Semi-supervised Instance Matching Using Boosted Classifiers

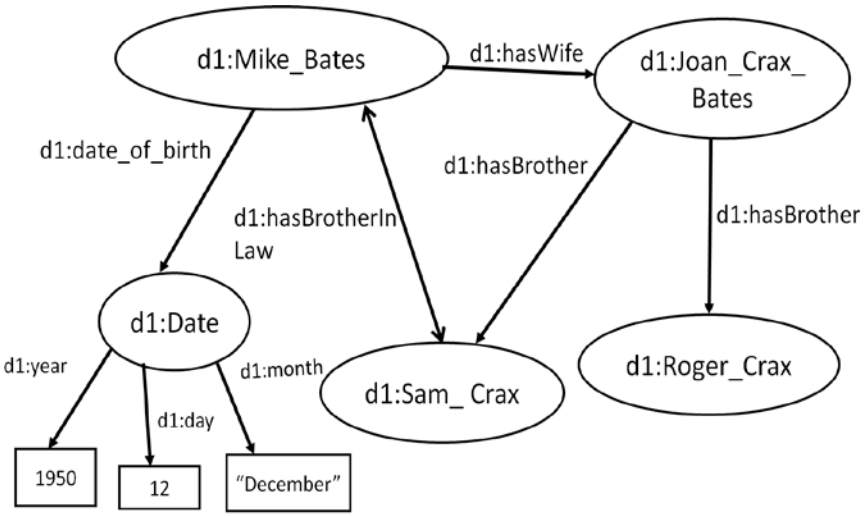
Mayank Kejriwal, Daniel P. Miranker

University of Texas at Austin

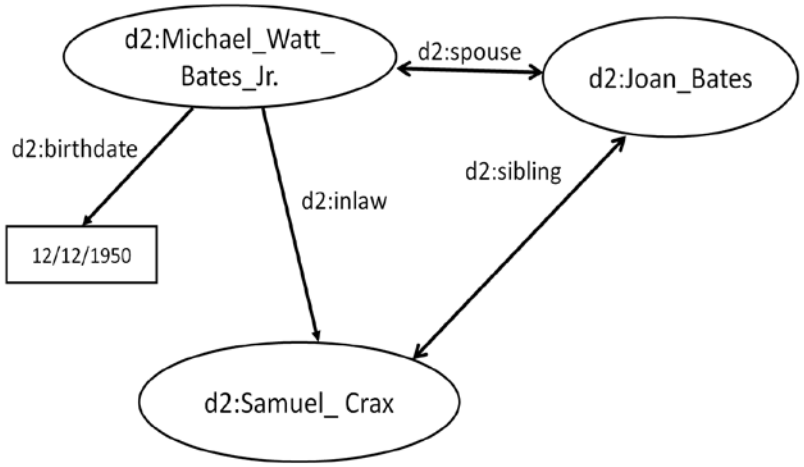
Acknowledgement: National Science Foundation

Instance Matching

- **High-level question**: Which pieces of data ('instances') refer to the same thing?



RDF Graph 1

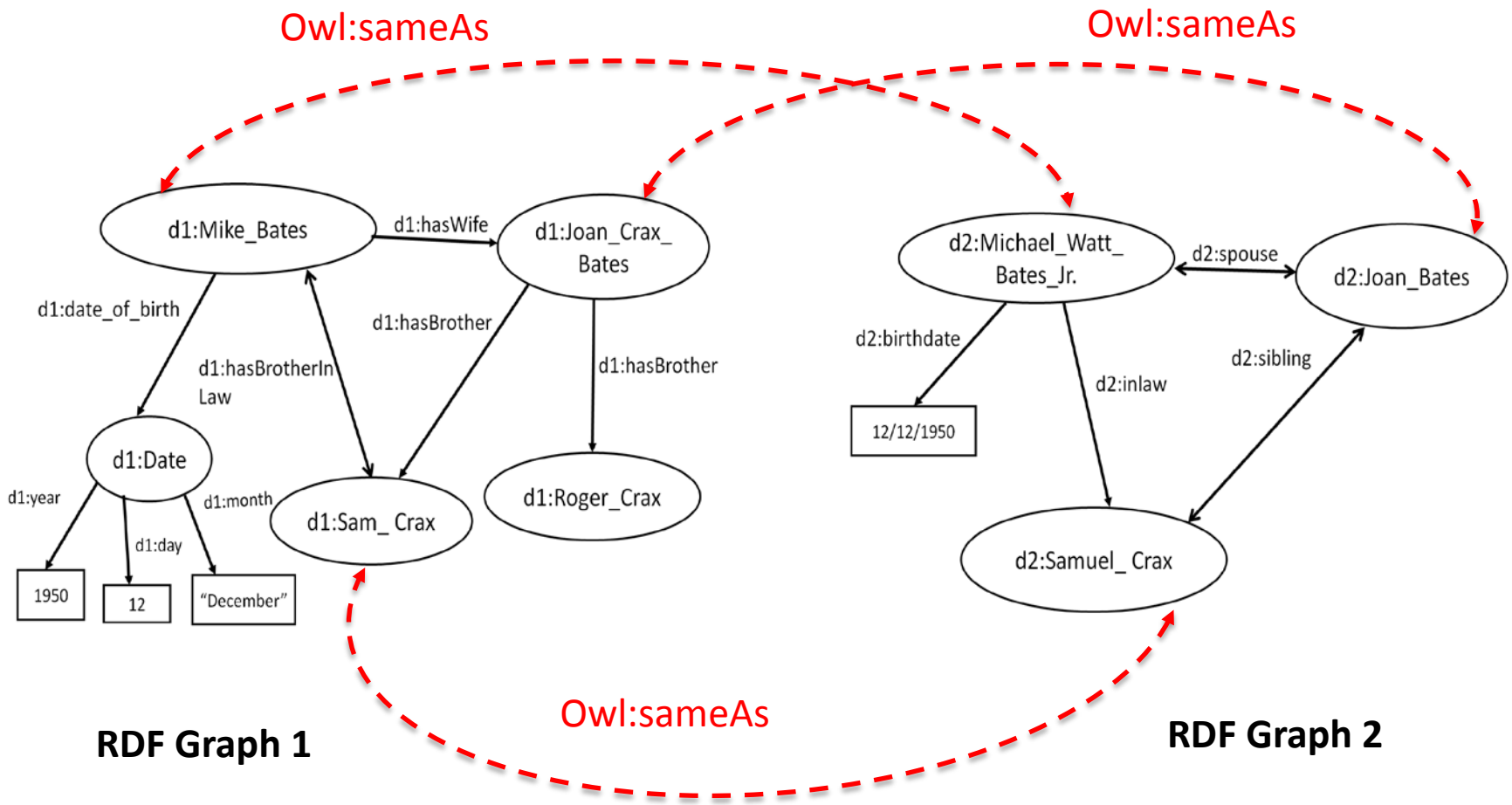


RDF Graph 2

- Usually easy for a **human being**...

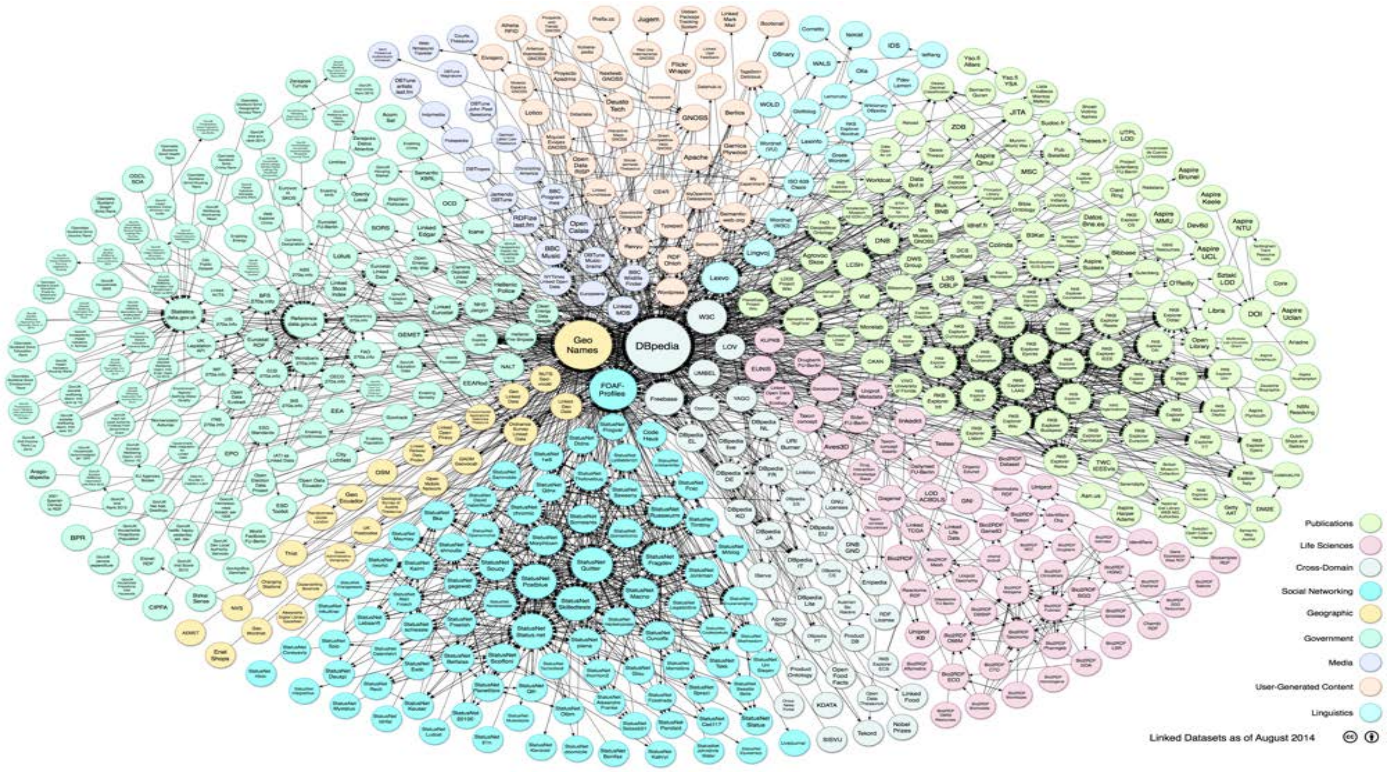
Jaffri et al. (2008)
Nikolov et al. (2011)

- Instance Matching: a 50 year old **Artificial Intelligence** problem!



Newcombe (1959)

Primary Application: Linked Open Data

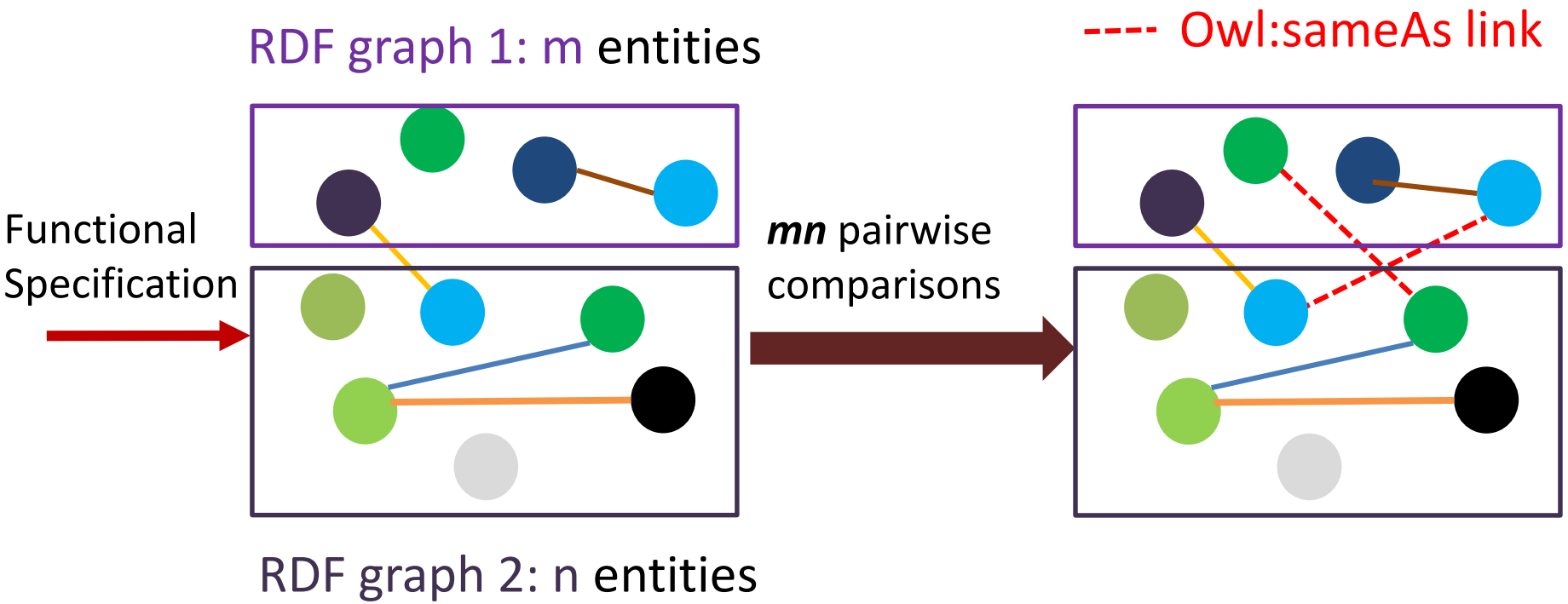


- Also, semantic search, knowledge graph identification...
- **Emphasized** in the keynote today

Linkeddata.org
Bizer et al.
Pujara et al.

Two key aspects

1. Efficiency
   - How do we locate matching instance pairs without comparing **all** pairs?



Instead, choose a small subset of the *nm* pairs that are good link candidates through a blocking phase

Hernandez and Stolfo (1995)

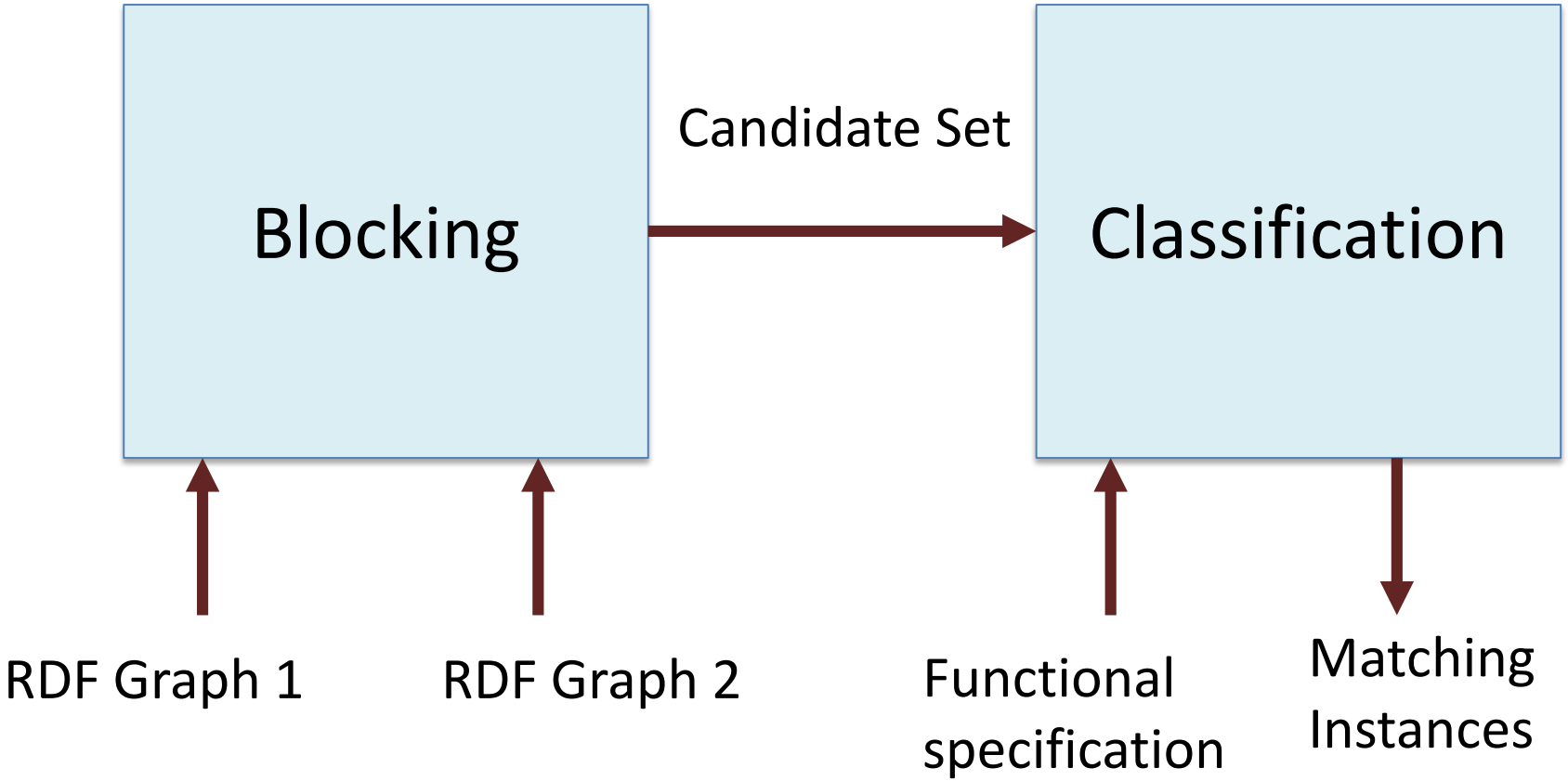
Two key aspects

2. Effectiveness:
   – How do we devise an **adaptive** functional specification to locate matching pairs with **high-quality**?

Answer: *Machine Learning*

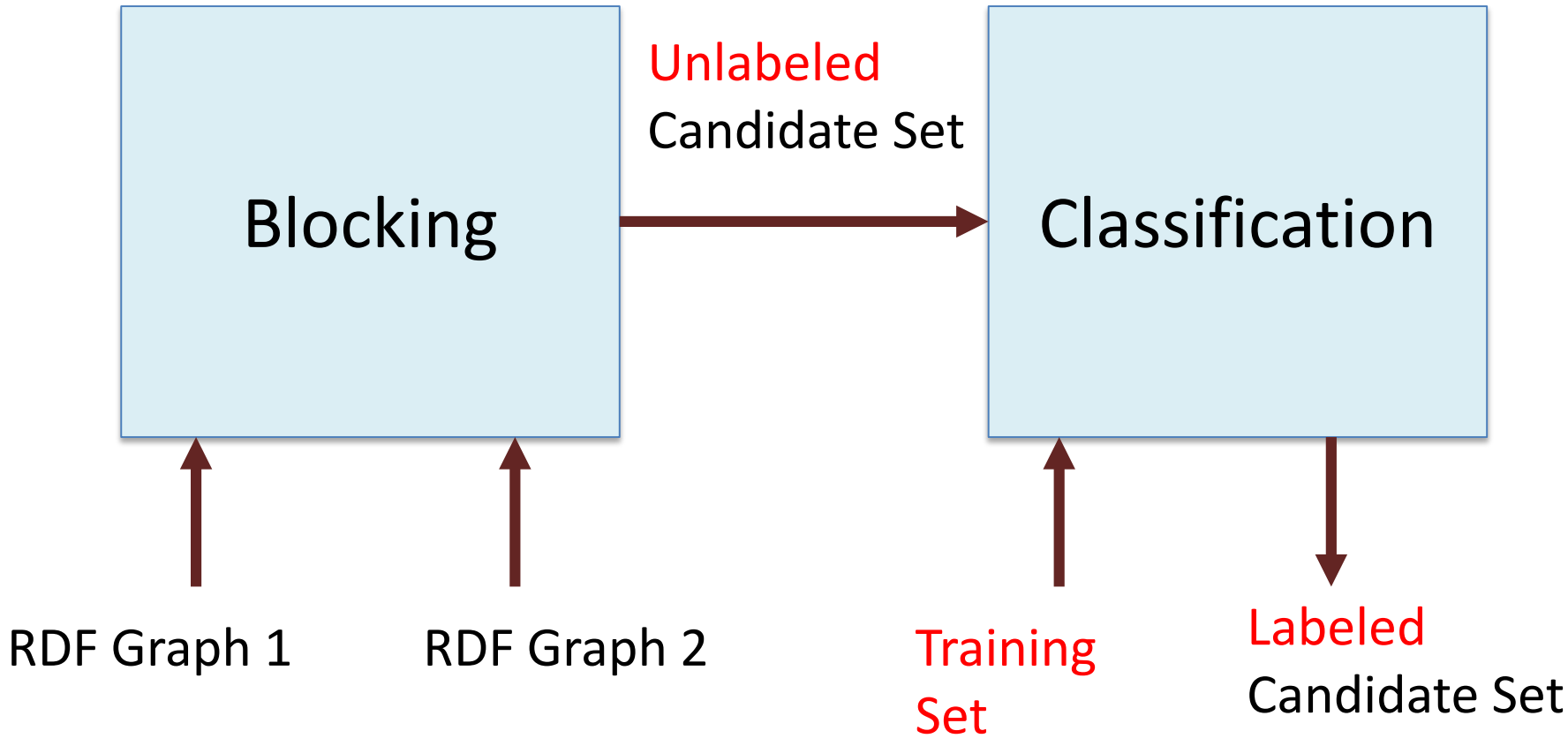
Bilenko et al. (2003)
Elmagarmid et al. (2007)

An Instance Matching pipeline



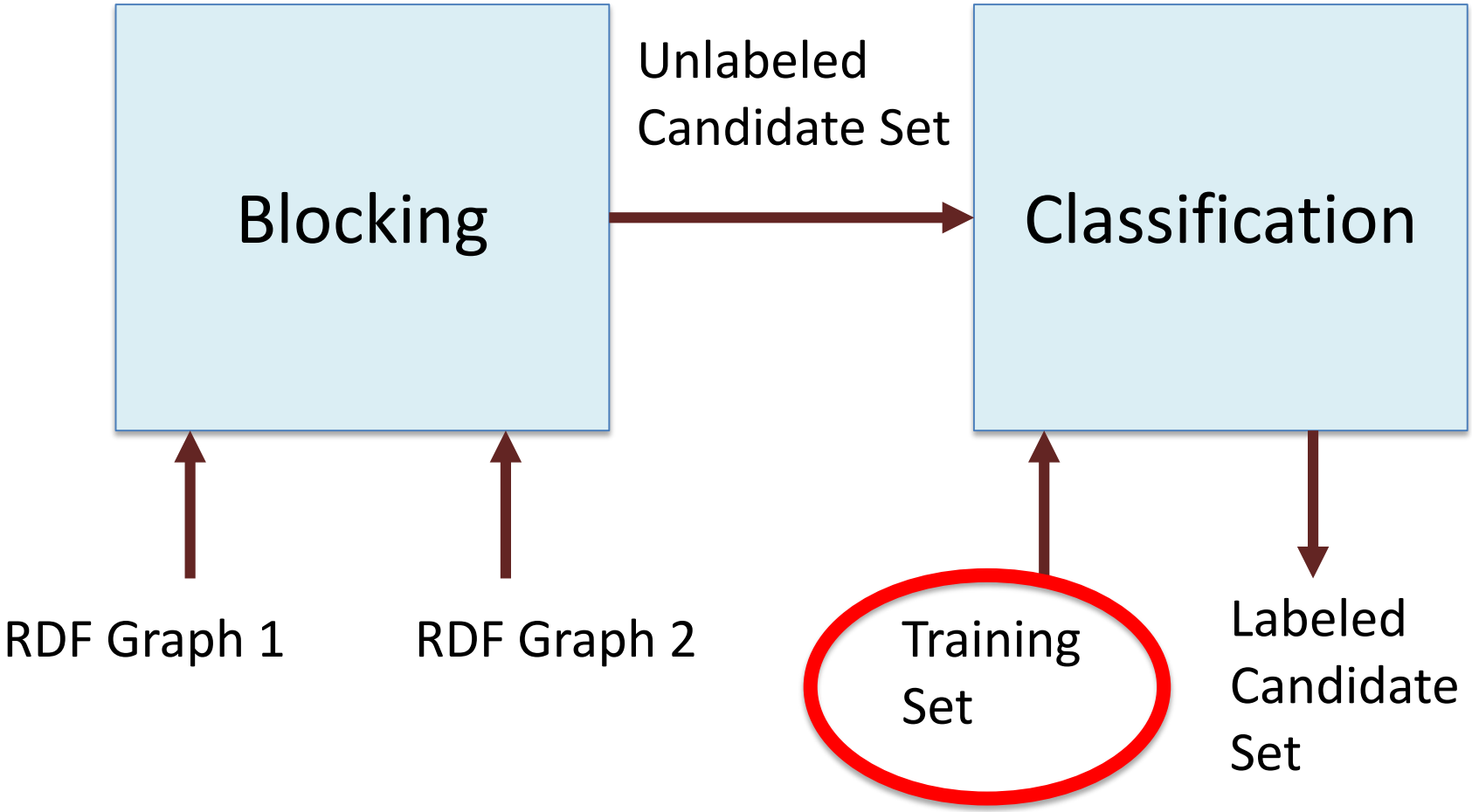
Christen (2012)

Machine learning perspective



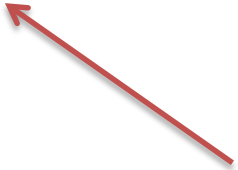
Also some preprocessing (generating restriction sets, extracting feature engineering…)

But…manual labeling is expensive



How can we minimize labeling effort without degrading quality (by much)?

Semi-supervised Learning



How can we minimize labeling effort without degrading quality (by much)?

Zhu (2005)

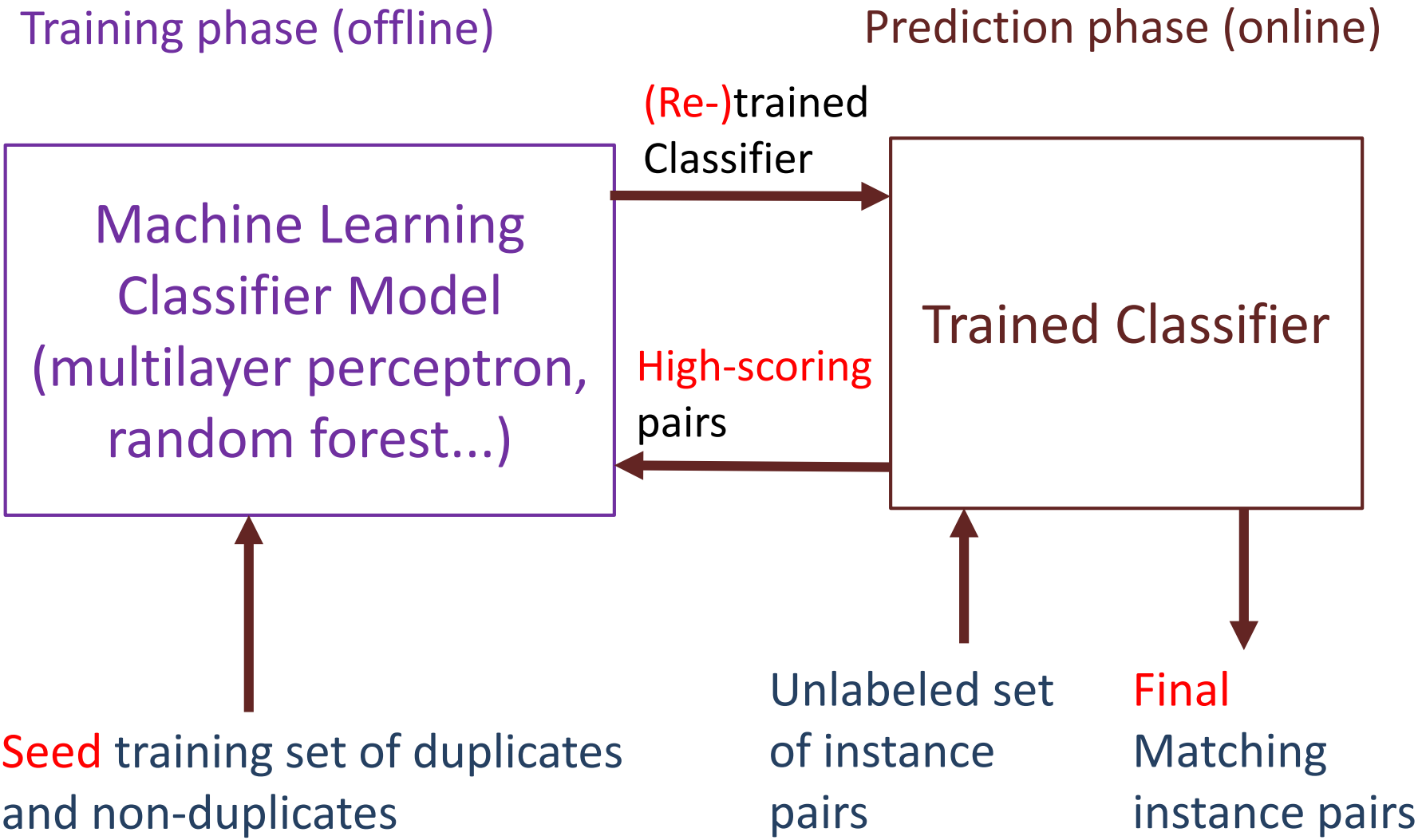
Semi-supervised Learning

How can we minimize labeling effort without degrading quality (by much)?

Boosting

Zhu (2005)
Freund and Schapire (1997)

Semi-supervised learning



Boosting (specifically *Adaboost*)

- Key idea is to train a committee of classifiers (called an ensemble classifier) in an iterative manner
- Training samples are re-weighted in each iteration, based on what previous classifiers got wrong
- The weight of each classifier is the normalized accuracy on the full training set

Adaboost algorithm

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X, y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : X \to \{-1, +1\}$ with error

$$\epsilon_t = \Pr_{i \sim D_t} [h_t(x_i) \neq y_i].$$

- Choose $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$.
- Update:

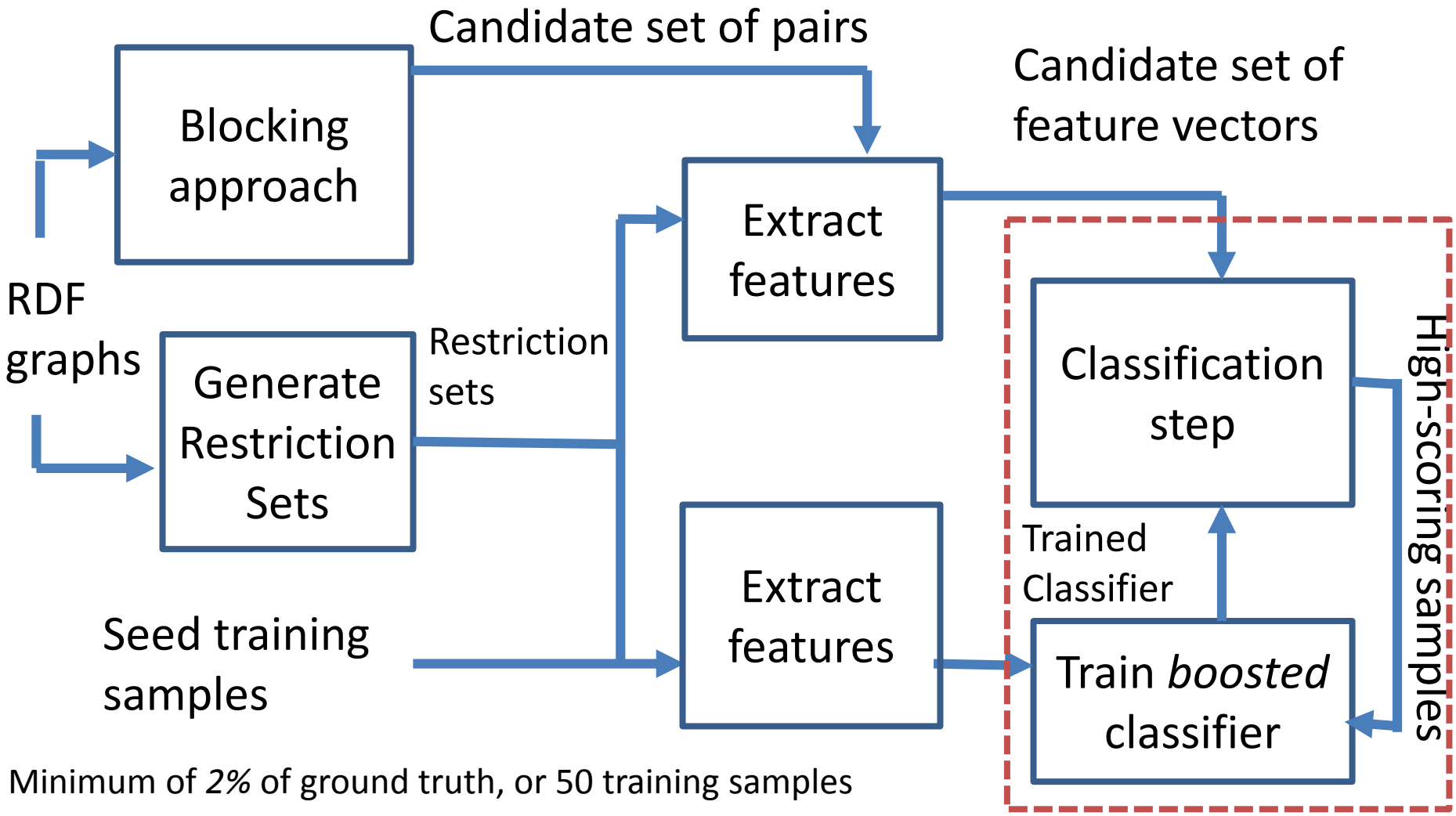
$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \end{aligned}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \text{sign} \left(\sum_{t=1}^{T} \alpha_t h_t(x) \right).$$

Proposed System



Proposed Algorithm

Algorithm 1 Classification step

Input: Seed training sets (comprising *feature vectors*) of positive and negative samples D and N resp., Candidate Set Γ, Base Classifier M, Iteration rounds num, Positive factors for positive and negative samples $factor_D$ and $factor_N$ resp.

Output: Ranked list L of pairs in Γ

1. Initialize list L of size $|\Gamma|$
2. Initialize $num_D := |D|$
3. Initialize $num_N := |N|$
4. Train classifier M with the *AdaBoost* ensemble method using D and N as training sets; let the trained classifier model be denoted as M'
5. Initialize $count := 0$
6. **while** $count < num$ **do**
   - Score each pair in Γ using M' and place pair in L
   - Sort L in ascending order using the scores as sorting keys
   - $num_D := num_D \times factor_D$
   - $num_N := num_N \times factor_N$
   - Repeat step 4 by using the first num_D elements in L as positive training examples, and last num_N elements in L as negative training examples
   - $count := count + 1$
7. **end while**
8. **return** L

Boosting

Semi-supervised learning

(Very high-level) Intuition

- Boosting and semi-supervised learning deal with separate but not mutually exclusive machine learning issues:
  - Semi-supervised learning: Small training set size
  - Boosting: Weak classifier made strong, more training set representation
- A different risk-return tradeoff: initial classification error rather than initial training set size determines overall system performance

Evaluation: Test Cases

- **Rationale:** Evaluate on public datasets that reflect real-world noise and have been extensively evaluated by other systems
- We picked six real-world test cases
  - Three cases (Persons 1, Persons 2 and Restaurants) from OAEI 2010 initiative
  - Two from real-world e-commerce domain (Amazon-GoogleProducts and AbtBuy)
  - One from bibliographic domain (ACM-DBLP)
- All datasets, experimental data and related materials released on project website

https://sites.google.com/a/utexas.edu/mayank-kejriwal/projects/semi-supervised-im-using-boosting

Evaluation: Metrics

- Let *R* be the set of true positives returned by the system and *T* be the ground-truth set of true positives

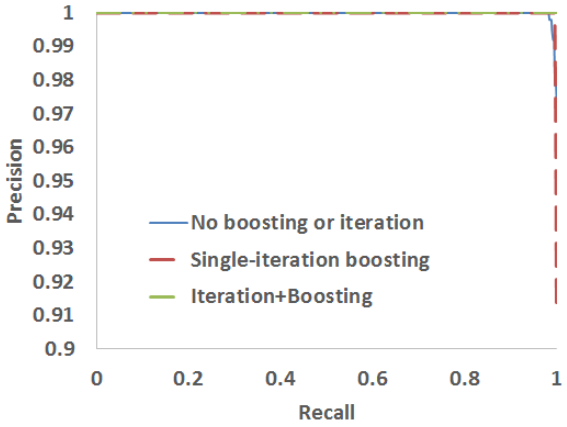
$$Precision = \frac{R \cap T}{R} \qquad Recall = \frac{R \cap T}{T}$$

$$F - Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

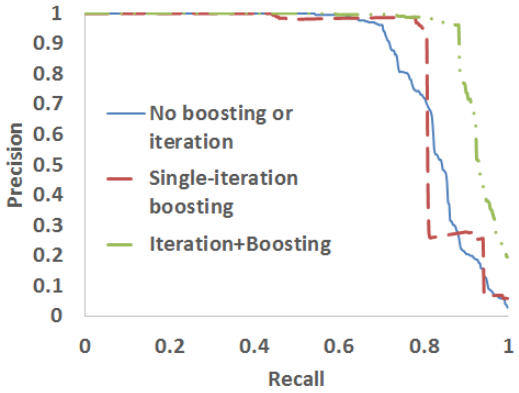
Evaluation: Baselines

- Contrast effects of boosting vs. boosting+semi-supervision vs. neither
- Contrast effects of using base classifiers with different characteristics (specifically Random Forest vs. Multilayer Perceptron)
- Compare against best f-scores reported in the literature from competitive supervised and unsupervised systems

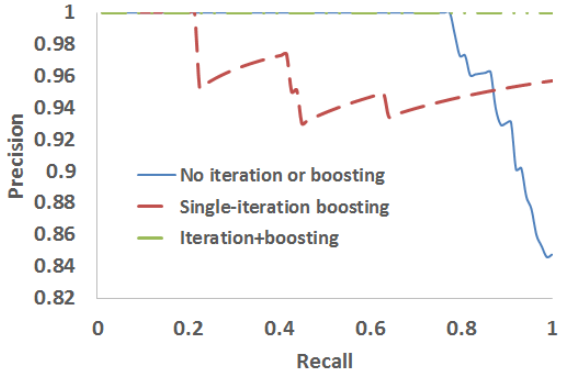
Results: Random Forest



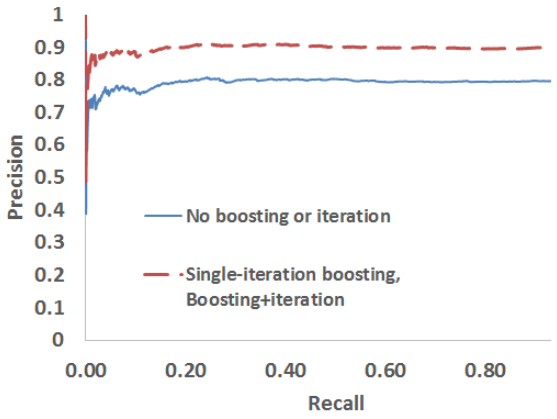
Persons1



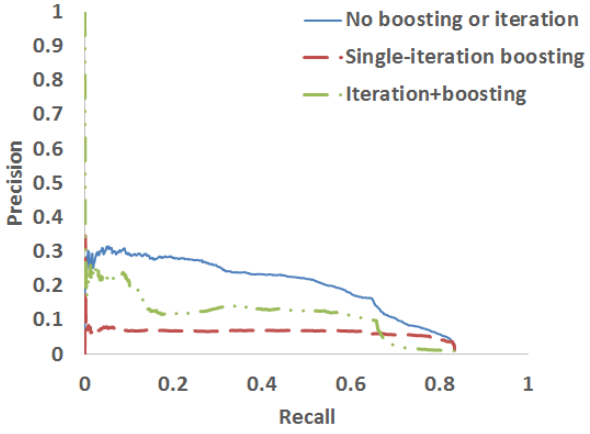
Persons2



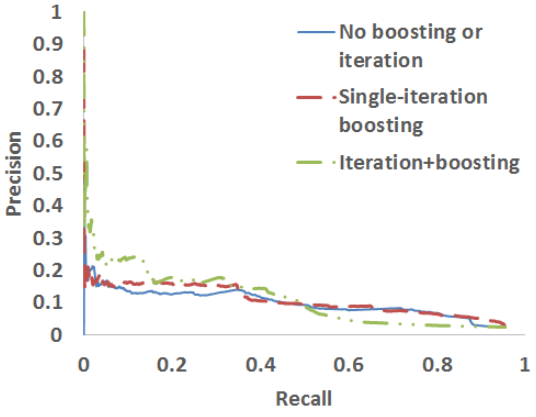
Restaurants



ACM-DBLP

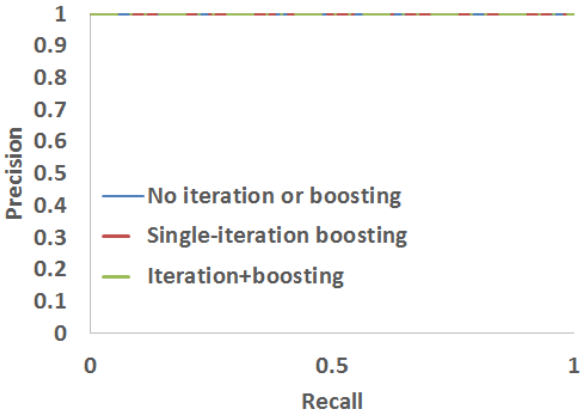


Amazon-GoogleProducts

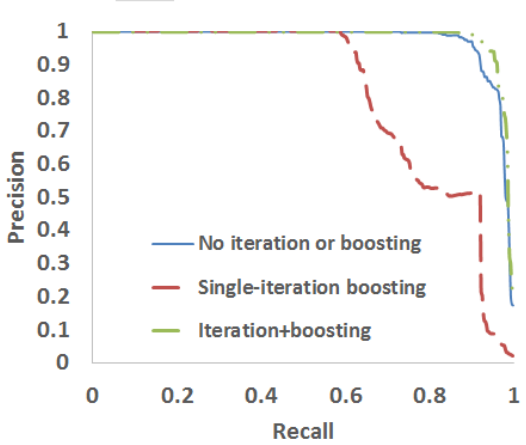


Abt-Buy

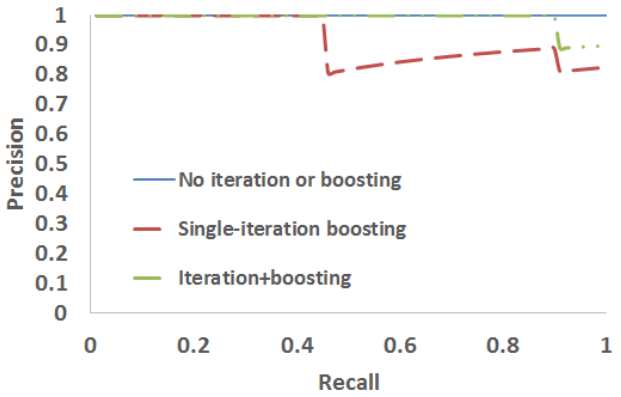
Results: Multilayer Perceptron



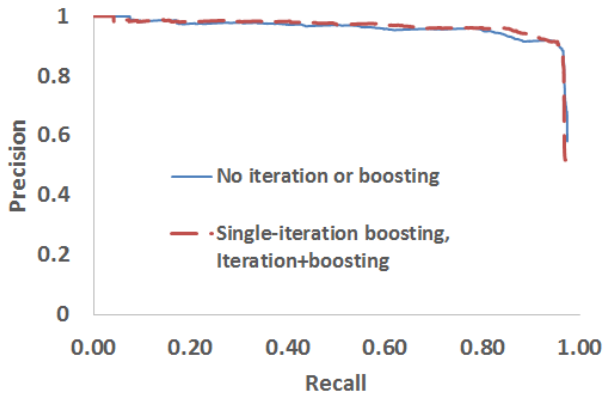
Persons1



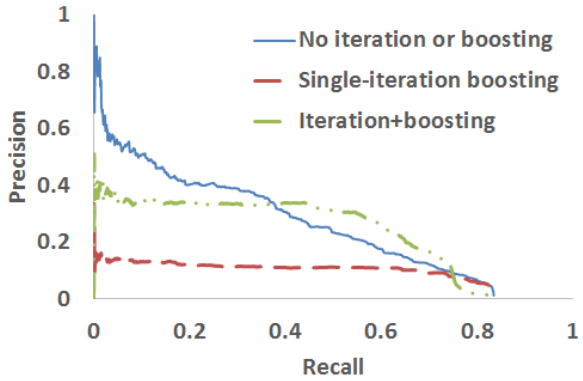
Persons2



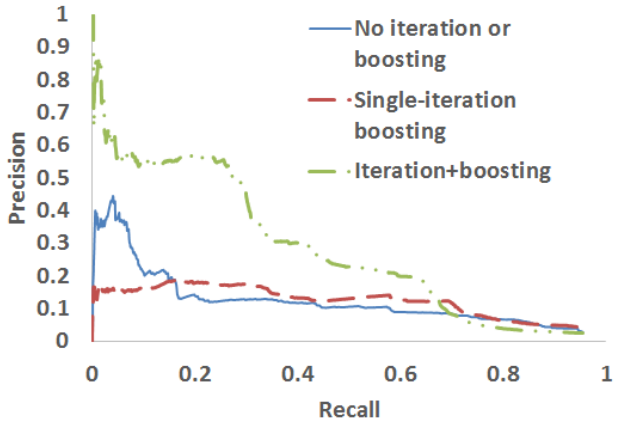
Restaurants



ACM-DBLP



Amazon-GoogleProducts



Abt-Buy

Results: External Baselines

Test Case	Linear	Boolean	Genetic	Proposed
Persons1	**100.00%**	99.50%	**100.00%**	**100.00%**
Persons2	41.45%	59.12%	37.04%	**97.19%**
Restaurants	88.56%	88.56%	88.56%	**94.68%**
ACM-DBLP	**97.96%**	97.46%	97.71%	93.42%
Amazon-GP	**49.08%**	39.97%	43.11%	39.13%
Abt-Buy	**48.60%**	37.66%	45.08%	36.27%
Average	70.94%	72.18%	68.58%	**76.78%**

- Average best F-score is within 2.5% of supervised variants of machine learning instance matchers evaluated in the literature
- Performance is also comparable to that of state-of-the-art supervised instance matchers (that also employ preprocessing, feature engineering etc.) such as FEBRL and Marlin

Key Takeaways

- Semi-supervised learning+boosting is an effective technique for minimally supervised instance matching
- Good performance is achieved with a non-linear, expressive classifier (e.g. multilayer perceptron)
- Offers a non-traditional risk-return tradeoff (early errors quickly propagate)

Conclusion

- Minimally supervised instance matching is a hard problem
  - Has many applications, but especially useful for discovering links in Linked Open Data
- We combine boosting with semi-supervised learning to address the problem
- We implement a full instance matching system with state-of-the-art components to evaluate the methods