Introduction

- R-KwS systems: take queries as a set of keywords and return JNTs – joint networks of tuples – that fulfill the user needs
- Candidate Networks (CNs): Relational Join Expressions automatically generated from input keywords so that when evaluated by a RDBMS produce relevant JNTs
- Claim: Although the number of generated CNs can be very high, only very few produce answers indeed relevant
- Proposal: Ranking CNs based on their probability of producing relevant answers and select just a few to be evaluated
- Contribution: A probabilistic ranking model that uses a Bayesian belief network to estimate the relevance of a CN given the current state of the DB
- Results: Considerable positive impact, not only on the performance of processing keyword queries, but also on the quality of the answers produced

Architecture

CNRank

- The user inputs an unstructured keyword query
- The system generates a number of Candidate Networks (CNs)
- The CNs generated are ranked using the algorithm CNRank
- Top-k CNs from the ranking are evaluated
- The evaluation process can be carried out by any of the many algorithms proposed in the literature
- Finally, the ranked tuple networks are presented to the user.

Algorithm

1. Input: A set of CNs $C = \{C_1, \ldots, C_l\}$
2. Output: A rank of CNs $R = (C_1, \ldots, C_l)$
3. for each $C_i \in C$ do
4.   $\text{copred} \leftarrow 1$
5.   for each $C_{ij} \in C_i$ do
6.     $\text{let } C_{ij} \leftarrow B_j \supseteq K_j$
7.     $\text{wsum} \leftarrow 0$
8.     for each $K_i \in K_j$ do
9.       $C_{ij} \leftarrow \text{TermIndexLookUp}(C_{ij}, B_j)$
10. if $K_i \neq 0$ then
11.       $w \leftarrow \text{tfis}(K_i, B_j)$
12.       $\text{wsum} \leftarrow \text{wsum} + w$
13. end if
14. end for
15. cos $\leftarrow \text{wsum}/\text{anorm}(B_j)$
16. $\text{copred} \leftarrow \text{copred} \times \cos$
17. end for
18. score $\leftarrow \text{copred} \times \text{copred} \times \frac{1}{|T|}$
19. end for
20. Build $R$ such that $C_{ij} \subseteq C_{ik}$ if $\text{score}_{ji} \geq \text{score}_{ki}$

Given a set $C = \{C_1, C_2, \ldots, C_m\}$ of CNs generated for a keyword query $\alpha$, assigns to each CN a score value that estimates the likelihood of this CN representing the user intention when formulating $\alpha$. The score of $c$ is computed as the joint probability of the keywords in each $K_j$ compose of values of some attribute of $K_i$, considering the current state of the database.

Probabilistic Ranking Model

Datasets and Relevant CNs per query

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Size [MB]</th>
<th>Relations</th>
<th>Tuples</th>
<th>WIC</th>
<th>Query set</th>
<th>Relevant CN</th>
<th>Relevant CN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mondial</td>
<td>9</td>
<td>28</td>
<td>17,115</td>
<td>104</td>
<td>Coffman</td>
<td>129</td>
<td>100%</td>
</tr>
<tr>
<td>IMDb</td>
<td>516</td>
<td>6</td>
<td>1,673,074</td>
<td>4</td>
<td>SPARK</td>
<td>45</td>
<td>60%</td>
</tr>
<tr>
<td>Wikipedia</td>
<td>550</td>
<td>6</td>
<td>206,318</td>
<td>5</td>
<td>INEX</td>
<td>14</td>
<td>100%</td>
</tr>
<tr>
<td>DBLP</td>
<td>40</td>
<td>6</td>
<td>86,080</td>
<td>5</td>
<td>Total</td>
<td>138</td>
<td>85%</td>
</tr>
</tbody>
</table>

Results

Effect of CNRank on CN evaluation in terms of MRR.

Impact of CNRank. SMRR on Coffman query set.

Impact on CN evaluation - MAP – SPARK and INEX (left), Coffman (right).

Impact of CNRank on CN evaluation – Performance.