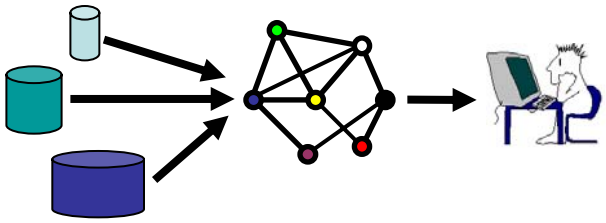
Applications and Algorithms for Semantic Graphs

Tina Eliassi-Rad
Lawrence Livermore National Laboratory

IPAM Graduate Summer School
July 11-29, 2005

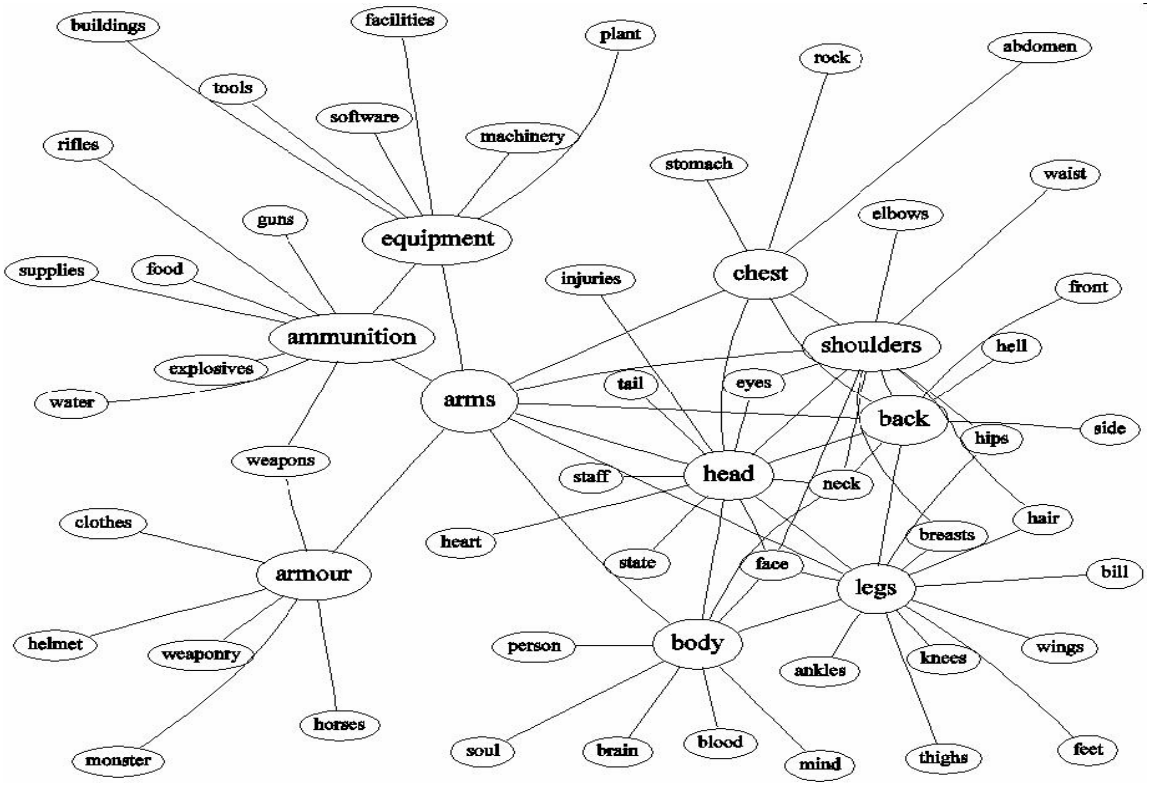
Motivation



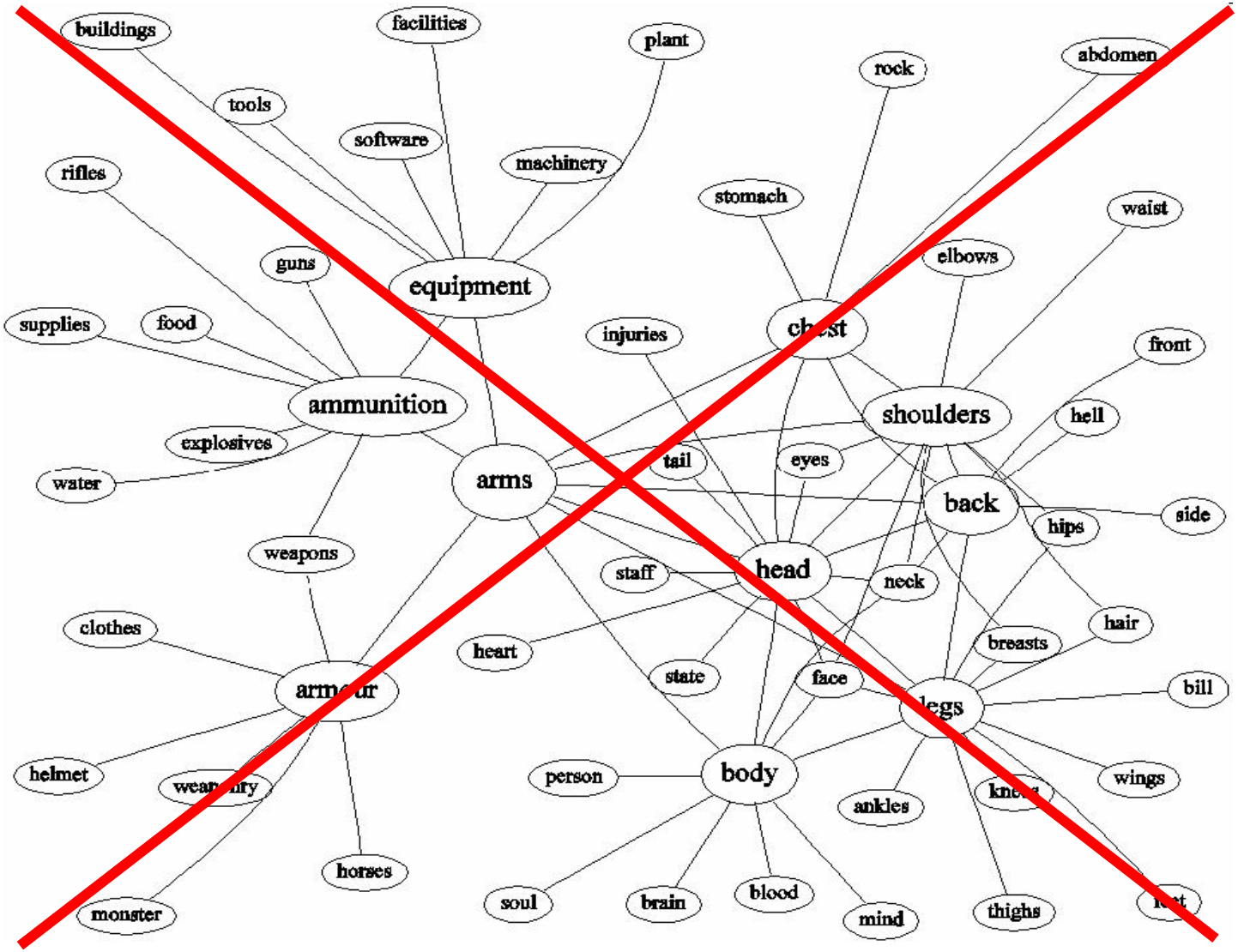
- Problem
  - Identify chains of relationships within a large collection of seemingly disjoint entities.
  - Data is collected from multiple (dynamic) sources.
- Challenge
  - Identify relationships and uncover patterns in a timely manner.
- Approach
  - Use *semantic graphs* to represent the data.
  - Exploit various techniques from graph theory, probability theory, information theory, social network analysis, informed search algorithms, statistical relational learning, parallel algorithms, …

Google query: semantic graph

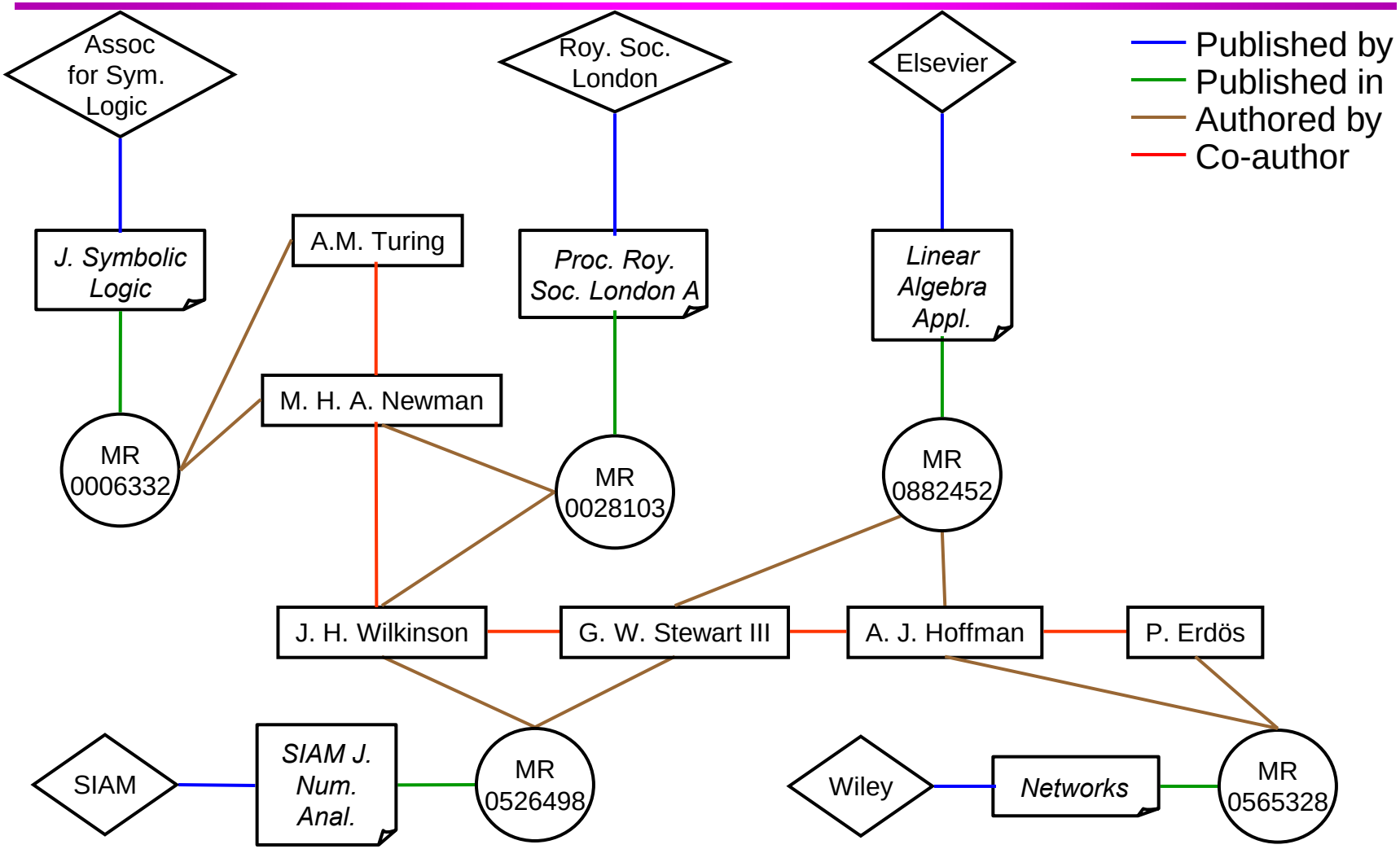
- Top link points to http://infomap.stanford.edu/graphs/
- “The Infomap Project is developing graph models representing words and their relationships.”
- “This work is important for developing automatic dictionaries.”



Our semantic graph is not this!

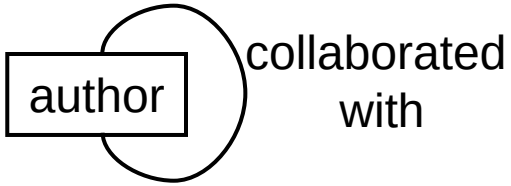


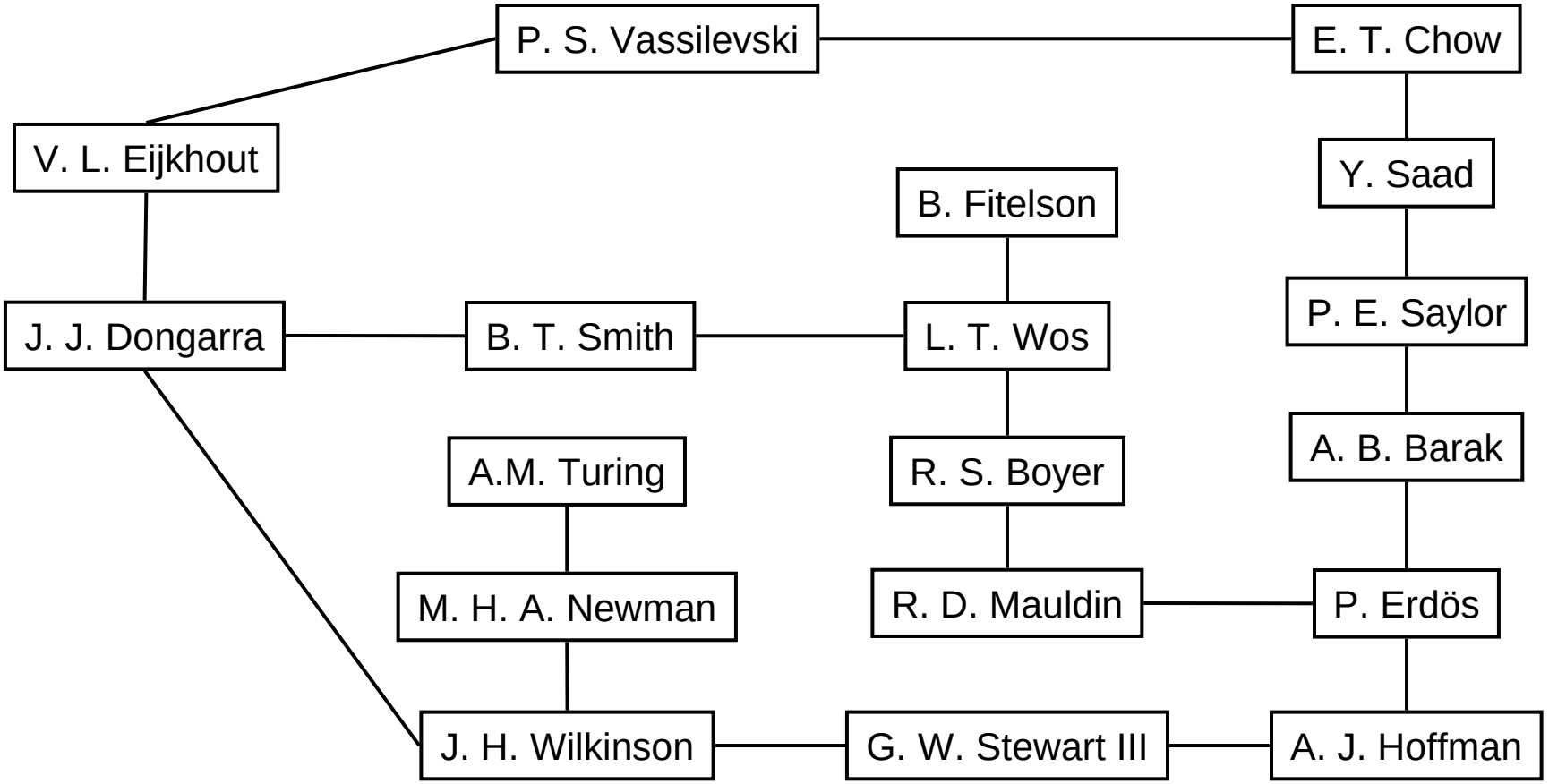
Our semantic graph is a *heterogeneous* complex network.



A section from MathSciNet's Collaboration Network at http://www.ams.org/mathscinet/

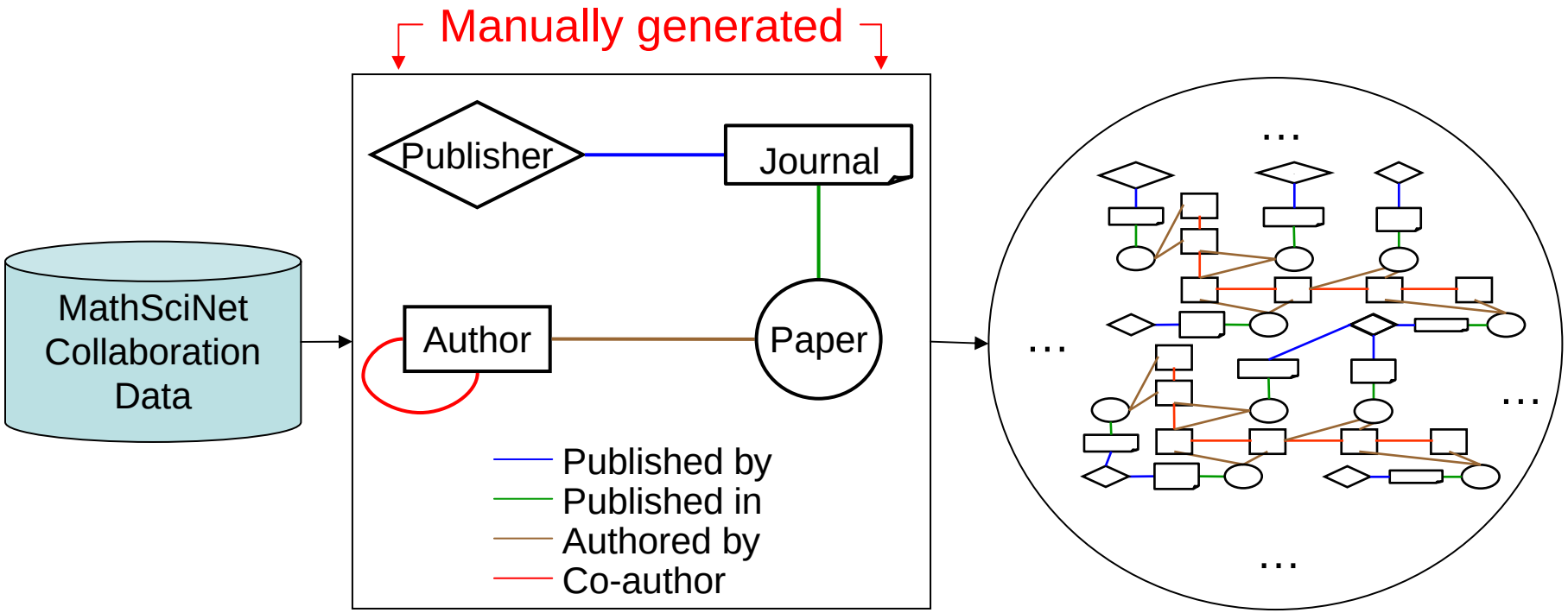
Social network is a *homogenous* complex network.





A section from MathSciNet's Collaboration Network at
http://www.ams.org/mathscinet/

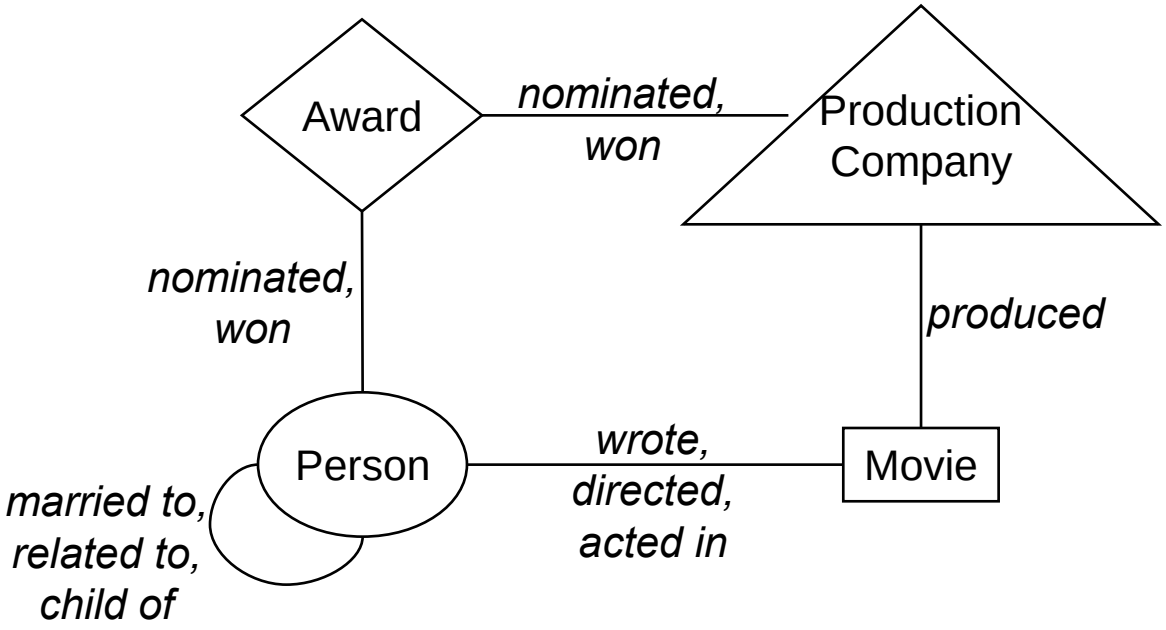
To generate a semantic graph, you need datasets and *human-defined* ontology graphs.



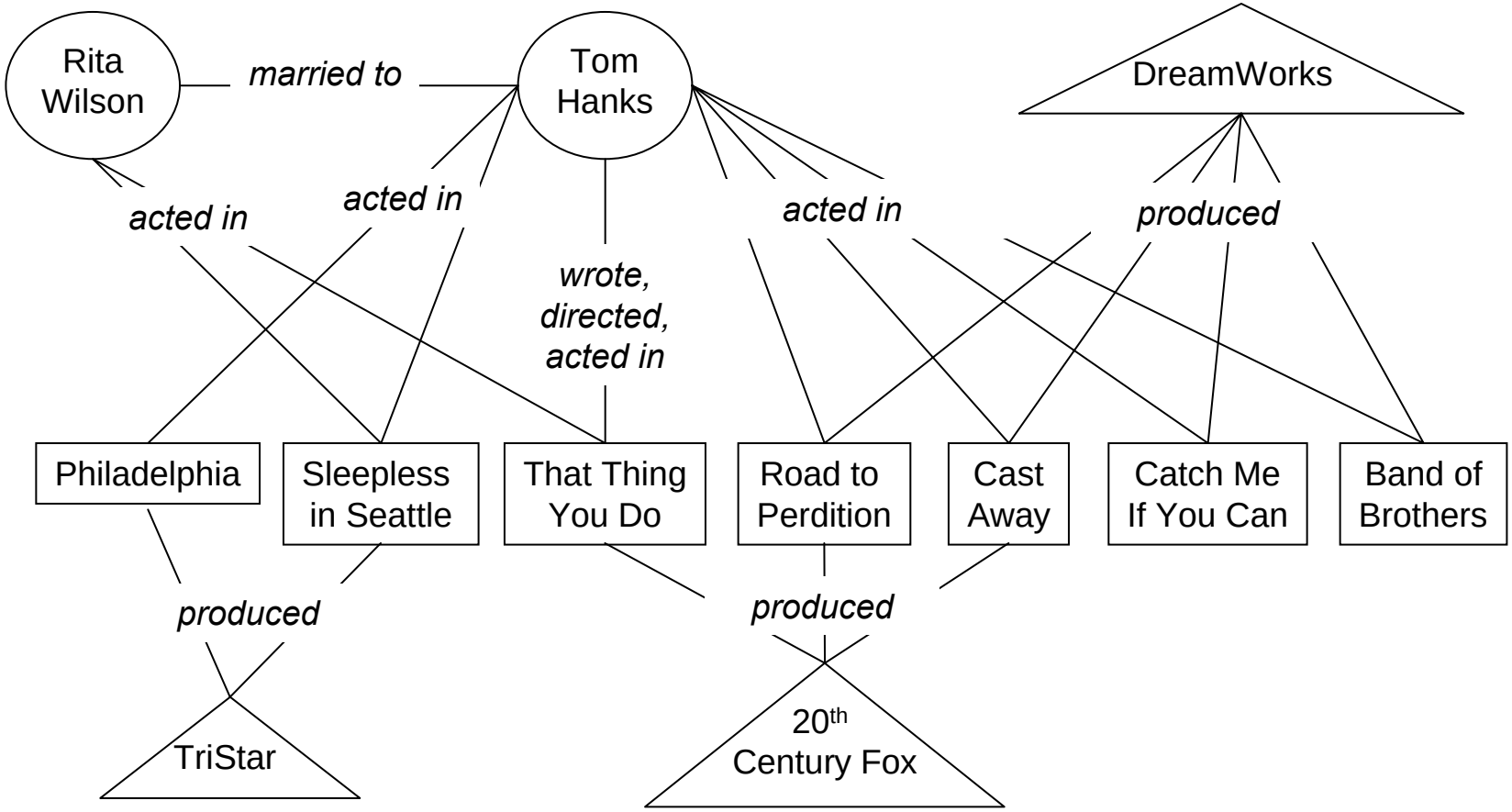
Ontology graph
Data schema
Type graph

Semantic graph
Attributed relational graph
Heterogeneous social network
Instance graph

A simple type graph for movies

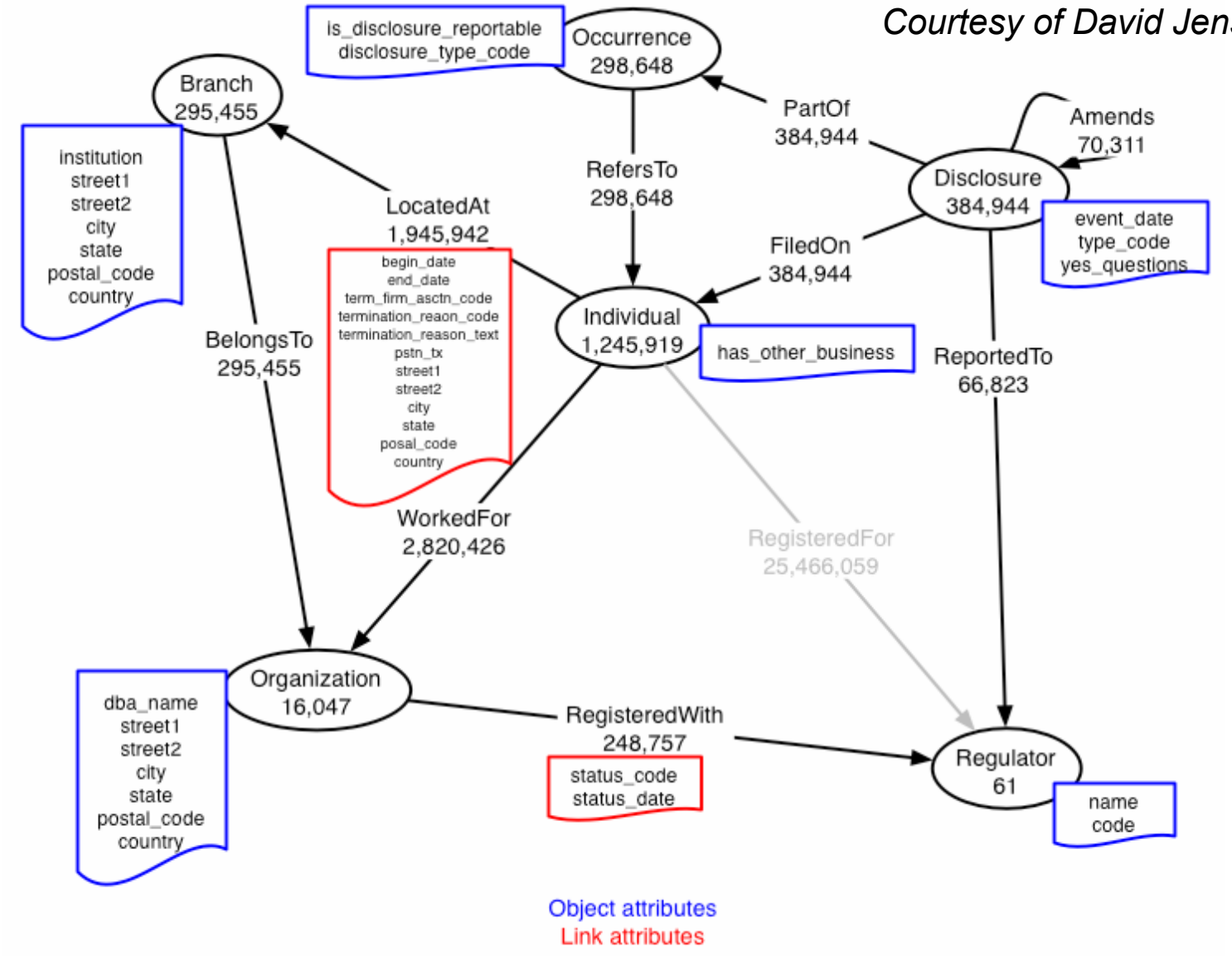


Instance graph: a section from the IMDB data at http://www.imdb.com

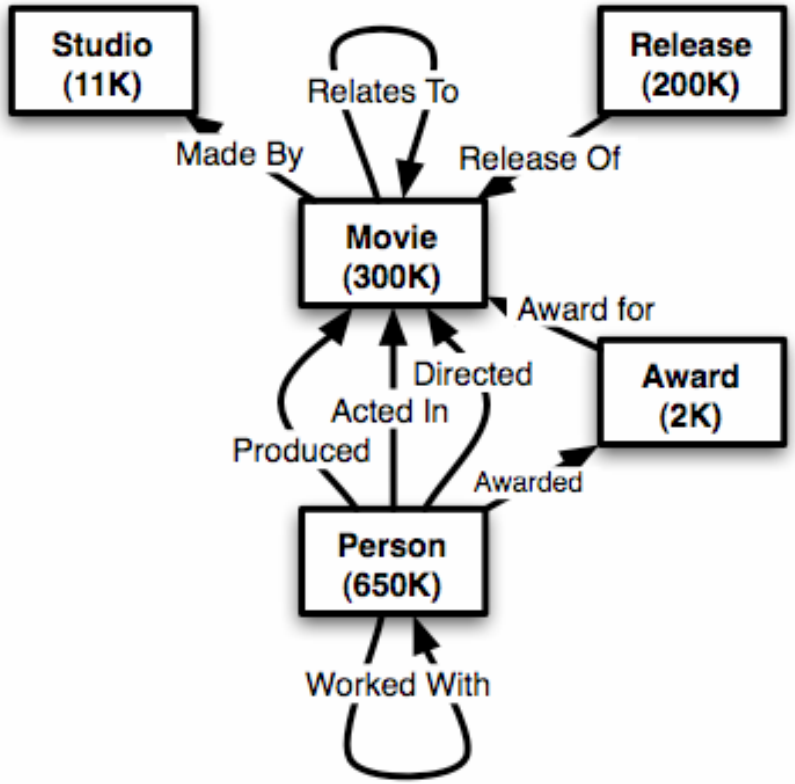


Type graph for National Association of Securities Dealers (NASD)

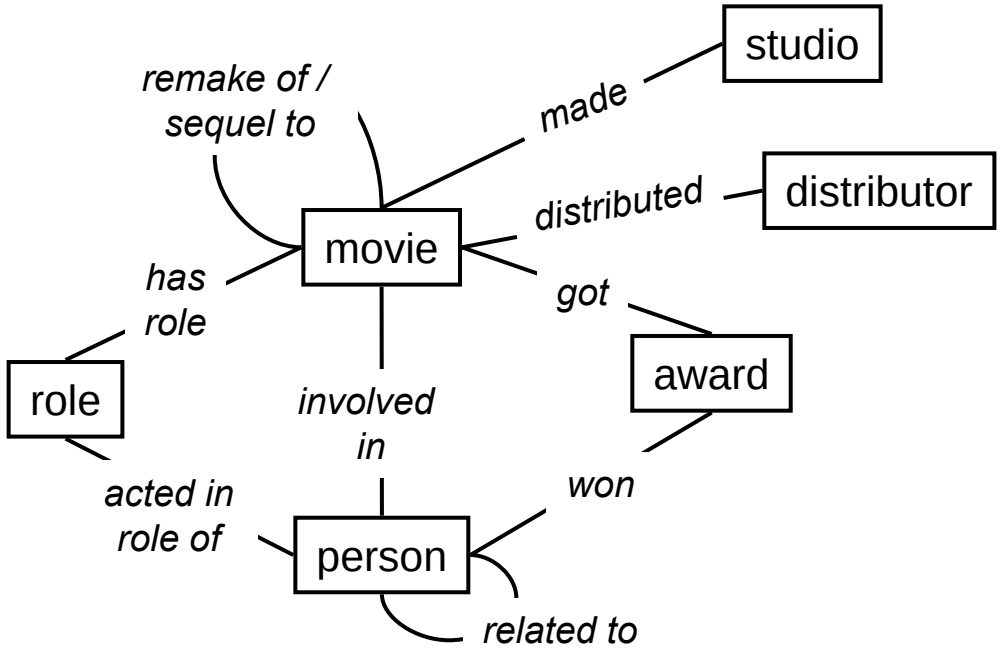
Courtesy of David Jensen, UMass



Sample type graphs for IMDB



Courtesy of David Jensen, UMass

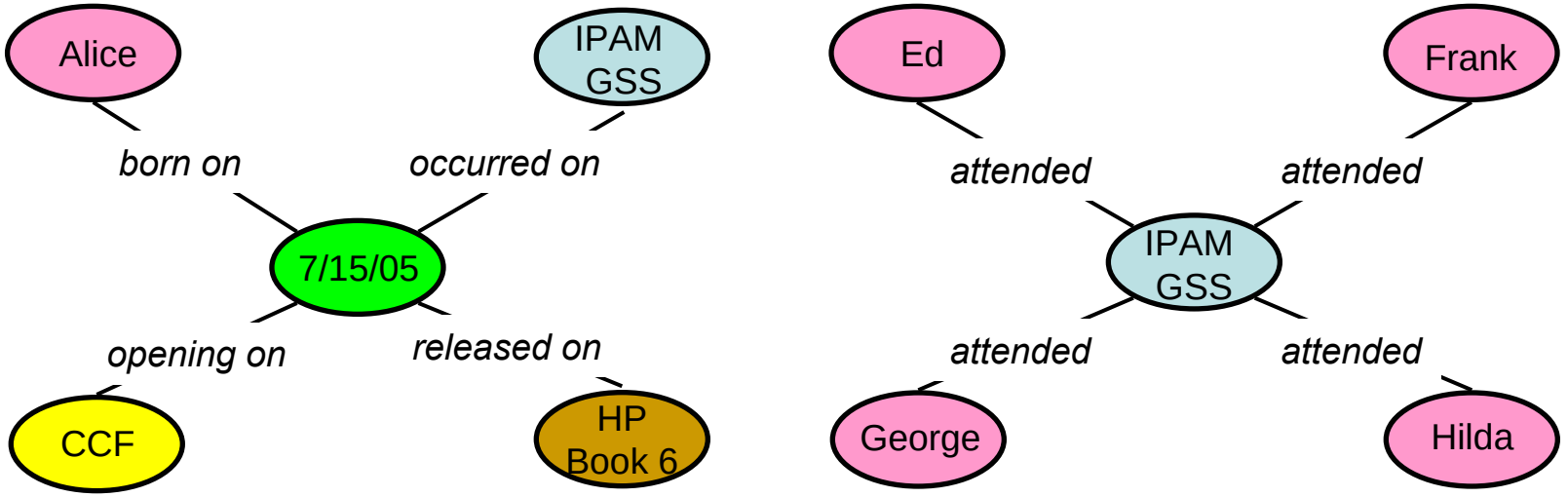


Humans introduce biases into the manually defined type graph!

Generating type graphs: human vs. computer

- Current framework (1)
  - Expert manually defines type graph based on his/her prior knowledge.
- Alternative framework (2)
  - Automatically define/learn type graph from data.
  - Requires getting the expert to trust and accept the computer-generated type graph.
- Desired framework
  - Combine (1) and (2)

Finding flaws in the type graph such as disparity of connected types



- Some node types have connections to many different other node types, even when these nodes are not connected to many other nodes
- This can be quantified as the disparity of connected types (measured by one number)
- High disparity node types are often not relevant for path finding (i.e., search).

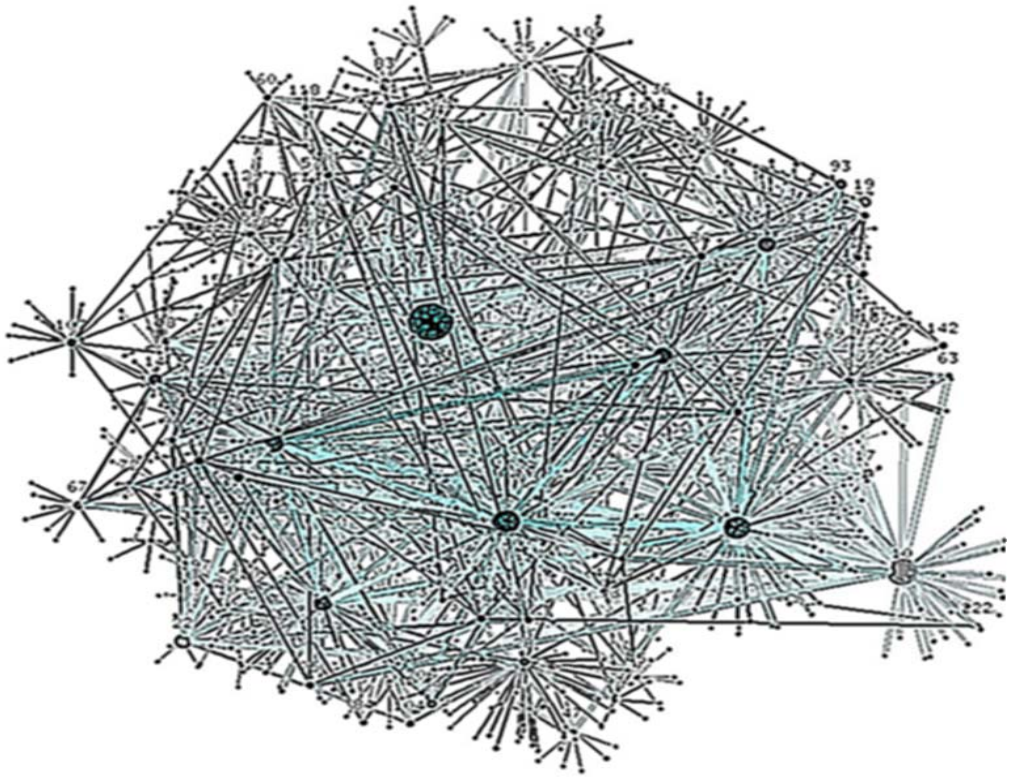
Terrorism instance graph

- Data about world-wide terrorist events from ADL
- Ontology from Niles and Pease, 2001 (http://ontology.teknowledge.com/)
  - Part of IEEE Standard Ontology Working Group
- High disparity: Nation, Region, City
- Low disparity: Kidnapping, Shooting, Bombing, Suicide Bombing, Car Bombing, Knife Attack
- Semantically similar nodes have similar values of average number of neighbors per type, and similar values of disparity

See Barthelemy, Chow, & Eliassi-Rad, AAAI SS 2005.

Basics of instance graphs

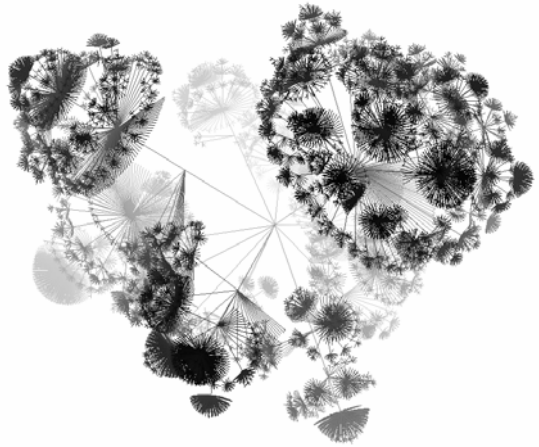
- Set of vertices
- Set of *directed* edges
- Confidence values on vertices and edges



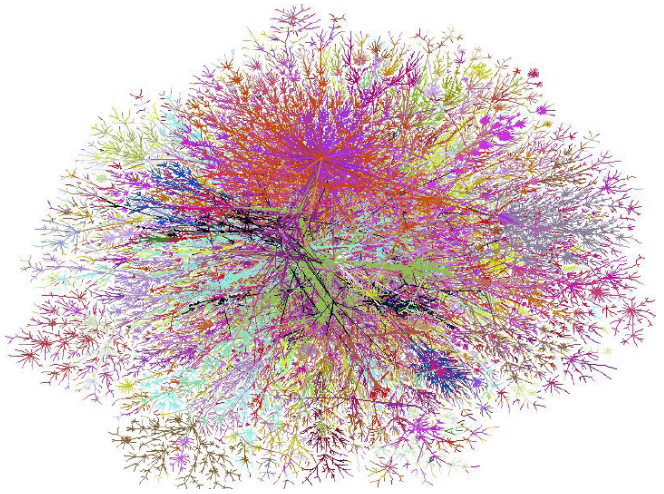
Visualization courtesy of Jonathan McPherson, UC-Davis

Topology of instance graphs

- Small world
  - High clustering coefficient
  - Low diameter (or short average path length)
  - Karinthy 1929, Pool & Kochen 1978, Milgram 1967, Watts & Strogatz 1998
- Scale free
  - Degree distribution follows power-law
  - Preferential or proportional attachment
    - Polya 1923, Yule 1925, Zipf 1949, Simon 1955, Price 1976, Barabasi & Albert 1999
- Strength of weak ties
  - Granovetter 1973



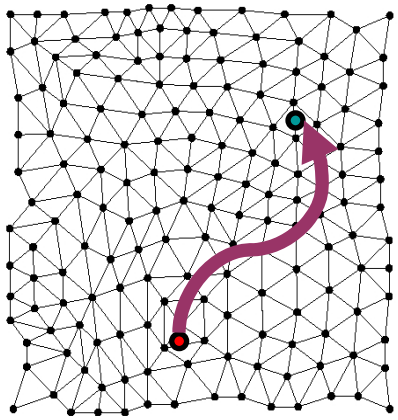
Citation network from arXiv.org.
Visualization by J. McPherson, UC-Davis



Internet autonomous systems
Courtesy of Burch & Cheswick

Milgram's small world experiment (1967)

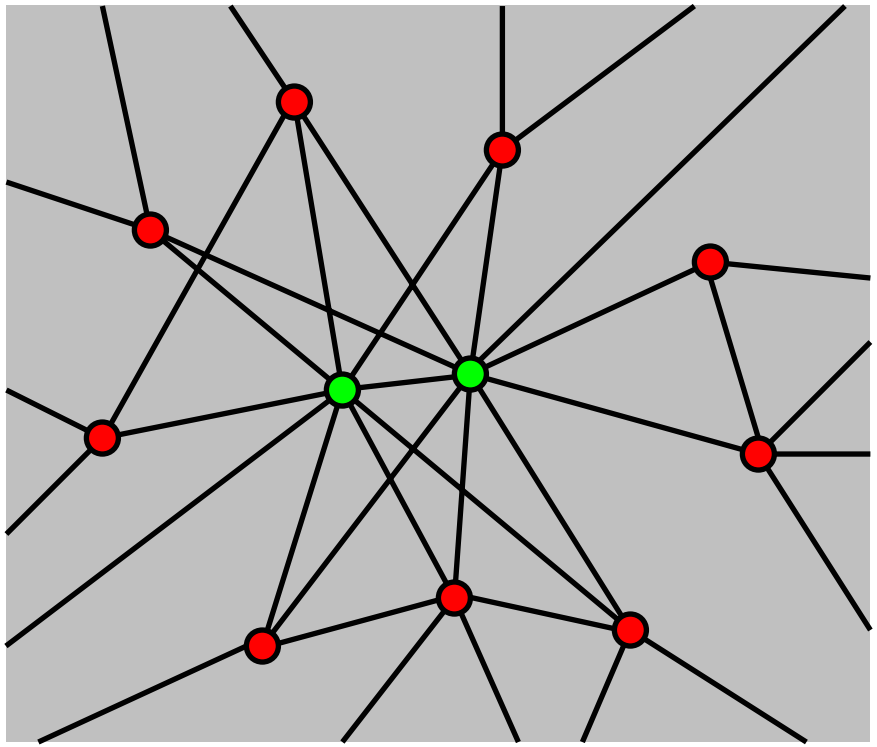
- Try to send a letter to a person in Boston via forwarding the letter to a friend
- Senders from Omaha, Nebraska, and Boston
- Of the completed chains (20%), average chain length was *6.5*
- How are people able to find these chains, using only *local* information?



Greedy search:
Forward to neighbor most like the target

Clustering coefficient

- Two neighboring vertices are likely to have many of the same neighbors



Normally clustering coefficient =

$$C_i = \frac{2E_i}{k_i(k_i - 1)}$$

k_i = num. neighbors of vertex i

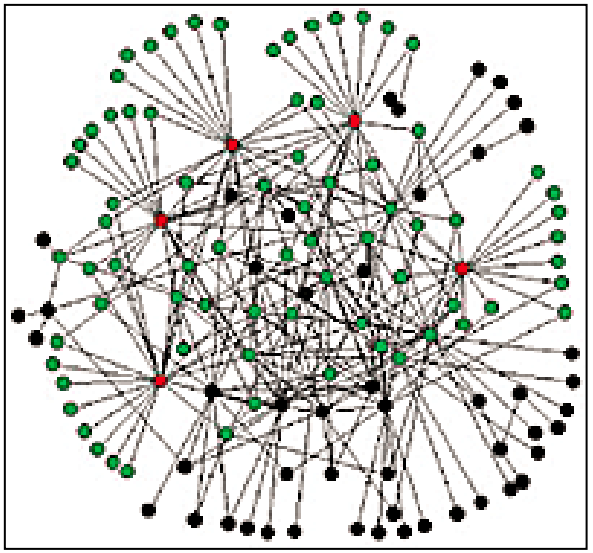
E_i = num. edges between the k_i nodes

$$C_i = \frac{2E_i}{E_{i;\tau}}$$

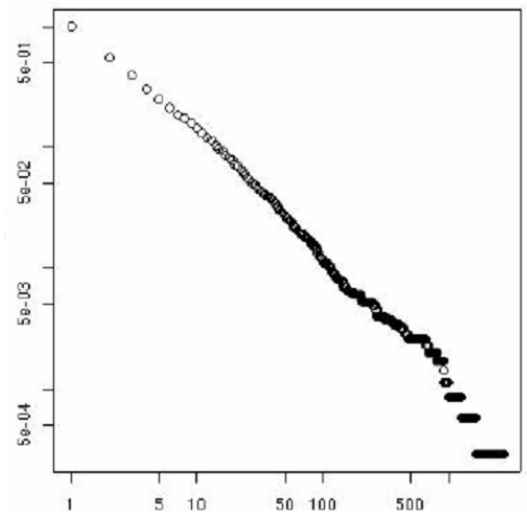
where $E_{i;\tau}$ denotes the maximum number of links allowed by the type graph.

Instance graphs have a wide range of vertex degrees.

- Scale-free network: frequency of vertex with degree k is approximately proportional to $k^{-\alpha}$ (with $2 < \alpha < 3$ generally)
- Vertices with very high degree (hubs) form a "vertex separator"-- removing them will disconnect the graph



A scale-free network (Barabasi & Albert, 1999). Red vertices are hubs.



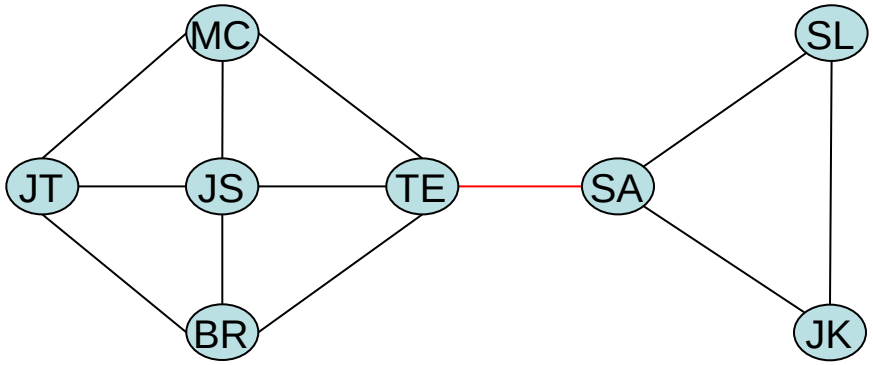
Courtesy of http://www.caida.org

k vs. P(k)

The ratio of the number of hubs to the number of non-hubs remains constant as the graph changes in size.

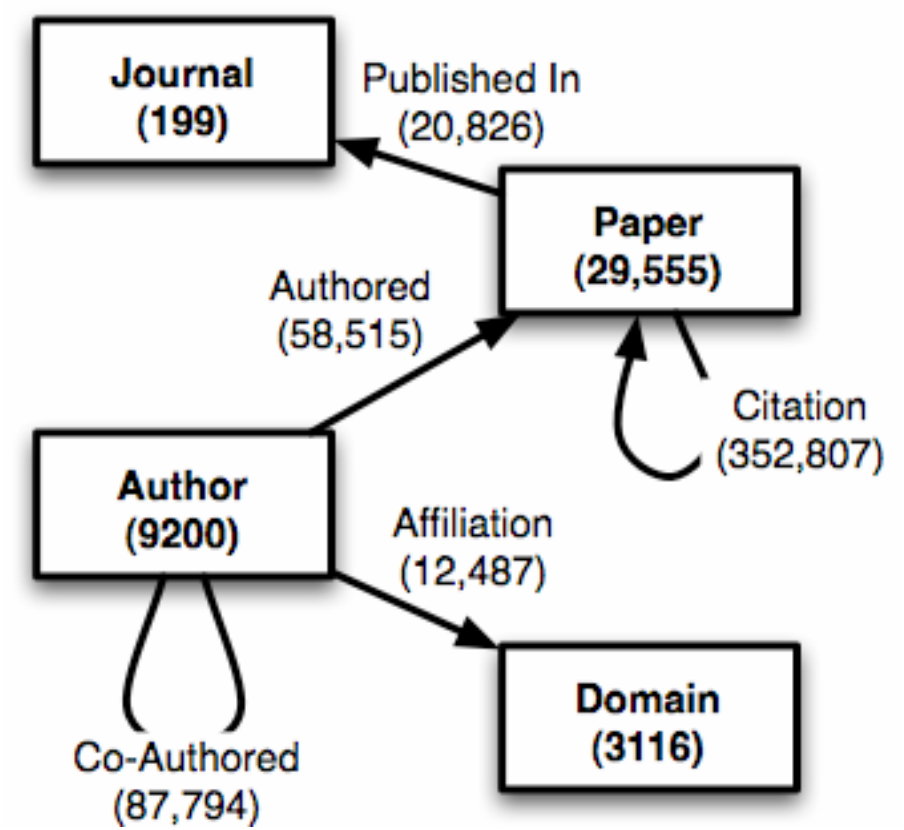
Strength of Weak Ties (SWT)

- M. Granovetter (1973, 1983)
  - Acquaintances are weak ties.
  - Close friends are strong ties.
  - Acquaintances are less likely to be connected with close friends.



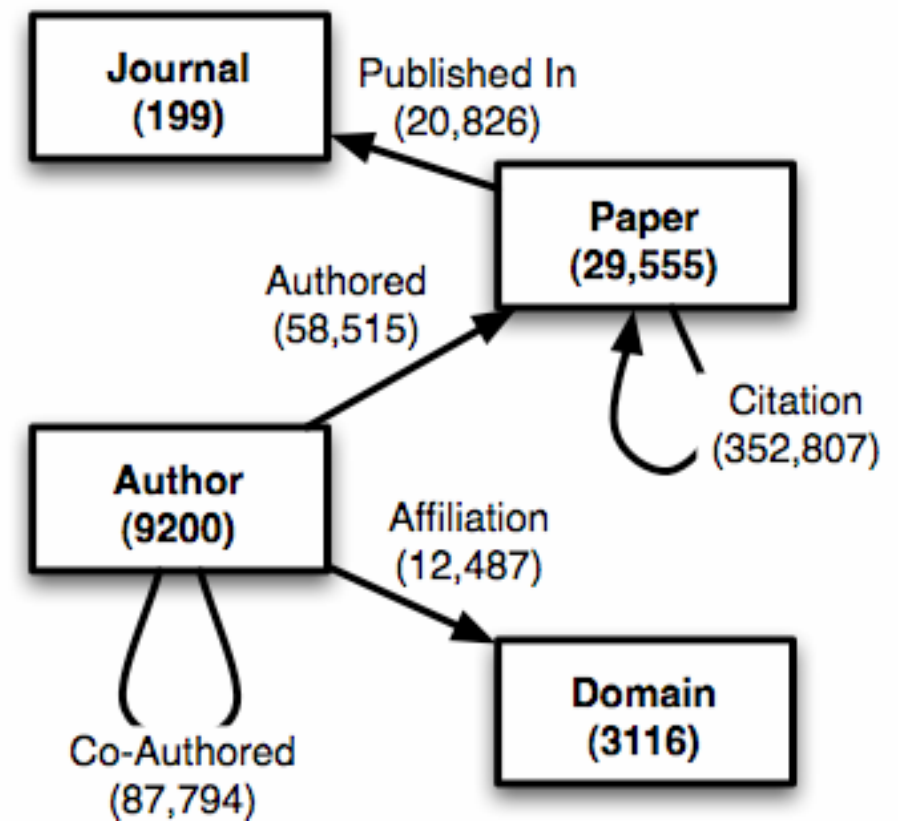
- SWT captures betweenness or rush (Anthonisse, 1971)
  - A vertex or edge with high betweenness has great influence over the flow of information in the graph.
  - A good place to partition the graph.

Type Graph from Physics Preprint Archive (www.arxiv.org)

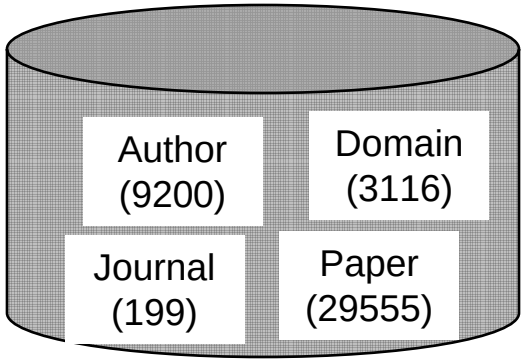


A. McGovern, L. Friedland, M. Hay, B. Gallagher, A. Fast, J. Neville, and D. Jensen, Exploiting relational structure to understand publication patterns in high-energy physics, *SIGKDD Explorations* 5(2):165-173, 2003.

Unconditional probability distribution of vertex types in the ArXiv instance graph

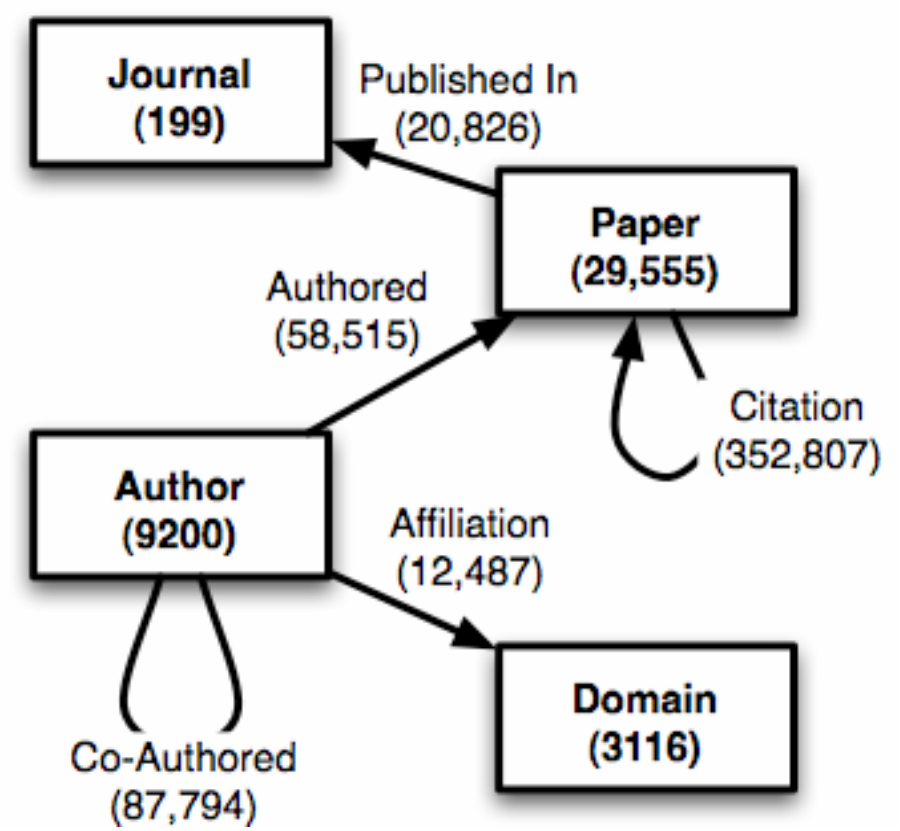


An Urn Filled with Vertices →



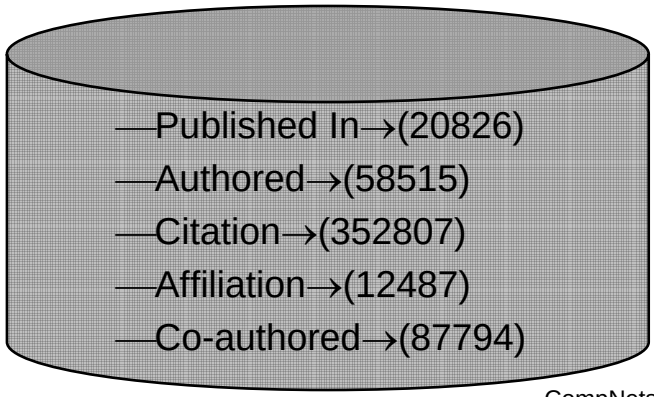
Vertex Type	Probability of Vertex Type Occurrence in Semantic Graph
Author	Pr(*v*'s type is author)=0.218
Domain	Pr(*v*'s type is domain)=0.074
Journal	Pr(*v*'s type is journal)=0.005
Paper	Pr(*v*'s type is paper)=0.703

Unconditional probability distribution of edge types in the ArXiv instance graph

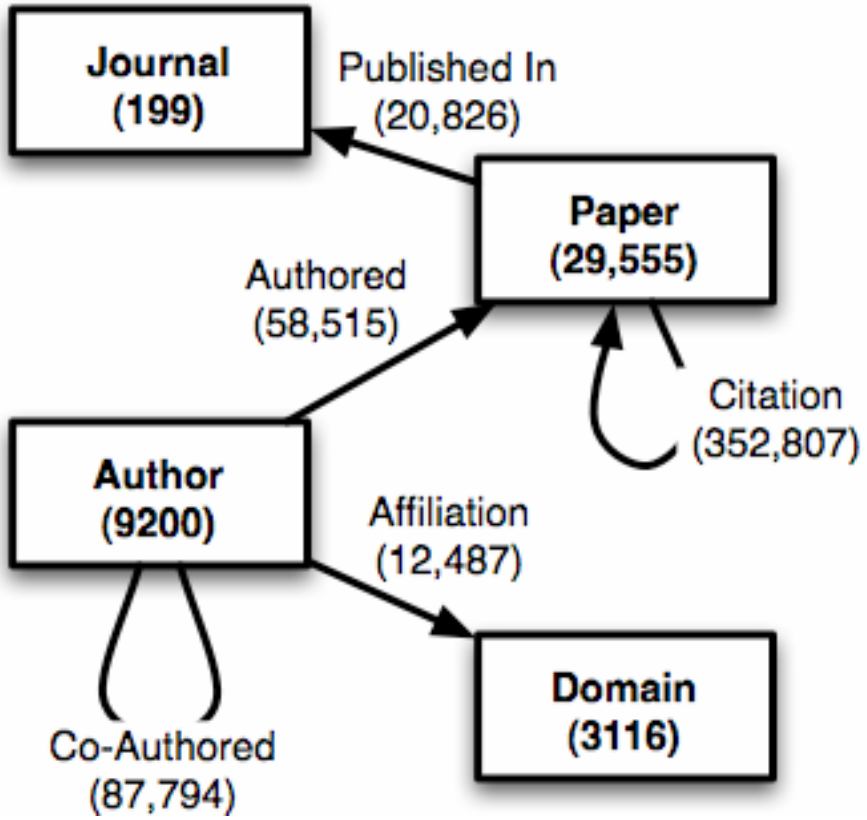


Edge Types	Probability of Edge Type Occurrence in Semantic Graph
Published In	Pr(*e* is published in)=0.04
Authored	Pr(*e* is authored)=0.11
Citation	Pr(*e* is citation)=0.66
Affiliation	Pr(*e* is affiliation)=0.02
Co-Authored	Pr(*e* is co-authored)=0.17

An Urn Filled with Edges →



Conditional probability distribution on edge types based on end points

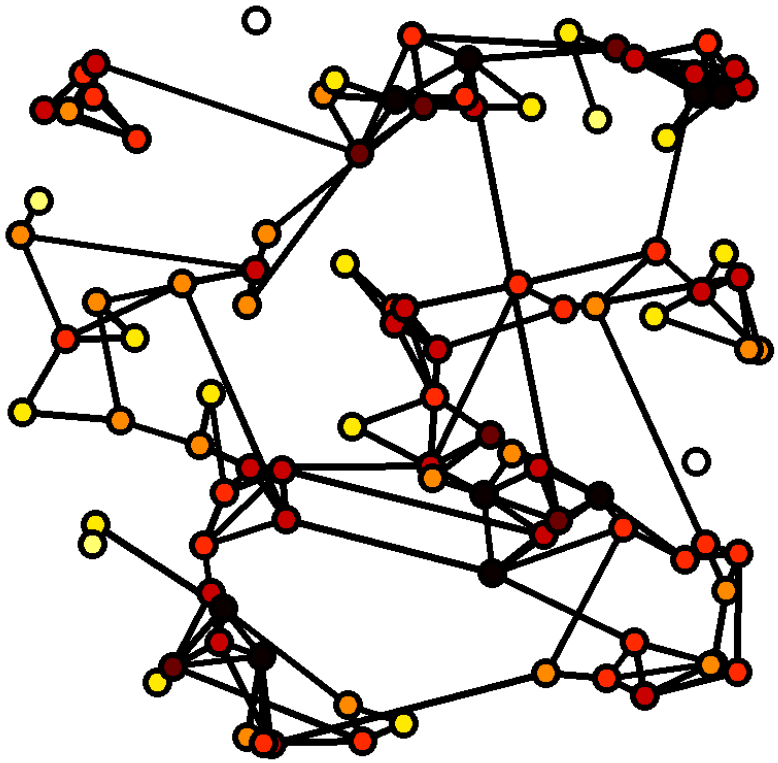


- *Assuming uniformity of edge distribution across end points*
  - *Can get around this with sampling*

- Conditional on start point:
  - $Pr(A_v \text{—} AU_e \rightarrow [*]_v \mid A_v) = 0.37$
  - $Pr(A_v \text{—} AF_e \rightarrow [*]_v \mid A_v) = 0.08$
  - $Pr(A_v \text{—} CA_e \rightarrow [*]_v \mid A_v) = 0.55$
  - $Pr(P_v \text{—} PI_e \rightarrow [*]_v \mid P_v) = 0.06$
  - $Pr(P_v \text{—} CI_e \rightarrow [*]_v \mid P_v) = 0.94$
  - $Pr(D_v \text{—} [*]_e \rightarrow [*]_v \mid D_v) = 0$
  - $Pr(J_v \text{—} [*]_e \rightarrow [*]_v \mid J_v) = 0$
- Conditional on end point:
  - $Pr([*]_v \text{—} CA_e \rightarrow A_v \mid A_v) = 1.00$
  - $Pr([*]_v \text{—} AU_e \rightarrow P_v \mid P_v) = 0.14$
  - $Pr([*]_v \text{—} CI_e \rightarrow P_v \mid P_v) = 0.86$
  - $Pr([*]_v \text{—} AF_e \rightarrow D_v \mid D_v) = 1.00$
  - $Pr([*]_v \text{—} PI_e \rightarrow J_v \mid J_v) = 1.00$

Heterogeneous complex networks

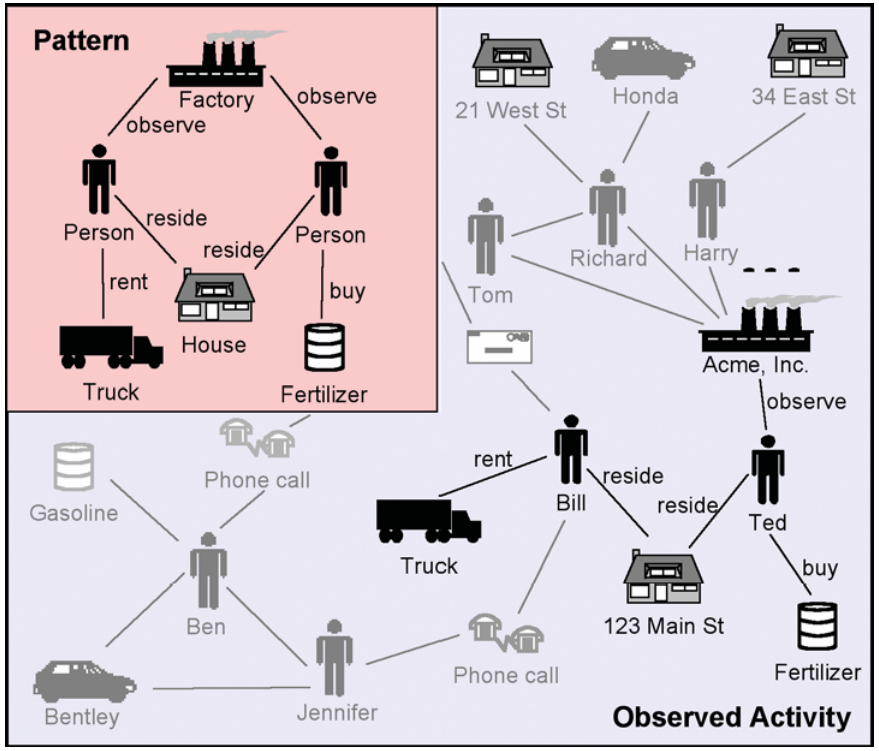
- Irregular structure of connections
- Short average path lengths
- Special degree distributions
- High clustering coefficient



- Correlations between nodes that are linked (homophily)
- Conditional and unconditional probability distributions over the vertex and edge types
- Semantics associated with vertex and edge types

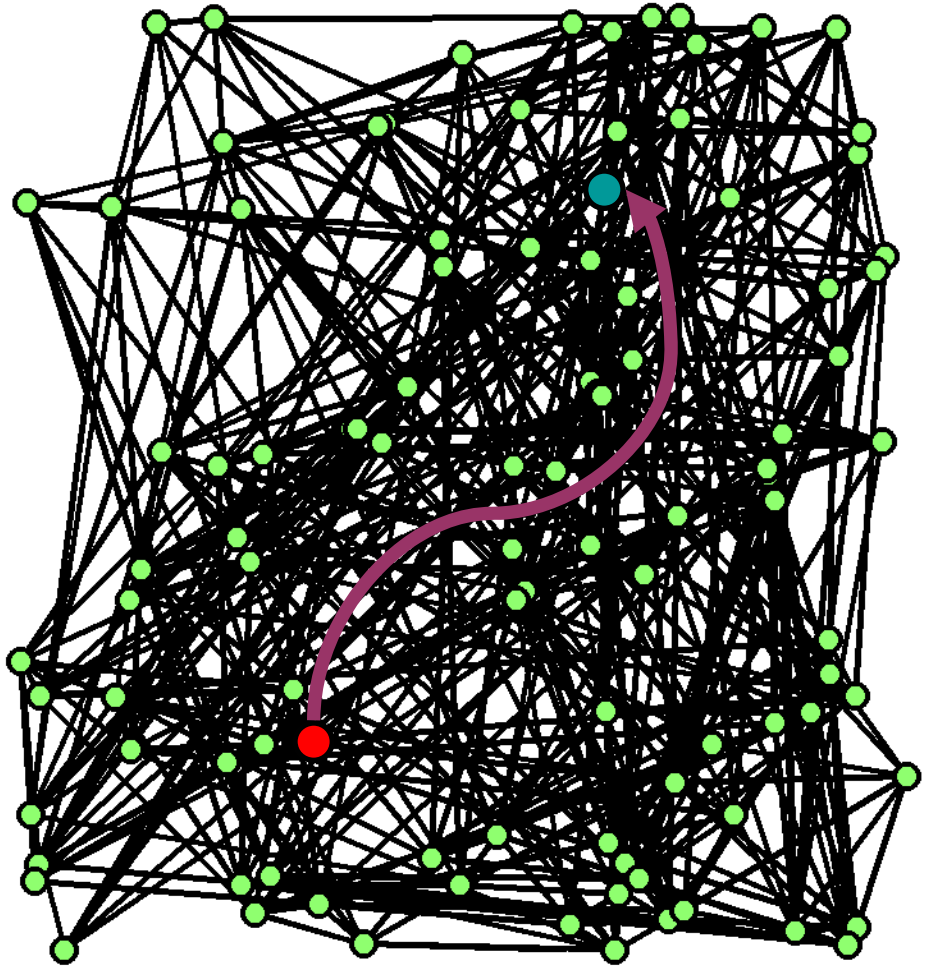
Relationship detection problems for heterogeneous complex networks

- Search with constraints
  - Heuristic search techniques
  - Distributed semantic shortest path search
  - Multi-objective partitioning for semantic graphs
  - Incremental repartitioning for dynamic semantic graphs
- Inexact subgraph isomorphism
- Identifying rare and "interesting" events
- Gap analysis: "What do I need to know?"



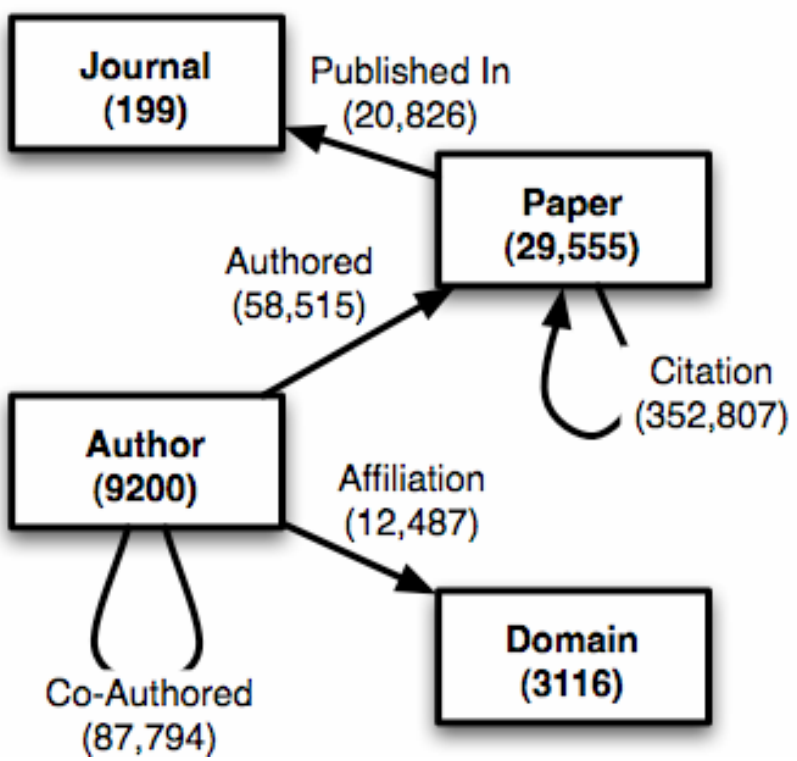
T. Coffman, S. Greenblatt, and S. Marcus, Comm. of the ACM, March 2004.

Given a graph and two vertices, find a shortest distance path between the vertices



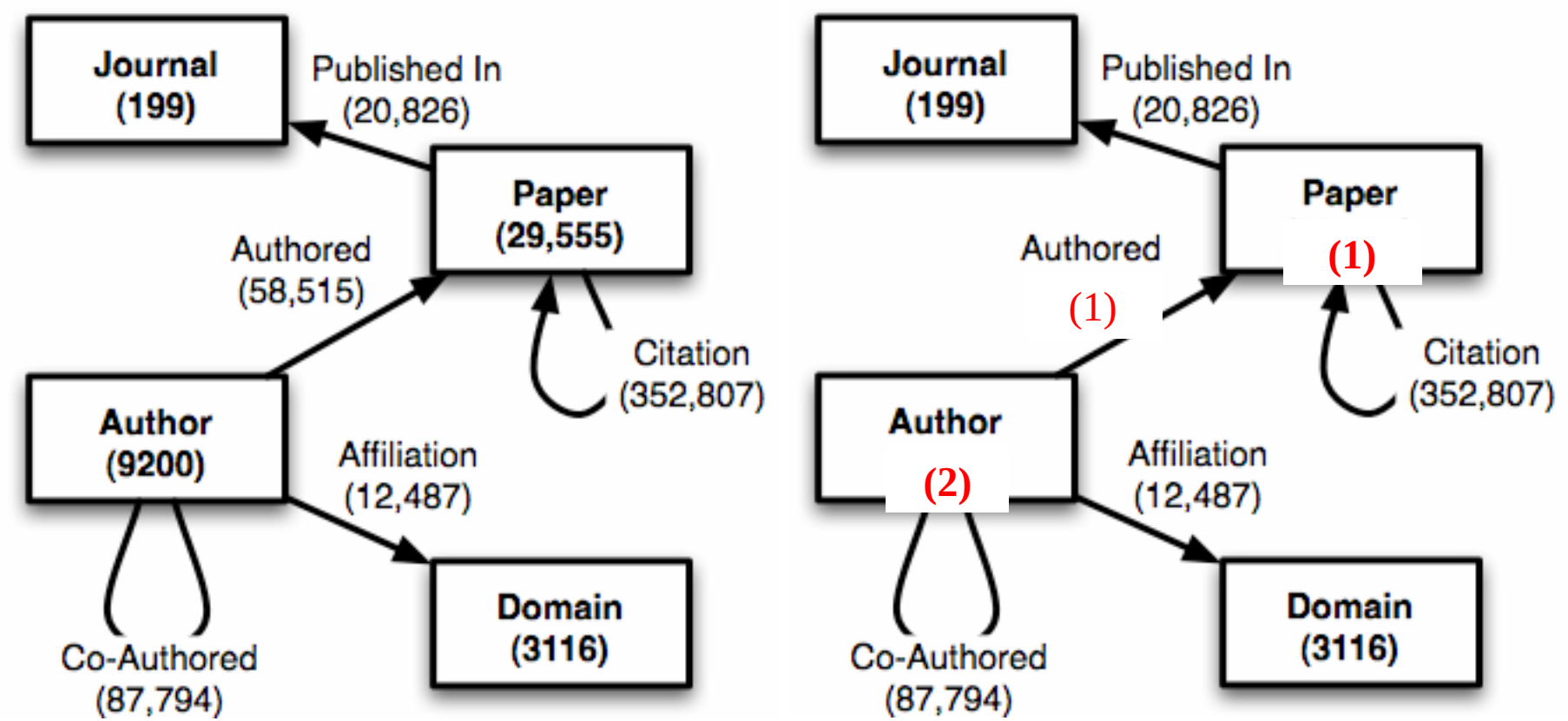
Probability model for edge existence

- Need a fast statistical model that
  - Operates on the type graph
  - Uses frequency information on vertex and edge types
  - Produces a probability for the existence of any edge type, given a particular vertex type
- Will help speed-up search



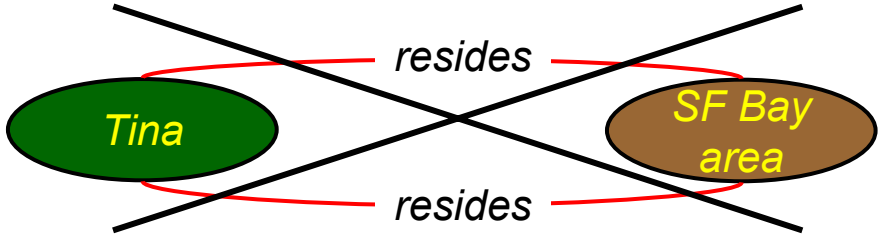
For example: Want to compute the probability that an edge of type *authored* exists from a randomly selected vertex of type *author*

Want a probability model that highlights the difference between these two graphs

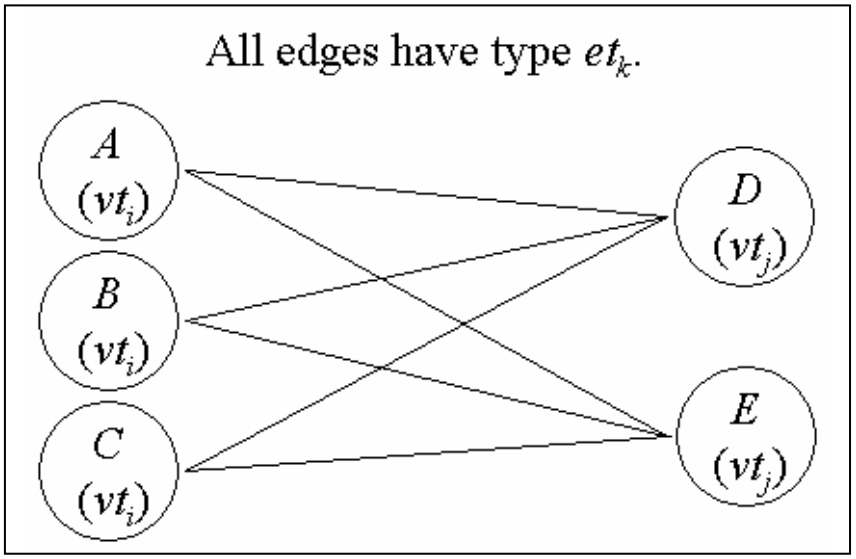


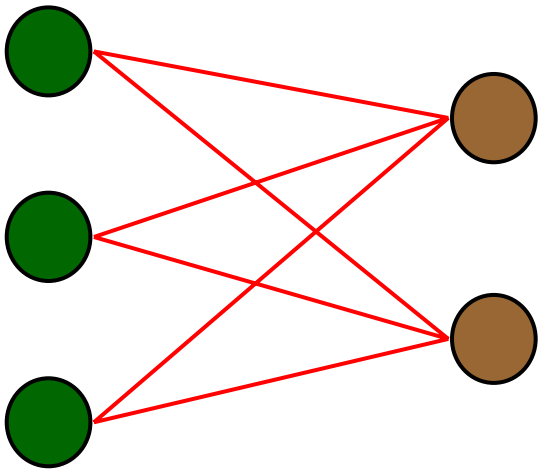
Prohibition on redundant edges and a simple axiom

- Assumption: There are no redundant edges in the semantic graph.



- Axiom: For any edge type et_k connecting vt_i and vt_j, $|et_k| \leq |vt_i| \times |vt_j|$.





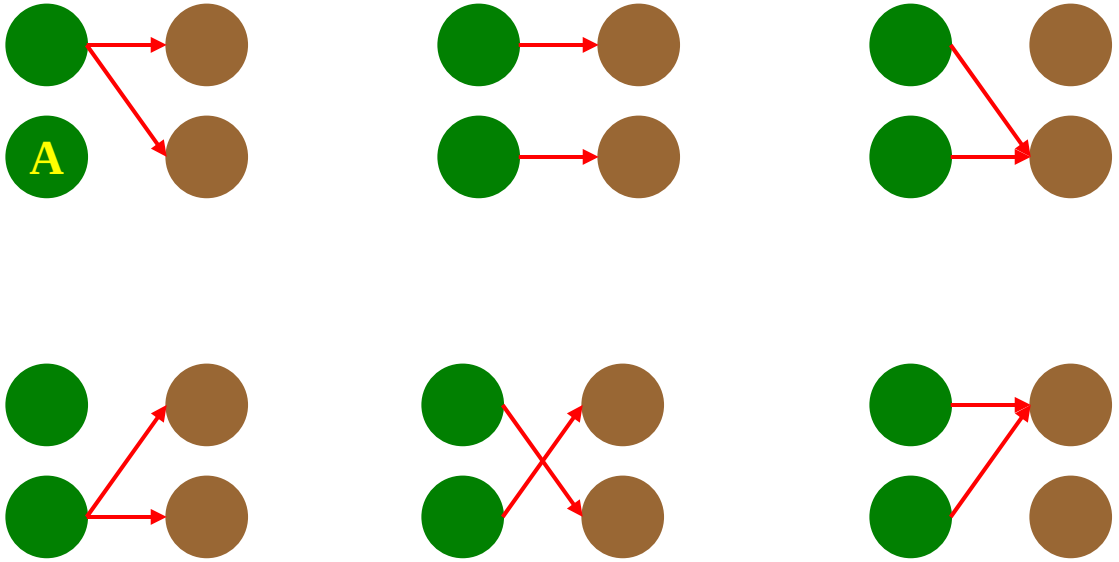
Probability function for measuring an edge's existence

$$Pr\left(\exists \text{ at least one edge } et_k : vt_i \xrightarrow{et_k} vt_j \middle| vt_i\right) = 1 - \frac{\binom{(m-1)n}{l}}{\binom{mn}{l}}$$

$$= 1 - \frac{\prod_{i=0}^{n-1}(mn - l - i)}{\prod_{i=0}^{n-1}(mn - i)} \geq \frac{l}{mn}, \text{ where } l \equiv |et_k|, m \equiv |vt_i|, \text{ and } n \equiv |vt_j|$$

- Denominator = total number of all possible graph structures consistent with the type graph
- Numerator = number of graphs in which the given vertex of type vt_i has at least one edge of type et_k
- With more terms, the product becomes smaller since each factor is less than 1!
- Additional factors become important when $|et_k|$ is small compared to $|vt_i| \times |vt_j|$.

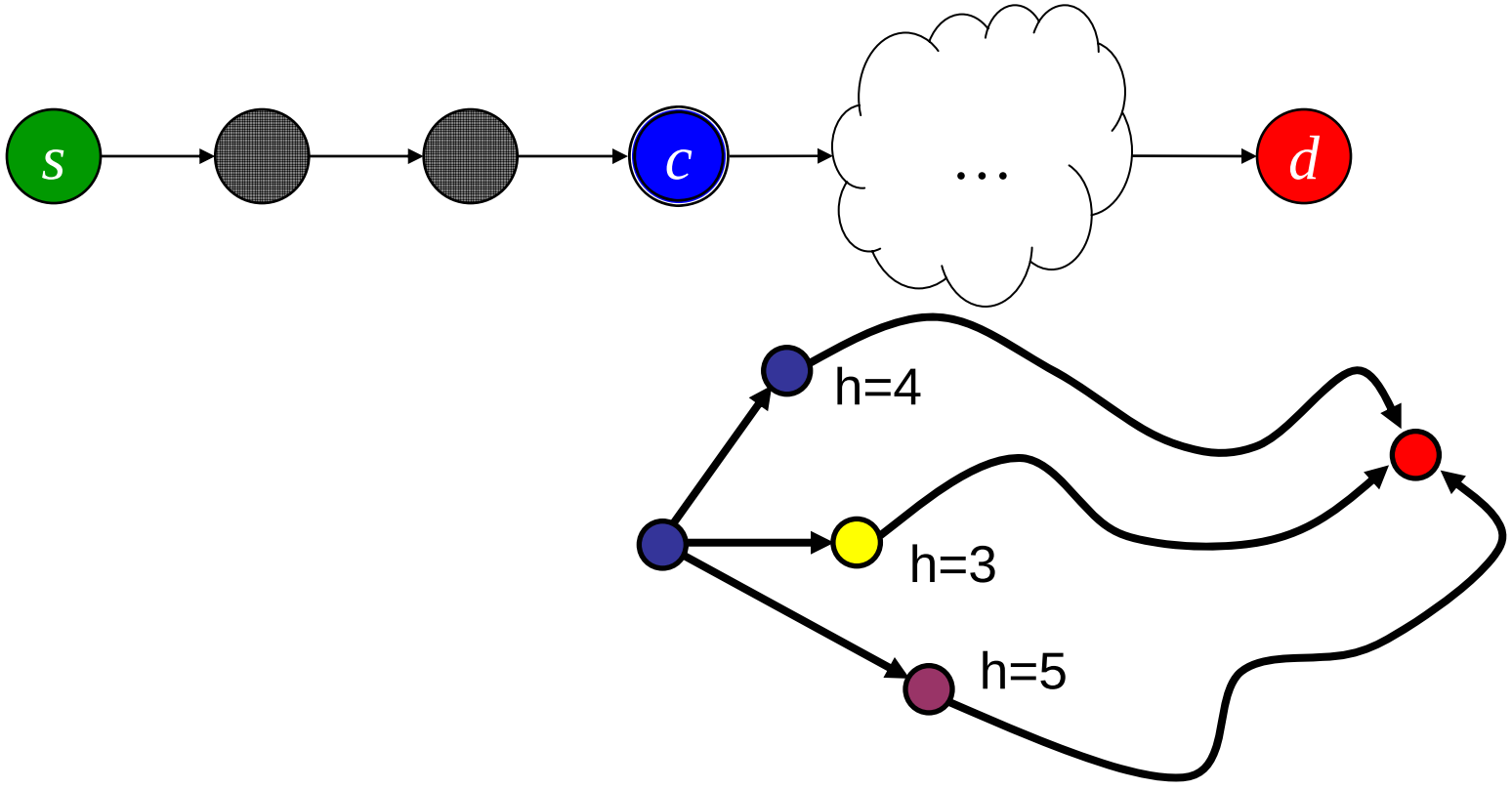
An example



- How many graphs can connect 2 *green* nodes to 2 *brown* nodes via 2 red edges? 6 possible graphs
- How many times is a (uniformly) randomly picked *green* node has *at least one red* out-going edge? 5 times
- Probability that there exists a *red* out-going edge from a *green* node is 5/6 = 0.833.

Heuristics can be used to guide the search

- A* search (Hart, Nilsson, and Raphael 1968)
  - Cost function has the form:
    $$f(s, c, d) = g(s, c) + h(c, d)$$
  - h is admissible if it never over-estimates actual cost



Experiments on Real Graphs

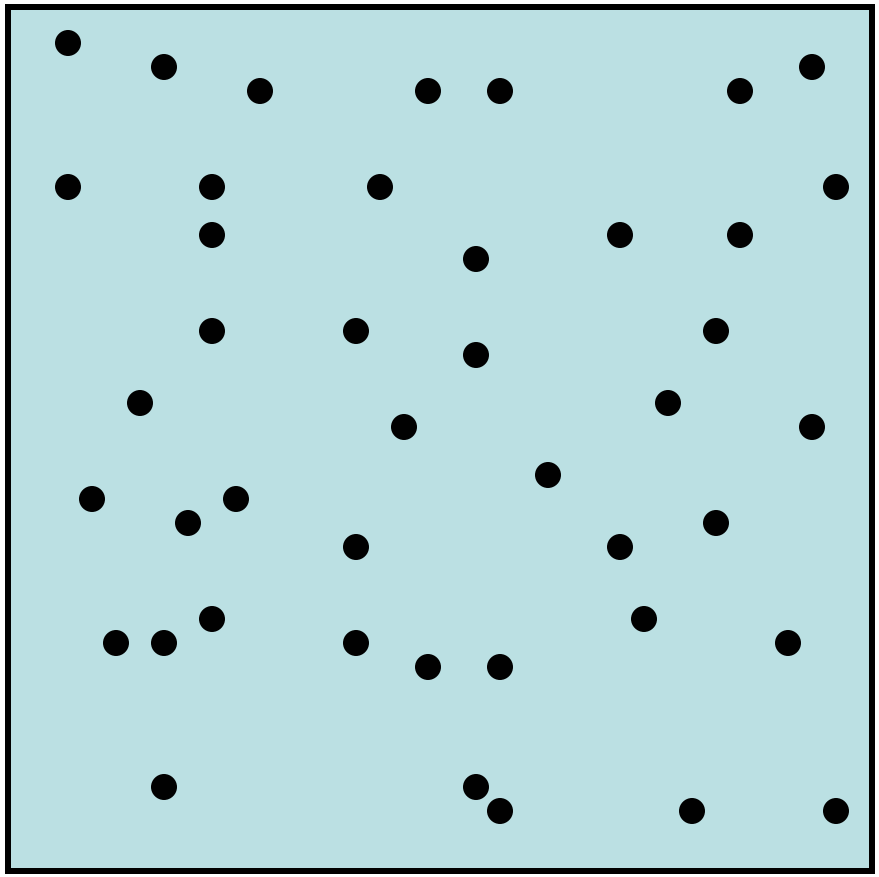
- Terrorism domain
  - # of vertices = 2,436
  - # of edges = 25,234
  - # of vertex types = 59
  - # of edge types = 522
  - Instance graph connectivity = 0.0043
  - Type graph connectivity = 0.15
  - Instance graph avg path length = 2.837
  - Type graph avg path length = 0.964 (has an unconnected component – bad type graph)
  - Avg degree = 10.4
- IMDB domain
  - # of vertices = 42,026
  - # of edges = 528,756
  - # of vertex types = 8
  - # of edge types = 30
  - Instance graph connectivity = 0.0003
  - Type graph connectivity = 0.47
  - Instance graph Avg path length = 3.385
  - Type graph avg path length = 1.5
  - Avg degree = 12.6

Was able to cut the number of nodes visited by more than 50% compared to breadth-first search!

2D Partitioning

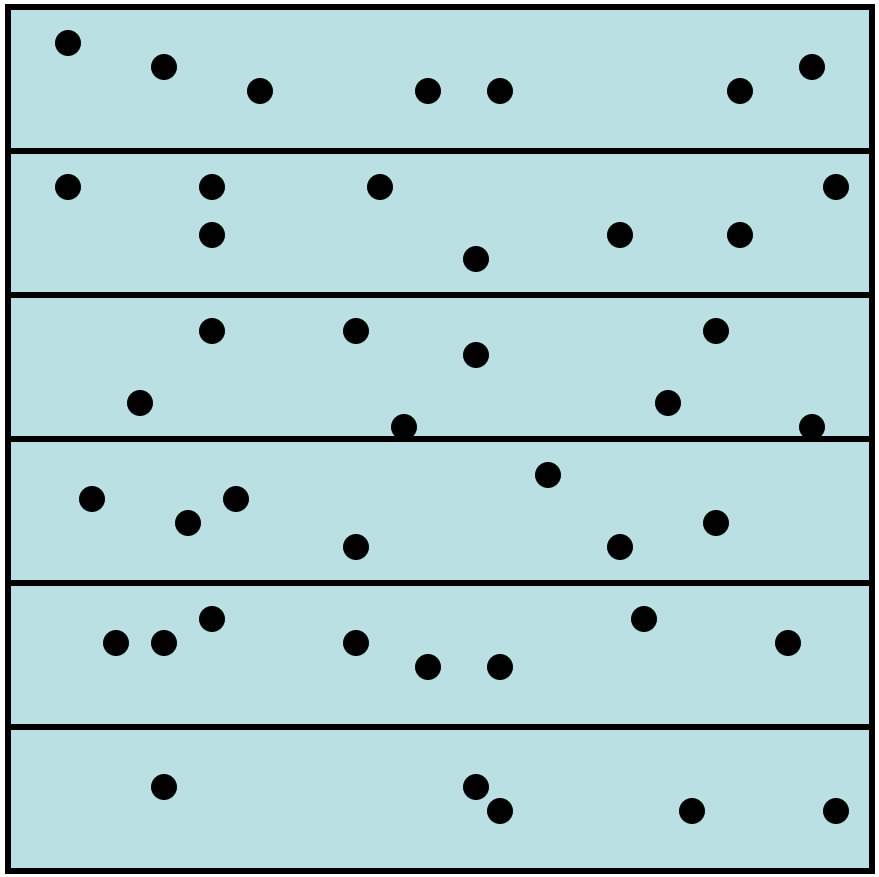
- Partition vertices (1D) or edges (2D)
- 2D partitioning has been advocated for sparse matrices where the sparsity pattern is difficult to exploit (Hendrickson, Leland, and Plimpton 1995).
- Many variants of 2D partitioning (Catalyurek 1999)
- 2D checkerboard variant is perhaps most useful.
  - Redistribution-free, transpose-free doubling/halving (Lewis and van de Geijn 1993, Lewis, Payne, and van de Geijn 1994).
  - 2D checkerboard (Catalyurek 1999, Catalyurek and Aykanat 2001)

Example: adjacency matrix

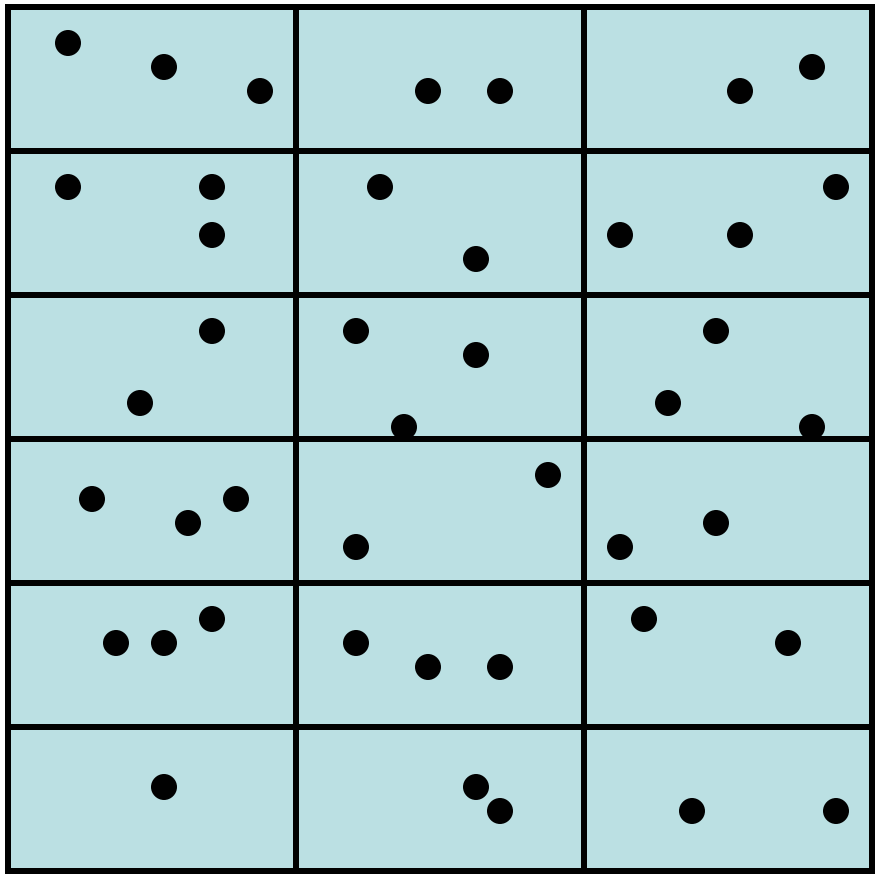


Partition to minimize processor communication while maintaining load balance

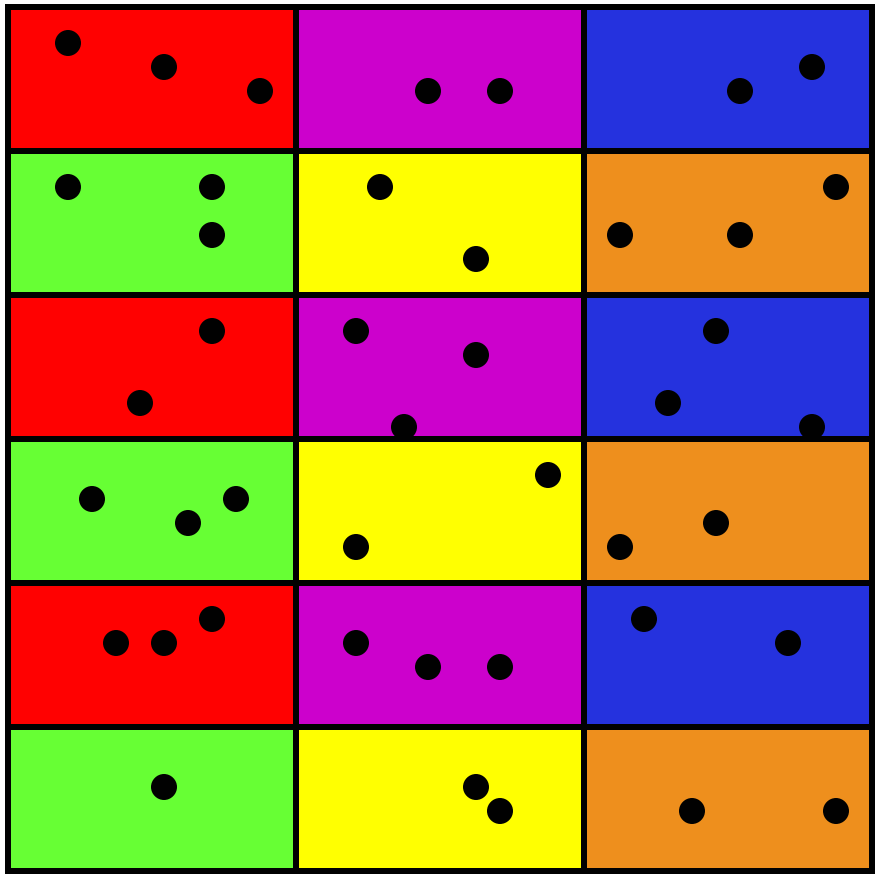
Example: 6-way vertex partitioning

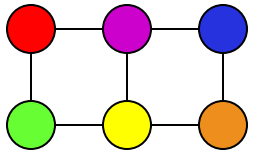


Example: 2x3 edge partitioning



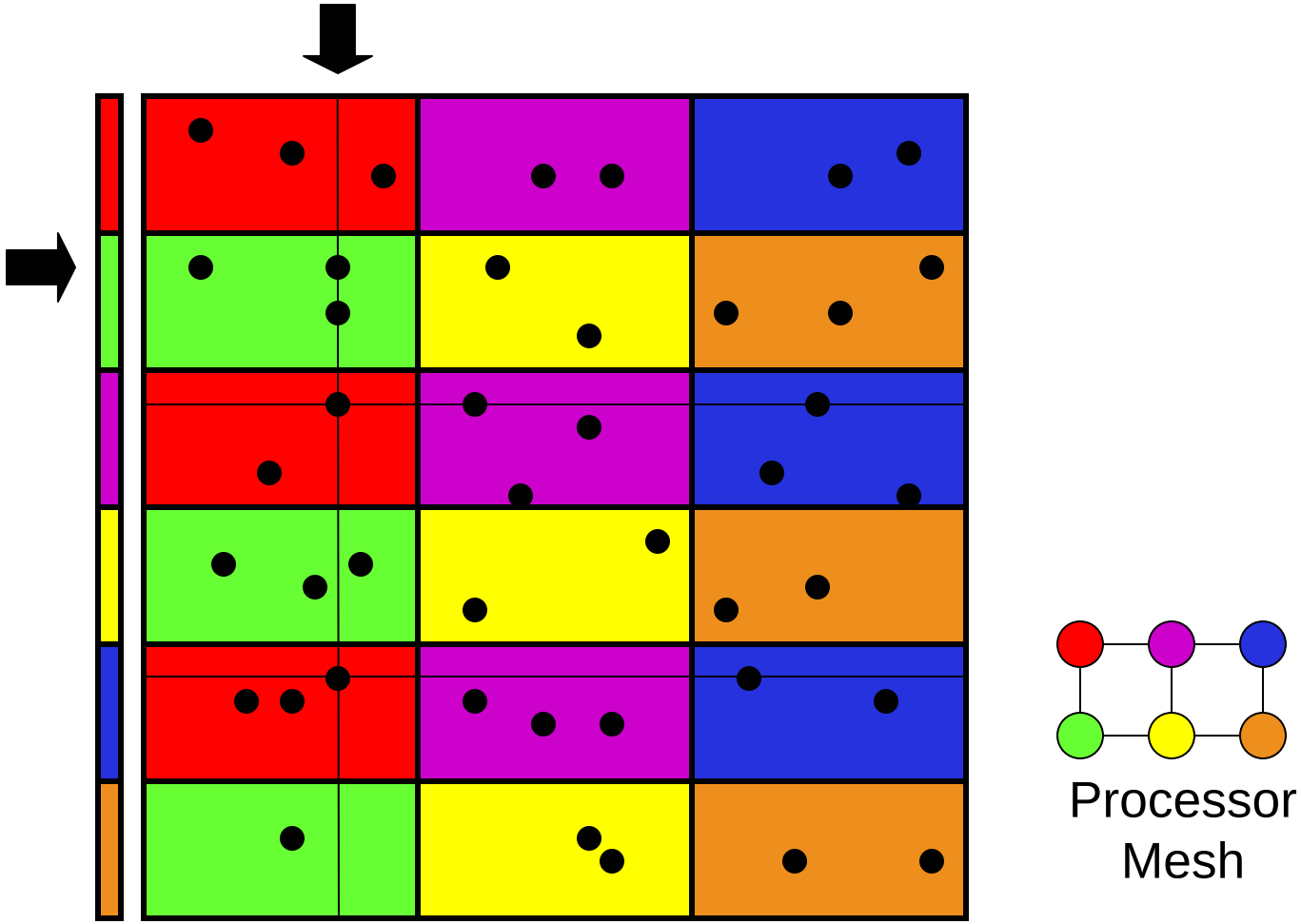
Example: 2x3 edge partitioning





Processor Mesh

Example: 2x3 edge partitioning



Parallel search experimental setup

- Parallel Breadth First Search (BFS) algorithm
  - Level-synchronized algorithm
  - Report average time for 100 pairs
  - Does not take into account increasing graph avg. path length (varies from 5 to 9)
- Input graphs
  - Undirected Poisson random graphs with degree 10
  - Random 2D checkerboard partitioning
  - Vertices and edges accessed from memory
- Machines
  - BlueGene/L

Level-synchronized parallel search

Do l=0 to … until target is found

F = set of assigned vertices with level l

Column Expand communication (send F, receive F')

N = set of neighbor vertices of F'

Row Fold communication (send N, receive N')

Update levels of vertices in N'

End do

- Expand is all-gather or all-to-all.
- Reduce is all-to-all or reduce-scatter.
- Must store *vertex* lists in sparse mode.
- Storage is scalable for random graphs.
- If the blocks are balanced, then the communication is balanced for any graph.

BlueGene/L timings, up to 32K processors

Constant local problem size of 100k vertices/processor for a random graph with average degree 10.

Number of Vertices	Search Time (s)
1.00 Billion	4.37
1.96 Billion	4.64
3.28 Billion	4.90

2-D partitioning is effective for unstructured graphs with high average degree.

Can we use information theory to identify rare and/or interesting events?

$$H(X) = \left(-\sum_{x \in X} p(x) \log p(x) \right) \geq 0$$

$$H(Y \mid X) = -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(y \mid x) \leq H(Y)$$

$$H(X, Y) = -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(x, y) = H(X) + H(Y \mid X)$$

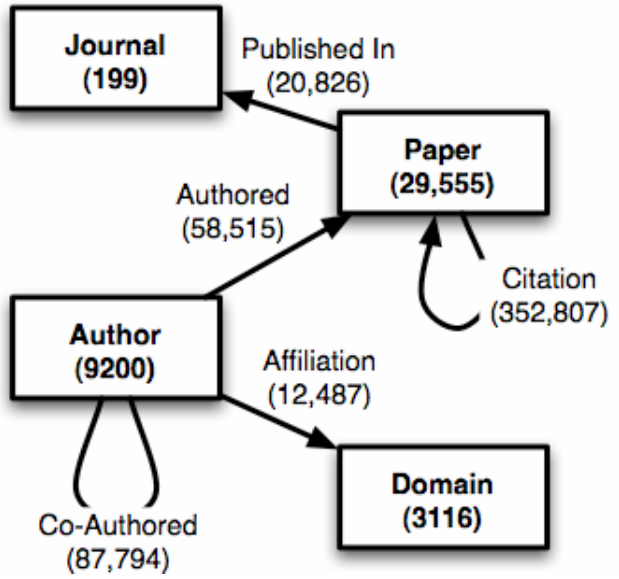
$$I(X;Y) = \left(\sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x) p(y)} \right) \geq 0$$

$$\begin{aligned} I(X;Y) &= H(X) - H(X \mid Y) \\ &= H(Y) - H(Y \mid X) \\ &= H(X) + H(Y) - H(X, Y) \end{aligned}$$

H(*X*) = Entropy of a Discrete Random Variable

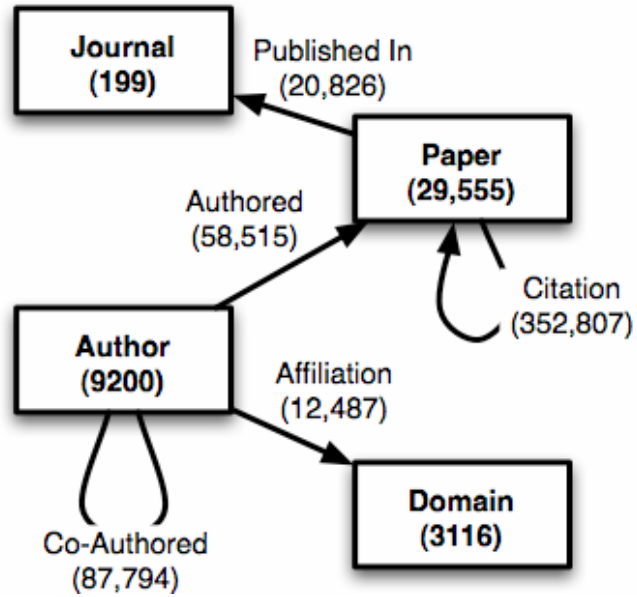
- If *X* is an edge type, then *H*(*X*) measures the uncertainty that an instance of *X* exists in the graph.
- *p*(*Authored* | *Type Graph*)
  = *p*(*Author*) × *p*(*Authored* | *Author*) + *p*(*Paper*) × *p*(*Authored* | *Paper*)
  = 0.22 * 0.99 + 0.70 * 0.86 = 0.82
- *H*(*Authored* | *Type Graph*)
  = – (*p*(*Auth'ed*) × *Log*[*p*(*Auth'ed*)] + *p*(¬*Auth'ed*) × *Log*[*p* (¬*Auth'ed*)])
  = – (0.82 * *Log*[0.82] + 0.18 * *Log*[0.18]) = 0.68

- *H*(*Published-In*) = 0.94
- *H*(*Citation*) = 0.88
- *H*(*Affiliation*) = 0.80
- *H*(*Co-Authored*) = 0.76
- *H*(*Authored*) = 0.68



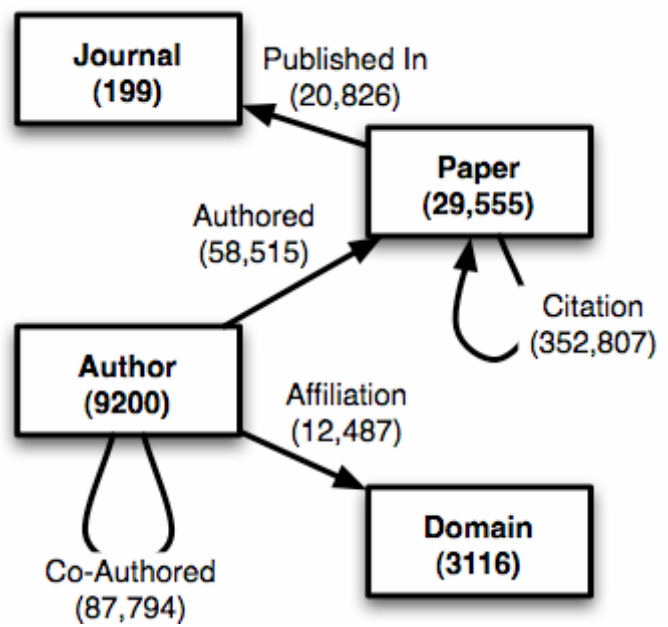
Mutual Information: $I(X; Y) = H(X) - H(X \mid Y)$

- Need $P(X \mid Y)$
  - E.g., P(Published-in | Authored)
- Ove Frank (1978)
  - Sampling and estimation in large social networks
- Procedure for instance graphs
  - Draw a set of vertices by simple random sampling
  - Observe the type of each vertex and the frequency of edge types between each sampled vertex and its immediate neighbors.
- Avoids assumptions about uniform distribution of incident edges



I(*X; Y*) = Measure of amount of information that *X* contains about *Y*

- *I*(*Co-Authored; Affiliation*) = 0.24
- *I*(*Published-in; Citation*) = 0.21
- *I*(*Authored; Co-Authored*) = 0.06
- *I*(*Authored*; *Affiliation*) = 0.04
- *I*(*Authored; Citation*) = 0.02
- *I*(*Authored; Published-in*) = 0.003



Finding interesting relationships with information theory

- For each edge type, *X*
  - Compute *H*(*X*).
  - For each edge type, *Y*
    - Compute *H*(*X*|*Y*) and *I*(*X*;*Y*).
- If *I*(*X*;*Y*) is relatively small (i.e., reduction in uncertainty of *X* due to knowing *Y* is relatively small), then the *X*, *Y* chain of relationships are rare or interesting.
- Otherwise, the *X*, *Y* chain of relationships are not rare or interesting.
- This analysis can be done for chains of relationships with more than two links.

$$I(X;Y \mid Z) = H(X \mid Z) - H(X \mid Y,Z)$$

$$I(X_1, X_2, \cdots, X_n; Y) = \sum_{i=1}^{n} I(X_i; Y \mid X_{i-1}, X_{i-2}, \cdots X_1)$$

Conclusions and perspectives

- Heterogeneous complex networks (or semantic graphs) are currently popular for the task of “connecting the dots”.
  - The sizes of such networks in terms of vertices and edges are enormous.
- Standard graph theory metrics need to be updated for characterizing node types by taking into account the type graph which specifies the permitted connections in the instance graph.
- Watch out for biases in the human-defined type graph.
- Topological and semantic properties may be useful tools for designing ontology graphs.
- Bidirectional search, heuristic search, machine-optimized communication primitives, and improved partitioning will improve performance significantly for path finding.
- Algorithms should be evaluated with respect to graphical and statistical properties of the networks.
- Use information theory metrics to find interesting or rare chains of relationships (as defined by the type graph).

Some related work

Pattern analysis with semantic graphs

- Jensen, et al.
- Popp, Armour, Senator, and Numrych
- Coffman, Greenblatt, and Marcus

Relational data and networks in machine learning

- Daphne Koller
- Lise Getoor
- David Jensen
- …

Project team and collaborators

- Keith Henderson and Andy Yoo (Lawrence Livermore National Lab)
- Edmond Chow (D.E. Shaw Research and Development)
- Bruce Hendrickson and William McLendon (Sandia National Labs)
- David Jensen (University of Massachusetts)
- Umit Catalyurek and Doruk Bozdag (Ohio State University)
- For more information
  - Visit http://www.llnl.gov/casc/compnets
  - Contact me at eliassi@llnl.gov

This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48. UCRL-PRES-212773-REV-1.