OUTLINE

• DBPedia:
  o What is it?
  o How's the data structured?
  o Where does the data come from?
  o Accessing the data
• Query Examples
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
Example of class ontology:

```xml
<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

- 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
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

(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)
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-1.png |
  state = {{Tyrol}} |
  regbz = {{Statutory city}} |
  population = 117,342 |
  population_as_of = 2006 |
  pop_dens = 1,119 |
  area = 104.91 |
  elevation = 674 |
  lat_deg = 47 |
  lat_min = 16 |
  lat_hem = N |
  lon_deg = 11 |
  lon_min = 23 |
  lon_hem = E |
  postal_code = 6010-8080 |
  area_code = 0612 |
  licence = I |
  mayor = Hilde Zach |
  website = [http://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 exist for the article
ACCESSING DATA

• Browse Data:
  o either looking through RDF manually
  o using tool like RelFinder
  o hard to get value

• Third Party Tools:
  o use underlying SPARQL queries
  o Display search results in html format with links to resource information
  o SPARQL queries require an intimate knowledge of data set
  o Not practical for a wide web use
QUERY EXAMPLES

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

All “Things” about Atlanta:
SELECT * WHERE {
  FILTER (LANG(?o)='en') .
}
People who were born in Germany before the year 1800, but died in Paris:

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

SELECT * WHERE {
  ?person dbo:birthPlace :Germany .
  ?person foaf:name ?name .
  FILTER (LANG(?description) = 'en') .
  FILTER (?birth < "1800-01-01"^^xsd:date) .
}
ORDER BY ?name
Schools within 10km of Atlanta:
SELECT DISTINCT ?Link ?SchoolName ?EstablishedDate ?lat ?long
WHERE
{
  FILTER ( bif:st_intersects ( ?matchgeo, ?sourcegeo, 5 ) ) .
  ?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?