Empowering Investigative Journalism with Graph-based Heterogeneous Data Management

Angelos-Christos Anadiotis
Ecole Polytechnique and Institut Polytechnique de Paris
Conflicts of Interest database

“A conflict of interest is any situation where a public interest may interfere with a public or private interest, in such a way that the public interest may be, or appear to be, unduly influenced.”

French transparency law, 2011
Biomedical domain

• **Experts in the biomedical area** advise national and international officials on decisions with impact on public health

• **Companies with interests in this area** may recruit experts likely to be auditioned by regulatory boards

• **Goal**: establish a database of CoIs where it would be easy to "find the declared links of Dr. Alice with HealthStar"
Biomedical domain

• **Experts in the biomedical area** advise national and international officials on decisions with impact on public health

• **Companies with interests in this area** may recruit experts likely to be auditioned by regulatory boards

• **Goal**: establish a database of CoIs where it would be easy to "find the declared links of Dr. Alice with HealthStar"

Usually available, but *technically buried* information
Landscape of heterogeneous data
ConnectionLens graph processing pipeline

ConnectionLens graph construction
Extraction policies

Nodes+edges

Relational DB

P-GAM Parallel Query Engine

Optimized Graph Layout

GAM KS algorithm
ConnectionLens graph processing pipeline

ConnectionLens graph construction
Extraction policies

Relational DB
Nodes+edges

P-GAM Parallel Query Engine
Optimized Graph Layout
In-memory migration
GAM KS algorithm

Querying the graph
Problem statement

• Given the graph $G = (N, E)$ built out of the datasets $D$ and a query keywords $Q = \{w_1, \ldots, w_m\}$, return the $k$ highest-score minimal answer trees

• An **answer tree** is a set of edges which (i) form a tree, and (ii) for each $w_i$, contain at least one node whose label matches $w_i$

• We are interested in **minimal answer trees**, that is:
  • Removing an edge from the tree should make it lack some query keywords $w_i$
  • If a query keyword $w_i$ matches the label of more than one nodes in the answer tree, then all these matching nodes must be equivalent
Problem statement

• Given the graph $G = (N, E)$ built out of the datasets $D$ and a query keywords $Q = \{w_1, ..., w_m\}$, return the $k$ highest-score minimal answer trees.

• An answer tree is a set of edges which (i) form a tree, and (ii) for each $w_i$, contain at least one node whose label matches $w_i$.

• We are interested in minimal answer trees, that is:
  • Removing an edge from the tree should make it lack some query keywords $w_i$.
  • If a query keyword $w_i$ matches the label of more than one nodes in the answer tree, then all these matching nodes must be equivalent.

Related to GSTP + bidirectional edges

Return $k$ highest-score trees among those found.
Grow and Aggressive Merge

Grow

$w_1$  $w_2$  ...  $w_m$

$N_{1,1}$  $N_{2,1}$  $N_{m,1}$
$N_{1,2}$  $N_{2,2}$  $N_{m,2}$
...  ...  ...
$N_{1,k_1}$  $N_{2,k_2}$  $N_{m,k_m}$
Grow and Aggressive Merge

\[ \begin{align*}
W_1, & \quad W_2, \\
N_{1,1}, & \quad N_{2,1}, \\
N_{1,2}, & \quad N_{2,2}, \\
\ldots, & \quad \ldots, \\
N_{1,k_1}, & \quad N_{2,k_2}, \\
N_{1,2}, & \quad N_{2,2}, \\
N_{1,k_1}, & \quad N_{2,k_2}, \\
N_{1,2}, & \quad N_{2,2}, \\
N_{1,k_1}, & \quad N_{2,k_2}, \\
N_{1,2}, & \quad N_{2,2}, \\
N_{1,k_1}, & \quad N_{2,k_2},
\end{align*} \]
Grow and Aggressive Merge

$N_{1,1}$ $N_{2,1}$ $N_{m,1}$

$N_{1,2}$ $N_{2,2}$ $N_{m,2}$

$\ldots$ $\ldots$ $\ldots$

$N_{1,k1}$ $N_{2,k2}$ $N_{m,km}$

Grow

$N_{1,2}$ $N_{2,1}$
Grow and Aggressive Merge

Grow

Merge

\[ N_{1,1} \quad N_{2,1} \quad \ldots \quad N_{m,1} \]

\[ N_{1,2} \quad N_{2,2} \quad \ldots \quad N_{m,2} \]

\[ N_{1,k_1} \quad N_{2,k_2} \quad \ldots \quad N_{m,k_m} \]
Grow and Aggressive Merge

Grow

\[ N_{1,1} \rightarrow N_{2,1} \rightarrow N_{m,1} \]

Merge

\[ N_{1,2} \rightarrow N_{2,1} \rightarrow N_{2,1} \rightarrow N_{3,1} \]

\[ N_{1,2} \rightarrow N_{2,1} \rightarrow N_{3,2} \]
Grow and Aggressive Merge

\[
\begin{align*}
W_1, W_2, \ldots, W_m \\
N_{1,1}, N_{1,2}, \ldots, N_{1,k1} \\
N_{2,1}, N_{2,2}, \ldots, N_{2,k2} \\
N_{m,1}, N_{m,2}, \ldots, N_{m,km}
\end{align*}
\]
Which tree to Grow or to Merge?

• Assign priorities to answer trees resulting from Grow/Merge
  1. Prefer trees matching many query keywords
  2. Prefer trees of smaller size
Which tree to Grow or to Merge?

- Assign priorities to answer trees resulting from Grow/Merge
  1. Prefer trees matching many query keywords
  2. Prefer trees of smaller size

Priority Queue
Which tree to Grow or to Merge?

• Assign priorities to answer trees resulting from Grow/Merge
  1. Prefer trees matching many query keywords
  2. Prefer trees of smaller size
Which tree to Grow or to Merge?

• Assign priorities to answer trees resulting from Grow/Merge
  1. Prefer trees matching many query keywords
  2. Prefer trees of smaller size

Priority Queue

Apply priority rules

Matches most keywords
Which tree to Grow or to Merge?

• Assign priorities to answer trees resulting from Grow/Merge
  1. Prefer trees matching many query keywords
  2. Prefer trees of smaller size

1. Grow answer tree
2. Merge with same-rooted answer trees
Which tree to Grow or to Merge?

• Assign priorities to answer trees resulting from Grow/Merge
  1. Prefer trees matching many query keywords
  2. Prefer trees of smaller size

Priority Queue

1. Grow answer tree
2. Merge with same-rooted answer trees

Mixed BFS/DFS approach of graph search
In-memory graph layout

### Keyword Index

<table>
<thead>
<tr>
<th>w₁</th>
<th>Node 367</th>
</tr>
</thead>
<tbody>
<tr>
<td>w₂</td>
<td>Node 212</td>
</tr>
<tr>
<td>w₃</td>
<td>Node 452</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>wₙ</td>
<td>Node 231</td>
</tr>
</tbody>
</table>

### Edge

- source node
- target node
- specificity
- metadata

### Edge metadata

- edge type
- edge label

### Node

- data source
- representative
- connection 1
- ... connection K
- metadata
- connections heap

### Node metadata

- node type
- node label

### Node connections

- connection K+1
- ...
In-memory graph layout

Row-major, CPU-cache-friendly data layout
Duplicate work elimination

• The same answer tree may be created following different combinations of Grow and Merge
  ➢ Duplicate work

• Maintain a history of explored trees

• Every answer tree is inserted only once:
  • in the history of explored trees
  • in the priority queue
Parallel search

• Cannot partition the graph:
  • expensive, and we do not know which parts we will need
  • no assumption on the shape of the graph

• DFS/BFS alternation incurs mixed scalability requirements

• P-GAM bottlenecks
  • size of intermediate results
Parallel search

• Cannot partition the graph:
  • expensive, and we do not know which parts we will need
  • no assumption on the shape of the graph
• DFS/BFS alternation incurs mixed scalability requirements
• P-GAM bottlenecks
  • size of intermediate results

Shared-everything
Concurrent data structures
Experimental evaluation – CoI application

• 450,000 PubMed bibliographic notices (2019, 2020)
• 42,000 PDF articles transformed to JSON
• 781 HTML pages describing relationships between people and organizations
• Load the graph in the main memory
• Query thresholds:
  • 1000 solutions
  • 1 minute of execution time
CoI application results (anonymized)

<table>
<thead>
<tr>
<th>#</th>
<th>Keywords</th>
<th>$T^1$</th>
<th>$T^{last}$</th>
<th>$T$</th>
<th>$S$</th>
<th>#DS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A1, A2</td>
<td>200</td>
<td>4840</td>
<td>4840</td>
<td>1000</td>
<td>1-6,5</td>
</tr>
<tr>
<td>2</td>
<td>A3, I1</td>
<td>1263</td>
<td>20547</td>
<td>60000</td>
<td>13</td>
<td>2-4,2,3</td>
</tr>
<tr>
<td>3</td>
<td>A5, A6, I3</td>
<td>2602</td>
<td>4203</td>
<td>60000</td>
<td>15</td>
<td>6,8,8</td>
</tr>
<tr>
<td>4</td>
<td>A8, I2, I4</td>
<td>667</td>
<td>51186</td>
<td>60000</td>
<td>63</td>
<td>4-7,6</td>
</tr>
<tr>
<td>5</td>
<td>A9, H3, I2</td>
<td>264</td>
<td>59831</td>
<td>60000</td>
<td>516</td>
<td>3-8,5</td>
</tr>
<tr>
<td>6</td>
<td>H2, I1, P1</td>
<td>1267</td>
<td>60212</td>
<td>60000</td>
<td>148</td>
<td>6-8,6</td>
</tr>
<tr>
<td>7</td>
<td>A5, A10, I2</td>
<td>19077</td>
<td>23160</td>
<td>60000</td>
<td>9</td>
<td>8,8</td>
</tr>
<tr>
<td>8</td>
<td>A9, I1, I4, I5</td>
<td>6327</td>
<td>55762</td>
<td>60000</td>
<td>38</td>
<td>8-9,11,8</td>
</tr>
<tr>
<td>9</td>
<td>A7, I1, I6, P1</td>
<td>1857</td>
<td>3057</td>
<td>60000</td>
<td>8</td>
<td>7,8,7,8</td>
</tr>
<tr>
<td>10</td>
<td>A7, A8, I1, I2, I4</td>
<td>3389</td>
<td>28237</td>
<td>60000</td>
<td>4</td>
<td>7-8,11,11</td>
</tr>
</tbody>
</table>
Conclusion

• ConnectionLens introduces an end-to-end pipeline for constructing and querying graphs from heterogeneous data.

• In-memory storage engine stores the graph data required for querying.

• P-GAM queries the graph in parallel.
Find out more about our work

• A. -C. Anadiotis, O. Balalau, C. Conceição, H. Galhardas, M. Y. Haddad, I. Manolescu, T. Merabti, J. You. Graph integration of structured, semistructured and unstructured data for data journalism. Information Systems (accepted for publication).

• A. -C. Anadiotis, O. Balalau, T. Bouganim, F. Chimienti, H. Galhardas, M. Y. Haddad, S. Horel, I. Manolescu, Y. Youssef. Empowering Investigative Journalism with Graph-based Heterogeneous Data Management. IEEE Data Engineering Bulletin (accepted for publication).


SourcesSay project
https://sourcessay.inria.fr