Towards a Query-by-Example System for Knowledge Graphs

Nandish Jayaram (UT-Arlington)
Arijit Khan (UCSB, now at ETH Zurich)
Chengkai Li (UT-Arlington)
Xifeng Yan (UCSB)
Ramez Elmasri (UT-Arlington)

GRADES 2014
June 22, Snowbird, UT
Knowledge Graphs

Large and complex graphs capturing millions of entities and relationships between them!

Ubiquitous today:
- Linking Open Data: 52 billion RDF triples
- Freebase: 1.8 billion facts
- DBpedia: 470 million facts
- YAGO: 120 million facts
How to Query Knowledge Graphs?

Graph Search / Structured Querying

- Graph Search / Structured Querying
- Expertise in constructing structured queries required.
- A good knowledge of the schema of the knowledge graph is required.

```
SELECT F.obj, F.src
FROM F, G, H, L, P
WHERE F.prop = 'founded'
AND G.prop = 'education'
AND H.prop = 'headquartered_in'
AND L.prop = 'places_lived'
AND P.prop = 'place_founded'
AND F.obj = H.src
AND F.obj = P.src
AND F.src = L.src
AND L.obj = H.obj
AND F.src = G.src
```
Improving Usability of Knowledge Graphs: Prior Arts

- **Keyword Search**
  
  "Software companies located in the Silicon Valley and their founders who studied at Stanford University.”
  
  - Keyword search on graphs [Karger11].
  - Keyword based query formulation [Pound10] [Yao12].

- **Natural Language Query**
  
  - Natural language questions based querying [Yahya12].

- **Visual Query Interfaces**
  
  - Interactive and form based query construction [Demidova12] [Jarrar12].
  - Visual interface for query graph construction [Chau08] [Jin10].

- **Schemaless Graph Querying**
  
  - Use transformations to find matches to a naïve query graph [Yang14].
Query by Example Entity Tuples

Given an input \( n \)-entity tuple(s) (called \( n \)-tuple), a knowledge graph, and \( k \), find top-\( k \) \( n \)-tuples that are most similar to the input tuple(s).
### Answer Tuples

**Input Tuple**: Donald Knuth, Stanford University, Turing Award

<table>
<thead>
<tr>
<th>Stanford University</th>
<th>IEEE John von Neumann Medal</th>
<th>Donald Knuth</th>
<th>View Answer Graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stanford University</td>
<td>Turing Award</td>
<td>John McCarthy</td>
<td>View Answer Graph</td>
</tr>
<tr>
<td>Stanford University</td>
<td>Turing Award</td>
<td>Niklaus Wirth</td>
<td>View Answer Graph</td>
</tr>
<tr>
<td>Stanford University</td>
<td>National Medal of Science for Mathematics and Computer Science</td>
<td>Donald Knuth</td>
<td>View Answer Graph</td>
</tr>
<tr>
<td>Stanford University</td>
<td>Kyoto Prize</td>
<td>Donald Knuth</td>
<td>View Answer Graph</td>
</tr>
<tr>
<td>Stanford University</td>
<td>Turing Award</td>
<td>Robin Milner</td>
<td>View Answer Graph</td>
</tr>
</tbody>
</table>
Discover an hidden query graph behind the input tuples.

Query lattice to model space of all approximate matches.

Find approximate matching answers to the MQG.

Obtain user feedback to better understand the query intent.

- Exemplar Queries [Mottin14]
Maximal Query Graph Discovery

- Given an example tuple like \(<\text{Jerry Yang}, \text{Yahoo!}>\)

- Define importance of edges by assigning weights to them.

- Find a small sub-graph with important edges and nodes in the neighborhood of \(\text{Jerry Yang}\) and \(\text{Yahoo!}\), to form the Maximal Query Graph (MQG).
Every other node is a sub-graph of the MQG.
Query Processing

Upper bound based bottom-up lattice exploration.

Lattice evaluation terminated after top-k answers are obtained!

Pruning nodes based on null nodes.
Finding Matching Answer Graphs

- **Exact sub-graph matching**, based on indexing techniques.
  - Search on graph databases [Shasha02] [Yan04] [Zhao07] [Zou08].
  - Search on single large graph [Ullman76] [Cordella04] [Shang08] [Zhang09].

- **Approximate sub-graph matching**.
  - Use various indexes to quickly find approximate matches [Tian08] [Mongiovi10] [Khan13].
  - NESS : uses neighborhood-based indexes to quickly find approximate matches to a query graph [Khan11].
Experiments

QUERIES:
- 20 Queries on Freebase dataset (47 M edges, 27 M nodes, 5.4 K properties)
- 8 Queries on DBpedia dataset (2.6 M edges, 759 K nodes, 9 K properties)

Accuracy Comparison with NESS:
Efficiency Results

Single Query Execution Times (in seconds)

Query Processing Time (secs.)

# edges in MQG

Query

<table>
<thead>
<tr>
<th>Query</th>
<th>GQBE</th>
<th>NESS</th>
<th>Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>12</td>
<td>13</td>
<td>18</td>
</tr>
<tr>
<td>F2</td>
<td>10</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>F3</td>
<td>8</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>F4</td>
<td>8</td>
<td>8</td>
<td>12</td>
</tr>
<tr>
<td>F5</td>
<td>8</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>F6</td>
<td>8</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>F7</td>
<td>9</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>F8</td>
<td>9</td>
<td>7</td>
<td>11</td>
</tr>
<tr>
<td>F9</td>
<td>9</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>F10</td>
<td>7</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>F11</td>
<td>7</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>F12</td>
<td>7</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>F13</td>
<td>7</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>F14</td>
<td>7</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>F15</td>
<td>7</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>F16</td>
<td>7</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>F17</td>
<td>7</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>F18</td>
<td>7</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>F19</td>
<td>7</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>F20</td>
<td>7</td>
<td>10</td>
<td>11</td>
</tr>
</tbody>
</table>
Work in Progress

Maximal Query Graph Discovery:
- Does not capture the user-intent exactly.
- Iterative and interactive edge suggestion.

Query Processing:
- Materializing intermediate join results (millions of rows) can be expensive.
- Is a better join mechanism when we have more memory at our disposal possible?
- Distributed lattice exploration mechanism.

Obtaining User Feedback:
- User feedback on relevance of answer tuples to re-weight edges.
Work by Xifeng Yan’s group at UCSB

(SIGMOD 2014 demo, VLDB 2014)
Demo and Technical Details:

- **Demo:**
  - URL: idir.uta.edu/gqbe
  - Demo paper: *GQBE: Querying knowledge graphs by example entity tuples*, ICDE 2014.

- **Technical Details:**
  - Full paper under review