A Unified Framework for Flexible Query Answering over Structured and Semi-Structured Data
Antonio Maccioni (maccioni@dia.uniroma3.it)
University Roma Tre, Rome, Italy

Abstract

Data-centric applications are generating and consuming data at a faster pace than never before. Although the great availability of data, paradoxically, it is also increasing the difficulty that users face to satisfy their information need, specially when data are directly exposed to final users (e.g., on the Web). It is desirable to supply information systems with features to flexibly help these users in finding the information they require. The goal of this work is to propose a framework (FleQSy) to answer, in an uniform way, different forms of flexible queries over different sources of structured and semi-structured data.

The aim of FleQSy is to govern the heterogeneity of data sources, thus giving a unified interface for accessing and managing these data. Furthermore, the framework guarantees interesting properties such as the monotonical ranking and the possibility to process data stored into distributed databases.

Logical Architecture

FleQSy is a meta-approach that adopts, as a conceptual foundation, a graph model G of the structured and semi-structured data sources. Then it considers the following steps:

1. Indexing (off-line): paths of G are indexed.
2. Pre-processing: the query Q is analysed to individuate some criteria on the structure of the final answers. Then, all paths P of G relevant for Q are retrieved from the index.
3. Clustering: we group together, in a cluster, the paths of P that are similar with respect to the criteria individuated during the pre-processing.
4. Building: we generate the final answers a1, ..., ak by combining, at each run, the most relevant paths in every cluster.

This generation process meets the features of FleQSy.

FleQSy Features

Scalability: approximate algorithms with low (possibly linear) time and space computational complexity.

Monotonicity: return the best (top-k) answers in the first generated instead of enduring to process blocks of n candidates, with n > k, out of which select the best k in a second time.

Distributivity: run the algorithms in a parallel-fashion over data that are not located on the same computer machine.

Results Achieved So Far

We have addressed some flexible query answering problem with FleQSy improving the current state of the art and contributing to solve, in some cases, existing drawbacks.

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>Theoretical Time Complexity</th>
<th>Experimental Time Complexity</th>
<th>Monotonic Ranking</th>
<th>Distributed Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>KS-REL</td>
<td>O(</td>
<td>P</td>
<td>)</td>
<td>O(</td>
</tr>
<tr>
<td>KS-RDF</td>
<td>O(</td>
<td>P</td>
<td>)</td>
<td>O(</td>
</tr>
<tr>
<td>AGP-RDF</td>
<td>O(</td>
<td>P</td>
<td>^2)</td>
<td>O(</td>
</tr>
</tbody>
</table>

KS-REL = keyword search over relational data.
KS-RDF = keyword search over RDF data.
AGP-RDF = approximate graph Matching over RDF data.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Fullfilled* | *Not fulfilled yet*