Q2Semantic: A Lightweight Keyword Interface to Semantic Search

Haofen Wang[1], Kang Zhang[1], Qiaoling Liu[1],
Thanh Tran[2], and Yong Yu[1]

[1] Apex Lab, Shanghai Jiao Tong University

[2] Institute AIFB, University Karlsruhe, Germany

AIFB

Agenda

- Introduction
- Q2Semantic
  - Workflow
  - Data Pre-Processing
  - Query Interpretation
  - Query Ranking
- Experiments
- Demo
- Conclusions and Future Work

Introduction

- Semantic Web can be seen as an ever growing web of structured and interlinked data
  - Large repositories of such data are available in RDF (DBpedia, TAP, DBLP and etc.)
  - Increasing available of these semantic data offers opportunities for semantic search engines to support more expressive queries
- Query interface in semantic search engines
  - Formal query interface (e.g. SPARQL) is supported in current semantic search engines
  - Natural language query interface as one solution
  - **keyword query interface** is the most popular one (our focus)

Information need

Find specifications about "SVG" whose author's name is "Capin"

The SPARQL query

```
PREFIX  tap:  http://tap.stanford.edu/tap#
SELECT ?spec
WHERE  {
?spec     tap:hasAuthor    ?person.
?spec     tap:label         "SVG".
?person   tap:name          "Capin".
}
```

The keyword query

"SVG" + "Capin"

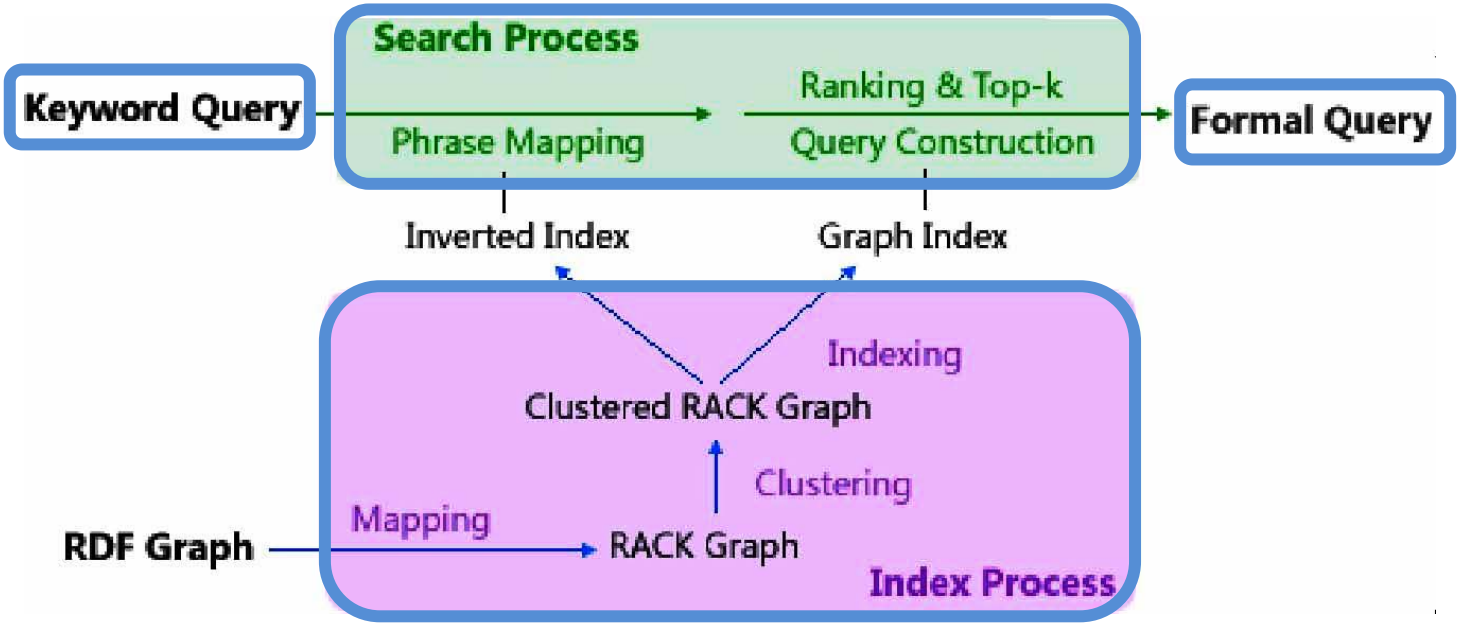
Introduction (cont'd)

- Many studies have been carried out to bridge the gap between keyword queries and formal queries
  - Keyword interfaces for DB or XML
  - Keyword Interfaces for semantic search engines
- Challenges
  - How to deal with keyword phrases which are expressed in the user's own words which do not appear in the RDF data?
  - How to find the relevant query when keywords are ambiguous (ranking)?
  - How to return the relevant queries as quickly as possible (scalability)?

Our Contributions

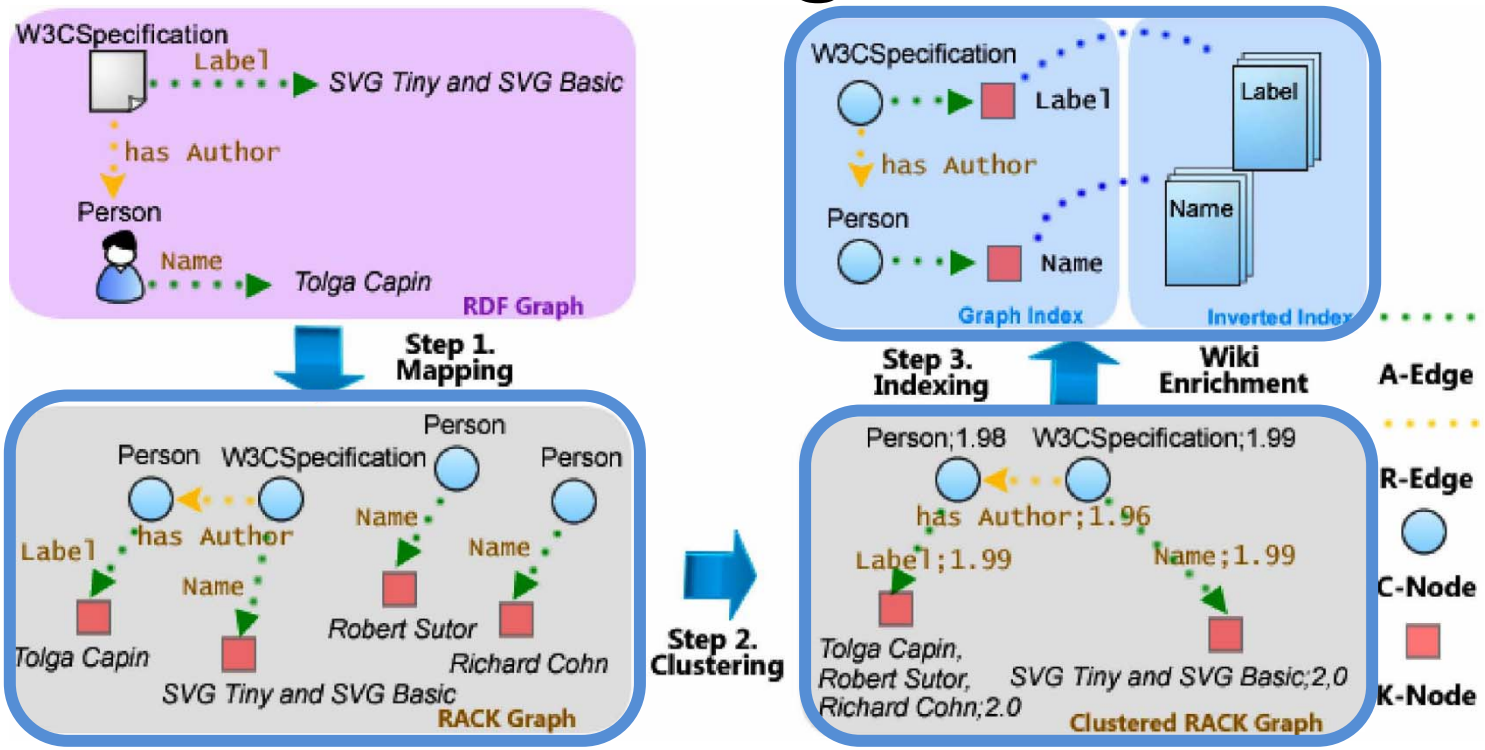
- We leverage terms extracted from Wikipedia to enrich literals described in the original RDF data.
- We adopt several mechanisms for query ranking, which can consider many relevant factors.
- We propose a novel graph data structure called clustered graph and an exploration algorithm.
- Additionally, the exploration algorithm also allows for the construction of the top-*k* queries.

Workflow of Q2Semantic



- Input: a keyword query *K* composed of keyword phrases {k_1, k_2, …, k_n}.
- Search Process
  - Phrase Mapping
  - Query Construction and Ranking
- Index Process
  - Mapping, Clustering and Indexing
- Output: a formal query *F* as a tree of the form <*r*, {p_1, p_2, …, p_n}>, where *r* is the root node of *F* and p_i is a path in *F.*
- In our example, *K* includes k_1 = "Capin" and k_2 = "SVG", and *F* = <*r*, {p_1, p_2}>, where *r* = W3CSpecification, p_1 = <*x1*, *label*, *SVG*> and p_2 = <*x1, hasAuthor, x2, name, Capin*>.

Data Pre-Processing in Q2Semantic



Four rules for mapping from RDF graph to RACK graph

-Every instance of the RDF graph is mapped to a *C-Node* labeled by the concept name that the instance belongs to.

-Every attribute value is mapped to a *K-Node* labeled by the value literal.

-Every relation is mapped to a *R-Edge* that is labeled by the relation name and connects two C-Nodes.

-Every attribute is mapped to an *A-Edge* that is labeled by the attribute name and connects a C-Node with a K-Node.

Four rules for clustering RACK graph

-Two C-Nodes are clustered to one if they have the same label.

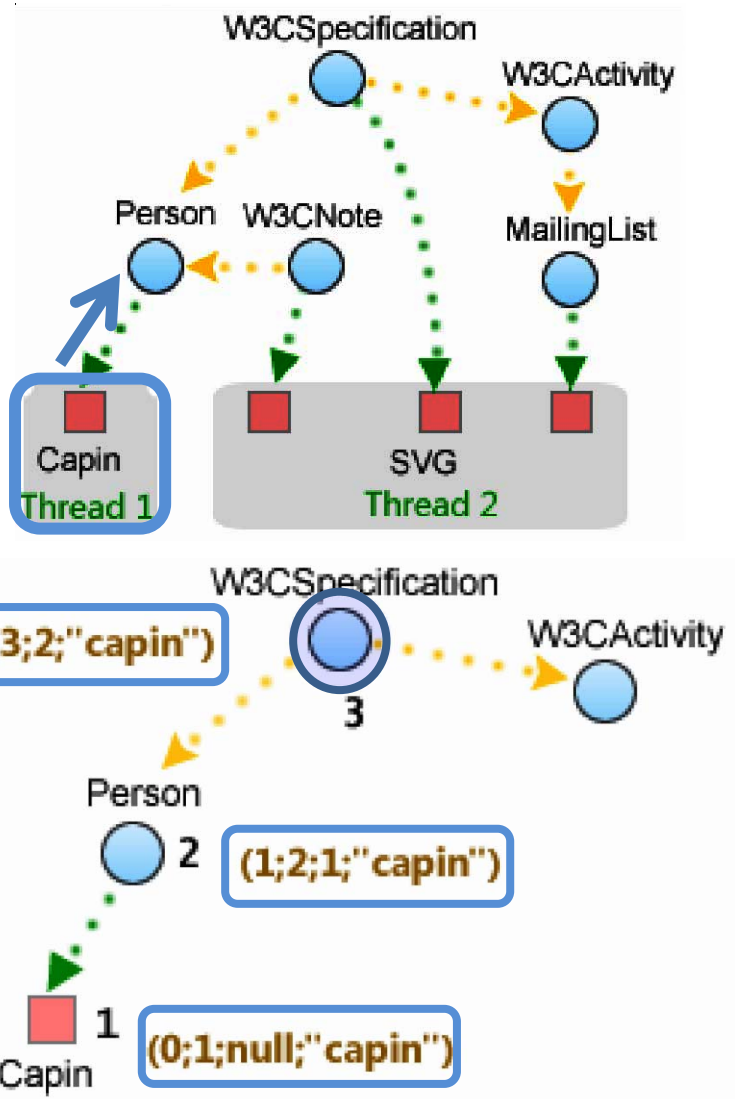
-Two R-Edges are clustered to one if they have the same label and connect the same pair of C-Nodes.

-Two A-Edges are clustered to one if they have the same label and connected to the same C-Node.

-Two K-Nodes are clustered to one if they are connected to the same A-Edge. The resulting node inherits the labels of both these K-Nodes.

Query Interpretation in Q2Semantic

- Phrase Mapping
- Query Construction
  - Thread Expansion (*T-Expansion*)
  - Cursor Expansion (*C-Expansion*)
  - Two strategies for expansion
    - Intra-Thread Strategy
    - Inter-Thread Strategy
  - Optimization for Top-*k* Termination
  - Optimization for Repeated Expansion



Query Ranking in Q2Semantic

- Path only

$$R_1 = \sum_{1 \le i \le n} (\sum_{e \in p_i} 1)$$

- Adding matching relevance

$$R_2 = \sum_{1 \le i \le n} (\frac{1}{D_i} \sum_{e \in p_i} 1)$$

- Adding importance of edges and nodes

$$R_3 = cost_r \sum_{1 \le i \le n} (\frac{1}{D_i} \sum_{e \in p_i} cost_e)$$

$$cost_{node} = 2 - \log_2(\frac{|node|}{N} + 1)$$

$$cost_{edge} = 2 - \log_2(\frac{|edge|}{M} + 1)$$

Experiment Setup

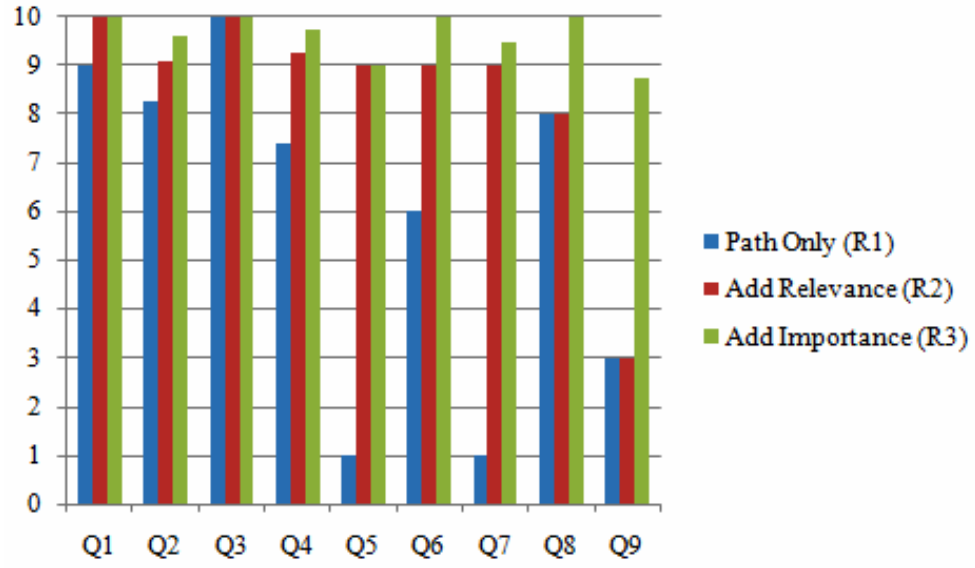
- TAP (220K triples)

Query	Keywords	Potential information need
Q3	Supergirl	Who is called “supergirl"
Q5	Strip, Las Vegas	What is the well-known “Strip" in Las Vegas
Q9	Web Accessibility Initiative, www-rdf-perllib	Find persons who work for Web Accessibility Initiative and involve in the activity with mailing list “www-rdf-perllib"

- DBLP (26M triples)
  - 100 valid queries by combining literals from different attributes (from one to three keywords)
- LUBM(1,0), LUBM(20,0) and LUBM(50,0)
  - 8 queries from the LUBM Query Set (LQ) are used by removing 2 cyclic queries and 4 queries requiring reasoning support

Effectiveness Evaluation

- A simple but effective metric *Target Query Position* (*TQP*): $TQP = 11 - P_{target}$
- TQPs of different ranking schemes on TAP

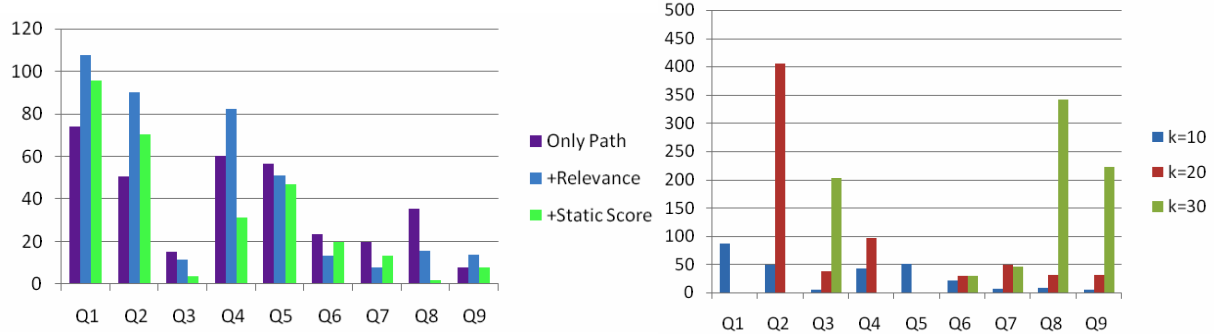


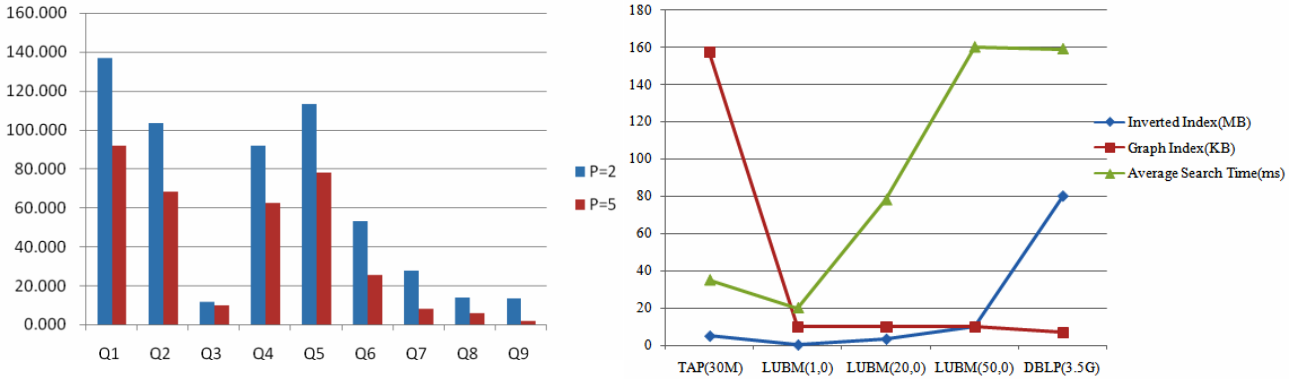
- TQPs on LUBM benchmark queries

TQP	9	10	9	10	8	8	9	10
Query	LQ1	LQ3	LQ4	LQ6	LQ7	LQ8	LQ10	LQ14

Efficiency Evaluation

- Search time under different ranking schemes
- Search time under different top-*k*
- Performance of penalty parameters
- Index size and search time on different datasets
- RACK graph vs. clustered RACK graph

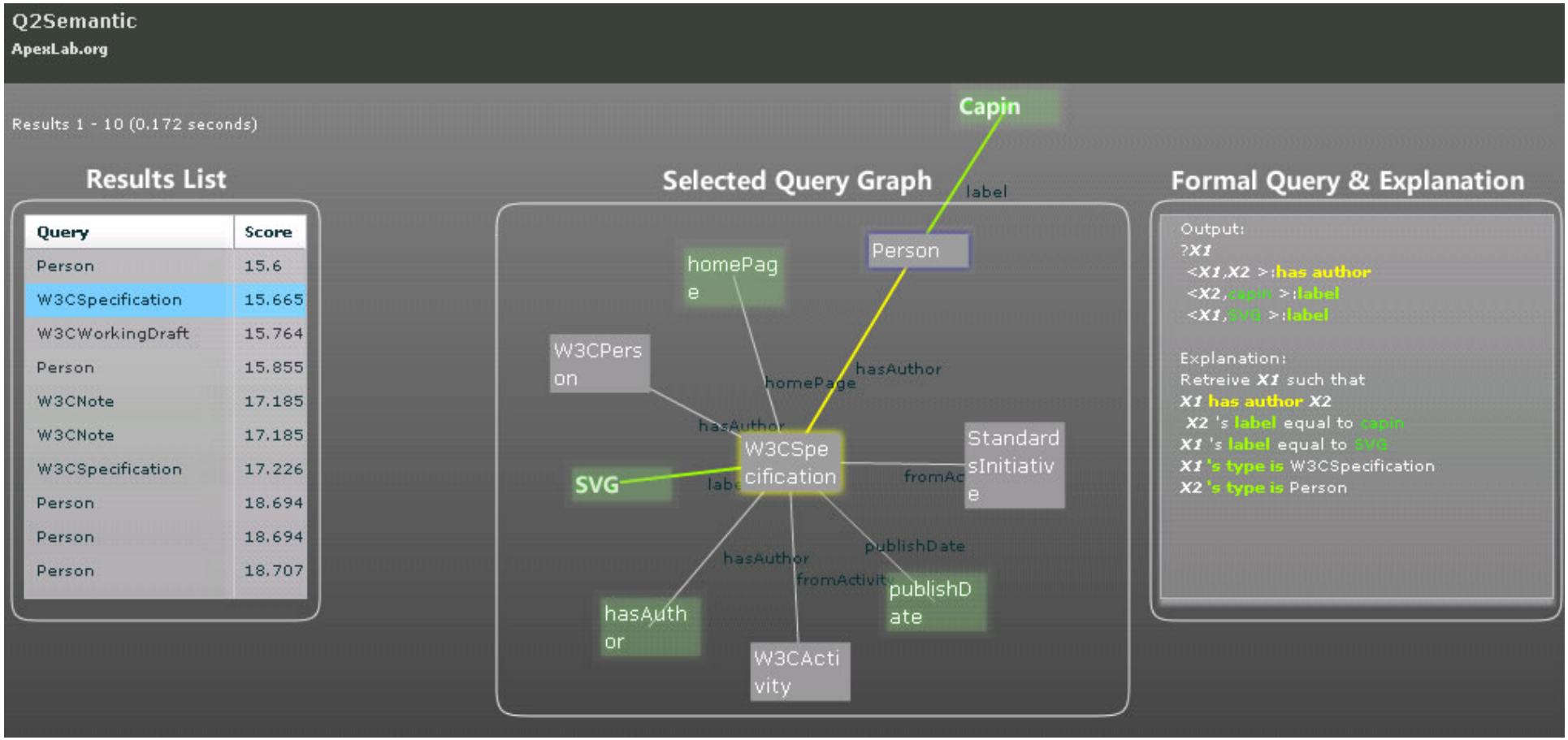




	R-Edge		A-Edge		C-Node		K-Node	
TAP	41914	**158**	87796	**666**	167656	**314**	87796	**666**
LUBM(1,0)	41763	**43**	30230	**39**	16221	**13**	30230	**39**
LUBM(20,0)	1127823	**43**	815511	**39**	411815	**13**	815511	**39**
LUBM(50,0)	2788382	**43**	2015672	**39**	1018501	**13**	2015672	**39**
DBLP	5619110	**19**	12129200	**23**	1366535	**5**	12129200	**23**

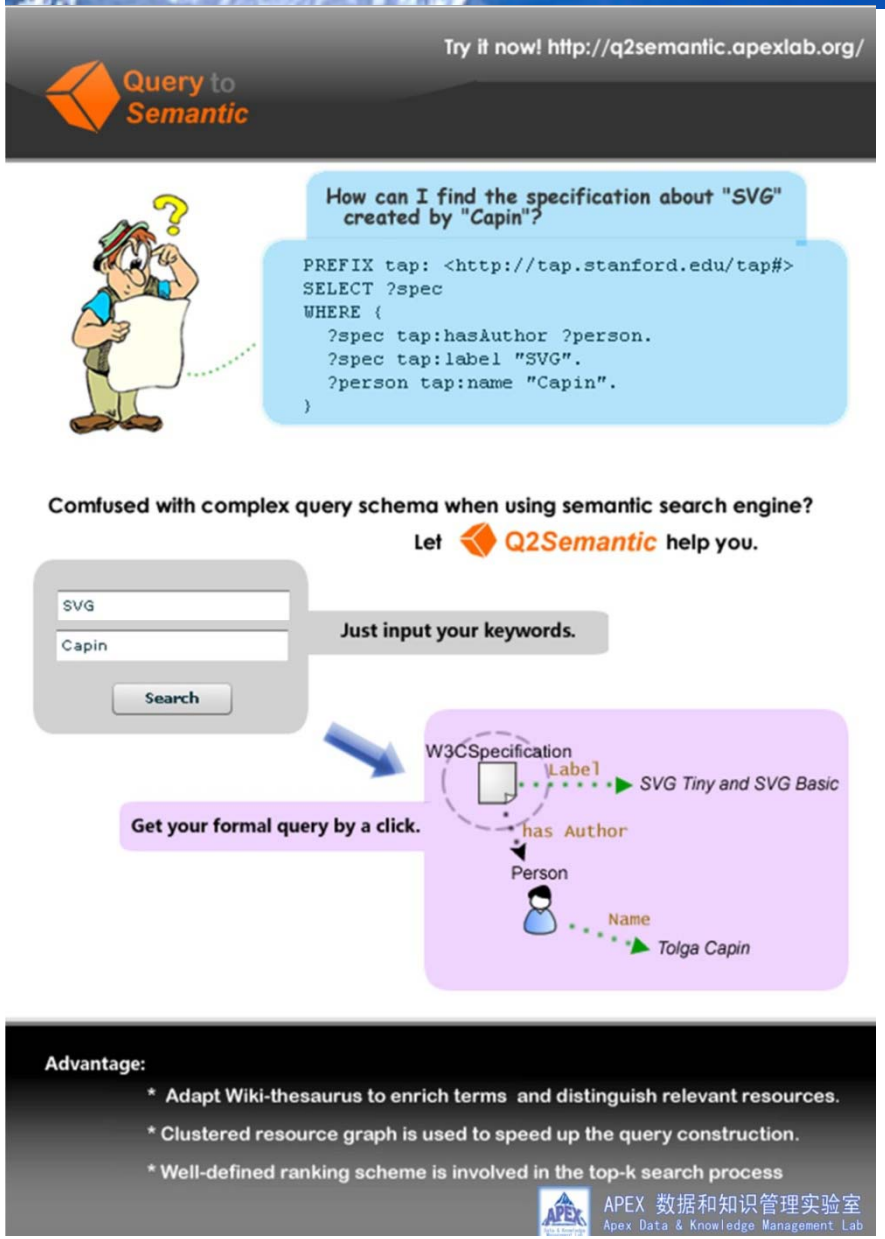
Demo

- Q2Semantic
  - http://q2semantic.apexlab.org



Conclusions and Future Work

- For the efficiency purpose, we propose a new clustered graph index structure as a summary of the original RDF data and support top-*k* formal query construction on it.
- For the effectiveness purpose, we design well-performed ranking schemes. Additionally, we leverage knowledge from Wikipedia to enrich and disambiguates the keyword queries.
- Future Work
  - Query Capability Extension
  - Clustering Method



Questions?

Thank you for your attending!

AIFB