Comparison Evaluation for SPARQL-to-SQL Translation Model

Hajung Sung¹, Suckhoon Lee¹, and Doo-Kwon Baik¹
¹Dept. of Computer Science and Radio Communications Engineering, Korea University, Seoul, Republic Of Korea
octoom@korea.ac.kr, leha82@korea.ac.kr, baikdk@korea.ac.kr

Abstract - Ontology use has increased with the increase in the use of the semantic Web. This has led to numerous research studies being conducted on ontology storage. SPARQL is a description language for the semantic Web environment and it searches ontology in an ontology database. However, it cannot search ontology in a relational database (RDB). To solve this problem, a data translation model and a query translation model have been proposed previously. However, these previous models have the disadvantage of requiring additional storage to store ontology. In this paper, we propose an SPARQL-to-SQL translation model that does not require additional ontology storage. The proposed model translates SPARQL into SQL to search data in an RDB. The data structure in such an RDB is a general data structure. As a result, the proposed model can search data stored in an RDB using SPARQL without requiring ontology storage. In addition, we describe models for comparison evaluation. The execution times of the models were evaluated in terms of five factors. The results of our evaluation showed that the proposed model is more efficient than other data translation models.

Keywords: Ontology, Ontology Storage, SPARQL, SQL

1 Introduction

The use of ontology has increased with the increase in the use of the semantic Web. Much research [1-5] has been conducted on ontology storage for efficient ontology management. SPARQL, a query language for ontology recommended by W3C, can be used to search ontology in an ontology database. However, it cannot be used to search data stored in a relational database (RDB). For the expansion of the semantic Web, it is necessary to consider data stored in an RDB. Therefore, to solve this problem, a data translation model has been proposed. This model converts data stored in an RDB to ontology and then stores the ontology in an ontology database. However, this model suffers from the following limitation. This model requires additional storage, which leads to increased costs for converting data stored in an RDB to ontology. Therefore, to overcome this limitation of translation costs, query translation models have been proposed, a representative model being the SPARQL-to-SQL translation model [6-10]. However, these models require additional ontology storage. Therefore, a query translation model that does not require additional ontology storage needs to be developed.

In this paper, we propose an SPARQL-to-SQL translation model that does not require additional ontology storage. We further present models for comparison with our proposed model, and compare their performances in terms of SPARQL-to-SQL translation time, loading time of ontology storage, RDB-to-RDF translation time, query execution time, total translation time, and total runtime. We give a clear account of the comparison results to prove the efficiency of the proposed SPARQL-to-SQL translation model.

The rest of this paper is organized as follows. Section 2 gives an account of related work. Section 3 describes our proposed SPARQL-to-SQL translation model that does not require additional ontology storage. Section 4 presents models for comparison with our proposed model and the results of a quantitative evaluation. Finally, Section 5 gives concluding remarks.

2 Related Work

Currently, extensive research has been conducted on ontology storage based on RDB with SPARQL. These studies can be divided mainly based on SPARQL-to-SQL translation models and data translation models. Representative studies of data translation models include the Virtuoso, D2RQ, and RDBToOnto [11,12,13]. A data translation model converts data stored in an RDB to ontology, and then stores the ontology in an ontology database.

Virtuoso is a server-based system that supports Windows and Linux operating systems. It is an ontology database that is also designed to enable the integration and access of data in the form of XML and RDF. Virtuoso is built on the basis of a physical storage. It can search data in an ontology database using SPARQL. Virtuoso can be either commercial or non-commercial and it uses Jena and Sesame APIs through Jena and Sesame providers, respectively. In this paper, we
conducted experiments on the noncommercial version of Virtuoso.

D2RQ converts data stored in an RDB into ontology using a mapping approach. The ontology is automatically generated by the system and stored in the memory. This generated ontology can be retrieved by using SPARQL.

RDBToOnto is a study in progress in the TAO project through which data stored in an RDB is converted into ontology. RDBToOnto describes the rich semantics of data stored in the RDB and supports automatic classification, which is performed using RTAXON learning methods. The relation of a word to its category is recognized as a subClassOf relationship according to the rules defined in RTAXON. In addition, the constraints defined by the user can be added, and a normalization function is provided to avoid duplication of data. Lastly, a convenient user interface is also provided. However, RDBToOnto only converts data stored in the RDB into ontology and then stores it as a file. Therefore, a model such as Fuseki can retrieve ontology using SPARQL. The above data translation model does not guarantee data consistency of ontology in the ontology database and the data stored in RDB. This problem is caused by the additional ontology storage. Changes in the data stored in the RDB and that stored in the ontology database cannot be changed simultaneously. In addition, data translation is time-consuming. Therefore, it is not suitable for processing large amounts of data.

3 Proposed SPARQL-to-SQL Model

Much research on SPARQL-to-SQL translation models has been conducted to overcome the disadvantages of data translation models. However, thus far, proposed SPARQL-to-SQL translation models support ontology storage based on RDB only. Therefore, this model needs additional ontology storage. In this paper, we propose an SPARQL-to-SQL translation model for general data in an RDB and compare the performance of the model with a data translation model.

Figure 1 illustrates the proposed model, which consists of the following four steps.

- Preprocessing: The SPARQL entered by the user is analyzed and converted into SQL for ontology storage based on RDB.

- RDB to Triple Mapping: The RDB components are mapped to RDF components in this step. The mapping information is stored in the form of OWL files should be prepared in advance by the user.

- SPARQL-to-SQL Translation: SQL statements obtained in the preprocessor step are converted into statements for querying data stored in an RDB referred by a mapping file. The RDB has a general schema.

- Search Results: Data stored in the RDB are retrieved and the query results are sent to the user.

4 Comparison Evaluation

In this section, we describe the preliminary constraints, factors, and notations relevant to the model for a comparison evaluation. In addition, we compare the performances of Virtuoso, Fuseki, D2RQ, and the proposed model.

4.1 Factors and Notation

The models to be compared are affected by the following five main factors:

- f1 : Number of RDB instances
- f2 : Loading time of ontology storage
- f3 : RDB-to-RDF translation time
- f4 : Query execution time
- f6 : SPARQL-to-SQL translation time
- f5 : Query pattern, which affect the translation time

Table 1 shows notations and their description.

<table>
<thead>
<tr>
<th>Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STT</td>
<td>SPARQL-to-SQL Translation Time</td>
</tr>
<tr>
<td>LT</td>
<td>Loading Time of ontology storage</td>
</tr>
<tr>
<td>RTT</td>
<td>RDB-to-RDF translation time</td>
</tr>
<tr>
<td>QET</td>
<td>Query execution time</td>
</tr>
<tr>
<td>TT</td>
<td>Total execution time</td>
</tr>
<tr>
<td>TRT</td>
<td>Total translation time</td>
</tr>
</tbody>
</table>
4.2 Models for comparison and evaluation

Figure 2 shows the models for comparison evaluation using SPARQL to query the data stored in the RDB. We determine the TRT of Virtuoso and Fuseki through the following two steps. In the first step, the RTT is obtained by RDToOnto. Then, the LT, which is the time to load ontology to an ontology database, of these models can be obtained. Finally, we can obtain the TT, which is the sum of RTT and LT. D2RQ converts data stored in the RDB automatically into ontology and then loads it to an ontology database. Therefore, the TT of D2RQ is composed of only LT. Similarly, the TT of the SPARQL-to-SQL translation model is composed of only STT. For each model, the TRT is the sum of TT and QET. Therefore, TRT is the main factor for comparison evaluation. The next section shows the results for the TRT. The query languages used in the experiment were presented by the LUBM Q1, Q2, Q3, Q11, and Q14 [14]. The rest of the query was excluded because it did not fit in the search data stored in RDB.

4.3 Assumptions

For comparison evaluation, we made the following assumptions and considered various definitions:

- All data are stored in an RDB.
- The TRT is composed of QET and TT.
- The TT is composed of RTT and LT in Virtuoso and Fuseki.
- The TT is composed of LT in D2RQ.
- The TT is composed of STT in the SPARQL-to-SQL translation model.
- Virtuoso and Fuseki convert data stored in the RDB into ontology using a file created using RDToOnto.
- Virtuoso and Fuseki use ontology created by RDToOnto.
- The time taken for creating mapping information is not considered in D2RQ and the proposed model.

4.4 Evaluation Results

In this section, we present results of comparison evaluation of models defined in Section 4.2. Figure 1 shows the QET. The execution time of a query is respectively similar. Therefore, QET negligibly affects the performance of the compared models.

![Figure 3 Query Execution Time in 714,370 triples](image)

Tables 2 and 3 show the performance evaluation results of the proposed SPARQL-to-SQL translation model, Virtuoso, Fuseki, and D2RQ. The values in the table show TRT. As shown in the tables, the proposed SPARQL-to-SQL translation model is more efficient than other models. Virtuoso, Fuseki, and D2RQ need data loading and translation time because these models need additional ontology storage. The time it takes to convert the data stored in the RDB using RDToOnto is 65 s for 714,370 triples and 14.27 s for 71,437 triples. In addition, these models need time to load ontology in an ontology database.

Table 4 shows the time to load ontology using Virtuoso, Fuseki, and D2RQ. Due to an increase in the number of triple loading, the time to load ontology is long. On the other hand, the proposed SPARQL-to-SQL translation model has an STT that is independent of the increase in the number of triples. Therefore, a model without additional ontology storage is more efficient than one that requires additional ontology storage.
Table 2 Total Runtime in 71,437 triples

<table>
<thead>
<tr>
<th>Query Pattern</th>
<th>Virtuoso</th>
<th>Fuseki</th>
<th>D2RQ</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>QP1</td>
<td>19.94</td>
<td>21.57</td>
<td>4.83</td>
<td>0.04</td>
</tr>
<tr>
<td>QP2</td>
<td>19.96</td>
<td>21.56</td>
<td>4.83</td>
<td>0.99</td>
</tr>
<tr>
<td>QP3</td>
<td>19.90</td>
<td>21.54</td>
<td>4.93</td>
<td>0.04</td>
</tr>
<tr>
<td>QP11</td>
<td>20.20</td>
<td>21.51</td>
<td>4.83</td>
<td>0.04</td>
</tr>
<tr>
<td>QP14</td>
<td>21.80</td>
<td>24.69</td>
<td>4.95</td>
<td>0.04</td>
</tr>
</tbody>
</table>

Table 3 Total Runtime in 714,370 triples

<table>
<thead>
<tr>
<th>Query Pattern</th>
<th>Virtuoso</th>
<th>Fuseki</th>
<th>D2RQ</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>QP1</td>
<td>114.02</td>
<td>429.08</td>
<td>8.26</td>
<td>0.06</td>
</tr>
<tr>
<td>QP2</td>
<td>113.71</td>
<td>428.49</td>
<td>7.77</td>
<td>1.06</td>
</tr>
<tr>
<td>QP3</td>
<td>113.54</td>
<td>428.66</td>
<td>8.06</td>
<td>0.06</td>
</tr>
<tr>
<td>QP11</td>
<td>114.24</td>
<td>428.38</td>
<td>8.47</td>
<td>0.06</td>
</tr>
<tr>
<td>QP14</td>
<td>115.02</td>
<td>428.57</td>
<td>8.18</td>
<td>0.06</td>
</tr>
</tbody>
</table>

Table 4 Loading Time of ontology storage

<table>
<thead>
<tr>
<th>The number of triple</th>
<th>Virtuoso</th>
<th>Fuseki</th>
<th>D2RQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>71,437</td>
<td>5.07</td>
<td>7.23</td>
<td>4.82</td>
</tr>
<tr>
<td>714,370</td>
<td>47.81</td>
<td>363.37</td>
<td>7.76</td>
</tr>
</tbody>
</table>

5 Conclusions and future works

This paper compared the performance efficiency of the proposed SPARQL-to-SQL translation model with other data translation models in terms of the time taken to execute an SPARQL query. The evaluation result showed that the proposed SPARQL-to-SQL model required less time than other data translation models, and hence, is more efficient.

In future, the proposed SPARQL-to-SQL translation model will be improved to be able to support all queries provided by the LUBM for improving its performance. Further, all RDF properties must be supported through a translation algorithm. Therefore, we need to develop a more powerful translation algorithm for RDB instances.

6 Acknowledgement

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (2012M3C4A7033346). The corresponding author is Doo-Kwon Baik.

7 References