Karma: A System for Mapping Structured Sources into the Semantic Web*

Shubham Gupta, Pedro Szekely, Craig A. Knoblock, Aman Goel,
Mohsen Taheriyan, and Maria Muslea

University of Southern California
Information Sciences Institute and Department of Computer Science
{shubhamg,pszekely,knoblock,amangoel,mohsen,mariam}@isi.edu

1 Introduction

The Linked Data cloud contains large amounts of RDF data generated from databases. Much of this RDF data, generated using tools such as D2R, is expressed in terms of vocabularies automatically derived from the schema of the original database. The generated RDF would be significantly more useful if it were expressed in terms of commonly used vocabularies. Using today's tools, it is labor-intensive to do this. For example, one can first use D2R to automatically generate RDF from a database and then use R2R to translate the automatically generated RDF into RDF expressed in a new vocabulary. The problem is that defining the R2R mappings is difficult and labor intensive because one needs to write the mapping rules in terms of SPARQL graph patterns.

In this work, we present a semi-automatic approach for building mappings that translate data in structured sources to RDF expressed in terms of a vocabulary of the user's choice. Our system, Karma, automatically derives these mappings, and provides an easy to use interface that enables users to control the automated process to guide the system to produce the desired mappings. In our evaluation, users need to interact with the system less than once per column (on average) in order to construct the desired mapping rules. The system then uses these mapping rules to generate semantically rich RDF for the data sources.

We demonstrate Karma using a bioinformatics example and contrast it with other approaches used in that community. Bio2RDF [7] and Semantic MediaWiki Linked Data Extension (SMW-LDE) [2] are examples of efforts that integrate bioinformatics datasets by mapping them to a common vocabulary. We applied our approach to a scenario used in the SMW-LDE that integrate ABA, Uniprot, KEGG Pathway, PharmGKB and Linking Open Drug Data datasets using a

* This research is based upon work supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Air Force Research Laboratory (AFRL) contract number FA8650-10-C-7058. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, AFRL, or the U.S. Government.

common vocabulary. In this demonstration, we first show how a user can interactively map these datasets to the SMW-LDE vocabulary, and then we use these mappings to generate RDF for these sources.

2 Application: Karma

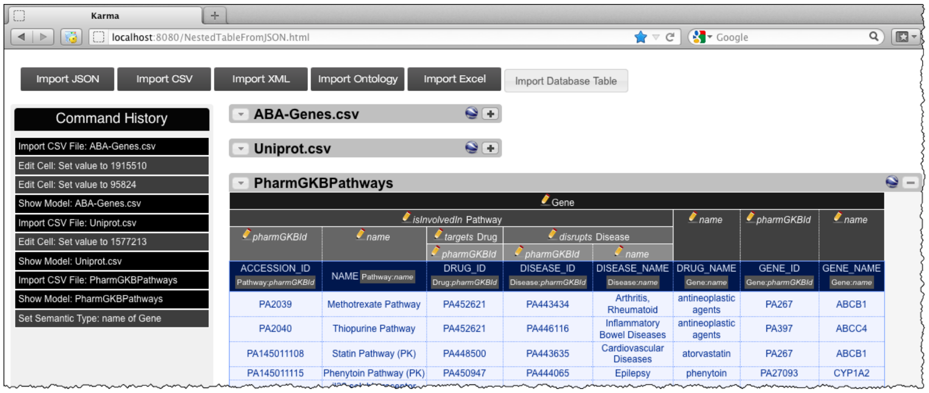
Karma[1] is a web application that enables users to perform data-integration tasks by example [8]. Karma provides support for extracting data from a variety of sources (relational databases, CSV files, JSON, and XML), for cleaning and normalizing data, for modeling it according to a vocabulary of the user's choice, for integrating multiple data sources, and for publishing in a variety of formats (CSV, KML, and RDF). In this demonstration we focus on the capabilities to interactively model sources according to a chosen vocabulary and to publish data in RDF.

The modeling process takes as input a vocabulary defined in an OWL ontology, one or more data sources to be modeled, and a database of semantic types learned in previous modeling sessions. It outputs a formal mapping between the source and the ontology that can be then used to generate RDF. The key technologies that this process exploits are the learning of semantic types using conditional random fields (CRF) [6] and a Steiner tree algorithm to compute the relationships among the schema elements of a source.

Semantic types characterize the meaning of data. For example, consider a dataset with a column containing PharmGKB accession identifiers for pathways. The syntactic type of the values is *String*. In our formulation, we represent their semantic type as a pair consisting of the class Pathway and the property pharmGKBId to capture the idea that these values are a particular type of pathway identifier. In RDF terms, the values are the objects of triples whose subject is of type Pathway and whose property is pharmGKBId. Karma infers semantic types automatically using the semantic types it has been trained to recognize. When Karma is unable to infer the semantic type for a column, users can interactively assign the desired type; Karma uses the assigned type and the data in the column to train a CRF model to recognize the type in the future [4]. The semantic types are used by our Steiner tree algorithm to compute the source model as the minimum tree that connects the assigned semantic types via properties in the ontology (the details of the approach are published elsewhere [5]). Because the minimum model is not always the desired model, Karma provides a user interface to enable users to force this algorithm to include specific properties in the model.

Most of the existing mapping generation tools, such as Clio [3], Altova MapForce (altova.com), or NEON's ODEMapster [1], rely on the user to manually specify the mappings in a graphical interface. In contrast, Karma provides a semi-automatic approach to achieve the same objective, enabling domain experts (and not just DB administrators or ontology engineers) to specify the mappings.

[1] https://github.com/InformationIntegrationGroup/Web-Karma-Public.



Command History

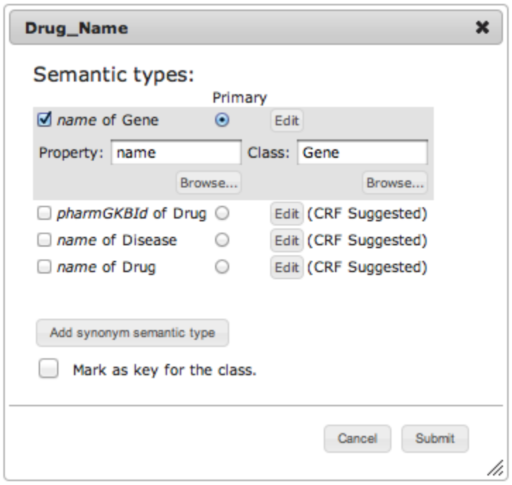
- Import CSV File: ABA-Genes.csv
- Edit Cell: Set value to 1915510
- Edit Cell: Set value to 95824
- Show Model: ABA-Genes.csv
- Import CSV File: Uniprot.csv
- Edit Cell: Set value to 1577213
- Show Model: Uniprot.csv
- Import CSV File: PharmGKBPathways
- Show Model: PharmGKBPathways
- Set Semantic Type: name of Gene

Gene							
isInvolvedIn Pathway					*name*	*pharmGKBId*	*name*
pharmGKBId	*name*	*targets* Drug	*disrupts* Disease				
		pharmGKBId	*pharmGKBId*	*name*			
ACCESSION_ID Pathway:*pharmGKBId*	NAME Pathway:*name*	DRUG_ID Drug:*pharmGKBId*	DISEASE_ID Disease:*pharmGKBId*	DISEASE_NAME Disease:*name*	DRUG_NAME Gene:*name*	GENE_ID Gene:*pharmGKBId*	GENE_NAME Gene:*name*
PA2039	Methotrexate Pathway	PA452621	PA443434	Arthritis, Rheumatoid	antineoplastic agents	PA267	ABCB1
PA2040	Thiopurine Pathway	PA452621	PA446116	Inflammatory Bowel Diseases	antineoplastic agents	PA397	ABCC4
PA145011108	Statin Pathway (PK)	PA448500	PA443635	Cardiovascular Diseases	atorvastatin	PA267	ABCB1
PA145011115	Phenytoin Pathway (PK)	PA450947	PA444065	Epilepsy	phenytoin	PA27093	CYP1A2

Fig. 1. Karma workspace showing a bioinformatics source and its model.

3 Demonstration

In this demonstration, we first show how users model structured sources according to an ontology they select; then we show how Karma can use the model to generate RDF represented using the classes and properties defined in the ontology. We will illustrate the process using a bioinformatics example.



☑ *name* of Gene ●
Property: name Class: Gene

☐ *pharmGKBId* of Drug ○ (CRF Suggested)

☐ *name* of Disease ○ (CRF Suggested)

☐ *name* of Drug ○ (CRF Suggested)

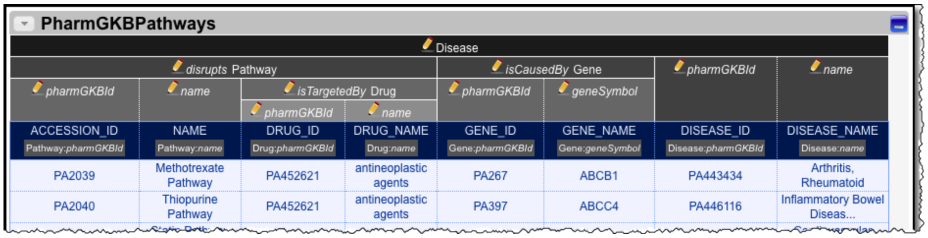
☐ Mark as key for the class.

Fig. 2. Semantic type selection dialog box.

In the first part of the demonstration we provide an overview of the Karma workspace (Figure 1) and show how to import data into Karma.

In the second part we show the model that Karma automatically infers for a source. Karma builds the initial model using the existing database of semantic types and visualizes it as hierarchical headings over the worksheet data. The inferred semantic types are shown in the grey boxes nested inside the dark blue boxes that show the column names.

In the third part we show how users can adjust the automatically generated model. We show how users can fix incorrectly assigned semantic types, and how users can adjust the model when Karma infers incorrect relationships between columns.



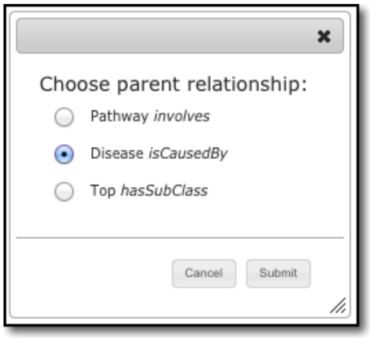
PharmGKBPathways

Disease							
disrupts Pathway				*isCausedBy* Gene		*pharmGKBId*	*name*
pharmGKBId	*name*	*isTargetedBy* Drug		*pharmGKBId*	*geneSymbol*		
		pharmGKBId	*name*				
ACCESSION_ID Pathway:*pharmGKBId*	NAME Pathway:*name*	DRUG_ID Drug:*pharmGKBId*	DRUG_NAME Drug:*name*	GENE_ID Gene:*pharmGKBId*	GENE_NAME Gene:*geneSymbol*	DISEASE_ID Disease:*pharmGKBId*	DISEASE_NAME Disease:*name*
PA2039	Methotrexate Pathway	PA452621	antineoplastic agents	PA267	ABCB1	PA443434	Arthritis, Rheumatoid
PA2040	Thiopurine Pathway	PA452621	antineoplastic agents	PA397	ABCC4	PA446116	Inflammatory Bowel Diseas...

Fig. 3. Source model for PharmGKB Pathways data before model refinement.

In our example shown in Figure 1, when the user loads the source, Karma incorrectly assigns the semantic type Gene.name to the DRUG_NAME column. To correct the problem, users click on the semantic type to bring up the semantic type specification dialog (Figure 2). The dialog shows the top options computed by the CRF model. When the correct option is in the list, users can select it with a single click. Otherwise, users specify the class and property by typing it (with completion) or by selecting the appropriate class or property from an ontology browser. In our example, the correct semantic type Drug.name is the fourth option. After each adjustment to the semantic types, Karma retrains the CRF model and invokes the Steiner tree algorithm to recalculate the set of properties that tie together the semantic types. Figure 3 shows the updated model incorporating the user changes.

The model proposed by Karma in Figure 3 is not correct because it specifies that the Gene columns contain information about genes that *cause* the disease described in the Disease columns (it models the relationship using the isCausedBy property). The correct model is that the genes are involved in the pathways that are disrupted by the disease. Users can specify the correct properties by clicking on the pencil icons.



Choose parent relationship:

○ Pathway *involves*

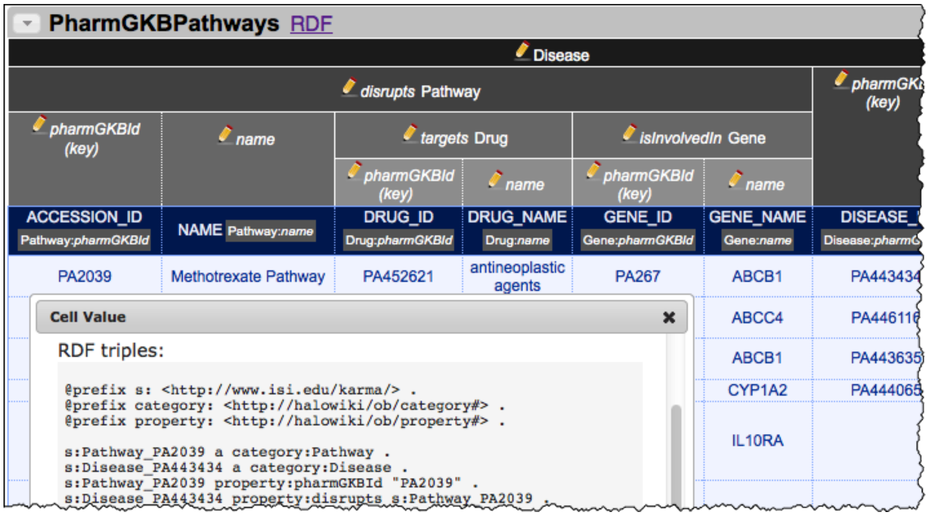
● Disease *isCausedBy*

○ Top *hasSubClass*

Fig. 4. Relationship selection dialog box.

Figure 4 shows the pop-up that appears by clicking on the pencil icon on the isCausedBy Gene cell. The pop-up shows *domain/property* pairs that satisfy two conditions. First, the class *domain* is a valid domain for the *property* and second, the class the user clicked (Gene in our example) is a valid range for the *property*. In our example, the correct choice is the first one because the information in the table is about Pathways that involve our Gene. After users make a selection, Karma recomputes the Steiner tree, which is now required to include the class/property selections users make [5]). Figure 5 shows the correct, updated model.

In the last part of the demonstration we show the RDF generation process. Once users are satisfied with the source model, they can generate and down-



PharmGKBPathways RDF

Disease						
disrupts Pathway						pharmGK (key)
pharmGKBId (key)	name	targets Drug		isInvolvedIn Gene		
		pharmGKBId (key)	name	pharmGKBId (key)	name	
ACCESSION_ID Pathway:pharmGKBId	NAME Pathway:name	DRUG_ID Drug:pharmGKBId	DRUG_NAME Drug:name	GENE_ID Gene:pharmGKBId	GENE_NAME Gene:name	DISEASE_ Disease:pharmG
PA2039	Methotrexate Pathway	PA452621	antineoplastic agents	PA267	ABCB1	PA443434
					ABCC4	PA44611
					ABCB1	PA443635
					CYP1A2	PA444065
					IL10RA	

Cell Value ✖

RDF triples:

```
@prefix s: <http://www.isi.edu/karma/> .
@prefix category: <http://halowiki/ob/category#> .
@prefix property: <http://halowiki/ob/property#> .

s:Pathway_PA2039 a category:Pathway .
s:Disease_PA443434 a category:Disease .
s:Pathway_PA2039 property:pharmGKBId "PA2039" .
s:Disease_PA443434 property:disrupts s:Pathway_PA2039 .
```

Fig. 5. RDF generation with Karma.

load the RDF for the whole source or view the RDF generated for a single cell (Figure 5). A movie of the whole user-interaction process is available online[2].

References

1. Barrasa-Rodriguez, J., Gómez-Pérez, A.: Upgrading relational legacy data to the semantic web. In: Proceedings of WWW Conference. pp. 1069–1070 (2006)
2. Becker, C., Bizer, C., Erdmann, M., Greaves, M.: Extending smw+ with a linked data integration framework. In: Proceedings of ISWC (2010)
3. Fagin, R., Haas, L.M., Hernndez, M.A., Miller, R.J., Popa, L., Velegrakis, Y.: Clio: Schema mapping creation and data exchange. In: Conceptual Modeling: Foundations and Applications - Essays in Honor of John Mylopoulos. pp. 198–236 (2009)
4. Goel, A., Knoblock, C.A., Lerman, K.: Using conditional random fields to exploit token structure and labels for accurate semantic annotation. In: Proceedings of AAAI-11 (2011)
5. Knoblock, C.A., Szekely, P., Ambite, J.L., Goel, A., Gupta, S., Lerman, K., Mallick, P., Muslea, M., Taheriyan, M.: Semi-automatically mapping structured sources into the semantic web. In: Proceedings of the Ninth Extended Semantic Web Conference. Crete, Greece (2012)
6. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning. pp. 282–289 (2001)
7. Peter, Ansell: Model and prototype for querying multiple linked scientific datasets. Future Generation Computer Systems 27(3), 329 – 333 (2011), `http://www.sciencedirect.com/science/article/pii/S0167739X10001706`
8. Tuchinda, R., Knoblock, C.A., Szekely, P.: Building mashups by demonstration. ACM Transactions on the Web (TWEB) 5(3) (2011)

[2] http://www.isi.edu/integration/videos/karma-source-modeling.mp4