On Implementing a SPARQLoid Query Coding Support – Vocabulary Discovery for Queries with Weighted Ontology Mappings

Hiroki Noguchi, Takahisa Fujino, and Naoki Fukuta

Graduate School of Informatics
Shizuoka University
Hamamatsu, Japan
{gs13025@s,gs12033@s,fukuta@cs}.inf.shizuoka.ac.jp

Abstract. When writing SPARQL queries that will access to multiple data sources (i.e., a federated querying), one has to know the corresponding vocabulary where one can name the property URIs to be used in the query, and to know the exact URIs for such vocabulary is not easy task since those vocabularies are often separately defined and managed. To avoid such a situation, the techniques to use some ontology mappings in a SPARQL query have been actively developed. In this paper, we present our preliminary implementation of SPARQLoid query coding support system that can utilize familiar vocabularies that may be contained in some ontology mappings by using weighted ontology mappings associated to specified keywords in the query.

Keywords: SPARQL, ontology mapping, Linked Open Data

1 Introduction

The use of ontology mapping, alignment, and matching techniques [5, 6, 14] could be useful for writing queries when it allows us to use any ontologies in the queries unless those ontologies are not directly used in the endpoint [7]. There are some ontology alignment methods to produce precise and crisp mappings among their entities. However, in some cases, producing a precise mapping is very difficult[5, 14], especially when the two ontologies have semantic gaps. Alignment methods generally require much time to produce an alignment when the two ontologies are huge. Hence, an efficient way to get an alignment has also been proposed (e.g.,[18]). To utilize these ontology mappings that have a variety of confidence values, it is important to explicitly control their effective use in a query, since without such control a query can produce so much unwanted data in the process of its execution so that they cannot be processed. Although there are some approaches that can utilize mappings on querying with SPARQL[13, 16], they will not provide a functionality to explicitly control how to process queries with such 'somewhat unreliable' weighted ontology mappings. To solve this issue, SPARQLoid, a small extension to SPARQL that allows us to utilize such weighted ontology mappings effectively, has been proposed[8, 9].

However, on SPARQLoid, finding a base ontology and understanding their vocabulary defined in the ontology is still remained as one of the user's tasks, and it would be still difficult for various novice users. In this paper, we present our preliminary implementation of SPARQLoid query coding support system that can help users to utilize vocabularies that users are familiar with in the query, by providing a recommendation mechanism for sufficient ontologies and endpoints in combination with weighted ontology mapping-based querying functionalities implemented on SPARQLoid engine.

2 Background

2.1 Linked Open Data and SPARQL

The datatype of Linked Open Data (LOD) is basically written by RDF like Semantic Web. As an example, there is DBpedia that has been converted from the original Wikipedia content to the form of LOD. Furthermore, to realize "Open Government", various countries have been spent much efforts to realize their open data access site, such as "data.gov". In 2013, there are at least 447 registered public SPARQL endpoints with over 31 billion RDF triples, which are interlinked by around 504 million RDF links, and the amount of data are increasing[1].

There are many SPARQL endpoints (e.g., DBpedia[3]) that allow us to retrieve and see the data by queries that are written in SPARQL query language. We can retrieve data not only from one endpoint but also from several endpoints using federated queries [11, 12, 15]. However, it is not always easy to understand the ontology used at an endpoint well enough to write a query based on it. The main reason could be that, often an ontology is constructed with some specific terms that were not used in others. Furthermore, it often lacks detailed documentations.

2.2 Ontology Mapping

In open systems such as Semantic Web, how to deal with heterogeneity of data or ontology is becoming an important issue to be solved. To solve the issue, lots of efforts have been done about developing and evaluating approaches for ontology mapping[6]. Also, some practical software that implemented such ontology mapping techniques are widely used (e.g. LogMap[10], Alignment API[2], etc).

LogMap uses HermiT[3] as a reasoner to allow the method to generate higher quality mappings based on the reasoning results among the ontologies. Therefore, to fully utilize it, we need to prepare target ontologies that are ready for reasoning on HermiT. In this way, sometimes we need to utilize various approaches to generate mappings and their quality would be varied. For example,

[1] http://sparqles.okfn.org/availability
[2] http://alignapi.gforge.inria.fr/
[3] http://hermit-reasoner.com/

AlignmentAPI allows us to generate mappings based on string similarity or other mapping methods that users have chosen. It generates mappings in short time but it may produce mappings with lower precision.

In OAEI2012[1], 23 types of mapping generation methods were presented. We prepared a set of mapping generation and updating mechanism to obtain results by querying to heterogeneous LODs.

2.3 SPARQLoid

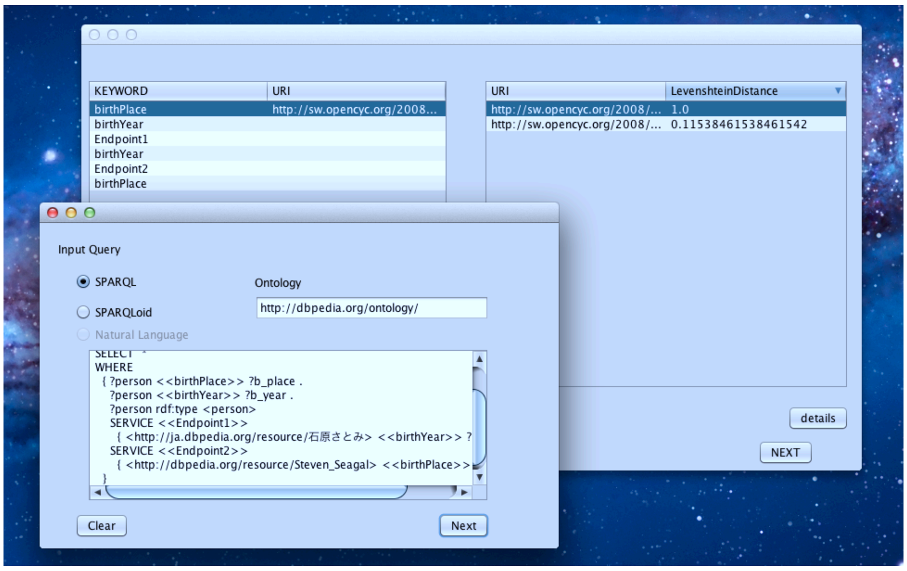
SPARQLoid enables users to control their utilization of different ontology mappings as well as confidence values in a query[7–9]. It allows users to choose and utilize ontology mappings to obtain the data in the target endpoint and specify the criterion for sorting the results and cutting output data that are far from accurate.

Here, we show a typical SPARQLoid query in listing 1.1. Notice that the SPARQLoid query in listing 1.1 obtains the matched instances as ?a, ?b, and ?c using the vocabularies in the conference namespace. However, each target endpoint to be accessed by a URI (*targetEndpoint* or *federatedQueryEndpoint* in listing 1.1) is expected to be constructed using each different ontologies.

```
SELECT ?a ?b ?c
WHERE
{
        ?a rdf:type conference:Regular_author
        ?a conference:has_the_last_name ?b {
                SERVICE < federatedQueryEndpoint > {
                        ?a conference:has_the_first_name ?c
                }
        } RANKING < targetEndpoint > {
                conference:Regular_author *0.5 +
                conference:has_the_last_name *0.2
        } THRESHOLD < targetEndpoint > {
                conference:Regular_author =0.3 ,
                conference:has_the_last_name =0.3
        } RANKING < federatedQueryEndpoint > {
                conference:has_the_first_name *0.3
        } THRESHOLD < federatedQueryEndpoint > {
        conference:has_the_first_name =0.2
        }
}
```

Listing 1.1. Example Query in SPARQLoid

For example, the class *conference:Regular_author* and the property *conference:has_the_first_name* require appropriate mappings to run the queries in listing 1.1 on the endpoints. In listing 1.1, *conference:Regular_author* was specified as more important than *conference:has_the_first_name* by the ranking clause. In case of a federated SPARQL querying, the weights of mappings are merged in the specified ways. In listing 1.1, some threshold values are specified to cut the



KEYWORD	URI
birthPlace	http://sw.opencyc.org/2008...
birthYear	
Endpoint1	
birthYear	
Endpoint2	
birthPlace	

URI	LevenshteinDistance
http://sw.opencyc.org/2008/...	1.0
http://sw.opencyc.org/2008/...	0.11538461538461542

Input Query

(● SPARQL) ○ SPARQLoid ○ Natural Language

Ontology: http://dbpedia.org/ontology/

```
SELECT *
WHERE
 { ?person <<birthPlace>> ?b_place .
   ?person <<birthYear>> ?b_year .
   ?person rdf:type <person>
   SERVICE <<Endpoint1>>
    { <http://ja.dbpedia.org/resource/石原さとみ> <<birthYear>> ?
   SERVICE <<Endpoint2>>
    { <http://dbpedia.org/resource/Steven_Seagal> <<birthPlace>>
 }
```

Fig. 1. Overview of System

irrelevant results that come from lower-weight mappings. For example, the mappings for *conference:Regular_author* whose confidence values are higher than 0.3 are used to execute the query on the target endpoint.

3 Proposed System

Our system, SuPARQooL(SUPport system for spARql Query cOding with Ontology mapping on sparqLoid) realizes a WOM(weighted ontology mapping)-based SPARQL query coding support with vocabulary discovery support technique. The users can write a query in SPARQL or SPARQLoid[7–9] that includes some natural language texts as the concepts (i.e., types) or properties to be used in them, and the system recommends sufficient URIs for them. Although other systems[4, 13, 16, 17] are not aware of weighted ontology mappings, SPARQLoid allows us to utilize weighted ontology mappings effectively. The system semi-automatically finds candidates of vocabularies or the entire ontologies that can be mapped to the users' familiar ontologies. Then, after clicking the "NEXT" button on the main window (see Figure1), our system calculates the matching degree to the endpoints that have been registered to the system. The endpoints in the list are scored and ranked based on the matching degrees. By pressing the button "more", the user can see the endpoints to be accessed and test-run some queries and investigate the endpoint itself to confirm it is sufficient to be used based on the investigated details of the endpoint. When the user has successfully coded the queries to be used, the system will prompt the user to select the endpoints to be accessed from the list. Then, based on the ontologies used in the endpoints, the SPARQLoid query is translated to a standard SPARQL

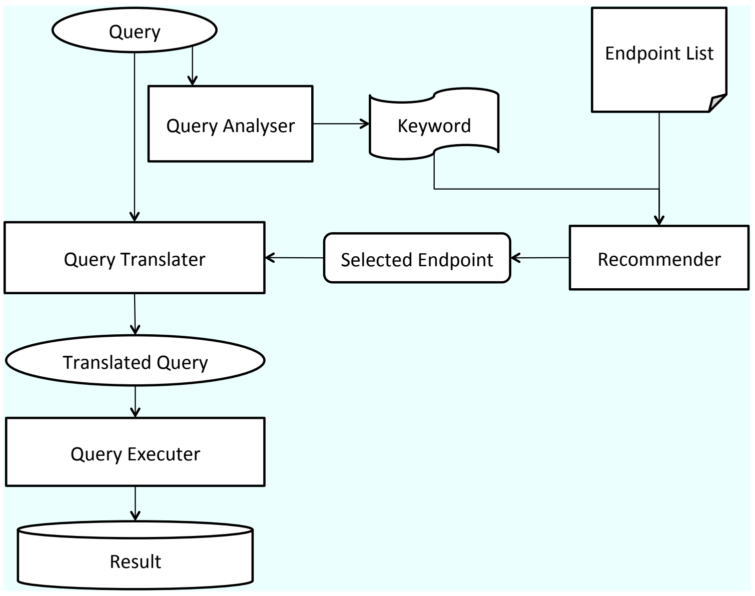


Fig. 2. Processing model of our system

query to be run on various SPARQL endpoints. The user can manually rewrite the generated SPARQL query for fine tunings of performance or adjust detailed behaviors of the query. The results of the query are also shown in the system, as well as converting the query that can embed some external applications to access external SPARQL endpoints.

Figure2 shows the structure of our preliminary implemented system. Query Analyser analyzes and extracts keywords. Recommender recommends endpoints by the specified keywords or related endpoint list. Query Translater translates SPARQLoid queries to standard SPARQL queries that can be run on various endpoints.

4 Conclusion

In this paper, we presented a SPARQL query coding support system with vocabulary discovery function for queries with weighted ontology mappings, which can be utilized in SPARQLoid queries. Our system allows users to query various open endpoints with weighted ontology mappings, as well as finding such endpoints and user's familiar base vocabularies to be used in the query.

References

1. Aguirre, J. L., Eckert, K., Euzenat, J., Ferrara, A., Van Hage, W. R., Hollink, L., Meilicke, C., Nikolov, A., Ritze, D., Scharffe, F., Shvaiko, P., Zamazal, O. Šváb, Trojahn, C., Jiménez-Ruiz, E., Grau, B. C. and Zapilko, B. Preliminary results of the

Ontology Alignment Evaluation Initiative 2012. In: Proc. of 7th Ontology Matching Workshop (OM2012), at International Semantic Web Conference (ISWC2012), 2012.
2. Atencia, M., Borgida, A., Euzenat, J., Ghidini, C., Serafini, L.: A formal semantics for weighted ontology mappings. In: Proc. of the 11th International semantic web conference (ISWC2012). (2012) 17-33
3. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: DBpedia: A Nucleus for a Web of Open Data. In: Proc. of the 6th International Semantic Web Conference (ISWC2007). (2007) 722-735
4. Bizer, C., Schultz, A.: The R2R Framework: Publishing and Discovering Mappings on the Web. In: Proc. of the 1st International Workshop on Consuming Linked Data (COLD 2010). (2010)
5. Duan, S., Fokoue, A., Srinivas, K., Byrne, B.: A clustering-based approach to ontology alignment. In: Proc. of the 10th International Semantic Web Conference (ISWC2011). (2011) 146-161
6. Euzenat, J., Shvaiko, P.: Classifications of ontology matching techniques. In: Ontology matching. (2007) 61-72
7. Fujino, T., Fukuta, N.: A SPARQL Query Rewriting Approach on Heterogeneous Ontologies with Mapping Reliability. In: Proc. of the IIAI International Conference on Advanced Applied Informatics (IIAI-AAI2012). (2012) 230-235
8. Fujino, T., Fukuta, N.: SPARQLoid - a querying system using own ontology and ontology mappings with reliability. In: Proc. of the International Semantic Web Conference (Posters & Demos) (ISWC2012). (2012)
9. Fujino, T., Fukuta, N.: Utilizing Weighted Ontology Mappings on Federated SPARQL Querying. In: Proc. of the 3rd Joint International Semantic Technology Conference (JIST2013). (2013) (to appear)
10. Jiménez-Ruiz, E., Grau, B, C.: LogMap: Logic-based and Scalable Ontology Matching. In: Proc. of the 11th International Semantic Web Conference (ISWC 2011), Part I, LNCS 7031, pp.273-288, 2011.
11. Ladwig, G., Tran, T.: Linked data query processing strategies. In: Proc. of the 9th International semantic web conference (ISWC2010). (2010) 453-469
12. Ladwig, G., Tran, T.: SIHJoin: querying remote and local linked data. In: Proc. of the 8th European Semantic Web Conference (ESWC2011). (2011) 139-153
13. Makris, K., Bikakis, N., Gioldasis, N., and Christodoulakis, S.: SPARQL-RW: Transparent Query Access over Mapped RDF Data Sources, In: Proc. of the 15th International Conference on Extending Database Technology (EDBT2012), (2012).
14. Noy, N. F.: In: Proc. of the Ontology mapping. In Steffen, S., Rudi, S., eds.: Handbook on Ontologies. (2009) 573-590
15. Quilitz, B., Leser, U.: Querying distributed RDF data sources with SPARQL. In: Proc. of the 5th European SemanticWeb Conference (ESWC2008). (2008) 524-538
16. Rivero, C., Hernandez,I., Ruiz,D., and Corchuelo, R.: Mosto: Generating SPARQL Executable Mappings Between Ontologies. In: Proc. of the 30th International Conference on Conceptual Modeling Demos and Posters. (2011).
17. Rivero, C., Hernandez,I., Ruiz,D., and Corchuelo, R.: Generating SPARQL Executable Mappings to Integrate Ontologies. In: Proc. of the 30th International Conference on Conceptual Modeling. (2011) 118-131
18. Seddiqui, M.H., Aono, M.: An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size. In: Web Semantics. 7(4). (December 2009) 344-356