On Combining RDF Streams and Remotely Stored Background Data

S. Dehghanzadeh, D. Dell'Aglio, S. Gao, E. Della Valle, A. Mileo, A. Bernstein
Overview

Evaluation of **Continuous Queries** where RDF streams are combined with background data.

![Diagram showing RDF Stream Generator, Window, Join, and Background data](image-url)
Initial RSP research (2009), ...

Assumption: Background data is static

RDF Stream Generator

Window

Join

Background data

Not realistic!
Assumption: Background data is **static changes** and it is stored **locally**

And the **Web**?
Assumption: Background data is static changes and it is stored locally on the Web

... RSP research goes further (2015)
Streaming data vs Background data

RDF Stream
- Pushed in the RSP Engine
  - Dynamic
    - Small volume
- Background data
  - Pulled by the RSP engine
    - Quasi-static
    - Huge volume

Is it possible to avoid a continuous access to the BKG data?
Streaming data vs Background data

In other words: **what** to pull and **when** to pull BKG data?
Views and Maintenance processes

A Local View (caching) of data is key but when data changes, the local view becomes stale.

How to design a Maintenance Process to refresh the local view?
Requirements

The Maintenance Process

1. should take into account the change rates of the data elements in the REST API;
2. should consider the dynamicity of the change rate values;
3. should satisfy the Quality of Service constraints on responsiveness and freshness of the answer;
4. may consider the sliding window operator.
Initial solution: WSJ+WBM

- Window
  - Current window
  - Validity (V), how many future windows will involve the element (L)

- JOIN
- WSJ
- WBM
  - Rank and top-k select

- Local View

- Refresher

- REST API

\( \Omega_{join} \)
Experiments

- Cumulative staleness vs. evaluation for different models (WSJ and WBM)
Future work

• Future directions:
  – Design and development of adaptive caches
    • Implementation in C-SPARQL/CQELS
  – Queries with multiple SPARQL endpoints invocations

• Open challenges:
  – Extensions in the Query Language?
    • Quality of Service constraints – Freshness, Responsiveness, Precision, Recall, etc.
  – Extension of SPARQL endpoints?
    • To produce update streams (sparqlPuSH?)
    • Descriptions about the dynamicity of the data, rate limits, etc.
On Combining RDF Streams and Remotely Stored Background Data

S. Dehghanzadeh, D. Dell'Aglio, S. Gao, E. Della Valle, A. Mileo, A. Bernstein

Backup Slides
Example

Window

W_1 W_2 W_3 W_4

Local View

 mappings
compatible mappings
Solution (1): WSJ

WSJ: Focus on the stale mappings involved in the current evaluation
Solution (2): WBM

V: When would the mappings become stale if refreshed now?

<table>
<thead>
<tr>
<th>V</th>
<th>L</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Solution (2): WBM

L: For how many future evaluations the mappings is involved?

<table>
<thead>
<tr>
<th>V</th>
<th>L</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
Solution (2): WBM

WBM ranks the mappings by using a score:
Score = min(L, V)

Window View

Local View

V | L | S
---|---|---
3 | 3 | 3
4 | 1 | 1
1 | 3 | 1