Hierarchical Cost-sensitive Web Resource Acquisition for Record Matching

Yee Fan Tan and **Min-Yen Kan**

School of Computing
National University of Singapore

{tanyeefa,kanmy}@comp.nus.edu.sg

Contents

Record Matching using Web Resources

Example: linkage of short forms (*A*) to long forms (*B*)

A		*B*
KDD		Knowledge Discovery and Data Mining
PAKDD	? ↔	Pacific-Asia Conference on Knowledge Discovery and Data Mining
DKMD		Research Issues on Data Mining and Knowledge Discovery
KDID		Knowledge Discovery in Inductive Databases

For each $(a, b) \in A \times B$, to classify whether *a* and *b* is a match

---

Web resources: conference or workshop websites, etc.

Google

Web Intelligence

WI-IAT

Web Intelligence 2009 and Intelligent Agent Technology 2009
2 Feb 2009 ... The 2009 IEEE / WIC / ACM Joint Conference on **WI/IAT** will provide a leading international forum to bring together researchers and ...
www.**wi-iat**09.disco.unimib.it/ - Cached - Similar

Hit counts

Webpage similarity

Cooccurrence counts

Snippet similarity

URL information

Record Matching using Web Resources

- **Typical scheme for comparing records *a* and *b***
  - Query search engine with queries of the form:

    a, b, and/or $a \wedge b$

    (Optionally, append additional terms or tokens)
  - Extract information and construct test instance $\mathbf{x}_{a,\,b}$
  - Classify $\mathbf{x}_{a,\,b}$ as match/mismatch (e.g., using SVM)

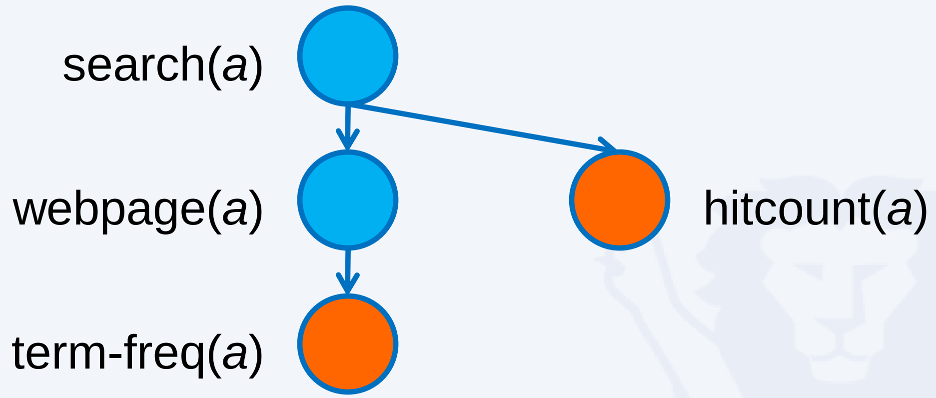
- **Applications (IR, NLP, IE, data mining, linkage, etc.)**
  - Mihalcea and Moldovan (1999), Cimiano et al. (2005), Tan et al. (2006), Bollegala et al. (2007), Elmacioglu et al., (2007), Oh and Isahara (2008), Kalashnikov et al. (2008)

Cost of Acquiring Web Resources

- **Acquiring web resources are slow and may incur other access costs**
  - Google SOAP Search takes a day or more to query 1000 records individually, even longer for pairwise queries
- **For large datasets, not feasible to query and download everything**
  - Must acquire only selected web resources

Hierarchical Dependencies

- **Basically ignored in existing work**
  - Including work on cost-sensitive attribute value acquisition, e.g., Ling et al. (2006), Saar-Tsechansky et al. (2009)

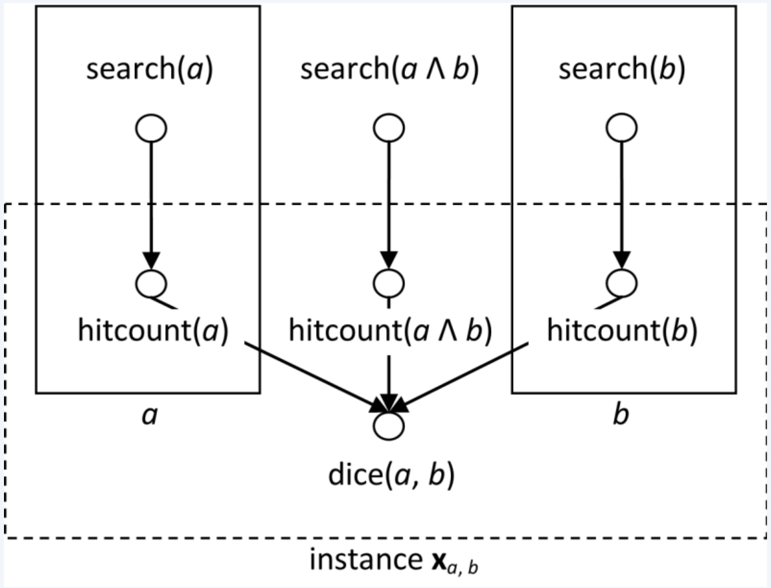


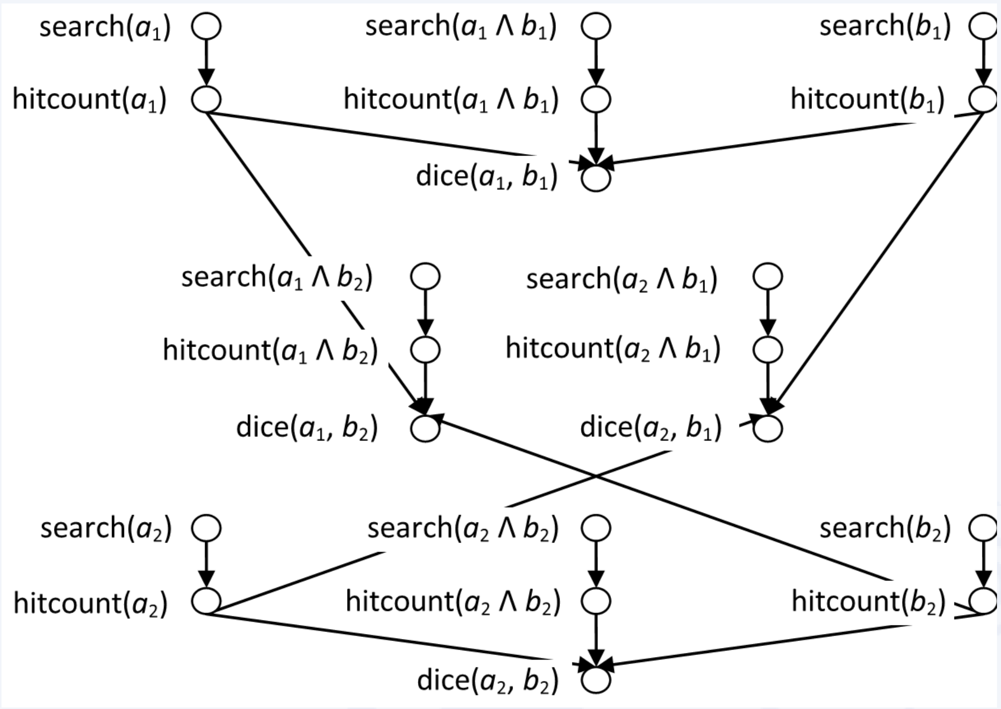
  - Same resource may be used to extract different types of attribute values
  - Different instances can share attribute values

Contents

- **Background and Motivation**
- **Cost-sensitive Record Acquisition Framework**
  - Resource dependency graph
  - Resource acquisition problem
  - Challenges for record matching problems
- **Algorithm for Record Matching Problems**
- **Evaluation**
- **Conclusion**

Resource Dependency Graph

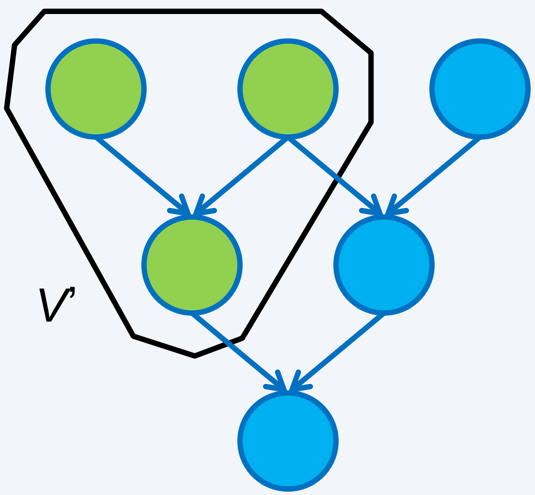


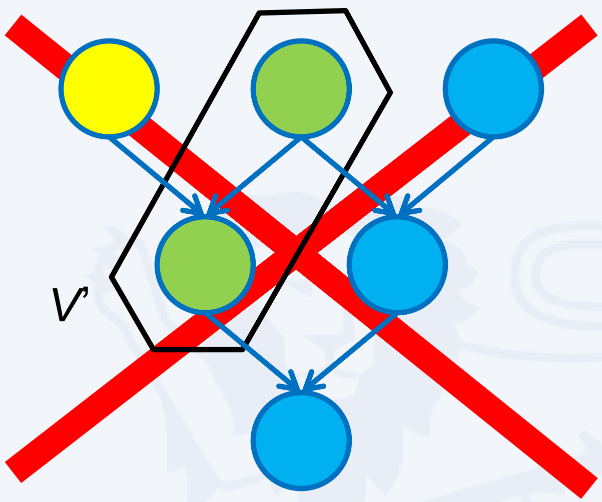


- Directed acyclic graph $G = (V, E)$
- Vertex types T

Resource Acquisition Graph

- **Feasible vertex set *V'* ⊆ *V***
  - *V'* can be acquired as-is without violating acquisition dependencies





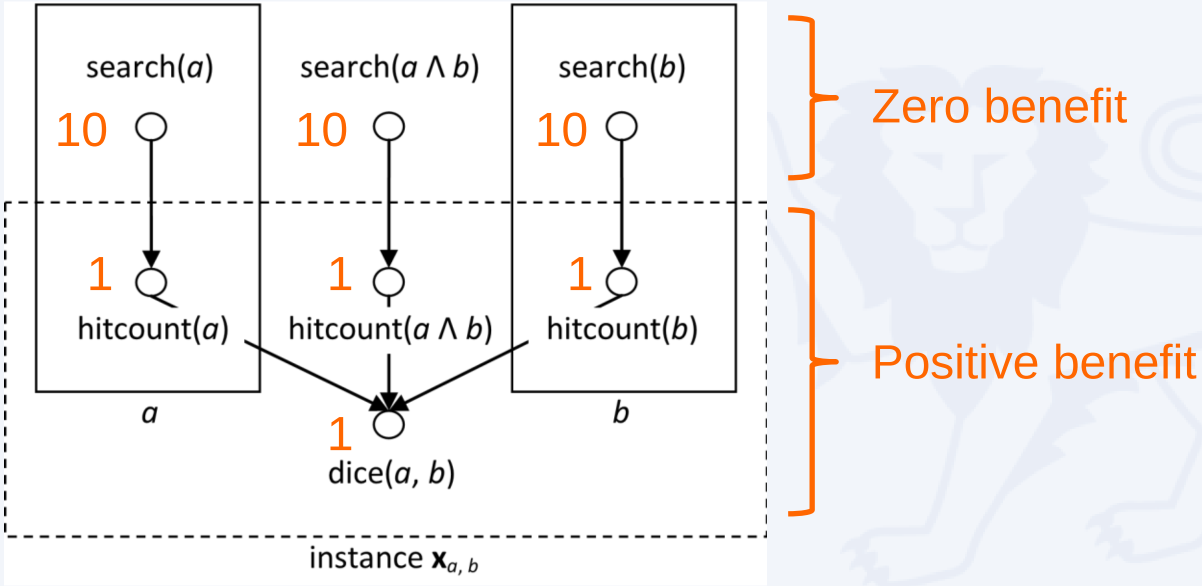
Cost and Benefit Functions

Acquisition cost function

- search(.) cost 10 each
- webpage(.) cost 100 each
- All other cost 1 each

Benefit function

- Positive benefit for vertices that are attribute values in test instances
- Zero benefit otherwise

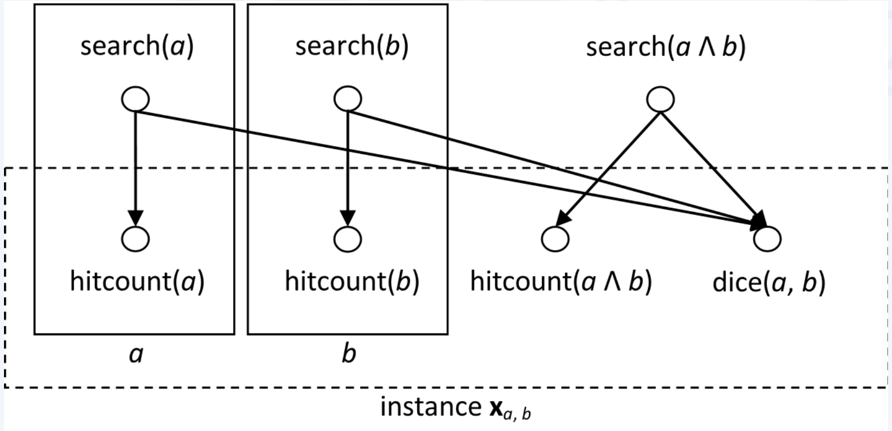


Resource Acquisition Problem

- **Given**
  - Resource dependency graph $G = (V, E)$
  - Acquisition cost function *cost*: $V \rightarrow \mathbf{R}^{+} \cup \{0\}$
  - Benefit function *benefit*: $F(G) \rightarrow \mathbf{R}$
  - Vertex type function *type*: $V \rightarrow T$
  - Budget *budget*
- **Acquire feasible vertex set *V'* with *cost*(*V'*) ≤ *budget* to maximize *obj*(*V'*) = *benefit*(*V'*) – *cost*(*V'*)**
- **Can be applied on many problems and variants**

Challenges for Record Matching

- **Heuristic search required**
  - Not feasible to enumerate all possible *V'* to acquire
- **Many local maxima**
  - *V'* = φ is a local maxima
  - Easy for states to be revisited
- **No guidance as to which root and child vertices to acquire**
  - Each search(.) has the same objective value
  - Vertex out-degree can be very large



Contents

- **Background and Motivation**
- **Cost-sensitive Record Acquisition Framework**
- **Algorithm for Record Matching Problems**
  - Application of Tabu search
  - More intelligent legal moves
  - Propagation of benefit values
- **Evaluation**
- **Conclusion**

Tabu Search

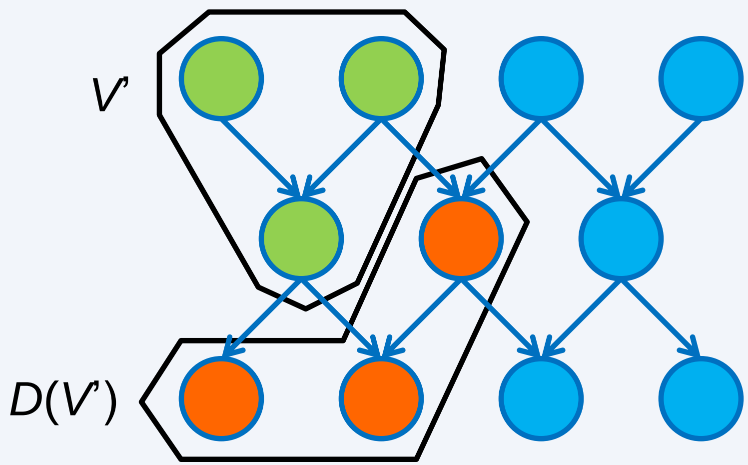
- **Simple hill climbing**
  - In each iteration, move to state *V'* with best *obj*(*V'*)

- **Simple Tabu search (Glover, 1990)**
  - Simple hill climbing with Tabu list
  - Tabu list disallows moves that reverse effect of moves that have been made for a limited number of iterations
    - e.g., after adding a vertex *v* to *V'*, disallow removing *v* from *V'* for next *k* iterations

1. Avoids getting stuck in local maxima
2. Better exploration of state space

Legal Moves

Let *D*(*V'*) be the set of children from any vertex in *V'* that are not in *V'*

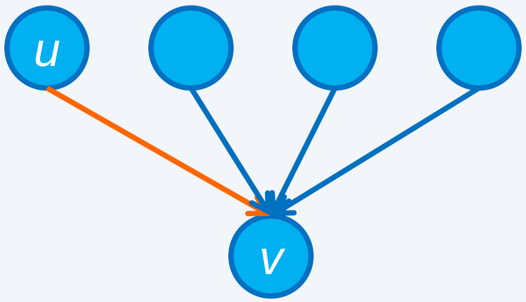


If *V'* is empty, then set *D*(*V'*) = *V*

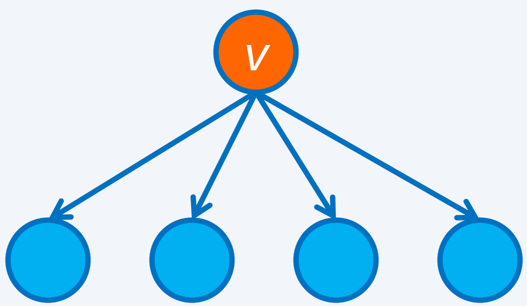
- **Add(*v*)**
  - Add *v* from *D*(*V'*) and its ancestors into *V'*
- **Remove(*v*)**
  - Remove *v* and its descendants from *V'*
- **AddType(*t*)**
  - Add vertices of type *t* from *D*(*V'*) into *V'* (as many as the budget allows)

Propagated Benefit

Benefit propagation from leaf to root: vertices → edges → vertices → edges → …



$$prop\text{-}benefit(u \to v) = \frac{\lambda \cdot (prop\text{-}benefit(v) - cost(v))}{|pa(v)|}$$



$$prop\text{-}benefit(v) = benefit(v) + max\text{-}pb(v) + \beta \cdot rest\text{-}pb(v)$$

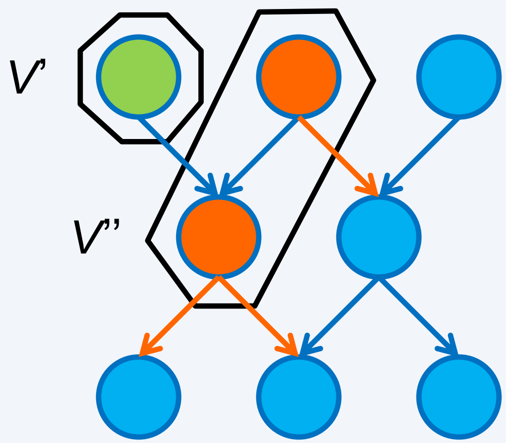
where

$$max\text{-}pb(v) = \max_{u \in ch(v)} prop\text{-}benefit(v \to u)$$

$$rest\text{-}pb(v) = \sum_{u \in ch(v)} prop\text{-}benefit(v \to u) - max\text{-}pb(v)$$

Surrogate Benefit and Objective

- **Current state *V'* and we consider adding *V''***
- **Surrogate benefit function**



$$surr\text{-}benefit(V', V'') = \sum_{v \to u \in E'} prop\text{-}benefit(v \to u)$$

- **Surrogate objective function**
  - *surr-obj*(*V'*, *V''*) = *surr-benefit*(*V'*, *V''*) – *cost*(*V''*)

Benefit Function for SVM

- We use SVM to classify test instances
- Apply *cost-sensitive attribute value acquisition* work in Tan and Kan (2010) to compute benefit function

Contents

Experimental Setup

Algorithms

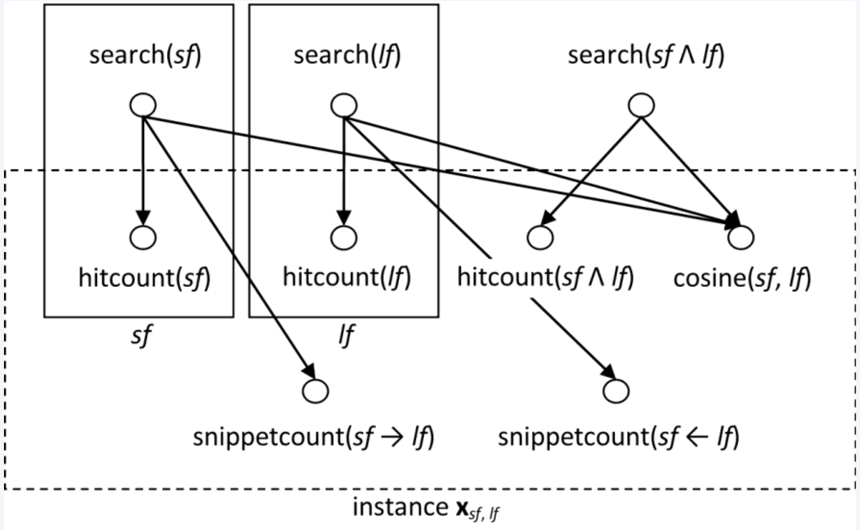
- **Baselines**
  - Random
  - Least cost
  - Best benefit
  - Best cost-benefit ratio
  - Best type
- **Our algorithm**
- **Manual (with access to solution set)**

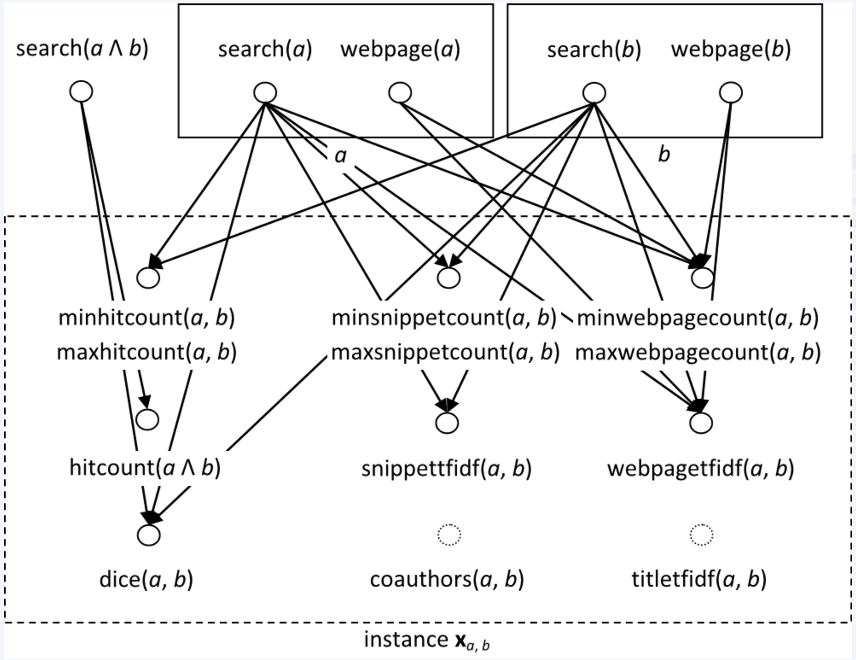
Evaluation Methodology

- **Start with no vertices acquired**
- **For each iteration**
  - Run acquisition algorithm with budget
  - Record
    - Total acquisition cost
    - Total misclassification cost
- **Run for 200 iterations**

Datasets

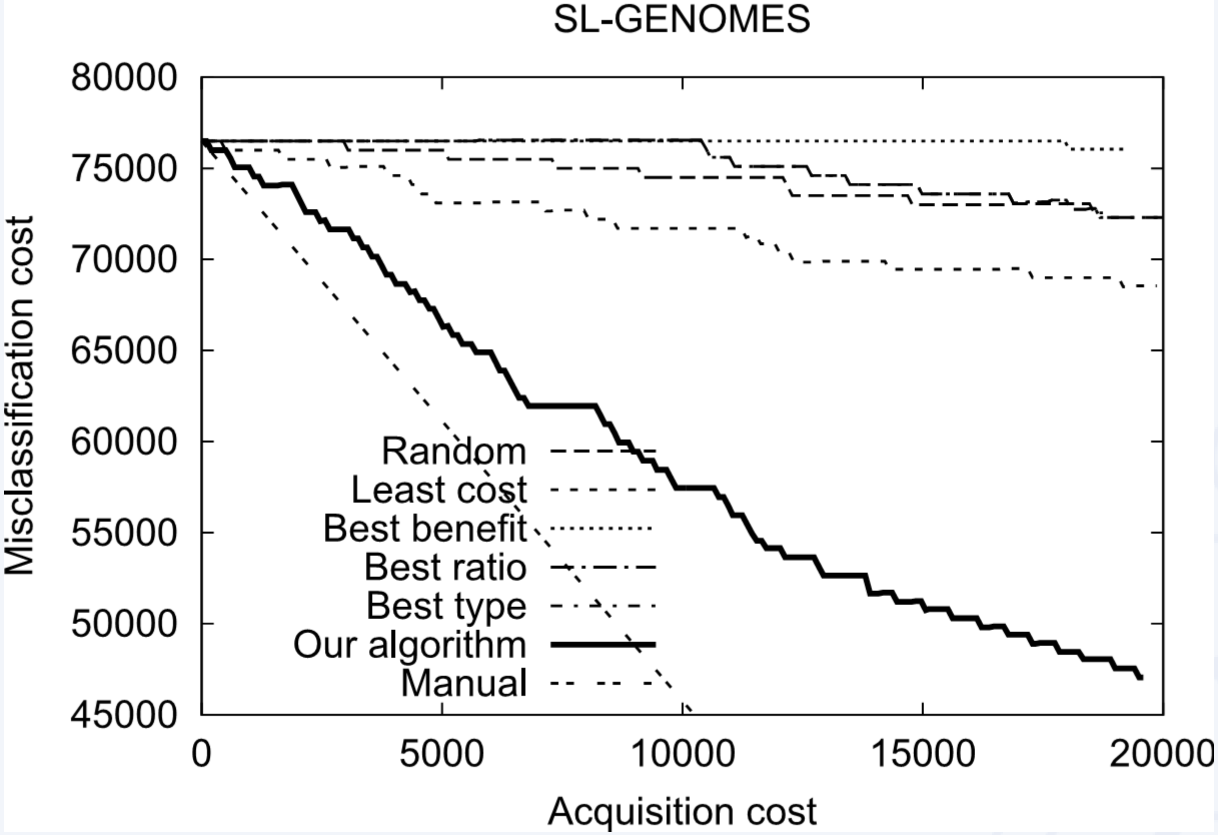
- **Linking short forms to long forms**
  - SL-GENOMES: Human genomes
  - SL-DBLP: Publication venues
- **Disambiguating author names in publication lists**
  - AUTHOR-DBLP: 352 ambiguous author names





Results

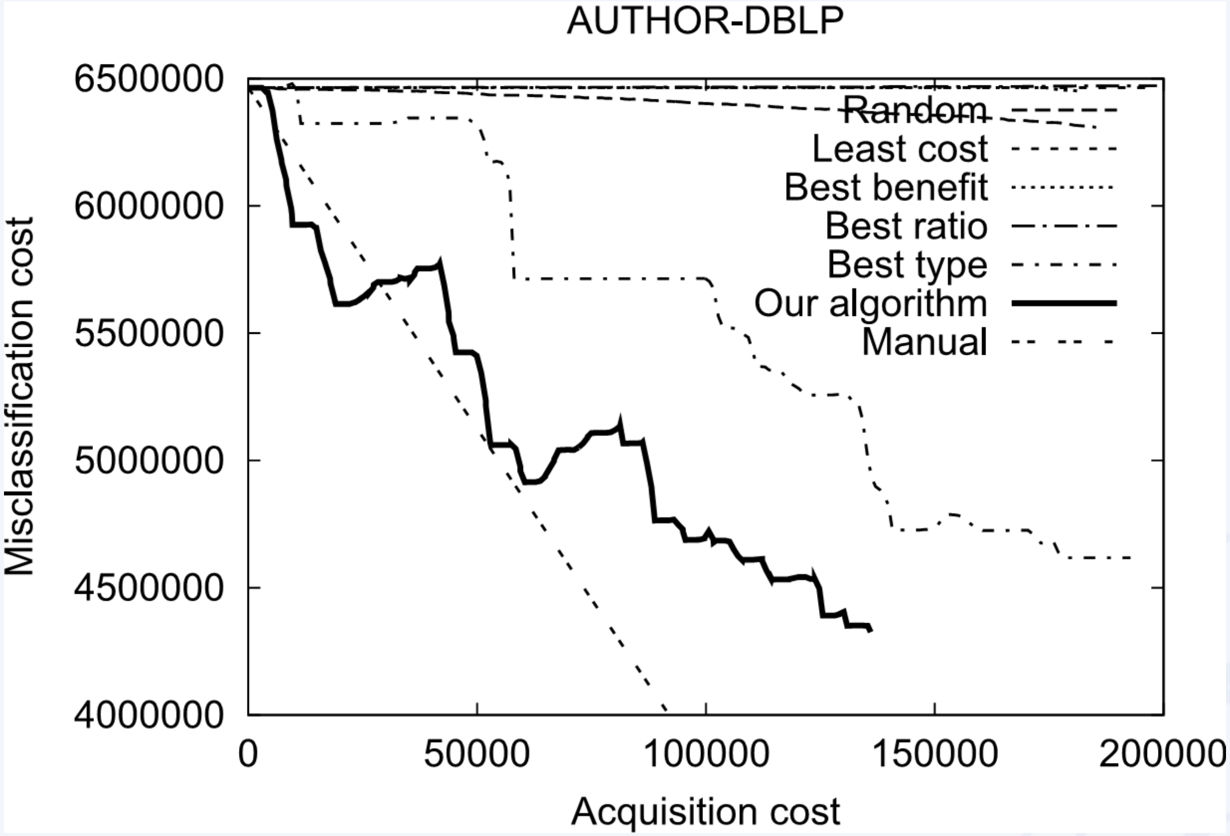
- **SL-GENOMES**



Average improvement of difference with Manual over second best algorithm: 49%

Results

- **AUTHOR-DBLP**



Average improvement of difference with Manual over second best algorithm: 74%

Search Engine Queries

- **Saved 90% of acquisitions of search(.) vertices for test instances correctly classified by our algorithm**
  - Mainly acquired search(*a*) or search(*b*) vertices, which services many test instances
  - Typically, no need to acquire search($a \wedge b$) for correct classification

Contents

- Background and Motivation
- Cost-sensitive Record Acquisition Framework
- Algorithm for Record Matching Problems
- Evaluation
- **Conclusion**
  - Contributions

Contributions

Framework for cost-sensitive acquisitions of web resources with hierarchical dependencies

- Model using resource dependency graph
- Versatile, applicable to many problems
- Acquisition algorithm for record matching problems
- Effective on record matching problems of different domains

Thank you for your attention!