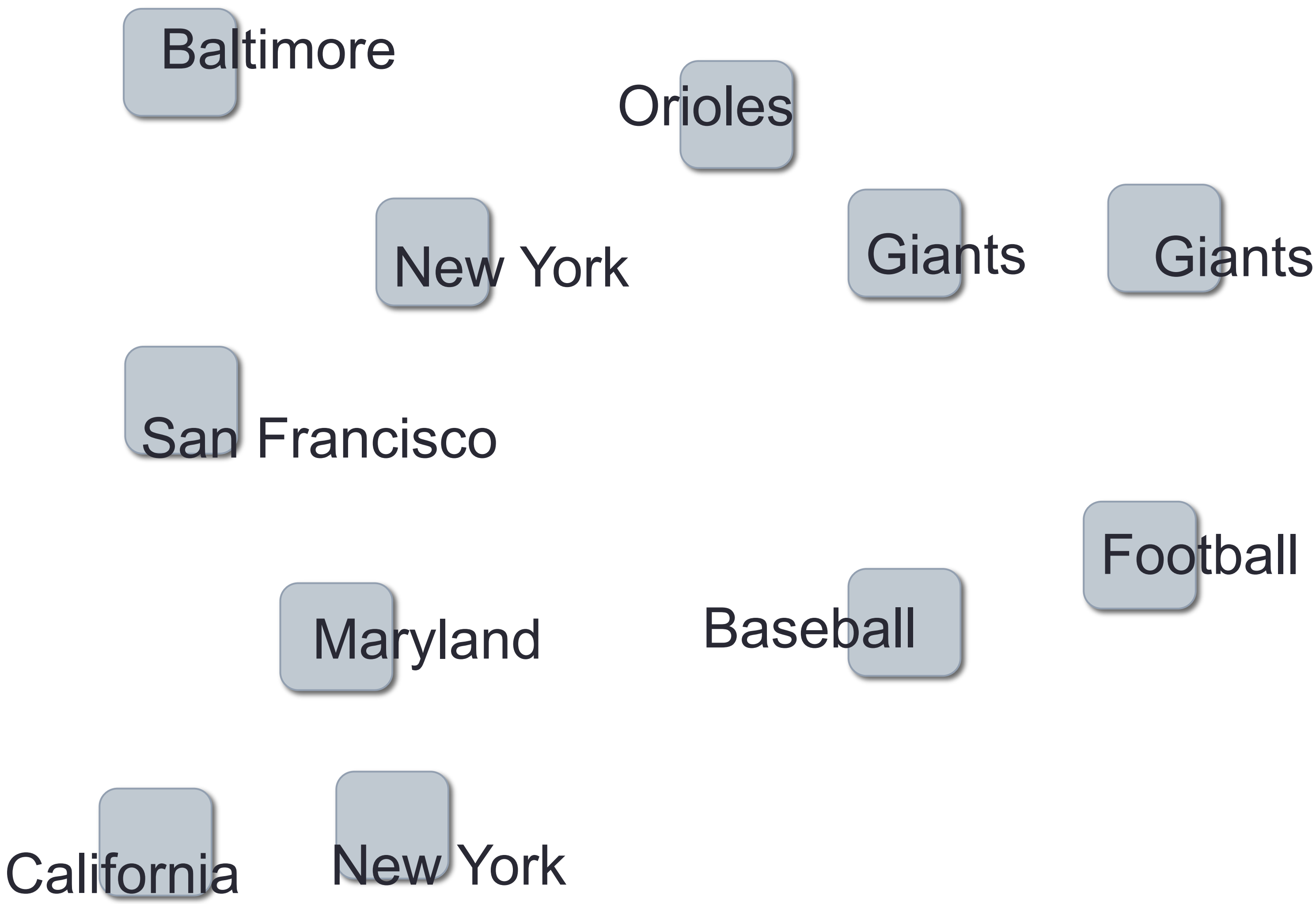
Ontology-Aware Partitioning for Knowledge Graph Identification

Jay Pujara, Hui Miao, Lise Getoor, William Cohen

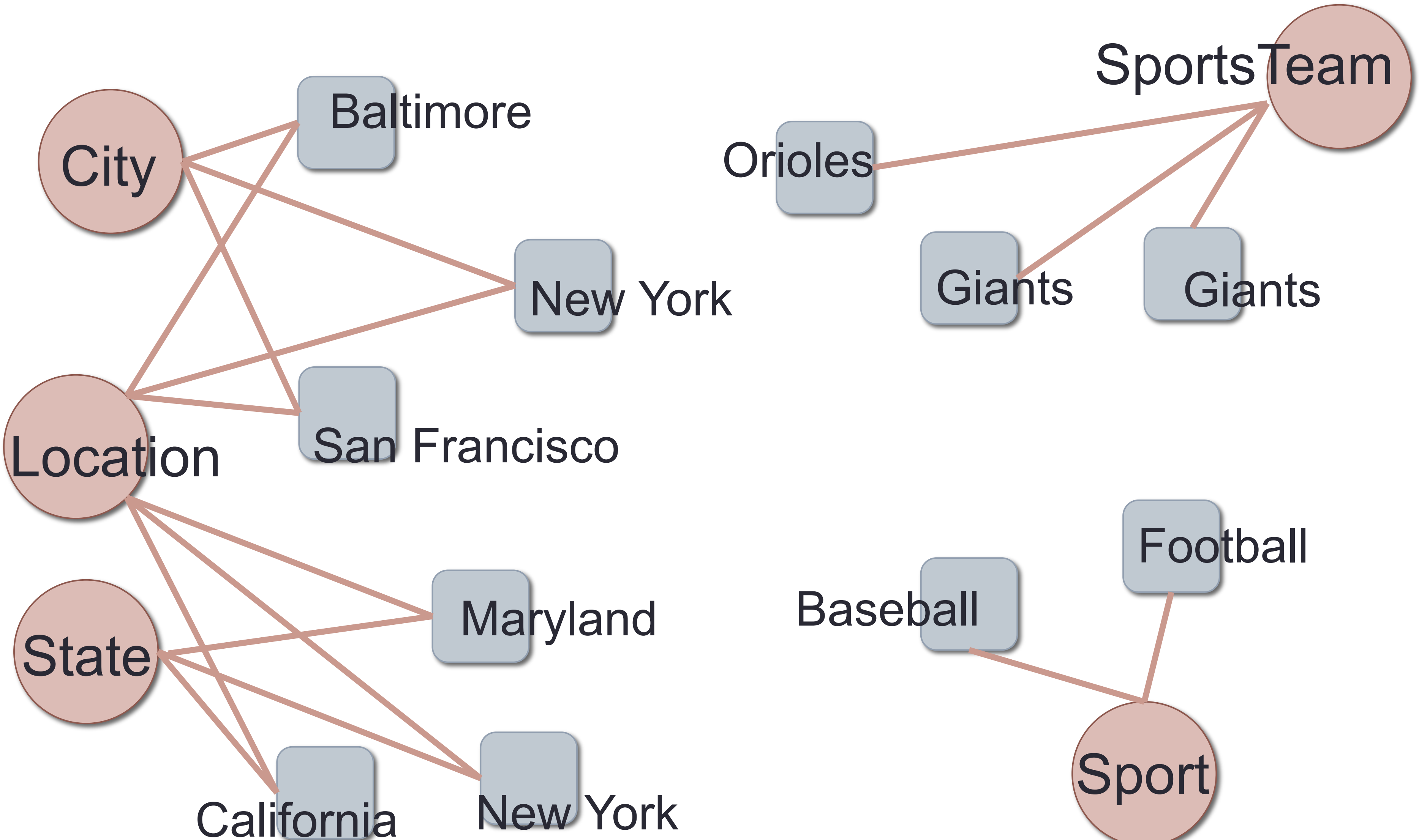
UNIVERSITY OF MARYLAND 1856

Workshop on Automatic Knowledge Base Construction
10/27/2013

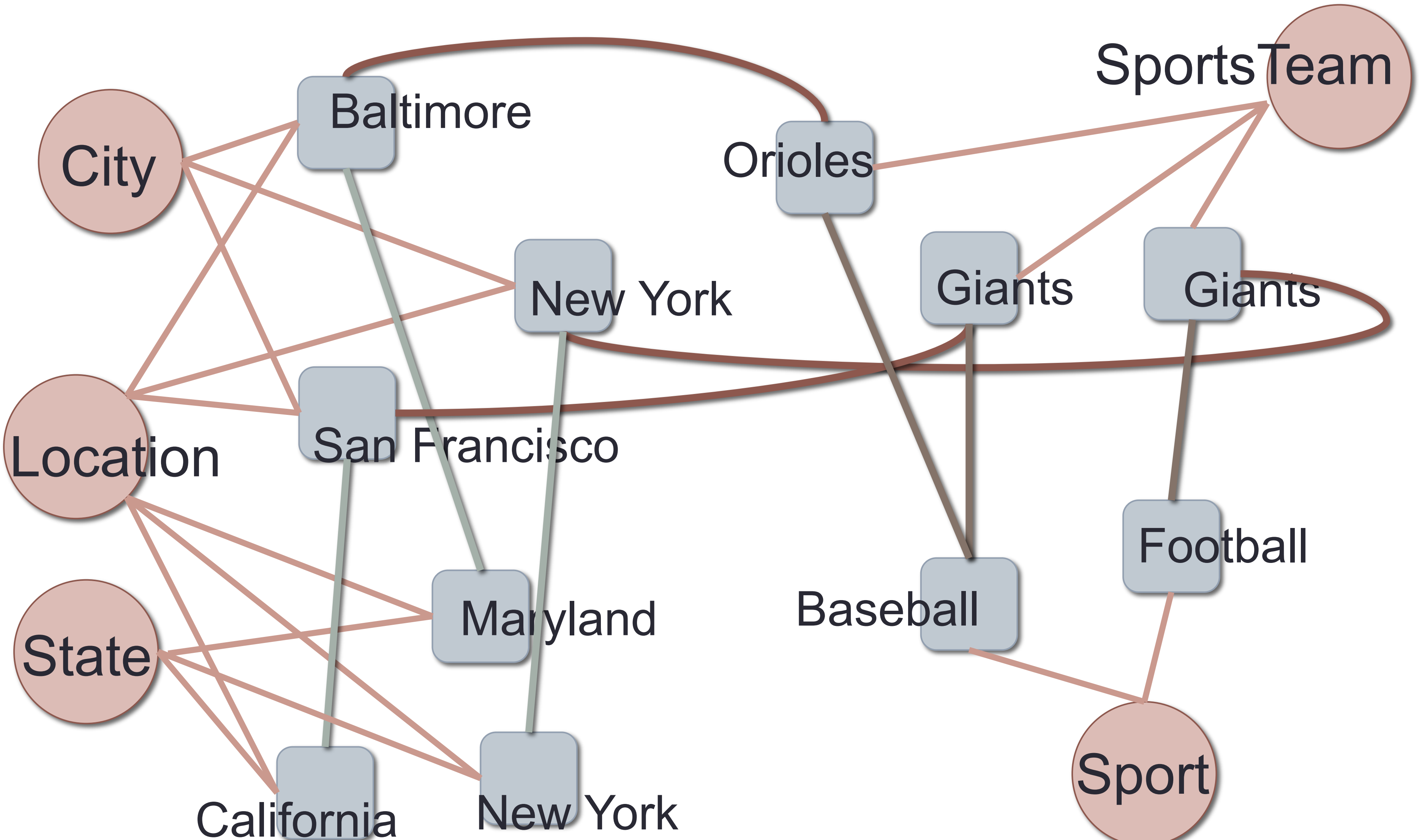
Knowledge Graph Ingredients: Entities



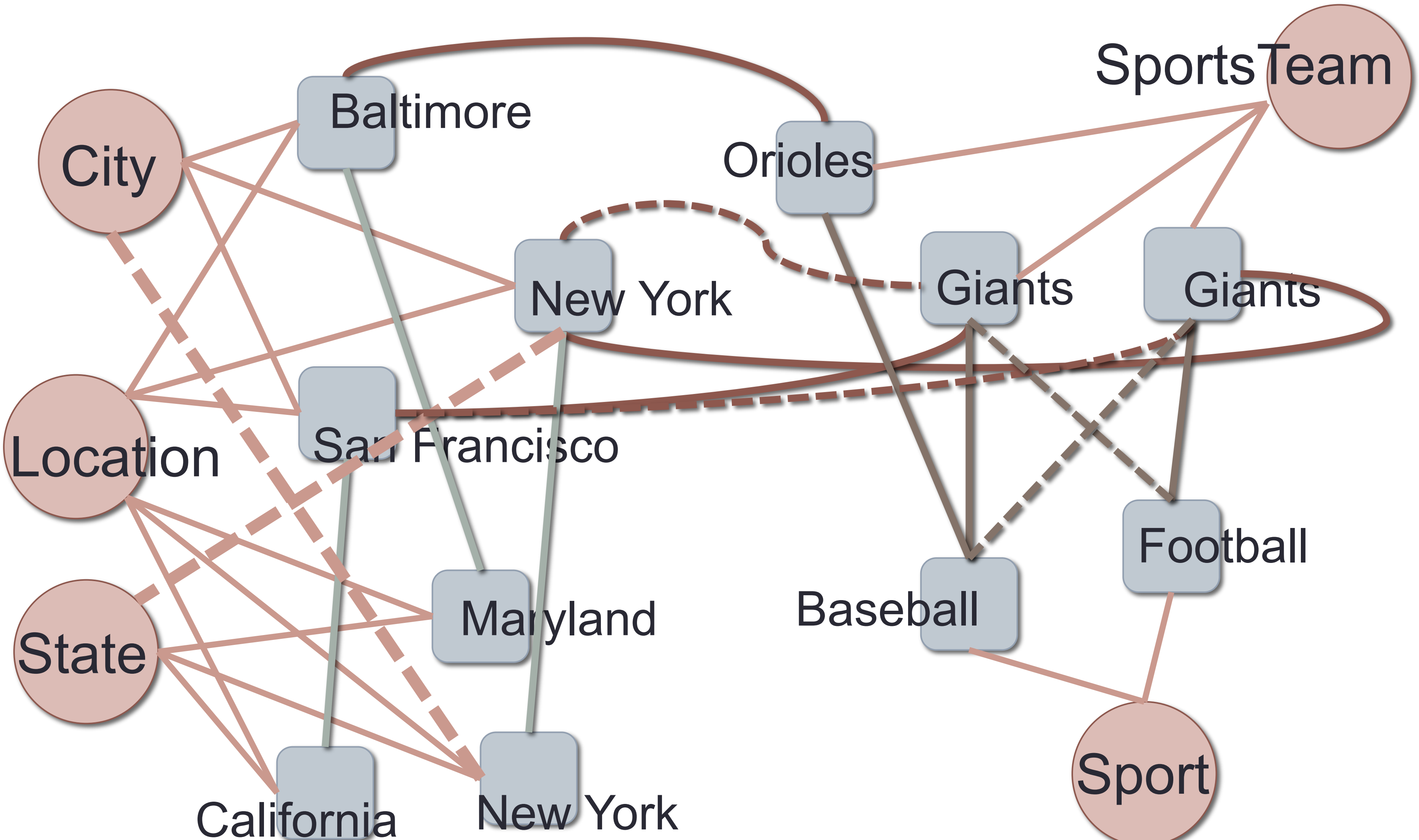
Knowledge Graph Ingredients: Labels



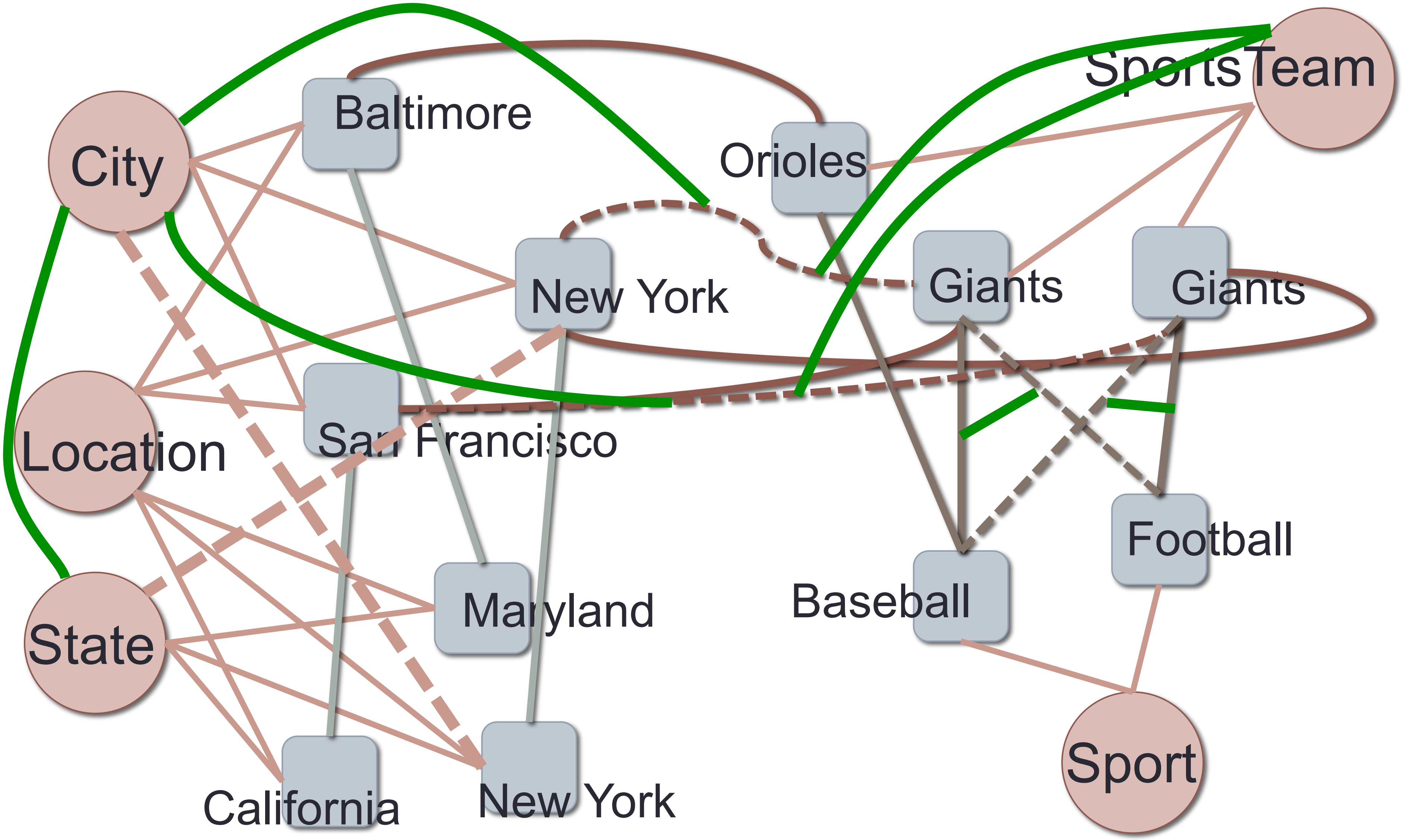
Knowledge Graph Ingredients: Relations



Knowledge Graph Nuisance: Noise!



Knowledge Graph Ingredients: Ontology



Ontological rules for Knowledge Graphs

Inverse:

$$\mathbf{w_O} : \textsc{Inv}(R, S) \;\tilde{\wedge}\; \textsc{Rel}(E_1, E_2, R) \Rightarrow \textsc{Rel}(E_2, E_1, S)$$

Selectional Preference:

$$\mathbf{w_O} : \textsc{Dom}(R, L) \;\tilde{\wedge}\; \textsc{Rel}(E_1, E_2, R) \Rightarrow \textsc{Lbl}(E_1, L)$$

$$\mathbf{w_O} : \textsc{Rng}(R, L) \;\tilde{\wedge}\; \textsc{Rel}(E_1, E_2, R) \Rightarrow \textsc{Lbl}(E_2, L)$$

Subsumption:

$$\mathbf{w_O} : \textsc{Sub}(L, P) \;\tilde{\wedge}\; \textsc{Lbl}(E, L) \Rightarrow \textsc{Lbl}(E, P)$$

$$\mathbf{w_O} : \textsc{RSub}(R, S) \;\tilde{\wedge}\; \textsc{Rel}(E_1, E_2, R) \Rightarrow \textsc{Rel}(E_1, E_2, S)$$

Mutual Exclusion:

$$\mathbf{w_O} : \textsc{Mut}(L_1, L_2) \;\tilde{\wedge}\; \textsc{Lbl}(E, L_1) \Rightarrow \tilde{\neg}\textsc{Lbl}(E, L_2)$$

$$\mathbf{w_O} : \textsc{RMut}(R, S) \;\tilde{\wedge}\; \textsc{Rel}(E_1, E_2, R) \Rightarrow \tilde{\neg}\textsc{Rel}(E_1, E_2, S)$$

Adapted from Jiang et al., ICDM 2012

Knowledge Graph Identification (KGI)

- Joint inference over possible knowledge graphs
  - Resolves co-referent entities
  - Removes spurious labels and relations
  - Infers missing labels and relations
  - Uses many uncertain sources
  - Enforces ontological constraints

Pujara, Miao, Getoor, Cohen, "Knowledge Graph Identification" International Semantic Web Conference, 2013

KGI: Under the Hood

- Define a probability distribution over knowledge graphs:

$$P(G \mid D) = \frac{1}{Z} \exp\left[-\sum_{r \in R} w_r \varphi_r(G) \right]$$

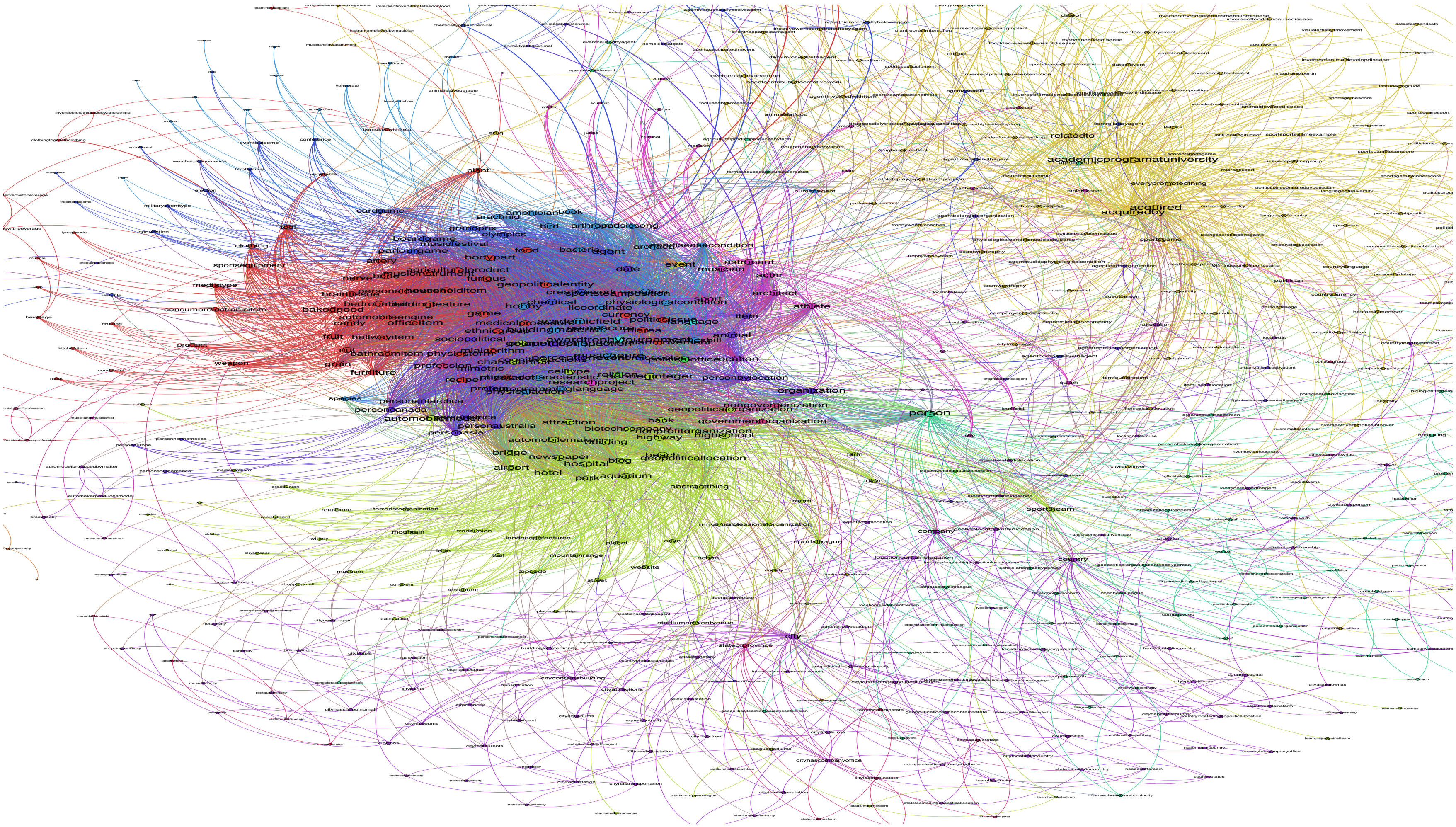
Grounding of logical rule in model

weighted potential takes the form of a hinge-loss

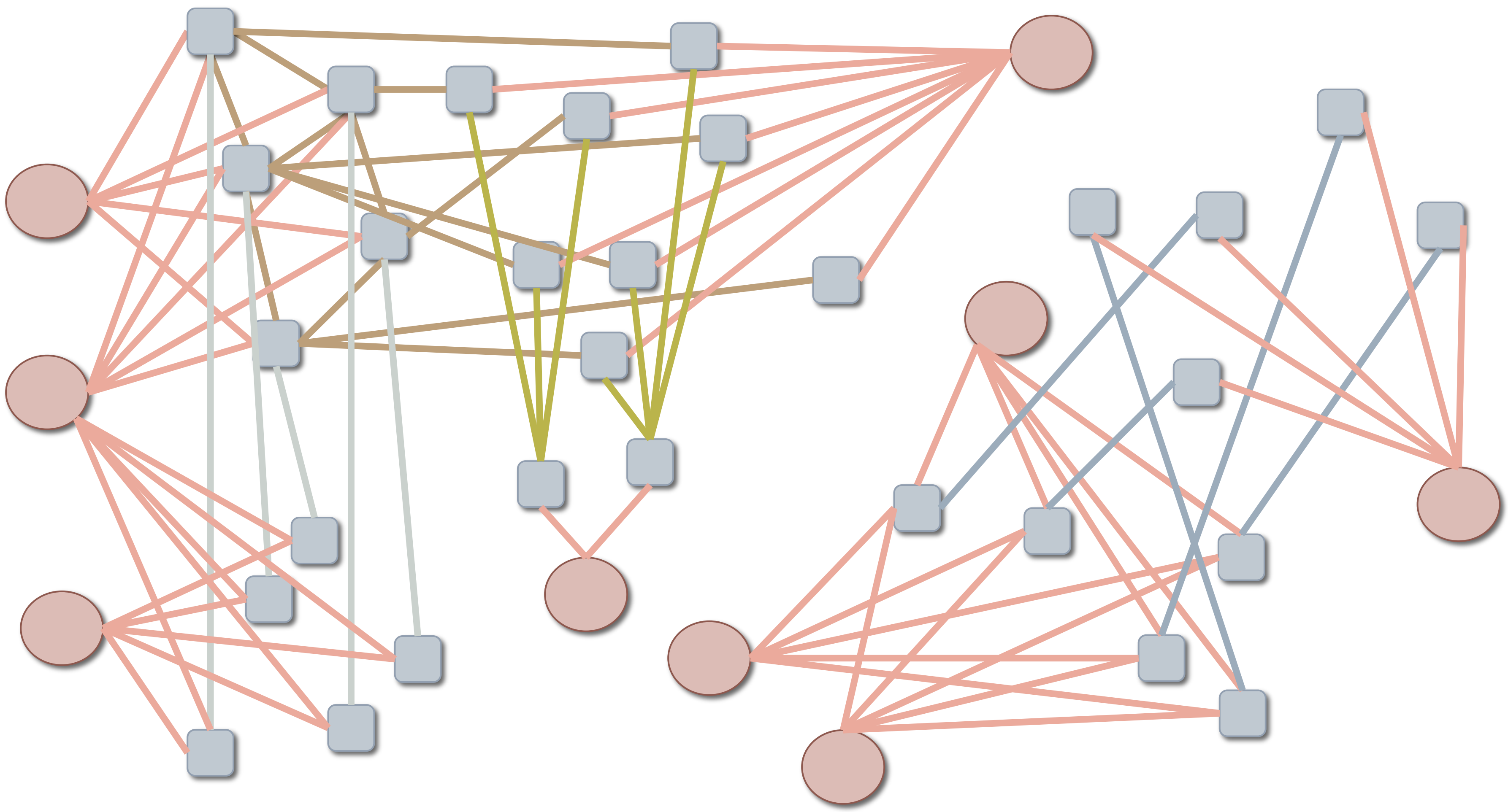
- Hinge-Loss Markov Random Fields
  - Templated: easy to define with probabilistic soft logic
  - Continuous: atoms have continuous truth values in [0,1] range
  - Efficient: inference via convex optimization in O(|R|)
- Superior speed and quality for KGI (Pujara, ISWC13)

Pujara, Miao, Getoor, Cohen, "Knowledge Graph Identification" International Semantic Web Conference, 2013

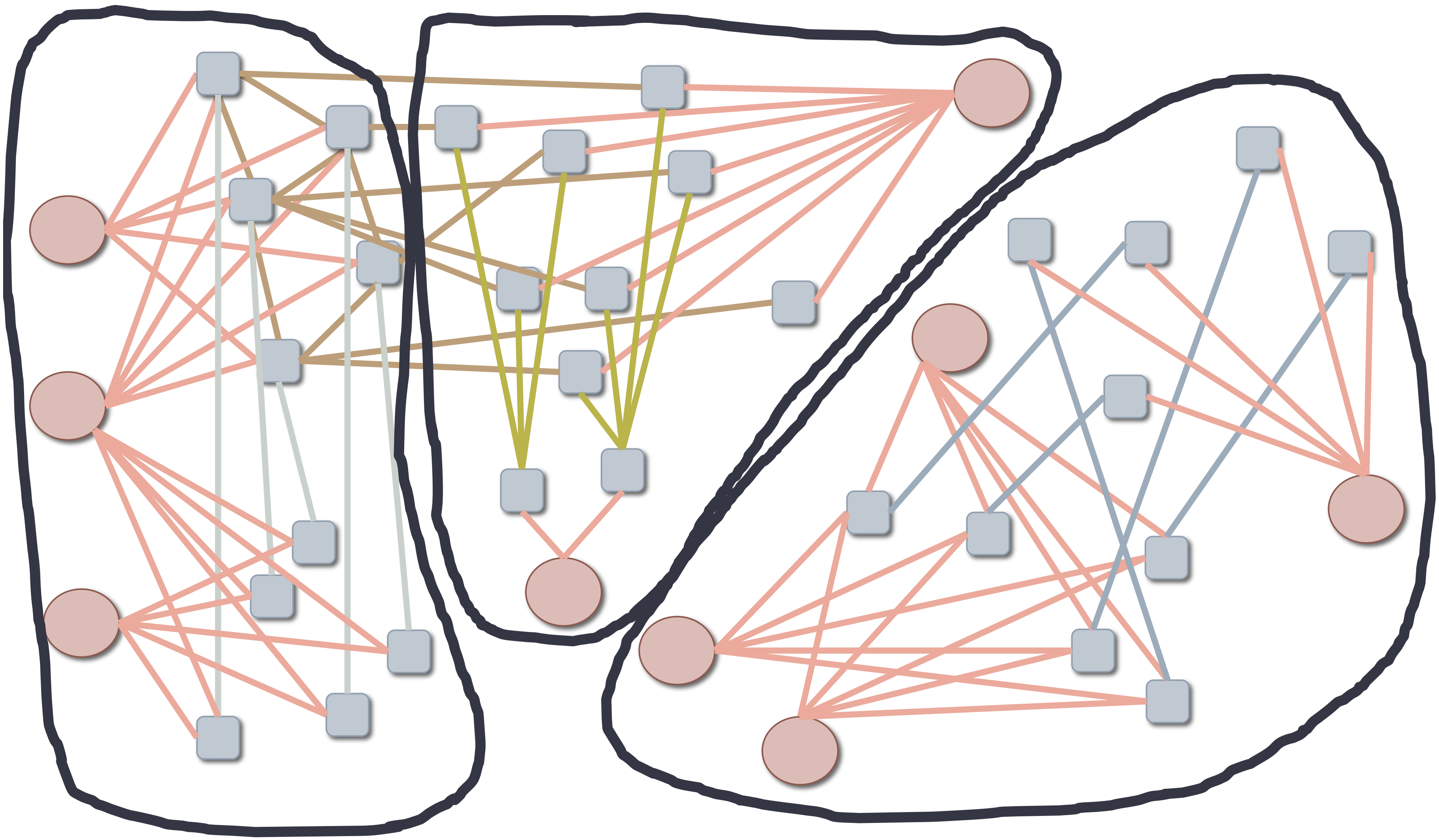
Problem: Knowledge Graphs are HUGE



Problem: Knowledge Graphs are HUGE



Solution: Partition the Knowledge Graph



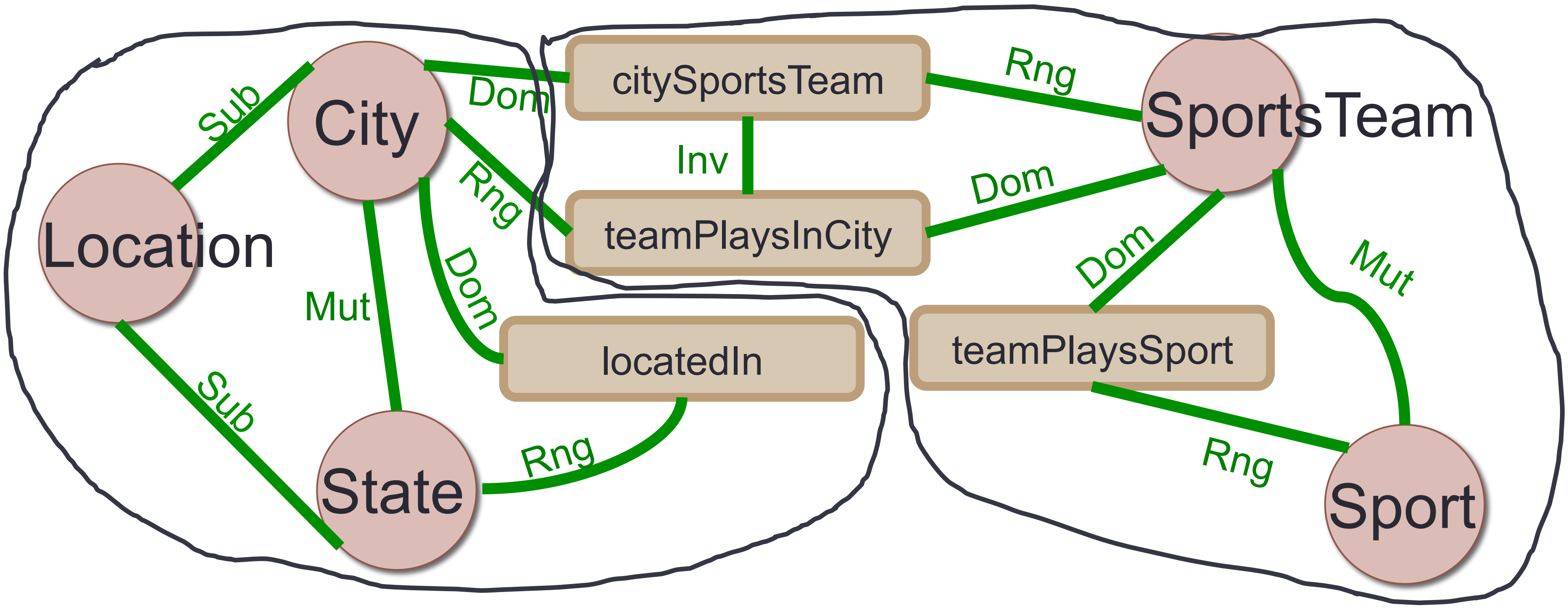
Partitioning: advantages and drawbacks

- Advantages
  - Smaller problems
  - Parallel Inference
  - Speed / Quality Tradeoff

- Drawbacks
  - Partitioning large graph time-consuming
  - Key dependencies may be lost
  - New facts require re-partitioning

Key idea: Ontology-aware partitioning

- Partition the *ontology* graph, not the knowledge graph



- Induce a partitioning of the knowledge graph based on the ontology partition

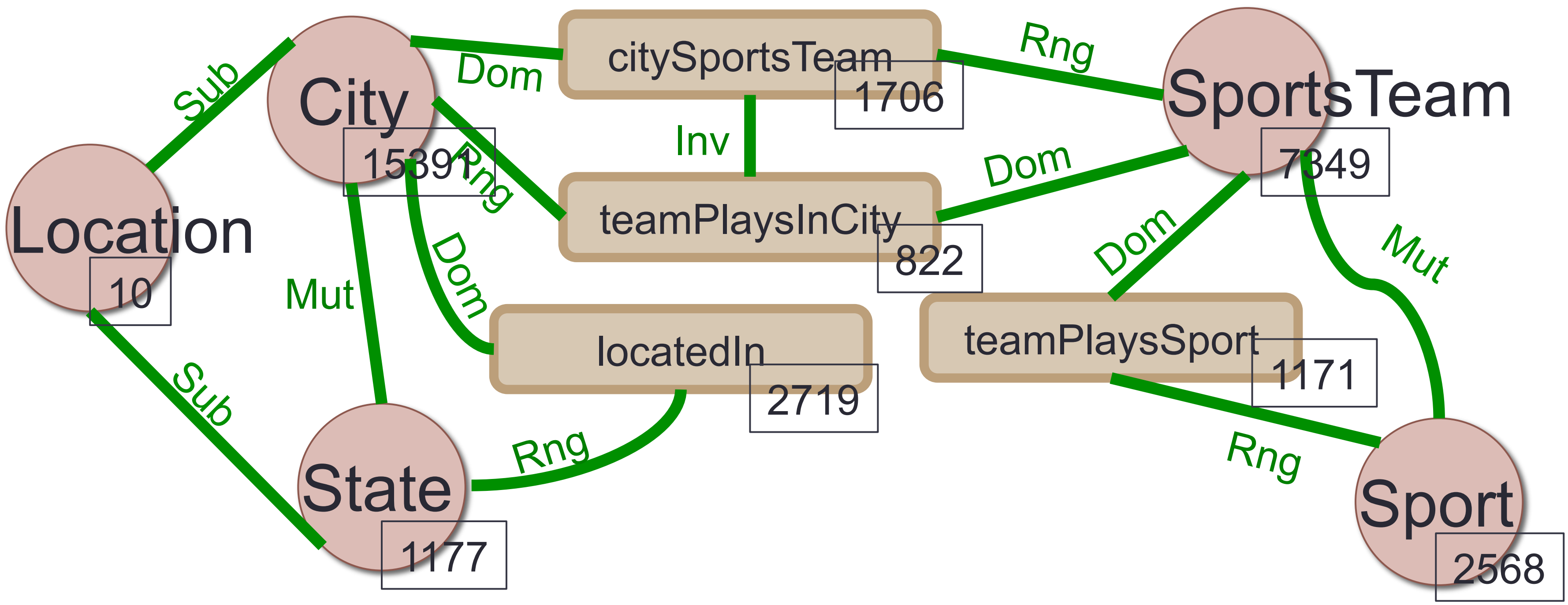
Considerations: Ontology-aware Partitions

- Advantages:
  - Ontology is a smaller graph
  - Ontology coupled with dependencies
  - New facts can reuse partitions

- Disadvantages:
  - Insensitive to data distribution
  - All dependencies treated equally

Refinement: include data frequency

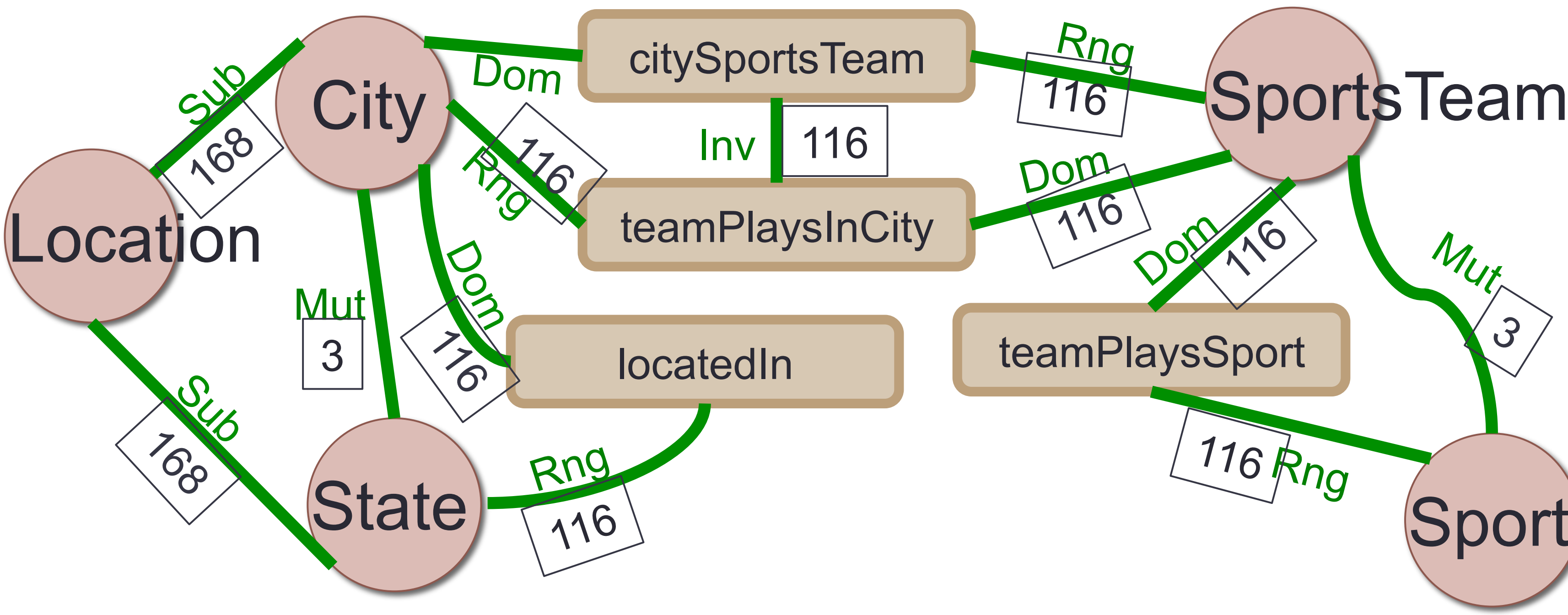
- Annotate each ontological element with its frequency



- Partition ontology with constraint of equal vertex weights

Refinement: weight edges by type

- Weight edges by their ontological importance



Datasets & Metrics

- Data from Never-ending Language Learner (NELL) from iteration 165
- Consists of over 1M extractions and a rich ontology
- Evaluation set from (Jiang, ICDM12) with 4.5K labeled extractions
- Report AUC-PR and running time, optimization terms of slowest partition

Inputs	
Candidate Labels	1.2M
Candidate Relations	100K
Types	
Unique Labels	235
Unique Relations	221
Ontology	
Dom	418
Rng	418
Inv	418
Sub	288
RSub	461
Mut	17.4K
RMut	48.5K

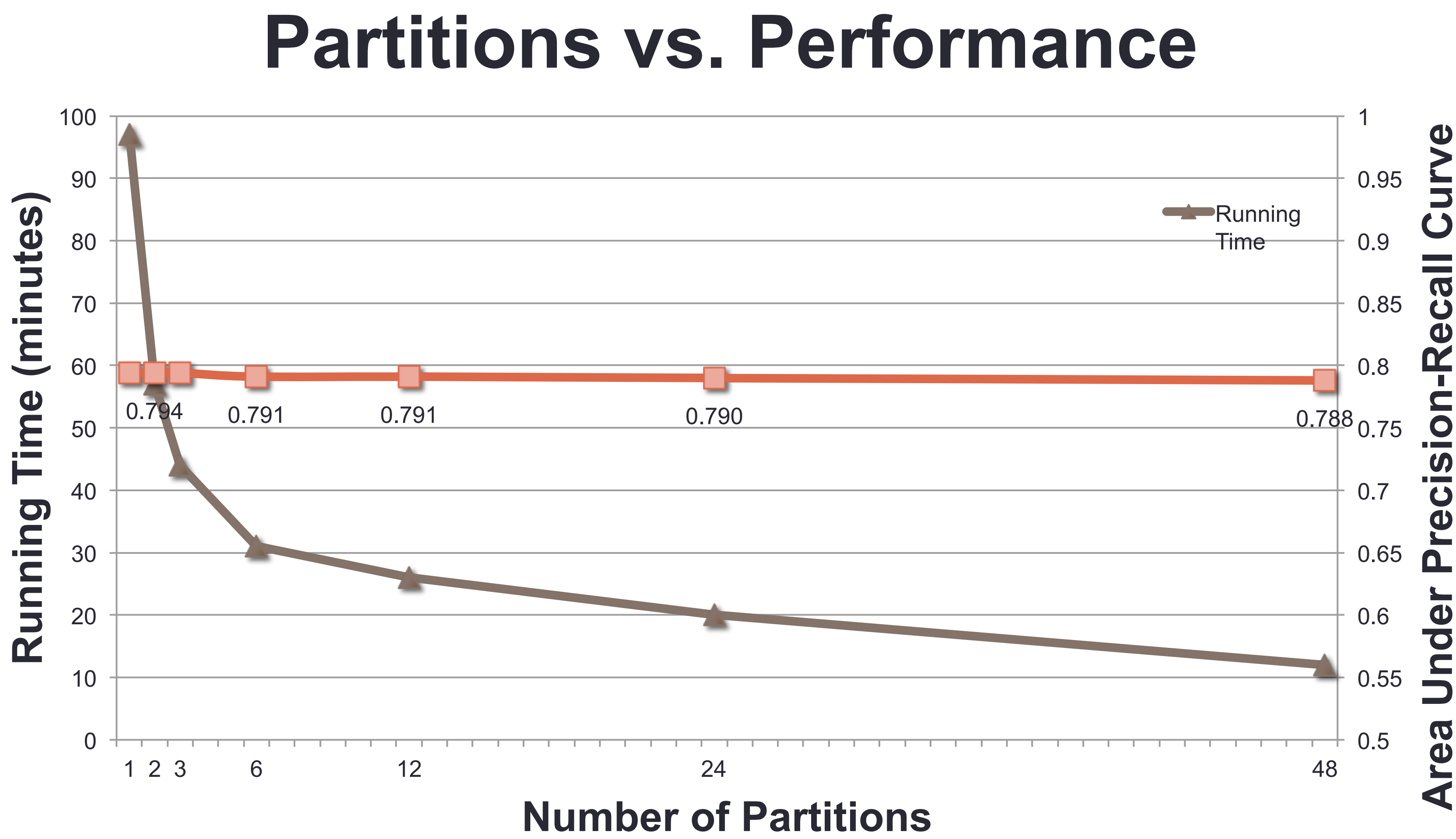
Experiments: Partitioning Approaches

Comparisons (6 partitions):

NELL	Default promotion strategy, no KGI
KGI	No partitioning, full knowledge graph model
baseline	KGI, Randomly assign extractions to partition
Ontology	KGI, Edge min-cut of ontology graph
O+Vertex	KGI, Weight ontology vertices by frequency
O+V+Edge	KGI, Weight ontology edges by inv. frequency

	AUPRC	Running Time	Opt. Terms
NELL	0.765	-	
KGI	**0.794**	97	10.9M
baseline	0.780	**31**	3.0M
Ontology	0.788	42	4.2M
O+Vertex	**0.791**	**31**	3.7M
O+V+Edge	0.790	**31**	3.7M

Experiments: Scalability (Ont+Vertex)



Conclusion

- Knowledge Graph Identification: a powerful technique for constructing consistent knowledge graphs…
- Scalability is still a concern for real-world applications

- Partitioning can address scalability concerns
- Ontology-aware partitioning partitions the *ontology* instead of the knowledge graph
- Including data distribution information balances partitions

- Results on the NELL dataset show this strategy reduces running time from 97 minutes to 12 minutes without significant AUC degradation

Code Available on GitHub:

https://github.com/linqs/KnowledgeGraphIdentification