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. Organizations today collect and store large amounts of data in various formats and locations, however they are sometimes required to locate all instances of a certain type of data. Data classification enables efficient retrieval of information when needed. This work presents 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, using a classification model in RDF/OWL describing the elements to discover and classify. At the end of the process the results are also stored in RDF, enabling simple navigation between the input model and the findings in different databases.
Recorded demo link: https://www.research.ibm.com/haifa/info/demos/piidiscovery_full.htm

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

1 Introduction

Organizations today collect and store large amounts of data in various formats and locations. When an organization is required to meet certain legal or regulatory requirements, for instance to comply with regulations or perform discovery during civil litigation, it needs to find all the places where the required data is located. Data discovery and classification is about finding and marking enterprise data in a way that enables quick and efficient retrieval of the relevant information when needed. Most existing approaches either require re-classification of the data each time the organization's policies change, can only be applied to a single data type or format, or only identify predefined sets of known fields.

In this work we demonstrate the concept of enterprise data classification using Semantic Web technologies described in [2]. The goal of the solution is to provide organizations with a tool that automatically locates and annotates valuable information, provides manageable results and enables quick and easy access to the data when needed. For example, if in order to comply with a privacy regulation an organization is required to mask all Social Security Numbers (SSN), all the occurrences of SSN must be found.

This reference implementation demonstrates the automatic discovery and classification of Personally Identifiable Information (PII) stored in relational databases. The classification process starts with creating a model described using the Resource Description Framework (RDF), containing the entities to discover and classify as well as additional information that can help the discovery process (e.g., type and format); this is referred to as a *classification model*. In this demo we used a model representing PII, but any model that follows the meta-model described in [2] can be used. 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 tables and columns. Using RDF to define the classification model makes it easy to expand, merge and combine existing models and generate new models for different purposes. The fact that the classification results are also represented in RDF that follow the same shema as the classification model, enables us to unify the results from different classifiers, navigate easily between the model entities and the data sources (thanks of the use of URIs), annotate, reason, and query the classified data, and 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 to the classification model, using SPARQL and various 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.

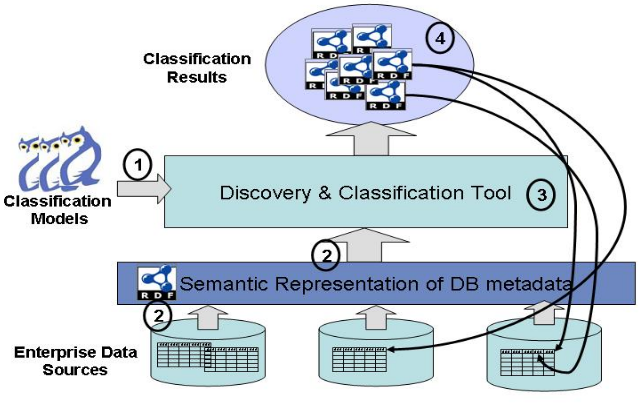

Fig. 1. The classification process

2 Implementation

Our reference implementation is based on the NeOn toolkit [4], an open-source, Eclipse-based ontology engineering environment. In addition we used the Eclipse Data Tools Platform (DTP)[1] to define connections with local and remote databases. We chose RelationalOWL [5] as the basis to create an RDF representation of the database metadata. We also used the Jena framework [3] to access and query the different RDF representations. To those we added a set of "home-made" plug-ins that perform the discovery and integrate between the different components in the system. The discovery component uses the syllabifying techniques described in [2], 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.

Using our classification tool users can create projects, build 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. Users can perform the discovery process on any combination of models and databases. 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 Figure 2. Figure 2 shows part of the discovery results (in this case - all table columns in which a first name was discovered).

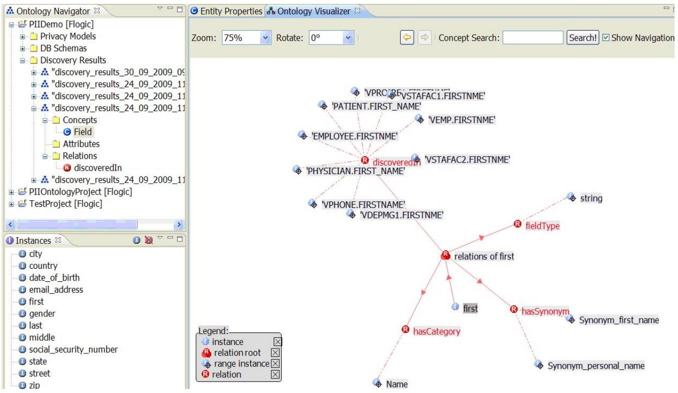

Fig. 2. A partial view of the discovery results in the NeOn-based tool

For the purpose of this demonstration, we execute 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. [3]).

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

As noted previously, we use RDF to represent the discovery results, making it 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 so it is more accurate in subsequent runs.

3 Summary

In this demonstration we exhibited the main advantages of our approach. By combining different discovery techniques and extracting most of the search logic to external files, we created a highly flexible and adaptable solution. Using RDF to represent both the ontologies and the results maximizes the modularity and extensibility of the classification input and facilitates easy navigation between the results, the models and the data sources. The ontology can thus 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, such as semantic query languages, reasoning engines and rule languages.

References

1. Eclipse Data Tools Platform (DTP) Project. data sheet, `http://www.eclipse.org/datatools/`
2. Ben-David, D., Domany, T., Tarem, A.: Enterprise Data Classification using Semantic Web Technologies. In: ISWC (2010)
3. 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`
4. 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`
5. 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)