Semantic Information Push for Cultural Heritage Applications

Assist. Prof. Christos Tryfonopoulos
University of the Peloponnese, Tripoli, Greece
Talk outline

- Introduction & motivation
- (Semantic) Information push
  - background
  - usefulness to Cultural Heritage
- Our approach
  - overview
  - experimental results
- Future work
Semantic Web for Cultural Heritage

- Semantic Web (SW) has found a new fascinating field:
  - annotation
  - integration
  - linking
  of Cultural Heritage (CH) data

- CH data on the other hand are typically:
  - (physically) distributed
  - (continuously) evolving
  - (inherently) diverse
  i.e., difficult to handle/exploit!
Who uses this data? (1/2)

- Pretty much everybody in the loop!
  - all stakeholders ranging from:
    • the simple museum visitor to
    • the humanities researcher to
    • the data scientist...

- Museum visitors
  - increasingly more demanding in their museum experience
  - varied in needs (e.g., targeted/exploratory/... visit)
  - personalisation is of paramount importance!
Who uses this data? (2/2)

- Humanities researchers
  - not satisfied with “static” data manipulations any more!
  - interested in emerging views (e.g., new facets/interpretations/stories/...) or
  - aim at patterns in CH data

- Data/IT scientists
  - aim at more practical data qualities
    - accuracy (moderation)
    - richness (integration)
  - need appropriate tools
Technological cross-cuts

- But what is the common technological factor that cross-cuts the desires of CH stakeholders?
  
  **Knowledge bases/graphs!**

- For the museum visitor the key lies in
  - the appropriate *exploitation* of the knowledge base
  - i.e., timely (aka in real-time) identify the proper (and relevant) data

- For the humanities researcher the key lies in
  - the appropriate *evolution monitoring* of the knowledge base
  - i.e., monitor data for new/interesting patterns

- For the data/IT scientist the key lies in
  - the appropriate *curation* of the knowledge base
  - i.e., overview the stream of changes using appropriate tools
Key idea

- Propose a technological solution that
  - accounts for the **dynamicity, vastness and heterogeneity** of the knowledge bases/graphs at hand
  - is able to address all functional requirements in a unifying way
  - enrich the technological arsenal of CH by providing a fundamental building block for a series of applications

- (Semantic) **Information push** may play this role!
  - publish/subscribe
  - information alert
  - information dissemination
  - information filtering
  - ...

Information push in a nutshell

First generation information push systems

- **channel-based** (aka group-based)
  - a set of groups designated by the system
  - each event published to one such group
  - user subscribes to one or more groups of interest
    - think of mailing lists or IP multicast

- **topic-based** (aka subject-based)
  - (a bit) more flexible
  - each event is tagged with a subject (from a vocabulary or arbitrary)
  - user subscribes by specifying the subject (and operations - *, ?, ∨, ∧, ...)

- fast, simple to implement but ...
  - no flexibility, cognitive overload
Information push in a nutshell

Second generation alerting systems

- **content-based** (our case)
  - rich data/query models
  - index queries, match against events/updates
  - matching is
    - complicated (specialised data structures, algorithms)
    - expensive (time, computational effort)

- **Applications**
  - news dissemination
  - digital libraries
  - electronic marketplaces / stock market updates
  - but not applied in a CH domain before!
Key idea (cont’d)

- We have to deal both with **text and structure!** → expressiveness
- We have to do it real-time for **bursty updates!** → efficiency

Semantic Information Push

Personalization

User profiling

Recommender systems

- Semantic Information Push
- IDOC
- GeoNames
- CRM
- W3C
- SKOS
- DBpedia
- Getty
So, how is this of benefit to CH stakeholders?

Information push for end users:

- Angela is a museum visitor and WWII aficionado
  - explicit (active) profile creation for interests
    - e.g., poetry, WWII, art
  - implicit profile augmentation on user interactions
    - visited sites/reads, context (e.g., location, device type)

- Receives notifications on events of interest, e.g.,
  - a WWII antique fair as she passes nearby (LBS)
  - a connection of a museum artifact to the bombing of Nagasaki
  - a new interpretation of H. Goldbaum’s “In the Shadow of Great Times” poem
So, how is this of benefit to CH stakeholders?

Information push for humanities researchers:

- Amalia is a majoring in the history of History of Art
  - regularly searches relevant online resources (e.g., SemScholar or MSA)
  - mainly interested in
    - retrieving scientific publication in the relevant domain
    - following prominent works in the area
  - has to deal with field particularities

- Receives notifications on events of interest, e.g.,
  - on long-term information needs
    - new papers
    - interpretations of existing art pieces
    - art reviews of prolific authors are published
So, how is this of benefit to CH stakeholders?

Information push for data/IT scientists:

- Nikki is a computer scientist working on CH ontology maintenance
  - resorts both to automated and crowdsourced methods
  - occasionally needs to integrate with new resources
  - aims at both quality and quantity of the knowledge base

- Receives notifications on events of interest, e.g.,
  - spurious or unusual connections in the knowledge base
    - e.g., mistakenly linking a painting to an oratorio composer as opposed to his namesake painter
  - trending items (e.g., receiving many upvotes)
  - creation/evolution of certain patterns/subgraphs (e.g., clique patterns shared between different artifacts)
Our approach

- Expressive continuous (SPARQL) queries with
  - textual constraints
    - Boolean expressions over keywords
    - word proximity/phrases
  - structural (graph) constraints
    - predefined (chains, stars, cycles, cliques)
    - ... arbitrary(sub)graph patterns

- Over vast, evolving graphs → graph streams
  - edge/node additions
  - edge/node removals
  - attribute/label updates (attribute graphs)

- Matching constraints produce appropriate notifications!
Our Contribution (1/2)

- Extend SPARQL with textual information push
  - Boolean, word proximity, phrase operators

```sparql
SELECT ?event
WHERE {?event type Artifact.
  ?publication description ?descr.
  FILTER contains(?title, "alexander" NEAR [0,1] "great")
  FILTER contains(?descr, "doctor" AND ("friendship" OR "trust"))
}
```
Our Contribution (2/2)

- Algorithm STIP (Structural and Textual Information Push)
  - inverted index to accommodate both
    - structural constraints and
    - textual constraints
  - unified structure
  - emphasis on efficiency

- Identify the tradeoff between:
  - expressiveness and
  - efficiency

- Note that we want this to be a lower-level building block for CH → keep it as generic as possible!
Obvious solution: Brute Force

- No index on the cont. queries...
- Sequentially evaluate them against every graph update!
- Perfect to motivate info push...

<table>
<thead>
<tr>
<th>Query ID</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AB, AC, B: w, B: z, C: xyz</td>
</tr>
<tr>
<td>2</td>
<td>AB, BC, CA</td>
</tr>
<tr>
<td>3</td>
<td>AB, AC, EA, DA</td>
</tr>
</tbody>
</table>
Obvious solution: Brute Force

- No index on the cont. queries...
- Sequentially evaluate them against every graph update!
- Perfect to motivate info push...

<table>
<thead>
<tr>
<th>Query ID</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AB, AC, B: w, B: z,</td>
</tr>
<tr>
<td></td>
<td>C: xyz</td>
</tr>
<tr>
<td>2</td>
<td>AB, BC, CA</td>
</tr>
<tr>
<td>3</td>
<td>AB, AC, EA, DA</td>
</tr>
</tbody>
</table>
Proposed solution: STIP

- Exploits cont. query commonalities
- Post-process potential matches

<table>
<thead>
<tr>
<th>Key</th>
<th>Query IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>2</td>
</tr>
<tr>
<td>BD</td>
<td>1</td>
</tr>
<tr>
<td>AC</td>
<td>1, 3</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td>C: xyz</td>
<td>1</td>
</tr>
<tr>
<td>AB</td>
<td>1, 2, 3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query ID</th>
<th>Total atomic constraints</th>
<th>Matched atomic constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>
Experimental evaluation

- **Data set**
  - 1M timestamped DBpedia triples
  - added to an initially empty graph as publications/events
  - end result: a graph with 1.2M vertices

- **Continuous query set**
  - artificially created
    - matching: extracted from final graph
    - non-matching: random
  - equiprobably chosen to be chains/stars/cycles/arbitrary graphs
  - 10% had also a textual constraint

- **Baselines**
  - query DB: 10/30/50K queries, query length: 4/5/6 atomic constraints, query selectivity: 5, 10, 15%
Key findings

- **STIP** four orders of magnitude faster in filtering than Brute Force
  - ~2M updates/sec on a commodity PC (but is this enough?)
  - deals effectively with bursts

- **STIP** insensitive to
  - graph size (since evaluation is per update!)
  - continuous query length
  - continuous query selectivity

- Results are
  - preliminary
  - but ... highly promising
Future focus

- **Expressiveness**
  - add vector space queries (non-trivial: window or thresholding? how?)
  - add more query classes (without much performance compromise)
    - shortest path queries (critical to unveil interesting connections)
    - clustering coefficient queries
    - ...

- **Efficiency**
  - further exploit query commonalities
    - e.g., tree structures, automata
  - devise parallelisation
    - e.g., exploit multi-core or cluster environments

- **Deployment**
  - provide a query store and the accompanying event filtering engine
Thank you…

…for your attention!
And thanks to L. Zervakis for his help with the experimental part!

Questions?

For more info:  www.uop.gr/~trifon and soda.dit.uop.gr

- **Information push**
  [TKDE’17, DEBS’15, TLDKS’14, TOIS’09, TKDE’06, SIGIR’04, SIGMOD Record’03, ECDL’02]

- **Semantic Information Management**
  [ESWC’16, K-CAP’15, EDBT’14, ESWC’12, ISWC’11]

- **Digital libraries**
  [JCDL’09, ECDL’08,’07,’05, DELOS’07]

- **Distributed data/information management**
  [AAMAS15, ECIR13, Coopis13 & 11, DAPD09, Int. Comp.07, WISE08, SIGIR05 & 08, P2P08, ICDE06, SIGMOD04, EDBT04]

- **Security/privacy/anonymisation (of users or data)**
  [SIGIR’16, Medical Data Privacy Book ‘15, EDBT’14, CoopIS’09, PODC’08]