GRIN: A Graph Based RDF Index

Octavian Udrea
Andrea Pugliese
V. S. Subrahmanian

Presented by Yuchen(Alex) GAO
Introduction

• Problem: processing graph-based queries
• Data structure: A Graph based RDF INDEX
• Query answering: GrinAnswer algorithm
• Experiments: on TAP and ChefMoz
Outline

• RDF graph queries
• The GRIN index
• Query evaluation
• Experimental evaluation
Define: P-path

P={location, locatedIn}

(ColdStone, location, Lincoln)
(Lincoln, locatedIn, NE/USA)
RDF graph query

SELECT ?v1 ?v2 ?v3
WHERE {{(?v1 attire ?v3).(?v1 cuisine Italian)}
{(?v2 attire ?v3).(?v2 cuisine Italian).
 (?v2 location Norfolk)}
{(Norfolk locatedIn NE/USA)}}

Extension of SPARQL: path-at-most
Outline

• RDF graph queries
• The GRIN index
• Query evaluation
• Experimental evaluation
GRIN index example

ROOT

<table>
<thead>
<tr>
<th>Center</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Italian</td>
<td>3</td>
</tr>
<tr>
<td>DairyQueen</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Center</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grivanti</td>
<td>3</td>
</tr>
<tr>
<td>Charlie's</td>
<td>2</td>
</tr>
<tr>
<td>Norfolk</td>
<td>2</td>
</tr>
<tr>
<td>FastFood</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(Review #16472, date, 12/15/06)</th>
<th>(Review #16472, rating, 8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Grivanti, cuisine, Italian)</td>
<td>(Charlie's, location, Norfolk)</td>
</tr>
<tr>
<td>(Grivanti, attire, businessCasual)</td>
<td>(Lincoln, locatedIn, NE/USA)</td>
</tr>
<tr>
<td>(DairyQueen, cuisine, FastFood)</td>
<td>(Arby's, cuisine, FastFood)</td>
</tr>
<tr>
<td>(Fazoli, attire, casual)</td>
<td>(Coldstone, cuisine, Dessert)</td>
</tr>
</tbody>
</table>
Building the index: Clustering

• # of clusters $C$ (leaf nodes):

\[
\frac{|R|}{C} \leq M \Rightarrow C = 2^{\log_2 \left( \frac{|R|}{M} \right)}
\]

$R$ is the set of resources

$M$ is the maximum number of RDF graph vertices per page

• Use a modification of the clustering algorithm by Kaufman & Rousseeuw 1987
Building the index: Clustering
Building the index: Intermediate

Choose a inter-cluster distance function $d_c$:

$$d_c(S, S') = \min_{x \in S, y \in S'} (d(x, y))$$

$$d_c(S, S') = \max_{x \in S, y \in S'} (d(x, y))$$

$$d_c(S, S') = \frac{\sum_{x \in S, y \in S'} (d(x, y))}{|S| \times |S'|}$$
Building the index: Intermediate

**ROOT**

- **Center:** Italian
  - Distance: 3
  - (Review #16472, date: 12/15/06)
  - (Review #16472, rating: 8)
  - (Grivanti, cuisine, Italian)
    - (Grivanti, attire, business Casual)

- **Center:** Charlie’s
  - Distance: 2
  - (Charlie’s, location, Norfolk)
  - (Review #21765, date: 11/08/03)
    - (Review #21765, rating: 6)

- **Center:** Norfolk
  - Distance: 2
  - (Lincoln, locatedIn, NE/USA)

- **Center:** Dairy Queen
  - Distance: 3
  - (Dairy Queen, cuisine, FastFood)
  - (Arby’s, cuisine, Fast Food)
  - Coldstone, cuisine, Dessert

- **Center:** Fast Food
  - Distance: 2
Outline

• RDF graph queries
• The GRIN index
• Query evaluation
• Experimental evaluation
Query Evaluation: Constraints

\[ d(\text{v1}, \text{NE/USA}) \leq 2 \]
\[ d(\text{v2}, \text{NE/USA}) \leq 2 \]
\[ d(\text{v3}, \text{Italian}) \leq 2 \]

...
Query Evaluation: Algorithm

For every node, ACCEPT it or REJECT it!
Rule 1: Reject $\text{circle}(c,r)$ if any constant in query is outside the circle
Rule 2: Reject $\text{circle}(c,r)$ if we cannot guarantee that every variable in inside the circle
Is ?v2 in circle (Grivanti, 2)?
\[ d(?v2, Norfolk) \leq 1 \] but
\[ d(\text{Grivanti, Norfolk}) = 3 > 2 \]

So ?v2 cannot be satisfied by circle (Grivanti, 2)
Is \(?v_1\) in circle (Grivanti, 2)?
\[ d(\text{Grivanti,}\ ?v_1) \leq d(\text{Grivanti, Italian}) + d(\?v_1, \text{Italian}) \leq 2 \]

So \(?v_1\) can be satisfied
Query Evaluation: Algorithm

- GRINAnswer(D,G,q,n_i)
- Start from root
- If either child of the node $n_i$ passes rule 1 and rule 2, go down. If not, run subgraph matching on $n_i$ itself
Outline

• RDF graph queries
• The GRIN index
• Query evaluation

• Experimental evaluation
Experimental Evaluation

• Compare with Jena, Sesame and RDFBroker
  i) time to build index
  ii) index size
  iii) time to answer queries

• Dataset
  TAP (up to 300MB)
  ChefMoz (about 220MB)
Experimental Evaluation
Experimental Evaluation

![Graph showing query time for different numbers of variables for GRIN, RDFBroker, Sesame, and Jena. The x-axis represents the number of variables, ranging from 1 to 25, and the y-axis represents time in milliseconds, ranging from 0 to 25,000. The graph demonstrates how query time increases with the number of variables for each system.]
Comparison to related work

• New format for indexing triples

• Grouping information around “Center”

• Avoid time-consuming join operation
Thank you! =)

Any Questions?
Discussion: Question

• It is mentioned in the paper: We find experimentally that the determining factor in choosing GRIN over one of the other systems depends on the average degree of a node in the query graph.

• WHY?
Discussion: Limitation

- Good efficiency only for small-sized data

  worst case complexity when building index:
  \(O(|R|^4 \cdot \log_2(|R|))\)

  Index stored in memory
Discussion: Limitation

• Scalability is an issue
  Data is up to 300MB, up to 3.5 million triples

  Algorithm main-memory based

  Hard to adapt for a distributed context
Discussion: Limitation

• GRIN focus on path-like queries on RDF data, which cannot be expressed using current SPARQL language
Discussion: Future work

• Disk-based index

• Implementation

• Query optimization