Keyword Search over RDF Graphs

Elisa Menendez
emenendez@inf.puc-rio.br
Summary

• Motivation

• Keyword Search over RDF
  • Process
  • Challenges
  • Example

• QUIOW System

• Next Steps
Motivation
Motivation

• Keyword search is an easy way to retrieve information
  • Whether in Web pages or in databases
Motivation

• Keyword search is useful for databases
  • It eliminates the overhead of writing structured queries
  • The user does not need to know the database schema

```
SELECT movie.title FROM movie, director
WHERE movie.director = director.id
AND director.name = 'Tim Burton'
```
Motivation

• From the **Semantic Web** and **Linked Data**
  • **RDF graphs** emerged as a new kind of databases
  • That models data as triples

```
nflx:Beetlejuice mo:direcetedBy nflx:Tim_Burton

subject          predicate          object

PREFIX mo: <http://www.movieontology.org/>
PREFIX nflx: <http://netflix.com/>
```
Motivation

• A collection of RDF triples forms an RDF graph

```
"Beetlejuice" mo:title

"1988" mo:releaseYear

"A couple of ghosts contract..." mo: synopsis

nflx:Beetlejuice rdf:type mo:Movie

nflx:Tim_Burton rdf:type mo:Director
```

```
mo:directedBy
```

```
rdf:type
```

```
rdf:type
```

```
rdf:type
```

```
rdf:type
```

Motivation

- RDF graphs are a type of databases

LOD Cloud 2017

1139 datasets

- Cross Domain
- Geography
- Government
- Life Sciences
- Linguistics
- Media
- Publications
- Social Networking
- User Generated
Motivation

• RDF graphs are interesting sources of knowledge
  • Give more flexibility to describe resources
  • Use W3C standardized graph formats
  • Use popular ontologies
  • Keep increasing
Motivation

• RDF graphs are usually queried with **SPARQL** queries

```
SELECT ?title
WHERE { ?movie rdf:type :Movie .
    ?movie mo:directedBy ?director .
    ?director :name "Tim Burton" .
    ?movie mo:title ?title }
```

⇒ Keyword search is also useful for RDF Graphs
Motivation

• Advantages of RDF Graphs for Keyword Search
  • Well-known graph theories
  • Data and metadata all together
    • And easy to access with SPARQL
  • Integration of multiple sources
    • And easy to access with SPARQL
Keyword Search over RDF Graphs
Keyword Search over RDF Graphs

- Keyword Search Process

Keywords
  - query formulation
  - Query

Documents
  - indexing
  - Indexed Documents

Matching + Ranking

Retrieved Documents

feedback
Keyword Search over RDF Graphs

- Keyword Search over RDF Process

```
Keywords

query formulation

Query

feedback

Matching + Ranking + Connecting

Retrieved Sub Graphs

Graph

indexing

Indexed Triples
```
Keyword Search over RDF Graphs

- Main Challenges
  - Find pieces of information in the graph
    - Indexing and Matching problems
  - Connect the pieces of information
    - NP-Complete problem
  - Ranking sub graphs
Keyword Search over RDF Graphs

• Problem definition
  • Given an RDF graph $G$ and a set of keywords $K$
  • An answer for $K$ is
    • A minimal sub graph of $G$ that covers all keywords in $K$
  • Goal:
    • Find all possible answers for $K$
    • Rank them by relevance
Keyword Search over RDF Graphs

• Example: **Graph G**
Keyword Search over RDF Graphs

- Index literals values

```
R1 :type :Actor :name "Denzel Washington"
R2 :type :Actor :name "Dakota Fanning"
R3 :type :Movie :starring R1
R4 :type :Movie :starring R2
R5 :type :Movie :starring R3

"South Dakota": :place

"Washington": :place
```

Keyword Search over RDF Graphs

- Associating with the corresponding resources

```
R1: type :Actor
   :starring R2: type :Actor
   :starring R3: type :Movie
   :starring

R2: :name "Dakota Fanning"

R3: :starring
    :place "South Dakota"

R4: :type :starring
    :place "Washington"

R5: :type :starring
    :place "Washington"

"Denzel Washington"

"Dakota Fanning"
```
Keyword Search over RDF Graphs

- Simple indexing example

<table>
<thead>
<tr>
<th>Words</th>
<th>Resource (Property)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dakota</td>
<td>R₂ (name), R₃ (place)</td>
</tr>
<tr>
<td>Denzel</td>
<td>R₁ (name)</td>
</tr>
<tr>
<td>Fanning</td>
<td>R₂ (name)</td>
</tr>
<tr>
<td>South</td>
<td>R₃ (place)</td>
</tr>
<tr>
<td>Washington</td>
<td>R₁ (name), R₅ (place)</td>
</tr>
</tbody>
</table>
Keyword Search over RDF Graphs

- Example: \( K = \{ \text{"Dakota", Washington"} \} \)
Keyword Search over RDF Graphs

- “Dakota” matches with \( R_2 \) (name) and \( R_3 \) (place)

```
"Dakota Fanning"
```
```
"Denzel Washington"
```
```
"South Dakota"
```
```
"Washington"
```
Keyword Search over RDF Graphs

• “Washington” matches with $R_1$ (name) and $R_5$ (place)
Keyword Search over RDF Graphs

- Combinations = \{ (R_2, R_5), (R_1, R_3), (R_3, R_5), (R_1, R_2) \}
Keyword Search over RDF Graphs

- Combinations = \{ (R_2, R_5), (R_1, R_3), (R_3, R_5), (R_1, R_2) \}
Keyword Search over RDF Graphs

• A movie in Washington with Dakota Fanning
Keyword Search over RDF Graphs

- Combinations = \{ (R_2, R_5), (R_1, R_3), (R_3, R_5), (R_1, R_2) \}
Keyword Search over RDF Graphs

• A movie in South Dakota with Denzel Washington
Keyword Search over RDF Graphs

- Combinations = \{ (R_2, R_5), (R_1, R_3), (R_3, R_5), (R_1, R_2) \}
Keyword Search over RDF Graphs

- A movie in South Dakota and a movie in Washington
Keyword Search over RDF Graphs

- Combinations = \{ (R_2, R_5), (R_1, R_3), (R_3, R_5), (R_1, R_2) \}

- "Denzel Washington"
- "Dakota Fanning"
- "South Dakota"
Keyword Search over RDF Graphs

- Actor Denzel Washington and actress Dakota Fanning
Keyword Search over RDF Graphs

- A movie with Denzel Washington and Dakota Fanning

```
R1: type Actor :starring R2

R2: name "Dakota Fanning"

R3: type :starring R4

R3: place "South Dakota"

R4: type :starring R5

R5: type :place "Washington"

"Denzel Washington"

"Dakota Fanning"
```
Keyword Search over RDF Graphs

- Problem definition
  - Given an RDF graph $G$ and a set of keywords $K$
  - An answer for $K$ is
    - A minimal sub graph of $G$ that covers all keywords in $K$
- Goal:
  - Find all possible answers for $K$  
  - Rank them by relevance

\textit{Not reasonable!}
QUIOW
A Keyword Search over RDF System
### Production Wellbore

<table>
<thead>
<tr>
<th>Production Wellbore</th>
<th>Well Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>30/6-E-1 A</td>
<td>Den norske stats oljeselskap a.s</td>
</tr>
<tr>
<td>30/6-E-1 A</td>
<td>Rocksource ASA</td>
</tr>
<tr>
<td>30/6-E-1 A</td>
<td>Total E&amp;P Norge AS</td>
</tr>
<tr>
<td>30/6-E-1 A</td>
<td>Amerada Hess Norge AS</td>
</tr>
<tr>
<td>30/6-E-1 A</td>
<td>Elf Petroleum Norge AS</td>
</tr>
<tr>
<td>30/6-E-1 A</td>
<td>Esso Exploration and Production Norway A/S</td>
</tr>
</tbody>
</table>
Oracle 12C

RDF

Auxiliary tables

Join

Class

Prop

Value

Parser

Matching

Greedy

Steiner Tree

SPARQL

executing query

graph

cores

Executor

Matching

Parser

result

Presentation
QUIOW

• QUIOW approaches
  • Find pieces of information in the graph
    • Auxiliary tables for data and metadata
    • Indexing and Matching with Oracle Text
  • Scoring cores based on
    • The keywords they match
    • The type of resource: Class > Property > Value
    • Some basic statics
QUIOW

• QUIOW approaches
  • Find pieces of information in the graph
    • Greedy algorithm to select minimal cores
  • Connect the pieces of information
    • Steiner Tree approximation for the nucleuses
    • Synthesize a SPARQL query to represent the Steiner Tree
  • Rank the results based on the resources score

• Evaluation
  • Petrobras Petroleum Dataset
  • Mondial - 64% of the 50 queries
  • IMDb - 72% of the 50 queries
Next Steps
Next Steps

• QUIOW problems
  • The score of cores are mostly based on keyword terms
    • It is basically a **Boolean Model**
  • Not suitable for ambiguous data as in IMDb and Mondial

• How can we improve the quality of QUIOW results?
  • More sophisticated Information Retrieval models
Next Steps

• Which IR models we can adapt to RDF graphs?
  • Vector Space Model?
  • Language Model?
  • Page Rank?
  • User profiling?

• Which of them improves the quality of QUIOW results?
Next Steps

• Which of them improves the quality of QUIOW results?
  • IMDb and Mondial Benchmarks
  • DBpedia
  • Adapt other benchmarks to RDF
    • Social Book Search
    • Last.fm
Thank you!

Elisa Menendez
emenendez@inf.puc-rio.br