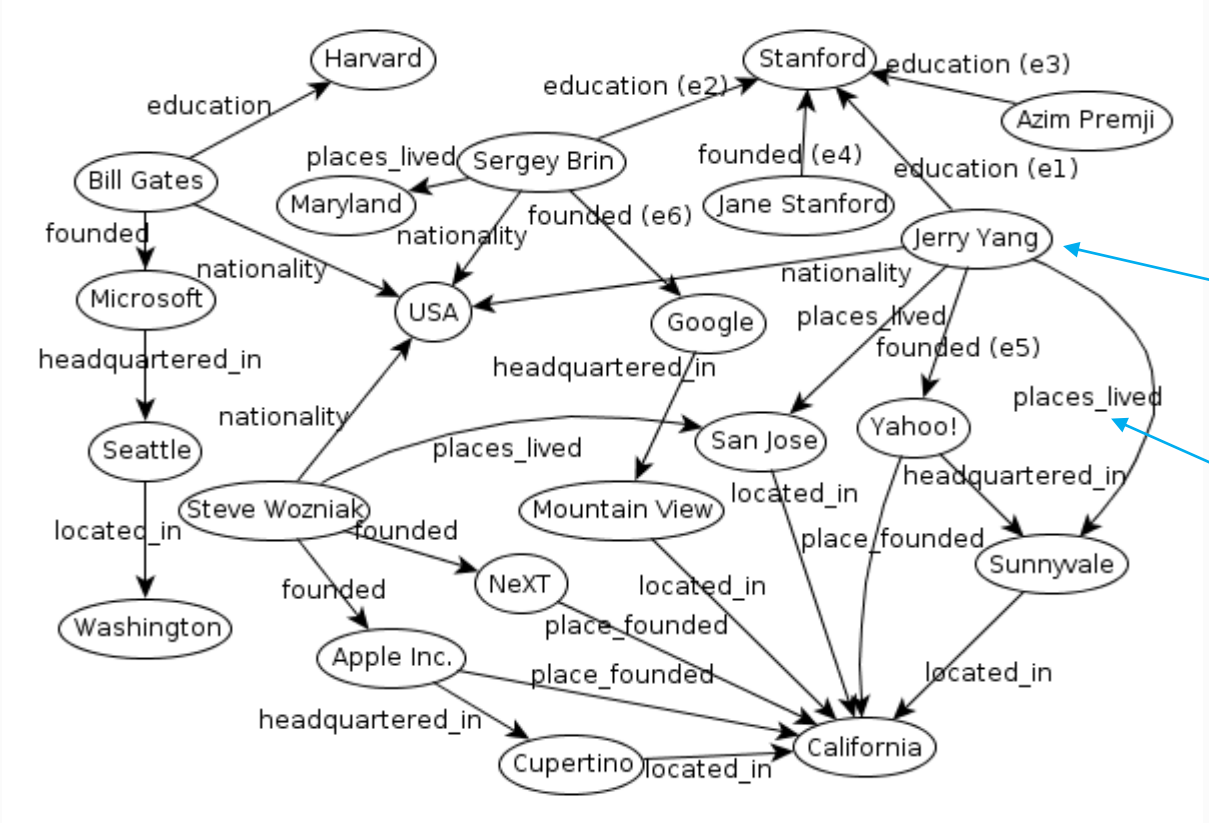
Towards a Query-by-Example System for Knowledge Graphs

Nandish Jayaram (UT-Arlington)
Arijit Khan (UCSB, now at ETH Zurich)
Chengkai Li (UT-Arlington)
Xifeng Yan (UCSB)
Ramez Elmasri (UT-Arlington)

GRADES 2014
June 22, Snowbird, UT

Knowledge Graphs

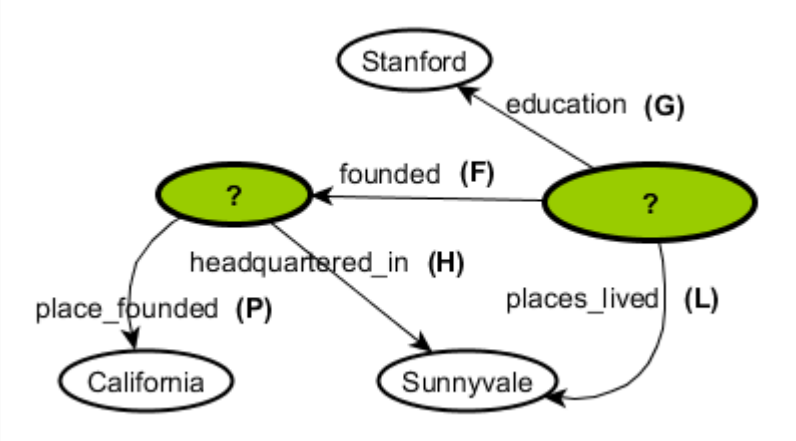
Large and complex graphs capturing millions of entities and relationships between them!



Ubiquitous today: Linking Open Data : 52 billion RDF triples
Freebase : 1.8 billion facts
DBpedia : 470 million facts
YAGO : 120 million facts

How to Query Knowledge Graphs?

- Graph Search / Structured Querying




```
SELECT F.obj, F.src
FROM F, G, H, L, P
WHERE F.prop = 'founded'
AND G.prop = 'education
AND H.prop = 'headquartered_in'
AND L.prop = 'places_lived'
AND P.prop = 'place_founded'
AND F.obj = H.src
AND F.obj = P.src
AND F.src = L.src
AND L.obj = H.obj
AND F.src = G.src
```

- Expertise in constructing structured queries required.
- A good knowledge of the schema of the knowledge graph is required.

Improving Usability of Knowledge Graphs: Prior Arts

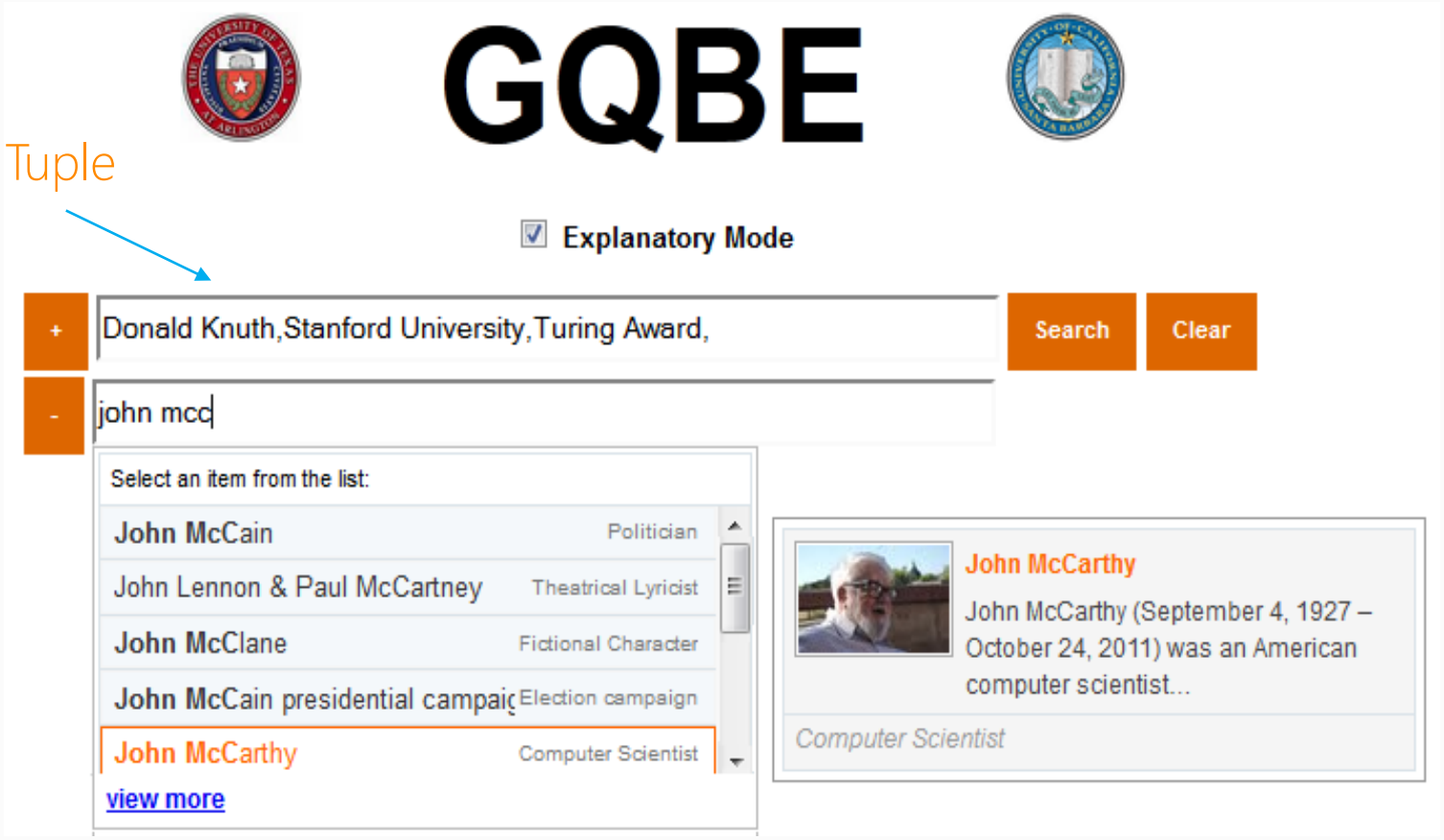
- Keyword Search

  "Software companies located in the Silicon Valley and their founders who studied at Stanford University."

  - Keyword search on graphs [Karger11].
  - Keyword based query formulation [Pound10] [Yao12].
- Natural Language Query
  - Natural language questions based querying [Yahya12].
- Visual Query Interfaces
  - Interactive and form based query construction [Demidova12] [Jarrar12].
  - Visual interface for query graph construction [Chau08] [Jin10].
- Schemaless Graph Querying
  - Use transformations to find matches to a naïve query graph [Yang14].

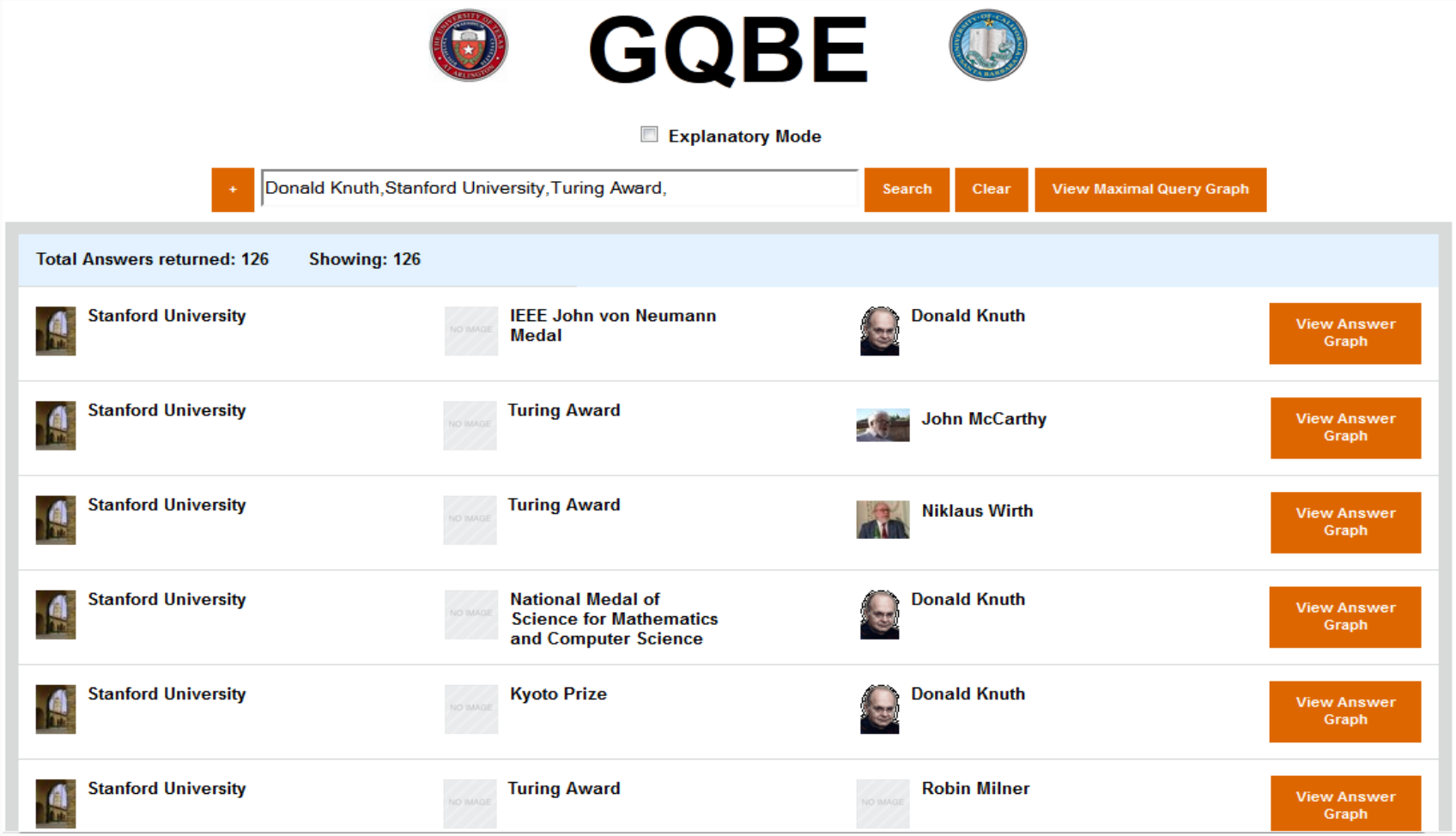
Query by Example Entity Tuples

Given an input *n*-entity tuple(s) (called *n*-tuple), a knowledge graph, and *k*, find top-*k* *n*-tuples that are most similar to the input tuple(s).

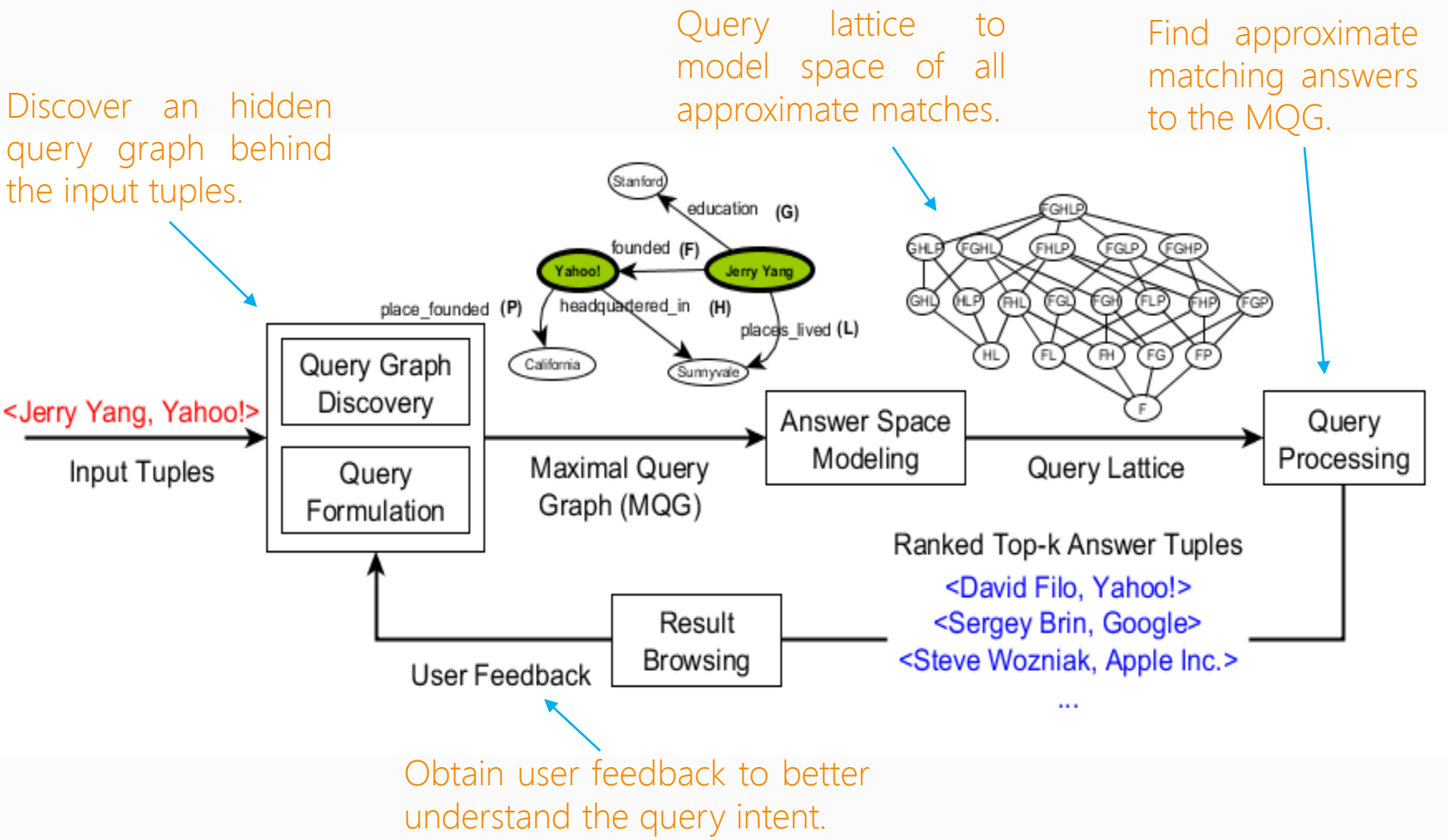


Answer Tuples

Input Tuple : Donald Knuth, Stanford University, Turing Award



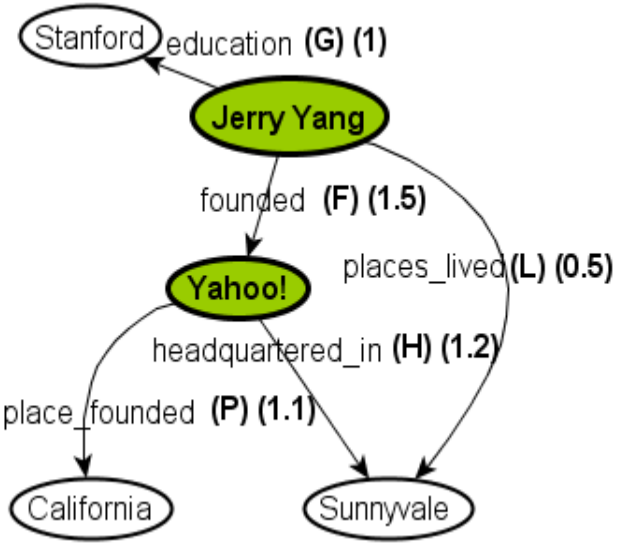
Overall Architecture



- Exemplar Queries [Mottin14]

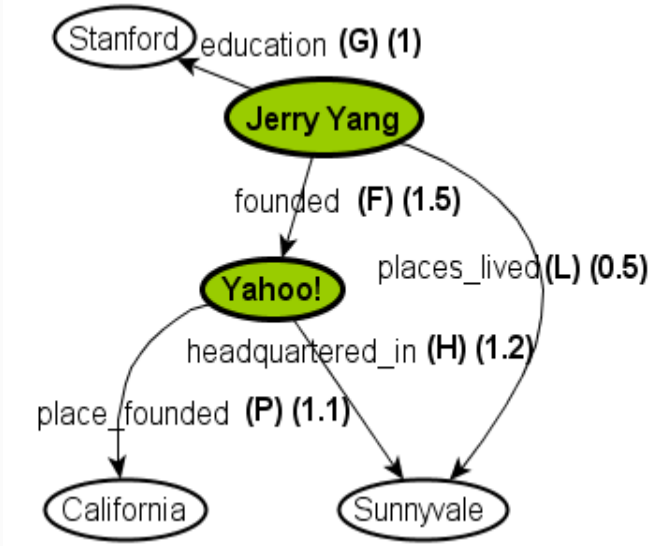
Maximal Query Graph Discovery

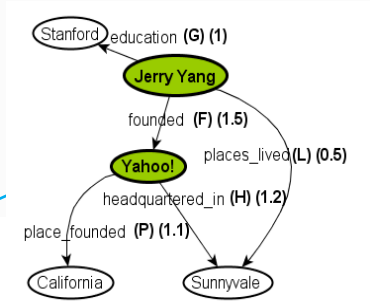
- Given an example tuple like <Jerry Yang, Yahoo!>
  - Define importance of edges by assigning weights to them.
  - Find a small sub-graph with important edges and nodes in the neighborhood of *Jerry Yang* and *Yahoo!*, to form the Maximal Query Graph (MQG).

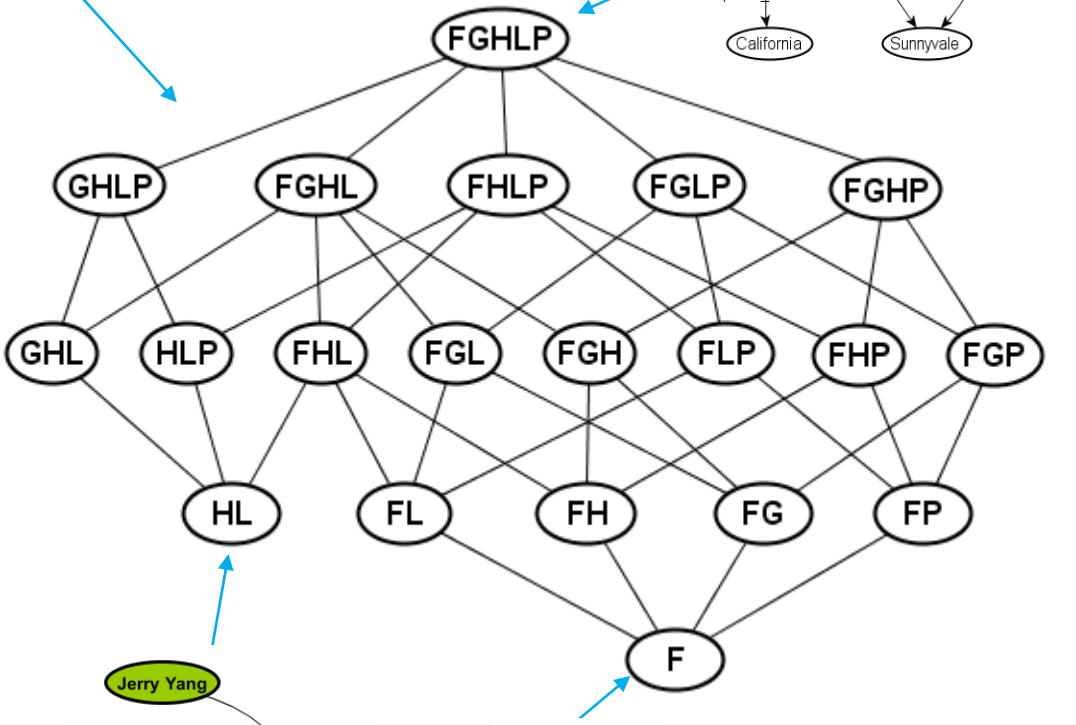


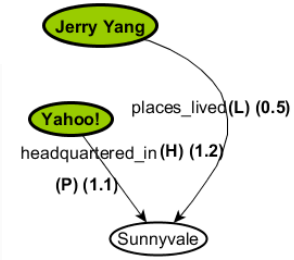
Answer Space Modeling

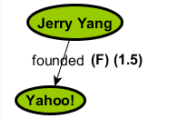
Every other node is a sub-graph of the MQG.





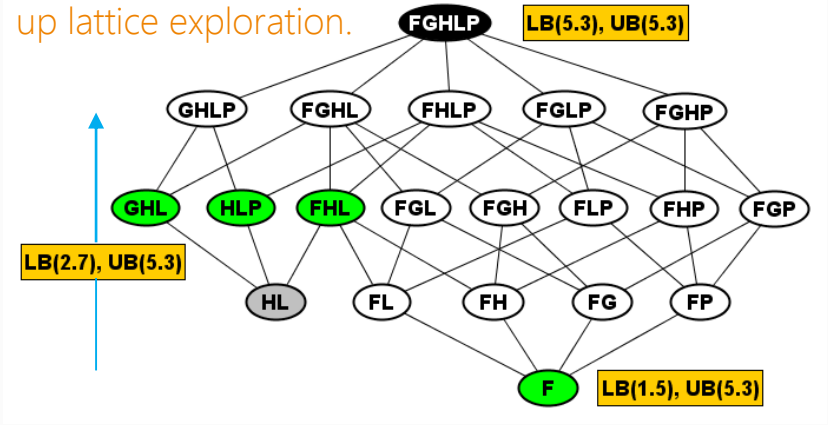




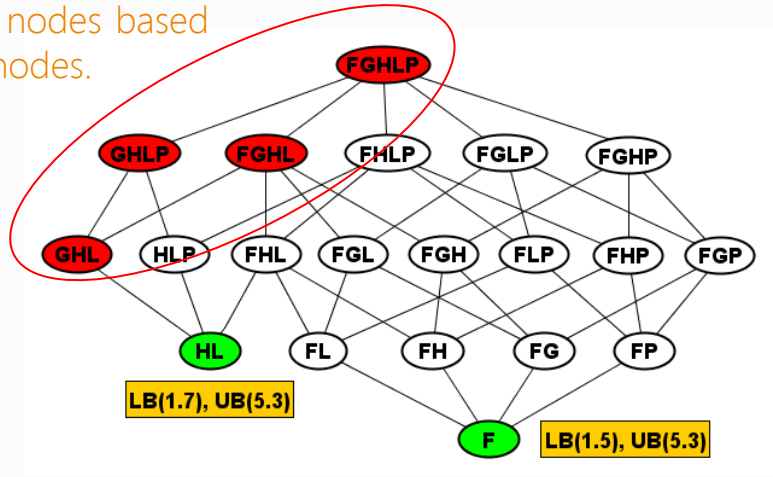


Query Processing

Upper bound based bottom-up lattice exploration.



Pruning nodes based on null nodes.



Lattice evaluation terminated after top-k answers are obtained!

Finding Matching Answer Graphs

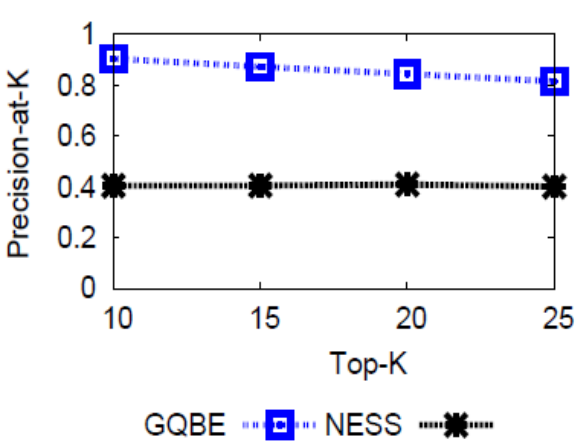
- Exact sub-graph matching, based on indexing techniques.
  - Search on graph databases [Shasha02] [Yan04] [Zhao07] [Zou08].
  - Search on single large graph [Ullman76] [Cordella04] [Shang08] [Zhang09].
- Approximate sub-graph matching.
  - Use various indexes to quickly find approximate matches [Tian08] [Mongiovi10] [Khan13].
  - NESS : uses neighborhood-based indexes to quickly find approximate matches to a query graph [Khan11].

Experiments

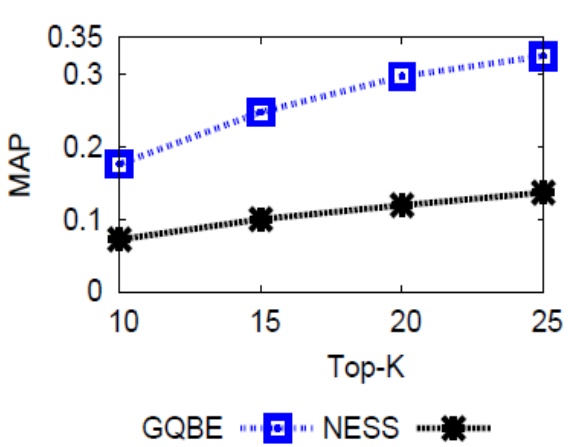
QUERIES:

- 20 Queries on **Freebase** dataset (47 M edges, 27 M nodes, 5.4 K properties)
- 8 Queries on **DBpedia** dataset (2.6 M edges, 759 K nodes, 9 K properties)

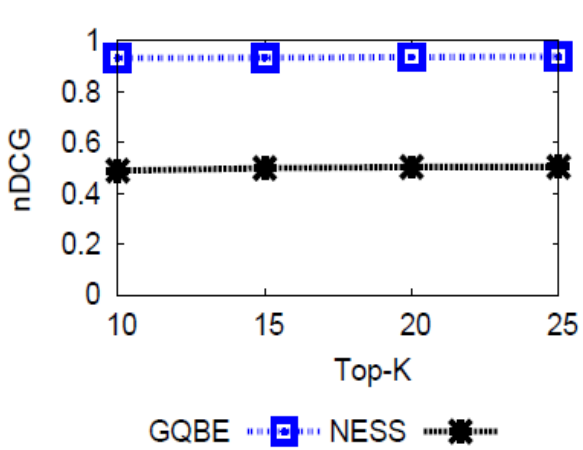
Accuracy Comparison with NESS:



(a) P@k



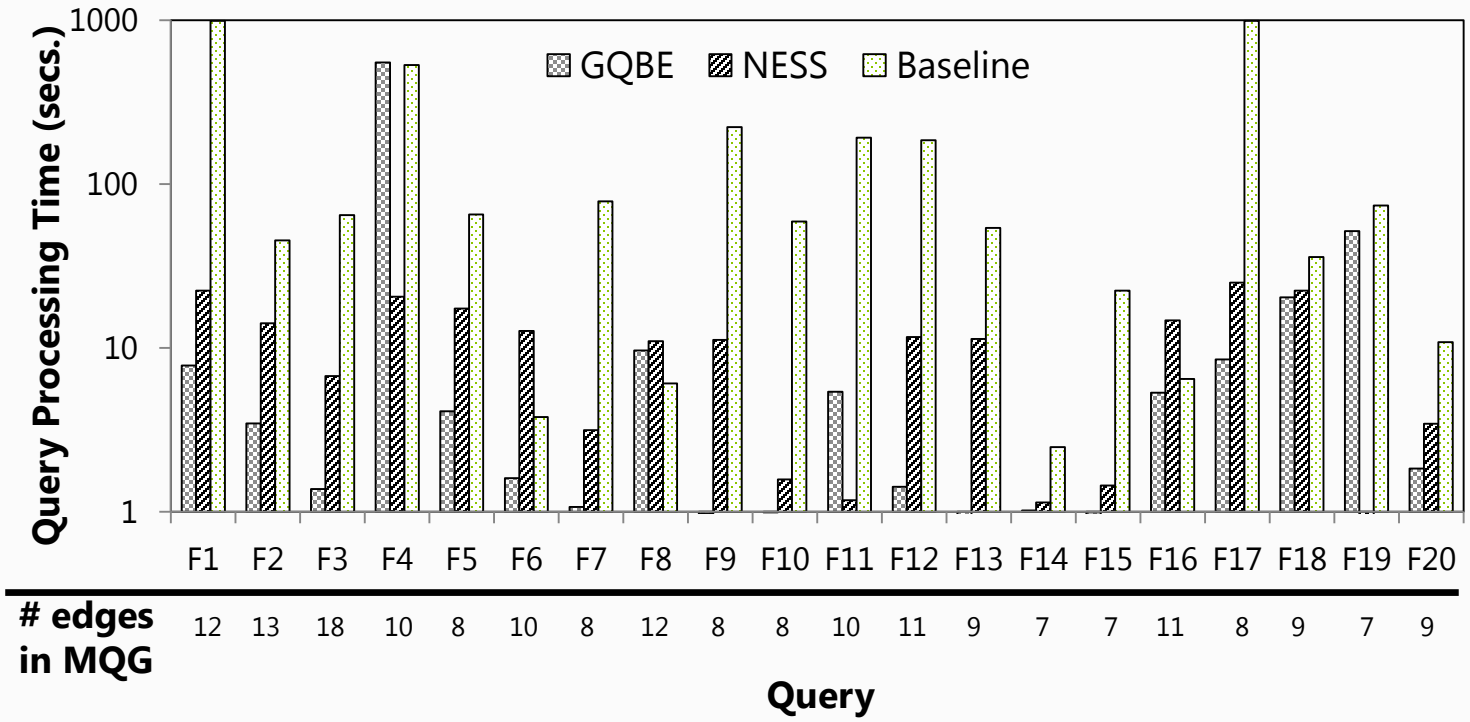
(b) MAP



(c) nDCG

Efficiency Results

Single Query Execution Times (in seconds)



Work in Progress

➢ Maximal Query Graph Discovery:

- ➢ Does not capture the user-intent exactly.
- ➢ Iterative and interactive edge suggestion.

➢ Query Processing:

- ➢ Materializing intermediate join results (millions of rows) can be expensive.
- ➢ Is a better join mechanism when we have more memory at our disposal possible?
- ➢ Distributed lattice exploration mechanism.

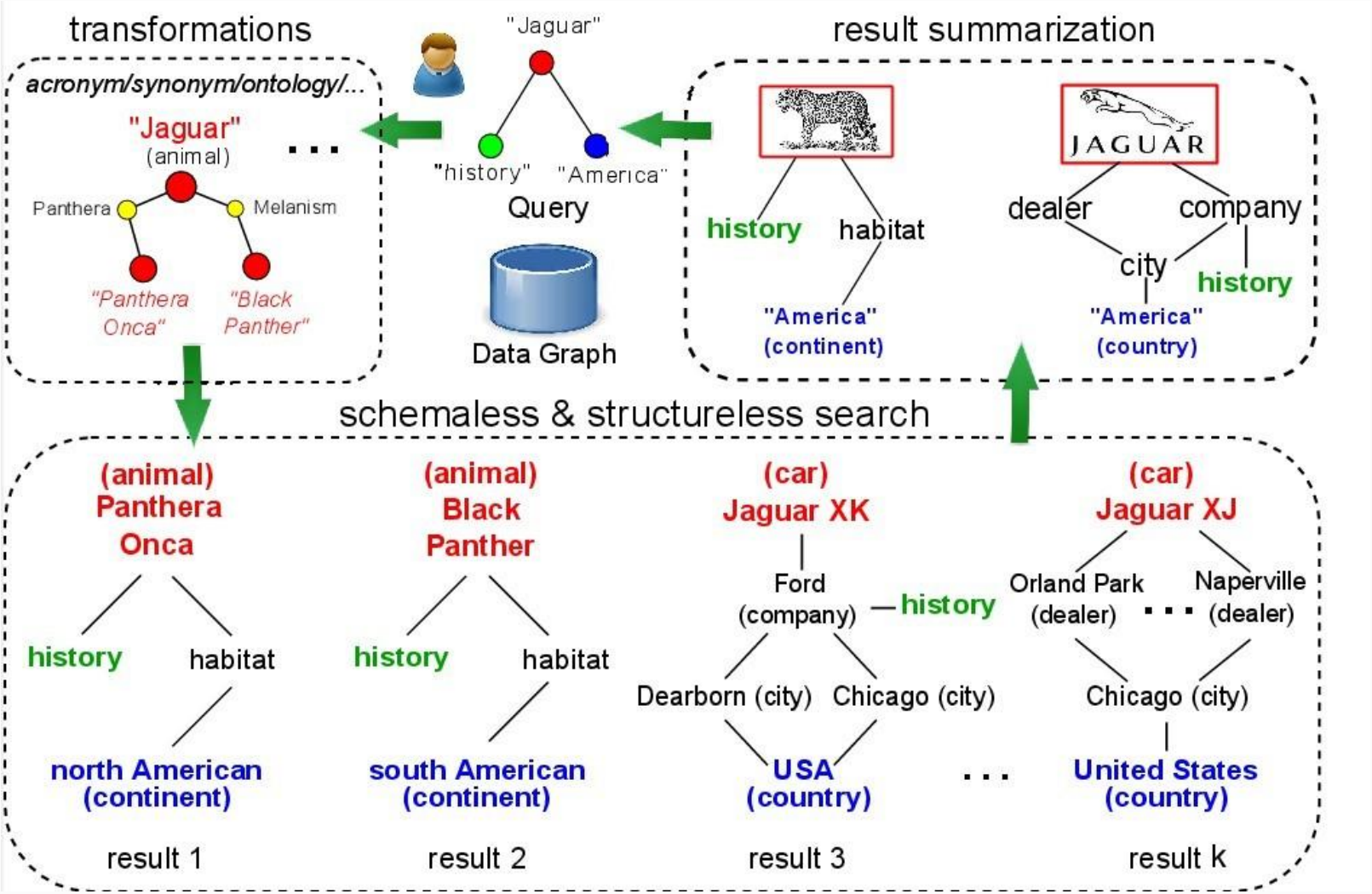
➢ Obtaining User Feedback:

- ➢ User feedback on relevance of answer tuples to re-weight edges.

Work by Xifeng Yan's group at UCSB

(ICDE 2013)

(VLDB 2014)



(SIGMOD 2014 demo, VLDB 2014)

Demo and Technical Details:

- Demo:
  - URL: idir.uta.edu/gqbe
  - Demo paper: *GQBE: Querying knowledge graphs by example entity tuples*, ICDE 2014.
- Technical Details:
  - Full paper under review
  - Archived version: http://arxiv.org/abs/1311.2100