H-0286 (H1007-012) July 27, 2010
Computer Science

IBM Research Report

Enterprise Data Classification Using Semantic Web Technologies

David Ben-David
Technion - Israel Institue of Technology
Haifa 32000, Israel

Tamar Domany, Abigail Tarem
IBM Research Division
Haifa Research Laboratory
Mt. Carmel 31905
Haifa, Israel

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Enterprise Data Classification using Semantic Web Technologies

David Ben-David[1], Tamar Domany[2], and Abigail Tarem[2]

[1] Technion – Israel Institute of Technology, Haifa 32000, Israel,
davidbd@cs.technion.ac.il
[2] IBM Research – Haifa, University Campus, Haifa 31905, Israel,
{tamar,abigailt}@il.ibm.com

Abstract. In this work we present a reference implementation for Enterprise Data Classification using Semantic Web Technologies. We demonstrate automatic discovery and classification of Personally Identifiable Information (PII) in relational databases. The process starts with a classification model (in RDF/OWL) containing the elements to discover and classify, continues with the discovery process itself and finally documents the results in RDF, enabling simple navigation between the input model and the findings in different databases.

Keywords: Semantic Techniques, RDF, Classification, modeling, NeON, RelationalOWL

1 Introduction

In this work we demonstrate the concept of Enterprise Data Classification using Semantic Web Technologies described in [3]. This reference implementation demonstrates automatic discovery and classification of Personally Identifiable Information (PII) in relational databases. The goal of the solution is to provide organizations with a tool that automatically locates and annotates sensitive and valuable information, providing manageable results and enabling quick and easy access to the data in time of need (e.g to comply with regulations). The process starts with creating an RDF model containing the entities to discover and classify as well as additional information that can help the discovery process (e.g. type and format of the data), referred to as a *classification model*. In this demo we used a model representing PII, but any other model that follows the meta-model described in [3] can be used just as easily. The result of the classification process is a set of RDF triples linking between entities in the classification model and locations in the data stores, in this case database elements (tables or columns). Using RDF as the language in which to represent the classification model and results enables exploiting existing and evolving tools for annotating, reasoning, querying, etc. the classified data; common vocabularies (such as RDFS, OWL) can be used to express stronger relations and properties, best suited for ontologies; the powerful features of RDF make it easy to expand, merge and combine existing classification models and generate new models for different purposes.

The fact that the classification results are also represented in RDF has many additional advantages suach as unification of results from different classifiers, easy navigation between the model entities and the data sources (thanks of the use of URIs), inferencing, and many more.

The classification process is composed of four stages:

1. Creating or loading an existing classification model.
2. Importing database schemas.
3. Discovering and classifying the data according the the classification model using SPARQL and different classification algorithms.
4. Representing the results in a way that allows navigation between the classification model and the specific columns where the information was found.

A high-level view of this process is depicted in figure 1.

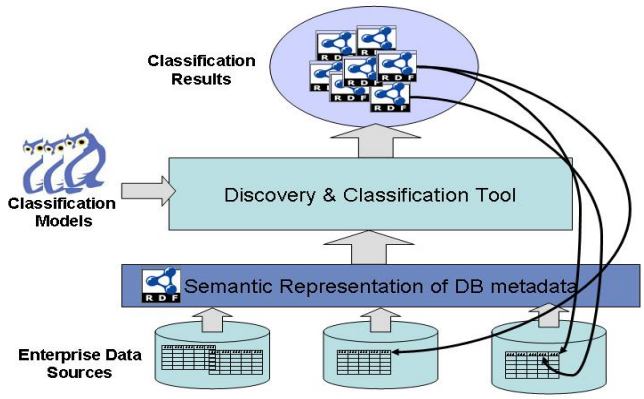

Fig. 1. The classification process

2 Implementation

Our reference implementation is based on the NeOn toolkit [5], an open-source, Eclipse-based ontology engineering environment. In addition we used the Eclipse Data Tools Platform (DTP)[3] for defining connections with local and remote databases. We chose the RelationalOWL ontology (with some slight changes) and implementation [6] for creating an RDF representation of the database metadata. We used the Jena framework [4] for accessing and querying the different RDF representations. To those we added a set of "home-made" plug-ins which perform the discovery and integrate between the different components in the system. The

[3] `http://www.eclipse.org/datatools`

discovery component uses the syllabifying techniques described in [3], as well as type checking. Future extensions are planned to include additional linguistic techniques (such as stemming) and the use of sample content to verify the data's format.

In the classification tool, a user can create projects, create and edit models, import existing models (in both cases, the models are validated against the meta-model) and import or create database metadata RDF representations. The discovery process can be performed on any combination of models and databases, and the results are also displayed in the project. The results (as well as the models and database metadata) can be viewed in both a hierarchical view and a graph view, as depicted in Figures 2 and 3. Figure 2 shows part of the initial PII model and Figure 3 shows part of the discovery results (in this case - all table columns in which a first name was discovered).

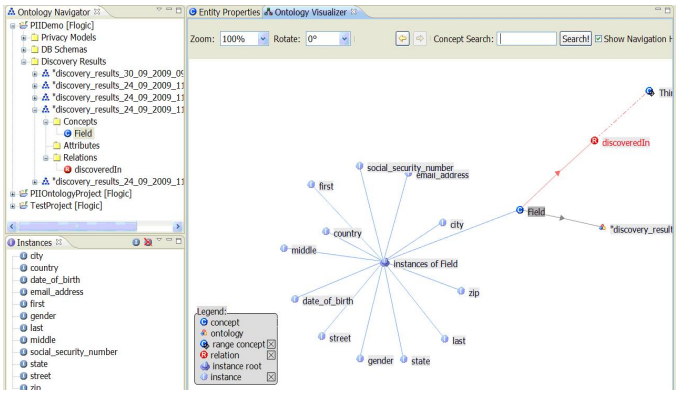

Fig. 2. A partial view of the PII model in the NeOn-based tool

For the purpose of this demonstration we are executing our classification on two externally available databases: one representing employee records in an organization (taken from the sample database created by the DB2® software installation) and the other representing medical records of patients (taken from "Avitek Medical Records Development Tutorials" by BEA Systems, Inc. [4]).

Thanks to the use of RDF to represent the discovery results (each triple in the result represents a data field that was discovered as belonging to one of the classification fields), it is possible to navigate from any node in the result back to both the classification field in the model, and to the data field (column) in the database representation. This easy navigation allows verifying the classification results, refining them (adding or removing triples), and enriching the model to be more accurate in subsequent runs.

[4] `http://download.oracle.com/docs/cd/E13222_01/wls/docs100/medrec_tutorials/index.html`

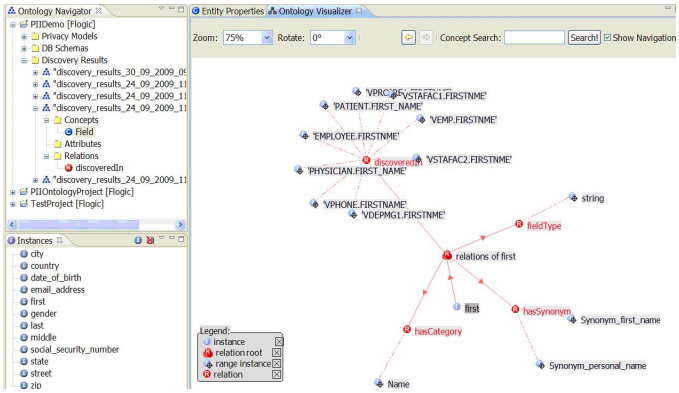

Fig. 3. A partial view of the discovery results in the NeOn-based tool

3 Summary

In this demonstration we exhibit the main advantages of our approach: the combination of different matching techniques and the extraction of most of the search logic to external files that can be easily changed, as opposed to changing the actual searching algorithms, creating a highly flexible and adaptable solution. Using RDF to represent both the ontologies and the results maximizes modularity and extensibility and facilitates easy navigation between the results, the models and the data sources. The ontology can serve as a centralized point to manage all valuable information in the organization and enables easy location of all related pieces of data in one click. In addition, all of the information created and used by the system (models, metadata RDF representations, results) can be exposed to existing and evolving Semantic Web tools.

References

1. Avitek Medical Records Development Tutorials, `http://download.oracle.com/docs/cd/E13222_01/wls/docs100/medrec_tutorials/index.html`
2. Eclipse Data Tools Platform (DTP) Project. data sheet, `http://www.eclipse.org/datatools/`
3. Ben-David, D., Domany, T., Tarem, A.: Enterprise Data Classification using Semantic Web Technologies. In: ISWC (2010)
4. Carroll, J.J., Dickinson, I., Dollin, C., Seaborne, D.R.A., Wilkinson, K.: Jena: Implementing the Semantic Web Recommendations. Tech. rep., HP Laboratories (2003), `http://www.hpl.hp.com/techreports/2003/HPL-2003-146.pdf`
5. Holger, P.H., Studer, L.R., Tran, T.: The NeOn Ontology Engineering Toolkit. In: ISWC (2009), `http://www.aifb.uni-karlsruhe.de/WBS/pha/publications/neon-toolkit.pdf`
6. de Laborda, C.P., Conrad, S.: RelationalOWL: a data and schema representation format based on OWL. In: Conferences in Research and Practice in Information Technology. pp. 89–96 (2005)