SPARQL Query Re-Writing for Spatial Datasets Using Partonomy Based Transformation Rules

Prateek Jain  
*Wright State University - Main Campus*

Cory Andrew Henson  
*Wright State University - Main Campus*

Amit P. Sheth  
*Wright State University - Main Campus, amit.sheth@wright.edu*

Peter Z. Yeh

Kunal Verma  
*Wright State University - Main Campus*

Follow this and additional works at: [http://corescholar.libraries.wright.edu/knoesis](http://corescholar.libraries.wright.edu/knoesis)

Part of the Bioinformatics Commons, Communication Technology and New Media Commons, Databases and Information Systems Commons, OS and Networks Commons, and the Science and Technology Studies Commons

Repository Citation

[http://corescholar.libraries.wright.edu/knoesis/667](http://corescholar.libraries.wright.edu/knoesis/667)

This Presentation is brought to you for free and open access by the The Ohio Center of Excellence in Knowledge-Enabled Computing (Kno.e.sis) at CORE Scholar. It has been accepted for inclusion in Kno.e.sis Publications by an authorized administrator of CORE Scholar. For more information, please contact corescholar@www.libraries.wright.edu.
SPARQL Query Re-writing for Spatial Datasets Using Partonomy Based Transformation Rules

Prateek Jain, Cory Henson, Amit Sheth
Kno.e.sis,
Wright State University,
Dayton, OH

Peter Z. Yeh, Kunal Verma
Accenture Tech. Labs
San Jose, CA
Increased availability of spatial information
But accessing this information can be difficult
User expected to ask for this information in the “right” way
Proposed approach

Automatically align conceptual mismatches between a user’s query and spatial information of interest through a set of semantic operators. Our approach will reduce the user’s burden of having to know how information of interest is structured, and hence improve accuracy and relevance of the results.
Outline

• Introduction
• Existing Mechanisms for querying RDF Data
  – Existing approaches
  – How well do they work?
• Proposed Approach
• Future Work
Why is it important?

• Spatial data becoming more significant day by day.

• Crucial for multitude of applications:
  – GPS
  – Military
  – Location Aware Services
  – weather data...

• Spatial Data availability on Web continuously increasing.
  – Sensor streams, satellite imagery
  – Naïve users contribute and correct spatial data too which can lead to discrepancies in data representation.
    • E.g. Geonames, Wikimapia
What’s the problem

- Existing approaches only analyze spatial information and queries at the lexical and syntactic level.

- Mismatches are common between how a query is expressed and how information of interest is represented.
  - Question: “Find schools in NJ”.
  - Answer: Sorry, no answers found!
  - Reason: Only counties are in states.

- Natural language introduces much ambiguity for semantic relationships between entities in a query.
  - Find Schools in Greene County.
What needs to be done?

• We need to reduce users’ burden of having to know how information of interest is represented and structured in order to enable access to this information by a broad population.

• We need to resolve mismatches between a query and information of interest due to differences in granularity in order to improve recall of relevant information.

• We need to resolve ambiguous relationships between entities due to natural language in order to reduce the amount of wrong information retrieved.
Existing mechanism for querying RDF
Known approaches

- SPARQL
- Regular Expression Based Querying Approaches
Common query for testing all approaches!

“Find schools located in the state of Ohio”
In a perfect scenario

School → parent feature → Ohio
In a not so perfect scenario
And finally..
Proposed Approach
Proposed Approach

- Define operators to ease writing of expressive queries by implicit usage of semantic relations between query terms and hence remove the burden of expressing named relations in a query.

- Define transformation rules for operators based on work by Winston’s taxonomy of part-whole relations.

- Rule based approach allows applicability in different domains with appropriate modifications.

- Partonomical Relationship Based Query Rewriting System (PARQ) implements this approach.
Architecture

User submits SPARQL Query

Mapping of ontology properties to Winston’s categories
• Triple Constraints
• Query Variables

Meta rules for Winston’s Categories
= Transformation Rules
• Altered Triple Constraints
• Altered Query Variables

Rewritten Query according to the data structure

SELECT ?school
WHERE { ?school geo:parentFeature Ohio. }

SELECT ?school
WHERE { ?state geo:name "Ohio"
  ?county geo:parentFeature ?schools .}
Meta Rules for Winston’s Categories

- **Transitivity**
  - \((a \text{ φ-part of } b) (b \text{ φ-part of } c) \Rightarrow (a \text{ φ-part of } c)\)
  - \((\text{Dayton place-part of } \text{Ohio}) (\text{Ohio place-part of } \text{US}) \Rightarrow (\text{Dayton place-part of } \text{US})\)

- **Overlap**
  - \((a \text{ place-part of } b) (a \text{ place-part of } b) \Rightarrow (b \text{ overlaps } c)\)
  - \((\text{Sri L. place-part of } \text{Indian Ocean}) (\text{Sri L. place-part of } \text{Bay of Bengal}) \Rightarrow (\text{Indian Ocean overlaps with } \text{Bay of Bengal})\)

- **Spatial Inclusion**
  - \((a \text{ instance of } b) (c \text{ spatially included in } a) \Rightarrow (c \text{ spatially included in } b)\)
  - \((\text{White House instance of } \text{Building}) (\text{Barack is in } \text{White House}) \Rightarrow (\text{Barack is in building})\)
Perfect Scenario

```sparql
SELECT ?school
WHERE {
    ?state geo:featureClass geo:A
    ?schools geo:featureClass geo:S.
    ?state geo:name "Ohio"
}
```

Query Re-Writer

```sparql
SELECT ?school
WHERE {
    ?state geo:featureClass geo:A
    ?schools geo:featureClass geo:S.
    ?state geo:name "Ohio"
    ?state geo:parentFeature ?schools
}
```
Slight and Severe Mismatch

SELECT ?school
WHERE {
  ?state geo:featureClass geo:A
  ?schools geo:featureClass geo:S.
  ?state geo:name "Ohio".
}

Query Re-Writer

SELECT ?school
WHERE {
  ?state geo:featureClass geo:A
  ?schools geo:featureClass geo:S.
  ?state geo:name "Ohio".
}
So where do we stand with all these mechanisms..

<table>
<thead>
<tr>
<th></th>
<th>Ease of writing</th>
<th>Expressivity</th>
<th>Works in all scenarios</th>
<th>Schema agnostic</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPARQL</td>
<td>X</td>
<td>✓</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PSPARQL</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Our Approach</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Evaluation
Evaluation

- Evaluation performed on publicly available datasets such as Geonames and British Ordnance Survey Ontology.

- Utilized 120 questions from National Geographic Bee and 46 questions from trivia related to British Administrative Geography.

- Questions serialized into SPARQL Queries by 4 human respondents unfamiliar with the ontology.

- Performance of PARQ compared with PSPARQL and SPARQL.
Sample Queries

• “In which English county, also known as "The Jurassic Coast" because of the many fossils to be found there, will you find the village of Beer Hackett?”

• “The Gobi Desert is the main physical feature in the southern half of a country also known as the homeland of Genghis Khan. Name this country.”
## PARQ Vs SPARQL

<table>
<thead>
<tr>
<th>Respondent</th>
<th>System</th>
<th># of queries answered</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PARQ</td>
<td>82</td>
<td>100%</td>
<td>68.3%</td>
</tr>
<tr>
<td></td>
<td>SPARQL</td>
<td>25</td>
<td>100%</td>
<td>20.83%</td>
</tr>
<tr>
<td>2</td>
<td>PARQ</td>
<td>93</td>
<td>100%</td>
<td>77.5%</td>
</tr>
<tr>
<td></td>
<td>SPARQL</td>
<td>26</td>
<td>100%</td>
<td>21.6%</td>
</tr>
<tr>
<td>3</td>
<td>PARQ</td>
<td>61</td>
<td>100%</td>
<td>50.83%</td>
</tr>
<tr>
<td></td>
<td>SPARQL</td>
<td>19</td>
<td>100%</td>
<td>15.83%</td>
</tr>
<tr>
<td>4</td>
<td>PARQ</td>
<td>103</td>
<td>100%</td>
<td>85.83%</td>
</tr>
<tr>
<td></td>
<td>SPARQL</td>
<td>33</td>
<td>100%</td>
<td>27.5%</td>
</tr>
</tbody>
</table>
## PARQ Vs PSPARQL

<table>
<thead>
<tr>
<th>System</th>
<th>Precision</th>
<th>Recall</th>
<th>Execution time/query in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARQ</td>
<td>100%</td>
<td>86.7%</td>
<td>0.3976</td>
</tr>
<tr>
<td>PSPARQL</td>
<td>6.414%</td>
<td>86.7%</td>
<td>37.59</td>
</tr>
</tbody>
</table>

Comparison for National Geographic Bee over Geonames

<table>
<thead>
<tr>
<th>System</th>
<th>Precision</th>
<th>Recall</th>
<th>Execution time/query in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARQ</td>
<td>100%</td>
<td>89.13%</td>
<td>0.099</td>
</tr>
<tr>
<td>PSPARQL</td>
<td>65.079%</td>
<td>89.13%</td>
<td>2.79</td>
</tr>
</tbody>
</table>

Comparison for British Admin. Trivia over Ordnance Survey Dataset
Conclusion

• Query engines expect user to know the structure of ontology and pose well formed queries.

• Query engines ignore semantic relations between query terms.

• Need to exploit semantic relations between concepts for processing queries.

• Need to provide systems to perform behind the scene rewrite of queries to remove burden of knowing structure of data from the user.
Future Work

- Investigating support for more SPARQL constructs such as FILTER, OPTIONAL pattern

- Testing our approach for its applicability across domains.

- Systematic comparison between resolving mismatches using query re-writing method viz-a-viz a reasoner.

- Development of additional transformation operators which cannot be defined in terms of Winston’s categorization such as involving “containment”.
References

- SPARQL [http://www.w3.org/TR/rdf-sparql-query/](http://www.w3.org/TR/rdf-sparql-query/)


- Prateek Jain, Peter Z. Yeh, Kunal Verma, Cory Henson and Amit Sheth, SPARQL Query Re-writing for Spatial Datasets Using Partonomy Based Transformation Rules, Third International Conference on Geospatial Semantics (GeoS 2009), Mexico City, Mexico, December 3-4, 2009.

Thank You!
Geonames Dataset

- Description at http://www.geonames.org/ontology/

- 100395794 (100 Million) RDF triples present in the dataset.

- Most interesting properties “parentFeature” (Administrative Region which contains the entity) and “nearbyFeature” (Entities close to this region).