Intuitive and Interactive Query Formulation to Improve the Usability of Query Systems for Heterogeneous Graphs

Nandish Jayaram
University of Texas at Arlington
PhD Advisors: Dr. Chengkai Li, Dr. Ramez Elmasri

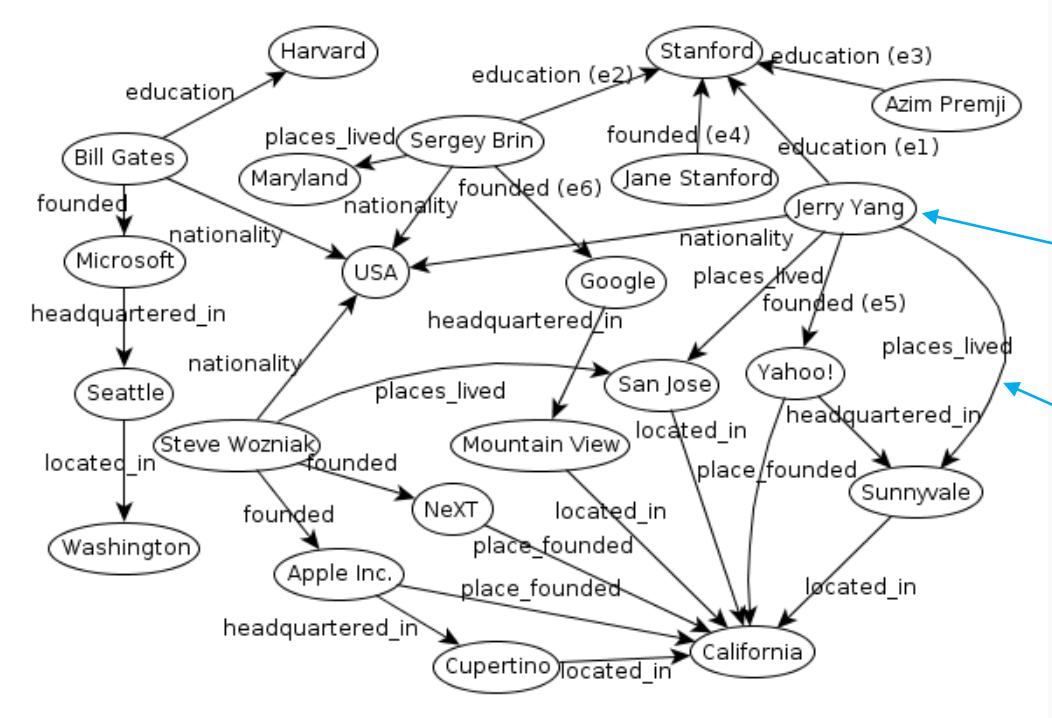
VLDB 2015 Phd Workshop
August 31st 2015

Outline

- Motivation: Graph Data Usability
- Visual Interface for Recommendation Based Interactive Graph Query Formulation (Orion)
- Graph Query By Example (GQBE)

Large Heterogeneous Graphs

Large, complex and schema-less graphs capturing millions of entities and relationships between them!



Entity

Relationship

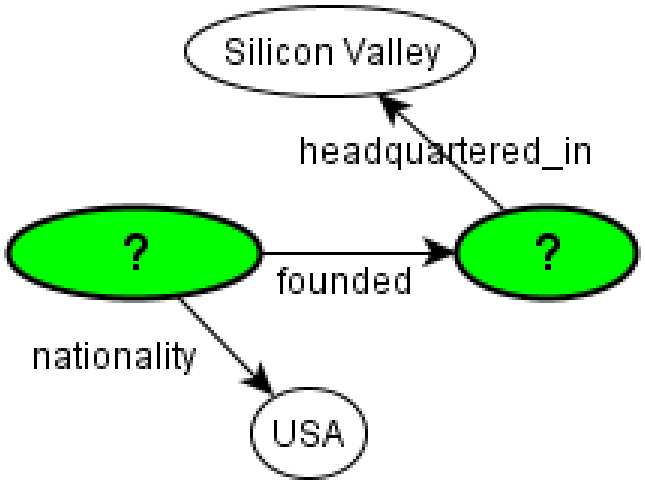
Linking Open Data : 52 billion RDF triples

Freebase : 1.8 billion triples

DBpedia : 470 million triples

Yago : 120 million triples

Specifying Queries for Graphs



SQL QUERY:

```
SELECT  Founder.subj, Founder.obj
FROM Founder, Nationality, HeadquarteredIn
WHERE
    Founder.property = 'founded' AND
    Founder.subj = Nationality.subj AND
    Nationality.property = 'nationality' AND
    Founder.obj = HeadquarteredIn.subj AND
    HeadquarteredIn.property = 'headquartered_in';
```

SPARQL QUERY:

```
SELECT  ?company  ?founder  WHERE {
    :?founder  dbo:founded  :?company .
    :?founder  dbo:nationality  :USA .
    :?company  dbprop:headquartered_in  :Silicon Valley .
}
```

Simpler Querying Paradigms

- Keyword Search
  - Keyword search in Graphs [Kargar, VLDB'11], BLINKS [He, SIGMOD'07]
    - Limitation: Articulating keyword query for graphs is not simple
- Approximate Query Specification and Answering
  - NESS: uses neighborhood-based indexes to quickly find approximate matches to a query graph [Khan, SIGMOD'11]
  - TALE: approximate large graph matching [Tian, ICDE'08]
    - Limitation: Users still have to formulate the initial query graph

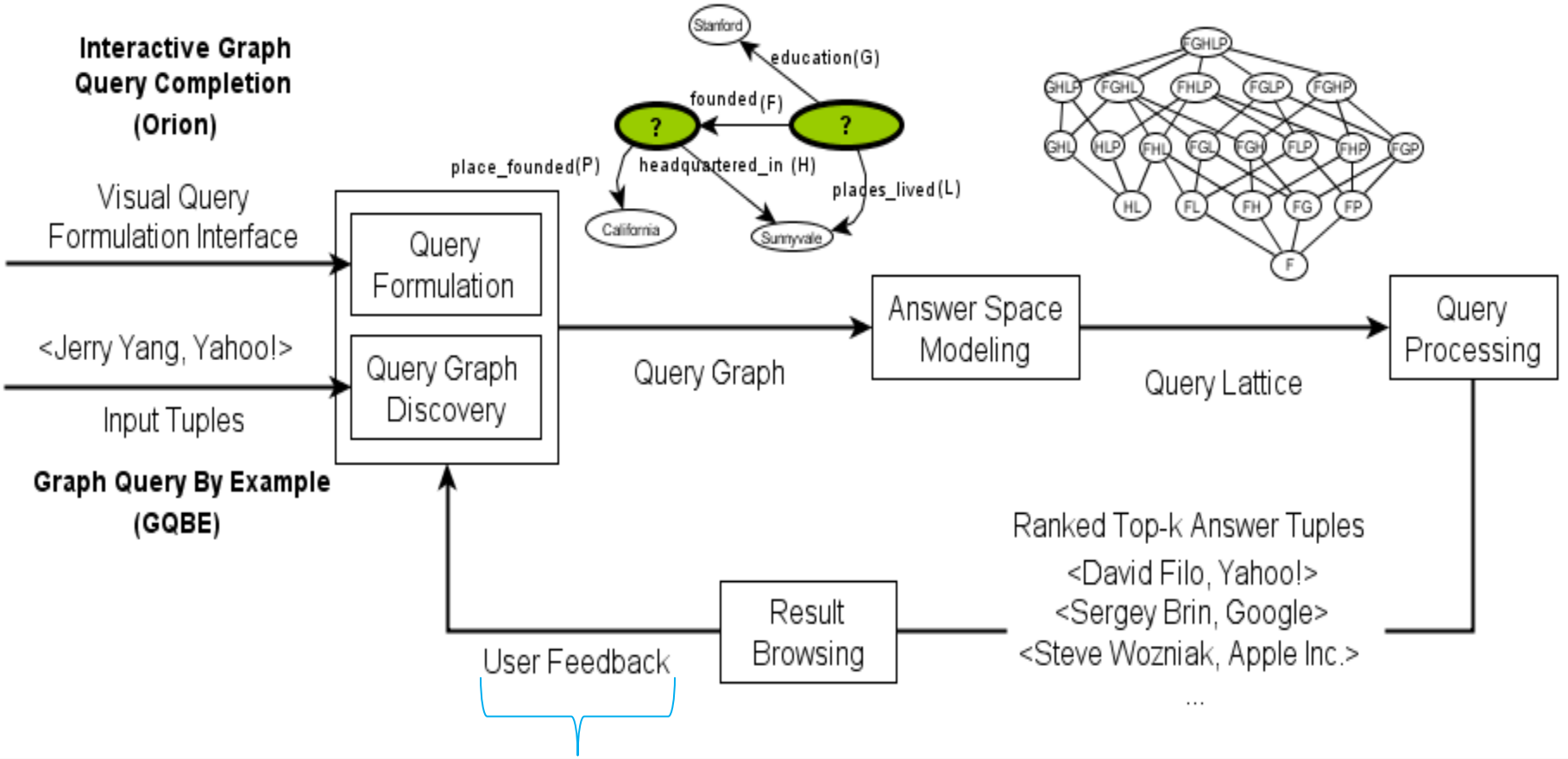
Visual Query Formulation Systems

- Relational Databases
  - CLIDE [Petropoulos, SIGMOD'06,07]
- Graph Databases
  - VOGUE, PRAGUE, Gblender, [Bhowmick, CIDR'13, ICDE'12, SIGMOD'11], GRAPHITE [Chau, ICDMW'08]
- Single Large Graphs
  - QUBLE [Bhowmick, VLDB'14]
- Limitations:
  - New relevant query components are not automatically recommended to users
  - Users require a good knowledge of the underlying schema

Desiderata of a User Friendly Query System

- Usability
  - An easy-to-use graphical interface for formulating query graphs
  - Easier paradigm to query complex heterogeneous graphs
- Ability to express exact query intent
  - Schema agnostic users assisted by an intelligent query system

Dissertation Research Outline



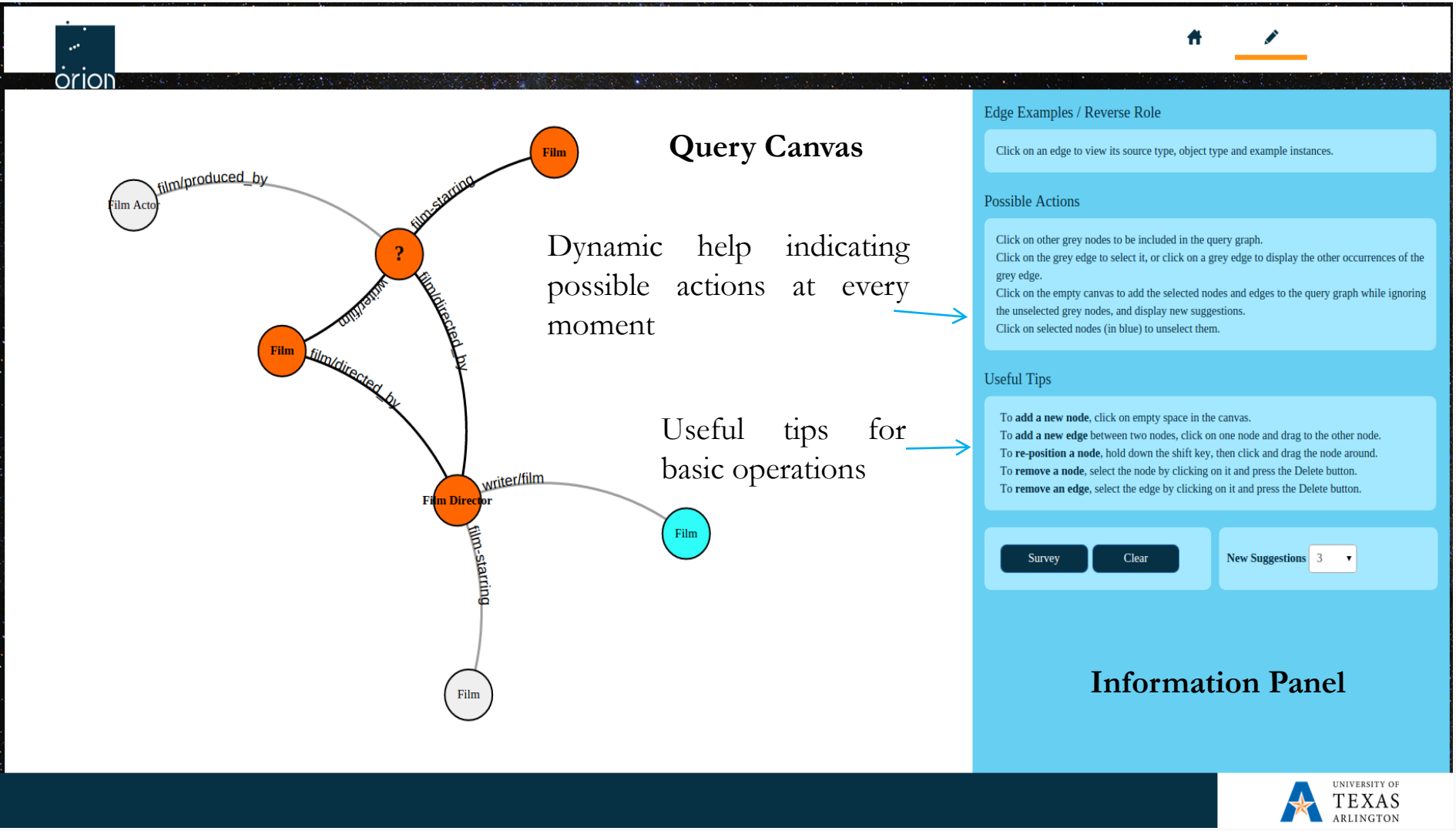
Visual Interface for Recommendation Based Interactive Query Formulation (Orion)

Ongoing work

Problem Statement

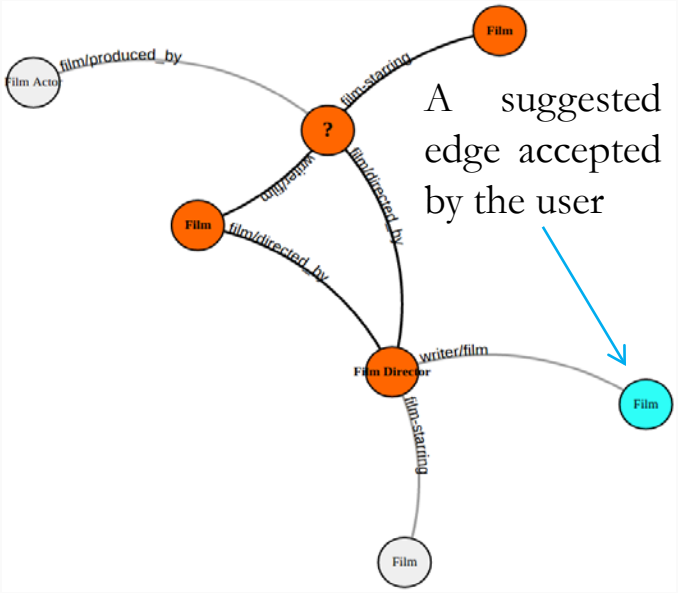
- Given a large heterogeneous graph, iteratively suggest edges to help build a query graph
  - An interactive graphical user interface for building query components
  - An edge recommendation system that ranks edges based on their relevance to the user's query intent

Orion Interface (idir.uta.edu/orion)



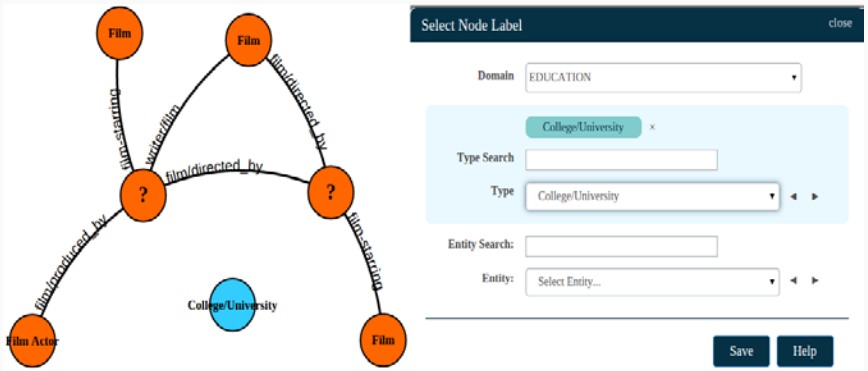
Modes of Operation: Passive and Active

Grey edges and nodes automatically suggested in **passive mode**

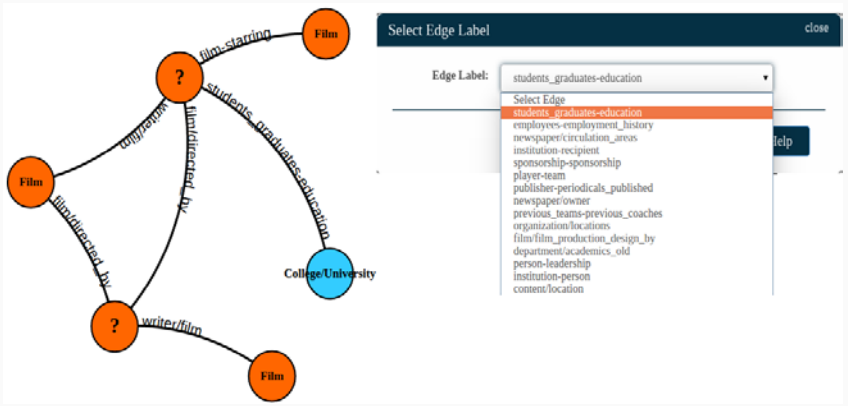


Suggested edges accepted by the user (with blue node) are **positive edges**. Grey edges ignored are **negative edges**.

A new node added in **active mode**



A new edge added in **active mode**



Preliminaries

Edges in partial query graph (positive edges)

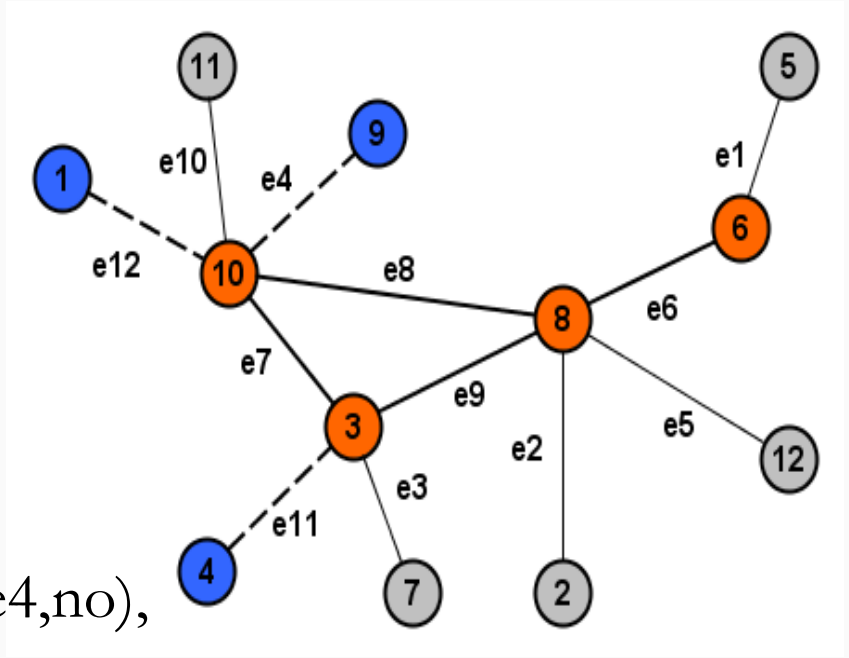
e6, e7, e8, e9

Edges rejected by users (negative edges)

e4, e11, e12

Candidate edges

e1, e2, e3, e5, e10



Query Session:

<(e6,yes), (e7,yes), (e8,yes), (e9,yes), (e4,no), (e11,no), (e12,no)>

represented as

(e6, e7, e8, e9, -e4, -e11, -e12)

Query Log

- Collection of several user sessions

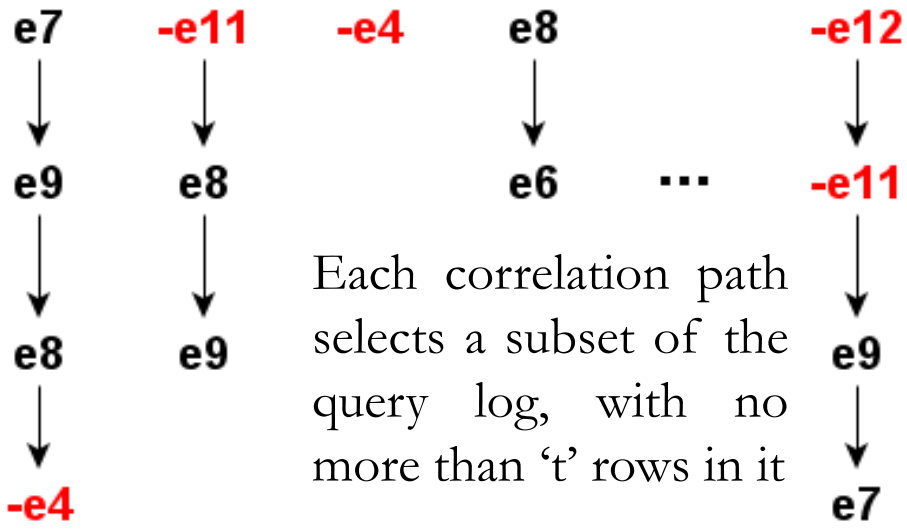
Session Id	Co-related Edges
w_1	(*education*, yes), (*founder*, yes), (*nationality*, no)
w_2	(*starring*, yes), (*music*, no), (*director*, yes)
w_3	(*nationality*, yes), (*education*, no), (*music*, yes), (*starring*, no)
w_4	(*artist*, no), (*title*, no), (*writer*, yes), (*director*, yes)
w_5	(*director*, no), (*founder*, yes), (*producer*, yes)
w_6	(*writer*, yes), (*editor*, no), (*genre*, yes)
w_7	(*award*, no), (*movie*, yes), (*director*, yes), (*genre*, no)
w_8	(*education*, yes), (*founder*, yes), (*nationality*, no)

Algorithms to Rank Candidate Edges

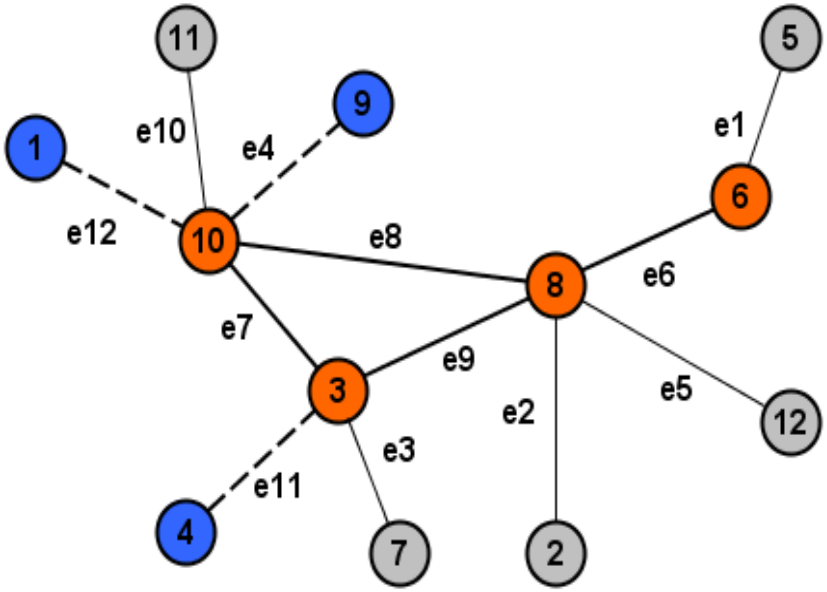
- Possible Solutions
  - Order alphabetically
  - Use standard machine learning methods
    - Recommendation system
    - Association rule mining based classification
    - Classification: naïve Bayesian classifier, random forests
- Query-specific random correlation paths based suggestion

Random Correlation Paths (RCPs) Based Ranking

- Choose edges from the query session randomly to form RCPs:



Each correlation path selects a subset of the query log, with no more than 't' rows in it



- Grow a path incrementally until its support in the query log drops below a threshold (t).
- For each RCP, use its corresponding query log subset to compute support for each candidate edge.

Session Id	Co-related Edges
w_1	(*education*, yes), (*founder*, yes), (*nationality*, no)
w_2	(*starring*, yes), (*music*, no), (*director*, yes)
w_3	(*nationality*, yes), (*education*, no), (*music*, yes), (*starring*, no)
w_4	(*artist*, no), (*title*, no), (*writer*, yes), (*director*, yes)
w_5	(*director*, no), (*founder*, yes), (*producer*, yes)
w_6	(*writer*, yes), (*editor*, no), (*genre*, yes)
w_7	(*award*, no), (*movie*, yes), (*director*, yes), (*genre*, no)
w_8	(*education*, yes), (*founder*, yes), (*nationality*, no)

Final score of each candidate is its average score across all RCPs.

Preliminary Results

Target Query Graphs		Edge Ranking Algorithms			
Query Graph	**# of edges**	**RCP**	**RCP (no negative edges)**	**Random Forest Classifier**	**Random**
ForrestGump-directorType	3	12	**11**	>100	37
FilmType-directorType	5	**39**	>100	41	>100
DirectorType-actorType	3	>100	>100	>100	>100
FilmType-DirectorType	4	**28**	>100	31	>100
FilmType-DirectorType	3	**14**	27	25	>100
FounderType-SchoolType	5	34	>100	**33**	>100
FounderType-SchoolType	4	>100	>100	>100	>100
JerryYang-SchoolType	5	**34**	85	>100	>100
JerryYang-Yahoo-Stanford	4	**14**	>100	33	>100

Evaluation Plan for Orion

- Compare with other standard machine learning algorithms
- User studies to gauge the effectiveness of our system and compare with naïve approaches like listing suggestions alphabetically
- Study effectiveness (number of suggestions required) using several simulated target query graphs
- Experiments with other datasets (DBpedia, YAGO)

Publication

- VIIQ: Auto-suggestion Enabled Visual Interface for Interactive Query Formulation, Nandish Jayaram, Sidharth Goyal, Chengkai Li, VLDB 2015, Demonstration description

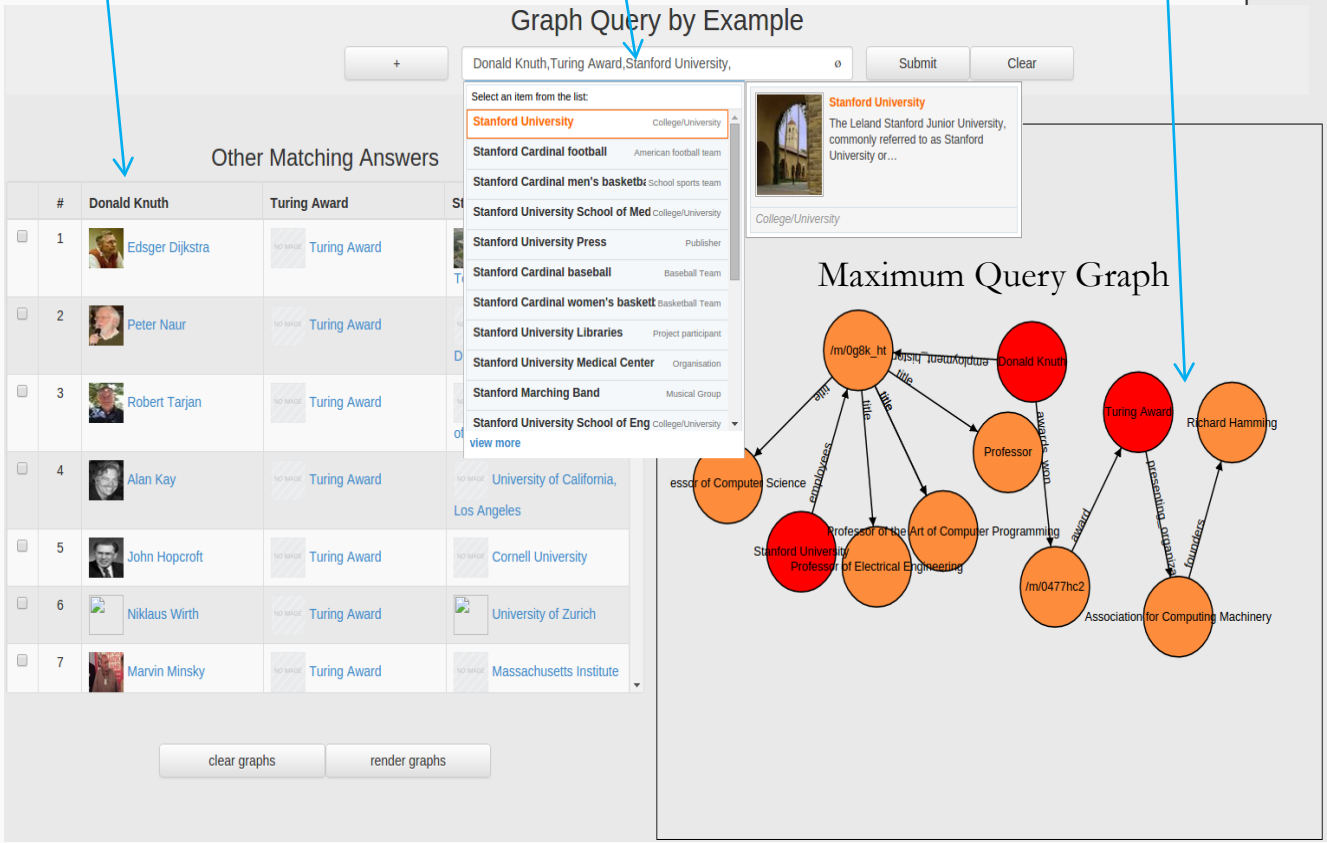
Graph Query By Example (GQBE)

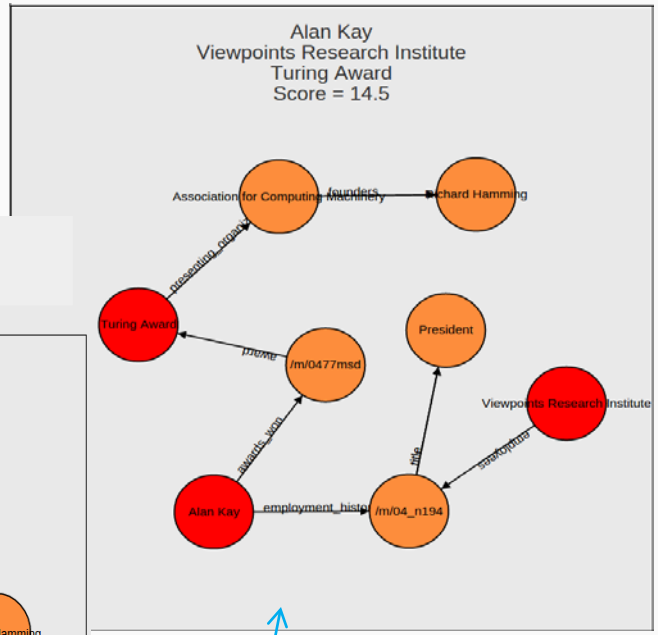
GQBE Interface (idir.uta.edu/gqbe)

Ranked similar answer tuples

Keyword completion powered query interface

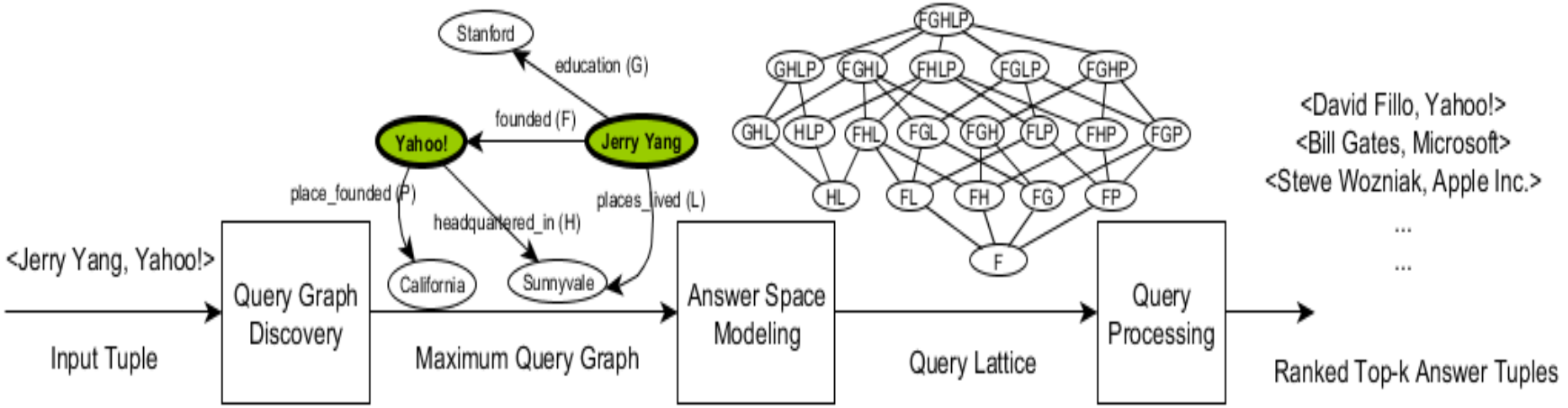
Query graph automatically discovered by the system





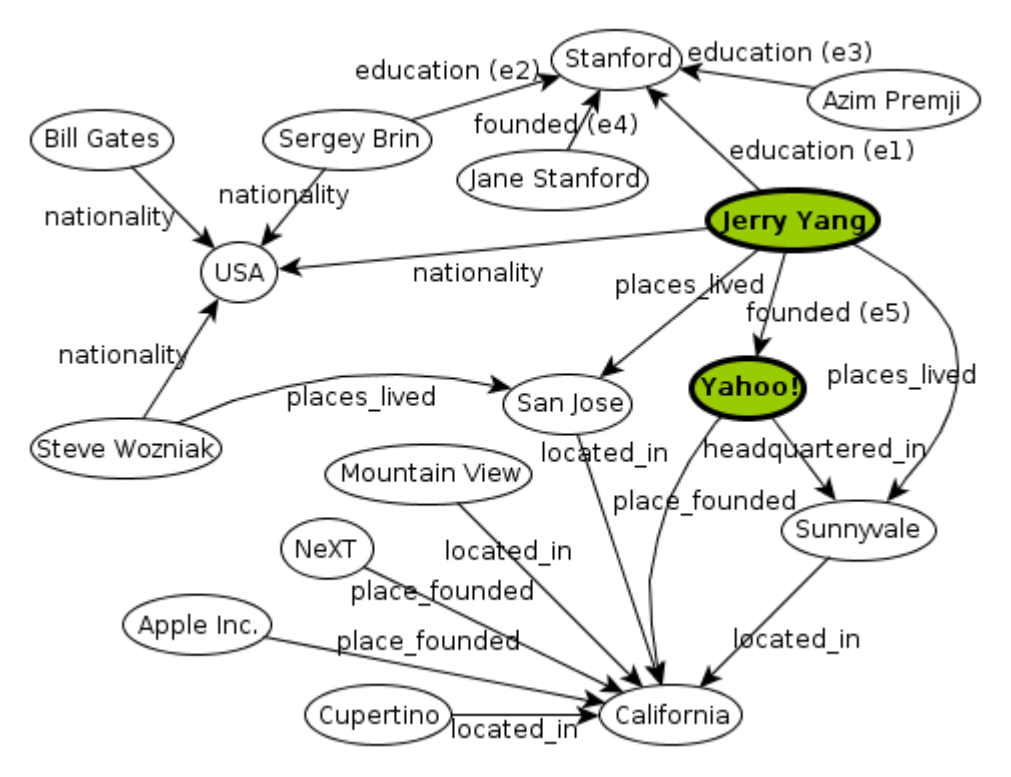
An example answer graph

Challenges

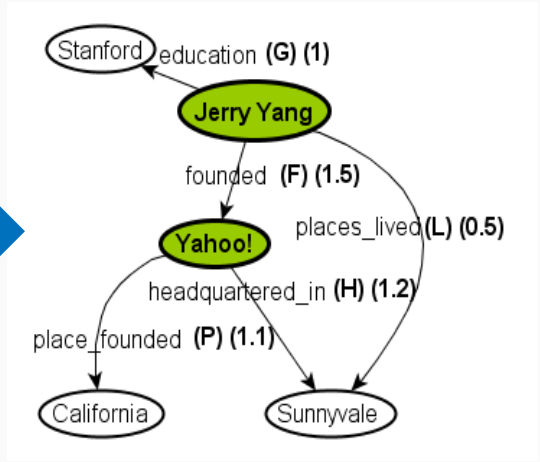

<Jerry Yang, Yahoo!>
Input Tuple
Query Graph Discovery
Stanford
education (G)
founded (F)
Yahoo!
Jerry Yang
place_founded (P)
headquartered_in (H)
places_lived (L)
California
Sunnyvale
Maximum Query Graph
Answer Space Modeling
FGHLP
GHLP
FGHL
FHLP
FGLP
FGHP
GHL
HLP
FHL
FGL
FGH
FLP
FHP
FGP
HL
FL
FH
FG
FP
F
Query Lattice
Query Processing
<David Fillo, Yahoo!>
<Bill Gates, Microsoft>
<Steve Wozniak, Apple Inc.>
...
...
Ranked Top-k Answer Tuples

Query Graph Discovery

Neighborhood Graph

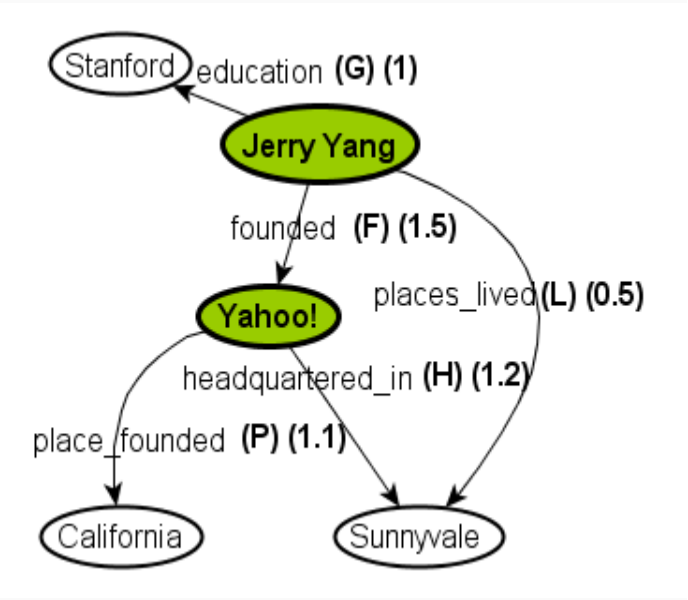


Query Graph

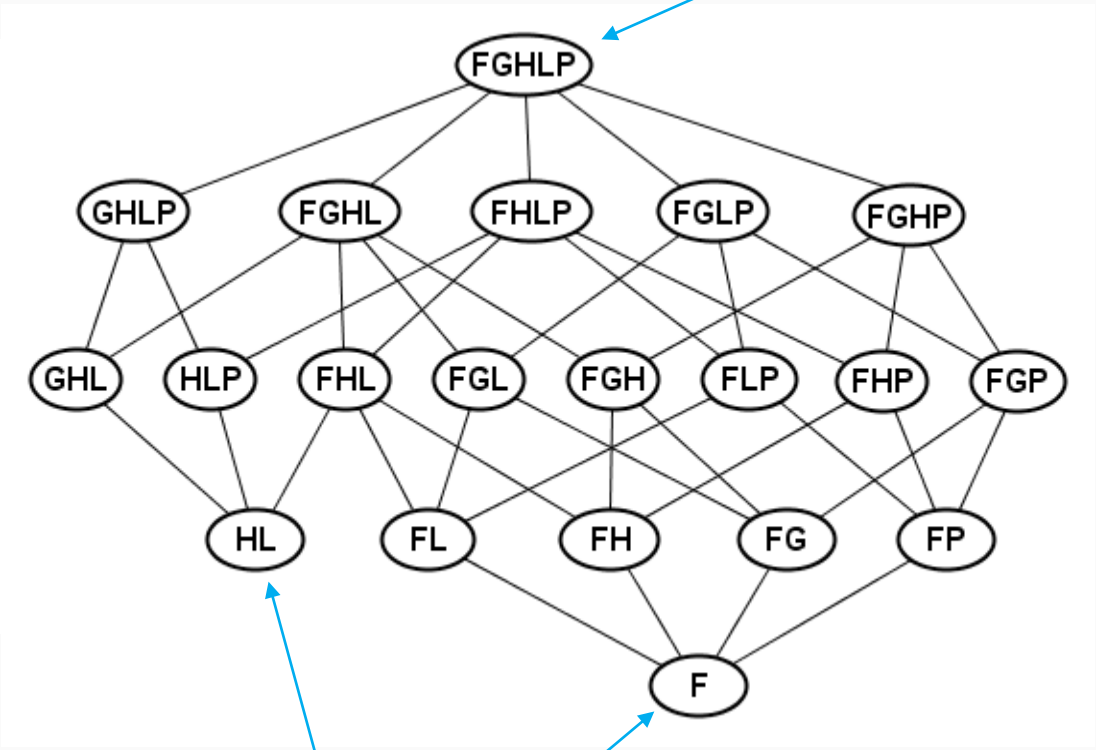


Query Processing

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



Maximum Query Graph (MQG)

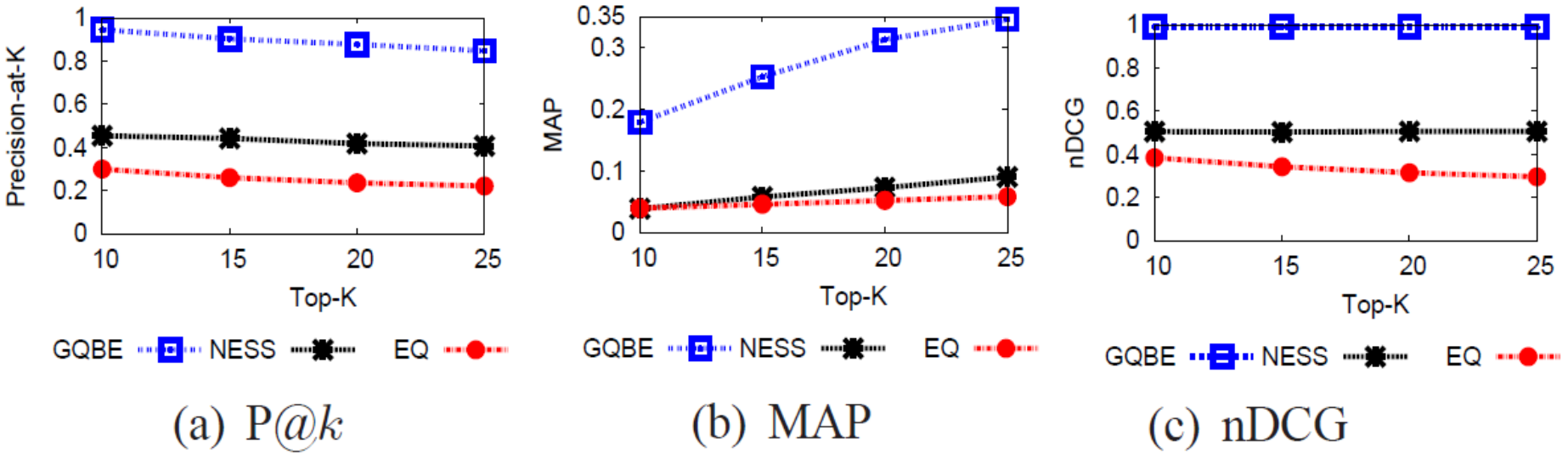


Minimal Query Trees

Experiments: Accuracy Comparison with NESS and EQ

Dataset:

Freebase (47 million edges, 27 million nodes, 5.4 K edge labels)



Experiments: User Study with Amazon MTurk

Query	PCC	Query	PCC	Query	PCC	Query	PCC
F_1	0.79	F_2	0.78	F_3	0.60	F_4	0.80
F_5	0.34	F_6	0.27	F_7	0.06	F_8	0.26
F_9	0.33	F_{10}	0.77	F_{11}	0.58	F_{12}	undefined
F_{13}	undefined	F_{14}	0.62	F_{15}	0.43	F_{16}	0.29
F_{17}	0.64	F_{18}	0.30	F_{19}	0.40	F_{20}	0.65

PEARSON CORRELATION COEFFICIENT (PCC) BETWEEN GQBE AND AMAZON MTURK WORKERS, k=30

[0.5, 1.0] : Strong positive correlation
[0.3, 0.5) : Medium positive correlation
[0.1, 0.3) : Small positive correlation

Publications

- Querying Knowledge Graphs by Example Entity Tuples, Nandish Jayaram, Arijit Khan, Chengkai Li, Xifeng Yan, Ramez Elmasri, TKDE (to appear)
- GQBE: Querying Knowledge Graphs by Example Entity Tuples, Nandish Jayaram, Mahesh Gupta, Arijit Khan, Chengkai Li, Xifeng Yan, Ramez Elmasri, ICDE' 14, Demonstration description
- Towards a Query-by-Example System for Knowledge Graphs, Nandish Jayaram, Arijit Khan, Chengkai Li, Xifeng Yan, Ramez Elmasri, GRADES' 14

Orion Demonstration at VLDB 2015

- **Demo Session 3 (Kona 4)**
- VIIQ: Auto-Suggestion Enabled Visual Interface for Interactive Graph Query Formulation

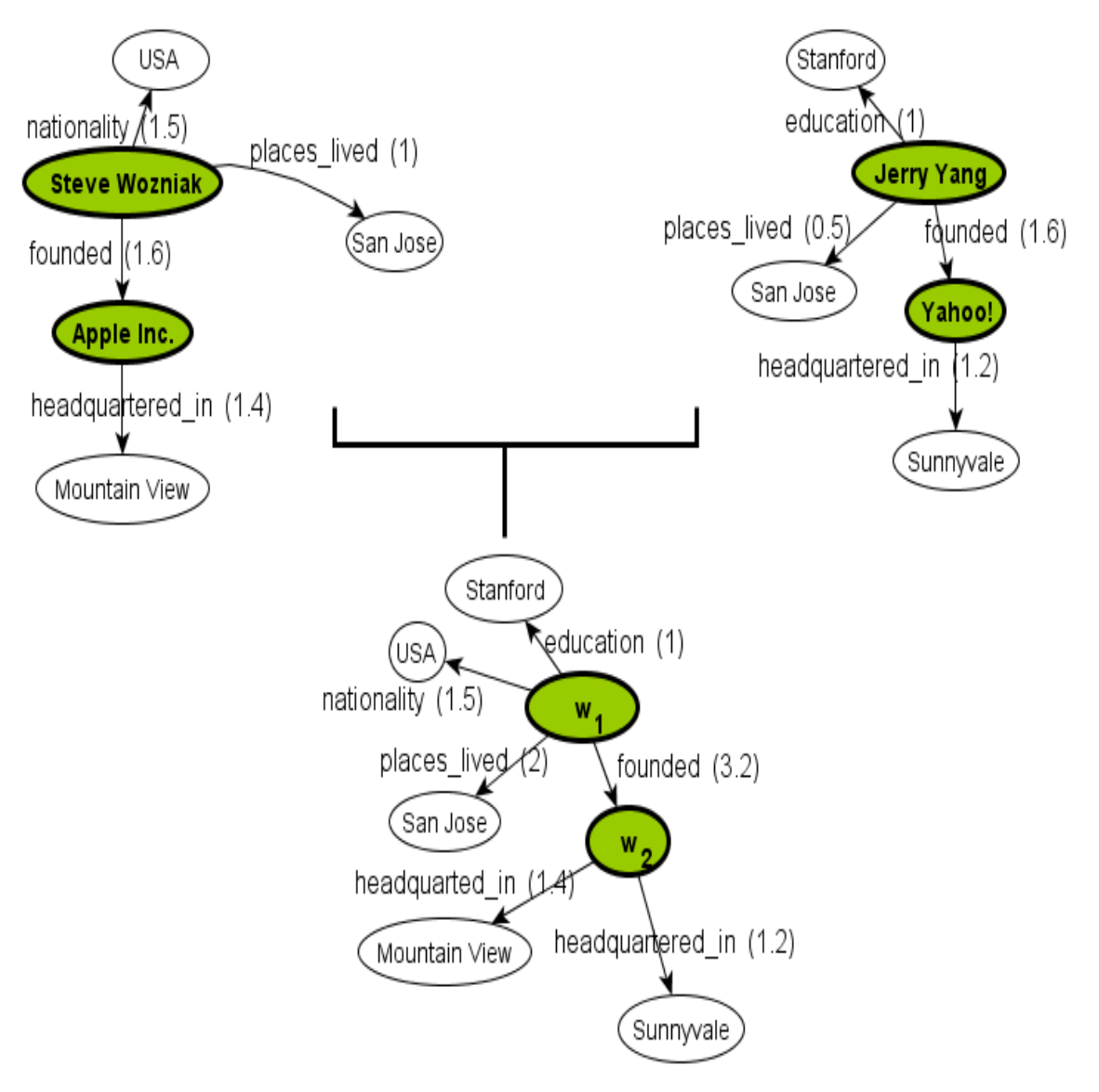
September 3rd, Wednesday (10:30 am to 12:00 pm)

September 4th, Thursday (3:30 pm to 5:00 pm)

Thank You!

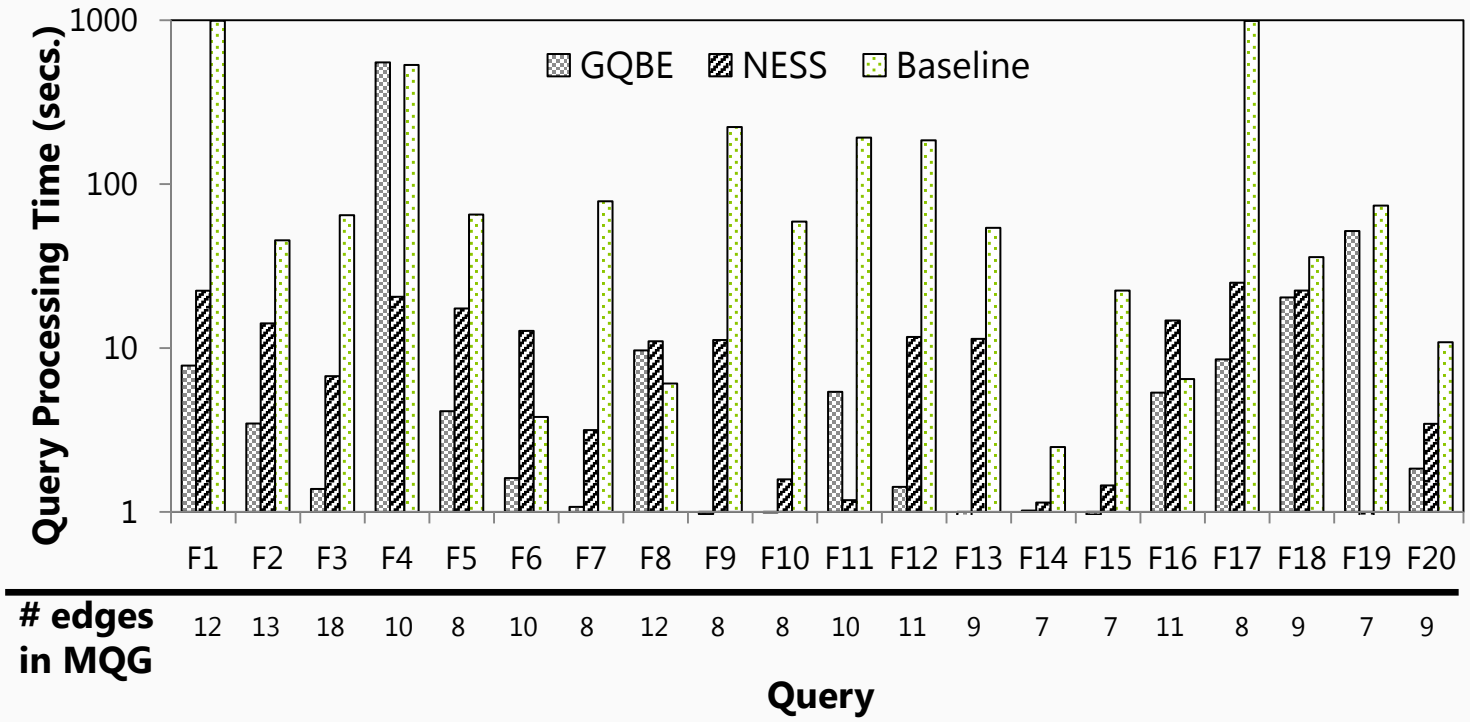
nandish.jayaram@mavs.uta.edu
https://sites.google.com/site/jnandish

Multiple Example Tuples



Experiments: Efficiency Results

Single Query Execution Times (in seconds)



Future Work

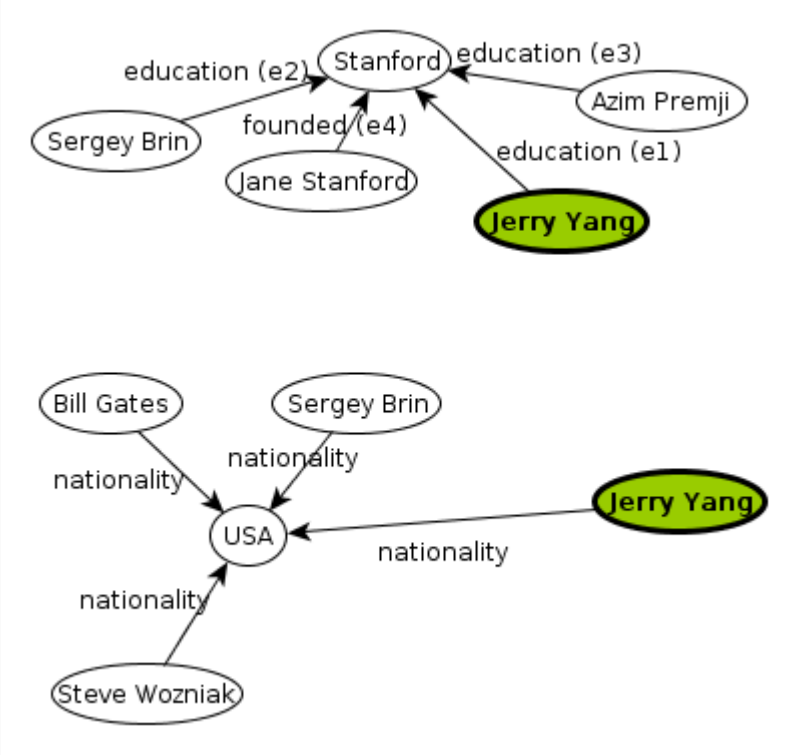
Future Work

- Comprehensive experiments and evaluation of Orion
- Evaluate the partial query graph at every iteration of the query formulation process in Orion
- User feedback loop after browsing the results

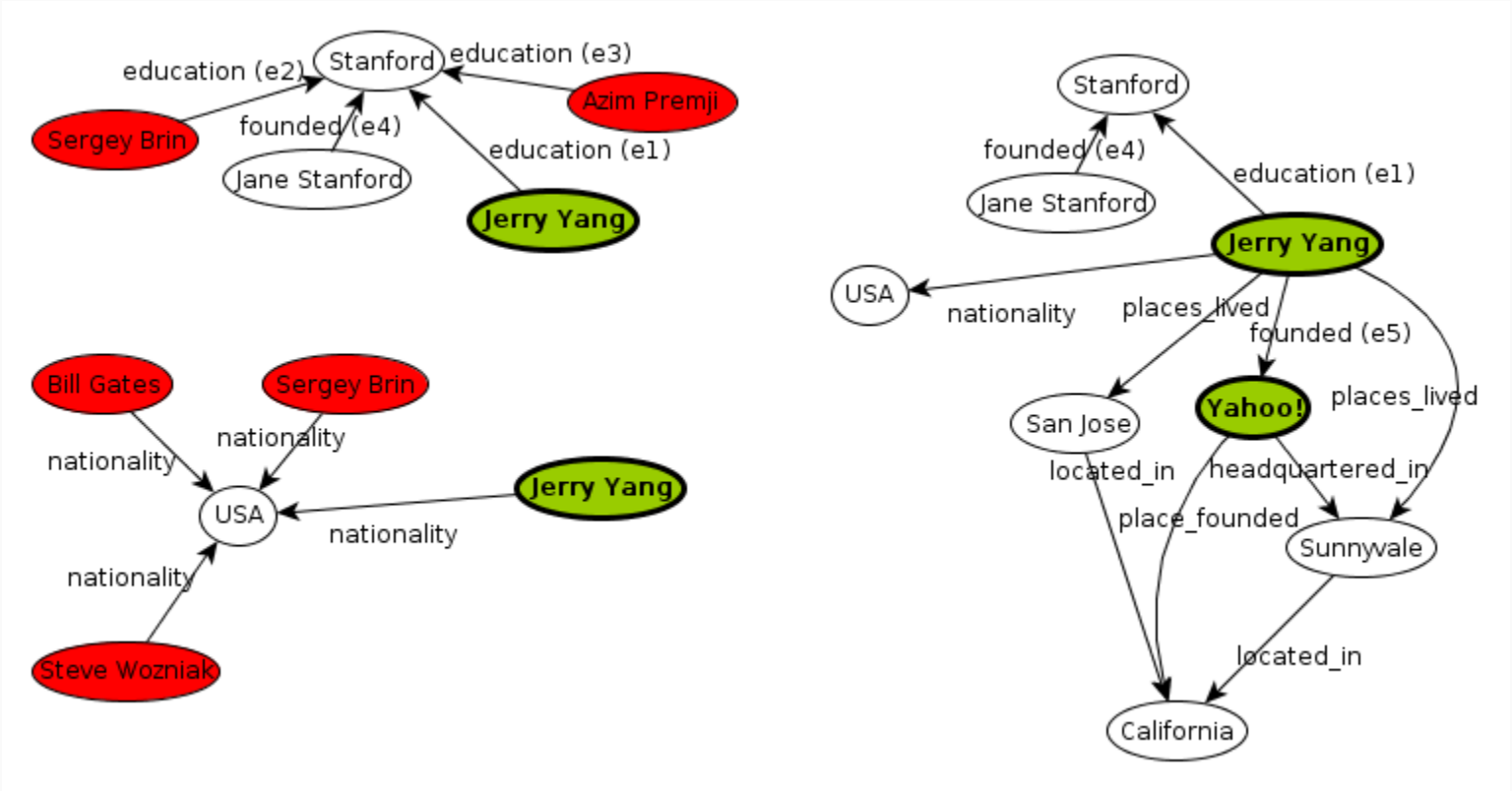
Cleaning Neighborhood Graph

- Neighborhood graphs can be large even for a small *d*; hundreds of thousands of edges and vertices!

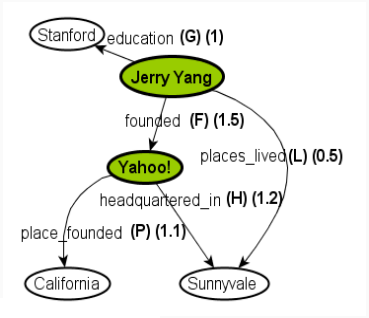
- Clean some clearly unimportant edges.

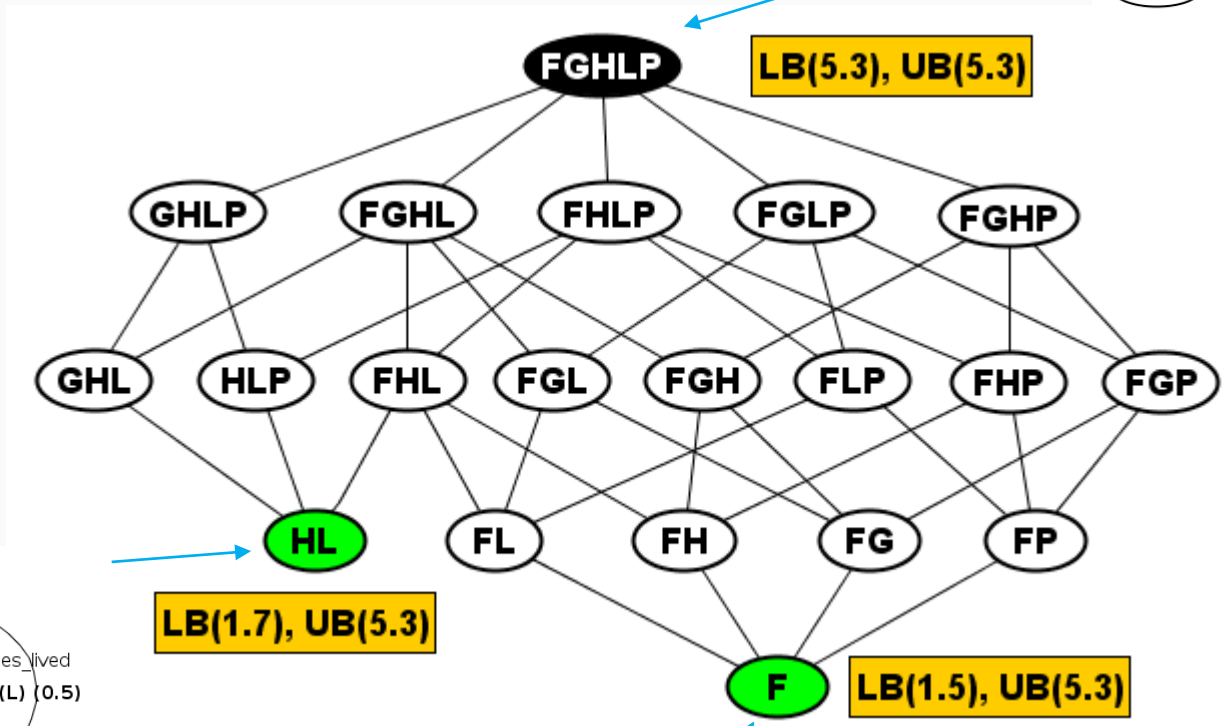


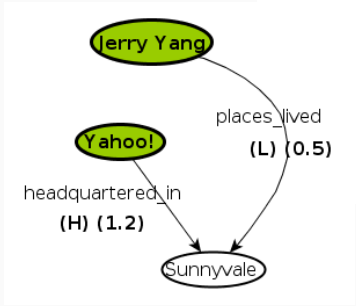
Reduced Neighborhood Graph

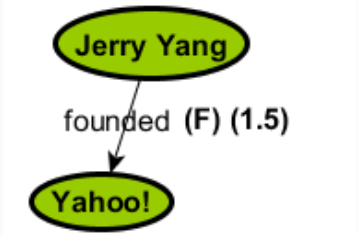


Query Processing

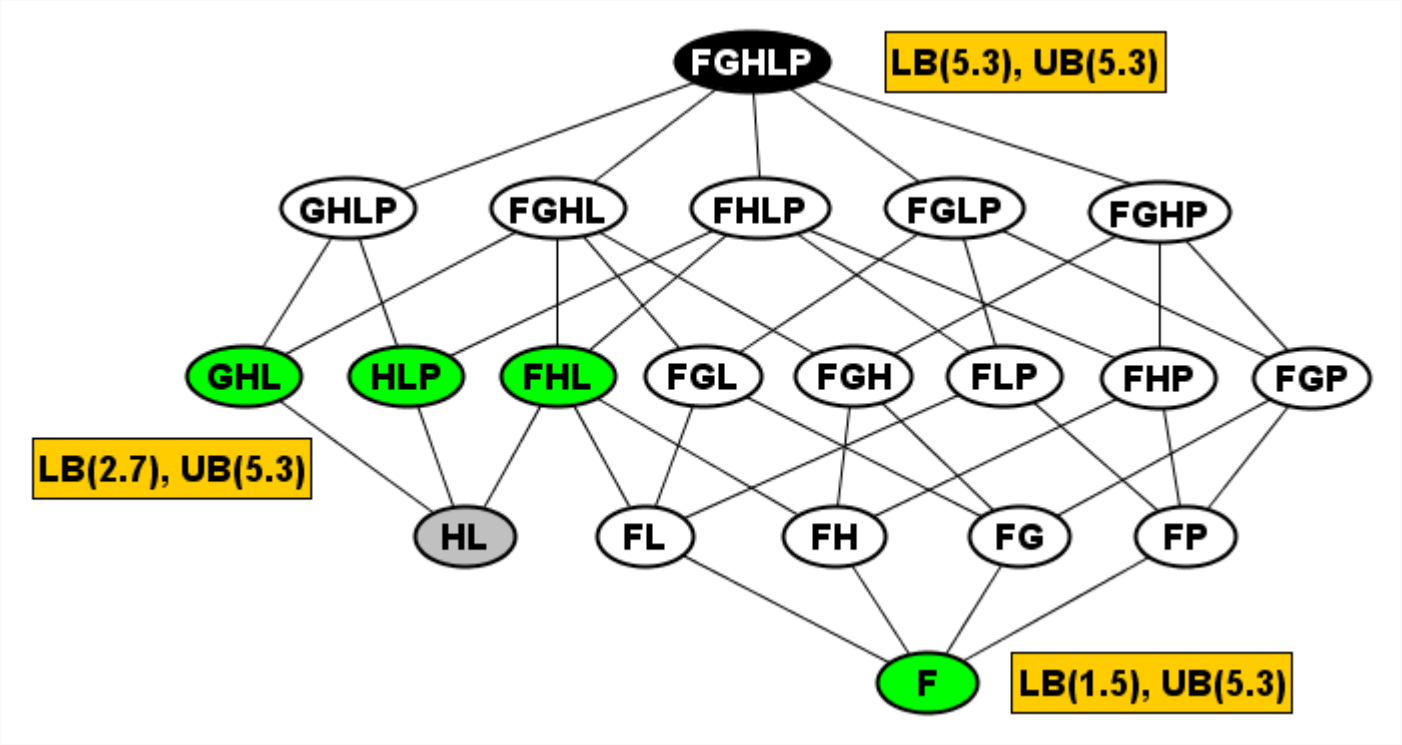




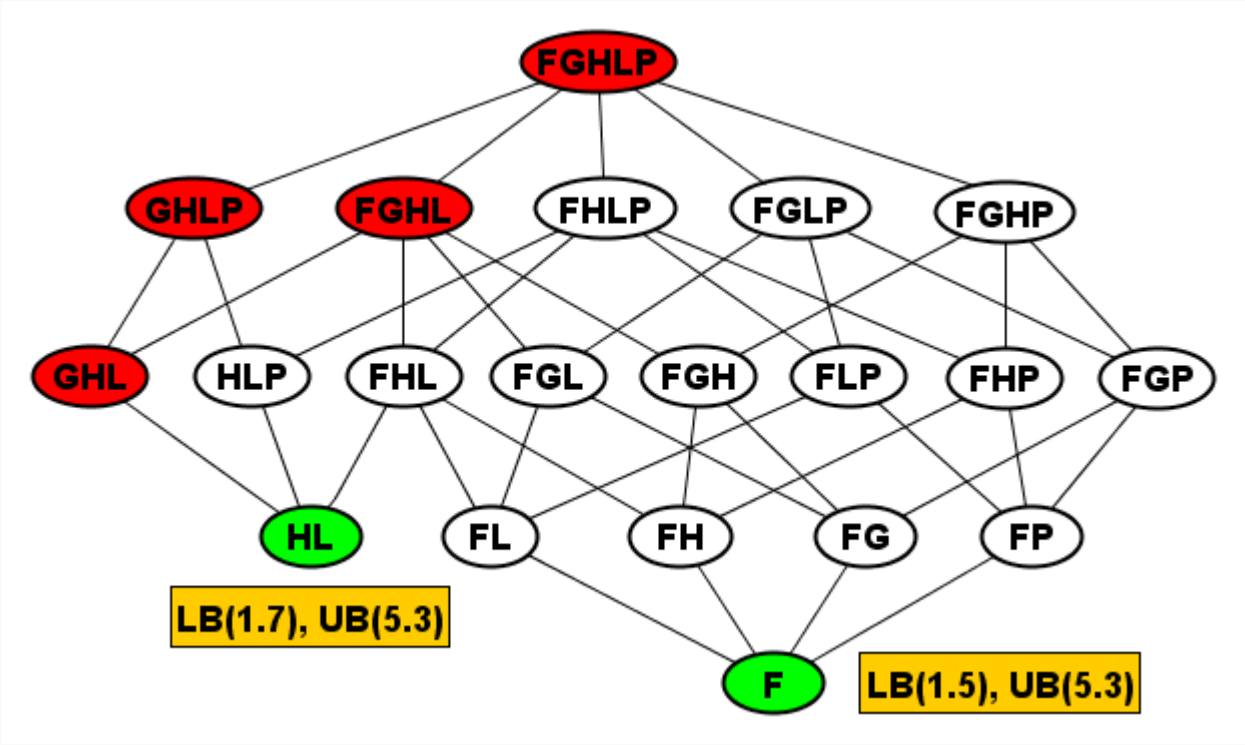




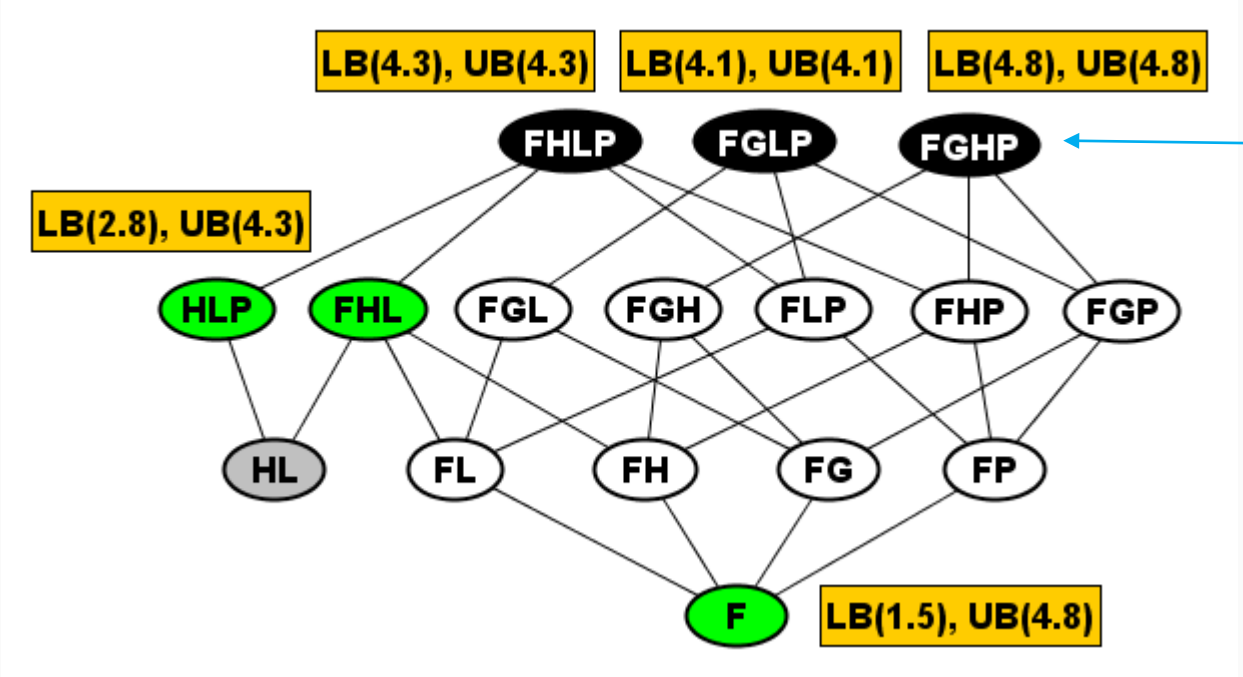
Query Processing (cont.)

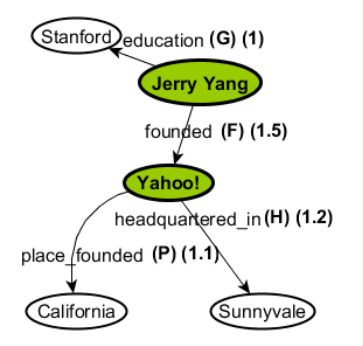


Query Processing (cont.)



Query Processing (cont.)





Evaluation Plan for Orion (cont.)

- Study effectiveness (number of suggestions required) using simulated target query graphs
- Experiments with other datasets (DBpedia, YAGO)
- Experiments to study effectiveness of simulated query log