Problem Solving using Semantic Search with Unstructured Data

Joe Presbrey
CTO, Qwobl
presbrey@qwobl.com
<http://presbrey.mit.edu/foaf#presbrey>
Example Problem

*Find events in Boston related to my interests and which of my friends may also be interested*

- Find all the events in Boston for each of my interests
  - Ticketmaster, TicketNetwork, StubHub, etc.
- Find friends with interests that correspond with my own
  - Facebook, copy-paste, Excel?
Example Problem

*Find events in Boston related to my interests and which of my friends may also be interested*

- Find all the events in Boston for each of my interests
  - Ticketmaster, TicketNetwork, StubHub, etc.
- Find friends with interests that correspond with my own
  - Facebook, copy-paste, Excel?

Linked Data and Semantic Search provide a better solution!
Semantic Search

Find events in Boston related to my interests

• Linked Data
  – Select relevant data-sets for the search
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix ex: <http://example.com/schema#>.

<http://presbrey.mit.edu/foaf#presbrey> a foaf:Person ;
   rdfs:label "Joe Presbrey" ;
   foaf:interest
      <http://rdf.freebase.com/ns/guid.9202a8c04000641f8000000000c914c7> ,
      <http://rdf.freebase.com/ns/guid.9202a8c04000641f8000000003ad880f> ;
   foaf:knows <http://www.w3.org/People/Berners-Lee/card#i> .

<http://rdf.freebase.com/ns/guid.9202a8c04000641f8000000000c914c7> rdfs:label “Taylor Swift”.
<http://rdf.freebase.com/ns/guid.9202a8c04000641f8000000003ad880f> rdfs:label “Jay-Z”.

<http://example.com/event/1> a ex:Event ;
   ex:artist <http://rdf.freebase.com/ns/guid.9202a8c04000641f8000000000c914c7> ;
<http://example.com/event/2> a ex:Event ;
   ex:artist <http://rdf.freebase.com/ns/guid.9202a8c04000641f8000000003ad880f> ;

<http://geonames.example.com/Boston> rdfs:label “Boston” ;
Semantic Search

Find events in Boston related to my interests

• Linked Data
  – Select relevant data-sets for the search
    • FOAF, Interests: Freebase+Wikipedia, Events, Geonames

• Query Language+Engine
  – Ontological knowledge of how resources relate
  – Data-specific representation of search constraints
Find events in Boston related to my interests

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX ex: <http://example.com/schema#>
SELECT ?event WHERE {
  <http://presbrey.mit.edu/foaf#presbrey> foaf:interest ?i .
  ?event a ex:Event ;
      ?p ?i ;
      ex:city ?location ;
  ?location rdfs:label "Boston" .
}
Find events in Boston related to my interests

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX ex: <http://example.com/schema#>

SELECT ?event WHERE {
  <http://presbrey.mit.edu/foaf#presbrey> foaf:interest ?i .
  ?event a ex:Event ;
    ?p ?i ;
    ex:city ?location ;
    ?location rdfs:label “Boston” .
}

$ roqet -r simple -D example0.ttl example0.rq
roquet: Querying from file example0.rq
roquet: Query has a variable bindings result
result: [event=uri<http://example.com/event/1>]
result: [event=uri<http://example.com/event/2>]
roquet: Query returned 2 results
Live Demo

PREFIX qs: <http://rdf.qwobl.com/schema#>
PREFIX es: <http://events.rdf.qwobl.com/schema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT * WHERE {
  ?t0 rdfs:label "presbrey" .
  { { ?t0 ?p0_1 ?t1 } UNION { ?t1 ?p0_1 ?t0 } ?t1 qs:class ?t1c . ?t1c rdfs:label "interest" }
  UNION { ?t0 ?p0_1 ?t1 . ?p0_1 rdfs:label "interest" }
  UNION { ?t1 ?p1_2 ?t2 . ?p1_2 rdfs:label "concert" }
}
Qwobl resolves keywords to URIs, infers intended relations, and executes SPARQL.
Simple Semantics

presbrey interests album

Joe Presbrey

Jay-Z

Nas

Infinity On High

Big L
Simple Semantics

No RDF, ontology, or SPARQL experience required
Linked Data
presbrey's interests' concerts in Boston

"interests":
FOAF profiles

"in Boston":
Geonames

Lexical inference:
Wordnet, Yago
Qwobl Linked Data

presbrey's interests' concerts in Boston

Qwobl Crawler links:

- Last.fm
- Facebook

“interests”:
- TicketMaster
- StubHub
- more...

“concerts”:
Qwobl SPARQL Engine

• Auto-follow owl:sameAs

Equating equivalent entities across data-sets provides for property corroboration and strengthened resource identities but can be significant overhead for the language and/or engine

```
SELECT ?event WHERE {
  <http://presbrey.mit.edu/foaf#presbrey> foaf:interest ?i .
```
• Auto-follow owl:sameAs

Equating equivalent entities across data-sets provides for property corroboration and strengthened resource identities but can be significant overhead for the language and/or engine

SELECT ?event WHERE {
  <http://presbrey.mit.edu/foaf#presbrey> foaf:interest ?i .

SELECT DISTINCT ?event WHERE {
  { <http://presbrey.mit.edu/foaf#presbrey> foaf:interest ?i }
  UNION
  { <http://presbrey.mit.edu/foaf#presbrey> foaf:interest ?i0 .
    ?i0 owl:sameAs ?i }
  UNION
  { <http://presbrey.mit.edu/foaf#presbrey> owl:sameAs ?presbrey .
    ?presbrey foaf:interest ?i }
  UNION
  { <http://presbrey.mit.edu/foaf#presbrey> owl:sameAs ?presbrey .
    ?presbrey foaf:interest ?i0 . ?i0 owl:sameAs ?i } ... }
Qwobl SPARQL Engine

• High-performance full-text rdfs:label's

Replace any rdfs:label variable FILTER(regex) constraints with calls to a full-text index and use matching URIs to prepopulate SPARQL variable bindings

```sparql
SELECT * WHERE {
  ?t0 rdfs:label ?t0l FILTER regex(?t0l, "presbrey")
  ?t0 ?p0_1 ?t1 . ?p0_1 rdfs:label ?p0_1l
    FILTER regex(?p0_1l, "interests")
  { { ?t1 ?p1_2 ?t2 } UNION { ?t2 ?p1_2 ?t1 }
    ?t2 qs:class ?t2c . ?t2c rdfs:label ?t2cl
      FILTER regex(?t2cl, "concerts") }
  UNION { ?t1 ?p1_2 ?t2 . ?p1_2 rdfs:label ?p1_2l
    FILTER regex(?p1_2l, "concerts") }
            FILTER regex(?cityl, "boston") }
            FILTER regex(?statel, "boston") }
            FILTER regex(?zipl, "boston") }
}
```
Qwobl SPARQL Engine

- High-performance full-text rdfs:label's
  Replace any rdfs:label variable FILTER(regex) constraints with calls to a full-text index and use matching URIs to prepopulate SPARQL variable bindings

```
SELECT * WHERE {
  ?t0 rdfs:label "presbrey" .
  ?t0 ?p0_1 ?t1 . ?p0_1 rdfs:label "interests" .
  { { ?t1 ?p1_2 ?t2 } UNION { ?t2 ?p1_2 ?t1 } }
  ?t2 qs:class ?t2c . ?t2c rdfs:label "concerts" }
UNION { ?t1 ?p1_2 ?t2 . ?p1_2 rdfs:label "concerts" }
{ ?t2 es:city ?city . ?city rdfs:label "boston" }
UNION { ?t2 es:state ?state . ?state rdfs:label "boston" }
UNION { ?t2 es:zip ?zip . ?zip rdfs:label "boston" }
}