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
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 2\(^{nd}\) 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 => Linked Open Data

Use Semantic Web Technology to publish shared data & knowledge

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

LOD beginning
Semantic Web => Linked Open Data

Use Semantic Web Technology to publish shared data & knowledge

Data is interlinked to support integration and fusion of knowledge

LOD growing
Semantic Web => Linked Open Data

Use Semantic Web Technology to publish shared data & knowledge

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

2010...growing faster
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 interlinked 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
<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
## 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 .
dbpedia:Michael Jordan a dbo:BasketballPlayer .

"Chicago Bulls"@en is rdfs:label of dbpedia:Chicago Bulls .
dbpedia:Chicago Bulls a yago:NationalBasketballAssociationTeams .

All this in a completely automated way

* DBpedia
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.
~ 400,000 datasets 😊є

~ < 1 % in RDF 😞
2010 Preliminary System

T2LD Framework

Predict Class for Columns

Linking the table cells

Identify and Discover relations

Class prediction

Entity Linking

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

Graph showing % of correct and incorrect instances linked by category:
- Person: 83.05% Correct, 16.95% Incorrect
- Place: 80.43% Correct, 19.57% Incorrect
- Organization: 61.90% Correct, 38.10% Incorrect
- Other: 29.22% Correct, 70.78% Incorrect

Categories

<table>
<thead>
<tr>
<th>Categories</th>
<th>Correct</th>
<th>Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person</td>
<td>83.05%</td>
<td>16.95%</td>
</tr>
<tr>
<td>Place</td>
<td>80.43%</td>
<td>19.57%</td>
</tr>
<tr>
<td>Organization</td>
<td>61.90%</td>
<td>38.10%</td>
</tr>
<tr>
<td>Other</td>
<td>29.22%</td>
<td>70.78%</td>
</tr>
</tbody>
</table>
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

Pre-processing modules

Sampling → Acronym detection

Query and generate initial mappings → Joint Inference/Assignment

Generate Linked RDF → Verify (optional) → Store in a knowledge base & publish as LOD

City | Mayor      | State | Population
--- | ---------- |------- |------------
Boston | T. Menino  | MA     | 610,000
New York | M. Bloomberg | NY   | 8,400,000
Philadelphia | M. Nutter  | PA     | 1,500,000
Baltimore | S. Dixon   | MD     | 640,000
Washington | A. Fenty   | DC     | 595,000
# Query Mechanism

**Team**

<table>
<thead>
<tr>
<th>Michael Jordan</th>
<th>Chicago Bulls</th>
<th>Shooting Guard</th>
<th>1.98</th>
</tr>
</thead>
</table>

{dbo:Place,dbo:City,yago:WomenArtist,yago:LivingPeople,yago:NationalBasketballAssociationTeams...}

**possible types**

**possible entities**

Chicago Bulls, Chicago, Judy Chicago ...

.........
Ranking the candidates

• $C_i = \text{"State"}$ ; $L_{ci} = \text{AdministrativeRegion}$

String in column header

• $f_1 = [\text{Levenshtein distance}(C_i, L_{Ci}), \text{Dice Score } (C_i, L_{Ci}), \text{Semantic Similarity } (C_i, L_{Ci}), \text{InformationGain}(L_{Ci})]$

Class from an ontology

String similarity metrics

• $\psi_1 = \exp(w_1^T f_1(C_i, L_{Ci}))$
Ranking the candidates

- $R_{ij} = \text{“Baltimore”}; E_{ij} = \text{Baltimore}_{-\text{Maryland}}$

- $f_2 = \left[ \text{Levenshtein distance}(R_{ij}, E_{ij}), \right.$
  $\text{Dice Score}(R_{ij}, E_{ij}), \text{PageRank}(E_{ij}), \text{KBScore}(E_{ij}), \text{PageLength}(E_{ij}) \left. \right]$

- $\psi_2 = \exp(w_2^T f_2(R_{ij}, E_{ij}))$
Joint Inference over evidence in a table

✓ Probabilistic Graphical Models
A graphical model for tables
Joint inference over evidence in a table
Parameterized graphical model

- Function that captures the affinity between the column headers and row values
- Captures interaction between row values
- Captures interaction between column headers
- Variable Node: Column header
- Factor Node
- Row value

Nodes: R11, R12, R13, R21, R22, R23, R31, R32, R33, C1, C2, C3, ψ4, ψ3, ψ5

Connecting arrows indicate relationships and interactions between nodes.
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>

- 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)

\[
\text{pmi}(x; y) \equiv \log \frac{p(x, y)}{p(x)p(y)} = \log \frac{p(x|y)}{p(x)} = \log \frac{p(y|x)}{p(y)}.
\]

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

• 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
### Mark Twain

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbpedia-owl:birthDate</td>
<td>1835-11-30 (xsd:date)</td>
</tr>
<tr>
<td>dbpedia-owl:birthName</td>
<td>Samuel Langhorne Clemens</td>
</tr>
<tr>
<td>dbpedia-owl:birthPlace</td>
<td>Florida, Missouri</td>
</tr>
<tr>
<td>dbpedia-owl:child</td>
<td>Jean Clemens, Susy Clemens, Clara Clemens</td>
</tr>
<tr>
<td>dbpedia-owl:deathDate</td>
<td>1910-04-21 (xsd:date)</td>
</tr>
<tr>
<td>dbpedia-owl:deathPlace</td>
<td>Redding, Connecticut</td>
</tr>
<tr>
<td>dbpedia-owl:genre</td>
<td>Essay, Historical_fiction, Social_commentary, Satire, Non-fiction, Philosophy_and_literature, Literary_criticisim, Travel_literature, Fiction, Children's_literature</td>
</tr>
<tr>
<td>dbpedia-owl:notableWork</td>
<td>Adventures_of_Huckleberry_Finn, The_Adventures_of_Tom_Sawyer</td>
</tr>
<tr>
<td>dbpedia-owl:occupation</td>
<td>Writer, Lecture</td>
</tr>
<tr>
<td>dbpedia-owl:pseudonym</td>
<td>Mark Twain</td>
</tr>
<tr>
<td>dbpedia-owl:thumbnail</td>
<td><a href="http://upload.wikimedia.org/wikipedia/commons/thumb/c/c5/Mark_Twain_by_AF_Brady_S.jpg">http://upload.wikimedia.org/wikipedia/commons/thumb/c/c5/Mark_Twain_by_AF_Brady_S.jpg</a></td>
</tr>
<tr>
<td>dbpedia-owl:pageTitle</td>
<td><a href="http://www.sfgate.com/cgi-bin/article.cgi?file=/c/a/2005/08/19/MNGOBEA9JI1.DTL">http://www.sfgate.com/cgi-bin/article.cgi?file=/c/a/2005/08/19/MNGOBEA9JI1.DTL</a></td>
</tr>
<tr>
<td></td>
<td><a href="http://www.pbs.org/marktwain/scrapbook/index.html">http://www.pbs.org/marktwain/scrapbook/index.html</a></td>
</tr>
<tr>
<td></td>
<td><a href="http://www.kamakurapens.com/TwainsConklin.html">http://www.kamakurapens.com/TwainsConklin.html</a></td>
</tr>
<tr>
<td>dbpedia-owl:wikipediaLink</td>
<td><a href="http://www.ucmerced.edu/faculty/facultybio.asp?facultyid=95">http://www.ucmerced.edu/faculty/facultybio.asp?facultyid=95</a></td>
</tr>
</tbody>
</table>
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 URIs (>1M entities!)
  – Term heterogeneity (Place vs. PopulatedPlace)

• Querying large LOD sets overwhelming

• Natural language query systems still a research goal

  - Best NLP techniques have difficulty to understand complex questions
  - Familiar with ontology terms
  - SPARQL syntax and URIs
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

Similarity metric is distributional similarity, LSA, and WordNet.
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

\[
\arg\max_{p_1, \ldots, p_m, c_1, \ldots, c_n \in H} \text{goodness}(G) = \arg\max_{p_1, \ldots, p_m, c_1, \ldots, c_n \in H} \sum_{i=1}^{m} \text{goodness}(L_i) \quad (1)
\]

Resolving direction

\[
\text{If } [\text{PMI}(c(O_i), p(R_i)) + \text{PMI}(p(R_i), c(S_i))]
- [\text{PMI}(c(S_i), p(R_i)) + \text{PMI}(p(R_i), c(O_i))] > \alpha
\]
Then \( S_i' = O_i, \ O_i' = S_i \)
Else \( S_i' = S_i, \ O_i' = O_i \)  \quad (2)

Link reasonableness

\[
\text{goodness}(L_i) = \max(\text{PMI}(c(S_i'), p(R_i)) \cdot \text{sim}(S_i', c(S_i')) \cdot \text{sim}(R_i, p(R_i)))
+ \text{PMI}(p(R_i), c(O_i')) \cdot \text{sim}(O_i', c(O_i')) \cdot \text{sim}(R_i, p(R_i)), \beta)
+ \text{PMI}(c(S_i'), c(O_i')) \cdot \text{sim}(S_i', c(S_i')) \cdot \text{sim}(O_i', c(O_i')) \quad (3)
\]
Example of Translation result

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

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

SELECT DISTINCT ?x, ?y WHERE {
  ?0 a dbo:Book .
  ?0 rdfs:label ?label0 .
  ?label0 bif:contains "The adventures of Tom Sawyer" .
  ?x a dbo:Writer .
  ?y a dbo:Place .
  {?0 dbo:author ?x} .
  {?x dbo:birthPlace ?y} .
}
```
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></th>
<th>Prec.</th>
<th>Recall</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>GoRelations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>regular</td>
<td>0.687</td>
<td>0.722</td>
<td>0.704</td>
</tr>
<tr>
<td>concise</td>
<td>0.736</td>
<td>0.803</td>
<td>0.768</td>
</tr>
<tr>
<td>PowerAqua</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1st triple</td>
<td>0.372</td>
<td>0.483</td>
<td>0.420</td>
</tr>
<tr>
<td>all triples</td>
<td>0.334</td>
<td>0.483</td>
<td>0.395</td>
</tr>
<tr>
<td>merged</td>
<td>0.255</td>
<td>0.291</td>
<td>0.272</td>
</tr>
<tr>
<td>True Knowledge</td>
<td>0.469</td>
<td>0.535</td>
<td>0.500</td>
</tr>
</tbody>
</table>
Please input relations in your query. One per line.

?x/American Football Player, date of birth, ?y/Date
?x, height, ?y/Number

Examples: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

Query  Relaxed Query  See Translations

Message: This query gives data about the height of American football players and their date of birth. Football fans may feel it interesting to know how the height of football players changes over time.

http://ebiq.org/GOR

Powered by DBpedia and Virtuoso
2011 Ebiquity Lab, UMBC.
<table>
<thead>
<tr>
<th>x</th>
<th>z</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://dbpedia.org/resource/Jack_Shaprio">http://dbpedia.org/resource/Jack_Shaprio</a></td>
<td>1.5494</td>
<td>1907-03-22</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Josh_Betts">http://dbpedia.org/resource/Josh_Betts</a></td>
<td>1.5748</td>
<td>1982-08-25</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Nate_Abrams">http://dbpedia.org/resource/Nate_Abrams</a></td>
<td>1.6256</td>
<td>1897-12-25</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Herbert_Clow">http://dbpedia.org/resource/Herbert_Clow</a></td>
<td>1.6256</td>
<td>1899-05-07</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Earl_Warweg">http://dbpedia.org/resource/Earl_Warweg</a></td>
<td>1.6764</td>
<td>1892-01-11</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Jacquizz_Rodgers">http://dbpedia.org/resource/Jacquizz_Rodgers</a></td>
<td>1.6764</td>
<td>1990-02-06</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/John_Bralier">http://dbpedia.org/resource/John_Bralier</a></td>
<td>1.6764</td>
<td>1876-12-27</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Menz_Lindsey">http://dbpedia.org/resource/Menz_Lindsey</a></td>
<td>1.6764</td>
<td>1897-07-25</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Stefan_Logan">http://dbpedia.org/resource/Stefan_Logan</a></td>
<td>1.6764</td>
<td>1981-06-02</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Cory_Ross">http://dbpedia.org/resource/Cory_Ross</a></td>
<td>1.6764</td>
<td>1982-09-22</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Eddie_Scharer">http://dbpedia.org/resource/Eddie_Scharer</a></td>
<td>1.6764</td>
<td>1990-01-26</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/John_Barrett_%28American_football%29">http://dbpedia.org/resource/John_Barrett_%28American_football%29</a></td>
<td>1.6764</td>
<td>1899-02-25</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Dick_Thornton_%28American_football%29">http://dbpedia.org/resource/Dick_Thornton_%28American_football%29</a></td>
<td>1.6764</td>
<td>1908-02-04</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Walter_French_%28baseball%29">http://dbpedia.org/resource/Walter_French_%28baseball%29</a></td>
<td>1.7018</td>
<td>1899-07-12</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/AL_Loeb">http://dbpedia.org/resource/AL_Loeb</a></td>
<td>1.7018</td>
<td>1890-03-11</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Antraun_Carter">http://dbpedia.org/resource/Antraun_Carter</a></td>
<td>1.7018</td>
<td>1981-09-09</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Terry_Caulley">http://dbpedia.org/resource/Terry_Caulley</a></td>
<td>1.7018</td>
<td>1984-06-22</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Kendall_Hunter">http://dbpedia.org/resource/Kendall_Hunter</a></td>
<td>1.7018</td>
<td>1988-09-16</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Mark_McMillian">http://dbpedia.org/resource/Mark_McMillian</a></td>
<td>1.7018</td>
<td>1970-04-29</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/AL_Consweet">http://dbpedia.org/resource/AL_Consweet</a></td>
<td>1.7018</td>
<td>1996-07-16</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Bodie_Weldon">http://dbpedia.org/resource/Bodie_Weldon</a></td>
<td>1.7018</td>
<td>1895-11-07</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Johnny_Bryan">http://dbpedia.org/resource/Johnny_Bryan</a></td>
<td>1.7018</td>
<td>1897-02-28</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Andrew_Hawkins">http://dbpedia.org/resource/Andrew_Hawkins</a></td>
<td>1.7018</td>
<td>1996-03-10</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Chad_Owens">http://dbpedia.org/resource/Chad_Owens</a></td>
<td>1.7018</td>
<td>1982-04-05</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Mark_Higgs">http://dbpedia.org/resource/Mark_Higgs</a></td>
<td>1.7018</td>
<td>1984-04-11</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Emmett_McLemore">http://dbpedia.org/resource/Emmett_McLemore</a></td>
<td>1.7018</td>
<td>1894-08-08</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Danny_Woodhead">http://dbpedia.org/resource/Danny_Woodhead</a></td>
<td>1.7018</td>
<td>1894-08-17</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/resource/Garrett_Wolfe">http://dbpedia.org/resource/Garrett_Wolfe</a></td>
<td>1.7018</td>
<td>1980-02-10</td>
</tr>
</tbody>
</table>
Three Top Interpretations of the User Query

Top One Interpretation

Before Step Three

```
* ?x/American Football Player has a candidate list including 0:AmericanFootballPlayer 1.00 1:Person 0.25 (the selected choice is AmericanFootballPlayer 1.00)
* ?z/Number has a candidate list including 0:Number 1.00 1:#Group 0.31 2:#Code 0.31 3:EthnicGroup 0.25 4:#Size 0.23 5:#Series 0.23 6:Single 0.21 7:OlympicResult 0.15 8:MountainRange 0.14 (the selected choice is Number 1.00)
* ?y/Date has a candidate list including 0:Date 1.00 1:Grape 0.19 2:Holiday 0.18 3:Name 0.18 4:Place 0.15 5:Event 0.15 6:GIVENAME 0.15 7:HistoricPlace 0.14 8:WrestlingEvent 0.14 9:PopulatedPlace 0.12 10:SportsEvent 0.12 11:WorldHeritageSite 0.12 12:NaturalPlace 0.12 13:MixedMartialArtsEvent 0.11 (the selected choice is Date 1.00)
```

After Step Three

```
* ?x/American Football Player => AmericanFootballPlayer
* ?z/Number => Number
* ?y/Date => Date
& ?x/American Football Player date of birth: ?y/Date => 0:@birthDate 1.00
& ?x/American Football Player height: ?z/Number => 0:@height 1.00
```

The Regular SPARQL Query

```
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?x, ?z, ?y WHERE {
  ?x a dbo:AmericanFootballPlayer .
  ?x dbo:birthDate ?y .
  ?x dbo:height ?z .
}
```

The Concise SPARQL Query

```
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?x, ?z, ?y WHERE {
  ?x dbo:AmericanFootballPlayer .
  ?x dbo:birthDate ?y .
  ?x dbo:height ?z .
}
```

Top Two Interpretation

Before Step Three

```
* ?x/American Football Player has a candidate list including 0:AmericanFootballPlayer 0.25 (the selected choice is AmericanFootballPlayer 1.00)
* ?z/Number has a candidate list including 0:Number 1.00 1:#Group 0.31 2:#Code 0.31 3:EthnicGroup 0.25 4:#Size 0.23 5:#Series 0.23 6:Single 0.21 7:OlympicResult 0.15 8:MountainRange 0.14 (the selected choice is Number 1.00)
* ?y/Date has a candidate list including 0:Date 1.00 1:Grape 0.19 2:Holiday 0.18 3:Name 0.18 4:Place 0.15 5:Event 0.15 6:GIVENAME 0.15 7:HistoricPlace 0.14 8:WrestlingEvent 0.14 9:PopulatedPlace 0.12 10:SportsEvent 0.12 11:WorldHeritageSite 0.12 12:NaturalPlace 0.12 13:MixedMartialArtsEvent 0.11 (the selected choice is Date 1.00)
```

http://ebiq.org/GOR
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 for more information & try our alpha version at 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