Knowledge Graph Completion

Mayank Kejriwal (USC/ISI)

What is knowledge graph completion?

- An ‘intelligent’ way of doing data cleaning
  - Deduplicating entity nodes (entity resolution)
  - Collective reasoning (probabilistic soft logic)
  - Link prediction
  - Dealing with missing values
  - Anything that improves an existing knowledge graph!
- Also known as knowledge base identification

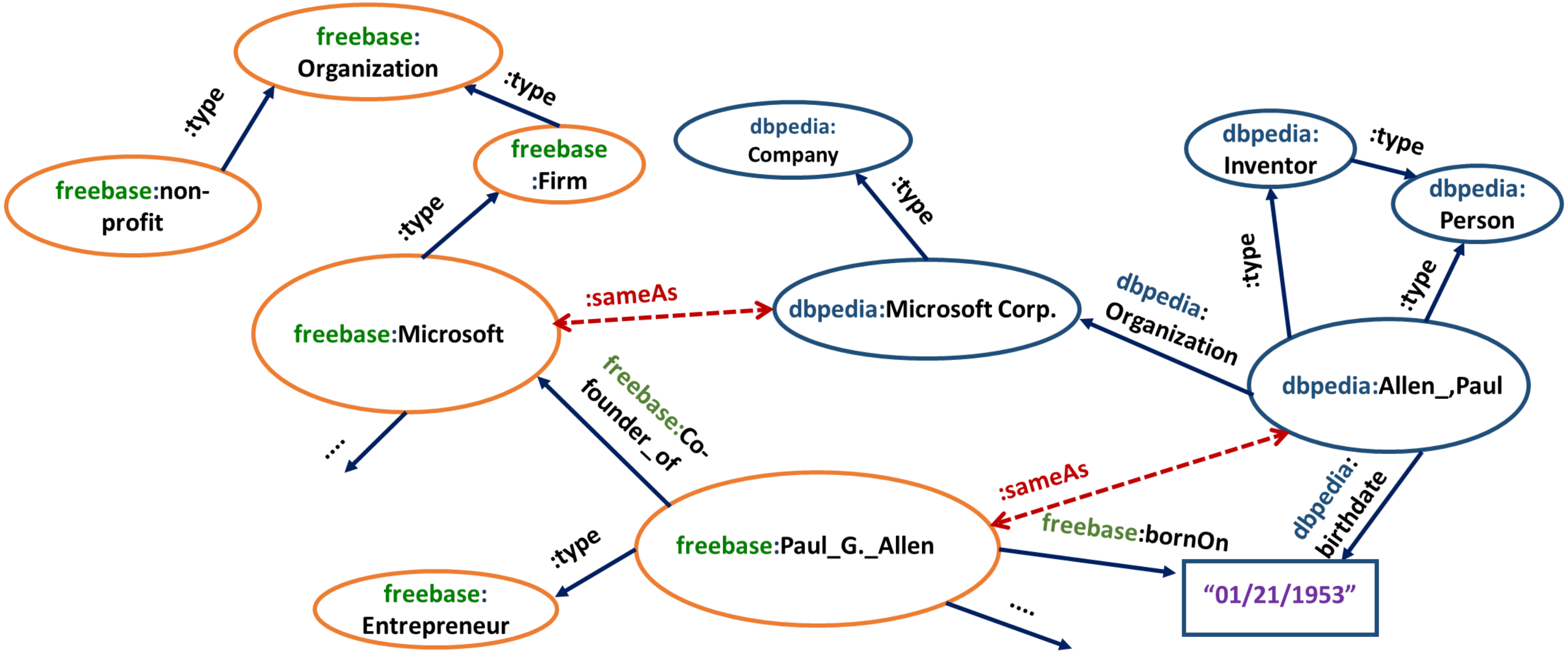
Some solutions we’ll cover today

- Entity Resolution (ER)
- Probabilistic Soft Logic (PSL)
- Knowledge Graph Embeddings (KGEs), with applications

Entity Resolution (ER)

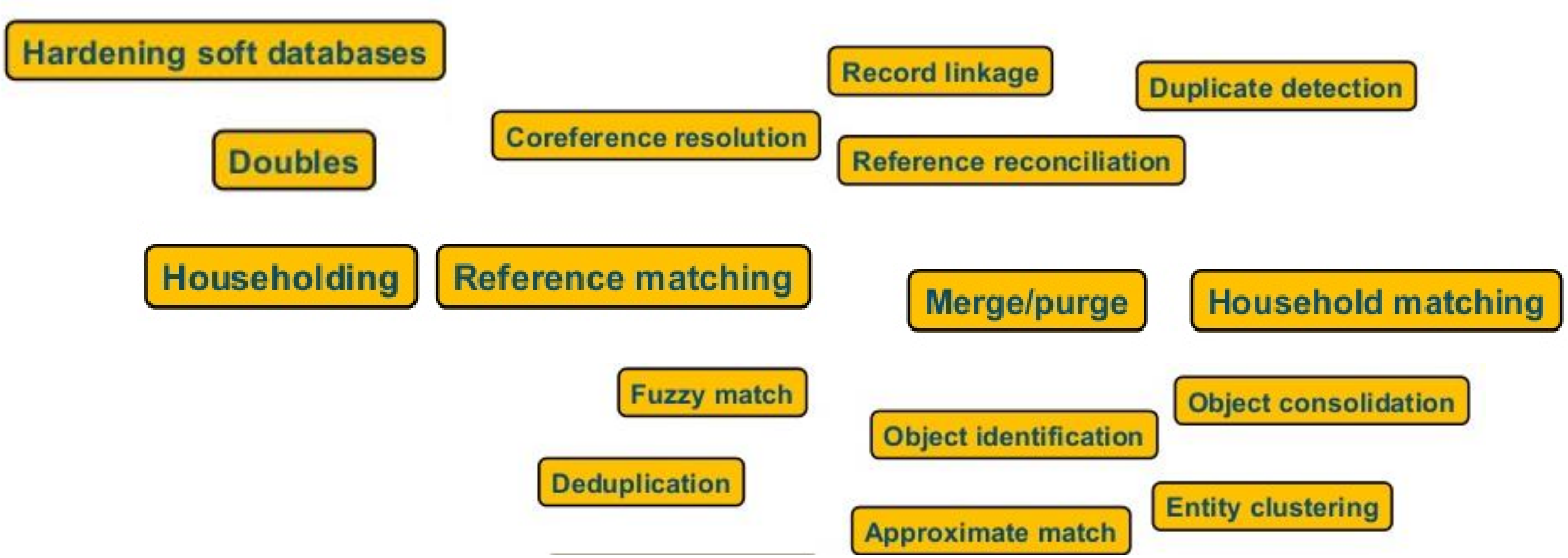
Entity Resolution (ER)

- The algorithmic problem of grouping entities referring to the same underlying entity



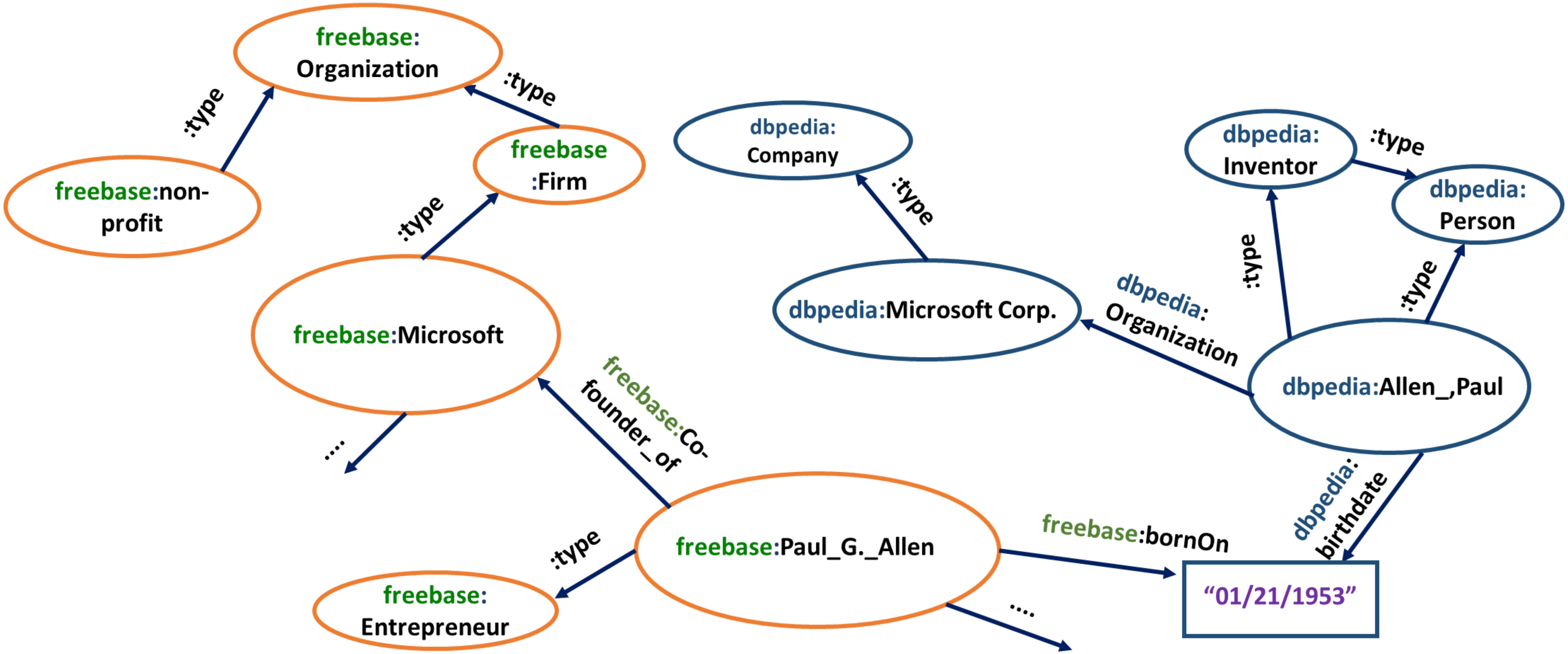
Aside: Resolving Entity Resolution

- Itself has many alternate names in the research community!

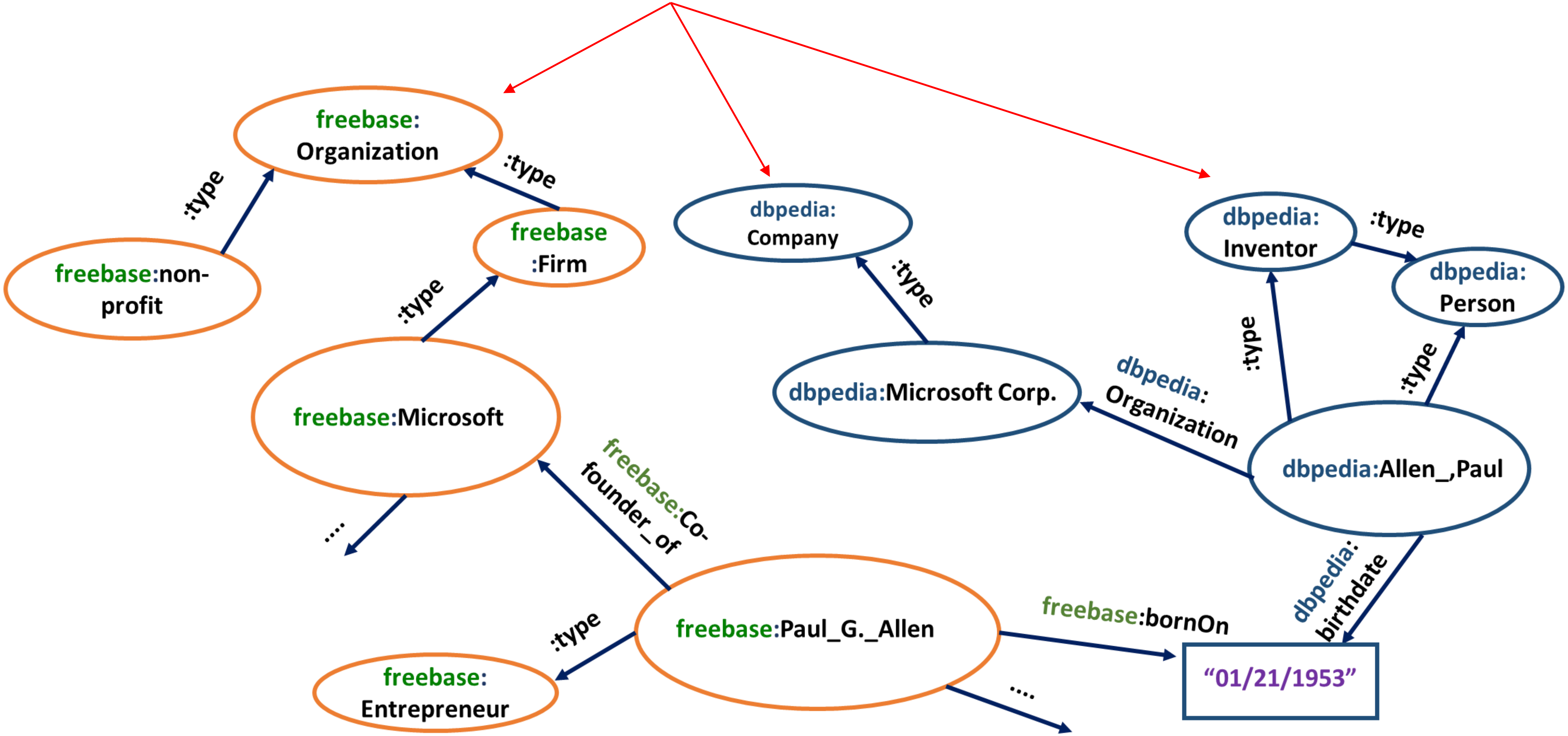


**Many thanks to Lise Getoor*

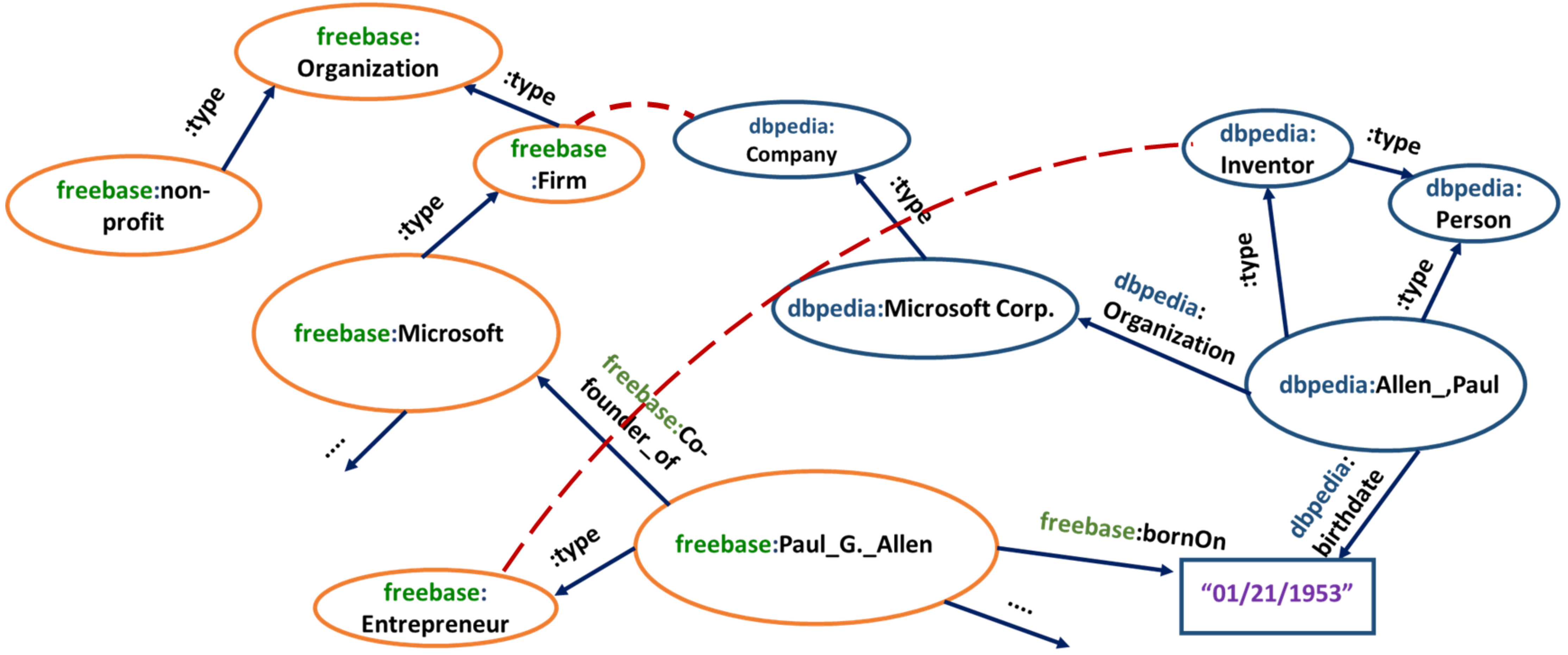
ER is less constrained for graphs than tables (why?)



KG nodes are multi-type



Two KGs may be published under different ontologies



How to do ER?

- Popular methods use some form of machine learning; see surveys by Kopcke and Rahm (2010), Elmagarmid et al. (2007), Christophides et al. (2015)

Probabilistic Matching Methods

Supervised, Semi-supervised

Marlin (SVM based)
Bilenko and Mooney (2003)

Active Learning

Rule Based

Distance Based

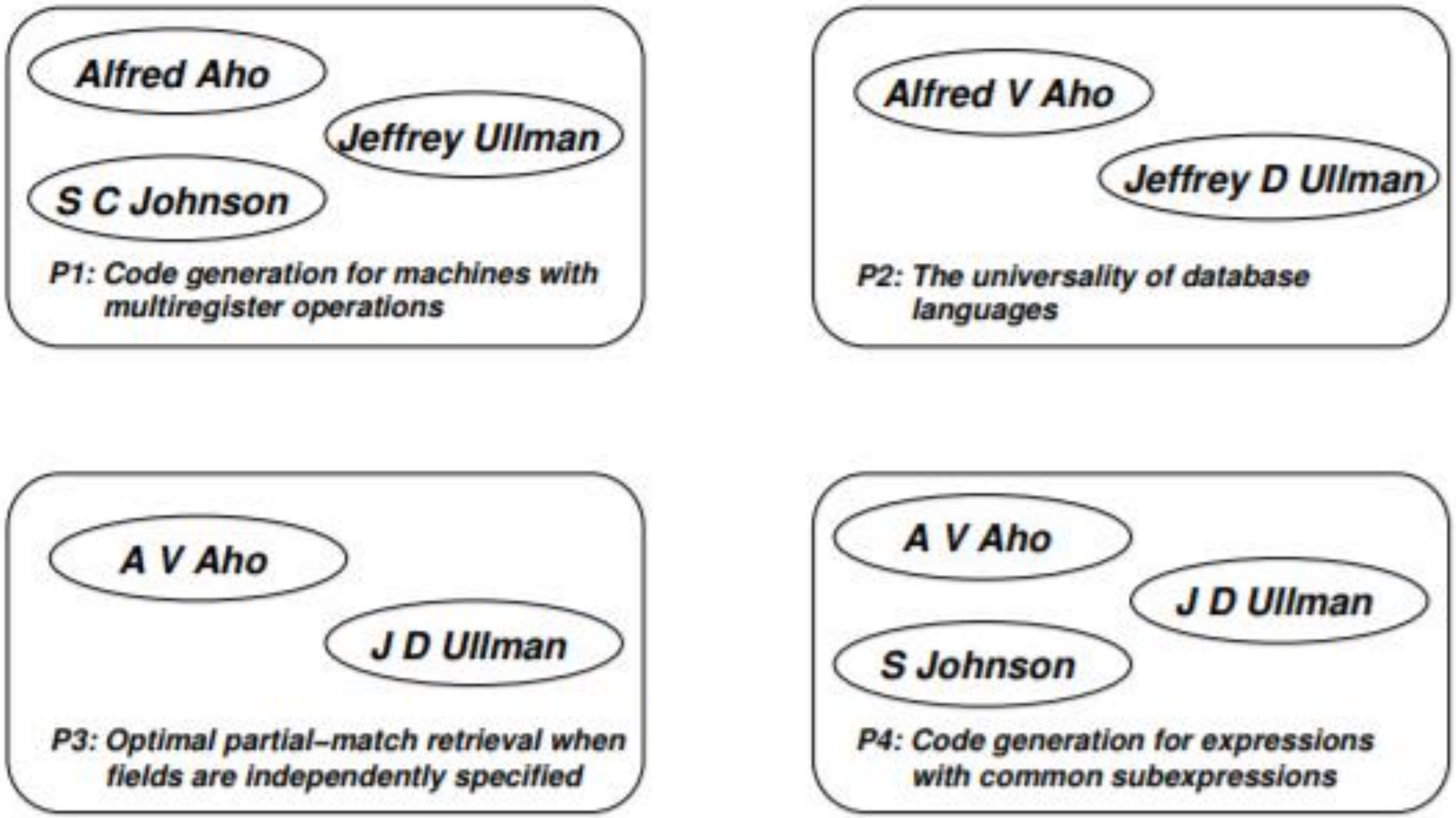
Unsupervised

EM
Winkler (1993)
Hierarchical Graphical Models
Ravikumar and Cohen (2004)
SVM
Christen (2008)

With graph representation

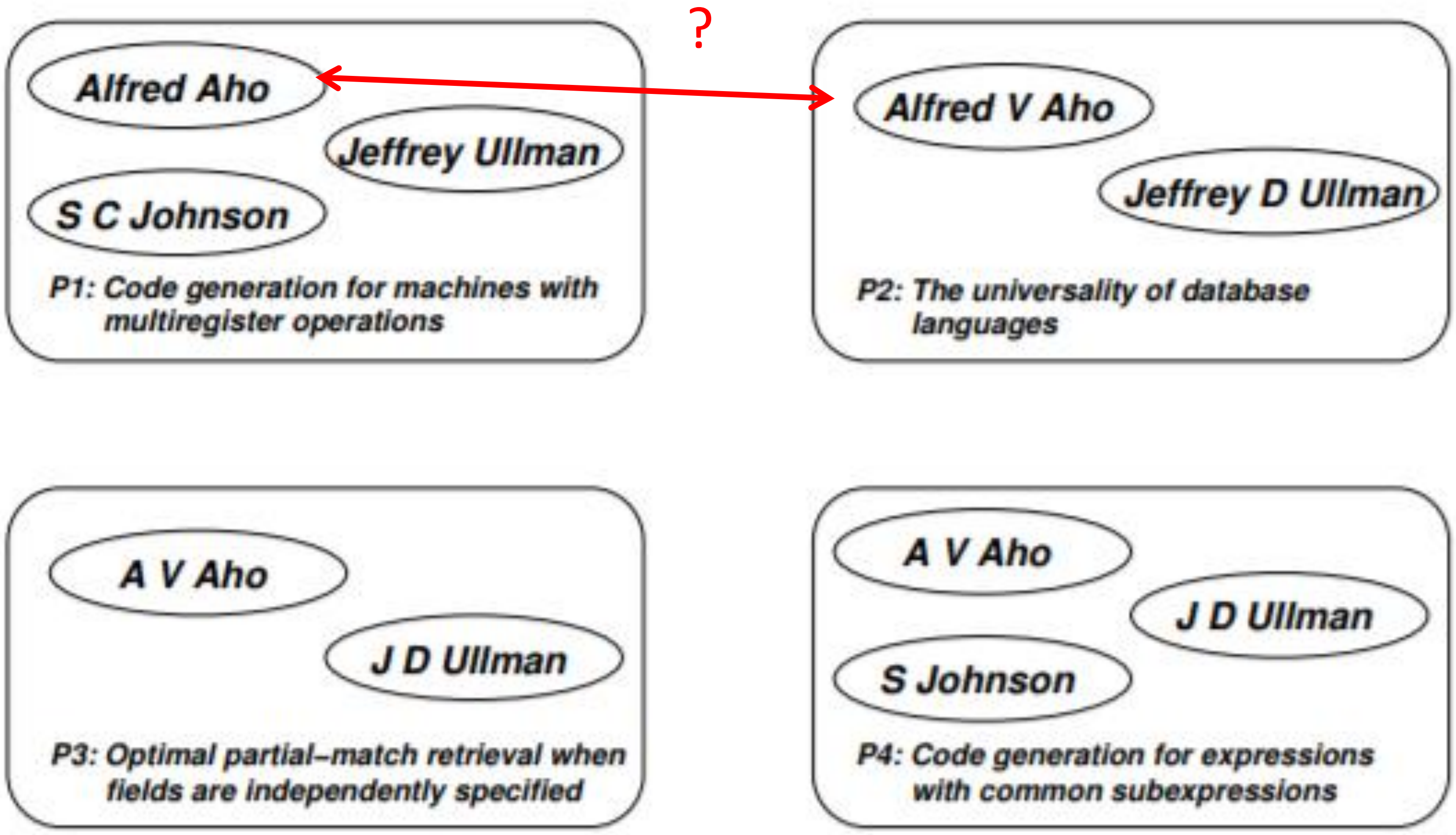
- Can propagate similarity decisions Melnik, Garcia-Molina and Rahm (2002)
  - More expensive but better performance
- Can be generic or use domain knowledge e.g., citation/bibliography domain Bhattacharya and Getoor (2006,2007)

Example (co-authorship)



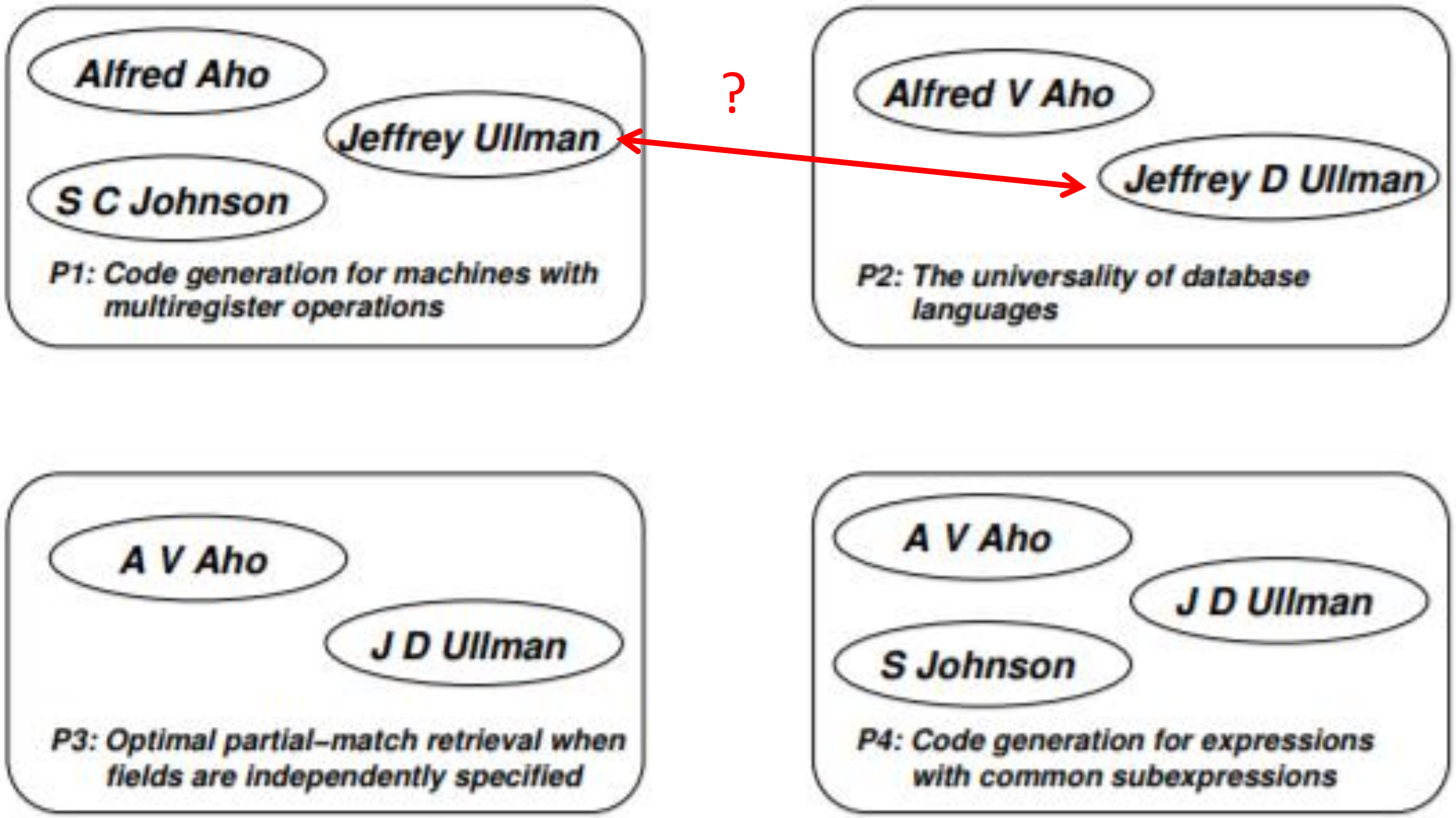
- Bhattacharya and Getoor (2006,2007)

Example (co-authorship)



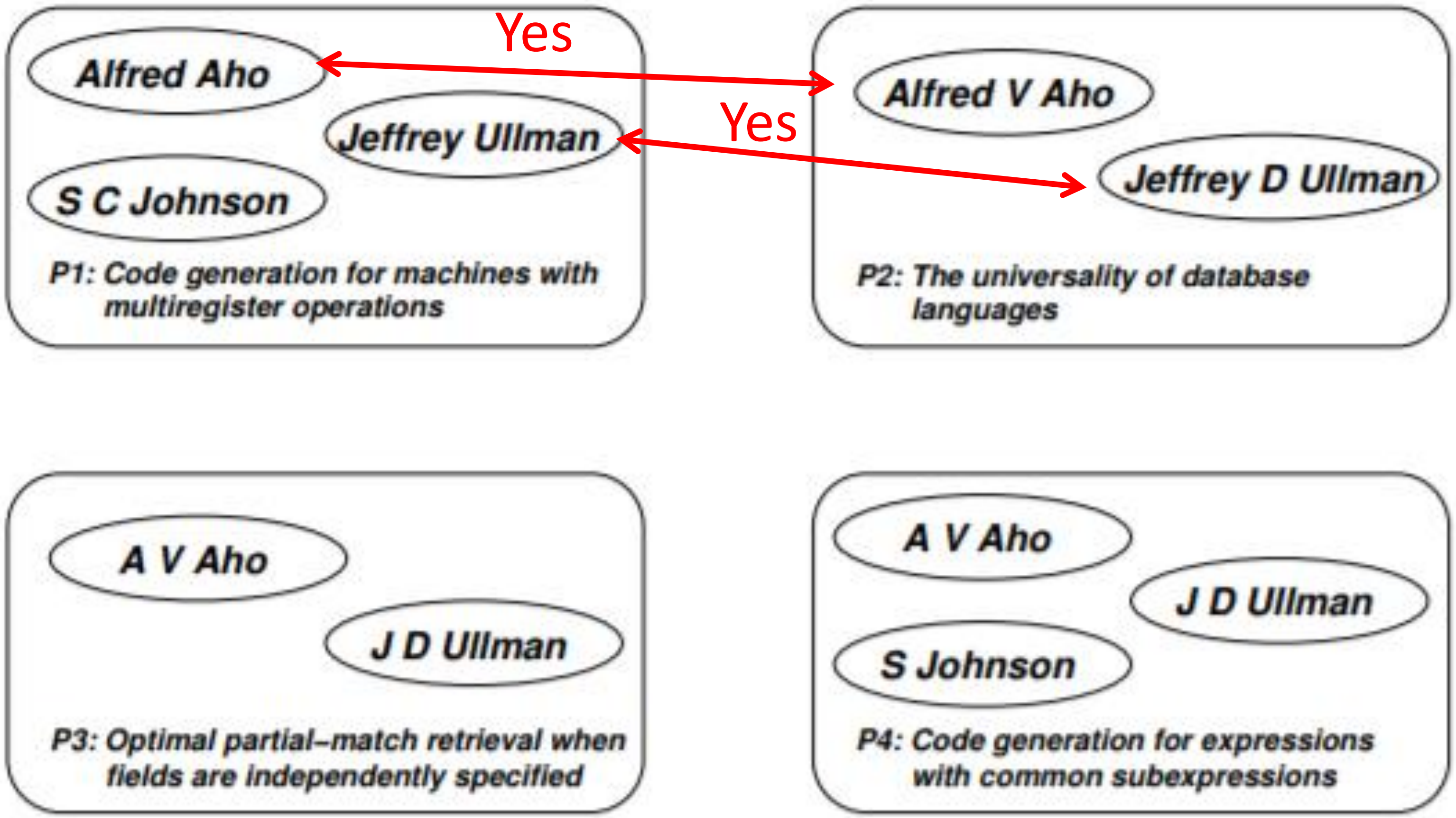
- Bhattacharya and Getoor (2006,2007)

Example (co-authorship)



- Bhattacharya and Getoor (2006,2007)

Example (co-authorship)



- Bhattacharya and Getoor (2006,2007)

Feature functions - I

- First line of attack is *string matching*

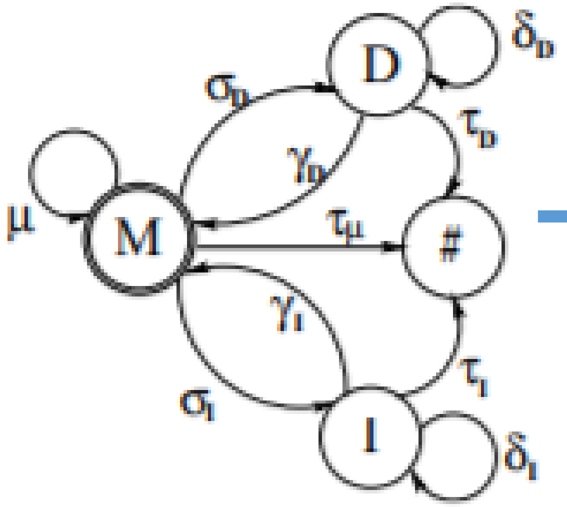
Character based	Token based	Phonetic based
Edit Distance Affine Gap Smith-Waterman Jaro Q-gram	Monge Elkan TF-IDF • Soft • Q-gram Jaccard	Soundex NYSIIS ONCA Metaphone Double Metaphone

Available Packages: SecondString, FEBRL, Whirl…

Learnable string similarity

- Example: adaptive edit distance

Sets of equivalent string pairs (e.g., <Suite 1001, Ste. 1001>

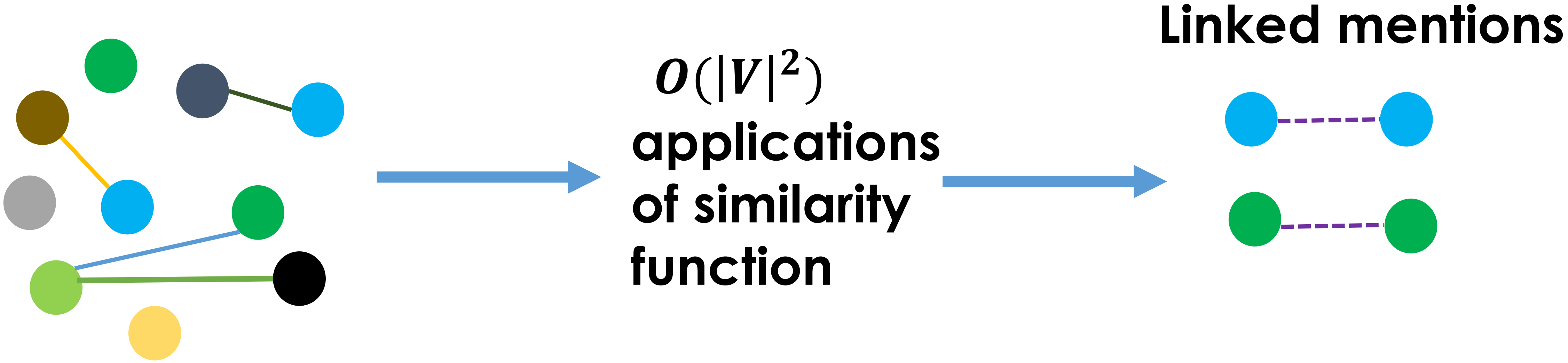


Learned parameters

Bilenko and Mooney (2003)

After training…

- Apply classifier i.e. link specification function to every pair of nodes? Quadratic complexity!



More formally

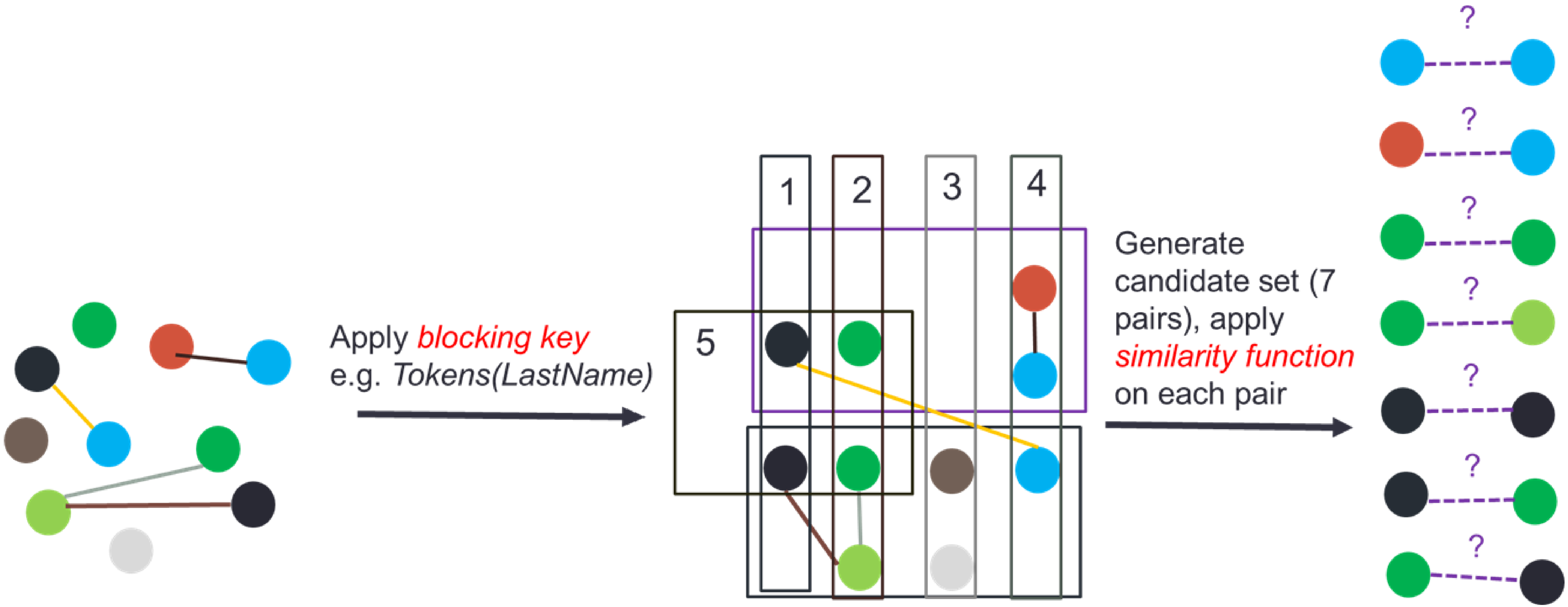
- Input: Two graphs G and H with $|V|$ nodes each, pairwise Link Specification Function (LSF) L
- Naïve algorithm: Apply L on $|V|X|V|$ node pairs, output pairs flagged (possibly probabilistically) by function

Complexity is quadratic: $O(T(L)|V|^2)$

How do we reduce the number of applications of L?

Blocking trick

- Like a configurable inverted index function



What is a good blocking key?

- Achieves high recall
- Achieves high reduction
- Good survey on blocking: Christen (2012)

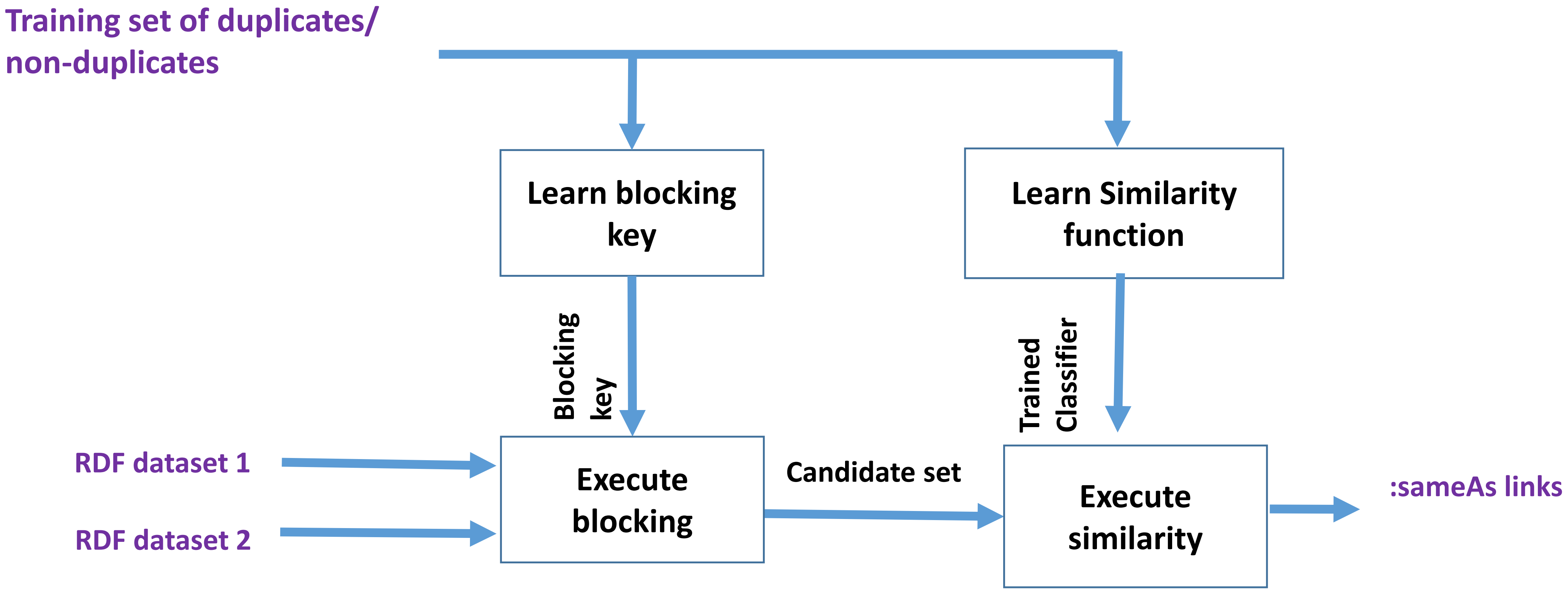
How do we learn a good blocking key?

- Key idea in existing work is to learn a DNF rule with indexing functions as atoms

CharTriGrams(Last_Name) U (Numbers(Address) X Last4Chars(SSN))

Michelson and Knoblock (2006), Bilenko, Kamath and Mooney (2006), Kejriwal and Miranker (2013; 2015)...

Putting it together



Post-processing step: soft transitive closure

- How do we combine :sameAs links into groups of unique entities?
  - Naïve transitive closure might not work due to noise!
- Clustering and 'soft transitive closure' algorithms could be applied
- Not as well-studied for ER
  - Has unique properties! ER is a micro-clustering problem
  - How to incorporate collective reasoning (better-studied)?
  - Efficiency!

ER packages

- Several are available, but some may need tuning to work for RDF
  - FEBRL was designed for biomedical record linkage (Christen, 2008)
  - Dedupe https://github.com/dedupeio/dedupe
  - LIMES, Silk mostly designed for RDF data (Ngonga Ngomo and Auer, 2008; Isele et al. 2010)

Not all attributes are equal

- Phones/emails important in domains like organizations
  - (names are unreliable)
- Names can be important in certain domains
  - (nothing special about phones)
- How do we use this knowledge?

Domain knowledge

- Especially important for unusual domains but how do we express and use it?
  -
  - Use rules? Too brittle, don't always work!
  - Use machine learning? Training data hard to come by, how to encode rule-based intuitions?

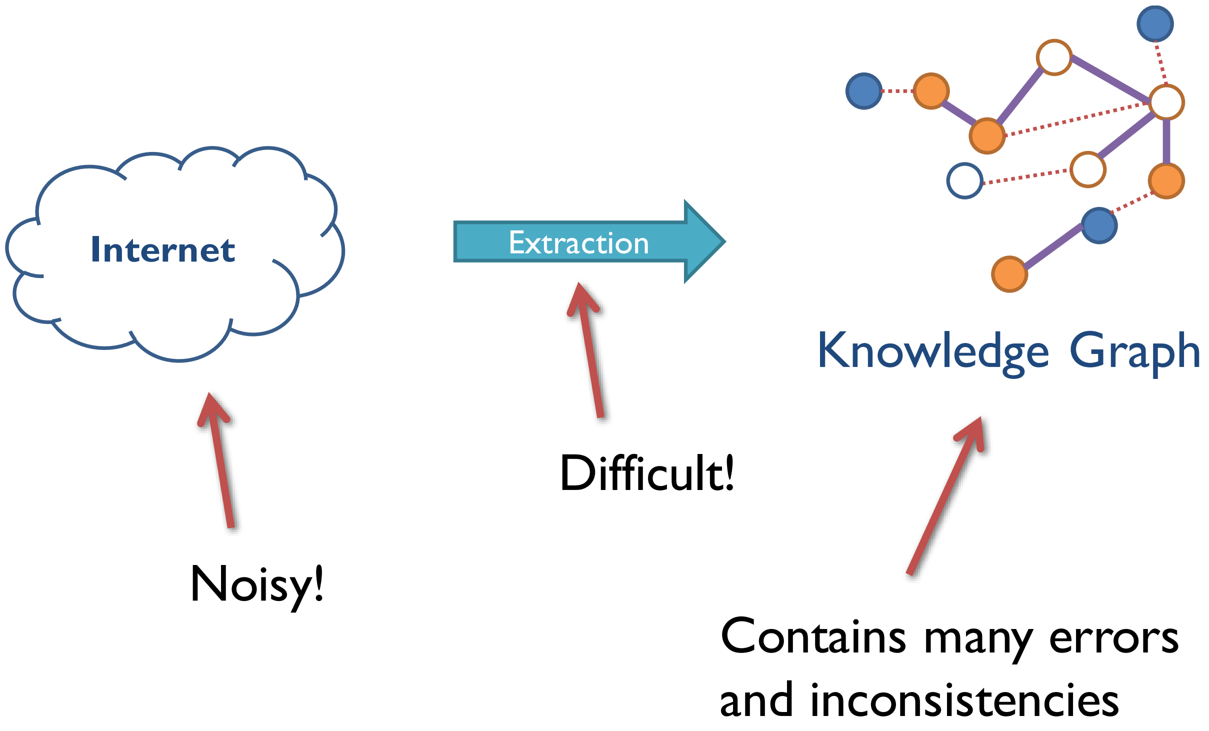
Summary

- Entity Resolution is the first line of attack for the knowledge graph completion problem
- The problem is usually framed in terms of two steps: blocking and similarity (or link specification)
  - Blocking is used for reducing exhaustive pairwise complexity
  - Similarity determines what makes two things the same
  - Both can use machine learning!
- Many open research sub-problems, especially in SW

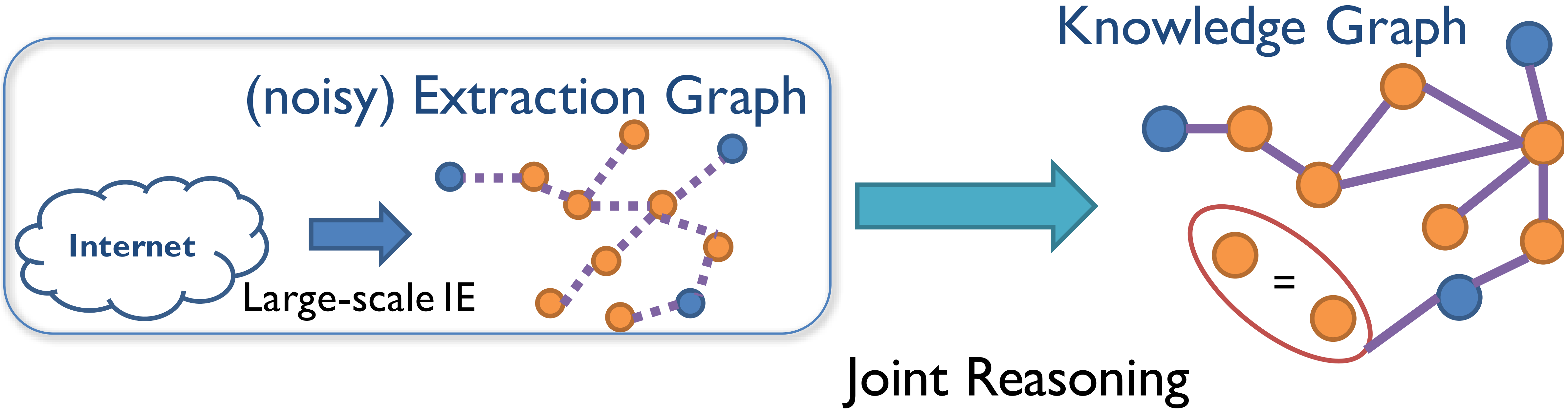
Probabilistic Soft Logic (PSL)

Many thanks to Jay Pujara for his inputs/slides

Collective Reasoning over Noisy Extractions



- Noise in extractions is not random
- Jointly reason over facts and extractions to converge to the most probable extractions
- Use a combination of logic, semantics and machine learning for best performance (but how?)



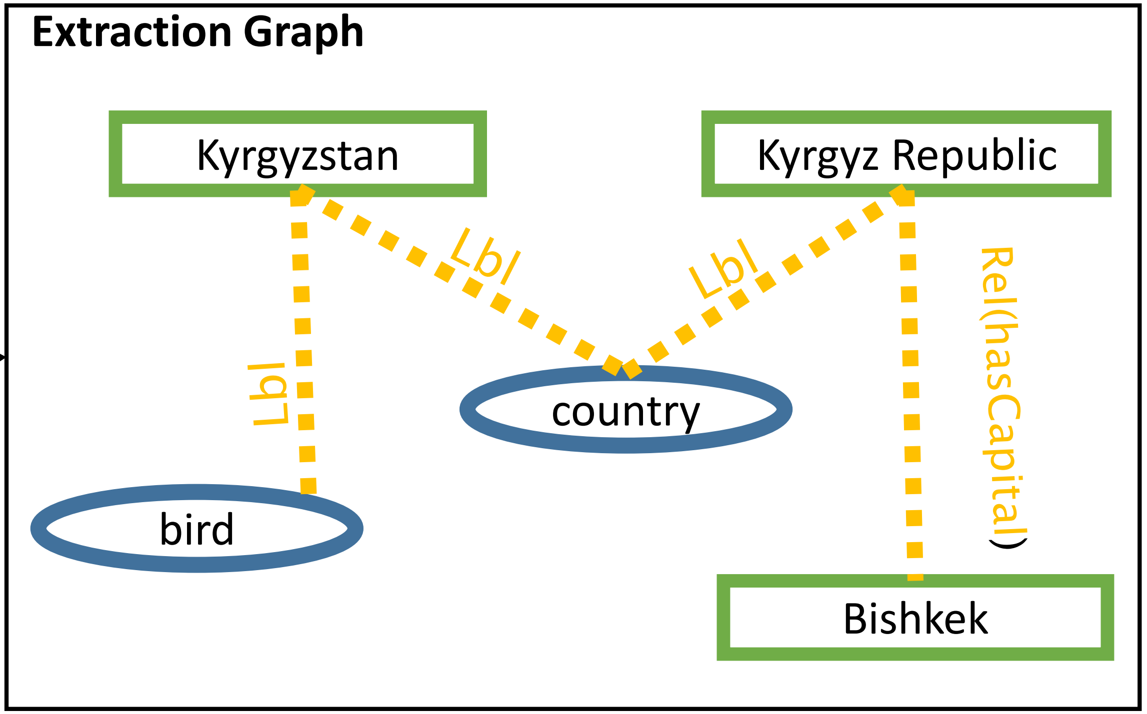
(noisy) Extraction Graph
Internet
Large-scale IE
Knowledge Graph
=
Joint Reasoning

Extraction Graph

Uncertain Extractions:

.5: Lbl(Kyrgyzstan, bird)
.7: Lbl(Kyrgyzstan, country)
.9: Lbl(Kyrgyz Republic, country)

.8: Rel(Kyrgyz Republic, Bishkek, hasCapital)

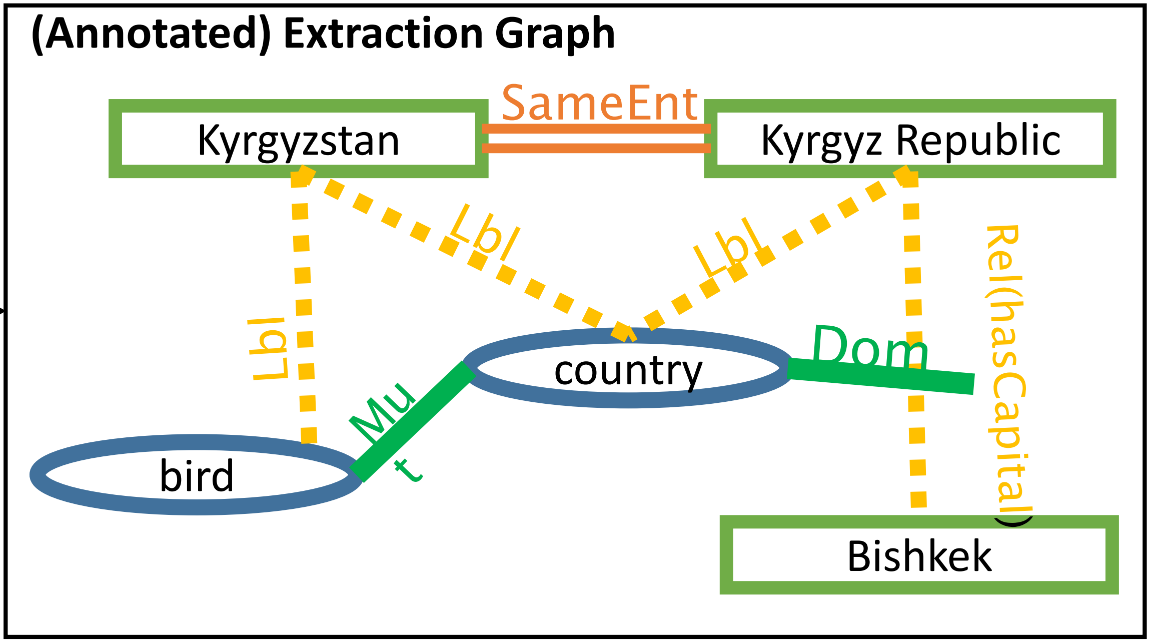


Extraction Graph+Ontology + ER

Uncertain Extractions:
.5: Lbl(Kyrgyzstan, bird)
.7: Lbl(Kyrgyzstan, country)
.9: Lbl(Kyrgyz Republic, country)
.8: Rel(Kyrgyz Republic, Bishkek, hasCapital)

Ontology:
Dom(hasCapital, country)
Mut(country, bird)

Entity Resolution:
SameEnt(Kyrgyz Republic, Kyrgyzstan)



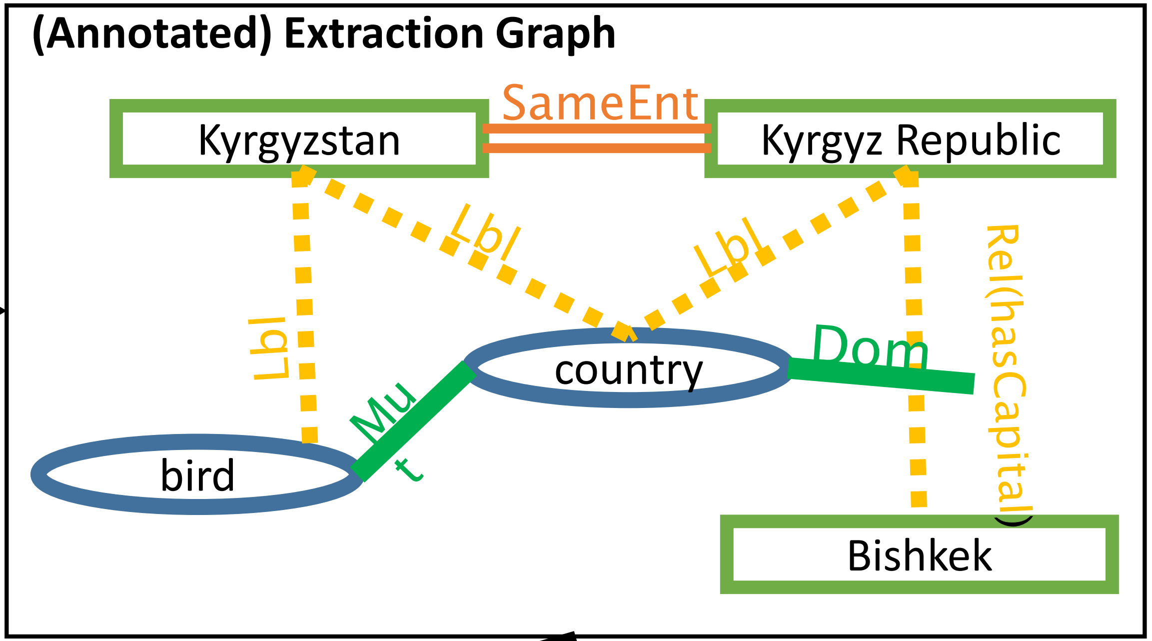
Extraction Graph+Ontology + ER+PSL

Uncertain Extractions:
.5: Lbl(Kyrgyzstan, bird)
.7: Lbl(Kyrgyzstan, country)
.9: Lbl(Kyrgyz Republic, country)

.8: Rel(Kyrgyz Republic, Bishkek, hasCapital)

Ontology:
Dom(hasCapital, country)
Mut(country, bird)

Entity Resolution:
SameEnt(Kyrgyz Republic, Kyrgyzstan)



Probabilistic Soft Logic (PSL)

- Templating language for hinge-loss MRFs, very scalable!
- Model specified as a **collection of logical formulas**

$$\textsc{SameEnt}(E_1, E_2) \;\tilde{\wedge}\; \textsc{Lbl}(E_1, L) \Rightarrow \textsc{Lbl}(E_2, L)$$

- Uses **soft-logic** formulation
  - Truth values of atoms relaxed to [0,1] interval
  - Truth values of formulas derived from Lukasiewicz t-norm

Technical Background: PSL Rules to Distributions

- Rules are *grounded* by substituting literals into formulas

$$\mathbf{w_{EL}} : \textsc{SameEnt}(\text{Kyrgyzstan}, \text{Kyrygyz Republic}) \ \tilde{\wedge}$$
$$\textsc{Lbl}(\text{Kyrgyzstan}, \text{country}) \Rightarrow \textsc{Lbl}(\text{Kyrygyz Republic}, \text{country})$$

- Each ground rule has a weighted *distance to satisfaction* derived from the formula's truth value

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

- The PSL program can be interpreted as a joint probability distribution over all variables in knowledge graph, conditioned on the extractions

Finding the best knowledge graph

- **Most probable explanation** (MPE) inference solves $\max_G P(G)$ to find the best KG
- In PSL, inference solved by **convex** optimization
- **Efficient:** running time scales with $O(|R|)$

PSL Rules: Uncertain Extractions

Weight for source T (relations)

Predicate representing uncertain relation extraction from extractor T

Relation in Knowledge Graph

$$\mathbf{w_{CR-T}} : \textsc{CandRel}_T(E_1, E_2, R) \Rightarrow \textsc{Rel}(E_1, E_2, R)$$

$$\mathbf{w_{CL-T}} : \textsc{CandLbl}_T(E, L) \Rightarrow \textsc{Lbl}(E, L)$$

Weight for source T (labels)

Predicate representing uncertain label extraction from extractor T

Label in Knowledge Graph

PSL Rules: Entity Resolution

$$\mathbf{w_{EL}} : \textsc{SameEnt}(E_1, E_2) \tilde{\wedge} \textsc{Lbl}(E_1, L) \Rightarrow \textsc{Lbl}(E_2, L)$$
$$\mathbf{w_{ER}} : \textsc{SameEnt}(E_1, E_2) \tilde{\wedge} \textsc{Rel}(E_1, E, R) \Rightarrow \textsc{Rel}(E_2, E, R)$$
$$\mathbf{w_{ER}} : \textsc{SameEnt}(E_1, E_2) \tilde{\wedge} \textsc{Rel}(E, E_1, R) \Rightarrow \textsc{Rel}(E, E_2, R)$$

ER predicate captures confidence that entities are co-referent

- Rules require co-referent entities to have the same labels and relations
- Creates an *equivalence class* of co-referent entities

PSL Rules: Ontology

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

Evaluated extensively: case study on NELL

Task: Compute a full knowledge graph from uncertain extractions

Comparisons:

NELL NELL's strategy: ensure ontological consistency with existing KB

PSL-KGI Apply full Knowledge Graph Identification model

Running Time: Inference completes in 130 minutes, producing 4.3M facts

	AUC	Precision	Recall	F1
NELL	0.765	0.801	0.477	0.634
PSL-KGI	0.892	0.826	0.871	0.848

Summary

- Probabilistic Soft Logic (PSL) is a powerful framework for producing knowledge graphs from noisy IE and ER outputs
- PSL can be used to enforce global ontological constraints and capture uncertainty in the model
- The model is scalable i.e. it infers complete knowledge graphs for datasets with millions of extractions

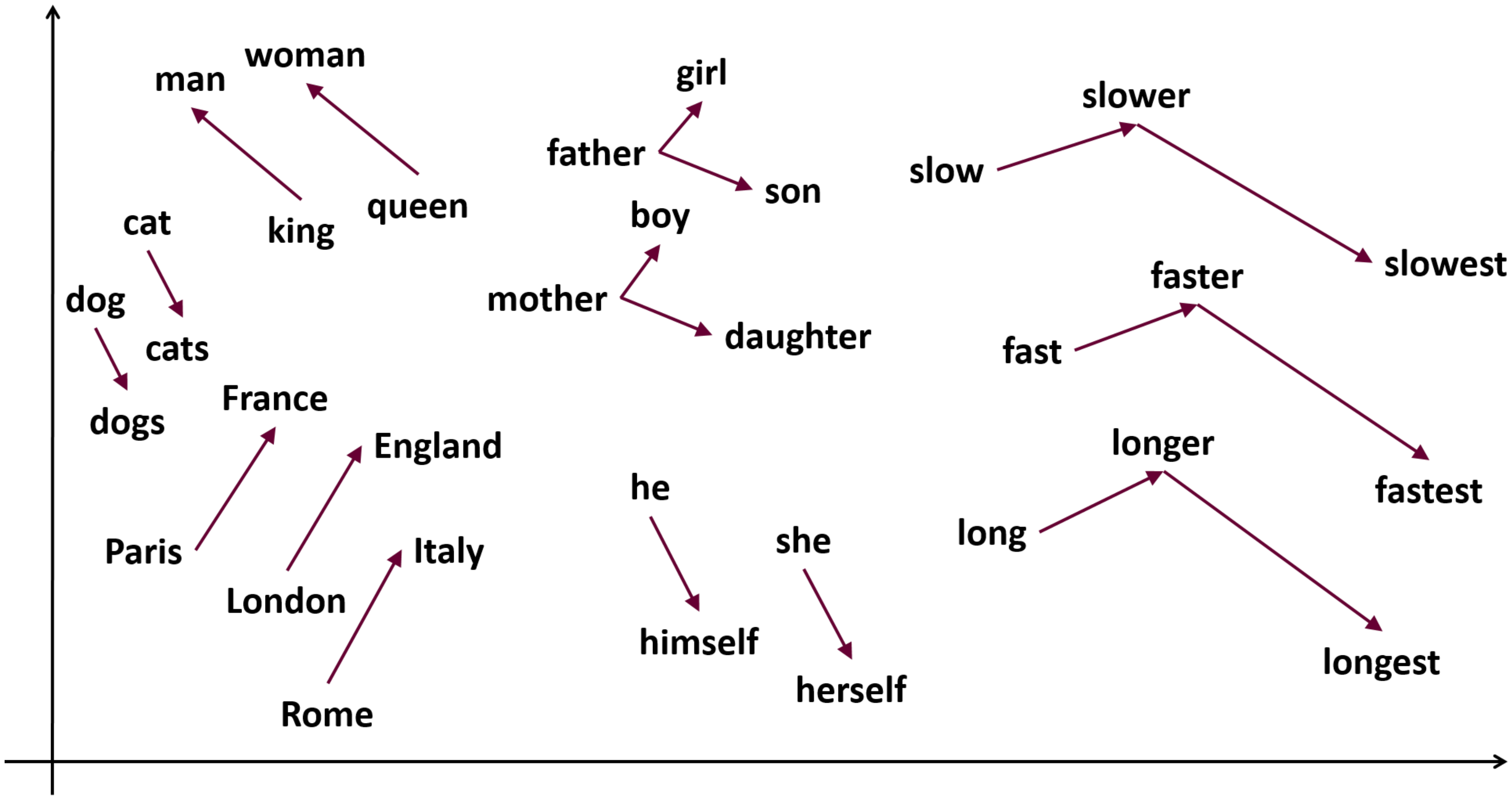
Very well-documented and maintained: code, tutorials and publications openly available:

https://github.com/linqs/psl

Knowledge Graph Embeddings (KGEs)

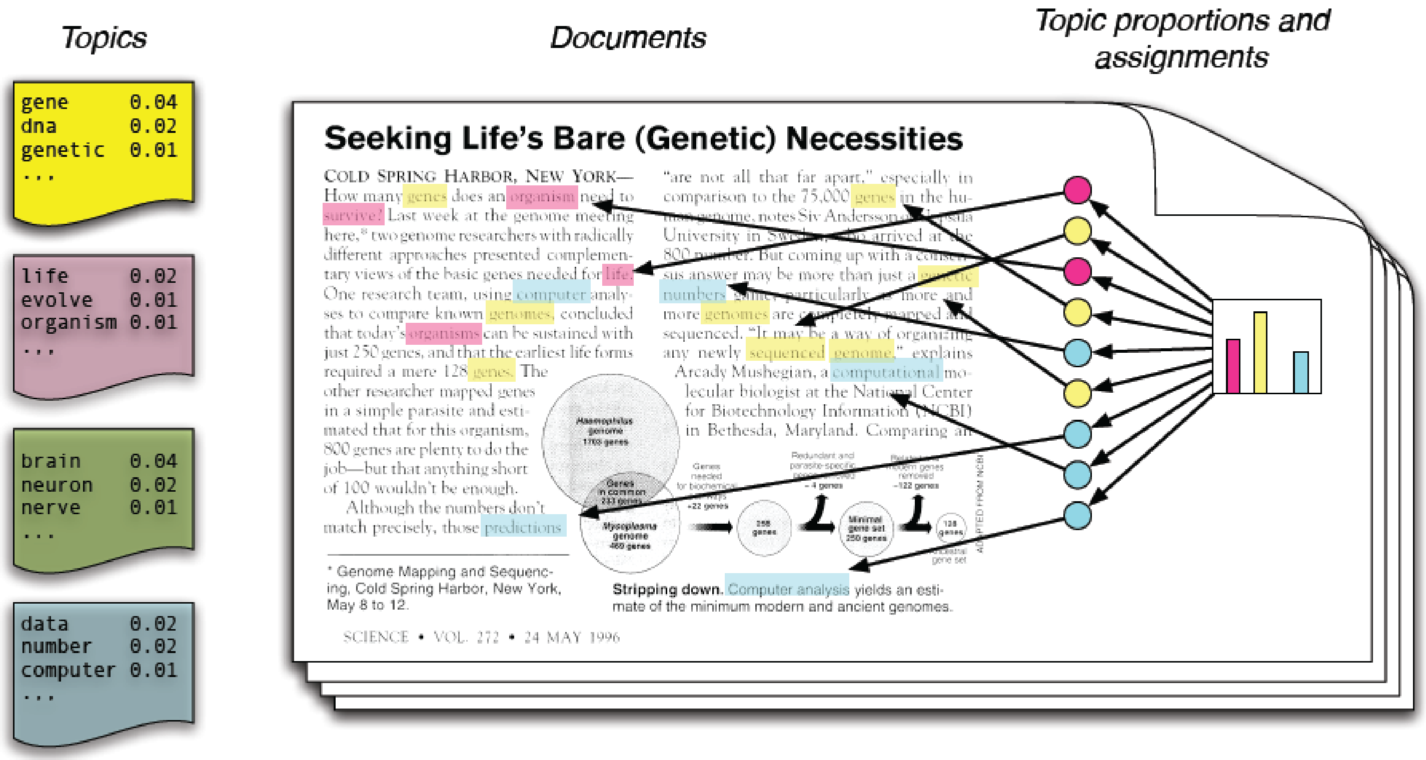
Low-dimensional vector spaces

- Very popular for documents, graphs, words...



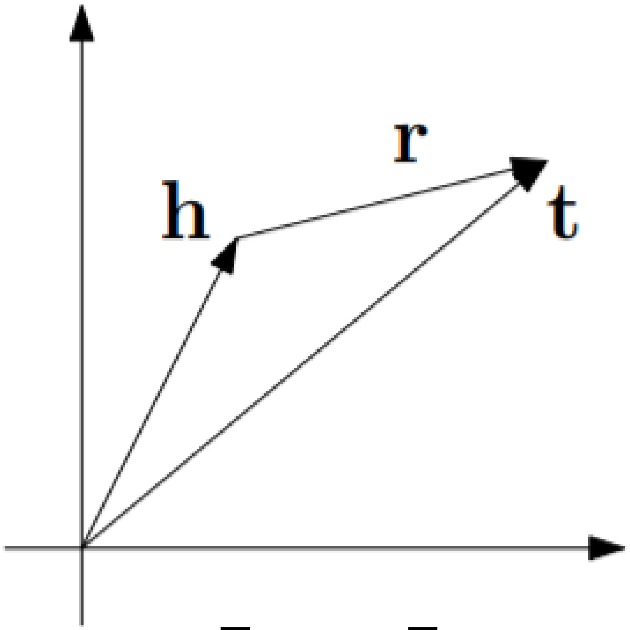
Some more intuition

- Embeddings are not a 'new' invention…topic models are an early example still widely used!

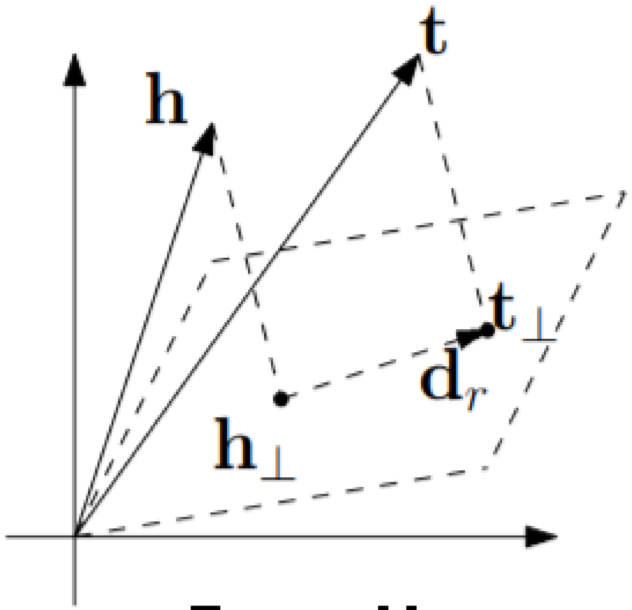


Knowledge graph embeddings

- Many ways to model the problem: entities are usually vectors, relations could be vectors or matrices



TransE



TransH

Objective/loss/energy functions

- What is an 'optimal' vector/matrix for an entity or relation?

Model	Score function $f_r(\mathbf{h}, \mathbf{t})$	# Parameters
TransE (Bordes et al. 2013b)	$\lVert\mathbf{h} + \mathbf{r} - \mathbf{t}\rVert_{\ell_{1/2}}$, $\mathbf{r} \in \mathbb{R}^k$	$O(n_e k + n_r k)$
Unstructured (Bordes et al. 2012)	$\lVert\mathbf{h} - \mathbf{t}\rVert_2^2$	$O(n_e k)$
Distant (Bordes et al. 2011)	$\lVert W_{rh}\mathbf{h} - W_{rt}\mathbf{t}\rVert_1$, $W_{rh}, W_{rt} \in \mathbb{R}^{k\times k}$	$O(n_e k + 2n_r k^2)$
Bilinear (Jenatton et al. 2012)	$\mathbf{h}^\top W_r \mathbf{t}$, $W_r \in \mathbb{R}^{k\times k}$	$O(n_e k + n_r k^2)$
Single Layer	$\mathbf{u}_r^\top f(W_{rh}\mathbf{h} + W_{rt}\mathbf{t} + \mathbf{b}_r)$ $\mathbf{u}_r, \mathbf{b}_r \in \mathbb{R}^s$, $W_{rh}, W_{rt} \in \mathbb{R}^{s\times k}$	$O(n_e k + n_r(sk + s))$
NTN (Socher et al. 2013)	$\mathbf{u}_r^\top f(\mathbf{h}^\top \mathbf{W}_r \mathbf{t} + W_{rh}\mathbf{h} + W_{rt}\mathbf{t} + \mathbf{b}_r)$ $\mathbf{u}_r, \mathbf{b}_r \in \mathbb{R}^s$, $\mathbf{W}_r \in \mathbb{R}^{k\times k\times s}$, $W_{rh}, W_{rt} \in \mathbb{R}^{s\times k}$	$O(n_e k + n_r(sk^2 + 2sk + 2s))$
TransH (	$\lVert(\mathbf{h} - \mathbf{w}_r^\top \mathbf{h}\mathbf{w}_r) + \mathbf{d}_r - (\mathbf{t} - \mathbf{w}_r^\top \mathbf{t}\mathbf{w}_r)\rVert_2^2$ $\mathbf{w}_r, \mathbf{d}_r \in \mathbb{R}^k$	$O(n_e k + 2n_r k)$

Existing work

- Typically evaluate on Freebase and WordNet

Data	WN18	FB15K	WN11	FB13
#Rel	18	1,345	11	13
#Ent	40,943	14,951	38,696	75,043
#Train	141,442	483,142	112,581	316,232
#Valid	5,000	50,000	2,609	5,908
#Test	5,000	59,071	10,544	23,733

Wang et al. (2008)

Application 1: Triples completion

Dataset	WN18				FB15k			
Metric	MEAN		HITS@10		MEAN		HITS@10	
	Raw	Filt.	Raw	Filt.	Raw	Filt.	Raw	Filt.
Unstructured (Bordes et al. 2012)	315	304	35.3	38.2	1,074	979	4.5	6.3
RESCAL (Nickel, Tresp, and Kriegel 2011)	1,180	1,163	37.2	52.8	828	683	28.4	44.1
SE (Bordes et al. 2011)	1,011	985	68.5	80.5	273	162	28.8	39.8
SME (Linear) (Bordes et al. 2012)	545	533	65.1	74.1	274	154	30.7	40.8
SME (Bilinear) (Bordes et al. 2012)	526	509	54.7	61.3	284	158	31.3	41.3
LFM (Jenatton et al. 2012)	469	456	71.4	81.6	283	164	26.0	33.1
TransE (Bordes et al. 2013b)	263	**251**	75.4	**89.2**	243	125	34.9	47.1
TransH (unif.)	318	**303**	75.4	**86.7**	211	**84**	42.5	**58.5**
TransH (bern.)	400.8	388	73.0	82.3	212	**87**	45.7	**64.4**

Wang et al. (2008)

Application 2: Triples classification

Dataset	WN11	FB13	FB15k
Distant Model	53.0	75.2	-
Hadamard Model	70.0	63.7	-
Single Layer Model	69.9	**85.3**	-
Bilinear Model	73.8	84.3	-
NTN	70.4	**87.1**	66.5 ($\approx 40h$)
TransE (unif.)	75.85	70.9	79.7 ($\approx 5m$)
TransE (bern.)	75.87	81.5	**87.3** ($\approx 5m$)
TransH (unif.)	**77.68**	76.5	80.2 ($\approx 30m$)
TransH (bern.)	**78.80**	83.3	**87.7** ($\approx 30m$)

Wang et al. (2008)

Code availability

- Code for replicating experiments can be found at https://github.com/glorotxa/SME ; implemented using both theano/tensorflow backend
- Unclear how to extend to new, sparse data, how to scale to much bigger KGs

Application 3: 'Featurizing' locations

- E.g. Convering 'locations' into feature vectors
- Relevant for toponym resolution, building rich graphs...

Kejriwal, Mayank; Szekely, Pedro (2017): Neural Embeddings for Populated GeoNames Locations. figshare. https://doi.org/10.6084/m9.figshare.5248120

https://github.com/mayankkejriwal/Geonames-embeddings

Features encode spatial proximity

- But could encode much else, lots of room for new research!

Embeddings and **extracted** knowledge graphs

- **Do embeddings work for extracted KGs?**
- Approach by Pujara et al. (2017): Evaluate on the NELL knowledge graph, containing millions of candidates extracted from WWW text
- **Observations:**
- Baseline (threshold input) wins against embeddings
- Best results from graphical model (PSL-KGI[8]) using rules & uncertainty
- More complex embedding methods have the worst performance
- **Conclusion:** Embeddings have poor performance on sparse & noisy KGs extracted from text
- **Key question for future research: How do we make embeddings work for extracted KGs?**

Method	AUC	F1
NELL	0.765	0.673
TransH	0.701	0.783
HolE	0.710	0.783
TransE	0.726	0.783
STransE	0.784	0.783
Baseline	0.873	0.828
PSL-KGI	**0.891**	**0.848**

Summary

- Knowledge graph embedding (KGE) is an active research area
- Uses machine learning and neural networks to ‘vectorize’ entities and relationships
- Implementations can be slow, recently this has started to change
- Unlike PSL, ecosystem not yet matured