]. J. Weaver and J. Hendler, "Parallel materialization of the finite RDFS closure for hundreds of millions of triples," in *Proc. ISWC*, Chantilly, VA, USA, 2009, pp. 682–697.

[21]. J. Guo, L. Xu, Z. Gong, C.-P. Che, and S. S. Chaudhry, "Semantic inference on heterogeneous e-marketplace activities," *IEEE Trans. Syst., Man, Cybern. A Syst., Humans*, vol. 42, no. 2, pp. 316–330, Mar. 2012.

[22]. M Madhu Bala and Rohit Dandamudi, "HUPM: Efficient High Utility Pattern Mining Algorithm for E-Business," 2018 IEEE 8th International Advance Computing Conference (IACC), Greater Noida, India, *IEEE Xplore*, pp. 191-195, 2018. DOI: 10.1109/IADCC.2018.8691944