ALGORITHM FOR ANSWER GRAPH CONSTRUCTION FOR KEYWORD QUERIES ON RDF DATA

SemSearch’11 – 4th Semantic Search Workshop

K.Parthasarathy, Dr Dominic Damien
Indian Space Research Organisation, Bangalore, India

Dr P.Sreenivasa Kumar
Indian Institute of Technology, Madras, India
Talk Outline...

- Introduction
- Motivation
- Definitions
- Our Approach
- Illustration
- Ranking
- Future directions and conclusions
Introduction

- Entity-relationship graphs are receiving great attention for representing data in domains such as biological networks, launch vehicles.
- RDF and RDFS has become very popular framework for representing data and metadata.
- Growing semantic data in RDF format provides a fertile soil for semantic search.
Introduction

- Formal keyword queries can express complex information needs but syntax is complex and prior knowledge of schema is needed.
- Keyword queries are user oriented and convenient to use but missing links are to be identified.
- Best approach is use a keyword interface and then translate into queries.
Introduction

- Much work has been reported on keyword search on tree structured data
- Recently there has been considerable interest on graph structured data
- Generic approach
  - Map keywords to data elements
  - Search for substructures on the graph
  - Output Top-k substructures based on scores
Motivation

- Only substructures in the form of trees with distinct trees are computed and root element is assumed to be the answer

- Exploration of all possible paths. Pruning is logical by means of scoring and ranking mechanism

- Semantic relationship (e.g. Class/Sub-Class) is not exploited suitably for answering queries

- Adopting instance based approach which takes longer processing time
Motivation

- Fixing a distance metric a-priori to limit the neighborhood related to graph traversal
- Conventional inverted index is inadequate to support queries over graph data as it is fairly difficult to identify answers that captures rich structural relationship
My research focus

Develop a framework for translating keyword queries on RDF graph data to SPARQL that computes not only trees but also general structures without distance neighbourhood restriction using specific indexing mechanisms in line with algorithmic approach.

Inspiration
Previous research efforts on this direction by [Tran, Camiano etc] & [Zhou, Wang etc]
Skeleton RDF data represented from AIFB
Definition

Data Graph The data graph $G = (N, E)$ where

$N$ is a finite set of nodes which is a disjoint union of
C-Nodes (representing types), EN-Nodes (representing entities) and D-Nodes (data values)

$$N = C-Nodes \cup EN-Nodes \cup D-Nodes$$

Examples

C-Nodes – FullProfessor, PhdStudent

D-Nodes – Rudi Studer, Semantic Web Services
E is the finite set of edges connecting nodes \( n_1, n_2 \) with \( n_1, n_2 \in N \). The different types of edges are IE-Edges (inter-entity edges), EA-Edges (entity-attribute edges) and Type/SubClass edges.

Examples

IE-Edges – Author, WorksAt

EA-Edges – Title, Name

ResearchTopic SubClassOf Topic
**Definition (Continued)**

**CR-Node** is a **C-Node** which has an inter-entity relationship with another **C-Node** either through an **EN-Node** or through edge with the label **SubClassOf**. For example the **C-Node** **PhdStudent** is a CR-Node for the C-Node **Project** and vice-versa.
Queries

Given a list of keywords \(k_1, k_2, \ldots, k_n\)

Answer graph is a minimal sub-graph s.t

- Every keyword is contained in at least one node or edge in \(G\)
- The graph consists only of Nodes mapped to keywords, CR-Nodes connected by inter-entity edges, dummy D-Nodes created during the algorithmic phase and labels belonging to \(L(IE-Edges)\) and \(L(EA-Edges)\)
- Answer A is minimal in the sense that no sub-graph of A can be an answer to Q. If a keyword node is removed then that keyword is not matched. If a non keyword node is removed the graph becomes disconnected.
We focus on efficient algorithm for construction of ranked answer graphs to the keyword query on RDF data represented as a graph, that exploits graph semantics (SubClassOf relationship) and detects not only trees but also cycles.
Our approach (CPH)

- Form component clusters for keywords using C-Nodes and CR-Nodes and then expand the clusters based on the closeness. Closeness is determined at two levels
  - CR-Nodes that are common to a pair of clusters
  - Path of CR-Nodes between clusters where there are no common CR-Nodes
- Prune the hanging nodes after taking union of all C-Nodes/CR-Nodes which is the compliment of original set of nodes
Our approach (CPH)

- Hook the clusters using *similarity property* CR-Nodes as hooks and then merge clusters.
- Continue the process till we get the final answer cluster. Prune hanging nodes if any.
Our approach (CPH)

- Uses *similarity* property during pruning/hooking
  - Nodes are same
  - Nodes are related by Type/SubClassOf relationship
  - Exists a chain of C-Nodes connecting the nodes
Illustration

**Keyword Query:** titles topic webservice student studer

- titles → title
- topic → topic
- webservice → webservice
- student → PhdStudent
- Studer → RudiStuder

**Possible interpretation:** titles with topic webservice by Phd students of Rudi Studer
Component Cluster

- \( \text{Nodes}_{\text{Rudi Studer}} = \{\text{‘Rudi Studer’}, \text{Event}, \text{FullProfessor}, \text{Project}, \text{PhDStudent}\} \)
- \( \text{Nodes}_{\text{PhdStudent}} = \{\text{PhdStudent}, \text{InProceedings}, \text{Project}, \text{FullProfessor}\} \)
- \( \text{Nodes}_{\text{title}} = \{\text{Dummytilenode}, \text{InProceedings}, \text{Author FullProfessor}, \text{ResearchTopic}\} \)
- \( \text{Nodes}_{\text{topic}} = \{\text{Topic}, \text{ResearchTopic}, \text{InProceedings}, \text{Project}\} \)
- \( \text{Nodes}_{\text{webservice}} = \{\text{‘Semantic Web service’}, \text{ResearchTopic}, \text{Project}, \text{InProceedings}\} \)
Some clusters
Hanging node *Event* along with associated edge *ChairOf* is pruned

*Topic* and *Webservice* has isomorphic structure with respect to Class *ResearchTopic* segment

We merge the clusters by pushing D-node Semantic Web Service to the other cluster
Hooking

- InProceedings, PhDStudent, and FullProfessor acts as hook nodes
- Final Cluster will be the answer graph
Clusters formed during hooking
Final answer graph
Ranking

- Three possibilities
  - Different node mappings
  - Different chains of C-Nodes
  - Different hooking
Ranking

- **Compactness Relevance (CR)**
  - Larger the C-Node chain between mapped nodes, smaller the compactness

- **Term Relevance (TR)**
  - AIFB can be mapped to node AIFB or title of a publication with string AIFB.

- **Node Relevance (NR)**
  - Hierarchical score for different categories of nodes/edges

- **Relationship relevance (RR)**
  - Overall C-Nodes that contributes by direct connections vs indirect connections

\[
\text{Rankval}(AG) = \sum_{i \leq n} (SC(k_i, k_j) + (TR(k_i | AG) + TR(k_j | AG))) + NR(AG) + RR(AG)
\]
Conclusions and Future works

- The algorithm is under implementation.
- Standard benchmark sets AIFB, DBLP, TAP and LUBM will be used to validate the approach.
- Enhancement of the approach to include an indexing scheme to aid the algorithmic process.


THANK YOU

QUESTIONS?