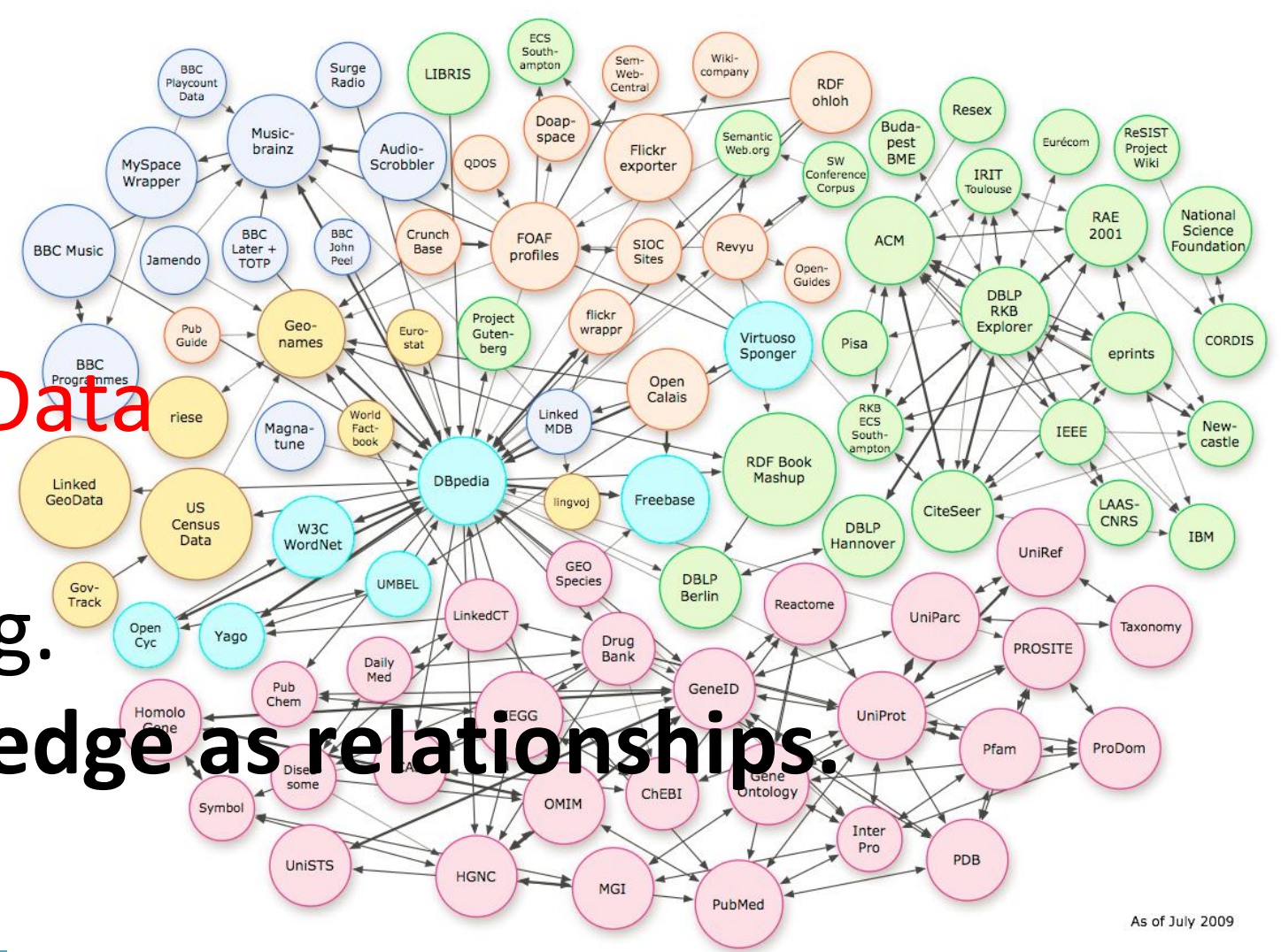
Pragmatics Aware Querying in Heterogeneous RDF Graphs

Amar Viswanathan
Rensselaer Polytechnic Institute

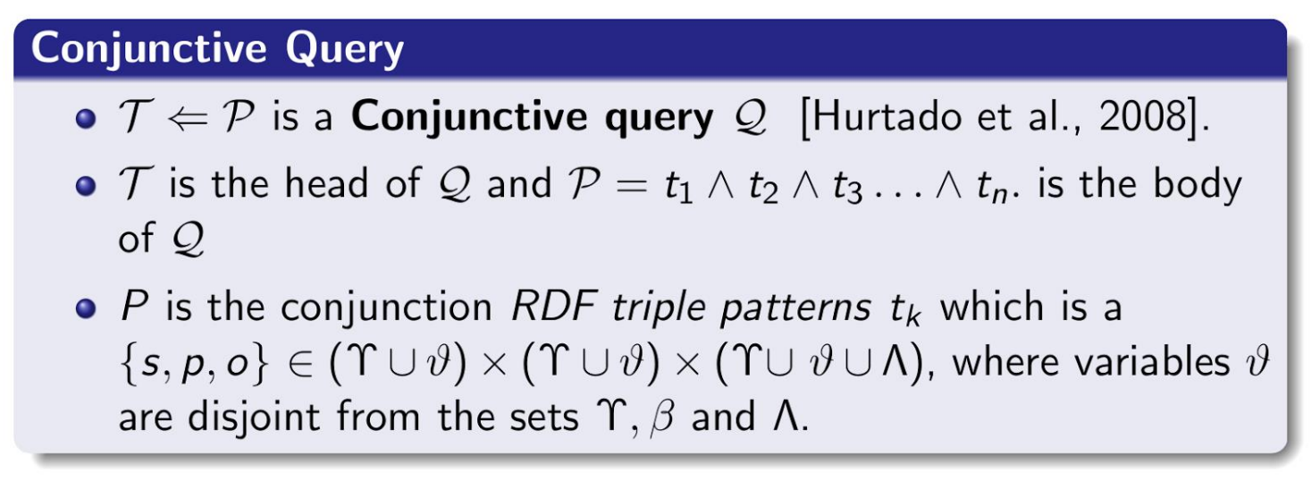
Motivation

- Handling SPARQL **Query Failure** is largely unexplored.
- No intuitive mechanism to aid user when a query fails i.e. **returns zero results.**
- User left to figure out the reason for query failure and system doesn't provide relevant clues.
- Cooperative Answering approach where both the participants engage in conversation to achieve mutual conversational ends.

Why Linked Data?



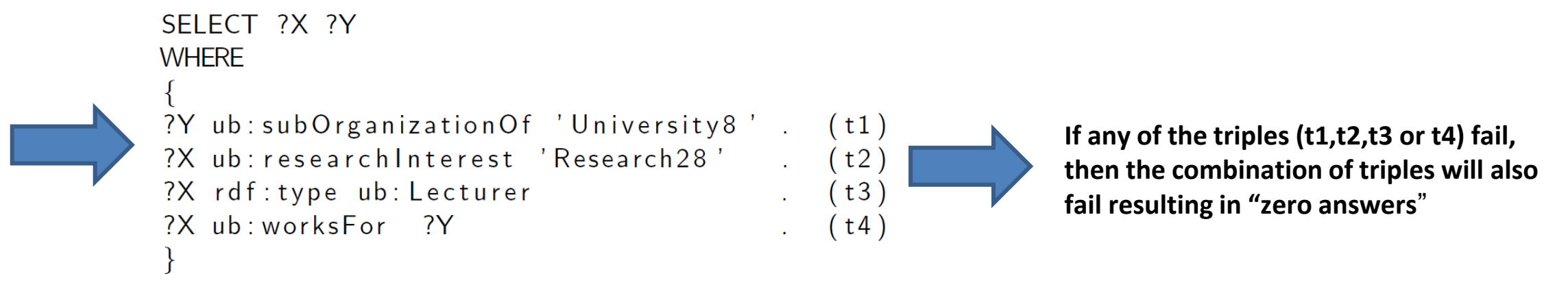
- Proliferation of RDF based Linked Data
- Expressivity of RDF Graphs.
- Relationship based Graph Querying.
- Ability to express extracted **knowledge as relationships.**



Conjunctive Query

- $\mathcal{T} \Leftarrow \mathcal{P}$ is a **Conjunctive query** $\mathcal{Q}$ [Hurtado et al., 2008].
- $\mathcal{T}$ is the head of $\mathcal{Q}$ and $\mathcal{P} = t_1 \wedge t_2 \wedge t_3 \ldots \wedge t_n$. is the body of $\mathcal{Q}$
- P is the conjunction *RDF triple patterns* t_k which is a $\{s, p, o\} \in (\Upsilon \cup \vartheta) \times (\Upsilon \cup \vartheta) \times (\Upsilon \cup \vartheta \cup \Lambda)$, where variables ϑ are disjoint from the sets Υ, β and Λ.

Theoretical Basis for a SPARQL Query



```
SELECT ?X ?Y
WHERE
{
?Y ub:subOrganizationOf 'University8' .   (t1)
?X ub:researchInterest 'Research28'   .   (t2)
?X rdf:type ub:Lecturer                .   (t3)
?X ub:worksFor  ?Y                     .   (t4)
}
```

If any of the triples (t1,t2,t3 or t4) fail, then the combination of triples will also fail resulting in "zero answers"

Example Figure : 1
Sample SPARQL Query

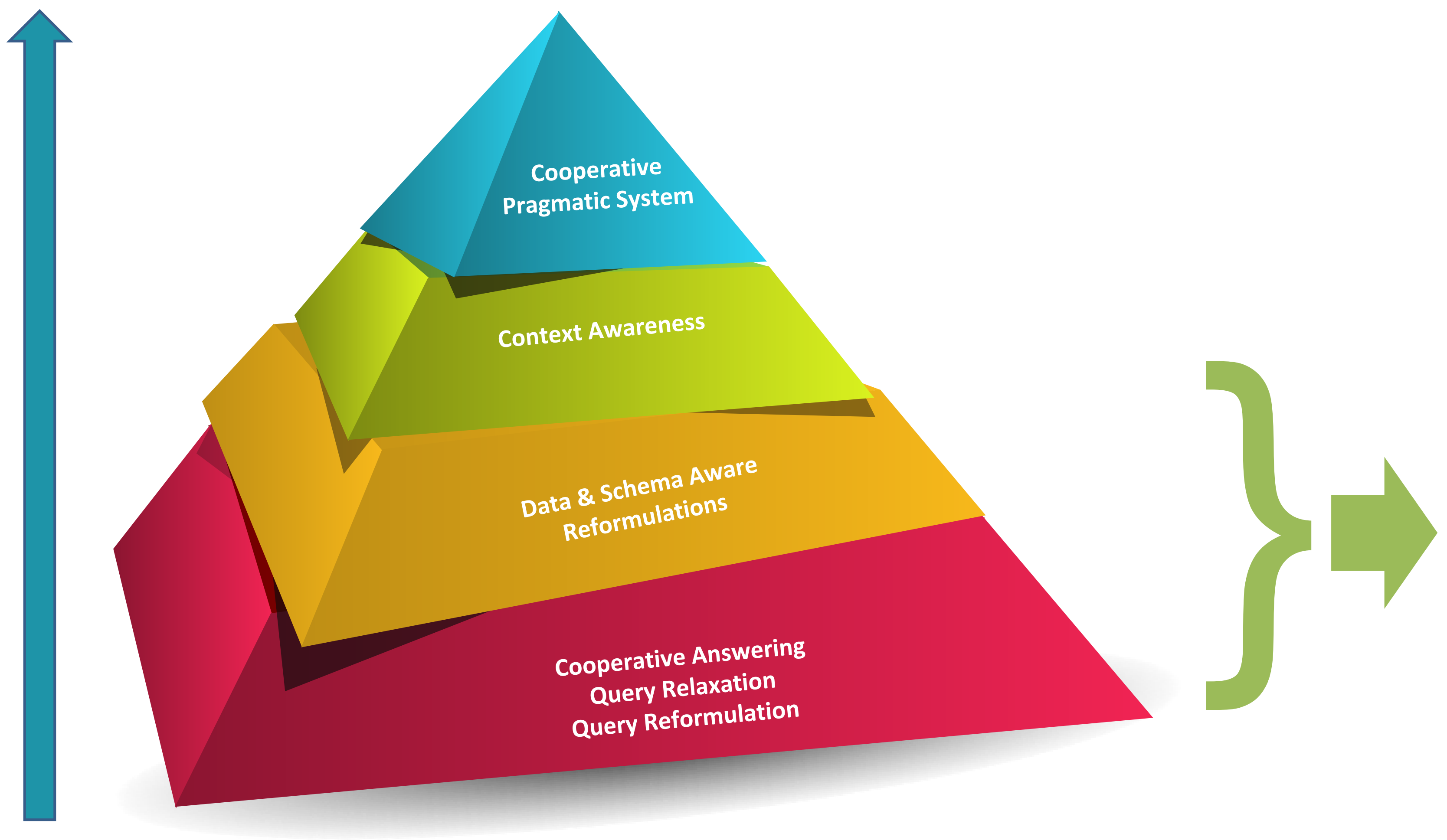
Challenge

> "It must be possible to express a query that doesn't fail when some specified part of the original query fails to match."
>
> --W3C RDF Data Access Group

Questions

- ❑ Can "Query Reformulation" solve the **Query Failure** problem?
  Using the query Example Figure :1 the triple (t3) was rewritten as `?X rdf:type ub:FullProfessor` then we call it a "reformulated query".
- ❑ If yes how many reformulations would you need?
- ❑ Are there rules to be followed? Where would you apply them?
- ❑ Is this process automatic or human aided?
- ❑ How do you identify parts of the query that need reformulation?
- ❑ Does the reformulated query **preserve the semantics** of the original query?

Long term Research Vision



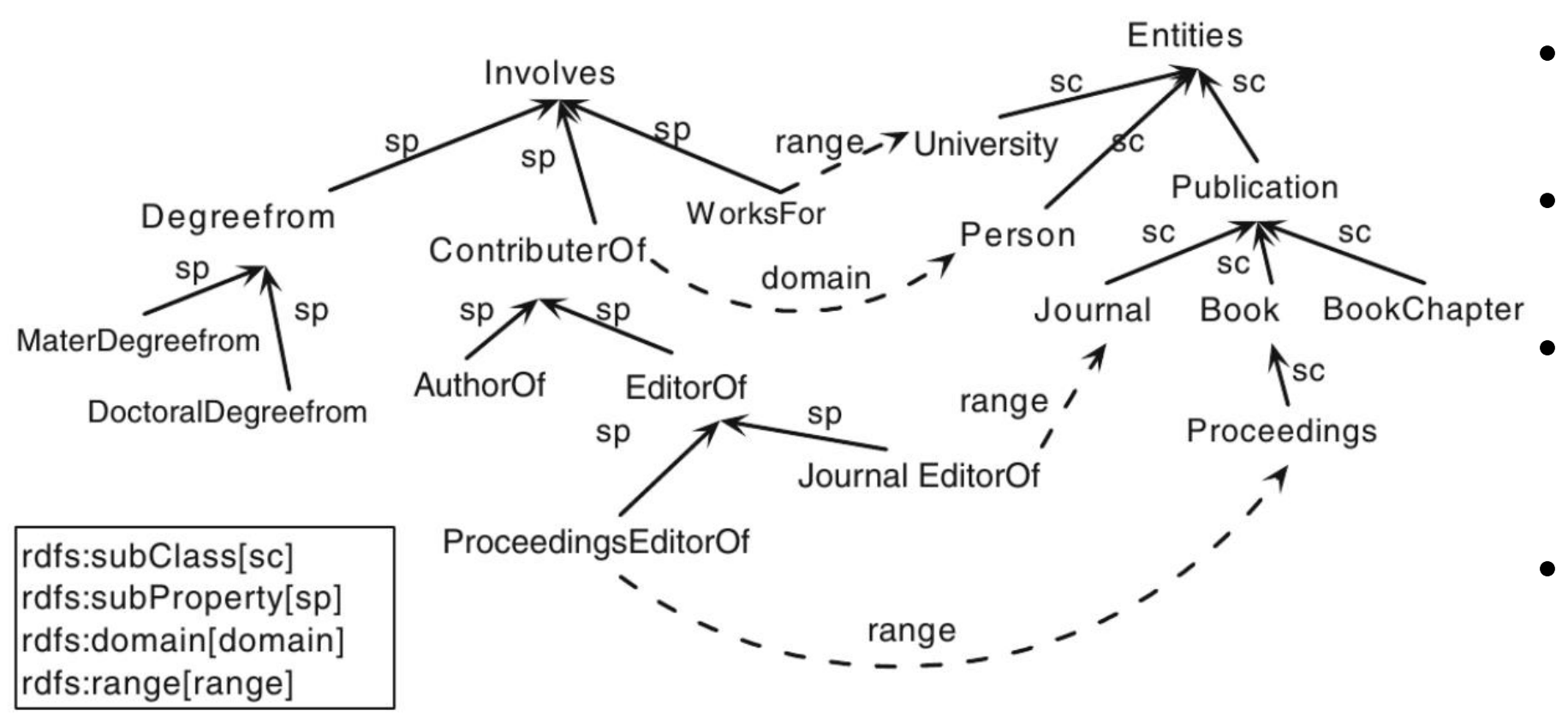
Thesis Contributions

- ❑ A Query Reformulation methodology that takes into account:
  1. RDF Entailment Rules
  2. Data Distribution
  3. Data Availability
  4. Schema Awareness

  } ➡ Data Awareness
- ❑ Finding **failing triples** – extending the Minimal Failing Subquery problem
- ❑ Using data distribution along with the **hierarchy** information to rank the reformulated queries according to their similarities.
- ❑ Using the generated reformulations as an input to further explore the addition of human preferences

Evaluation

Dataset



rdfs:subClass[sc]
rdfs:subProperty[sp]
rdfs:domain[domain]
rdfs:range[range]

- **LUBM** rdf graph used.
- Describes a **University Ontology**.
- Interactions between Students, Professors, Employees and Lecturers.
- Expressive and easily "scalable".

- LUBM100 with 10,115,551 triples, 43 classes, 32 properties as the synthetic dataset.

Evaluation Plan and Baseline

We evaluate our algorithms based on the number of reformulations required to generate a given "k" results when a query failure occurs.

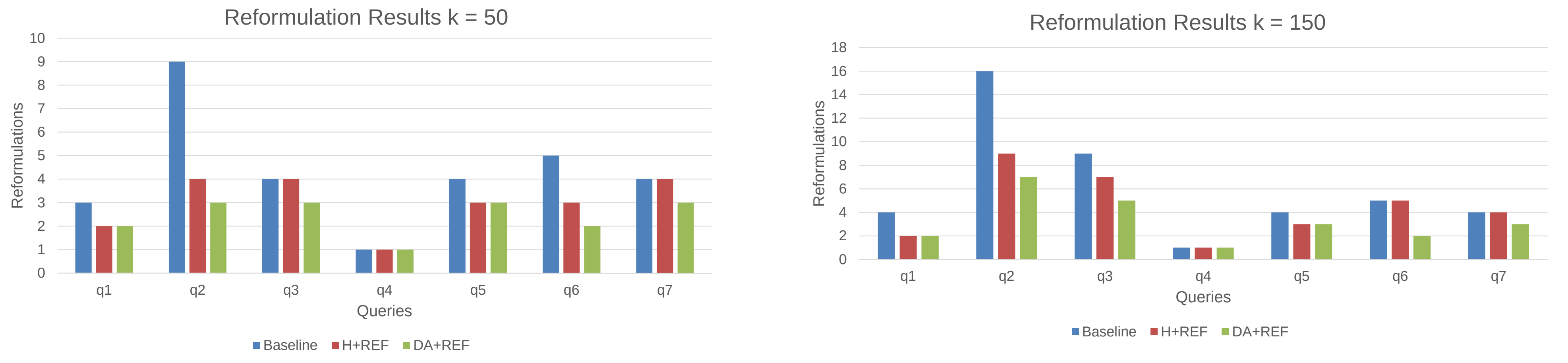
Baseline : Baseline Algorithm by Huang et.al, which reformulates based based on a heuristic and similarity tree.

H+REF : Our algorithm which uses Minimal Failing Subqueries and Hierarchical Similarity.

REF + DA : Uses Data Awareness along with REF

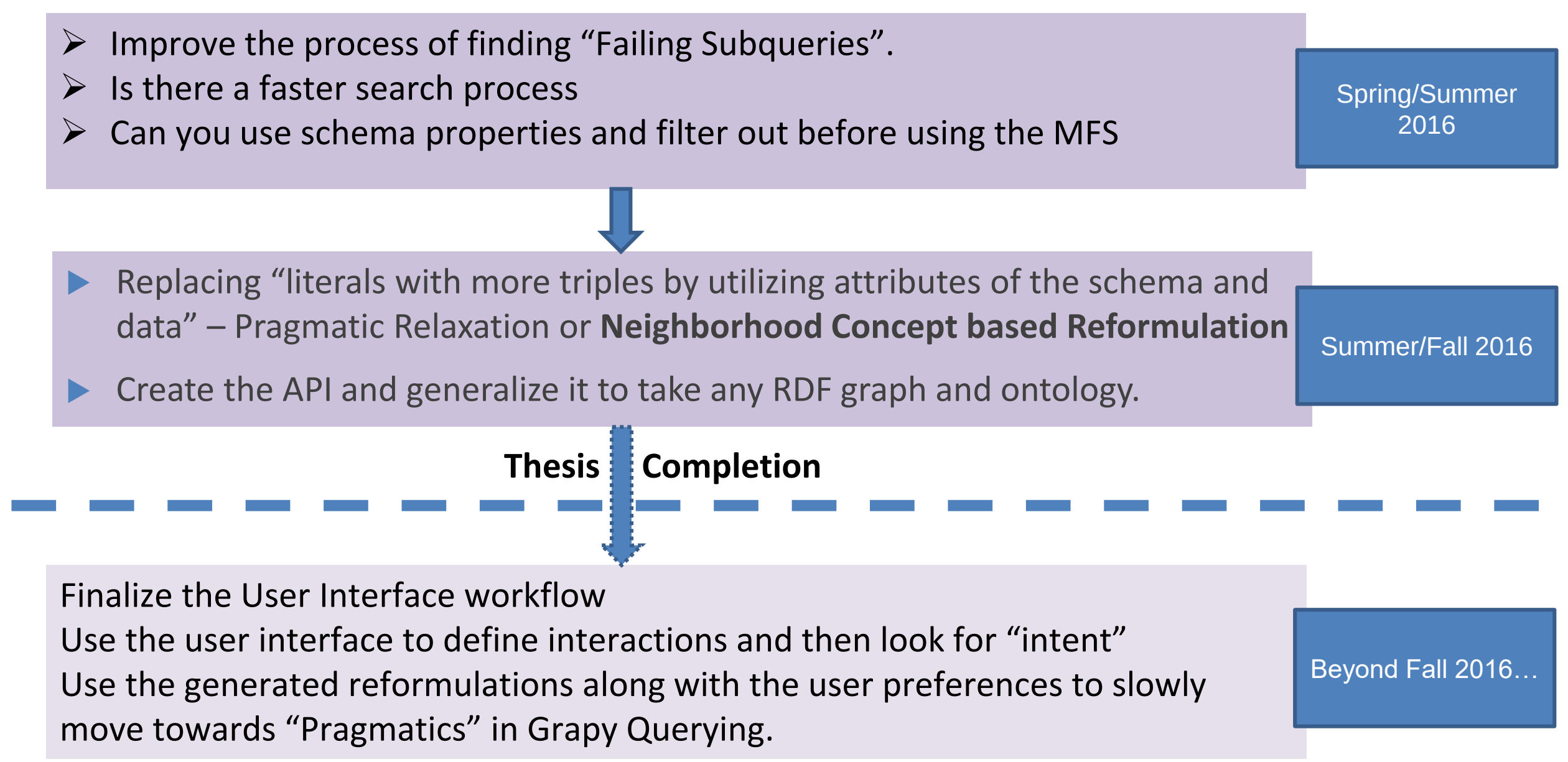
Key Terms
Pragmatic – *Being Aware of Data Distribution and Schema Hierarchy and User preferences*
Query Reformulation - *Using Ontological Relaxation along with Data distribution to rewrite failing queries.*
Query Relaxation – Loosen or "relax" query constraints such that more results or answers are matched.

Current Results



Early results suggest significant reduction in the number of reformulations when compared with baselines.

Future Work Timeline



- Improve the process of finding "Failing Subqueries".
- Is there a faster search process
- Can you use schema properties and filter out before using the MFS

Spring/Summer 2016

- Replacing "literals with more triples by utilizing attributes of the schema and data" – Pragmatic Relaxation or **Neighborhood Concept based Reformulation**
- Create the API and generalize it to take any RDF graph and ontology.

Summer/Fall 2016

Thesis Completion

Finalize the User Interface workflow
Use the user interface to define interactions and then look for "intent"
Use the generated reformulations along with the user preferences to slowly move towards "Pragmatics" in Grapy Querying.

Beyond Fall 2016…

Acknowledgments

I would like to thank my advisor Prof. James A. Hendler and my mentor at IBM Geeth De Mel for their valuable suggestions and guidance in shaping up this thesis. I also thank Prof. Héctor Muñoz-Avila, Prof. Yolanda Gil for their mentoring at AAAI-DC 2016 and David Aha for tolerating my persistent emails.