Building Semantic Descriptions of Sources

Craig Knoblock
University of Southern California
Linked Open Data and Services

- Vast collection of interlinked information
- Various sources and services with different schemas
Where do the Semantics Come From?

• **Linked Open Data**
  - Populated by manually linking or writing procedures that define the links across sources
  - But we don’t know how the sources are related
  - In many cases there is no or very limited semantic descriptions of sources

• **Linked Open Services**
  - Manually constructed or built by wrapping existing Web services
  - Constructing the lifting and lowering rules that relate the services to existing ontologies is a difficult task
  - Even when done, it may only provide a partial description
    - e.g., descriptions of the inputs and outputs, but not the function of a service

Monday, August 22, 2011
Outline of the Talk

• Linked Open Data
  • Building and linking ontologies of linked data

• Linked Open Services
  • Building semantic web services from the Deep Web

• Discussion
  • Remaining challenges
Outline of the Talk

- Linked Open Data
  - Building and linking ontologies of linked data
- Linked Open Services
  - Building semantic web services from the Deep Web
- Discussion
  - Remaining challenges
Building and linking ontologies of linked data [Parundekar et al., ISWC 2010]
Disjoint Schemas

<table>
<thead>
<tr>
<th>Source 1</th>
<th>Source 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Schema Level</strong></td>
<td><strong>Instance Level</strong></td>
</tr>
<tr>
<td>Ciudad</td>
<td>City</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>NO LINKS!!</td>
<td>Los Angeles</td>
</tr>
<tr>
<td>owl:sameAs</td>
<td></td>
</tr>
</tbody>
</table>

Monday, August 22, 2011
Objective 1: Find Schema Alignments

<table>
<thead>
<tr>
<th>Source 1</th>
<th>Source 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Schema Level</strong></td>
<td></td>
</tr>
<tr>
<td>Ciudad</td>
<td>City</td>
</tr>
<tr>
<td><strong>Instance Level</strong></td>
<td></td>
</tr>
<tr>
<td>Los Angeles</td>
<td>City of Los Angeles</td>
</tr>
</tbody>
</table>
Ontologies of Linked Data

- Ontologies can be highly specialized
  - e.g. DBpedia has classes for Educational Institutions, Bridges, Airports, etc.

- Ontologies can be rudimentary
  - e.g. in Geonames all instances only belong to a single class – ‘Feature’
  - Derived from RDBMS schemas from which Linked Data was generated

- There might not exist exact equivalences between classes in two sources
• Only subset relations possible with difference in class specializations

<table>
<thead>
<tr>
<th>Schema Level</th>
<th>Geonames</th>
<th>DBpedia</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature</td>
<td>⊂</td>
<td>Educational Institution</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance Level</th>
<th>University of Southern California</th>
<th>University of Southern California</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>owl:sameAs</td>
</tr>
</tbody>
</table>
• A specialized class can be created by restricting the value of one or more properties

• The following Venn diagram explains a restriction class in Geonames with a restriction on the value of the featureCode property as ‘S.SCH’

Set of all instances in Original Class – rdf:type=Feature

Set of all instances in Restricted Class – rdf:type=Feature & featureCode=S.SCH
Objective 2: Find Alignments Between Restriction Classes

- Find and model specialized descriptions of classes

<table>
<thead>
<tr>
<th>Geonames</th>
<th>DBpedia</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Schema Level</strong></td>
<td></td>
</tr>
<tr>
<td>rdf:type=Feature &amp; featureCode=S.SCH</td>
<td>rdf:type=Educational Institution</td>
</tr>
<tr>
<td><strong>Instance Level</strong></td>
<td></td>
</tr>
<tr>
<td>University of Southern California</td>
<td>University of Southern California</td>
</tr>
</tbody>
</table>

Monday, August 22, 2011
Nature of Restriction Classes

• Instances belonging to a restriction class also belong to parent restriction class
  • e.g. restrictions from Geonames below

```
(rdf:type=geonames:Feature)  (featureCode=geonames:A.PCLI)  (featureClass=geonames:A)
```

```
(rdf:type=geonames:Feature & featureClass=geonames:A)  
(rdf:type=geonames:Feature & featureCode=geonames:A.PCLI)  
(featureCode=geonames:A.PCLI & featureClass=geonames:A)  
```

• This also results in a hierarchy in the alignments, which our algorithm exploits
## Extensional Approach to Ontology Alignment

<table>
<thead>
<tr>
<th>ClassA is disjoint from ClassB</th>
<th>ClassA is equivalent to ClassB</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Diagram 1" /></td>
<td><img src="image2" alt="Diagram 2" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ClassA is subset of ClassB</th>
<th>ClassB is subset of ClassA</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image3" alt="Diagram 3" /></td>
<td><img src="image4" alt="Diagram 4" /></td>
</tr>
</tbody>
</table>
Alignment Hypotheses

- An alignment hypothesis considers aligning:
  - a restriction class from ontology $O_1$
  - another restriction class from ontology $O_2$

- Find relation between the two restriction classes:
  - using extensional comparison on set of instances belonging to each restriction class
  - Use instance pair identifiers from pre-processing step (combination of URIs of linked instances)
Exploration of Hypotheses Search Space

Seed hypotheses generation

Seed hypothesis pruning (owl:Thing covers all instances)

Pruning on empty set $r_2 = \emptyset$

Prune as no change in the extension set
## Example Alignments from LinkedGeoData, Geonames, and DBpedia

<table>
<thead>
<tr>
<th>#</th>
<th><strong>LinkedGeoData</strong> restriction</th>
<th><strong>DBpedia</strong> restriction</th>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>rdf:type=lgd:node</code></td>
<td><code>rdf:type/owl:Thing</code></td>
<td>$r_1 = r_2$</td>
</tr>
<tr>
<td>2</td>
<td><code>rdf:type=lgd:aerodrome</code></td>
<td><code>rdf:type=dbpedia:Airport</code></td>
<td>$r_1 = r_2$</td>
</tr>
<tr>
<td>3</td>
<td><code>rdf:type=lgd:island</code></td>
<td><code>rdf:type=dbpedia:Island</code></td>
<td>$r_1 = r_2$</td>
</tr>
<tr>
<td>4</td>
<td><code>lgd:gnis.%3AST_alpha=NJ</code></td>
<td><code>dbpedia:Place#type=http://dbpedia.org/resource/City_(New_Jersey)</code></td>
<td>$r_1 = r_2$</td>
</tr>
<tr>
<td>5</td>
<td><code>rdf:type=lgd:village</code></td>
<td><code>rdf:type=dbpedia:PopulatedPlace</code></td>
<td>$r_1 \subseteq r_2$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#</th>
<th><strong>Geonames</strong> restriction</th>
<th><strong>DBpedia</strong> restriction</th>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td><code>geonames:featureClass=geonames:P</code></td>
<td><code>rdf:type=dbpedia:PopulatedPlace</code></td>
<td>$r_1 = r_2$</td>
</tr>
<tr>
<td>7</td>
<td><code>geonames:featureClass=geonames:H</code></td>
<td><code>rdf:type=dbpedia:BodyOfWater</code></td>
<td>$r_1 = r_2$</td>
</tr>
<tr>
<td>8</td>
<td><code>geonames:parentFeature=http://sws.geonames.org/3174618/</code></td>
<td><code>dbpedia:City_region=http://dbpedia.org/resource/Lombardy</code></td>
<td>$r_1 = r_2$</td>
</tr>
<tr>
<td>9</td>
<td><code>geonames:featureCode=geonames:S.SCH</code></td>
<td><code>rdf:type=dbpedia:EducationalInstitution</code></td>
<td>$r_1 = r_2$</td>
</tr>
<tr>
<td>11</td>
<td><code>geonames:featureCode=geonames:T.MT</code></td>
<td><code>rdf:type=dbpedia:Mountain</code></td>
<td>$r_1 \subseteq r_2$</td>
</tr>
</tbody>
</table>
Outline of the Talk

• Linked Open Data
  • Building and linking ontologies of linked data

• Linked Open Services
  • Building semantic web services from the Deep Web

• Discussion
  • Remaining challenges
Building semantic web services from the Deep Web [Ambite et al., ISWC 2009]

- Automatically build semantic models for data and services available on the larger Web
- Construct models of these sources that are sufficiently rich to support querying and integration
  - Build models for the vast amount of structured and semi-structured data available
    - Not just web services, but also form-based interfaces
    - E.g., Weather forecasts, flight status, stock quotes, currency converters, online stores, etc.
  - Learn models for information-producing web sources and web services
Approach

- Start with some initial knowledge of a domain
  - Sources and semantic descriptions of those sources
- Automatically
  - Discover related sources
  - Determine how to invoke the sources
  - Learn the syntactic structure of the sources
  - Identify the semantic types of the data
  - Build semantic models of the source
Monday, August 22, 2011
Automatically Discover and Build Semantic Web Services for Related Sources

Unisys Weather

Unisys Weather: Forecast for Washington, DC (20502) [0] 2

Latest Observation for Washington, DC (20502)

Partly Cloudy

Site: KJDN (Washington/Natl, VA)

Almanac

Sunrise: 7:02 AM

Sunset: 4:45 PM

Temperature:

Temp: 45°F (7°C)

Dewpoint: 22°F (-5°C)

Relative Humidity: 45%

Winds: W at 7 knt

Wind Chill: 41°F

Pressure: 1010.1 mb (29.84 in)

Visibility: 10 mi

Skies: partly cloudy

Weather:

Alerts

No alerts

Forecast Summary

WEDNESDAY THURSDAY FRIDAY SATURDAY SUNDAY MONDAY TUESDAY

Sunny

Hi: 45

Lo: 32

Sunny

Hi: 52

Lo: 35

Sunny

Hi: 48

Lo: 35

Sunny

Hi: 48

Lo: 35

Sunny

Hi: 52

Lo: 32

Detailed forecast from National Weather Service

DISTRICT OF COLUMBIA-ARLINGTON-FALLS CHURCH-LEXANDRIA

INCLUDING THE CITIES OF: WASHINGTON, ALEXANDRIA, FALLS CHURCH

3 PM EST TUE NOV 29 2008

TODAY

Hi: 46

LO: 32

MOSTLY CLOUDY. LOWS IN THE LOWER 30S. SOUTHWEST WINDS AROUND 10 MPH.

WEDNESDAY

Hi: 45

MOSTLY SUNNY. HIGHS IN THE MID 40S. WEST WINDS 10 TO 15 MPH.

WEDNESDAY NIGHT

Lo: 35

PARTLY CLOUDY. LOWS IN THE MID 30S. WEST WINDS 5 TO 10 MPH.

THURSDAY NIGHT

Lo: 35

SUNNY. HIGHS IN THE LOWER 50S. SOUTHWEST WINDS 5 TO 10 MPH.

FRIDAY

Hi: 52

Lo: 32

SUNNY. HIGHS IN THE LOW 40S. WEST WINDS 5 TO 10 MPH.
discovery

Invocation & extraction

Semantic typing

Source modeling

Background knowledge

- Seed URL
- Sample input values
- Sample values
- Definition of known sources
- Patterns
- Domain types

unisys(Zip, Temp, ...)

http://wunderground.com

unisys(Zip, Temp, Humidity, ...)

Invocation & extraction

unisys

unisys

Monday, August 22, 2011
Semantic Typing
[Lerman, Plangprasopchok, & Knoblock]

✓ Idea: Learn a model of the content of data and use it to recognize new examples

**Background knowledge** 

**Patterns**

<table>
<thead>
<tr>
<th>Person</th>
<th>Address</th>
<th>Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>E Lewis</td>
<td>3518 Hilltop Rd</td>
<td>(419) 531 - 0504</td>
</tr>
<tr>
<td>Andrew Lewis</td>
<td>3543 Larchmont Pkwy</td>
<td>(518) 474 - 3326</td>
</tr>
<tr>
<td>C. S. Lewis</td>
<td>555 Willow Run Dr</td>
<td>(612) 578 - 2355</td>
</tr>
<tr>
<td>Carmen Jones</td>
<td>355 Morgan Ave N</td>
<td>(612) 522 - 3533</td>
</tr>
<tr>
<td>John Jones</td>
<td>3574 Brookside Rd</td>
<td>(555) 531 - 9566</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Location</th>
<th>State_prov</th>
<th>Postal_code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toledo</td>
<td>OH</td>
<td>64325-3000</td>
</tr>
<tr>
<td>Toledo</td>
<td>OH</td>
<td>64356</td>
</tr>
<tr>
<td>Seattle</td>
<td>WA</td>
<td>8422</td>
</tr>
<tr>
<td>Seattle</td>
<td>WA</td>
<td>8435</td>
</tr>
<tr>
<td>Omaha</td>
<td>NE</td>
<td>52456-6444</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Person</th>
<th>Address</th>
<th>Telephone</th>
</tr>
</thead>
<tbody>
<tr>
<td>E Lewis</td>
<td>3518 Hilltop Rd</td>
<td>(419) 531 - 0504</td>
</tr>
<tr>
<td>Andrew Lewis</td>
<td>3543 Larchmont Pkwy</td>
<td>(518) 474 - 4799</td>
</tr>
<tr>
<td>C. S. Lewis</td>
<td>555 Willow Run Dr</td>
<td>(612) 578 - 5555</td>
</tr>
<tr>
<td>Carmen Jones</td>
<td>355 Morgan Ave N</td>
<td>(612) 522 - 5555</td>
</tr>
<tr>
<td>John Jones</td>
<td>3574 Brookside Rd</td>
<td>(555) 531 - 9566</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Location</th>
<th>City</th>
<th>State</th>
<th>Zipcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toledo</td>
<td>OH</td>
<td>OH</td>
<td>64325-3000</td>
</tr>
<tr>
<td>Toledo</td>
<td>OH</td>
<td>OH</td>
<td>64356</td>
</tr>
<tr>
<td>Seattle</td>
<td>WA</td>
<td>WA</td>
<td>8422</td>
</tr>
<tr>
<td>Seattle</td>
<td>WA</td>
<td>WA</td>
<td>8435</td>
</tr>
<tr>
<td>Omaha</td>
<td>NE</td>
<td>NE</td>
<td>52456-6444</td>
</tr>
</tbody>
</table>
**Inducing Source Definitions**

- **Known Source 1**
- **Known Source 2**
- **Known Source 3**
- **New Source 4**

### Step 1: classify input & output semantic types

- `source1($zip, lat, long) :- centroid(zip, lat, long).`
- `source2($lat1, $long1, $lat2, $long2, dist) :- greatCircleDist(lat1, long1, lat2, long2, dist).`
- `source3($dist1, dist2) :- convertKm2Mi(dist1, dist2).`

```prolog
source4( $startZip, $endZip, separation)
```

- **zipcode**
- **distance**

---

Monday, August 22, 2011
Generating Plausible Definition
[Carman & Knoblock, 2007]

• Step 1: classify input & output semantic types
• Step 2: generate plausible definitions

known source 1
known source 2
known source 3

source1($zip, lat, long) :-
    centroid(zip, lat, long).

source2($lat1, $long1, $lat2, $long2, dist) :-
    greatCircleDist(lat1, long1, lat2, long2, dist).

source3($dist1, dist2) :-
    convertKm2Mi(dist1, dist2).

source4($zip1, $zip2, dist) :-
    source1(zip1, lat1, long1),
    source1(zip2, lat2, long2),
    source2(lat1, long1, lat2, long2, dist2),
    source3(dist2, dist).

source4($zip1, $zip2, dist) :-
    centroid(zip1, lat1, long1),
    centroid(zip2, lat2, long2),
    greatCircleDist(lat1, long1, lat2, long2, dist2),
    convertKm2Mi(dist1, dist2).
Invoke and Compare the Definition

- Step 1: classify input & output semantic types
- Step 2: generate plausible definitions
- Step 3: invoke service & compare output

```
source4($zip1, $zip2, dist):-
    source1(zip1, lat1, long1),
    source1(zip2, lat2, long2),
    source2(lat1, long1, lat2, long2, dist2),
    source3(dist2, dist).
```

```
source4($zip1, $zip2, dist):-
    centroid(zip1, lat1, long1),
    centroid(zip2, lat2, long2),
    greatCircleDist(lat1, long1, lat2, long2, dist2),
    convertKm2Mi(dist1, dist2).
```

<table>
<thead>
<tr>
<th>$zip1</th>
<th>$zip2</th>
<th>dist (actual)</th>
<th>dist (predicted)</th>
</tr>
</thead>
<tbody>
<tr>
<td>80210</td>
<td>90266</td>
<td>842.37</td>
<td>843.65</td>
</tr>
<tr>
<td>60601</td>
<td>15201</td>
<td>410.31</td>
<td>410.83</td>
</tr>
<tr>
<td>10005</td>
<td>35555</td>
<td>899.50</td>
<td>899.21</td>
</tr>
</tbody>
</table>

match
ForecastDay = one-of(0,1,2,3,4,5) ;;
0 is today, 1 is tomorrow, ...

DEIMOS generated Web Service

z90292 hasName 90292 .

w1 hasZIP z90292 .
w1 hasTemp 61° F .
...
w1 hasZIP z90292 .
w2 hasLowTemp 59° F .
## Accuracy of the Models

<table>
<thead>
<tr>
<th>domain</th>
<th>Precision</th>
<th>Recall</th>
<th>F$_1$-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>weather</td>
<td>0.64</td>
<td>0.29</td>
<td>0.39</td>
</tr>
<tr>
<td>geospatial</td>
<td>1.00</td>
<td>0.86</td>
<td>0.92</td>
</tr>
<tr>
<td>flights</td>
<td>0.69</td>
<td>0.35</td>
<td>0.46</td>
</tr>
<tr>
<td>currency</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>mutualfund</td>
<td>0.72</td>
<td>0.30</td>
<td>0.42</td>
</tr>
</tbody>
</table>
Outline of the Talk

• Linked Open Data
  • Building and linking ontologies of linked data

• Linked Open Services
  • Building semantic web services from the Deep Web

• Discussion
  • Remaining challenges
Discussion

• Initial work described here just scratches the surface of the problem
  • Goal is to both populate the Web of linked data and have rich semantic models of the data
  • Building semantic descriptions of linked open data will allow us to better understand the available sources and use the sources in a broad range of applications
  • Methods for automatically constructing linked open services will improve the coverage and quality of the sources available
Some Challenges

- **Linked Open Data**
  - How do we build an overall class hierarchy for a source
  - How do the relations map across sources
  - What do we do about missing and extraneous links

- **Linked Open Services**
  - How do we improve the accuracy of the learned semantic descriptions
  - How can we learn semantic descriptions that go beyond the current sources
  - How do we learn mappings between enumerated types (e.g., “Arrived” vs. “Landed”)