A Framework for Evaluating Database Keyword Search Strategies

Joel Coffman    Alfred C. Weaver

University of Virginia

28 October 2010
Outline

Introduction

Evaluation Framework
Outline

Introduction

Evaluation Framework

Experiments

Conclusion
Keyword Search

- Preferred means of data exploration and retrieval online
  - > 4 billion searches daily
- Desire to extend paradigm to relational databases
Keyword Search

- Preferred means of data exploration and retrieval online
  - > 4 billion searches daily
- Desire to extend paradigm to relational databases

Example
Who played Professor Henry Jones in *Indiana Jones and the Last Crusade*?
Keyword Search

- Preferred means of data exploration and retrieval online
  - > 4 billion searches daily
- Desire to extend paradigm to relational databases

Example
Who played Professor Henry Jones in *Indiana Jones and the Last Crusade*?

<table>
<thead>
<tr>
<th>Person</th>
<th>Character</th>
<th>Movie</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>name</td>
<td>id</td>
</tr>
<tr>
<td>10</td>
<td>Ford, Harrison</td>
<td>7</td>
</tr>
<tr>
<td>11</td>
<td>Connery, Sean</td>
<td>9</td>
</tr>
</tbody>
</table>
Keyword Search

- Preferred means of data exploration and retrieval online
  - > 4 billion searches daily
- Desire to extend paradigm to relational databases

Example

Who played Professor Henry Jones in *Indiana Jones and the Last Crusade*?

<table>
<thead>
<tr>
<th>Person</th>
<th>Character</th>
<th>Movie</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>name</td>
<td>id</td>
</tr>
<tr>
<td>10</td>
<td>Ford, Harrison</td>
<td>Raiders of the Lost Ark</td>
</tr>
<tr>
<td>11</td>
<td>Connery, Sean</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Indiana Jones and the Last Crusade</td>
</tr>
</tbody>
</table>

Definition (query result)

A tree of tuples that is reduced with respect to the query.
Keyword Search

- Preferred means of data exploration and retrieval online
  - > 4 billion searches daily
- Desire to extend paradigm to relational databases

Example
Who played Professor Henry Jones in *Indiana Jones and the Last Crusade*?

<table>
<thead>
<tr>
<th>Person</th>
<th>Character</th>
<th>Movie</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>name</td>
<td>id</td>
</tr>
<tr>
<td>10</td>
<td>Ford, Harrison</td>
<td>7</td>
</tr>
<tr>
<td>11</td>
<td>Connery, Sean</td>
<td>9</td>
</tr>
</tbody>
</table>

Definition (query result)
A tree of tuples that is reduced with respect to the query.

*Indiana Jones and the Last Crusade*

Sean Connery
Professor Henry Jones
Keyword Search

- Preferred means of data exploration and retrieval online
  - > 4 billion searches daily
- Desire to extend paradigm to relational databases

**Example**
Who played Professor Henry Jones in *Indiana Jones and the Last Crusade*?

<table>
<thead>
<tr>
<th>Person</th>
<th>Character</th>
<th>Movie</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>name</td>
<td>id</td>
</tr>
<tr>
<td>10</td>
<td>Ford, Harrison</td>
<td>7</td>
</tr>
<tr>
<td>11</td>
<td>Connery, Sean</td>
<td>9</td>
</tr>
<tr>
<td>18</td>
<td></td>
<td>19</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Which would you rather write?
Keyword Search

▶ Preferred means of data exploration and retrieval online
  ▶ > 4 billion searches daily
▶ Desire to extend paradigm to relational databases

Example
Who played Professor Henry Jones in Indiana Jones and the Last Crusade?

<table>
<thead>
<tr>
<th>Person</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>name</td>
</tr>
<tr>
<td>10</td>
<td>Ford, Harrison</td>
</tr>
<tr>
<td>11</td>
<td>Connery, Sean</td>
</tr>
<tr>
<td>id</td>
<td>name</td>
</tr>
<tr>
<td>7</td>
<td>Indiana Jones</td>
</tr>
<tr>
<td>9</td>
<td>Professor Henry Jones</td>
</tr>
</tbody>
</table>

Which would you rather write?

```sql
SELECT Person.name
FROM Person, Character, Movie, Cast
WHERE Person.id = Cast.personId
AND Character.id = Cast.characterId
AND Movie.id = Cast.movieId
AND Character.name = 'Professor Henry Jones'
AND Movie.title = 'Indiana Jones and the Last Crusade';
```
Keyword Search

- Preferred means of data exploration and retrieval online
  - > 4 billion searches daily
- Desire to extend paradigm to relational databases

Example
Who played Professor Henry Jones in *Indiana Jones and the Last Crusade*?

<table>
<thead>
<tr>
<th>Person</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>name</td>
</tr>
<tr>
<td>10</td>
<td>Ford, Harrison</td>
</tr>
<tr>
<td>11</td>
<td>Connery, Sean</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Movie</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>19</td>
</tr>
</tbody>
</table>

Which would you rather write?

```sql
SELECT Person.name
FROM Person, Character, Movie, Cast
WHERE Person.id = Cast.personId
AND Character.id = Cast.characterId
AND Movie.id = Cast.movieId
AND Character.name = 'Professor Henry Jones'
AND Movie.title = 'Indiana Jones and the Last Crusade';
```

or “Henry Jones Last Crusade”
The Problem

Relational keyword search has been a hot topic since 2002

- Evaluations ad hoc, no standardization

Example (Search Effectiveness)

DISCOVER \(\ll Hristidis \ et \ al. \ll Liu \ et \ al. \ll SPARK \ll Xu \ et \ al.\)
The Problem

Relational keyword search has been a hot topic since 2002

- Evaluations ad hoc, no standardization

Example (Search Effectiveness)

DISCOVER  ≪ Hristidis et al.  ≪ Liu et al.  ≪ SPARK  ≪ Xu et al.

- ≈ 16-fold improvement in search effectiveness during past decade
The Problem

Relational keyword search has been a hot topic since 2002
  - Evaluations ad hoc, no standardization

Example (Search Effectiveness)

DISCOVER ≪ Hristidis et al. ≪ Liu et al. ≪ SPARK ≪ Xu et al.
  - ≈ 16-fold improvement in search effectiveness during past decade
  - Liu et al. claim to be better than Google
The Problem

Relational keyword search has been a hot topic since 2002
► Evaluations ad hoc, no standardization

Example (Search Effectiveness)
DISCOVER ≪ Hristidis et al. ≪ Liu et al. ≪ SPARK ≪ Xu et al.
► ≈ 16-fold improvement in search effectiveness during past decade
► Liu et al. claim to be better than Google
► SPARK achieves Mean Reciprocal Rank (MRR) of 1.0
The Problem

Relational keyword search has been a hot topic since 2002

- Evaluations ad hoc, no standardization

Example (Search Effectiveness)

DISCOVER ≪ Hristidis et al. ≪ Liu et al. ≪ SPARK ≪ Xu et al.

- ≈ 16-fold improvement in search effectiveness during past decade
- Liu et al. claim to be better than Google
- SPARK achieves Mean Reciprocal Rank (MRR) of 1.0
  - Best systems at TREC score ≈ 0.8 (Webber, 2010)
The Problem

Relational keyword search has been a hot topic since 2002

- Evaluations ad hoc, no standardization

Example (Search Effectiveness)

DISCOVER ≪ Hristidis et al. ≪ Liu et al. ≪ SPARK ≪ Xu et al.

- ≈ 16-fold improvement in search effectiveness during past decade
- Liu et al. claim to be better than Google
- SPARK achieves Mean Reciprocal Rank (MRR) of 1.0
  - Best systems at TREC score ≈ 0.8 (Webber, 2010)

Hypothesis: Existing evaluations overstate retrieval effectiveness
Survey of Existing Evaluations

- Existing experiments unrepeateable
  - Few details included in literature
  - Datasets, query workloads, and relevance assessments not released
- Query workloads vary widely
  - 12–1100 queries included in experiments
  - Too few representative queries
- Experiments
  - Performance-focus, less than half consider search effectiveness
  - Little system comparison
Outline

Introduction
  Background
  Motivation

Evaluation Framework
  Datasets
  Queries
  Relevance Assessments

Experiments

Conclusion
Datasets

- 3 datasets
  - Subsets of IMDb and Wikipedia used in experiments
    - Evaluate systems that assume index fits in memory

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Size (MB)</th>
<th>Relations</th>
<th>Tuples</th>
</tr>
</thead>
<tbody>
<tr>
<td>MONDIAL</td>
<td>10</td>
<td>28</td>
<td>17K</td>
</tr>
<tr>
<td>IMDb</td>
<td>427</td>
<td>6</td>
<td>1.7M</td>
</tr>
<tr>
<td>Wikipedia</td>
<td>378</td>
<td>6</td>
<td>0.2M</td>
</tr>
<tr>
<td>IMDb (original)</td>
<td>9017</td>
<td>44</td>
<td>44.3M</td>
</tr>
<tr>
<td>Wikipedia (original)</td>
<td>670</td>
<td>42</td>
<td>1.6M</td>
</tr>
</tbody>
</table>

Table: Dataset characteristics.
Query Workload

- 50 information needs (*minimum* for evaluating retrieval systems)
- Query statistics are similar to those submitted to Internet search engines

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Search log</th>
<th>Synthesized</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Q</td>
</tr>
<tr>
<td>MONDIAL</td>
<td>101,903</td>
<td>1–96</td>
</tr>
<tr>
<td>IMDb</td>
<td>122,956</td>
<td>1–95</td>
</tr>
<tr>
<td>Wikipedia</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td>20,527,863</td>
<td>1–245</td>
</tr>
</tbody>
</table>

Legend
- |Q|: total number of queries
- [q]: range in number of query terms
- [q]: average number of terms per query
### Relevance Assessments

- **Binary relevance assessments**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Range</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>MONDIAL</td>
<td>1–35</td>
<td>5.90</td>
</tr>
<tr>
<td>IMDb</td>
<td>1–35</td>
<td>4.32</td>
</tr>
<tr>
<td>Wikipedia</td>
<td>1–13</td>
<td>3.26</td>
</tr>
<tr>
<td><strong>Overall</strong></td>
<td>1–35</td>
<td>4.49</td>
</tr>
</tbody>
</table>

**Legend**

- $[r]$ range in number of relevant results per query
- $\bar{r}$ average number of relevant results per query
Outline

Introduction
  Background
  Motivation

Evaluation Framework
  Datasets
  Queries
  Relevance Assessments

Experiments

Conclusion
Experiments

Objectives and Metrics

- Determine search effectiveness of different systems
  - Mean Reciprocal Rank (MRR)
  - Mean Average Precision (MAP)
- Impact the number of results retrieved has on metrics
  - Interpolated precision
- Correlation of results from different systems
  - Minimizing Kendall distance \( K_{\text{min}} \)
Systems

- 2 major approaches to keyword search in relational databases
  - Relational
    - Specific to relational databases
    - Use IR-style ranking functions
  - Proximity Search
    - Applicable to arbitrary data graphs
    - Minimizes the total weight of result trees
- 8 systems published in major proceedings
  - ... plus our own ranking scheme, structured cover density ranking (CD)
Proximity search systems handle single-entity queries well but not scalable.

Figure: Mean reciprocal rank for queries targeting a single tuple.
Overall Effectiveness

- No ranking scheme outperforms all others
  - IR-style ranking (excluding CD) generally not as good as proximity search

Figure: Mean average precision across all queries.
Additional Experiments

Number of results retrieved
  ▶ Precision-recall curve inaccurate above 40% recall for small $k$
    ▶ $k$ should be at least double the number of relevant results

Ranking Correlation
  ▶ Ranking functions derived from common ancestor produce similar results
    ▶ Prefer simpler ranking functions
Outline

Introduction
  Background
  Motivation

Evaluation Framework
  Datasets
  Queries
  Relevance Assessments

Experiments

Conclusion
Conclusions

- Existing evaluations ad hoc, lacking standardization
  - Standardized evaluation critical to progress
- Our evaluation benchmark is the first designed for keyword search within relational databases
  - Datasets, queries, and relevance assessments available for other researchers
- No existing ranking scheme is most effective on all workloads
  - Improve ranking by considering additional factors
Questions?

Download the datasets, queries, and relevance assessments:
http://www.cs.virginia.edu/~jmc7tp/projects/search