Distributed Discovery of Semantic Relationships

Juan Li [1] and Hui Wang[2]

[1]Computer Science Department, North Dakota State University, ND 58108, USA
[2]Computer Science Department, Stevens Institute of Technology, NJ 07030, USA

The availability of large volumes of Semantic Web data has created the potential of discovering vast amount of knowledge, among which semantic relation discovery fundamentally changes the way we acquire and use knowledge. Due to the decentralized and distributed nature of Semantic Web development, semantic data tend to be created and stored independently in different organizations; under such circumstances, the full exposition of the Semantic Web data faces numerous challenges such as usability, scalability, and heterogeneity. This paper proposes an effective strategy to discover semantic relationships over large-scale distributed systems such as peer-to-peer and grid network, which allows users to share their local knowledge to collectively make new discoveries.

***Index Terms*—distributed system, knowledge, relation discovery, Semantic Web.**

I. Introduction

With the development of semantic web technologies, more and more semantic web data are generated and widely used in Web applications and enterprise information systems. Semantic web data are structured with ontologies for the purpose of comprehensive and transportable machine understanding. Ontologies contain millions of entities interconnected by meaningful relationships. Automatically 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." [1]

Currently, most researches on semantic association query and discovery assume there is a global dataset, where all the entities and relationships are available for analysis. However, such assumption is not feasible in practice. On the other hand, analyzing a local knowledgebase can only obtain limited knowledge. With more knowledge and data sources created and owned by geographically distributed organizations, there is an increasing demand to collectively discover knowledge from distributed sources. This task brings a number of new challenges: (1) Due to the lack of global view or unified understanding of the distributed semantic data, it is difficult to achieve the global optimum of semantic association discovery with dispersed local operations. (2) The relationships between two entities may span over multiple distributed knowledge bases, which requires efficient communication protocols to forward search requests between knowldgebases and later to gather results. (3) It is difficult to achieve scalability and low latency for even simple entity queries in large-scale distributed system, not to mention the complex relationship queries.

To our knowledge, there are few researches on discovering semantic relationships in distributed environment. The most important work was proposed by Perry et al [2]. In their paper, the authors presented a super-peer-based approach to discover semantic relations in a peer-to-peer (P2P) network environment. In the proposed system, peers register to super peers. Super peers connect with each other through semantic links. The paper assumes each super-peer knows how to reach other super-peers. Therefore, relation discovery can be performed by finding semantic paths at the super peer layer. However, how related nodes can locate the same super-peer and how super-peers communicate in the network were unspecified. It is difficult to guarantee the scalability of the system without the design of these communication components.

In this paper, we focus on a discovery system that goes beyond the centralized scheme to a decentralized and distributed strategy. The work presented aims to support discovering of semantic relationships over geographically distributed knowledgebases on an unprecedented scale. The approach we proposed is fully decentralized and scalable. It not only efficiently solves the semantic relation discovery problem, but also improves the traditional search and discovery of semantic knowledge, thus improving the effectiveness and efficiency of semantic sharing in general.

The rest of the paper is organized as follows. Section II introduces the concept of "semantic relationship". Section III describes the architectural and system design of the discovery system. Related work and concluding remarks are provided in Sections IV and V, respectively.

II. Semantic Relation

The Resource Description Framework (RDF) is a World Wide Web Consortium (W3C) recommendation for describing Web resources. RDF provides a basic data model, like the entity-relationship model, for writing simple statements about Web objects (resources). RDF can make statements about resources in the form of *subject-predicate-object* expressions, called *triples* in RDF terminology. The *subject* denotes the resource which has a Universal Resource Identifier (URI). The *predicate* denotes traits or aspects of the resource and expresses a relationship between the *subject* and the *object*. *Predicates* in RDF are also identified by URIs. The *object* is the actual value, which can either be a resource or a literal. RDF can represent simple statements about resources as a directed, labeled graph with typed edges and nodes. In this model, a directed edge labeled with property name connects

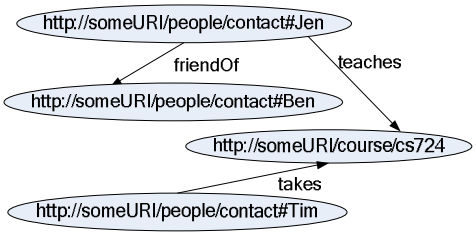

Fig. 1. Example of part of an RDF graph

the *subject* to the *object*. For example, the group of statements, "there is a person identified by *http://someURI/contact#Jen*. She is a friend of another person Ben and she teaches a course CS724 which a student Tim takes." could be represented as the RDF graph in Fig 1.

As shown in Fig. 1, paths of RDF graph can represent semantic relationships among the participating resources (entities), explicitly or implicitly. In [1], Anyawu and Sheth proposed ρ-path query as a way of expressing semantic associations between entities in RDF graph. A path $P=e_1,p_1,e_2,p_2,e_3,\ldots,e_{n-1}p_{n-1},e_n$ is a sequence of RDF statements where each e_i, p_i, e_{i+1} represents a single statement in which p_i is the rededicate and one of e_i or e_{i+1} is the subject and the other is the object. While a ρ-path has been defined as a directed path in [1], we treat paths as undirected in our paper for the following reasons: (1) this may simplify the path discovery. (2) As long as two entities are connected by undirected paths, they are semantically associated, even though there is no directed path connecting them. For example, in Fig. 1, Jen and Tim are related although there are no directed paths between them. We can consider directions of the edges after undirected paths have been located. Therefore, in this paper, two resources x and y are said to be ρ-path associated if there exists an undirected path P of length $n>0$ between them.

III. Discovery Framework

In this section we present the system design and the details of the relationship discovery plan, more specifically, how to efficiently find paths between two semantic entities from distributed knowledgebases.

A. Design Overview

Path discovery is much more difficult than entity discovery, because it needs to locate not only the entities but also all paths connecting them. Our solution is inspired by the strategy of Internet routing. Considering Internet routing that is scalable with millions of nodes, our semantic graph is very similar to Internet in that both are large-scale including millions of nodes and edges, and both are distributed without a global view at any individual node. Therefore, we believe that we can adopt a similar abstraction strategy for our semantic path-finding.

Given detailed links of the whole Internet, it is too expensive, if not impossible at all, to compute the route between nodes. Instead of working at such a low level of details, Internet routing is planned at the Autonomous System (AS) level. Autonomous System (AS) corresponds to an administrative domain. Once the path reaches an AS border, the best route is computed from AS to AS. The Border Gateway Protocol (BGP) is the core exterior gateway routing protocol of the Internet.

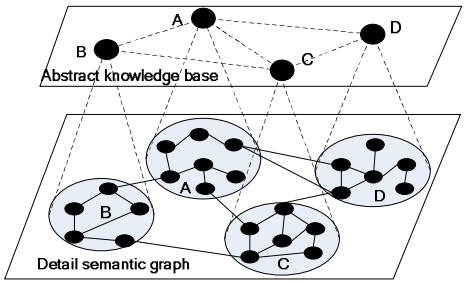

Fig. 2. Two layer structure

We adapt a similar routing scheme as BGP for semantic relationship discovery. As shown in Fig. 2, instead of starting from millions of semantic entities and relationships at the lower level, we consider each knowledgebase which contain multiple entities and relations as an abstract unit (as an AS in internet). We assume that each peer hosts an individual knowledgebase. We treat these knowledgebases as black boxes and ignore the detailed semantic entities and their relations. Next, through ontology-mappings, we connect knowledgebase to form a graph. The graph, called the *semantic graph*, will act as the blueprint of our Semantic Web. Based on the semantic graph, the semantic path discovery problem is analogous to the route discovery of Internet. This abstraction dramatically reduces the size of a potentially huge search space.

With the semantic graph constructed, the path finding problem is reduced to two steps: Firstly, locate the source and destination entities. Secondly, search at the peer level for paths from the source peer containing the source entity to the peer containing the goal entity. The result is a much faster search. To efficiently locate the source and destination semantic entities, we adopt a distributed hash table (DHT)-based overlay to index the semantic graph, with which semantic entities can be efficiently located. The focus of this paper is to discover semantic paths between the semantic entities. We propose a novel routing algorithm to efficiently locate semantic paths. In this section, we present each part of the relation discovery strategy in details.

B. Semantic Graph Formation

To link the dispersed knowledgebases to form a connected semantic graph, we propose a semantics-based topology adaptation scheme to connect knowledgebases containing similar semantic properties and facilitate establishing semantic mappings. The foundation of this scheme is a metric that measures peers' semantic similarity. We adopt the idea of distance-based approach [3], [4], [5] to measuring the semantic similarity between ontologies. The basic idea behind the distance-based approach is to identify the shortest path between two concepts in terms of the number of edges in the ontology graph and then translate that distance into semantic distance. Our approach improves the accuracy by integrating factors, such as the depth of a node in the ontology hierarchy

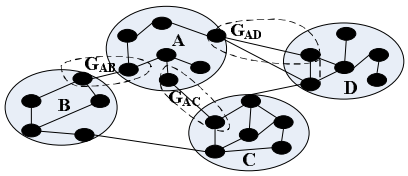

Fig. 3. Connection of knowledge bases and their gateways

and the type of links. With the semantic similarity defined, we can reconfigure the network topology accordingly. Due to the lack of space, we are restricted to provide only surface-level information pertaining to semantics-based topology adaptation. We encourage the readers to find more details about our proposed solution by referring to [19, 3].

C. Semantic Entity Location

Another task for semantic path finding is to efficiently locate specific semantic entities (e.g., the source and destination entity) in the semantic graph. To fulfill this task, we propose to construct index on entities. As mentioned, entities are subjects and objects in RDF triples. Triples in distributed knowledgebase that share common entities (i.e., the same subjects and/or objects) should be indexed together in one of the distributed peers, where they can be located later. The challenge in this scenario lies in assigning "index rendezvous points" for entities. To avoid the centralized bottleneck, we use a DHT overlay to provide decentralized and scalable rendezvous for RDF triple entities. Each triple is sent to two rendezvous peers for its subject and object respectively, which ensures that triples with common subjects and/or objects will be co-located. Unlike RDFPeer's data indexing [6], we do not index predicates (i.e., edges of the semantic graph), because normally we only need to locate entities of the semantic graph not the edges. We store each triple twice by applying a hash function to its subject and object. The DHT indexing guarantees the entities can be located within *log(N)* hops, where *N* is number of peers in the Semantic Web.

D. Semantic Relation Discovery

With the semantic graph created and both source and goal entities located, the next step is to locate paths between the source and goal entities. Next, we explain the details.

1) Path Cost of the Semantic Graph

To find a *k*-hop limited semantic path, we need to count the path length. Therefore, each peer records a set of distance (in terms of semantic edges in the semantic graph) before they can be treated as a black box. The distance that matters is shortest semantic hops between knowledgebases. As shown in Fig. 3, Knowledgebase *A* is linked to knowledgebases *B*, *C*, and *D* though ontology mappings. The entities in *A* that are mapped (most possibly with "equivalent_to" mapping) to other knowledgebases are called *gateway* nodes of *A*. For example, G_{AB}, G_{AC}, G_{AD} are knowledgebase *A*'s gateways to neighboring knowledgebases. If two knowledgebases *A* and *B* have more than one gateway nodes, we pick the one that contributes to the shortest semantic path. *A* records the shortest distances between all of its gateways. In Fig. 3, we can see that from *A*'s local knowledgebase *dist(* G_{AB}, G_{AC} *)=2* (i.e., the path cost of from *B* to *C* via *A* is *2*), *dist (*G_{AB}, G_{AD}*)=3, dist (*G_{AC}, G_{AD}*)=5*.

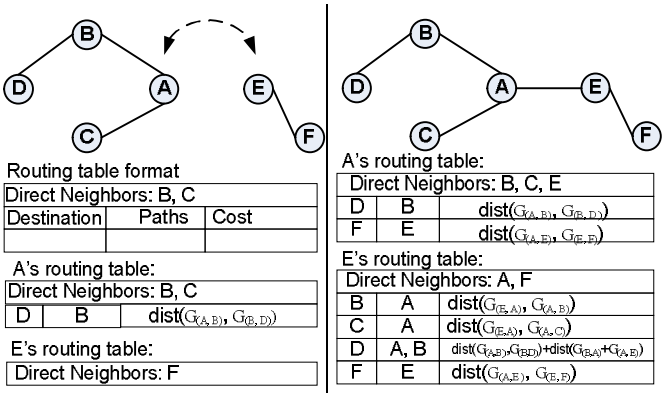

Fig. 4. Routing table updating after a new link has been established

2) Semantic Border Gateway Protocol (SBGP)

To locate paths between peers, we propose the Semantic Border Gateway Protocol (SBGP). This routing protocol was inspired by the BGP routing, but the protocol itself is different from BGP: BGP only needs to locate one shortest path from source to destination, while our SBGP has to locate multiple paths to discover more than one relationships between entities. The cost computation of SBGP is also different from BGP's. Unlike BGP, SBGP does not consider the cost of an edge but consider the cost of the paths between gateway nodes.

The SBGP is a path-vector protocol. In particular, for each peer *j*, peer *i* stores the peer paths of the lowest costs (maybe more than one) from *i* to *j*; in this vector, peers are identified by their peer ID. SBGP's route computation is similar to all path-vector routing protocols. Each peer sends its routing table to its neighbors, and each peer can then, based on this information, compute its own routing information.

Fig.4 illustrates the SBGP updating process when a new path between node *A* and *E* is found. The left column of Fig.4 shows the routing tables of *A* and *E* before the connection is established. The routing tables record the path (nodes traversed to reach a destination node) and cost information to other nodes. Note: the cost information is special. It is the distance between gateways. For example, in *A*'s routing table, the distance between *A* and *D* is the distance between two gateways: G_{BA} and G_{BD}, which is reported by *A*'s neighbor *B*. *A* does not record the cost to direct neighbors, since there is no gateway passed. *A* and *E* then exchange their routing tables. Based on the exchanged routing tables, they compute their updated routing tables. As shown in the right column of Fig.4, *E*'s new routing table now includes all nodes in *A*'s routing table and *A*'s direct neighbors. The cost is updated by adding the distance of gateway G_{AE} to *A*'s other boundaries in the path. For example, in *A*'s routing table, destination *D* can be reached with cost *dist(*G_{BA}, G_{BD}*)*. Then *E* can reach *D* through *A*'s boundaries G_{AE} and G_{AB}, therefore, in *E*'s routing table the cost is $dist(G_{AB},G_{BD})+dist(G_{AE},G_{A,B})$. If the cost is greater than the predefined *k* hop limit, the path is ignored. In this way, nodes can construct and update their routing tables. According to the routing table, a query looking for a destination node can be forwarded between peers in the routing path.

3) Semantic Relation Retrieval

Given a query of discovering the relationships between entity *A* and entity *B*, the system first locates the two peers, say, P_A, P_B, in charge of these two entities. There may exist multiple such peers. We consider each individual possible combination of them. With SBGP, the system can find the path connecting P_A and P_B. To retrieve semantic relations, the system has to go back to the detailed semantic graph (at the lower level shown in Fig. 2) in two steps: (1) Find the semantic path from entity *A* to one gateway node in P_A which is on the path to P_B. Similarly, find the semantic path from entity B to one gateway node in P_B which is on the path to P_A. (2) Retrieve the path from P_A to P_B, which in fact is a process of finding paths between all boundaries in the path.

IV. Related Work

Most current research on searching or querying Semantic Web uses an Information Retrieval (IR)-based search engine (e.g. [7–11]). The IR-based systems, such as Swoogle [8] and SWSE[9], indexes the Semantic Web by crawling and indexing the Semantic Web RDF documents found online and then offers a search interface over these documents. The IR-based search does not provide structured query capability.

Several groups [12-15] have developed technology to store RDF nodes, edges and labels into relational database systems, such as MySQL, Oracle, and DB2, so that Semantic Web data can be efficiently indexed and retrieved. They translate a SPARQL query into SQL statements which are evaluated on the triple store in relational databases.

To address the scalability issue, researchers have utilized P2P technologies to Semantic Web. For example, systems such as Edutella [16] and InfoQuilt [17] use broadcast or flooding to search RDF data, while many other projects, like RDFPeer [6] and OntoGrid [18] attempt applying DHT techniques to the retrieval of the ontology encoded knowledge. The queries that we try to address in this paper are fundamentally different from those described in any of the above approaches. Queries in the above mentioned approaches are mostly concerned with locating specific resources satisfying specific constraints, while we focus on locating relations between resources.

The query supporting semantic relationships was first proposed by Anyanwu and Sheth [1]. They define a semantic association as a complex relationship between two resources, and introduce a set of operators for querying semantic associations. Perry et al. [2] propose a method for computing semantic associations over P2P network. They use a super-peer based query planning algorithm for ρ-path quires. In their proposed systems, knowledgebases are stored at the peer level, while indexes are stored at the super peer level. Each super peer is in charge of a group of peers. A super peer knows about all other super-peers in the network and can query them to determine the semantic paths. This is an effective approach, but the scalability is still an unsolved issue. How to organize the peer group to reflect the semantic closeness and how super peers efficiently communicate is unaddressed.

V. Conclusion

This paper proposed a novel approach to discovering complex relationships from distributed knowledgebases. By correlating isolated islands of knowledge, individuals can gain new insights through the discovery of new relations. This technology is one of the key factors for realizing the full potential of utilizing the knowledge scattered over the Internet. Inspired by Internet routing, our proposed discovery system uses a two-level abstraction to reduce the search space and make discovery efficient and scalable.

This is an ongoing project. Currently, we are evaluating the performance of the system in terms of scalability and network latency. In the future we will work on alternative discovery algorithms and compare the performance between different approaches.

References

[1] Anyanwu, K., & Sheth, A. “r-Queries: Enabling Querying for Semantic Associations on the Semantic Web.” The Twelfth International World Wide Web Conference, Budapest, Hungary. 2003.

[2] Perry M., Janik M., Ramakrishnan C., Arpinar B. and Sheth A. “Peer-to-Peer Discovery of Semantic Associations.” In Workshop on P2P Knowledge Management, pages 1–12, 2005.

[3] J. Li and S. Vuong, “SOON: A Scalable Self-Organized Overlay Network for Distributed Information Retrieval”, in 19th IFIP/IEEE International Workshop on Distributed Systems (DSOM), 2008, pp.1-13.

[4] T. Pedersen, S. Patwardhan, J. Michelizzi, “WordNet: Similarity-Measuring the Relatedness of Concepts,” in 19th National Conference on Artifical Intelligence (AAAI), 2004.

[5] R. Rada, H. Mili, E. Bicknell, M. Blettner. “Development and Application of a Metric on Semantic Nets,” IEEE Transaction on Systems, Man, and Cybernetics, vol. 19, no. 1, pp. 17-30, 1989.

[6] M. Cai, M. Frank, “RDFPeers: A scalable distributed RDF repository based on a structured peer-to-peer network”, in proc of WWW conference, May 2004.

[7] Zhang, L., Liu, Q., Zhang, J., Wang, H., Pan, Y., Yu, Y.: Semplore: An IR approach to scalable hybrid query of semantic web data. In: Proceedings of the 6th International Semantic Web Conference, 2007.

[8] Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi,V., Sachs, J.: Swoogle: a search and metadata engine for the semantic web. In: Proc. of the 13th ACM CIKM Conf. (2004)

[9] Hogan, A., Harth, A., Umbrich, J., and Decker, S. “Towards a scalable search and query engine for the web”. In WWW 2007.

[10] Guha, R., McCool, R., Miller, E.: Semantic search. In: Proc. of the 12th Intl. Conf. on World Wide Web. (2003)

[11] Rocha, C., Schwabe, D., Aragao, M.P.: A hybrid approach for searching in the semantic web. WWW2004.

[12] Li Ma, Chen Wang, Jing Lu, Feng Cao, Yue Pan, Yong Yu, “Effective and Efficient Semantic Web Data Management over DB2.”, SIGMOD’08, June , 2008.

[13] Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: A generic architecture for storing and querying RDF and RDF Schema. In: Proc. of the ISWC2002

[14] Chong, E.I., Das, S., Eadon, G., Srinivasan, J.: An efficient SQL-based RDF querying scheme. In: Proc. of the VLDB 2005. (2005)

[15] S. Harris. SPARQL query processing with conventional relational database systems. In International Workshop on Scalable Semantic Web Knowledge Base System, 2005.

[16] 16 W. Nejdl et al. “EDUTELLA: a P2P Networking Infrastructure Based on RDF”. In Proc. of the WWW 2002

[17] M. Arumugam, A. Sheth, and I. B. Arpinar. “Towards peer-to-peer semantic web: A distribuited environment for sharing semantic knowledge on the web.” In Proc. of the International World Wide Web Conference 2002 (WWW2002), Honolulu, Hawaii, USA, 2002.

[18] OntoGrid project: http://www.ontogrid.net/

[19] J. Li and S. U. Khan, "MobiSN: Semantics-based Mobile Ad Hoc Social Network Framework," in IEEE Globecom 2009.