Relational Clustering for Multi-type Entity Resolution

Indrajit Bhattacharya and Lise Getoor
Department of Computer Science, University of Maryland

Presented by Martin Leginus

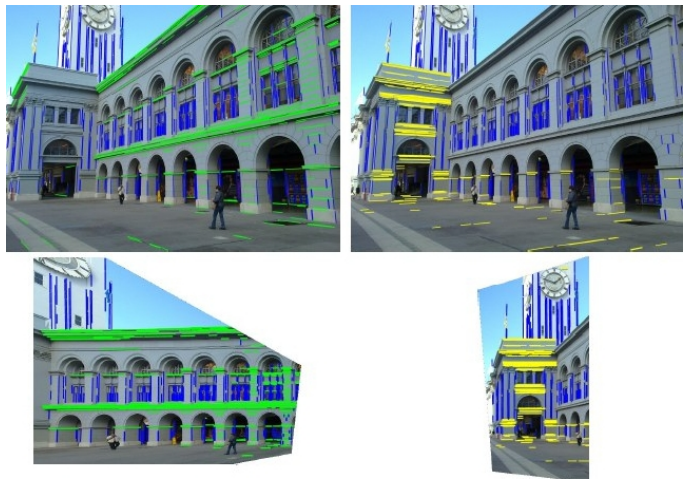
13th of March, 2013

Agenda

- Motivation
- Related work
- Use case scenarios
- Problem formulation
- Relational clustering
- Similarity measures
- Results
- Discussion

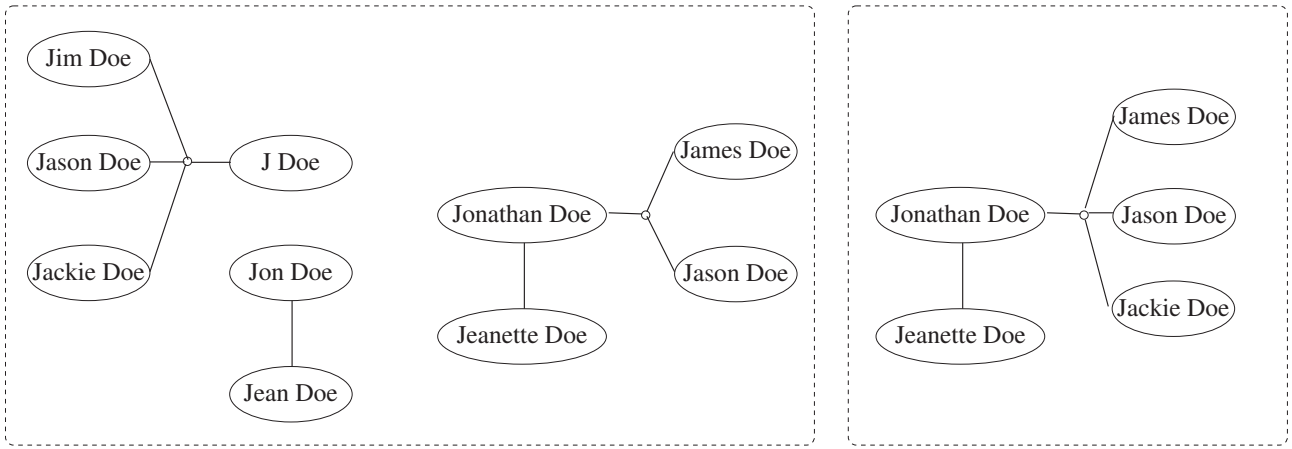
Why there is a need for entity resolution?

- The correspondence problem - 2 pictures refer to the same entity.
- Natural language processing - recognizing which noun phrases refer to the same entity.
- Data preprocessing - detection of duplicates.



Why there is a need for relational entity resolution?

Traditional approaches utilize textual similarity measures.



Relational evidences might improve the accuracy of the resolution.

Related work

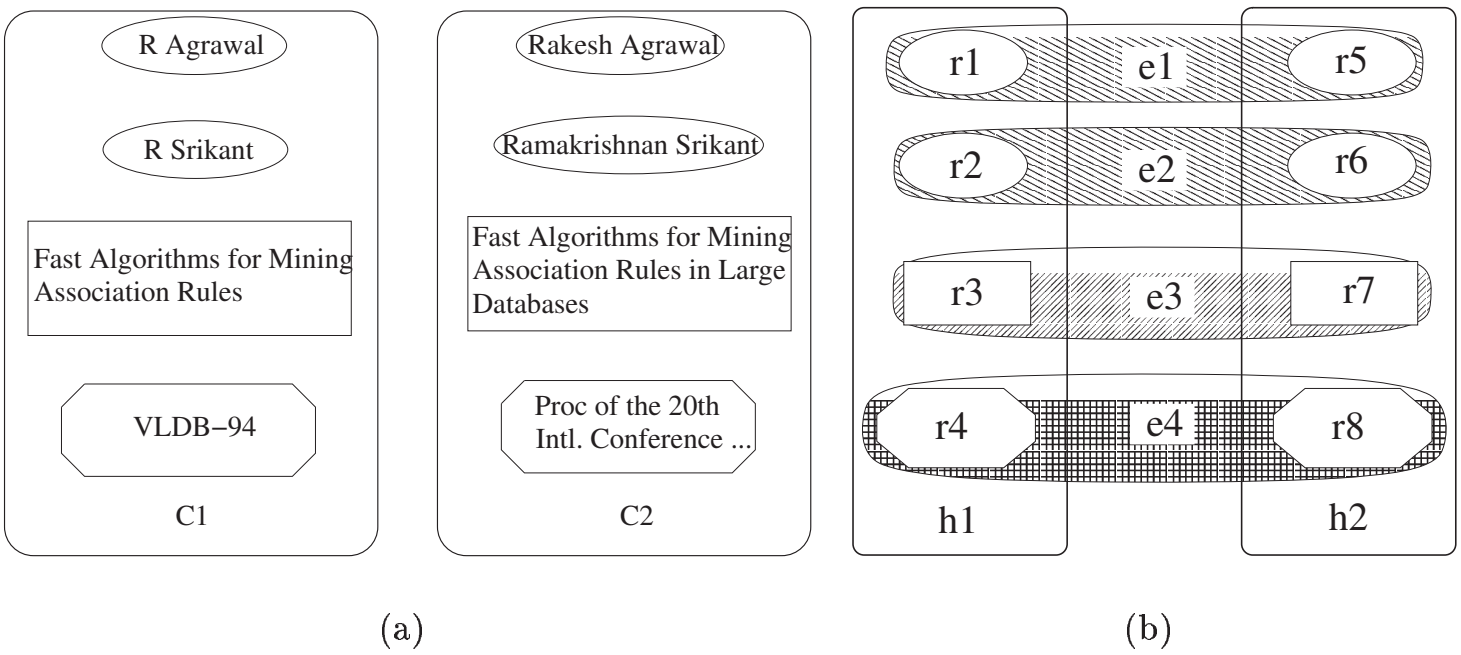
- Textual similarity calculated for the descriptions of two entities.
- Supervised alg. that learn string similarity measures from labelled data.
- Performance is improved with blocking approach.
- Relational features considered for data integration problems.

Use case example

Two citation examples of the same paper:
Fast algorithms for mining association rules in large databases. Agrawal, Rakesh and Srikant, Ramakrishnan. In Proc. 20th Int. Conf. Very Large Data Bases, VLDB, 1994
Fast algorithms for mining association rules. Agrawal, R., Srikant, R. in VLDB-94,1994

- String edit distance does not work.
- Multiple entity resolution problem i.e., *author, paper and venue entities*.

Joint resolution using entity relations

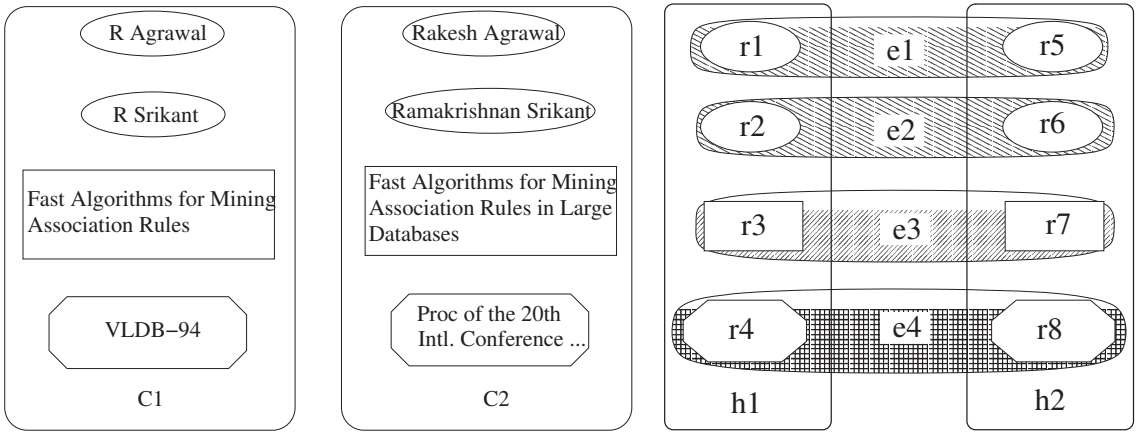


- Local and global resolution.
- Positive and negative relational evidence.

Problem formulation

Entities and references are denoted by e and r.
Assigned variables of e and r are denoted by $e.A$ and $r.A$.
References are typed and $r.T$ is observed. Each reference r corresponds to a hidden entity so that each r has assigned entity label $r.E$.
The problem is to discover the hidden set of entities $E = \{e_i\}$ and entity labels $r.E$ for each reference.
References are observed as members of hyper-edges. The membership of a reference is stored in hyper-edge label $r.H = h$ (if reference $r \in h$).

Problem formulation



The set of hidden entities is $E = \{e_1, e_2, e_3, e_4\}$ where

- $r_1.E = r_5.E = e_1$,
- $r_2.E = r_6.E = e_2$,
- $r_3.E = r_7.E = e_3$,
- $r_4.E = r_8.E = e_4$

Resolution by clustering

The goal is to group all the references corresponding to the same entity into one cluster. The membership of a reference to a cluster is represented with $r.C$. All references from the cluster are of the same type.

1. At the beginning, each reference belongs to the separate cluster.
2. At each step, the cluster pair, with the highest similarity to be the same entity, is merged.

The general similarity is defined as:

$$sim(c_i, c_j) = (1 - \alpha) \times \text{sim}_{attr}(c_i, c_j) + \alpha \times \text{sim}_{rel}(c_i, c_j)$$

where $0 \leq \alpha \leq 1$

Attribute a relational similarity

Attribute similarity
Any basic similarity measure for two reference attributes. The similarity for two clusters is calculated between two most representative attributes of those clusters.

Relational similarity
The measure between two clusters considering the clusters that they link to via observed edges.

- Edge detail similarity
- Neighborhood similarity

Edge detail similarity

Each cluster is associated with the set of hyper-edges:

$$c.H = \{h | r.H = h \wedge r.C = c\}$$

The similarity between two edges is defined as:

$$\text{sim}(h_i, h_j) = \sum_t (\text{sim}_t(h_i, h_j))$$

where:

$$\text{sim}_t(h_i, h_j) = \text{Jaccard}(\pi_t(h_i), \pi_t(h_j)))$$

and

$$\pi_t(h) = \{c | r.C = c \wedge c.T = t \wedge r.H = h\}$$

The final similarity is defined as:

$$\text{sim}_{rel}(c_i, c_j) = \max(h_i, h_j)\{\text{sim}(h_i, h_j)\}$$

where $h_i \in c_i.H, h_j \in c_j.H$

Neighborhood similarity

The similarity between two clusters is defined as:

$$\text{sim}_{rel}(c_i, c_j) = \text{Jaccard}(N_t(c_i), N_t(c_j)))$$

where $N_t(c) = \cup_m \pi_t(h), h \in c.H$
The obtained neighborhoods are multisets.

Implementation

Greedy agglomerative clustering that merges closest cluster pair at each step.
All candidate pairs are sorted by their similarities in a priority queue - blocking approach.
During the initial phase, references with the identical attributes $v_1 = v_2$ or with a reference which is initialed form of the other are merged.

Datasets and baseline methods

CiteSeer dataset contains 2892 references with 1165 authors, contained in 1504 documents.
arXiv dataset contains 58515 references with 9200 authors, contained in 29555 papers.
Baseline method *ATTR* based on SoftTF-IDF where the secondary distance measures can be Jaro-Winkler, Jaro or Scaled Levenstein distance.

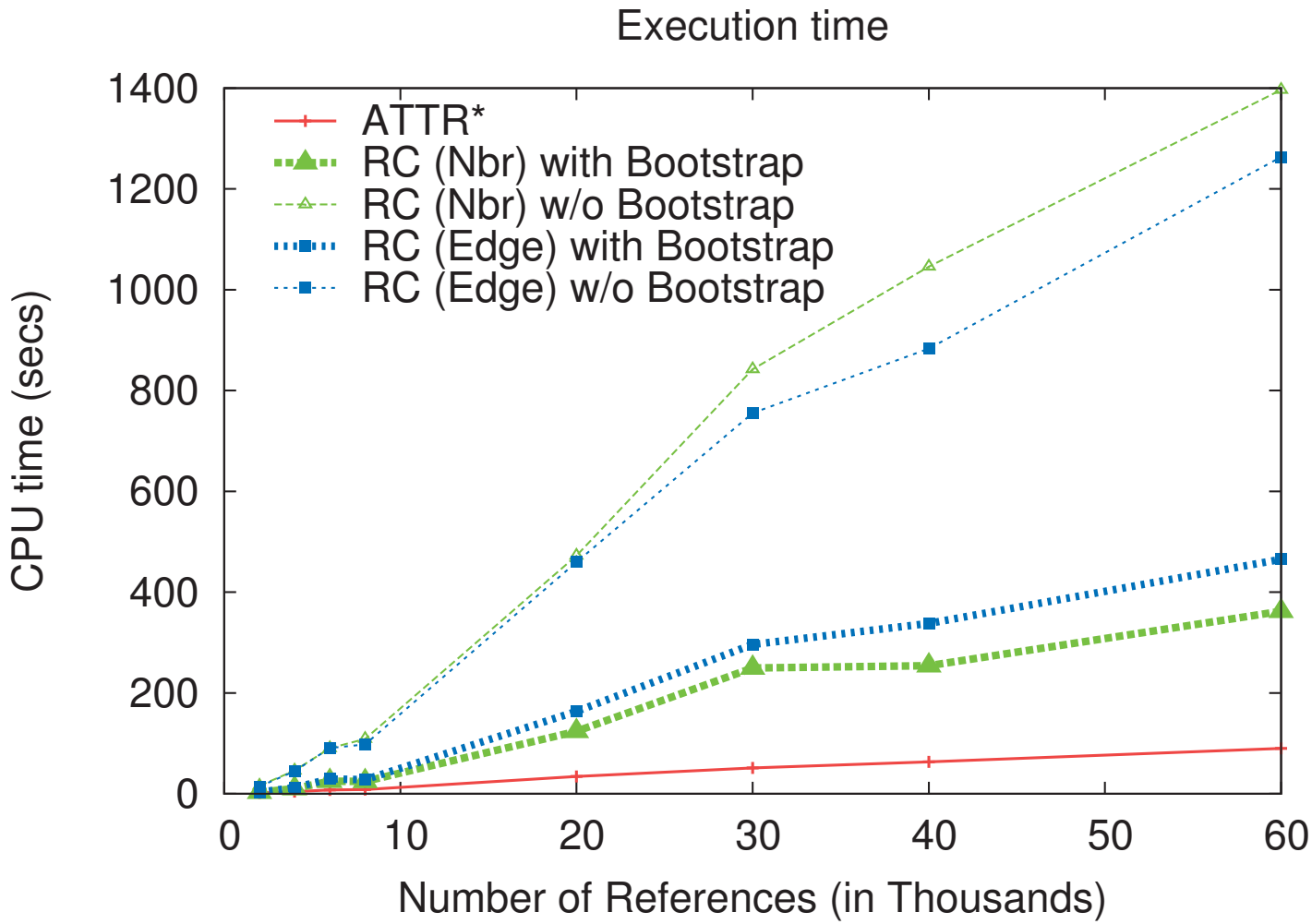
Accuracy results with different similarity measures

	CiteSeer			HEP		
	SL	JA	JW	SL	JA	JW
ATTR	0.980	**0.981**	0.980	**0.976**	0.976	0.972
ATTR*	0.989	**0.991**	0.990	**0.971**	0.968	0.965
RC(Nbr)	**0.994**	**0.994**	**0.994**	0.979	**0.981**	0.981
RC(Edge)	**0.995**	**0.995**	**0.995**	0.982	**0.983**	0.982

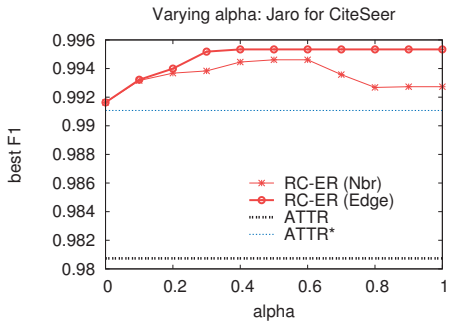
Precision, recall and F1 results for both datasets

	CiteSeer			HEP		
	P	R	F1	P	R	F1
ATTR	0.990	0.971	0.981	0.987	0.965	0.976
ATTR*	0.992	0.988	0.991	0.976	0.965	0.971
RC(Nbr)	0.998	0.991	**0.994**	0.990	0.972	**0.981**
RC(Edge)	0.997	0.993	**0.995**	0.992	0.974	**0.983**

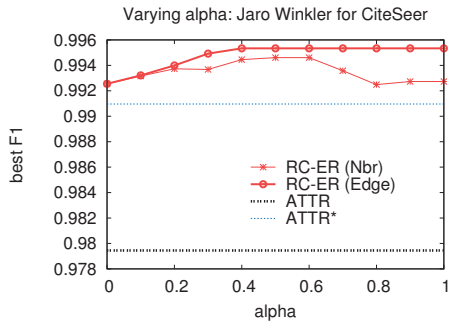
Performance



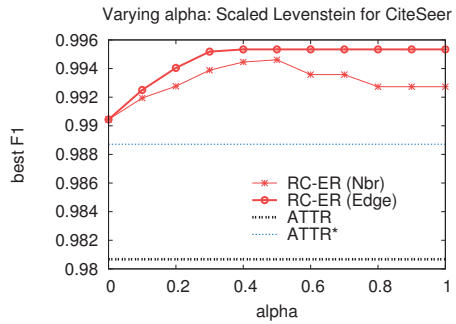
Attribute vs relational similarity effects on accuracy



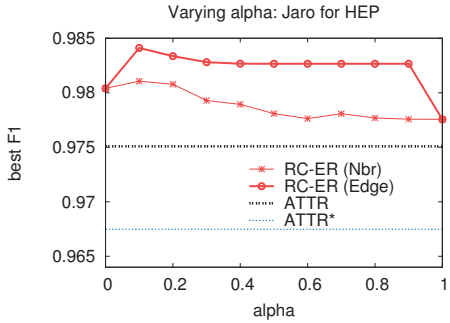
(a)

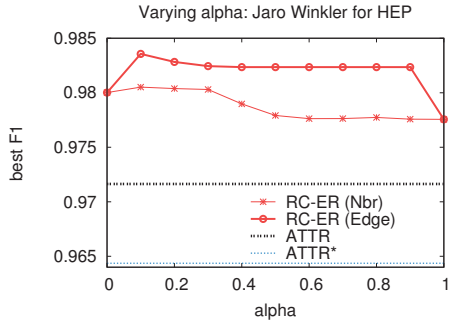


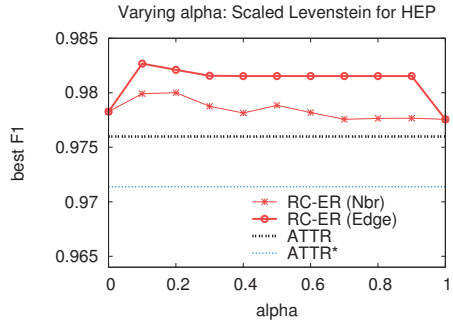
(b)



(c)







Conclusions

- Introduced two relational similarity measures
- Relational similarity in combination with attributes similarity outperform other non-relational approaches.
- Successful usage of bootstrapping and blocking approach for improved performance.