Singaporean Journal of Scientific Research(SJSR)
Journal of Technology and Engineering System(JTES)
Vol.8.No.3 2016 Pp.199-206
available at :www.iaaet.org/sjsr
Paper Received : 08-03-2016
Paper Accepted: 19-04-2016
Paper Reviewed by: 1.Prof. Cheng Yu 2. Dr.M. Akshay Kumar
Editor : Dr. Chu Lio

AN INCREMENTAL AND DISTRIBUTED INFERENCE METHOD FOR LARGE-SCALE ONTOLOGIES USING ONE-CLASS CLUSTERING TREE

A.Vijayalakshmi[1], Dr.S.Babu[2]
[1]PG Scholar, [2]Associate Professor
[1,2]IFET College of Engineering Villupuram, India.

ABSTRACT— Reasoning on a Web scale becomes increasingly challenging because of the large volume of data involved and the complexity of the task by means of ontology mapping. Here, an IDIM concept is used to deal with large-scale incremental RDF datasets. Resource Description Framework (RDF) is an important data, presenting standard of the semantic web to process the increasing RDF data. Map Reduce is a widely-used parallel programming model that can be used to represent uncertain similarities created by both syntactic and semantic similarity algorithms. The proposed One-Class Clustering Tree (OCCT) characterizes the entities by identifying different entries that should be linked together. The construction of TIF and EAT significantly reduces the re-computation time for the incremental inference as well as the storage for RDF triples. Therefore, users can execute their query more efficiently without computing and searching over the entire RDF closure used in the prior work. The final results are evaluated by comparing it against benchmark models in web information gathering.

INDEX TERMS— Ontology reasoning, RDF, MapReduce, IDIM, Hadoop, OCCT, MLE.

I. INTRODUCTION

Semantic reasoning of data on a Web raises the task to tedious process. Ontology mapping in the context of Question Answering can provide more correct results if the mapping process can deal with unreliability that is caused by the incomplete and inconsistent information used and produced by the mapping process.

In the year of 2009, the semantic web [2] contains 4.4 billion triples and has now reached over 20 billion triples. Its growth rate is still increasing. As it has evolved into a global knowledge-based framework to promise a kind of machine intelligence, supporting knowledge searching over such a big and increasing dataset has become an important issue. Resource Description Framework (RDF) is a data representation standard that plays a vital role to describe knowledge in the semantic web. Deriving inferences in the large-scale RDF [1] files, referred to as large-scale reasoning, poses challenges in three aspects:

i. Distributed data on the web make it difficult to acquire appropriate triples for appropriate inferences.
ii. The growing amount of information requires scalable computation capabilities for large datasets.
iii. Fast processing for inferences is required to satisfy the requirements of online query.

MapReduce can provide a solution for large scale RDF data processing which is a widely-used parallel programming model. It presents a novel approach can be used to represent uncertain similarities created by both syntactic and semantic similarity algorithms. In order to store the incremental RDF triples more efficiently, two novel concepts, transfer inference forest (TIF) and effective assertional triples (EAT) are used. Their use can largely reduce the storage and simplify the reasoning process. Based on TIF/EAT, we need not compute and store RDF closure and the reasoning time, so significantly decreases that a user's online query can be answered timely, which is more efficient than existing methods to our best knowledge. More importantly, the update of TIF/EAT needs only minimum computation since the relationship between new triples and existing ones is fully used.

II. RELATED WORKS

The prior methodologies used for semantic web search are discussed as follows:

A. Fuzzy Set Theory

Anagnostopoulos et al. proposed the method of fuzzy set theory where Context awareness (CA) is a very important computing paradigm. Context is an information that can be used to characterize the situation of a person, place, or object that is considered relevant to the integration between a user and an application, including the user and the application themselves. CA is the ability of a system to sense, interpret, and react to changes in the environment a user is situated in. The capability of a context (or situation)-aware system [6] to classify context and infer specific situations can be facilitated by proper knowledge-representation (KR) models. A Fuzzy-set-based model can accommodate the vagueness inherent in context capturing. A fuzzy set is used for representing imprecise context in a human understandable form. This methodology is generic and can be applied to different inference schemes in order to improve the inference capability of the classifier and deal with mutual-exclusion inference. This model generates specific complementary fuzzy rules used for increasing the accuracy of the classification process for the well-specified information in Semantic web.

Disadvantage:

Applications can handle context as flexibly as their users would expect by using this method, but it is not suitable for all situations of user.

B. RuleXPM

Guo et al. introduced a novel RuleXPM [2] (XML Product Map) approach is an integrated model that combines a set of representations of various types of concepts, some e-marketplace participating systems, and an inference process. The method consists of several major constituents that include a collaborative ConexNet (Concept exchange Network), an e-marketplace network (EMpNet), and an inference engine.

Disadvantage:

Although this method is interoperable and inferred from one entity to another, it is not possible to implement it on an automated offering system and an automated negotiation system.

C. Similarity Transition

Paulheim et al. presented a method of similarity transition [7], a linked dataset is a kind of labeled directed graph cross domain, which is used for knowledge presentation and cognitive model foundation. Each link represents a kind of relationship between two resources In these statements, the similarity between two subjects can be

calculated from the similarity between their corresponding sets of objects.

If the linked dataset is considered as a whole semantic graph, the similarity between the related subjects more accurate.

This calculation is referred as similarity transition that utilizes node and link types together with the topology of the semantic graph to derive a similarity graph from linked datasets. This method enables smooth interaction and visualization of the similarity graph which is derived based on the calculated similarity of two resources.

Disadvantage:

The effectiveness of this method is less as the similarity weight of each link type is given by experience.The above described methods are applicable for small databases. To deal with a large base, some researchers turn to distributed reasoning methods.

D. Parallel Materialization

Weaver et al. introduced Parallel Materialization [13], where the finite RDFS is the first method to provide RDFS inference on such large data sets in such low times and scalable manner. This maintains soundness and completeness without requiring any cumbersome preparation of the data.

Disadvantage:

It locks with scalability and expressivity.

E. Scalable Distributed Reasoning

Urbani et al. presented Scalable distributed reasoning method [4] which constitutes some non-trivial optimizations for encoding the RDFS ruleset in MapReduce and exploits the MapReduce [5] framework for efficient large-scale Semantic Web reasoning and implements on the top of Hadoop. This reasoning technique performs quick reasoning using HDFS and high data correlation.

Disadvantage:

It does not focus on quality of reasoning.

F. MapResolve

Schlicht et al. introduced an novel approach of MapResolve [10] that solves the problem by adapting the standard method for distributed resolution that avoids repetition of resolved inferences. For the limited expressivity of RDFS, the repetition can be avoided because every MapReduce job is executed only once.

Disadvantage:

For each iteration, the clause sets are parsed and written to disc by generating needless overhead.

G. WebPIE

Urbani et al. presented a scalable parallel inference method named WebPIE [3], is a Web-scale Parallel Inference Engine using MapReduce. This method calculates the RDF closure based on MapReduce for large-scale RDF dataset by adopting algorithms to process the statements based on input data as incremental reasoning. This technique identifies the accurate status, which does either exist or new ones.

Disadvantage:

It does not provide the relationship between the newly arrived and existing data.

However, the distributed reasoning methods considered no influence of increasing data volume and did not answer how to process users' queries. As the data volume increases and the ontology base are updated, these methods require the re-computation of the entire RDF closure every time when new data arrive. To avoid such time-consuming process, incremental reasoning methods are proposed.

H. Incremental Ontology Reasoning

Grau et al. proposed an Incremental Ontology Reasoning approach [12] based on modules that can reuse the information obtained from the previous versions of an ontology which is best suitable for OWL.

Disadvantage:

Reasoning speed is a huge problem while using this method.

III. EXISTING SYSTEM

In Existing system, the proposed concept of an incremental and distributed inference method [15] for large-scale ontologies using MapReduce realizes high-performance reasoning and runtime searching, especially for incremental knowledge base. By constructing, using novel concepts of transfer inference forest and effective assertional triples, the storage is largely reduced and the reasoning process is simplified and accelerated to satisfy end-users' online query needs. The processing was made via MapReduce, which is motivated by the fact that it can limit data exchange and alleviate load balancing problems by dynamically scheduling jobs on computing nodes.

Drawbacks of Existing System are as follows,

- The Query time for IDIM is affected when the incremental triples affect the structure of the inference forests.
- If an RDF dataset has few ontological triples, the size of constructing dataset TIF is also small.
- The changes in the structure of TIF affect the performance improvement with ontological triples.
- The advantages of TIF/EAT cannot be exploited well, if the size of the tree is small.

IV. PROPOSED METHODOLOGY

In order to overcome the existing drawbacks, the data clustering method is used in this paper that makes the processing of data more efficiently by means of linking the data sets.

A. One-Class Clustering Tree (OCCT)

A clustering tree is a tree in which each of the leaves contains a cluster instead of a single classification. Each cluster is generalized by a set of rules that is stored in the appropriate leaf. This data linkage method aimed at performing one-to-many linkage. The data linkage is performed among entities of different types.

For example, in a student database, we might want to link a student record with the courses she should take. It is done according to different features which describe the student and features describing the courses.

The OCCT [11] was evaluated using datasets from three different domains. They are

- Prevention from data leakage
- Acts as a recommender system
- Avoiding deception.

In the data leakage prevention domain, the goal is to detect abnormal access to database records that might indicate a potential data leakage or data misuse. The goal is to match an action, performed by a user within a specific context, with records that can be legitimately retrieved within that context.

In the recommender systems domain the proposed method is used for matching *new* users of the system with the items that they are expected to like based on their demographic attributes.

In the deception avoidance domain, the goal is to identify online purchase transactions that are executed by a fraudulent user and not the legitimate user.

The results show that the OCCT performs well in different linkage scenarios. In addition, it performs at least as accurate as the well known as decision tree data-linkage model, while incorporating the advantages of a one class solution. Additionally, the OCCT is preferable over the decision tree because it can easily be translated to linkage rules.

B. Algorithm: Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE) splitting criterion used in order to choose the attribute that is most appropriate to serve as the next splitting attribute. Each candidate attributes from the set of attributes splits the node data set into subsets according to its

possible values. For each of the subsets, a set of probabilistic models is created, one for each attribute of second dataset. Each probabilistic model is built to describe the probability given. In order to create the probabilistic models decision tree are used. Each of these trees represents the probability of its class attribute values given the values of all other attributes.

Once the set of models has been induced, the probability of each record given these models is calculated. A subset's score is calculated as the sum of all scores of the records belonging to it. The attribute's final score is determined by the sum of the subset's individual scores. The goal is to choose the split that achieves the maximal likelihood and therefore we choose the attribute with the highest likelihood score as the next splitting attribute in the tree. The computational complexity of building a decision model using the MLE method is dependent on the complexity of building a statistical model and the time it takes to calculate the likelihood.

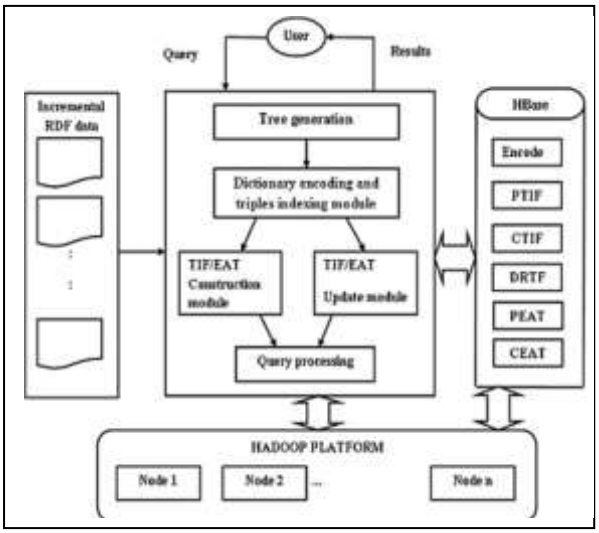

Figure.1 Overall System Architecture

In figure1, the input incremental RDF datasets are received by the core module IDIM of the system and it process the triples and performs the reasoning. It interacts with HBase for storing or reading the intermediate results and returns the query results to end-users. The HBase is designed with six tables to store the encoded ID, PTIF, CTIF, DRTF, PEAT, and CEAT.

The Hadoop framework is an open-source Java implementation of MapReduce that allows for the distributed processing of large data sets across clusters of computers. It can scale up from single server to thousands of machines by offering local computation and storage and manages execution details such as data transfer, job scheduling, and error management.

ADVANTAGES OF PROPOSED SYSTEM

- The OCCT model is better generalized and avoids over-fitting by means of pruning the data.
- Fraud detection is used to obtain the genuine matching data for legitimate users to access.
- Maximum Likelihood Estimation can handle multiple ways of splitting the data entities.
- It is easy and quick method to compare the datasets by obtaining the matching entities.

V. SYSTEM IMPLEMENTATION

The proposed system includes the following modules

A. Fetch User Query

Search engines rely on user to submit information that is subsequently indexed and catalogued. A query is a request for information from a search engine send by the user. When the users query a search engine to locate information, they are actually searching through the index that the search engine has created. The users' query information is collected and it is send to the server for analyzing. The information is retrieved by means of opinion mining which is the tone behind a series of words useful for monitoring as it allows us to gain an overview of the user opinion.

B. Tree generation

The user query has been formed as a Tree which is represented as an undirected graph by merging both user and database objects through the method of one-class clustering tree. This method links the data

sets and matches the entities by means of Maximum Likelihood Estimation splitting criterion and pruning is done in order to avoid the over-fitting as it removes the unnecessary entities. After creating the tree, we apply triple indexing method.

C. Resource description framework construction

The tree created will be converted as RDF data sets. RDF data usually contain many statements made of terms that are long sequences of Characters, their processing and storage have low performance as new RDF data arrives continuously. The dictionary encoding and triples indexing module encodes all the input triples into a unique and small identifier to reduce the physical size of input data and for each triple an index is built based on inverted index method. After that the incremental triples are separated into the incremental ontological triples and incremental assertional ones.

D. Query analysis

Query analysis improves the performance of query processing, which speeds up many database functions and aspects. A query optimizer analyzes a specific query statement and generates both remote and local access plans to be used based on the resource cost of each plan. The query given by the user is analyzed for the results by comparing with the tree generated by the server. It means that the requested query is compared and required data is extracted with matched entities as results. The given user query are being refined by changing or adding to the set of search terms to a better job of returning the pages user is seeking.

E. User's Query Evaluation

The extracted tree is structured to form XML code that places the root node as first element in the xml tag that follows parsing across the other nodes correspondingly. The generated code is executed to produce a link that leads the user to a dynamically designed web page that provides necessary information with respect to user's request. The final result is being brought to the user as link ordered by what it considers the items' relevance to the query, listing the best match first. This guides the user to determine if a page includes the information user is seeking or links to it and the results also gets stored in the database.

VI. RESULTS AND DISCUSSION

In table1, BTC dataset shows the sensible representation of semantic web. This can be used to deduce statistics for entire data. BTC consists of five major large datasets in which each constitutes several smaller ones.

Table:1 Basic Information of BTC Dataset

Dataset	No.of triples	Schema type		
		Domain & Range	*Sub-Property*	*Sub-Class*
Datahub	910078982	36338	15068	26146
Freebase	101241556	1	0	0
DBpedia	198090024	1136	0	275
Timbl	204806751	55086	24431	291095
Rest	22328242	2905	746	30373
OVERALL	1436545555	95466	40245	347889

We compare IDIM with WebPIE that acts as a state-of-the art for reasoning the RDF datasets in order to show the recital of our method. As the purpose of this paper is to speed up the query for users, we use WebPIE to produce the RDF closure and then search the related triples as the output for the query. The Hadoop configurations are the same to that in IDIM. Then the contrast can be concerted on the dissimilarity of reasoning methods.

Table:2 Reasoning results (Eight Nodes)

Dataset	No.of triples	TIF/EAT time (min)	No. Of RDF closure triples	RDF closure time (min)
Datahub	713574291	52	1079343655	71
Freebase	94134030	10	101241556	12
DBpedia	133242743	23	198091689	30
Timbl	114130464	24	326688386	33
Rest	17073633	7	26287842	9
OVERALL	1072155161	116	1731653128	155

The two methods WEbPIE and IDIM were run three times on each dataset and corresponding output triples were calculated along with the time it takes for reasoning. In table2, it shows the data of TIF/EAT which is related to IDIM and RDF closure data that relates to performance of WebPIE. Both the output triples are calculated and time consumed is recorded. By means of these results obtained while comparing both the methods, we can conclude that reasoning time for IDIM is less than WebPIE and the output triples are also much few lesser from WebPIE which in turn lesser than in original dataset. Note: The results are obtained when we use eight computing nodes in parallel.

A. Performance Evaluation

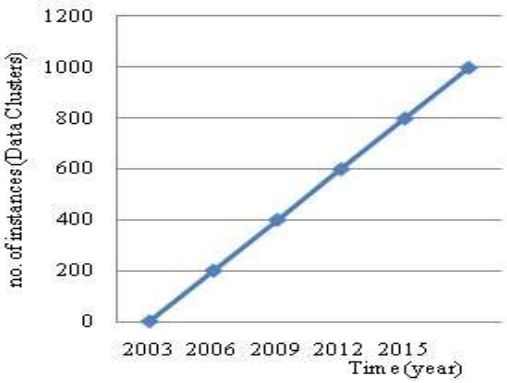

Fig.ure2 Map Reduce instances over Time

Figure 2 describes the significant growth of MapReduce against data clusters over time, from 0 in the beginning to nearly 1000 separate instances as of the late 2015.

The MapReduce library logs statistics about the computational resources used by the job at the end of each job. MapReduce can perform effectively even for a simple program that runs competently on thousand of machines that significantly quickening the development cycle.

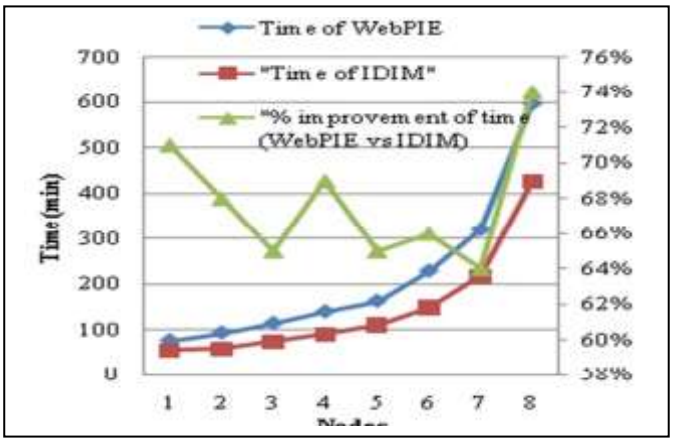

Figure.3 processing time on Different nodes

With the purpose of evaluating the scalable performance of IDIM against WebPIE, we take Freebase dataset as an example and we increase the number of nodes from 1 to 8 which reports the time for the reasoning in Figure 3. Clearly, the

increasing number of nodes can speed up the reasoning. IDIM achieves more performance enhancement than WebPIE. In particular, it needs roughly 68% of the processing time of WebPIE.

VII. CONCLUSION

With the upcoming data deluge of semantic data, the fast growth of ontology bases has brought significant challenges in performing efficient and scalable reasoning. Mapping process can deal with the uncertainty effect that is caused by the incomplete and inconsistent information used and produced by it for processing users' queries that can provide more correct results. MapReduce represents uncertain similarities created by both syntactic and semantic similarity algorithms. OCCT characterizes the entities that should be linked together using the splitting criterion of MLE. TIF and EAT construction significantly reduces the re-computation time for the incremental inference as well as the storage for RDF triples. Therefore, users can execute their query more efficiently without computing and searching over the entire RDF closure.

VIII. FUTURE SCOPES

In the future, the methods can validate for more datasets, such as other benchmarks and other types of datasets and also can be done in other ontology languages [9] that make the processing of data to the user's request in a highly efficient manner.

IX. REFERENCES

[1] N 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, 2012.

[2]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.

[3]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.

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

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

[6]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, Sept. 2009.

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

[8]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.

[9]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.

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

[11] Ma'ayanDror and AsafShabtai, "OCCT: A One-Class Clustering Tree for One-to-many Data linkage," *IEEE trans. on knowledge and data engineering*, tkde-2011-09-0577, 2013.

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