ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 8, February 2015

Discovery of Semantic Relationships over Distributed Networks

Sadhana Singh and A. Razia Sulthana
M.Tech Information Technology, SRM University, India.
Faculty of Engineering and Technology, Department of Information Technology, SRM University, India

Abstract*— The large volume of Web data has brought potential vast amount of knowledge. Semantic this means meaning, plays an important role in Semantic Web Technology. As data is been reposited all over Web in different datasets server, it become highly important to discover the relevant data for the purpose of many applications such as hospitality, medical, banks, security etc. Discovering such vast data around the Web and finding relationships between two different data from distributed datasets is a challenging task which includes isolation and scalability. This paper proposes a technique to discover semantic relationships between the data's through knowledge abstraction and nearest path discovery technique. The approach will help in finding out the common things between the two data's which can solve many real time applications.

***Index Terms*—Semantic Web, Semantic relation, Distributed network.**

I. INTRODUCTION

The development of Web technology has been increasing as the demand of data has tremendously increased. Semantic Web technologies has brought demand for more and more Semantic data which can applied in many Web applications. To effectively utilize the large amount of semantic data, efficient search mechanisms customized for Semantic Web data, especially for ontologies, have been proposed for both humans and software agents [1]. Many Semantic search engines have been developed [2], [3], [4] which retrieves the data from the datasets. It has helped in many data mining applications. But there are hidden information still in the datasets which is of quite important and can used in many real time applications like hospitality, medical and banks. For example how two people are related to each other i.e. common entities. In security concern, if we want to check the common entities between the two people, the aforementioned approaches fail to do it. Automatic discovery of semantic relationships between entities is a key issue in analytical domains, such as business intelligence, and homeland security, where "...the focus is on trying to uncover obscured relationships or associations between entities and very limited information about the existence and nature of any such relationship is known to the user..." [5]. we can find many applications in the literature that are similar to the above mentioned context. For example, detecting conflict of interest (COI) relationships among potential reviewers and authors of scientific papers [6] and detecting a connection between two suspected passengers on the same flight in the context of aviation safety [7]. Analysing the single knowledge base where only limited amount of knowledge is gained which sometimes are insufficient for some of the applications. So it's necessary to search all the repositories throughout the network. As data are stored by organizations; it becomes impractical to merge all the data which are dispersed throughout the network. Many organizations may not allow relocating or merging data because of legal reasons and business secrets [1]. Sometimes the knowledge from single datasets is not sufficient to solve many applications. So, there is a demand for the huge data which are reposited in distributed network. Analysing such huge data brings many challenges: 1) Less scalability due to the complexities of the query. 2) Finding relationship between two entities demand high level framework. 3) Deeper understanding for such semantic data. To overcome these challenges, a previous report [7] on discovering semantic relations which are scattered over the network. That work presents a super peer-based approach to discover semantic relations in a peer-to-peer (P2P) network environment. In the proposed system, peers register with superpeers that are also peers but likely to be more powerful in terms of computing capacity, memory storage, and bandwidth. Superpeers are connected with each other through semantic links. The system is built on the assumption that each superpeer knows how to reach other superpeers. Therefore, relational discovery can be performed by finding semantic paths at the superpeer layer. However, it was unspecified: 1) How do the related nodes can locate the same superpeer? 2) How do superpeers communicate in the network? As a consequence, it is difficult to guarantee the scalability of the system without the design of aforementioned communication components. In this paper we have proposed the technique which overcomes the above challenges. It helps in breaking down the traditional search which was centralised. The proposed approach is decentralised, it helps in efficiently solve semantic relation discovery problem, and it improves the traditional work. The layout of the paper is as follows: Section II discusses about the Design Framework. Section III explains about Path discovery. Section IV presents Experiments and Results. Section V concludes with the Conclusion.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 8, February 2015

II. DESIGN FRAMEWORK

In this section, we represent an overview of the architecture which resembles with real time distributed topology.

A. *Design Overview*

Fig.1. illustrate the architecture of proposed system. As the architecture can be assumed as the real time scenario, where node *a*, node *b*, node *c*, node *d*, node *e* acts as knowledge bases which are distributed via network links. Each node is interconnected with each other via networks and even the distance cost between the nodes has been taken randomly. Each node while sending request to its neighbour nodes calculate the cost distance and update its database. This will help in finding the shortest path to retrieve results. The distance cost factor helps in calculating the smallest path from source to destination. The path finding is constricted into three steps: 1) Locate the source and the destination. 2) Search all the intermediate nodes that arrive between source and destination. 3) Retrieve the path according to the cost distances. This helps in much faster search. The topology adaptation is a process of finding semantically related neighbors. This process can be performed at different states of the life time of a node. That is: 1) on joining the network, a node may choose neighbors based on their semantic similarities, or 2) after updating its knowledge base, a node will revaluate its neighbour relationships to connect to most related peers. This methodology ensures that the underlying node topology will always reflect the changing interests and data contents [1]. The topology adaptation is referred from [8] and [9]. The above architecture design is the small illusion or setup of the real time system which gives an idea how it can help in executing in real time.

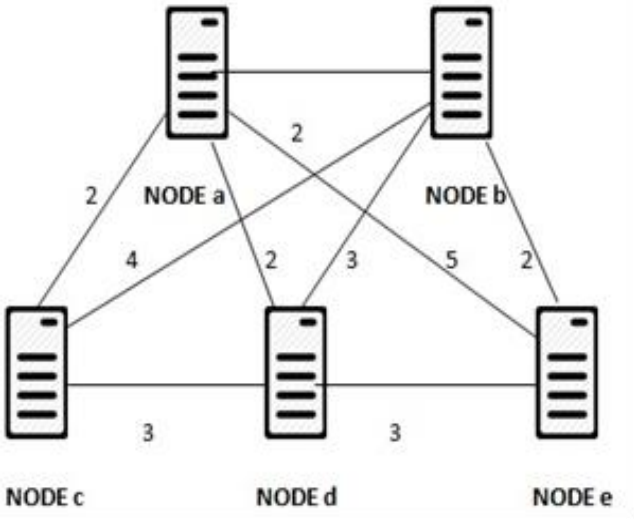

Fig.1. Architecture of proposed system.

The difficulties are quite less as compared with the real time system. It becomes tedious to work with real time. This small illusion gives us the idea about how it will work exactly and what output it can generate while working in real time. As in this paper we are just concentrating on the nearest path and how fast it will perform the output. The idea to work in real time system comes with different technique because it deals on network. As even we have proposed some of the strategy which can be helpful in implementing with real knowledge bases.

III. PATH DISCOVERY

To locate path between the source and the destination, the distance between them should be precomputed. In reality, we cannot add weights; it should be based on trust between the knowledge owners [1]. The path discovery can be computed by adding weights or cost on all the links connecting different knowledge bases. As in Fig.1. node *a* is linked to node *b*, node *c*, node *d* , node *e* through some common entities in datasets. The common entities that are mapped to node a are gateway nodes of node a. For example *min_dist*(*a to b*)=2; *min_dist*(*a to e*)=5. In real time designing the knowledge bases can use the technique of link state protocol where each node can send its link status to all the other nodes in the form of link state updates. The other nodes can check whether the coming request is new or the one which initiated the request by checking its table. Through the hop count which means how far one node from another node is, it can calculate the distance cost. The hop count limit should be *k.* As its vast network it has long paths which is difficult to explain and understand. In our experiment, the default value of k is equal to 6.

IV. EXPERIMENTS AND RESULTS

A. *Experimental Setup*

The above experiment is divided into four steps:
1) Deploying.
2) Loading data in respective datasets.
3) Finding out the nearest distance.
4) Query.

The first step is deploying. This process starts with deploying each datasets or even we can name as server which act as datasets of some location or area. Likely, we can add *n* numbers of datasets just for the execution of the above method. Eclipse is an integrated development environment (IDE) which is used to build applications. It uses java to build most of the applications. Even other programming languages like C, C++ etc. can be used to build applications. Through Eclipse software we have designed an user interface where we deploy server, as named above. For the experiment Purpose we have taken or deployed 8 servers. The next step is inserting the data into the database of above defined datasets. For building database we have used MySQL database to enter data to each servers. This database will act as a back end to our system where the data is retrieved, stored, updated, and deleted. The database which is been used in this experiment consists of name, age, blood group, country, state, title etc. Each of these groups in the database has been given corresponding values. This is to be followed for another datasets. Once all the data in the datasets has been uploaded,

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 8, February 2015

we can move on to the next step; calculating the distance cost from one datasets to another datasets. As by default we have fixed some distance cost value. This will calculate the entire nearest node it has with the starting node. In real time as the data is vast and is distributed completely over the network, so it would be tedious and can face network traffic. This can lead to slow down query processing. To solve the above mentioned problem, we can set an threshold value while picking up the path. If the value exceeds the value, it stops the further search and if it is less than the defined value, it starts searching the data with the nearest distance node. This feature can help in solving the aforementioned problem in real time applications. The last step is user query. As our experiment is finding out common relations between two data from different datasets. The following experiment can be best explained with a small example. To make things simpler we had used "*and*" connector between two data while inserting query. This helps in distinguishing one data from another data. Consider two person names Roha and Nitin which resides in two different datasets. After deploying and finding the nearest distance cost, if we give the query as Roha *and* Nitin, it will start from the start node and sends request to all the nodes and according to the nearest distance cost it will search that all datasets, if it find data related to their names and also some common relation between them such as school, blood group, age etc. It will give the results what common entities lies between the two data. If it fails to search and couldn't match any of the data in the datasets no results will be shown. The main advantage of this technique firstly it will send request to all the nodes and simultaneously calculate the path cost. After data are found in their respective datasets, it will take the shortest path to get the data. This helps in faster data searching and even finding common entities between them which can be helpful in much different kind of real time applications.

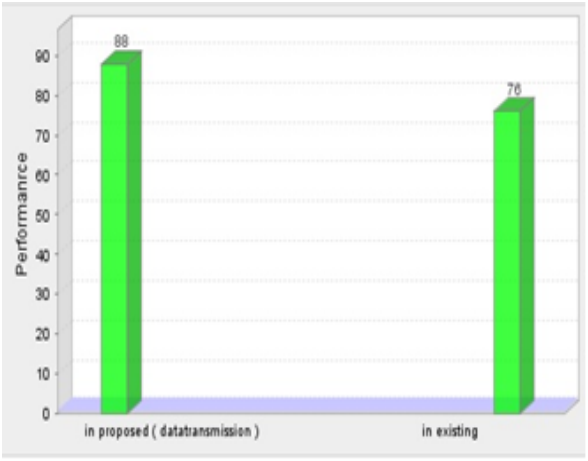

Fig.2. Performance of end results.

B. Results

The output will show the common entities found between two different data such as blood group, college name, city which has been used in different datasets. This helps in finding common entities or relationship which helps in many real time applications. This technique can be applied with the real time datasets which are distributed throughout network. The performance of the end results can be depicted in Fig.2. where we can see the performance rate has increased as compared to the existing system. This performance can even be checked with the real time applications.

V. CONCLUSION

In this paper, we presented a scalable approach to discover common relation between two data from distributed knowledge bases. This approach helps in discovering new knowledge which can help in many real time applications. As the above approach was just the small demonstration about how it can be done with real time knowledge base which are scattered over the internet. This paper is just the small illusion which we can apply on real datasets which are distributed throughout the network. The experimental results help in scalability and efficiency of the framework. The future work will be enhancing the above technique by adding security and privacy into system. This helps in gaining the security concepts such as authentication and authorization, where it can't violates organization confidential data. Only through authenticating the datasets the above experiment can be proceeded.

REFERENCES

[1] Juan Li, Hui Wang, Samee Ullah Khan, Qingrui Li, Albert.Y.Zomaya,"A Fully Distributed Scheme for Discovery of Semantic Relationships", IEEE transaction on service computing, Vol. 6, No. 4, October-December 2013.

[2] L. Ding, T. Finin, A. Joshi, R. Pan, R.S. Cost, Y. Peng, P. Reddivari,V. Doshi, and J. Sachs, "Swoogle: A Search and Metadata Engine for the Semantic Web," Proc. 13th ACM Int'l Conf. Information and Knowledge Management, 2004.

[3] R. Guha, R. McCool, and E. Miller, "Semantic Search," Proc. 12th Int'l Conf. World Wide Web (WWW '03), 2003.

[4] W3C, "SPARQL Query Language for RDF," http://www.w3.org/TR/ rdf-sparql-query/, 2013.

[5] K. Anyanwu and A. Sheth, "p-Queries: Enabling Querying for Semantic Associations on the Semantic Web," Proc. 12th Int'l Conf. World Wide Web (WWW '03), 2003.

[6] B.A. Meza, M. Nagarajan, C. Ramakrishnan, L. Ding, P. Kolari, A.P. Sheth, I.B. Arpinar, A. Joshi, and T. Finin, "Semantic Analytics on Social Networks: Experiences in Addressing the Problem of Conflict of Interest Detection," Proc. 15th Int'l Conf. World Wide Web (WWW '06), May 2006.

[7] A. Sheth, B. Aleman-Meza, I.B. Arpinar, C. Halaschek, C. Ramakrishnan, C. Bertram, Y. Warke, D. Avant, F.S. Arpinar, K. Anyanwu, and K. Kochut, "Semantic Association Identification and Knowledge Discovery for National Security Applications," J. Database Management, vol. 16, no. 1, pp. 33-53, Jan.-Mar. 2005.

[8] J. Li and S. Vuong, "SOON: A Scalable Self-Organized Overlay Network for Distributed Information Retrieval," Proc. 19th IFIP/ IEEE Int'l Workshop Distributed Systems:

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 4, Issue 8, February 2015

Operations and Management: Managing Large-Scale Service Deployment (DSOM '08), pp. 1-13, 2008.

[9] J. Li, "Grid Resource Discovery Based on Semantically Linked Virtual Organizations," J. Future Generation Computer Systems, vol. 26, no. 3, pp. 361-373, 2010.