Automated Mapping Generation for Converting Databases into Linked Data

Simeon Polfliet

Ensimag engineering school
Grenoble Institute of Technology (INPG)
Grenoble, France
simeon.polfliet@ensimag.imag.fr

Ryutaro Ichise

Principles of Informatics Research Division
National Institute of Informatics
Tokyo, Japan
ichise@nii.ac.jp

Abstract. Most of the data on the Web is stored in relational databases. In order to make the Semantic Web grow we need to provide easy-to-use tools to convert those databases into linked data, so that even people with little knowledge of the semantic web can use them. Some programs able to convert relational databases into RDF files have been developed, but the user still has to link manually the database attribute names to existing ontology properties and this generated "linked data" is not actually linked with external relevant data. We propose here a method to associate automatically attribute names to existing ontology entities in order to complete the automation of the conversion of databases. We also present a way - rather basic, but with low error rate - to add links automatically to relevant data from other data sets.

Keywords: Database, Linked Data, Semantic Integration, Semantic Web

1 Introduction

Even though significant research and development efforts have been made, the achievement of the vision of the Semantic Web remains remote. The amount of data on the Semantic Web remains marginal in comparison with the traditional Web. The importance of revealing relational data and making it available as RDF and as Linked Data[1] has been already acknowledged. Most notably, Virtuoso RDF views[4] and D2RQ[2] are production-ready tools for generating RDF representations from relational database contents. But the main restriction to their deployment is the complexity of generating a mapping, which is the last non-automated part of these programs.

In this paper, we will present a method to generate automatically the mapping between attribute names and existing ontology entities, completed with a method to add links automatically to external data. Then, we will present the application of this method on relational databases applied on the D2RQ Mapping system and the D2R Server[3], and the tests and results on different kind of relational databases.

A presentation of our software AuReLi (**Au**tomatic **Re**lational Database to **Li**nked Data Converter) can be found at `http://ri-www.nii.ac.jp/AuReLi/`

2 Method

In ontology matching, there are three types of methods to compare two entities: string-based, structure-based and knowledge-based methods. In the present problem, there

is on one side a database and on the other side we have several ontology descriptions. Thus, structure-based method are not relevant here. We are seeking to compare the name of an attribute in a database with the name of an ontology property. These names can be composed by one or several words: the first step is to split the name into a set of words in order to compare the words of each set. The success of the matching depends on the correctness of the word decomposition. The words composing names of ontology properties and of attributes in a database are usually either separated by special characters, for instance `product_name`, or by a change of case, e.g. `ProductName`. As sometimes it is not the case, we completed this simple splitting method with a method based on the presence of the words in a dictionary such as the WordNet dictionary[1] used here. After doing the previous splitting, it is necessary to check if the resulting words exist in the dictionary. If that is not the case, then we try to split it into words that are in the dictionary. However, because it is possible that the word is not in the dictionary but some part of it is, we will only keep the result if all the decomposed parts are present in the dictionary. With this method, even `productname` will be correctly split. The second step is to compare the resulting set of words of the attribute name with the sets of words of all the ontology entities, and then return the best match. In order to compare the words, we use string-based similarity measures[2], especially Jaro-Winkler, and WordNet similarity measures[3]: Lin[5] and Wu and Palmer[6] measures. We use WordNet measures if the words exist in the WordNet dictionary, otherwise we use the string-based ones.

Once the mapping is done, in order to have true linked data, we want to add links to relevant data. The idea is to make a SPARQL query on a given data set. If you know the target data and its ontology entities, you can specifically build SPARQL queries for this data set to get links. But here, in a more general setting, we do not have this information. However, there is a property common to most of the data sets: rdfs:label. Even better, this property is especially good because it is usually at the same time short and clearly defining the data. Therefore, if the rdfs:label property was correctly set on your data, the SPARQL query based on this property should not return wrong links and has good chances to find a result if there is a related data in the target data set.

3 Implementation

We produced a reusable Java library and used the D2RQ Map and the D2R Server[3] as a basis to implement and test our method. A Java graphical user interface was produced for the mapping generation, in order to simplify its use as much as possible. First, the user has to define the parameters to connect to the relational database, and to give to the program the ontology descriptions he wants to use, as shown in Fig. 1. We already provide some of the most common generic ontology descriptions along with some more specialized ones, but the user can add any other ontology by providing a file with its OWL definition. Then, the program generates the mapping of the table and attribute names with the ontology entities. It presents the resulting mapping to

[1] Princeton University: WordNet, Version 3.0:
`http://wordnet.princeton.edu/wordnet/download/`
[2] S. Chapman: SimMetrics Java library:
`http://www.dcs.shef.ac.uk/~sam/simmetrics.html`
[3] D. Hope: Java WordNet::Similarity:
`http://www.cogs.susx.ac.uk/users/drh21/`

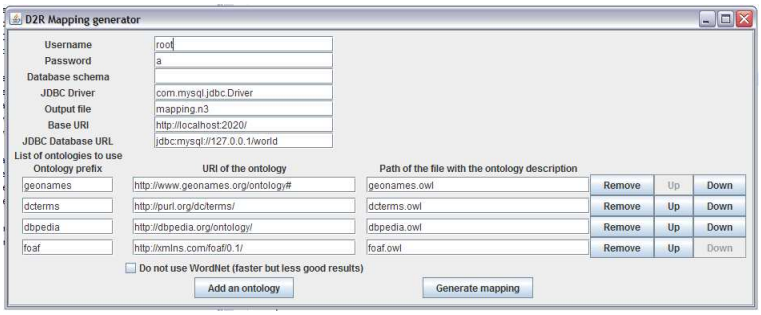

Fig. 1. Mapping generation graphical interface

the user so that he can check and make changes if necessary. It also allows the user to choose which attributes to use as labels for the rdfs:label property.

The D2R Server was also modified to add links automatically in the generated data. If the feature is activated, it makes a SPARQL query on DBpedia for each request of the user and add the link to the data if there was a result. We used DBpedia because it is currently one of the biggest and the most general linked database.

4 Test and Results

Five databases from different sources, with different size and about different topics were used for the tests: Northwind[4], World and Sakila[5], Automobile[6], World Development Indicator[7]

There are approximately three hundred attributes in those five databases: after a manual check of the mappings, 79,66% of the attributes were correctly mapped. The wrong mappings are explained by the fact that some attributes were too specific and consequently could not match any existing ontology property in the ontology descriptions used in the experiment. Another limit is the use of acronyms or short abbreviations, which did not produce a correct mapping either. The generation time was around one minute for each database. The mapping generated automatically can be seen in Fig. 2 for the World database. On the left and the middle are the table names and the attribute names, on the right are the matched ontology entities. We can observe for instance that the attribute *GNPOld* do not have good corresponding property and thus is mapped with *foaf:OnlineAccount* which is obviously irrelevant. But on the other hand, *Percentage* becomes *dbpedia:part*, which is quite good since a percentage is a part of something. This matching is due to WordNet because it would not have been found by a string-based similarity measure.

For the server, the use of the feature to automatically add links is slightly slowing down each request of the user because it needs the answer of the SPARQL query. It

[4] Example database for the Microsoft SQL Server:
`http://www.microsoft.com/downloads/details.aspx?FamilyID=06616212-0356-46a0-8da2-eebc53a68034`

[5] Two example database from the MySQL website:
`http://dev.mysql.com/doc/index-other.html`

[6] Data set from the UCI Machine Learning Repository:
`http://archive.ics.uci.edu/ml/datasets.html`

[7] database from the World Bank Data Catalog:
`http://data.worldbank.org/data-catalog`

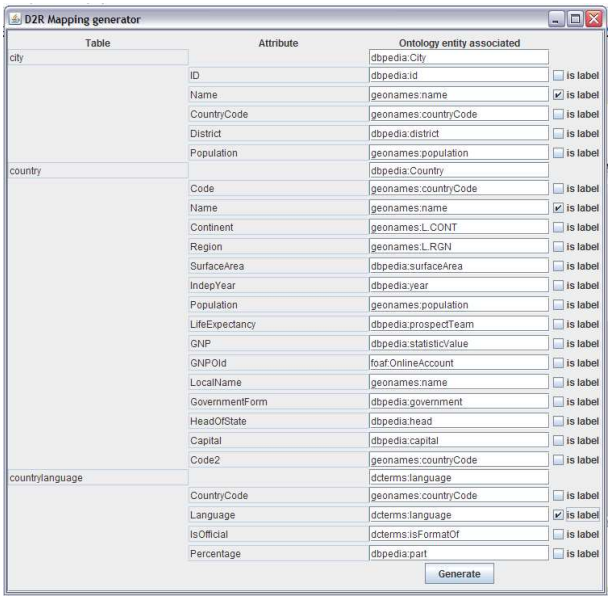

Fig. 2. Mapping generation result for the World database

becomes problematic if the external data set is slow or do not answer to the query. The results on the rdfs:label property on DBpedia are usually good, providing the labels in the mapping are correct. The principal case where the added links are wrong is in the case of homonyms, e.g. cities such as London, England and London, Canada.

5 Conclusion

The automatic mapping generation is a difficult problem which renders almost impossible the automatic production of a 100% correct mapping. Nevertheless, even if the user still needs some knowledge of the Semantic Web, we managed to simplify the process with a user-friendly interface where the user only has to check the correctness of the proposed mapping. The automatic addition of links in the generated RDF is simple and functional, and can easily be extended to add a greater variety of links.

References

1. T. Berners-Lee. Design issues: Linked data, 2006. `http://www.w3.org/DesignIssues/LinkedData.html`
2. C. Bizer and A. Seaborne. D2RQ - treating non-RDF databases as virtual RDF graphs. In ISWC2004 (posters), November 2004.
3. C. Bizer and R. Cyganiak: D2R Server, Version 0.7 `http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/`
4. O. Erling and I. Mikhailov. RDF support in the Virtuoso DBMS. In Proceedings of the 1st Conference on Social Semantic Web, volume P-113 of GI-Edition - Lecture Notes in Informatics (LNI), ISSN 1617-5468. Bonner Kollen Verlag, September 2007.
5. Lin, D. An information-theoretic definition of similarity. In Proceedings of the International Conference on Machine Learning, 1998
6. Z. Wu and M. Palmer. Verb semantics and lexical selection. In 32nd Annual Meeting of the Association for Computational Linguistics, 133-138, 1994