DBpedia.org

April 22nd, 2011 - Matt Harbers

Outline

- DBPedia:
  - What is it?
  - How's the data structured?
  - Where does the data come from?
  - Accessing the data
- Query Examples

DBpedia.org

- Community whose goal is to provide web based information from Wikipedia data
- Allows users to ask sophisticated questions
- Links data sets together across the web
- Describes more than 3.5 million things which are broken down into categories
  - People
  - Places
  - Music Album
  - Films
  - Video games
  - Organizations
  - Species
  - Diseases
    - etc

What kind of Data?

- Dataset is represented in a cross-domain ontology that was manually created by members of the community
- 272 classes based on information in Wikipedia infoboxes
    - organized in hierarchy under "owl:Thing"
    - infoboxes are grey "summary" boxes in top right of Wikipedia pages
- Organization of classes:
    - Means of Transportation parent of:
        - aircraft, ship, automobile, etc
    - Event parent of:
        - music festival, military conflict, convention, etc

Structure of Data

- OWL ontology describing all classes
- Data must be mapped from Wikipedia to DBpedia
  - data from Wikipedia not stored in standardized way
  - creation of data and properties decentralized by many users.
  - eg.
    - birthplace & placeofbirth property names describe same data

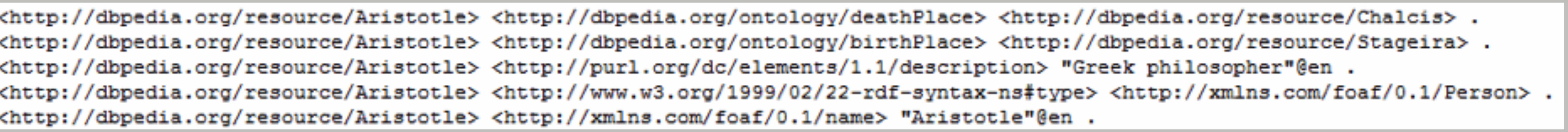
STRUCTURE OF DATA (CONTD)

- Example of class ontology:

```
<owl:Class rdf:about="http://dbpedia.org/ontology/Person">
    <rdfs:label xml:lang="en">person</rdfs:label>
    <rdfs:label xml:lang="de">Person</rdfs:label>
    <rdfs:label xml:lang="pt">pessoa</rdfs:label>
    <rdfs:label xml:lang="fr">personne</rdfs:label>
    <rdfs:subClassOf  rdf:resource = "http://www.w3.org/2002/07/owl#Thing"></rdfs:subClassOf>
     <owl:equivalentClass rdf:resource="http://xmlns.com/foaf/0.1/Person"></owl:equivalentClass>
</owl:Class>
```

- Support for multiple languages
- "Person" is one level below root.
- Mapping to FOAF makes machine-readable
- Ontology Classes
- http://dbpedia.org/ontology/Person

Structure of Instance



```
<http://dbpedia.org/resource/Aristotle> <http://dbpedia.org/ontology/deathPlace> <http://dbpedia.org/resource/Chalcis> .
<http://dbpedia.org/resource/Aristotle> <http://dbpedia.org/ontology/birthPlace> <http://dbpedia.org/resource/Stageira> .
<http://dbpedia.org/resource/Aristotle> <http://purl.org/dc/elements/1.1/description> "Greek philosopher"@en .
<http://dbpedia.org/resource/Aristotle> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
<http://dbpedia.org/resource/Aristotle> <http://xmlns.com/foaf/0.1/name> "Aristotle"@en .
```

- Instance Property description of "Person"
- Subject, predicate, object
- Predicates/Objects can be DBpedia defined (deathPlace) or standards defined (foaf)
- Objects can be literal values ("Greek Philosopher")
- Objects can be DBPedia/Standards defined:
    - foaf/Person
    - DBpedia defined

Relationships

- Resources may reference other resources by relationships
- Relationships can be represented as edges in a large web of data
- You can follow these relationships to other resources

Relationship Examples

(http://www.visualdataweb.org/relfinder/relfinder.php):

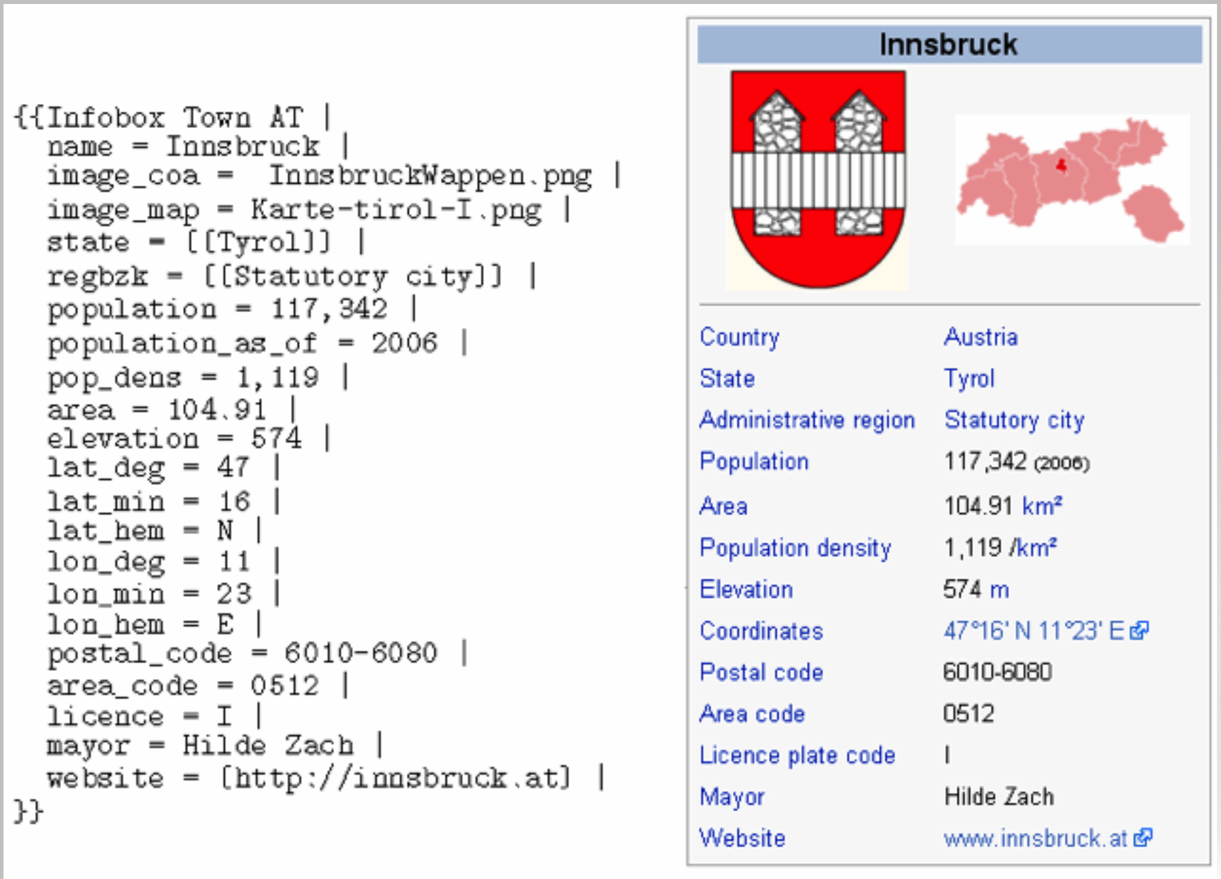
RelFinder -

Viewable Relationships:

- Porsche, Volkswagen, Allan McNish, Audi
- Physics, Albert Einstein, Literature (then + Barack Obama)
- George Clooney, O Brother, Where Art Though + John Turturro (start clicking on classes)

Wikipedia Data

- Most Wikipedia data is unstructured
- infobox templates, categorization information, images, geo information, and external url links are structured, however



```
{{Infobox Town AT |
  name = Innsbruck |
  image_coa =  InnsbruckWappen.png |
  image_map = Karte-tirol-I.png |
  state = [[Tyrol]] |
  regbzk = [[Statutory city]] |
  population = 117,342 |
  population_as_of = 2006 |
  pop_dens = 1,119 |
  area = 104.91 |
  elevation = 574 |
  lat_deg = 47 |
  lat_min = 16 |
  lat_hem = N |
  lon_deg = 11 |
  lon_min = 23 |
  lon_hem = E |
  postal_code = 6010-6080 |
  area_code = 0512 |
  licence = I |
  mayor = Hilde Zach |
  website = [http://innsbruck.at] |
}}
```

Innsbruck

Country	Austria
State	Tyrol
Administrative region	Statutory city
Population	117,342 (2006)
Area	104.91 km²
Population density	1,119 /km²
Elevation	574 m
Coordinates	47°16' N 11°23' E
Postal code	6010-6080
Area code	0512
Licence plate code	I
Mayor	Hilde Zach
Website	www.innsbruck.at

Wikipedia Data Gathering

- DBpedia gathers data using an automated extractor
  - pulls all infobox data from all articles in Wikipedia
  - pulls multiple languages
- Very little clean-up is done to the data
  - "June 2009 changed to 2009-06"
    - xml friendly
- Downside:
  - over 8000 property types exist
- Mapping of Wikipedia Infoboxes to DBpedia classes is done by hand to correct weaknesses in the Wikipedia model
  - more than 1 infobox may exists for the same article

Accessing Data

- Browse Data:
  - either looking through RDF manually
  - using tool like RelFinder
  - hard to get value
- Third Party Tools:
  - use underlying SPARQL queries
  - Display search results in html format with links to resource information
  - SPARQL queries require an intimate knowledge of data set
  - Not practical for a wide web use

QUERY EXAMPLES

DBpedia SPARQL (http://dbpedia.org/snorql/):

All “Things” about Atlanta:

```
SELECT * WHERE {
    <http://dbpedia.org/resource/Atlanta> ?p ?o .
    FILTER (LANG(?o)='en') .
}
```

QUERY EXAMPLES (SPARQL)

People who were born in Germany before the year 1800, but died in Paris:

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

SELECT * WHERE {
    ?person dbo:birthDate ?birth .
    ?person dbo:deathPlace :Paris.
    ?person dbo:birthPlace :Germany .
    ?person foaf:name ?name .
    ?person rdfs:comment ?description .
    FILTER (LANG(?description) = 'en') .
    FILTER (?birth < "1800-01-01"^^xsd:date) .
}
ORDER BY ?name
```

Query Examples (SPARQL)

Schools within 10km of Atlanta:

```
SELECT DISTINCT ?Link ?SchoolName ?EstablishedDate ?lat ?long
WHERE
  {
    <http://dbpedia.org/resource/Atlanta> geo:geometry ?sourcegeo .
    ?resource geo:geometry ?matchgeo .
    ?resource geo:lat ?lat .
    ?resource geo:long ?long .
    FILTER ( bif:st_intersects ( ?matchgeo, ?sourcegeo, 5 ) ) .
    ?Link ?somelink ?resource .
    ?Link <http://dbpedia.org/property/established> ?EstablishedDate .
    ?Link rdfs:label ?SchoolName .
    FILTER ( lang ( ?SchoolName ) = "en" )
  }
order by ?SchoolName
```

Third Party Tools

- Third Party Search Engines:
  - Text based searching
  - Facet based searching
    - Similar to Web Stores "filtering" results.
    - can also use text searching
    - Very powerful method of searching

Query Examples

Faceted Searching: (http://dbpedia.neofonie.de/browse)

- Large High Elevation Cities

SUMMARY

- Very powerful and meaningful results are produceable
- Relies heavily on crowd sourcing data and manual mapping
  - categorization of classes, wiki to dbpedia mapping, error correction.
- Data needs to be pre-formatted and stored in a place where accessing the data set is fast. (too big to cache)
- Error in data set makes searching difficult

Questions?