Towards data fusion in a multi-ontology environment

Andriy Nikolov
Victoria Uren
Enrico Motta
Public linked data

As of March 2009
Issues

• Pairwise linking of datasets
  – Scale will grow
  – More effort needed to include “yet another” dataset to the cloud

• Automation would be useful
Challenges

- **Instance matching**
  - Aggregated attribute similarity
  - Usually configured manually for each pair of datasets and for each class
    - SILK, LinkedMDB, ...

- **Schema heterogeneity**
  - Which datasets overlap?
  - Which attributes to compare?

- **Employ automatic schema matching**

- **Scope**
  - `dbpedia:Company` vs `sweto:Company ∩ sweto:Bank`

- **Granularity**
  - `foaf:Person` vs `dbPedia:Politician`

- **Modelling style**
  - “red” vs `#FF0000`

- **Terminological**
  - `Company` vs `Corporation`
Schema matching

• Many existing tools (OAEI)
  – Lily
  – Falcon-AO
  – CIDER,
  – ...

• Features
  – Produce DL relations between concepts and attributes (≡, ⊆)
  – Focus on terminological mismatches
KnoFuss

- Designed for the corporate knowledge management scenario
- Single common schema
- Workflow
  - Coreference resolution
    - Attribute-based similarity
  - Coreference refinement
    - Analysis of links, constraints and provenance
- Extendable library of methods
Task decomposition

Knowledge fusion

Ontology integration

Source KB

Ontology matching

Instance transformation

Coreference resolution

Coreference refinement

Knowledge base integration

Target KB

CIDER, Lily, SCARLET ...

SPARQL query translation

X-MEDIA
Filtering

- Produce candidate mappings
- Remove conflicting mappings based on the similarity score
SELECT ?uri WHERE {
  ?uri rdf:type sweto:Computer_Science_Researcher
}

SELECT ?uri WHERE {
  { ?uri rdf:type tap:ComputerScientist } 
  UNION 
  { ?uri rdf:type tap:MedicalScientist } 
  UNION 
  { ?uri rdf:type tap:CMUPerson }
}
Setup

- **Datasets**
  - TAP
  - SWETO
  - DBPedia

- **Ontology matching**
  - CIDER (Gracia & Mena, 2008)
  - Lily (Wang & Xu, 2008)

- **Instance coreference resolution**
  - String similarity (Jaro-Winkler, L2 Jaro-Winkler)
Tests (F1-measure)

<table>
<thead>
<tr>
<th>Datasets</th>
<th>manual</th>
<th>CIDER</th>
<th>Lily</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAP/SWETO</td>
<td>0.77</td>
<td>0.76</td>
<td>0.42</td>
</tr>
<tr>
<td>TAP/DBPedia</td>
<td>0.88</td>
<td>0.66</td>
<td>0.44</td>
</tr>
<tr>
<td>SWETO/DBPedia</td>
<td>0.89</td>
<td>0.81</td>
<td>0.70</td>
</tr>
</tbody>
</table>

• Instance coreference resolution
  – String similarity (Jaro-Winkler, L2 Jaro-Winkler)
Conclusions

• Schema-level recall is important (even at the expense of precision)
  – CIDER outperformed Lily
  – Finding overlapping classes
• Restrictions are very useful
  – Disjointness, cardinality
  – Public reference ontology may help?
• Provenance of linksets is crucial
  – Extending coreference bundles?
Questions?

Thanks for your attention
• CIDER
  – All schema mappings above the threshold are accepted

• Lily
  – One-to-one schema mappings
  – “Competitive” schema mappings are removed
  – (+) Higher schema alignment precision
  – (-) Negative impact at the data level
Schema mismatches

- Conceptualisation
  - Scope
  - Model coverage & granularity

- Explication
  - Modelling style
  - Terminological
  - Encoding
Future work

• Original version
  – Sequential workflow
  – Schema integration -> data integration
  – Omitted schema mappings – lower data-level recall

• To do:
  – Iterative workflow (as in (Udrea et al., 2007))
  – Discovery of omitted schema mappings based on instance-level matches