6-23-2010

How To Make Linked Data More than Data

Prateek Jain  
*Wright State University - Main Campus*

Amit P. Sheth  
*Wright State University - Main Campus, amit.sheth@wright.edu*

Kunal Verma

Pascal Hitzler  
pascal.hitzler@wright.edu

Peter Z. Yeh

Follow this and additional works at: [https://corescholar.libraries.wright.edu/knoesis](https://corescholar.libraries.wright.edu/knoesis)

Part of the *Bioinformatics Commons, Communication Technology and New Media Commons, Databases and Information Systems Commons, OS and Networks Commons,* and the *Science and Technology Studies Commons*

Repository Citation

[https://corescholar.libraries.wright.edu/knoesis/95](https://corescholar.libraries.wright.edu/knoesis/95)

This Presentation is brought to you for free and open access by the The Ohio Center of Excellence in Knowledge-Enabled Computing (Kno.e.sis) at CORE Scholar. It has been accepted for inclusion in Kno.e.sis Publications by an authorized administrator of CORE Scholar. For more information, please contact corescholar@www.libraries.wright.edu, library-corescholar@wright.edu.
How To Make Linked Data More than Data


Prateek Jain, Pascal Hitzler, Amit Sheth
Kno.e.sis: Ohio Center of Excellence on Knowledge-enabled Computing
Wright State University, Dayton, OH
http://www.knoesis.org

Peter Z. Yeh, Kunal Verma
Accenture Technology Labs
San Jose, CA
What is Semantic Web Semantics?

- Semantic Web Semantics:
  - shareable (independent of your particular software)
  - declarative (not dependent on imperative algorithms)
  - computable (otherwise we don’t gain much)

You can do Mashups without Semantic Web semantics.
You can do information integration without Semantic Web semantics.
You can do most things without Semantic Web semantics.

But then it will be one-off, less scalable, less reusable.
In other words

We capture the meaning of information

not by specifying its meaning directly (which is impossible)
but by specifying, precisely,

how information interacts with other information.

We describe the meaning indirectly through its effects.

- An example (from LoD) of unintended errors when adequate semantics is not used:
  Linked MDB links to Dbpedia URI for Hollywood for country 🤪
Linked Open Data

Where is the semantics?
## Example: GeoNames

<table>
<thead>
<tr>
<th>Populated Place Features (city, village,...)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2,518,403 P.PPL</td>
<td>populated place</td>
</tr>
<tr>
<td>48,483 P.PPLX</td>
<td>section of populated place</td>
</tr>
<tr>
<td>39,336 P.PPLL</td>
<td>populated locality</td>
</tr>
<tr>
<td>13,306 P.PPLQ</td>
<td>abandoned populated place</td>
</tr>
<tr>
<td>2,684 P.PPLA4</td>
<td>seat of a fourth-order administrative division</td>
</tr>
<tr>
<td>2,028 P.PPLA</td>
<td>seat of a first-order administrative division</td>
</tr>
<tr>
<td>1,847 P.PPLW</td>
<td>destroyed populated place</td>
</tr>
<tr>
<td>1,006 P.PPLF</td>
<td>farm village</td>
</tr>
<tr>
<td>930 P.PPLA3</td>
<td>seat of a third-order administrative division</td>
</tr>
<tr>
<td>695 P.PPLD</td>
<td>seat of a second-order administrative division</td>
</tr>
<tr>
<td>253 P.PPLS</td>
<td>populated places</td>
</tr>
<tr>
<td>249 P.STLMT</td>
<td>israeli settlement</td>
</tr>
<tr>
<td>235 P.PPLC</td>
<td>capital of a political entity</td>
</tr>
<tr>
<td>57 P.</td>
<td></td>
</tr>
<tr>
<td>29 P.PPLR</td>
<td>religious populated place</td>
</tr>
<tr>
<td>6 P.PPLG</td>
<td>seat of government of a political entity</td>
</tr>
<tr>
<td>2,629,547 Total for P</td>
<td></td>
</tr>
</tbody>
</table>

**Where Is the semantics?**
“Nancy Pelosi voted in favor of the Health Care Bill.”

Where is the semantics?
Don’t get us wrong

Linked Open Data is great, useful, cool, and a very important step.

But if we stay semantics-free, Linked Open Data will be of limited usefulness!
To leverage LoD, we require **schema knowledge**

- **application-type driven** (reusable for same kind of application)
- **less messy than LoD** (as required by application)
- **overarching several LoD datasets** (as required by application)
Schema on top of the LoD cloud
Schema on top of the LOD Cloud

• Obvious solution to create an ontology capturing the relationships on top of the LOD Schema datasets.

• Perform a matching of the LOD Schemas using state of the art ontology matching tools.

• The datasets can be mapped to an upper level ontology which can capture the relationships.

• Considering the size, heterogeneity and complexity of LOD, at least have results which can be curated by a human being.
## LOD Schema Alignment using state of the art tools

<table>
<thead>
<tr>
<th>Dataset</th>
<th>System-1</th>
<th></th>
<th>System-2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>Music, BBC</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Music, Dbpedia</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Geonames, DBpedia</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Average</td>
<td>0.0</td>
<td>0.0</td>
<td>0.33</td>
<td>0.0</td>
</tr>
</tbody>
</table>
LOD Schema Alignment

• State of the art Ontology Alignment systems have difficulty in matching LOD Schemas!
  ➢ Nation = Menstruation, Confidence=0.9 😞

• They are tuned to perform on the established benchmarks, but do not seem to work well in more unconstrained/preselected cases. Most current systems excel on Ontology Alignment Evaluation Initiative Benchmark.

• LOD Schemas are of very different nature
  • Created by community for community.
  • LOD has so far emphasized number of instances, not number of meaningful relationships.
  • Require solutions beyond syntactic and structural matching.
Research Agenda

Two components

• Enrich schemas to capture semantics – how data in different datasets/bubbles are logically related (BLOOM)

• Support Federated Queries – a system that automates query processing involving multiple, related datasets (LOCUS)
Step 1: Enrich Schemas

BLOOMS – Bootstrapping based Linked Open Data Ontology Matching Systems.
Step 1: Semantic Enrichment

• BLOOMS – Bootstrapping based Linked Open Data Ontology Matching Systems.

• At the highest level of abstraction our approach takes in two different ontologies and tries to match them using the following steps

   (1) Using Alignment API to identify direct correspondences.

   (2) Using the categorization of concepts using Wikipedia.

   (3) Running a reasoner on the results found using step (2) and directly on the ontologies.
Utilizes the Wikipedia Web service to identify the matching concepts.

- Thus for the term Conductor the following definitions are obtained
  - Electrical Conductor
  - Conducting
  - Conductor_(album)
  - Conductor (architecture)
  - Mr. Conductor
  - Conductor (ring theory)

These terms correspond to articles on Wikipedia for the concepts in the ontology.
• Next step utilize the Web service for identifying Wikipedia categories for building the Wikipedia category tree.
• For each different sense of concept c, match it with the different possible senses of the c’.

Conductor

Conducting

cat:Occupation s_in_music
cat:Music performance

cat: Arts_occupations

Artist
cat: Arts occupations
Using the position of the categories identify the relationships.

Thus this helps in identifying approximately the relationship between the various concepts.

Ponzetto & Strube, 2007
Disconnected Classes

• Some senses do not relate to each other

Thus this helps in identifying disconnected relationships.
Equivalent Classes

- Some senses are identical to each other

Lady_Finger → Okra → cat: Abelmoschus → cat: Hibusceae

Okra → cat: Abelmoschus → cat: Hibusceae → cat: Malvoideae

Thus this helps in identifying equivalence relationships.
## LOD Schema Alignment using BLOOMS

<table>
<thead>
<tr>
<th>Dataset</th>
<th>System-1</th>
<th></th>
<th>System-2</th>
<th></th>
<th>Our Approach</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>Precision</td>
<td>Recall</td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>Music, BBC</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td>0.63</td>
<td>0.78</td>
</tr>
<tr>
<td>Music, Dbpedia</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td>0.39</td>
<td>0.62</td>
</tr>
<tr>
<td>FOAF, DBpedia</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td>0.67</td>
<td>0.73</td>
</tr>
<tr>
<td>Average</td>
<td>0.0</td>
<td>0.0</td>
<td>0.33</td>
<td>0.0</td>
<td></td>
<td>0.56</td>
<td>0.71</td>
</tr>
</tbody>
</table>

Testing done on 10 different pairs of LOD schemas
Linked Schema’s

Music Ontology Schema

DBpedia Ontology

Geonames

Music Brainz

Jamendo

DBTunes

BBC Program

SWC

FOAF

SIOC

Pisa

IEEE

ACM

AKT Portal Ontology
Observations

- Heavy connections at instance level, do not translate to schema level.
  - Case in point: Geonames and Dbpedia. only SpatialThing in Geonames matches to Dbpedia concepts.

- No connections at instance level, DOES NOT mean anything.
  - Case in point: Dbpedia and AKT Reference Ontology have over 100+ relationship between concepts.
  - Possibility to create links between instance level. Example: Dbpedia “Scientist” Class can contain “Computer Scientist”.

- Schema level connections and reasoning can be used for cleaning up LOD Cloud.
  - dbpedia:Hollywood rdf:type dbpedia:Country
  - dbpedia:Country disjointWith uscensus:Community
  - uscensus:Hollywood rdf:type uscensus:Community
Step 2: Integrated Access/Federated Querying

LOQUS: Linked Open Data SPARQL Querying System (LOQUS)
Federated Querying

- Transform a query and broadcast it to a group of disparate and relevant datasets with the appropriate syntax.
- Merging the results collected from the datasets.
- Presenting them succinctly and unified format with least duplication.
- Automatically sort the merged result set.
Federated Querying Challenges

• User is required to have intimate knowledge about the domain of datasets.
• User needs to understand the exact structure of datasets.
• For each relevant dataset user needs to form separate queries.
• Entity disambiguation has to be performed on similar entities.
• Retrieved results have to be processed and merged.
Querying Federated Sources

Identify artists, whose albums have been tagged as punk and the population of the places they are based near.
Relevant Datasets

Music Ontology

Geonames Data

Location | Census ID
---|---
Malibu, CA | Census:5907

Census Data

Artist | Location
---|---
Lifehouse | Malibu, CA

Census ID | Population
---|---
Census:5907 | 12,575
Querying the Datasets

**Music Ontology**

Give me artists with punk as genre and their locations?

**Census Data**

Give me population figures of geographical entities?

**Geonames Data**

Give me the identifier used by Census Bureau for geographic locations?
• Linked Open Data SPARQL Querying System.

• User can pose federated queries without having to know the exact structure and links between the different datasets.

• Automatically maps user’s query to the relevant datasets using mapping repository created using BLOOMS.

• Executes individual queries and merges the results into a single, complete answer.
Traditionally to Retrieve Results

Perform disambiguation

Music Data
Geographic Data
Census Data

User has to ....

<table>
<thead>
<tr>
<th>Musician</th>
<th>Location</th>
<th>Population</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Clash</td>
<td>London</td>
<td>7,556,900</td>
</tr>
<tr>
<td>MC5</td>
<td>Lincoln Park, Michigan</td>
<td>40,008</td>
</tr>
<tr>
<td>Green Day</td>
<td>East Bay, California, U.S.</td>
<td>NA</td>
</tr>
</tbody>
</table>
A single source of reference consisting of mapping to the specific LOD datasets.

• Module to identify concepts contained in the query and perform the translations to the LOD cloud datasets.

• Module to split the query mapped to LOD datasets concepts into sub-queries corresponding to different datasets.

• Module to execute the queries remotely and process the results and deliver the final result to the user.
Identify artists, whose albums have been tagged as punk and the population of the places they are based near.

Query is decomposed into sub-queries. User looks up mapping repository to identify concepts of interest and formulates query.
Results are returned for the sub-queries.
Partial results are processed for union, join and disambiguation by LOQUS.
LOQUS combines the results and presents them back to the user.

<table>
<thead>
<tr>
<th>Musician</th>
<th>Location</th>
<th>Population</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Clash</td>
<td>London</td>
<td>7,556,900</td>
</tr>
<tr>
<td>MC5</td>
<td>Lincoln Park, Michigan</td>
<td>40,008</td>
</tr>
<tr>
<td>Green Day</td>
<td>East Bay, California, U.S.</td>
<td>NA</td>
</tr>
</tbody>
</table>
Technology Stack

Proprietary software

LOQUS  BLOOMS

Open Source Technologies

Jena/ARQ  SPARQL  RDF

Linked Open Data cloud  Java

Java
## LOQUS Advantage

<table>
<thead>
<tr>
<th>Traditional Query “Federation” (Manual)</th>
<th>LOQUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. User required to know different datasets individually</td>
<td>1. User looks at a single dataset which is mapped to the different datasets.</td>
</tr>
<tr>
<td>2. User has to form individual queries for the different datasets.</td>
<td>2. A single query expressed using the single dataset is necessary. Individual queries are formed automatically.</td>
</tr>
<tr>
<td>3. User has to execute the queries separately on each dataset.</td>
<td>3. Queries are automatically executed on the relevant datasets.</td>
</tr>
<tr>
<td>4. Query results have to be processed manually for unification, disambiguation and such.</td>
<td>4. Query results are processed automatically for join, unification and disambiguation.</td>
</tr>
</tbody>
</table>

LOQUS expects just the query from the user and does rest of the work.
Conclusions

• LOD cloud is an important start, but more needs to be done to make it useful – esp to make integrated use of multiple datasets

• Semantic relationships and descriptions across ontologies is a key enabler to provide integrated access/use (for example, federated queries)
Conclusions…. continued

• BLOOMS is one approach for semi-automatically linking different ontologies
  – A new approach for ontology mapping that leverages knowledge in DBPedia

• A more semantic LOD cloud can enable more intelligent applications such as open question answering
  – LOQUS shows how enriched schemas can enable automatic federated queries, making LOD significantly more useful
References

- Prateek Jain, Pascal Hitzler, Peter Z. Yeh, Kunal Verma, Amit P. Sheth, Linked Data is Merely More Data, AAAI Spring Symposium "Linked Data Meets Artificial Intelligence", March 22-24, 2010

- Prateek Jain, Kunal Verma, Pascal Hitzler, Peter Z. Yeh, Amit P. Sheth, “LOQUS: Linked Open Data SPARQL Querying System”
Thanks!

This work is funded primarily by NSF Award:IIS-0842129, titled "III-SGER: Spatio-Temporal-Thematic Queries of Semantic Web Data: a Study of Expressivity and Efficiency".

More at Kno.e.sis – Ohio Center of Excellence on Knowledge-enabled Computing: [http://knoesis.org](http://knoesis.org)