Interactively Mapping Data Sources into the Semantic Web

Craig A. Knoblock, Pedro Szekely, Jose Luis Ambite, Shubham Gupta, Aman Goel, Maria Muslea, Kristina Lerman
University of Southern California

Parag Mallick
Stanford University
• Huge amount of data has been published to the Linked Open Data (> 28.5M triples)
• Remarkably little of this data has a detailed semantic description
• Challenge is how to allow users to easily publish data with respect to an ontology
• Can we automate the mapping to such an ontology?
Motivating Example

- Integrate data from the Allen Brain Atlas (ABA) with standard neuroscience data sources [Bizer & Cyganiak, 2006]
  - UniProt, KEGG Pathway, PharmGKB, Linking Open Drug Data
• Challenge:
  — Create formal mappings from each of the sources into a shared ontology
  — Use the mappings to create RDF
Motivating Example (cont.)

<table>
<thead>
<tr>
<th>Accession_Id</th>
<th>Name</th>
<th>Gene_Accession_Id</th>
<th>Gene_Name</th>
<th>Drug_Accession_Id</th>
<th>Drug_Name</th>
<th>Disease_Accession_Id</th>
<th>Disease_Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA2039</td>
<td>Methotrexate Pathway</td>
<td>PA267</td>
<td>ABCB1</td>
<td>PA452621</td>
<td>antineoplastic</td>
<td>PA443434</td>
<td>Arthritis, Rheumatoid</td>
</tr>
<tr>
<td>PA2040</td>
<td>Thiopurine Pathway</td>
<td>PA397</td>
<td>ABCB1</td>
<td>PA452621</td>
<td>antineoplastic</td>
<td>PA446116</td>
<td>Inflammatory Bowel Disease</td>
</tr>
<tr>
<td>PA145011108</td>
<td>Statin Pathway (P)</td>
<td>PA267</td>
<td>ABCB1</td>
<td>PA448500</td>
<td>atorvastatin</td>
<td>PA443635</td>
<td>Cardiovascular Disease</td>
</tr>
<tr>
<td>PA145011115</td>
<td>Phenytoin Pathway</td>
<td>PA27093</td>
<td>CYP1A2</td>
<td>PA450947</td>
<td>phenytoin</td>
<td>PA444065</td>
<td>Epilepsy</td>
</tr>
<tr>
<td>PA164713560</td>
<td>iL22 soluble receptor</td>
<td>PA29779</td>
<td>IL10RA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PA164713561</td>
<td>alpha-synuclein</td>
<td>PA32942</td>
<td>PARK2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PA164713575</td>
<td>endocytotic role</td>
<td>PA24852</td>
<td>AP2A1</td>
<td>PA164743471</td>
<td>adenosine triphosphate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PA154423650</td>
<td>Biphosphonate</td>
<td>PA26366</td>
<td>CDC23</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. :Pathway/Accession_Id/PA2039 a :Pathway;
2. :Accession_Id "PA2039";
3. :Label "Methotrexate Pathway";
4. :Involves :Gene/Accession_Id/PA267;
5. :IsTargetedBy :Drug/Accession_Id/PA452621;
6. :IsDisruptedBy :Disease/Accession_Id/PA443434.
7. :Gene/Accession_Id/PA267 a :Gene;
8. :Accession_Id "PA267";
9. :Label "ABCB1".
10. :Drug/Accession_Id/PA452621 a :Drug;
11. :Accession_Id "PA452621";
12. :Label "antineoplastic agents".
13. :Disease/Accession_Id/PA443434 a :Disease;
14. :Accession_Id "PA443434";
15. :Label "Arthritis, Rheumatoid".
Overall Approach

Karma

Assign Semantic Types
Specify Relationships
Generate Source Descriptions
Generate RDF

Ontology
Source
Learned Semantic Types

RDF
Building the Ontology Graph

Drug

Pathway

Gene

Disease

isTargetedBy

targets

disrupts

isInvolvedIn

involves

causes

treats

isTreatedBy

isDisruptedBy

isDisruptedBy

isCausedBy

keggDrugId

label

description

alternativeLabel

keggDiseaseId

keggPathwayId

abaGeneId

treats

treats

alts

targets

isDisruptedBy

keggGeneld

uniprotId

geneSymbol

entrezGeneId

isTreatedBy

isTreatedBy

isCausedBy

isCausedBy

alternativeLabel

alternativeLabel

alternativeLabel
Building the Ontology Graph
Building the Ontology Graph

Drug
- isTargetedBy
- isTreatedBy
- targets
- disrupts
- treats
- causes
- involves

Pathway
- disrupts
- involves

Gene
- isDisruptedBy
- isInvolvedIn
- isTargetedBy
- isTreatedBy

Disease
- causes

keggDrugId
- label
- description
- alternativeLabel

keggPathwayId
- label
- alternativeLabel

abaGeneId
- alternativeLabel

entrezGeneId
- uniprotId

geneSymbol
- geneName

keggDiseaseId
- label
- alternativeLabel

DISEASE_ACCESSION_ID
- DISEASE_NAME

GENE_ACCESSION_ID
- GENE_NAME

DRUG_ACCESSION_ID
- DRUG_NAME

ACCESSION_ID
- NAME
Building the Ontology Graph
**Problem:** Given some columns of data, identify their semantic class.

Semantic classes: • DrugName
  • DiseaseID
  • DiseaseName
  • GeneName

**Solution:** Train a CRF model that learns the association between the features of the tokens and their labels.

• Tokenize each field and extract their features.
• Create feature functions and learn their weights.
  • DrugNameToken is alphabetic
  • DrugNameToken is lowercase
  • DrugNameToken is the word “agents”
  • Field with label DrugName will have a token of label DrugNameToken
• Predict label for new column based on how many high-weight feature functions apply.
Erroneous labeling due to similarity with GeneName and lack of semantic type PathwayID in the system.

Assigning correct label to a column of type PathwayID.

The CRF model discriminates between PathwayID and GeneName.
Inferring the Relationships

- **Apply a fast Steiner tree algorithm**
  - $G=(V,E)$, $S \subset V$, $c: E \rightarrow \mathbb{R}$
  - Find a tree of $G$ that spans $S$ with minimal total cost
- **Approximation Alg. [Kou & Markowsky, 1981]**
  - Worst case time complexity: $O(|V|^2|S|)$
  - Approximation Ratio: less than 2
- **Example**

<table>
<thead>
<tr>
<th>Drug_Name</th>
<th>Gene_Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antineoplastic</td>
<td>ABCB1</td>
</tr>
<tr>
<td>Antineoplastic</td>
<td>ABCC4</td>
</tr>
<tr>
<td>Atorvastatin</td>
<td>ABCB1</td>
</tr>
</tbody>
</table>

- **Diagram**

---

**Figure 1**: Example drug-gene pathway network with Steiner nodes.
Steiner Tree algorithm (cont.)

- **Step 1**: construct the complete graph
  - Nodes: Steiner Nodes
  - Links Weights: shortest path from each pair in original G
- **Step 2**: compute MST (minimal spanning tree)
- **Step 3**: replace each link with the corresponding shortest path in original G
- **Step 4**: compute MST again
- **Step 5**: remove extra links until all leaves are Steiner nodes

Gene_Name | Drug_Name | Disease | Gene_Name
--- | --- | --- | ---
1 | 1 | 1 | 1
No Change

Drug_Name | Gene_Name | Disease | Gene_Name
--- | --- | --- | ---
4 | 4 | 1 | 1
Drug | Gene | treats | causes
No Change because all leaves (degree = 1) are Steiner nodes
Steiner Tree Algorithm

1. Construct the complete graph (Nodes: Steiner Nodes, Links Weights: shortest path from each pair in original G)

2. Compute MST

3. Replace each link with the corresponding shortest path in original G

4. Compute MST

5. Remove extra links until all leaves are Steiner nodes
Interactive Refinement of the Relationships

Pathway has label Drug_Name X
Interactive Refinement of the Relationships

<table>
<thead>
<tr>
<th>Pathway</th>
<th>has label</th>
<th>Drug_Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methotrexate Pathway</td>
<td>PA267</td>
<td>PA452621</td>
</tr>
<tr>
<td>Methotrexate Pathway</td>
<td>PA397</td>
<td>PA452621</td>
</tr>
<tr>
<td>Methotrexate Pathway</td>
<td>PA267</td>
<td>PA448500</td>
</tr>
<tr>
<td>Methotrexate Pathway</td>
<td>PA27093</td>
<td>phenytoin</td>
</tr>
<tr>
<td>Methotrexate Pathway</td>
<td>PA29779</td>
<td>life22 soluble receptor</td>
</tr>
<tr>
<td>Methotrexate Pathway</td>
<td>PA32942</td>
<td>alpha-synuclein and p...</td>
</tr>
<tr>
<td>Methotrexate Pathway</td>
<td>PA24852</td>
<td>endocytotic role of nd...</td>
</tr>
<tr>
<td>Methotrexate Pathway</td>
<td>PA26266</td>
<td>Bisphosphonate Pathway</td>
</tr>
<tr>
<td>Methotrexate Pathway</td>
<td>PA267</td>
<td>Etoposide Pathway</td>
</tr>
</tbody>
</table>
Interactive Refinement of the Relationships
Interactive Refinement of the Relationships

Pathway is Targeted by a Drug which has label Drug_Name
Generation of the Source Descriptions: Idea

- From
  - sources combined by the user in the interface, and
  - selected steiner tree over the ontology
- Construct
  - GLAV rule (st-tgd): logical implication with conjunctive formulas in antecedent and consequent
  - Use function symbols to generate URIs (object IDs)
  - Typical of data integration (e.g., [Halevy 2001]) and data exchange (e.g., [Arenas et al, 2010])

- To generate RDF use the GLAV rule in data exchange mode
Generation of the Source Descriptions: rule antecedent

- From
  - sources combined by the user in the interface
    \[ \text{antecedent} \] of GLAV rule
  - selected steiner tree over the ontology
- Construct
  - logical GLAV rule (st-tgd)

\[
\text{PharmGKBPathways(}\text{Name, Accession Id, Gene Accession Id, Disease Name, Gene Name, Disease Accession Id, Drug Name, Drug Accession Id)}
\]

(One source predicate in this example, but in general it could be a conjunction (join) of several source predicates)
• From
  — sources combined by the user in the interface
    \(\rightarrow\) antecedent of GLAV rule
  — selected steiner tree over the ontology
    \(\rightarrow\) consequent of GLAV rule
• Construct
  — logical GLAV rule (st-tgd)

\[
\text{Pathway}(\text{uri}(	ext{Accession Id})) \land \text{label}(\text{uri}(	ext{Accession Id}), \text{Name}) \land \text{involves}(\text{uri}(	ext{Accession Id}), \text{uri}(	ext{Gene_Accession Id})) \land \text{isTargetedBy} (\text{uri}(	ext{Accession Id}), \text{uri}(	ext{Drug_Accession Id})) \land \text{isDisruptedBy} (\text{uri}(	ext{Accession Id}), \text{uri}(	ext{Disease_Accession Id})) \land \\
\text{Gene}(\text{uri}(	ext{Gene_Accession Id})) \land \text{label}(\text{uri}(	ext{Gene_Accession Id}), \text{Gene_Name}) \land \\
\text{Drug}(\text{uri}(	ext{Drug_Accession Id})) \land \text{label}(\text{uri}(	ext{Drug_Accession Id}), \text{Drug_Name}) \land \\
\text{Disease}(\text{uri}(	ext{Disease_Accession Id})) \land \\
\text{label}(\text{uri}(	ext{Disease_Accession Id}), \text{Disease_Name})
\]
Generation of the Source Descriptions

- From
  - sources combined by the user in the interface, and
  - selected steiner tree over the ontology
- Construct
  - logical GLAV rule (st-tgd)

\[
\text{PharmGKBPathways} \left( \text{Name}, \text{Accession ID}, \text{Gene Accession ID}, \text{Disease Name}, \right. \\
\text{Gene Name}, \text{Disease Accession ID}, \text{Drug Name}, \text{Drug Accession ID} \big) \rightarrow \\
\text{Pathway}(\text{uri}(\text{Accession ID})) \ ^ \ \text{label}(\text{uri}(\text{Accession ID}), \text{Name}) \ ^ \ \\
\text{involves}(\text{uri}(\text{Accession ID}), \text{uri}(\text{Gene Accession ID})) \ ^ \ \\
\text{isTargetedBy}(\text{uri}(\text{Accession ID}), \text{uri}(\text{Drug Accession ID})) \ ^ \ \\
\text{isDisruptedBy}(\text{uri}(\text{Accession ID}), \text{uri}(\text{Disease Accession ID})) \ ^ \ \\
\text{Gene}(\text{uri}(\text{Gene Accession ID})) \ ^ \ \text{label}(\text{uri}(\text{Gene Accession ID}), \text{Gene Name}) \ ^ \ \\
\text{Drug}(\text{uri}(\text{Drug Accession ID})) \ ^ \ \text{label}(\text{uri}(\text{Drug Accession ID}), \text{Drug Name}) \ ^ \ \\
\text{Disease}(\text{uri}(\text{Disease Accession ID})) \ ^ \ \\
\text{label}(\text{uri}(\text{Disease Accession ID}), \text{Disease Name})
\]
Generation of the Source Descriptions: rule consequent

Node \(\rightarrow\) Class (unary predicate)

Edge \(\rightarrow\) binary predicate
- Object property (class to class)
- Data property (class to literal)

Use function symbols to create URIs:
- Pathway Accession ID = PA164713560
- \(\text{uri}(PA164713560) = \text{http://www.semanticweb.org/ontologies/bio#Pathway\_PA164713560}\)

Pathway(\(\text{uri(Accession\_ID)}\)) ^ label(\(\text{uri(Accession\_ID)}, \text{Name}\) ) ^ involves(\(\text{uri(Accession\_ID)}, \text{uri(Gene\_Accession\_ID)}\) ) ^ isTargetedBy(\(\text{uri(Accession\_ID)}, \text{uri(Drug\_Accession\_ID)}\) ) ^ isDisruptedBy(\(\text{uri(Accession\_ID)}, \text{uri(Disease\_Accession\_ID)}\) ) ^ Gene(\(\text{uri(Gene\_Accession\_ID)}\) ) ^ label(\(\text{uri(Gene\_Accession\_ID)}, \text{Gene\_Name}\) ) ^ Drug(\(\text{uri(Drug\_Accession\_ID)}\) ) ^ label(\(\text{uri(Drug\_Accession\_ID)}, \text{Drug\_Name}\) ) ^ Disease(\(\text{uri(Disease\_Accession\_ID)}\) ) ^ label(\(\text{uri(Disease\_Accession\_ID)}, \text{Disease\_Name}\) )
Generating the RDF

Evaluating the GLAV rule generates the desired RDF

• Data exchange from relational to RDF data (triples)

• Unary predicate $\rightarrow$ rdf:type triple

• Binary predicates $\rightarrow$ object or data property triples
  — If uri() function in both arguments of predicate, then object property, otherwise data property
@prefix s: 

s:Pathway_PA145011115 a category:Pathway .
s:Gene_PA27093 a category:Gene .
s:Drug_PA450947 a category:Drug .
s:Disease_PA444065 a category:Disease .

s:Pathway_PA145011115 property:Label "Phenytoin Pathway (PK)" .
s:Pathway_PA145011115 property:Involves s:Gene_PA27093 .
s:Pathway_PA145011115 property:IsTargetedBy s:Drug_PA450947 .
s:Pathway_PA145011115 property:IsDisruptedBy s:Disease_PA444065 .
s:Gene_PA27093 property:Label "CYP1A2" .
s:Drug_PA450947 property:Label "phenytoin" .
s:Disease_PA444065 property:Label "Epilepsy" .
Evaluation Methodology

• We evaluated our approach by integrating the same bioinformatics sources integrated by Becker et al.
  — PharmGKB
  — ABA
  — KEGG Pathway
  — UniProt

• We measured the following metrics:
  — Equivalence of the mappings generated by Karma to the manually generated Becker et al. R2R mappings
  — The effort required to produce the mappings in terms of the user actions required per source
Thee were 41 mappings, but there was no data for 2 of the mappings.

Of the remaining 39 mappings, 38 were semantically equivalent to the R2R mappings.

The remaining case required a data normalization rule in the mapping.

<table>
<thead>
<tr>
<th>Source</th>
<th>Table Name</th>
<th># Columns</th>
<th># User Actions</th>
<th>Assigning Type</th>
<th>Choosing Path</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>PharmGKB</td>
<td>Genes</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Drugs</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Diseases</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Pathways</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>ABA</td>
<td>Genes</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>KEGG Pathway</td>
<td>Pathways</td>
<td>6</td>
<td>5</td>
<td>0</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Diseases</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Genes</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Drugs</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>UniProt</td>
<td>Genes</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>39</td>
<td>24</td>
<td>9</td>
<td></td>
<td>33</td>
</tr>
</tbody>
</table>

Avg. User Actions/Property = \( \frac{33}{39} = 0.85 \)
Related Work

• Mapping Databases into RDF
  — D2R [Bizer & Cyganiak, 2006]
    ▪ Maps a database into RDF using the DB schema
  — R2R [Bizer & Shultz, 2010]
    ▪ Manually defines the mappings of D2R triples to another ontology

• Ontology Matching
  — [Doan et al., 2000]
    ▪ Learn mappings to the ontology using data, but would be analogous to just doing the semantic typing

• Schema Matching
  — [Rahm et al., 2001]
    ▪ Generates alignments between schemas, not a fine-grained model of the data

• Semantic Integration of Bioinformatics Data
  — Bio2RDF [Belleau et al., 2008]
    ▪ Manual conversion of sources into RDF
Discussion

• Presented an approach to map existing data sources directly into an ontology and generate the RDF
  — Automates as much of the mapping as possible
  — Allows the user to easily refine the mapping

• Makes it possible to rapidly integrate data sources over an integrated domain model

• Using the generated mapping rule, we are now working on supporting a SPARQL endpoint
  — The RDF data will be generated on the fly
Focus of This Paper

KARMA

Web  WWW
Excel, CSV
Database
KML
XML, RDF

clean  normalize
extract  model
publish  integrate
Overall Karma Effort

KARMA

- extract
- publish
- clean
- integrate
- normalize
- model

Web
Excel, CSV
Database
KML
XML, RDF

WWW
Excel, CSV
Database
KML
XML, RDF

32
• **More information available on Karma:**
  — [http://www.isi.edu/~knoblock](http://www.isi.edu/~knoblock)

• **Contact:**
  — [knoblock@isi.edu](mailto:knoblock@isi.edu) or [pszekely@isi.edu](mailto:pszekely@isi.edu)

• **Software:**
  — Software will be available as open source under the Apache license as soon as we complete the next version