SQL QueRIE Recommendations: a query fragment-based approach

Jayad Akbarnejad, San José State University
Magdalini Eirinaki, San José State University
Suju Koshy, San José State University
Duc On, San José State University
Neoklis Polyzotis, University of California, Santa Cruz

Available at: https://works.bepress.com/magdalini_eirinaki/9/
SQL QueRIE Recommendations: A Query Fragment-based Approach

Javad Akbarnejad
Magdalini Eirinaki
Suju Koshy
Duc On
Neoklis Polyzotis
Motivation

- Scientific disciplines use relational DBMS for storage and retrieval of information
  - Biologists (e.g. UCSC Genome, BMRB)
  - Astronomers (e.g. Skyserver)
  - Chemists (e.g. PubChem)
- DBs are accessible online by users with diverse information needs
- Typical users do interactive exploration
Motivation (cont’d)

- Typical users are not SQL experts
- Scientific datasets increase in size
- Users may miss interesting information
  - They do not write the “right” query
  - They are not aware of all parts of the database

**Our goal:** Assist users in finding useful information
Web Collaborative Filtering

**Example:** Movie Recommendations

If Alice and Bob both like movie X and Alice likes movie Y then Bob is likely to be interested in seeing movie Y

If Alice and Bob both query data X and Alice queries data Y then Bob is likely to be interested in querying data Y
System Architecture

How do we generate meaningful queries?

How do we define the similarity metric between users?

Which parts of the database are interesting to the user?
Roadmap

- Introduction
- QueRIE Recommendation Framework
- Experiments
- QueRIE Prototype
- Conclusion
QueRIE Conceptual Framework

Current User
- Session Representation
  - S0
  - Similarity Function
  - Prediction
  - User / Item

Past Users
- Session Representation
  - S1 S2 ... Sn

Top-N List of Recommendations
- Recommendations Generator
  - Spred
  - Predicted Summary

Query Log
1. **Tuple-based recommendations [SSDBM09, ICDM09]**
   - Sessions represented by the tuples “touched” by respective queries
   - User-based similarity: 2 users are similar if they explore the same parts of the DB
   - Predict which parts of DB will interest the user and recommend queries that “touch” them

2. **Query fragment-based recommendations**
Session Representation

Relations: \( R(a, b, c) \)
\( S(d, e, f) \)

**Q₁**: SELECT \( R.a, R.b \) FROM \( R \) WHERE \( R.b = 2 \)

**Q₂**: SELECT \( R.a, R.b, S.e \) FROM \( R, S \) WHERE \( R.a = S.f \) AND \( R.b < 3 \)

Query parsing & relaxation

**Q₁**: SELECT \( R.a, R.b \) FROM \( R \) WHERE \( R.b \) EQU NUM

**Q₂**: SELECT \( R.a, R.b, S.e \) FROM \( R, S \) WHERE \( R.a \) EQU \( S.f \) AND \( R.b \) COMPARE NUM
Session Representation (cont’d)

**Binary Scheme**

- $Q_1 = \langle 1, 0, \ldots, 1, 1, 0, \ldots, 1, 0, 0 \rangle$
- $Q_2 = \langle 1, 1, \ldots, 1, 1, 1, \ldots, 0, 1, 1 \rangle$
- $S_0 = \langle 1, 1, \ldots, 1, 1, 1 \ldots, 1, 1, 1 \rangle$

**Weighted Scheme**

- $Q_1 = \langle 1, 0, \ldots, 1, 1, 0, \ldots, 1, 0, 0 \rangle$
- $Q_2 = \langle 1, 1, \ldots, 1, 1, 1, \ldots, 0, 1, 1 \rangle$
- $S_0 = \langle 2, 1, \ldots, 2, 2, 1 \ldots, 1, 1, 1 \rangle$

**QF** = \{R, S, ..., R.a, R.b, S.e, ..., R.b EQU NUM, R.b COMPARE NUM, R.a EQU S.f \}
Session Similarity

- Based on the item-based approach
  - Construct *fragment x fragment* similarity matrix offline
  - More efficient than the user-based approach
- Vector-space similarity functions can be used
- High similarity means that the query fragments co-appear frequently in sessions

=> the active user might also like to use them
Prediction

- For each fragment $\phi$, select top-k similar fragments $\rho \in R$
- Then compute “predicted summary”:

$$S_0^{pred}[\phi] = \frac{\sum_{\rho \in R} S_0[\rho] \cdot \text{sim}(\rho, \phi)}{\sum_{\rho \in R} \text{sim}(\rho, \phi)}$$
Prediction – the $\alpha$ factor

$S_0^{pred}$ contains:

- Only other users’ fragments
- Both $S_0$ and other users’ fragments
- Only $S_0$ fragments
Recommendations Generator

💡 Use queries of past users

Query Log Data

\[ q_1 = <1,0,0,...,0> \]
\[ q_2 = <0,1,0,...,0> \]
\[ \vdots \]
\[ q_N = <1,0,1,...,1> \]

\[ S^{\text{pred}} = <1,0,0,...,0> \]

Top-n fragments

Similarity Function

\[ (u^{\text{pred}}, q_i) \]

\[ \text{rank}(q_i) = \text{sim}(u^{\text{pred}}, q_i) \]

Return Top-\(m\) Queries

\[ \text{rank}(q_1) = \text{sim}(u^{\text{pred}}, q_1) \]
\[ \text{rank}(q_2) = \text{sim}(u^{\text{pred}}, q_2) \]
\[ \vdots \]
\[ \text{rank}(q_N) = \text{sim}(u^{\text{pred}}, q_N) \]
Roadmap

- Introduction
- QueRIE Recommendation Framework
- Experiments
- QueRIE Prototype
- Conclusions
Experimental Setup

- **SkyServer Dataset**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Sessions</td>
<td>180</td>
</tr>
<tr>
<td>#Distinct Queries</td>
<td>1400</td>
</tr>
<tr>
<td>#Distinct query fragments</td>
<td>755</td>
</tr>
<tr>
<td>#Non-zero pair-wise fragment similarities</td>
<td>30436</td>
</tr>
<tr>
<td>Avg. number of queries per session</td>
<td>9.3</td>
</tr>
<tr>
<td>Min. number of queries per session</td>
<td>3</td>
</tr>
</tbody>
</table>

- **Validation method: Holdout Set**
- **Evaluation Metrics: Precision, Recall, F-Score**
Experimental evaluation – top-n

- Precision and recall drop for large $n$.
- More fragments with low similarity included in the mix
Experimental Evaluation - $\alpha$

- Including user’s current session fragments is beneficial
- Expansion/Restructuring of posted queries
Experimental Evaluation – Weighting Scheme

Weighted scheme slightly outperforms the binary
Roadmap

- Introduction
- QueRIE Recommendation Framework
- Experiments
- QueRIE Prototype
- Conclusions
QueRIE Prototype

Query Recommendations for Interactive Data Exploration

Query Results:

Please provide your Query here:

```
select top 1000 * from field where fieldid=0x082802802c0000
```

Query Results:

<table>
<thead>
<tr>
<th>fieldID</th>
<th>skyVersion</th>
<th>num</th>
<th>camcol</th>
<th>Hold</th>
<th>Objects</th>
<th>nChild</th>
<th>nGalaxy</th>
<th>nStars</th>
<th>numStars</th>
<th>nStars_e</th>
<th>numStars_r</th>
<th>nStars_z</th>
<th>corr_nCR</th>
<th>gnCR</th>
<th>nCR_r</th>
<th>nCR_z</th>
<th>nCR_nBrgHt</th>
</tr>
</thead>
<tbody>
<tr>
<td>587735131425734081</td>
<td>273840</td>
<td>4</td>
<td>44</td>
<td>1103</td>
<td>328</td>
<td>668</td>
<td>274</td>
<td>472</td>
<td>757</td>
<td>757</td>
<td>757</td>
<td>735</td>
<td>139</td>
<td>394</td>
<td>164</td>
<td>154</td>
<td>141</td>
</tr>
</tbody>
</table>

Recommended Queries:

```
select top 1000 * from field where fieldid=0x082802802c0000
```

```
select top 1000 * from field where fieldid=0x082802802c0000
```

```
select top 1000 * from field where fieldid=0x082802802c0000
```

```
select top 1000 * from field where fieldid=0x082802802c0000
```
QueRIE Prototype (cont’d)

Recommendation Details

Recommendations:

1. Current active session is 61856
2. Queries in active session: select top 1000 * from field where fieldid=0x08280ab2602c0000
3. Top predicted items: 7735/7736/7737/7739/7740
4. Top predicted item names: twinframe, 
   JCV[3,6,8] EQU HXNUM
   PHOTOOBJ, 
   JCV[17,20] EQU HXNUM
5. Recommendation queries are:
6. Recommendation Query 1 select top 1000 * from frame where fieldid=0x08280ab2602c0000
7. Session ID for above Query 45
8. Recommendation Query 2 select top 1000 * from photoobj where objid=0x08280ab2602c0111
9. Session ID for above Query 45
QueRIE Prototype

- Demo @ VLDB
  - Session: Data Extraction, Integration and Mining
  - Tue & Wed, 2 – 3:30 PM
  - Lyrebird room
Conclusions

- Non-expert users need help in exploring databases
- Query recommendations can be an effective tool in guiding exploration
- Collaborative filtering provides a natural method to generate recommendations
- Experiments show promising results on real-world datasets

Ongoing & Future Work:
  - Comparison of two recommendation engines
  - Extend for form-based queries
Thank you!

Questions