Automatic RDF Query Generation from Person Related Heterogeneous Data

Hiroyuki Sato, Iko Pramudiono, Kyoji Iiduka, and Takahiko Murayama
NTT Information Sharing Platform Laboratories, NTT Corporation
3-9-11 Midoricho, Musashino-shi, Tokyo 180-8585 Japan
{sato.hiroyuki, iko.pramudiono, iiduka.kyo, murayama.takahiko}@lab.ntt.co.jp

ABSTRACT

With the advance of the Semantic Web, the amount of data based on RDF is increasing rapidly on the Web. RDF data can easily merge one another because of its simple graph-based data model, but it is difficult to extract useful knowledge from data once merged because the query specifications require the knowledge of the whole graph structure.

Therefore we propose Context Structure Matching (CSM) based on a method for extracting complex but characteristic frequent occurrence pattern called *common query pattern* by analysis of graph structure. CSM enables users who input a simple keyword to extract not only related information as subgraphs from pattern matching on the merged RDF data, but also similar results and comparison points by reusing the *common query pattern*.

To evaluate the feasibility of the proposed method to merge real data of the Web, we perform an experiment with the data of W3C related people. Around 25,000 RDF triples are aggregated from multiple Web sites. As the results, after the merge process with minimal modification effort, CSM can extract more than 20 *common query pattern*s which are useful to query the relationship between W3C related people and their interests.

Keywords

Semantic Web, RDF Query, Graph pattern matching, FOAF

1. INTRODUCTION

With the advance of the Semantic Web, many people and organizations have been producing a large amount of data based on the Resource Description Framework (RDF). There are also some services such as rdfdata.org [1] and Swoogle [2] that maintain links to RDF data on the Web and also provide search capability to access the data.

The idea of the Semantic Web is to provide frameworks that allow the sharing and reuse of data across various applications. Providing models and syntaxes for knowledge representation makes information on the Web more usable by machines and raises the quality and possibilities of processing by automatic tools. The goal of the Semantic Web is to turn the Web into a huge database of well-defined data that is easily reused by different machines that do not require knowledge of each other's functions.

RDF query specifications are also defined to allow users to access the data using pattern matching [3]. However currently available RDF query specifications require the users to have the knowledge of the data structure.

Semantic Web provides methods to find new knowledge with inference using ontology. However there is only a little ontology available that can be applied to a merged heterogeneous data. Knowledge discovery based on inference only is suffering from the difficulty to design proper ontology.

With such a background, we have proposed a method called Context Structure Matching (CSM) to automatically create RDF queries based on simple keywords that users input [4]. The method focuses on the analysis of graph patterns from an integrated RDF graph to find frequent and complex relationships across heterogeneous RDF data in the form of *common query pattern*s. The generated query allows users to find information clusters composed of various data aggregated from different sources. The query also allows users to find similarity and comparison points in the RDF graph.

In this paper, we show how CSM can provide added value which can not be acquired by simple keyword search. CSM is implemented as a query engine that gives not only the targets deeply related with the keyword but also the reasons why they are chosen based on the *common query pattern*s. We report experimental results using RDF datasets about W3C related people collected from various websites in order to evaluate the feasibility of CSM for the integration of real datasets available on the Web.

2. EXTRACTING INFORMATION FROM MERGED GRAPH DATA

2.1 Merging RDF Data using its Graph-based data model

The RDF specifies an interoperable model for describing the semantic attributes of information resources that are identified by uniform resource identifiers (URIs). An RDF statement consists of three elements: a "resource", a "property", and a "value," as shown in Fig. 1. An RDF statement consisting of these three elements is also called a "triple". RDF has an abstract syntax that reflects a simple graph-based data model [5].

Multiple items of data published on the Web in RDF format can be merged by placing same data resources at a single node in the graph representation. Multiple graphs, each of which has hetero graph pattern and uses different vocabularies developed independently, can be merged as a single graph, as shown in Fig 2.

WWW2006, May 22–26, 2006, Edinburgh, UK.
.

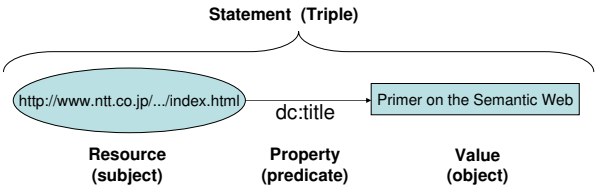

Figure 1: Graph representation of RDF data model.

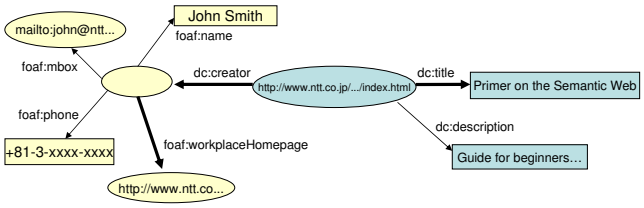

Figure 2: Merging of RDF data.

2.2 RDF Query

Various RDF query languages have been introduced to public. To provide applications with uniform access to RDF data, W3C has standardized SPARQL [6] since Feburary of 2004. Most RDF queries can be described by using a query graph pattern. The example below shows a SPARQL query [7] to find the workplace of the creator of "Primer on the Semantic Web" from the information in the RDF graph of Fig. 2.

```
PREFIX  foaf: <http://xmlns.com/foaf/0.1/>
PREFIX  dc:   <http://purl.org/dc/elements/1.1/>
SELECT ?workPlace
WHERE
{ ?document dc:title "Primer on the Semantic Web" .
  ?document dc:creator ?person .
  ?person   foaf:workplaceHomepage ?workPlace }
```

SPARQL query mainly consists of two parts, the SELECT clause and the WHERE clause. The SELECT clause identifies the variables of interest to the application, and the WHERE clause defines the triple patterns. This query contains a basic graph pattern of three triple patterns, each of which must be matched as the condition of the query. In this example, the query graph pattern matches the graph pattern that is indicated by a bold line in Fig. 2, so the result of the query will be the value that corresponds to the variable workplace of the person John Smith.

3. PROBLEM OF EXTRACTION

As mentioned above, by using query graph pattern, users can extract information from RDF graphs composed of a large amount of triples. However they cannot make it if they do not know about the graph structure in advance.

The Semantic Web allows anyone to use URIs to indicate various resources on the Web and lets users describe anything also using URIs and put it on the Web. The resources are then accessible from anywhere via the Internet. Anyone can use RDF Schema and OWL to define a vocabulary and ontology and make them available to everyone else via the Web.

As the consequence, there will be highly heterogeneous graph data patterns, so that end users can not understand all the graph data structures required to create a query. Note that the data structure may also change dynamically when new data is added.

4. CREATING QUERY

To solve the problem, we propose a method to automatically create query graph pattern based on a simple keyword that users input. The query allows users to obtain information consists of characteristic subgraphs related to the keyword. In addition, the method also provides similar or comparable information by reusing the generated query graph patterns.

The following subsection describes how our method automatically creates queries.

4.1 Extracting Characteristic Frequent Occurrence Pattern

Before making a query, this method analyzes graph of merged target data. The following shows the steps. Detailed process is also given in [4].

1. Search nodes whose value contains candidate keyword. The candidates are extracted from literal data of the RDF triples.

2. Search paths between the node and instance nodes of concepts.
   Target concepts, such as person and organization, might be predetermined by the service provider and are represented as a class of RDF Schema in RDF data. Determining the target concepts in advance will reduce the cost of path search. If the provider knows target class which is important for users, it is recommended to set target concepts before analysis.

3. Extract complex and frequent occurrence graph pattern.
   The pattern is created by combining multiple paths into one. Our proposed method merges combinations of paths which have the same two end nodes. For each combination, the method counts the number of nodes and arcs in the graph pattern to score the complexity. And to measure occurrence, the method counts the number of generated patterns which have same structure. The same structure means that all the corresponding arc labels of graph pattern, which are properties of RDF, are the same, but allows different values of corresponding node. We extract graph pattern which has high scores for both the complexity and the occurrence. This extracted pattern is called a *common query pattern*.

The *common query pattern* represents graph pattern which have commonly observed features. For example, a *common query pattern* may represent that it is quite common that "the person related to a certain keyword is the creator of more than two documents whose subject is same keyword." Once such *common query pattern* is indexed with the keyword, it is used for generating query when users input the keyword to search for the related people.

Note that a pattern which has only one path between the nodes representing a person and a keyword is removed,

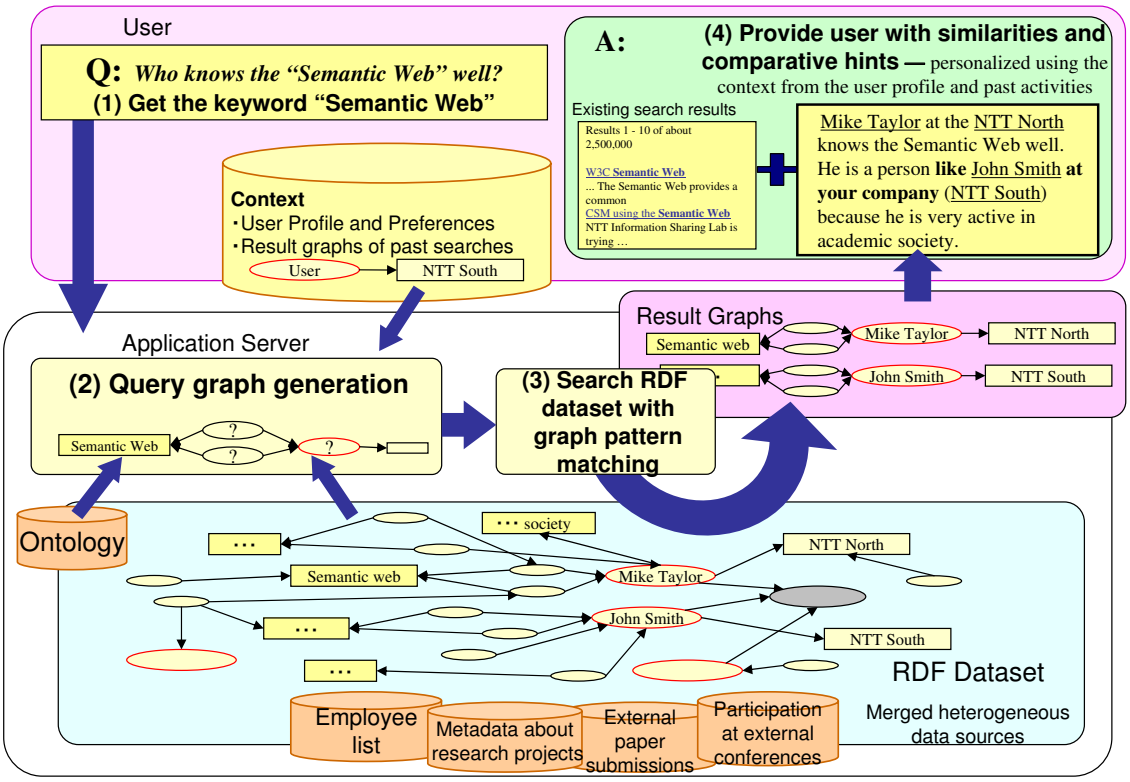

Figure 3: Applying Context Structure Matching (CSM) to the search system.

because the pattern is not so complex and when it is used for generating a query the result will be huge and may include too much noise. There exists other methods to create a *common query pattern*, but this subsection explains the one which is used in the experiments described below.

4.2 Context Structure Matching (CSM)

We implement the proposed method as a search engine called Context Structure Matching (CSM). CSM generates *common query pattern* to extract information with graph pattern matching.

Figure 3 shows how CSM works when it is applied to search data in an office. When a user input the word "Semantic Web", CSM searches the *common query pattern* which corresponds to the keyword. As (2) of Fig.3 shows, a query graph is generated by substituting the keyword for values of end node of the pattern. Moreover CSM reuses the pattern for enhancing the query. As the result of reusing, the engine extracts similar subgraph and comparison point shown in (4) of Fig. 3.

During the matching process, CSM uses ontology to find semantically similar graph patterns by changing target graph pattern flexibly. For example, when there is a definition about transitive property in the ontology data, the target for pattern matching is replaced by the result of inference.

We also introduce a way to use personal background information, which is called context, for the graph structure matching. In this research, we assume the following information as the context data.

- The history of individual user past searches in the form of result graphs.
- User profile information including ordinary personal data, such as address, workplace, and user preferences.

The context data is also stored as RDF data. The use of context data allows users to limit or expand the search according to their interest by enhancing query with context based patterns.

5. APPLYING CSM TO RDF DATA ABOUT PEOPLE

5.1 Experimental Conditions

Using the method mentioned in subsection 4.1, we conducted an experiment to generate queries and obtain subgraphs from aggregated RDF data on the Web. Target data for this experiment is data about W3C related people on the Web. We aggregated the following three kinds of datasets which are open to public on the Web.

1. Data about locations of people, research groups and projects related to W3C.
   This data is linked from the Web document about Semantic Web developer map [8] in SWAD-Europe Web site. The data contains 225 triples about 18 researchers.
2. Multiple Personal profile (FOAF) data which is aggregated from a place (Wiki) to put links to RDF files.

We use FOAFBulletinBoard [9] and AnRdfHarvesterStartingPoint [10] for harvesting. Most RDF files themselves are on the personal or their organization's Web sites. We use about 20 people's data and the total number of triples is about 1,500.

3. KnowWho metadata available from W3C KnowWho demo page [11].
   The data is automatically created from multiple W3C Web pages using natural language processing techniques. The RDF triples represent the relations between W3C related people and technological terms as well as the relations among those technological terms. It is composed of 22,685 triples.

Dataset 1 and 2 are mainly described by using vocabulary of the Friend-of-a-friend (FOAF) project [12]. The FOAF vocabulary is used to describe information such as personal profiles. The intention is to facilitate the formation of associations among people with similar interests, and backgrounds. The FOAF vocabulary covers items such as interest and nearby location as well as name and organization. A user is able to adopt unique IDs based on e-mail addresses and represent "people I know". The result of multiple FOAF definitions is a graph structure, which might be said to represent a social network.

We have to modify a part of dataset 3 in order to merge with the nodes of person in other datasets. In addition we also remove the blank nodes in dataset 3 which are originally used to represent the strength of the relationships. As the result, the resource nodes are directly connected to other nodes. After the merge of RDF triples, we extract *common query pattern* by using the method mentioned in subsection 4.1. The following are the parameters for the generation of *common query pattern*s.

- Total number of nodes in one path is less than 7.
- Contains at least two paths between end nodes.
- More than two same patterns are found in the merged graph.

5.2 Experimental Results

As the results of experiment, we obtained more than 20 *common query pattern*s. The followings are typical patterns extracted between the nodes which represent person and technical term. The three typical patterns are depicted in Fig. 4.

(a) A person who knows (has knowledge about) the keyword directly and also knows more than one term related with the keyword.

(b) A person who knows the keyword directly and more than one of his/her friend also know the same keyword.

(c) A person who knows the keyword directly and other person who belongs to the same organization/location and also knows the same keyword.

Those typical patterns are extracted from frequently occurred relations between people and technological term, beyond the direct path which exists between them. In addition, there are also a lot of relationships with similar form but different property. For example, there are some patterns

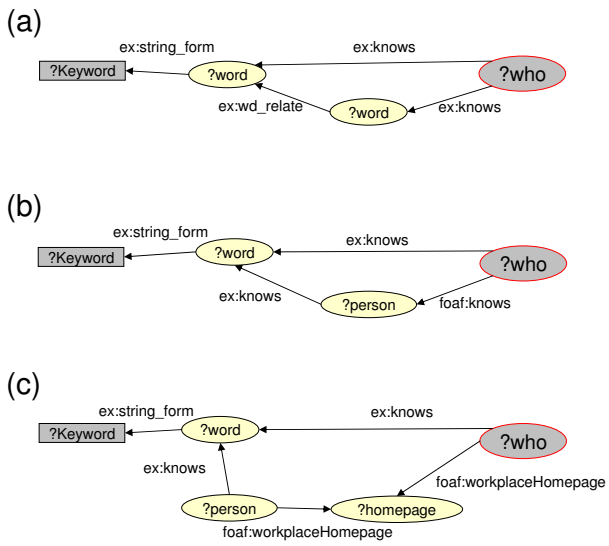

Figure 4: Typical common query patterns extracted from the experiment.

which replace the property foaf:workplaceHomepage in pattern (c) with foaf:groupHomepage or foaf:Interest. Combined patterns of those three types of patterns are also found. By using extracted patterns for creating query, the search system can provide persons who are deeply related with the keyword that users input. And the system also shows the reason why the person was chosen by showing the subgraphs of target graph, which are matched with query.

6. DISCUSSION ABOUT BENEFITS AND REMAINING ISSUES OF CSM

This research is based on an assumption that highly valuable search result has to fulfill the following conditions:

1. there are semantically complex relationships between the search keyword and the result nodes
2. the relationships are frequently occurred in the target graph

Those conditions are represented by the *common query pattern*. Here we evaluate the generation of *common query pattern*s for the application of search on heterogeneous data. We used three different RDF datasets available on the Web that also contain FOAF. We have confirmed that we can extract some *common query pattern*s. Using those *common query pattern*s, higher score can be given to search results that involve more patterns. When a user is looking for persons related to a keyword, persons who have various relationships with the keyword will appear as the top results.

We expect those results are more valuable and precise to the users. However the best query pattern for a certain user might be different to other user. Therefore, in the end, the users should be able to choose the most appropriate one for them. For this purpose, it is important to provide the users several results with corresponding reasons based on extracted typical patterns. CSM can provide reasons such as "person who also knows related word" or "person who

is belong to an organization whose people also knows the keyword." We are going to evaluate the precision of the search results by user's satisfaction. We are also examining some methods to filter unusable patterns.

There are some applications to visualize the relations between people by merging some FOAF files. The applications assume that the whole FOAF scheme is known. Our method does not require the target data is intended for merging. As long as the data can be represented in RDF, the method automatically generates the query and provides knowledge retrieval in the form of subgraphs of the merged data.

However as we have mentioned in the experiment conditions in subsection 5.1, there are some difficulties when merging the names of persons. There is a mechanism in FOAF to identify the same node on different RDFs such as email address or unique ID based on irreversible transformation of email address. However in reality, there are many RDFs that do not adopt the mechanism. Contradictions are also occurred when different versions of data related to the same person are merged from different websites. Butler has identified the problem as the record linkage problem of RDF data [13]. There is a research to utilize contexts of people to solve the problem [14]. We are going to examine a mechanism to decide whether to include a target graph from its creation time.

A research group from Southampton University has proposed CS Active Space to merge RDF data related to some researchers in order to retrieve knowledge [15]. Ontology is used heavily in CS Active Space. Our research differs since we focus on the analysis of graph patterns from target graph. However we are also planning to examine a better utilization of ontology in CSM.

7. CONCLUSION

Semantic Web technologies such as RDF promise a new paradigm on how to manage knowledge on the Web. Our contributions in this research are two-fold. The first is that we examine the usage of RDF as the framework for data integration, and we address several problems encountered for this purpose. The second is that we provide users a convenient way to discover knowledge using a novel graph analysis based semantic search engine called Context Structure Matching (CSM). The *common query pattern* which becomes the basis for pattern matching represents not only the frequent occurrence of the query target but also the complexity of the relationship between the query target and the search keyword. In addition, the *common query pattern* can be reused to find related similarities and comparison points.

Using the real RDF datasets available on the Web, we show that the *common query pattern*s can reveal relationships that can not be achieved by simple keyword matching. From around 25,000 RDF triples of W3C related people aggregated from several websites, we can extract more than 20 *common query pattern*s. The *common query pattern*s give not only the direct paths between a person and the technical term in his/her interests, but also the relations such as his/her social network with the same interest.

In the near future, we will expand the data domain for larger experiment and also improve algorithms for generating the *common query pattern*s.

8. REFERENCES

[1] rdfdata.org. http://www.rdfdata.org/

[2] Swoogle Semantic Web Search Engine. http://swoogle.umbc.edu/

[3] E. Prud'hommeaux and B. Grosof. RDF Query Survey. http://www.w3.org/2001/11/13-RDF-Query-Rules/

[4] H. Sato, K. Iiduka, T. Mukaigaito, T. Murayama. Finding similarity and comparability from merged hetero data of the Semantic Web by using graph pattern matching. In *Proc. of Activities on Semantic Web Technologies in Japan, WWW2005 Workshop.*

[5] G. Klyne and J. J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax. http://www.w3.org/TR/rdf-concepts/

[6] RDF Data Access Working Group. http://www.w3.org/2001/sw/DataAccess/

[7] E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF. http://www.w3.org/TR/rdf-sparql-query/

[8] L. Miller. Semantic Web Developer Map: representing locations of people, research groups and projects. http://www.w3.org/2001/sw/Europe/200303/geo/intro.html

[9] FOAFBulletinBoard - FOAF Wiki. http://rdfweb.org/topic/FOAFBulletinBoard

[10] AnRdfHarvesterStartingPoint - ESW Wiki. http://esw.w3.org/topic/AnRdfHarvesterStartingPoint

[11] H. Tsuda. W3C KnowWho. http://swada.w3.org/ htsuda/query.html

[12] the friend of a friend (foaf) project. http://www.foaf-project.org/

[13] M. H. Butler. Is the Semantic Web hype? http://www-uk.hpl.hp.com/people/marbut/

[14] J. Mori, Y. Matsuo, M. Ishizuka, B. Faltings. Keyword Extraction from the Web for FOAF Metadata. SWAD-Europe Final Workshop "Friend of a Friend, Social Networking and the Semantic Web". http://www.w3.org/2001/sw/Europe/events/foaf-galway/papers/fp/keyword_extraction_from_the_web/

[15] AKT - Technologies - CS AKTiveSpace from The University of Southampton. http://www.aktors.org/technologies/csaktivespace/