Making the Semantic Web Easier to Use

Tim Finin
University of Maryland, Baltimore County

Joint work with Lushan Han, Varish Mulwad, Anupam Joshi

Presented at the Workshop on Data Engineering Meets the Semantic Web of the IEEE International Conference on Data Engineering, 1 April 2012

http://ebiq.org/r/339

Overview

• Linked Open Data 101
• Two ongoing UMBC dissertations
  – Varish Mulwad, Generating linked data from tables
  – Lushan Han, Querying linked data with a quasi-NL interface

Linked Open Data (LOD)

• Linked data is just RDF data, typically just the instances (ABOX), not schema (TBOX)
• RDF data is a graph of triples
  – URI URI string
dbr:Barack_Obama dbo:spouse “Michelle Obama”
  – URI URI URI
dbr:Barack_Obama dbo:spouse dbpedia:Michelle_Obama
• Best linked data practice prefers the 2nd pattern, using nodes rather than strings for “entities”
• Liked open data is just linked data freely accessible on the Web along with any required ontologies

Semantic Web

Use Semantic Web Technology to publish shared data & knowledge

Semantic web technologies allow machines to share data and knowledge using common web language and protocols.

~ 1997

Semantic Web beginning
Use Semantic Web Technology to publish shared data & knowledge.

Data is inter-linked to support integration and fusion of knowledge.

**LOD beginning**

**LOD growing**

Use Semantic Web Technology to publish shared data & knowledge.

Data is inter-linked to support integration and fusion of knowledge.

**... and growing**

**...growing faster**

LOD is the new Cyc: a common source of background knowledge.
Linked Open Data

Use Semantic Web Technology to publish shared data & knowledge

LOD is the new Cyc: a common source of background knowledge

Data is inter-linked to support integration and fusion of knowledge

2011: 31B facts in 295 datasets interlinked by 504M assertions on ckan.net

Exploiting LOD not (yet) Easy

• Publishing or using LOD data has inherent difficulties for the potential user
  – It’s difficult to explore LOD data and to query it for answers
  – It’s challenging to publish data using appropriate LOD vocabularies & link it to existing data
• Problem: $O(10^4)$ schema terms, $O(10^{11})$ instances
• I’ll describe two ongoing research projects that are addressing these problems

Generating Linked Data by Inferring the Semantics of Tables

Research with Varish Mulwad

http://ebiq.org/j/96

Goal: Table => LOD*

<table>
<thead>
<tr>
<th>Name</th>
<th>Team</th>
<th>Position</th>
<th>Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>Michael Jordan</td>
<td>Chicago</td>
<td>Shooting guard</td>
<td>1.98</td>
</tr>
<tr>
<td>Allen Iverson</td>
<td>Philadelphia</td>
<td>Point guard</td>
<td>1.83</td>
</tr>
<tr>
<td>Yao Ming</td>
<td>Houston</td>
<td>Center</td>
<td>2.29</td>
</tr>
<tr>
<td>Tim Duncan</td>
<td>San Antonio</td>
<td>Power forward</td>
<td>2.11</td>
</tr>
</tbody>
</table>

* DBpedia

http://dbpedia.org/resource/Allen_Iverson

http://dbpedia.org/class/yago/NationalBasketballAssociationTeams

http://dbpedia.org/dataprop/team

Player height in meters
Goal: Table => LOD*

<table>
<thead>
<tr>
<th>Name</th>
<th>Team</th>
<th>Position</th>
<th>Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>Michael Jordan</td>
<td>Chicago</td>
<td>Shooting guard</td>
<td>1.98</td>
</tr>
<tr>
<td>Allen Iverson</td>
<td>Philadelphia</td>
<td>Point guard</td>
<td>1.83</td>
</tr>
<tr>
<td>Yao Ming</td>
<td>Houston</td>
<td>Center</td>
<td>2.29</td>
</tr>
<tr>
<td>Tim Duncan</td>
<td>San Antonio</td>
<td>Power forward</td>
<td>2.11</td>
</tr>
</tbody>
</table>

@prefix dbpedia: <http://dbpedia.org/resource/>
@prefix dbo: <http://dbpedia.org/ontology/>
@prefix yago: <http://dbpedia.org/class/yago/>

"Name"@en is rdfs:label of dbo:BasketballPlayer.
"Team"@en is rdfs:label of yago:NationalBasketballAssociationTeams.
"Michael Jordan"@en is rdfs:label of dbpedia:Michael Jordan.
"Chicago Bulls"@en is rdfs:label of dbpedia:Chicago Bulls.
"dbpedia:Michael Jordan a dbo:BasketballPlayer."
"yago:MichaelJordan a yago:BasketballTeam."

All this in a completely automated way

Tables are everywhere!! ... yet ...

The web – **154 million** high quality relational tables

Evidence-based medicine judges the efficacy of treatments or tests by meta-analyses of clinical trials. Key information is often found in tables in articles. However, the rate at which meta-analyses are published remains very low... hampers effective health care treatment...

<table>
<thead>
<tr>
<th># of Clinical trials published in 2008</th>
<th># of meta analysis published in 2008</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

~ **400,000** datasets 😊

~ < 1% in RDF 😞
2010 Preliminary System

Class prediction
Entity Linking

Examples of class labels:
Column – Nationality
Prediction – MilitaryConflict
Column – Birth Place
Prediction – PopulatedPlace

Sources of Errors

- The *sequential* approach let errors percolate from one phase to the next
- The system was biased toward predicting overly general classes over more appropriate specific ones
- Heuristics largely drive the system
- Although we consider multiple sources of evidence, we did not joint assignment

A Domain Independent Framework

Query Mechanism
Ranking the candidates

- \( C_i = \text{"State"; } L_{C_i} = \text{AdministrativeRegion} \)
- \( f_1 = \{ \text{Levenshtein distance}(C_i, L_{C_i}), \text{Dice Score } \langle C_i, L_{C_i} \rangle, \text{Semantic Similarity } \langle C_i, L_{C_i} \rangle, \text{InformationGain}(L_{C_i}) \} \)
- \( \psi_1 = \exp(w_1 \top f_1(C_i, L_{C_i})) \)

Ranking the candidates

- \( R_{ij} = \text{"Baltimore"; } E_{ij} = \text{Baltimore_Maryland} \)
- \( f_2 = \{ \text{Levenshtein distance}(R_{ij}, E_{ij}), \text{Dice Score } \langle R_{ij}, E_{ij} \rangle, \text{PageRank } (E_{ij}), \text{KBscore } (E_{ij}), \text{PageLength } (E_{ij}) \} \)
- \( \psi_2 = \exp(w_2 \top f_2(R_{ij}, E_{ij})) \)

Joint Inference over evidence in a table

- ✔ Probabilistic Graphical Models

A graphical model for tables

- Joint inference over evidence in a table

Class

- Team
  - Chicago
  - Philadelphia
  - Houston
  - San Antonio

Instance

- C1
- R11
- R21
- R31
- C2
- R12
- R22
- R32
- C3
- R13
- R23
- R33
Parameterized graphical model

Captures interaction between row values

Function that captures the affinity between the column headers and row values

Factor Node

Row value

Variable Node: Column header

Captures interaction between column headers

Challenge: Interpreting Literals

Many columns have literals, e.g., numbers

<table>
<thead>
<tr>
<th>Population</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>690,000</td>
<td>75</td>
</tr>
<tr>
<td>345,000</td>
<td>65</td>
</tr>
<tr>
<td>510,020</td>
<td>50</td>
</tr>
<tr>
<td>120,000</td>
<td>25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Profit in $K ?</th>
<th>Population</th>
</tr>
</thead>
<tbody>
<tr>
<td>690,000</td>
<td>120,000</td>
</tr>
<tr>
<td>345,000</td>
<td>120,000</td>
</tr>
<tr>
<td>510,020</td>
<td>120,000</td>
</tr>
</tbody>
</table>

- Predict properties based on cell values
- Cyc had hand coded rules: humans don’t live past 120
- We extract value distributions from LOD resources
  - Differ for subclasses: age of people vs. political leaders vs. athletes
  - Represent as measurements: value + units
- Metric: possibility/probability of values given distribution

Other Challenges

- Using table captions and other text is associated documents to provide context
- Size of some data.gov tables (> 400K rows!) makes using full graphical model impractical
  - Sample table and run model on the subset
- Achieving acceptable accuracy may require human input
  - 100% accuracy unattainable automatically
  - How best to let humans offer advice and/or correct interpretations?

PMI as an association measure

We use pointwise mutual information (pmi) to measure the association between two RDF resources (nodes)

\[ pmix\ y = \log \frac{p(x, y)}{p(x)p(y)} = \log \frac{p(x|y)}{p(x)} = \log \frac{p(y|x)}{p(y)}. \]

pmi is used for word association by comparing how often two words occur together in text to their expected co-occurrence if independent
PMI for RDF instances

- For text, the co-occurrence context is usually a window of some number of words (e.g., 50)
- For RDF instances, we count three graph patterns as instances of the co-occurrence of N1 and N2

N1 — N2

N1 — N2

N1 — N2

- Other graph patterns can be added, but we’ve not evaluated their utility or cost to compute.

PMI for RDF types

- We also want to measure the association strength between RDF types, e.g., a dbo:Actor associated with a dbo:Film vs. a dbo:Place
- We can also measure the association of an RDF property and types, e.g. dbo:author used with a dbo:Film vs. a dbo:Book
- Such simple statistics can be efficiently computed for large RDF collections in parallel

PREFIX dbo: <http://dbpedia.org/ontology/>

GoRelations: Intuitive Query System for Linked Data

Research with Lushan Han

http://ebiq.org/j/93

Dbpedia is the Stereotypical LOD

- DBpedia is an important example of Linked Open Data
  - Extracts structured data from Infoboxes in Wikipedia
  - Stores in RDF using custom ontologies Yago terms
- The major integration point for the entire LOD cloud
- Explorable as HTML, but harder to query in SPARQL
Querying LOD is Much Harder

- Querying DBpedia requires a lot of a user
  - Understand the RDF model
  - Master SPARQL, a formal query language
  - Understand ontology terms: 320 classes & 1600 properties!
  - Know instance URLs (>1M entities!)
  - Term heterogeneity (Place vs. PopulatedPlace)
- Querying large LOD sets overwhelming
- Natural language query systems still a research goal

Goal

- Allow a user with a basic understanding of RDF to query DBpedia and ultimately distributed LOD collections
  - To explore what data is in the system
  - To get answers to question
  - To create SPARQL queries for reuse or adaptation
- Desiderata
  - Easy to learn and to use
  - Good accuracy (e.g., precision and recall)
  - Fast

Key Idea

Structured keyword queries
Reduce problem complexity by:
- User enters a *simple graph*, and
- Annotates the nodes and arcs with *words and phrases*
Structured Keyword Queries

- Nodes denote entities and links binary relations
- Entities described by two unrestricted terms: *name* or value and *type* or concept
- Result entities marked with ? and those not with *
- A compromise between a natural language Q&A system and SPARQL
  - Users provide compositional structure of the question
  - Free to use their own terms in annotating the structure

**Translation – Step One**

finding semantically similar ontology terms

For each concept or relation in the graph, generate the k most semantically similar candidate ontology classes or properties

- Users provide compositional structure of the question
- Free to use their own terms in annotating the structure

**Another Example**

Football players who were born in the same place as their team’s president

**Translation – Step Two**

disambiguation algorithm

- To assemble the best interpretation we rely on *statistics of the data*
- Primary measure is *pointwise mutual information* (PMI) between RDF terms in the LOD collection
  - This measures the degree to which two RDF terms occur together in the knowledge base
- In a reasonable interpretation, *ontology terms associate* in the way that their corresponding *user terms* connect in the structured keyword query
Translation – Step Two
disambiguation algorithm

Three aspects are combined to derive an overall goodness measure for each candidate interpretation

Joint disambiguation:
\[
\text{argmax}_{P_1, P_2, \ldots, P_n} \text{goodness}(G) = \text{argmax}_{i=1}^{\infty} \text{goodness}(L_i)
\] (1)

Resolving direction:

If \( \text{PM}(c(O_1), p(R_1)) + \text{PM}(p(R_1), c(S_1)) \)
\( \text{PM}(c(S_1), p(R_1)) + \text{PM}(p(R_1), c(O_1)) > \alpha \)

Then \( S_i = O_i, O' = S_i \)
Else \( S_i = S_i, O' = O_i \)

Link reasonableness:

\[
\text{goodness}(L_i) = \text{PM}(c(S_i), p(R_i)) \cdot \text{sim}(S_i', c(S_i')) \cdot \text{sim}(R_i, c(R_i))
+ \text{PM}(p(R_i), c(S_i')) \cdot \text{sim}(O_i', c(O_i')) \cdot \text{sim}(R_i, p(R_i))
+ \text{PM}(c(S_i'), c(O_i')) \cdot \text{sim}(S_i', c(S_i')) \cdot \text{sim}(O_i', c(O_i'))
\] (3)

Example of Translation result

Concepts: Place => Place, Author => Writer, Book => Book
Properties: born in => birthPlace, wrote => author (inverse direction)

SPARQL Generation

The translation of a semantic graph query to SPARQL is straightforward given the mappings

Concepts
- Place => Place
- Author => Writer
- Book => Book

Relations
- born in => birthPlace
- wrote => author

Evaluation

- 33 test questions from 2011 Workshop on Question Answering over Linked Data answerable using DBpedia
- Three human subjects unfamiliar with DBpedia translated the test questions into semantic graph queries
- Compared with two top natural language QA systems: PowerAqua and True Knowledge

<table>
<thead>
<tr>
<th>System</th>
<th>Prec.</th>
<th>Recall</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>GoRelations</td>
<td>0.687</td>
<td>0.722</td>
<td>0.704</td>
</tr>
<tr>
<td>regular</td>
<td>0.736</td>
<td>0.803</td>
<td>0.768</td>
</tr>
<tr>
<td>concise</td>
<td>0.372</td>
<td>0.483</td>
<td>0.420</td>
</tr>
<tr>
<td>PowerAqua</td>
<td>0.334</td>
<td>0.483</td>
<td>0.395</td>
</tr>
<tr>
<td>all triples</td>
<td>0.255</td>
<td>0.291</td>
<td>0.272</td>
</tr>
<tr>
<td>merged</td>
<td>0.469</td>
<td>0.535</td>
<td>0.500</td>
</tr>
</tbody>
</table>
Current challenges

- Baseline system works well for DBpedia
- Current challenges we are addressing are
  - Adding direct entity matching
  - Relaxing the need for type information
  - Testing on other LOD collections and extending to a set of distributed LOD collections
  - Developing a better Web interface
  - Allowing user feedback and advice

See [http://ebiq.org/93](http://ebiq.org/93) for more information & try our alpha version at [http://ebiq.org/GOR](http://ebiq.org/GOR)
Final Conclusions

• Linked Data is an emerging paradigm for sharing structured and semi-structured data
  – Backed by machine-understandable semantics
  – Based on successful Web languages and protocols
• Generating and exploring Linked Data resources can be challenging
  – Schemas are large, too many URIs
• New tools for mapping tables to Linked Data and translating structured natural language queries help reduce the barriers