Automatic Methods for Disambiguating Author Names in Bibliographic Data Repositories

Anderson Almeida Ferreira

Marcos André Gonçalves

Alberto H. F. Laender

Outline

- Introduction
- Basic concepts
- Taxonomy
  - Author grouping methods
  - Author assignment methods
- Methods
  - HHC
  - SAND
  - INDi
- SyGAR
- Open challenges

Introduction

- Digital libraries: BDBComp, DBLP, Citeseer,...
  - Facilitate literature research and discovery
  - List millions of bibliographic citation records
  - Have become an important source of information
  - Allow the search and discovery of relevant publications in a centralized manner

Introduction

- Studies based on digital library content can lead to interesting results, such as:
  - Coverage of topics
  - Research tendencies
  - Quality and impact of publications
  - Patterns of collaboration in social networks
- These studies are used by funding agencies.
- Digital libraries must provide high quality content.

Author Name Ambiguity Problem

- Has required a lot of attention from the digital library research community
- Occurs when
  - The same author publishes articles under distinct names (synonyms)
  - Distinct authors publish articles with similar names (homonyms)

home | browse | search | about

dblp
computer science bibliography

DBLP FAQ: How does the 'author search' work?

Name: Mohammed Zaki [Submit] [Reset]

A query is interpreted as a **set of prefixes of name parts**. If you enter a few words, you get the names which include these words as prefixes of some name parts:

query = A Meyer — **answers** = Achim Meyer, Andrea Meyer, Anne Meyer, Hans-Albert Meyer, A. Meyers, Anton Smith-Meyer, ...

query = Ar b c — **answers** = Clark B. Archer, Arnold B. Calica, Arnab B. Chowdry, Armin B. Cremers, ...

XXIX SBBD
Simpósio Brasileiro de Banco de Dados

dblp
computer science bibliography

Search Results for ' mohammed zaki '

- Mohammed Zaki Ahmed
- Mohammed Zaki Hasan
- Mohammed Zaki Hussein
- Mohammed Zaki
- Mohammed J. Zaki
- Mohammed Javeed Zaki

May refer to the same person

Mohammed Zaki

M. Zaki

Systems and Computer Engineering Department, Al-Azhar University, Nasr City, Egypt

List of publications from the DBLP Bibliography Server - FAQ

Facets and more with CompleteSearch

Coauthor Index - Ask others: ACM DL/Guide - CiteSeer - CSB - Google - M

mohammedzaki:

Mohammed Zaki from Al-Azhar University, Nasr City, Cairo, Egypt

		2006
26	EE	Ashraf Elgohary, Tarek S. Sobh, Mohammed Zaki: Design of an enhancement for SSL/TLS protocols. Computers & Security 25(4): 297-306 (2006)
		2005
25	EE	Karlton Sequeira, Mohammed Zaki: SCHISM: a new approach to interesting subspace mining. IJBIDM 1(2): 137-160 (2005)
24	EE	Mohammed Zaki, Tarek S. Sobh: NCDS: data min ... ics. Information & Software Technology 47(3): 189-19
23	EE	M. Zaki, Tarek S. Sobh: Attack abstraction using a ... of Intelligent and Fuzzy Systems 16(2): 141-150 (200

Mohammed Javeed Zaki from the Department of Computer Science, Rensselaer Polytechnic Institute, USA

Refine by AUTHOR

Tarek S. Sobh (5)
Abdallah El-Ramsisi (3)
Rostom Omran (3)
G. Osman (2)
[top 4] [all 27]

Refine by VENUE

Journal of Systems and Software (JSS) (9)
Comput. Lang. (CL) (3)
Journal of Intelligent and Fuzzy Systems (JIFS) (3)

The author name disambiguation task

An illustrative example

A reference to an author

Citation Id	Citation
c_1	**(r_1) S. Godbole, (r_2) I. Bhattacharya, (r_3) A. Gupta, (r_4) A. Verma**. Building re-usable dictionary repositories for real-world text mining. CIKM, 2010.
c_2	**(r_5) Indrajit Bhattacharya, (r_6) Shantanu Godbole, (r_7) Ajay Gupta, (r_8) Ashish Verma, (r_9) Jeff Achtermann, (r_{10}) Kevin English**. Enabling analysts in managed services for CRM analytics. KDD, 2009.
c_3	**(r_{11}) T. Nghiem, (r_{12}) S. Sankaranarayanan, (r_{13}) G. E. Fainekos, (r_{14}) F. Ivancic, (r_{15}) A. Gupta, (r_{16}) G. J. Pappas**. Monte-carlo techniques for falsification of temporal properties of non-linear hybrid systems. HSCC, 2010.
c_4	**(r_{17}) William R. Harris, (r_{18}) Sriram Sankaranarayanan, (r_{19}) Franjo Ivancic, (r_{20}) Aarti Gupta**. Program analysis via satisfiability modulo path programs. POPL, 2010.

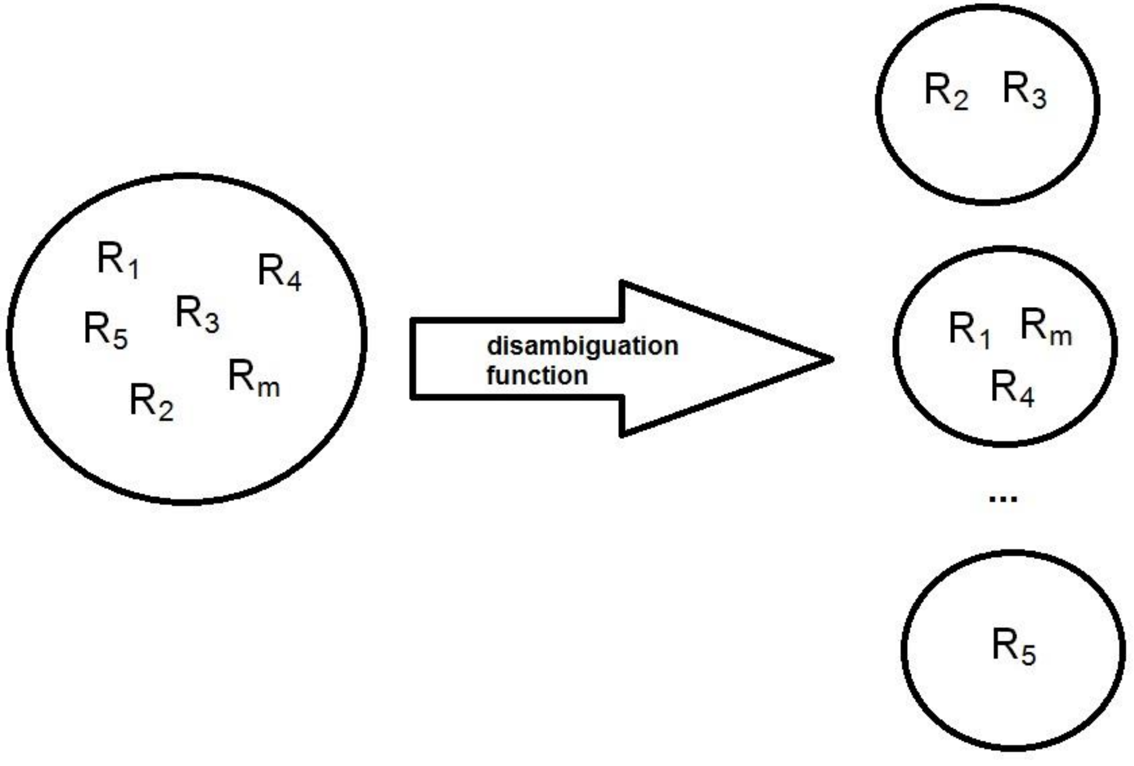
The author name disambiguation task

Definitions

- Citation record
  - A citation record *c* is a set of bibliographic data, such as author names, work title, publication venue title, publication year, etc., that is pertinent to a particular article.
- Reference
  - Each author name element is a *reference r* to an author. We associate a list of attributes to each reference *r*.
  - *r.author* – the author name attribute
  - *r.coauthors* - the other author names in a citation record
  - *r.title* - the work title attribute
  - *r.venue* - the publication venue title attribute
  - other attributes such as publication year, affiliation, e-mail, …
- Ambiguous group
  - An ambiguous group is a group of references whose value of the author name attribute are ambiguous.

The author name disambiguation task

Objective of a disambiguation method:



The author name disambiguation task

Preprocessing

Citation Id	Citation
c_1	**(r_1) S. Godbole, (r_2) I. Bhattacharya, (r_3) A. Gupta, (r_4) A. Verma**. Building re-usable dictionary repositories for real-world text mining. CIKM, 2010.
c_2	**(r_5) Indrajit Bhattacharya, (r_6) Shantanu Godbole, (r_7) Ajay Gupta, (r_8) Ashish Verma, (r_9) Jeff Achtermann, (r_{10}) Kevin English**. Enabling analysts in managed services for CRM analytics. KDD, 2009.
c_3	**(r_{11}) T. Nghiem, (r_{12}) S. Sankaranarayanan, (r_{13}) G. E. Fainekos, (r_{14}) F. Ivancic, (r_{15}) A. Gupta, (r_{16}) G. J. Pappas**. Monte-carlo techniques for falsification of temporal properties of non-linear hybrid systems. HSCC, 2010.
c_4	**(r_{17}) William R. Harris, (r_{18}) Sriram Sankaranarayanan, (r_{19}) Franjo Ivancic, (r_{20}) Aarti Gupta**. Program analysis via satisfiability modulo path programs. POPL, 2010.

The author name disambiguation task

Preprocessing – stop-word removal

Citation Id	Citation
c_1	**(r_1) S. Godbole, (r_2) I. Bhattacharya, (r_3) A. Gupta, (r_4) A. Verma**. building usable dictionary repositories real world text mining. CIKM, 2010.
c_2	**(r_5) Indrajit Bhattacharya, (r_6) Shantanu Godbole, (r_7) Ajay Gupta, (r_8) Ashish Verma, (r_9) Jeff Achtermann, (r_{10}) Kevin English**. enabling analysts managed services crm analytics. KDD, 2009.
c_3	**(r_{11}) T. Nghiem, (r_{12}) S. Sankaranarayanan, (r_{13}) G. E. Fainekos, (r_{14}) F. Ivancic, (r_{15}) A. Gupta, (r_{16}) G. J. Pappas**. monte carlo techniques falsification temporal properties linear hybrid systems. HSCC, 2010.
c_4	**(r_{17}) William R. Harris, (r_{18}) Sriram Sankaranarayanan, (r_{19}) Franjo Ivancic, (r_{20}) Aarti Gupta**. program analysis satisfiability modulo path programs. POPL, 2010.

The author name disambiguation task

Preprocessing - stemming

Citation Id	Citation
c_1	**(r_1) S. Godbole, (r_2) I. Bhattacharya, (r_3) A. Gupta, (r_4) A. Verma**. build usabl dictionari repositori real world text mine. CIKM, 2010.
c_2	**(r_5) Indrajit Bhattacharya, (r_6) Shantanu Godbole, (r_7) Ajay Gupta, (r_8) Ashish Verma, (r_9) Jeff Achtermann, (r_{10}) Kevin English**. enabl analyst manag servic crm analyt. KDD, 2009.
c_3	**(r_{11}) T. Nghiem, (r_{12}) S. Sankaranarayanan, (r_{13}) G. E. Fainekos, (r_{14}) F. Ivancic, (r_{15}) A. Gupta, (r_{16}) G. J. Pappas**. mont carlo techniqu falsif tempor properti linear hybrid system. HSCC, 2010.
c_4	**(r_{17}) William R. Harris, (r_{18}) Sriram Sankaranarayanan, (r_{19}) Franjo Ivancic, (r_{20}) Aarti Gupta**. program analysi satisfi modulo path program. POPL, 2010.

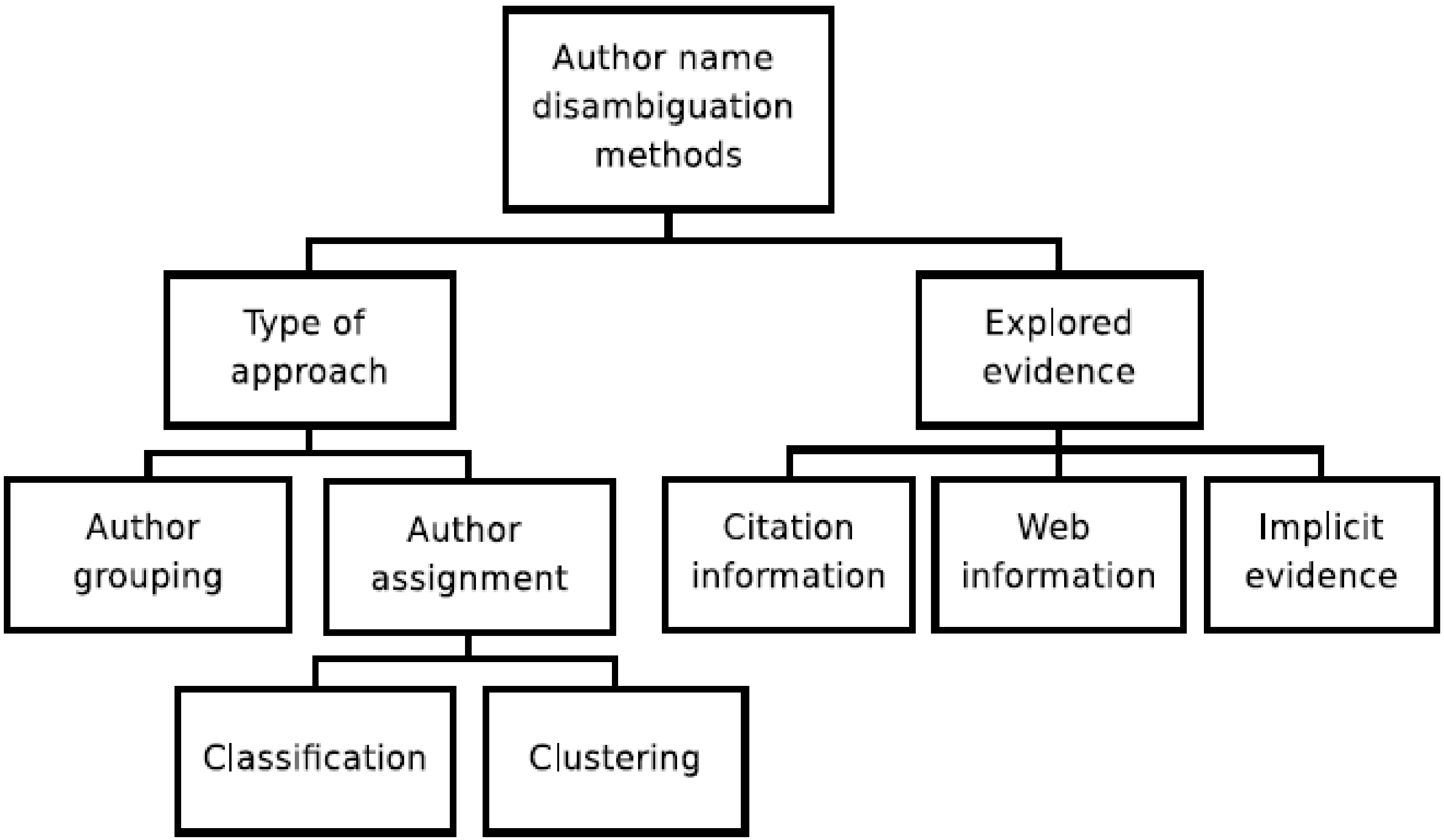
The author name disambiguation task

- { (r_1) S. Godbole, (r_2) I. Bhattacharya, (r_3) A. Gupta, (r_4) A. Verma, (r_5) Indrajit Bhattacharya, (r_6) Shantanu Godbole, (r_7) Ajay Gupta, (r_8) Ashish Verma, (r_9) Jeff Achtermann, (r_{10}) Kevin English, (r_{11}) T. Nghiem, (r_{12}) S. Sankaranarayanan, (r_{13}) G. E. Fainekos, (r_{14}) F. Ivancic, (r_{15}) A. Gupta, (r_{16}) G. J. Pappas, (r_{17}) William R. Harris, (r_{18}) Sriram Sankaranarayanan, (r_{19}) Franjo Ivancic, (r_{20}) Aarti Gupta }

- $a_1 = \{(r_1), (r_6)\}$ - Shantanu Godbole
- $a_2 = \{(r_2), (r_5)\}$ - Indrajit Bhattacharya
- $a_3 = \{(r_3), (r_7)\}$ - Ajay Gupta
- $a_4 = \{(r_4), (r_8)\}$ Ashish Verma
- $a_5 = \{(r_9)\}$ - Jeff Achtermann
- $a_6 = \{(r_{10})\}$ - Kevin English
- $a_7 = \{(r_{11})\}$ - T. Nghiem
- $a_8 = \{(r_{12}), (r_{18})\}$ Sriram Sankaranarayanan
- $a_9 = \{(r_{13})\}$ - G. E. Fainekos
- $a_{10} = \{(r_{14}), (r_{19})\}$ - Franjo Ivancic
- $a_{11} = \{(r_{15}), (r_{20})\}$ - Aarti Gupta
- $a_{12} = \{(r_{16})\}$ - G. J. Pappas
- $a_{13} = \{(r_{17})\}$ - William R. Harris

Author name disambiguation methods

A taxonomy



Author name disambiguation methods

Type of approach

- Author Grouping Methods
  - Apply a similarity function in order to group references using a clustering technique.
- Author Assignment Methods
  - Directly assign each reference to a given author by constructing a model that represents the author using either a supervised classification technique or a model-based clustering technique.

Author name disambiguation methods

Author Grouping Methods

- The similarity function
  - Aims to determine how similar two references (or groups of references) to authors are.
  - May be:
    - Predefined
    - Learned using a supervised machine learning technique
    - Extracted from the relationships among authors and coauthors

Author name disambiguation methods

Similarity function

- Using predefined function
  - A specific predefined similarity function *S* embedded in the algorithm to check whether two references or groups of references refer to the same author.
  - Examples of *S* includes:
    - Levenshtein distance
    - Jaccard coefficient
    - Cosine similarity
    - Soft-TFIDF
    - …
  - Ad-hoc combinations of functions have also been used.

Author name disambiguation methods

Similarity function

- Learning a Similarity Function
  - The methods receive a set of pairs of references (the training data) along a special variable that informs whether these two corresponding references refer to the same author.
  - A pair of references, r_i and r_j is usually represented by a similarity vector s_{ij}.
  - Each similarity vector s_{ij} is composed of a set of features $\{f_1, f_2, \ldots, f_q\}$.
  - Each feature f_p represents a comparison between attributes $r_i.A_l$ and $r_j.A_l$ of two references, r_i and r_j.
  - The value of each feature is usually defined using other functions
  - The training data is used to produce a similarity function
  - Usually need many examples and sufficient features to work well.

Author name disambiguation methods

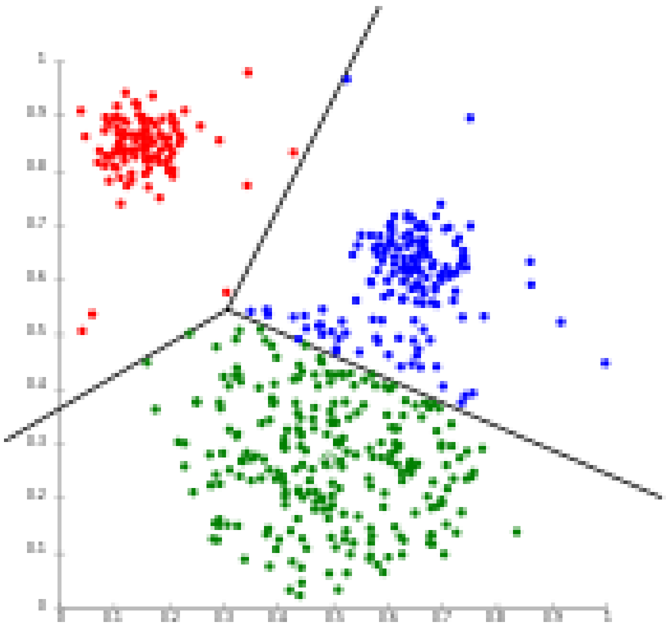
Similarity function

- Exploiting Graph-based Similarity Functions
  - Usually create a coauthorship graph $G=(V, E)$ for each ambiguous group.
  - Each element of the author name and coauthor name attributes is represented by a vertex $v \in V$.
  - The same coauthor names are usually represented by only a unique vertex.
  - For each coauthorship an edge $\{v_i,v_j\} \in E$ is created.
  - The weight of each edge $\{v_i,v_j\}$ is related to the amount of articles coauthored by the corresponding author names
  - A graph-based metric (e.g., shortest path) may be combined with other similarity functions on the attributes of the references to authors or used as a new feature in the similarity vectors.

Author name disambiguation methods

Clustering Techniques

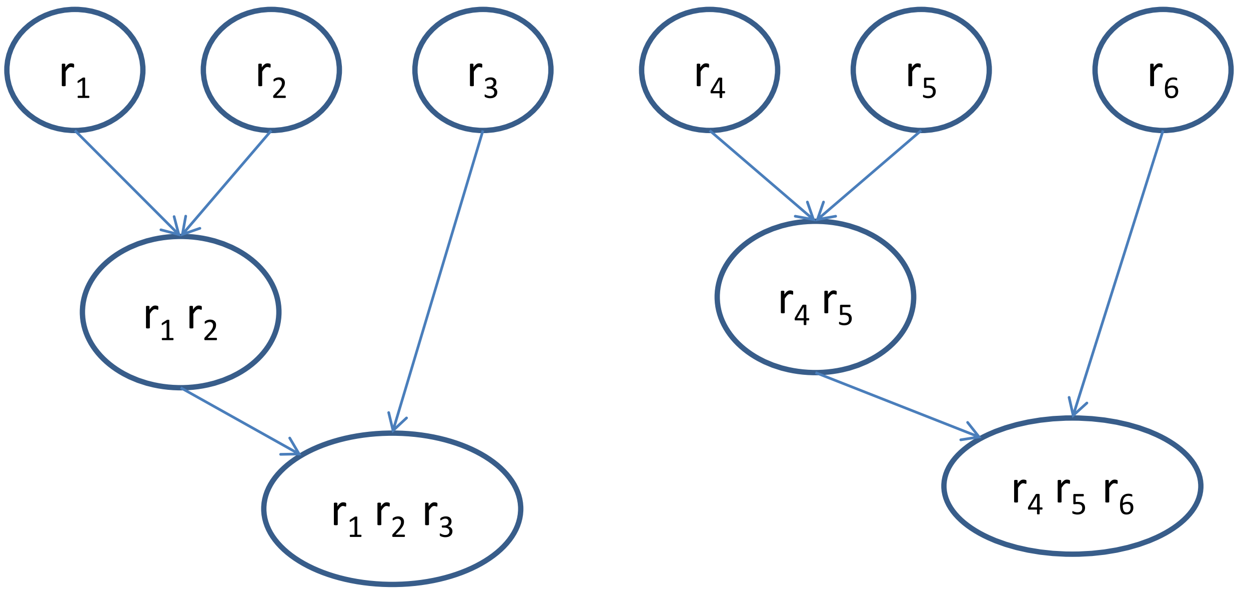
- Partitioning Clustering Technique



Author name disambiguation methods

Clustering Techniques

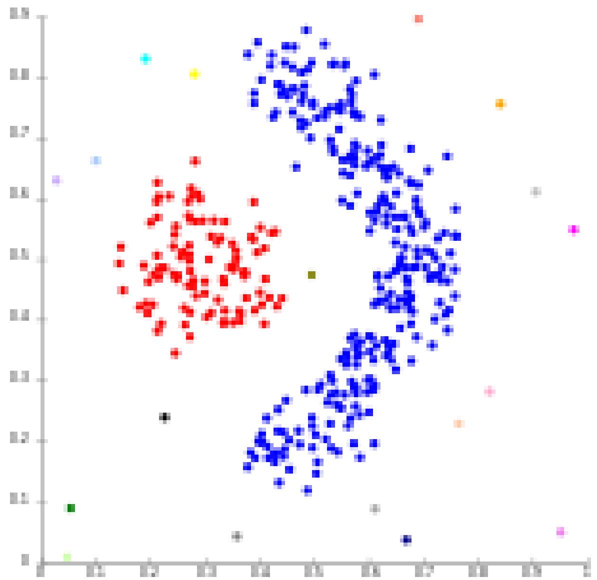
- Hierarchical Agglomerative Clustering



Author name disambiguation methods

Clustering Techniques

- Density-based Clustering



Author name disambiguation methods

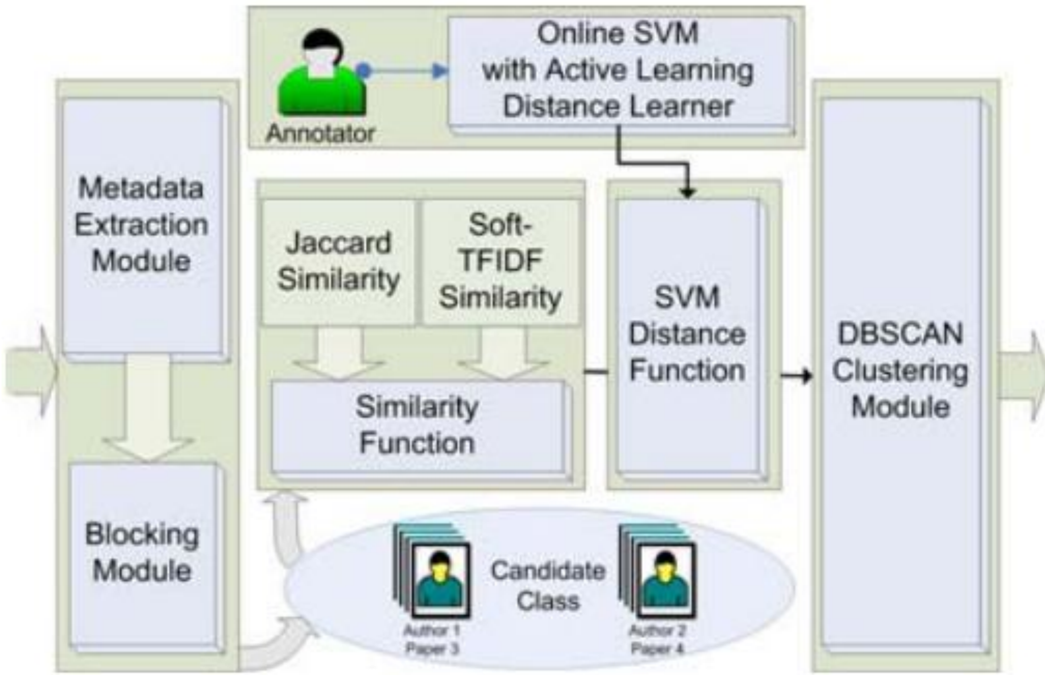
Author Grouping Methods

- Example
  - Jian Huang , Seyda Ertekin , C. Lee Giles. Efficient name disambiguation for large-scale databases, *PKDD*, 536—544, 2006.
  - LaSVM-DBSCAN
    - Uses an online SVM algorithm (LASVM) to build a supervised similarity function.
    - Uses the clustering algorithm DBSCAN to group references to the same author

Author name disambiguation methods

Author Grouping Methods

- LaSVM-DBSCAN



(Huang et al., 2006)

Author name disambiguation methods

Author Grouping Methods

- LaSVM-DBSCAN
  - Metadata Extraction Module
    - Extracts author metadata records from each paper.
  - Blocking Module
    - Blocks namesakes into ambiguous groups
  - Similarity function
    - Computes a similarity vector
      - $s^{(i,j)} = [sim_1(t^{(i)}_{u,1}, t^{(j)}_{v,1}), \ldots, sim_m(t^{(i)}_{u,m}, t^{(j)}_{v,m})]$
      - Edit distance → emails and URLs
      - Jaccard similarity → addresses and affiliations
      - Soft-TFIDF → name variations

Author name disambiguation methods

Author Grouping Methods

- LaSVM-DBSCAN
  - SVM
    - uses $s^{(i,j)}$ as a feature vector to classify whether $r^{(i)}_u$ and $r^{(j)}_v$ are references to the same author.
    - learns a distance pairwise function
  - DBSCAN
    - constructs clusters based on learned distance function

Author name disambiguation methods

Author assignment methods

- Directly assign each reference to a given author by constructing a model that represents the author using either a supervised classification technique or a model-based clustering technique.
  - Classification
  - Clustering

Author name disambiguation methods

Author assignment methods

- Classification
  - They receive as input a set of references to authors, called the *training data* (D), that consists of references for which the correct authorship is known.
  - Each example is composed of a set F of m features $\{f_1, f_2, \ldots, f_m\}$ along with a special variable called the *author*.
  - This *author* variable draws its value from a discrete set of labels $\{a_1, a_2, \ldots, a_n\}$, in which each label uniquely identifies an author.

Author name disambiguation methods

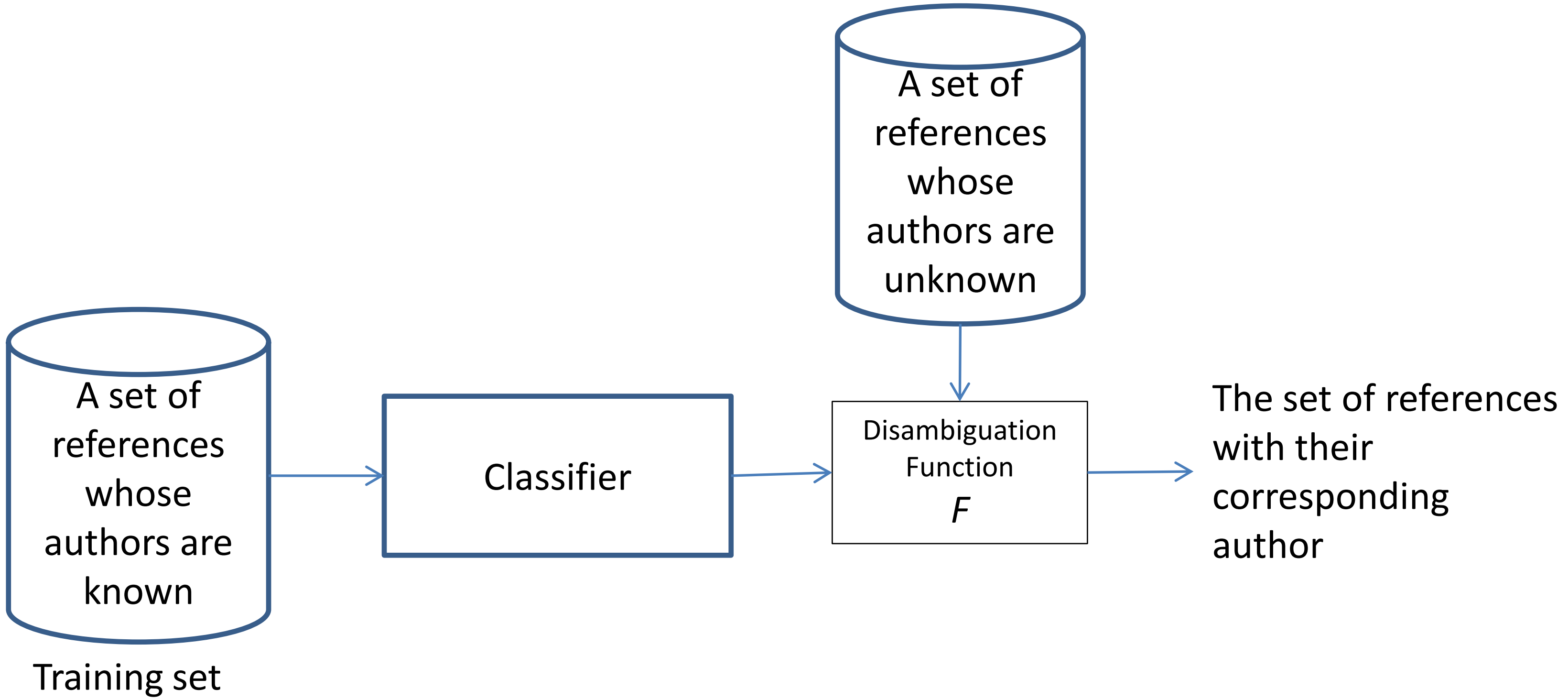
Author assignment methods

- Classification
  - The training examples are used to produce a disambiguation function that relates the features in the training examples to the correct author.
  - The *test set* (denoted as *T*)
    - A set of references for which the features are known while the correct author is unknown.
  - The disambiguator is used to predict the correct author for the references in *T*.
    - $F: \{f_1, f_2, \ldots, f_m\} \rightarrow \{a_1, a_2, \ldots, a_n\}$
  - The disambiguator essentially divides the records in *T* into n sets $\{a_1, a_2, \ldots, a_n\}$, where a_i contains (ideally all and no other) references in which the *i*th author is included.

Author name disambiguation methods

Author assignment methods

- Classification



Author name disambiguation methods

Author assignment methods

- Clustering
  - Work by optimizing the fit between a set of references to an author and some mathematical model used to represent that author.
  - Use probabilistic techniques to determine the author in a iterative way to fit the model (or estimate the parameters in probabilistic techniques) of the authors.

Author name disambiguation methods

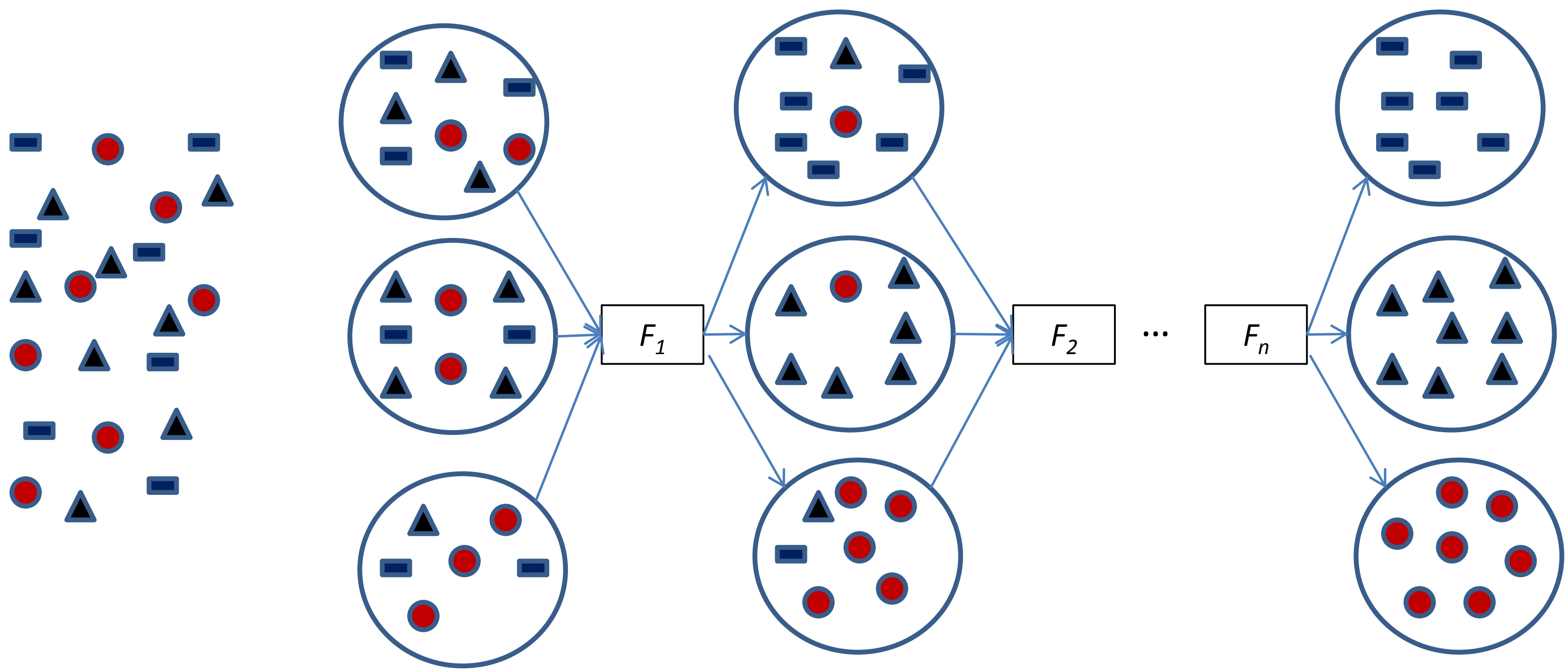
Author assignment methods

- Clustering
  - For instance,
    - In the first run, each reference may be randomly distributed to an author a_i and a function is derived using this distribution.
    - In the second iteration, this function is used to predict the author of each reference and a new function is derived to be used in the next iteration.
    - This process continues until a stop condition is reached, for instance, after a number of iterations.
  - These methods may be able to directly assign authors to their references in a new citations using the final derived function.

Author name disambiguation methods

Author assignment methods

- Clustering



Author name disambiguation methods

Author assignment methods

- Example
  - H. Han, L. Giles, H. Zha, C. Li, K. Tsioutsiouliklis. Two Supervised Learning Approaches for Name Disambiguation in Author Citations. *JCDL*, 296-305, 2004.
  - Naïve Bayes
  - Support vector machines - SVM

Author name disambiguation methods

Author assignment methods

- Example
  - The Naïve Bayes method
    - Assumes that each author's citation data are generated by a naive Bayes model.
    - Let X_i be an author class corresponding to a unique single person and let A be a reference.
    - A is attributed to a class that has the maximal posterior probability of producing it.

$$\max_i P(X_i|A) = \max_i P(A|X_i)\, P(X_i) / P(A)$$

Author name disambiguation methods

Author assignment methods

- Example
  - The Naïve Bayes method
    - *P(A)* is omitted because it does not depend on X_i

$$\max_i P(X_i|A) = \max_i P(A|X_i)P(X_i).$$

    - Assumes that attributes and distinct attribute elements are independent

$$P(A|X_i) = \prod_j P(T_j|X_i) = \prod_j \prod_k P(T_{jk}|X_i)$$

Author name disambiguation methods
Author assignment methods

- Example
  - The Naïve Bayes method
    - The conditional probabilities
      - $P(T_1|X_i)$ – an author publishes with coauthors
      - $P(T_2|X_i)$ – an author writes a work title
      - $P(T_3|X_i)$ – an author publishes in a venue
      - $P(T_4|X_i)$ – an author uses a name

Author name disambiguation methods

Author assignment methods

- Example
  - The SVM method
    - Uses SVMs to produce a model that predict the authors of the references in the test set
    - The model is produced using the training set
    - Han et al. (2004) associate each author name (individual person) with an author class.
    - Each reference is represented by a feature vector
      - Elements of their attributes (author and coauthor names, and words of work and publication venue titles)
      - TFIDF as the feature weight

Author name disambiguation methods

Explored evidence

- Citation information

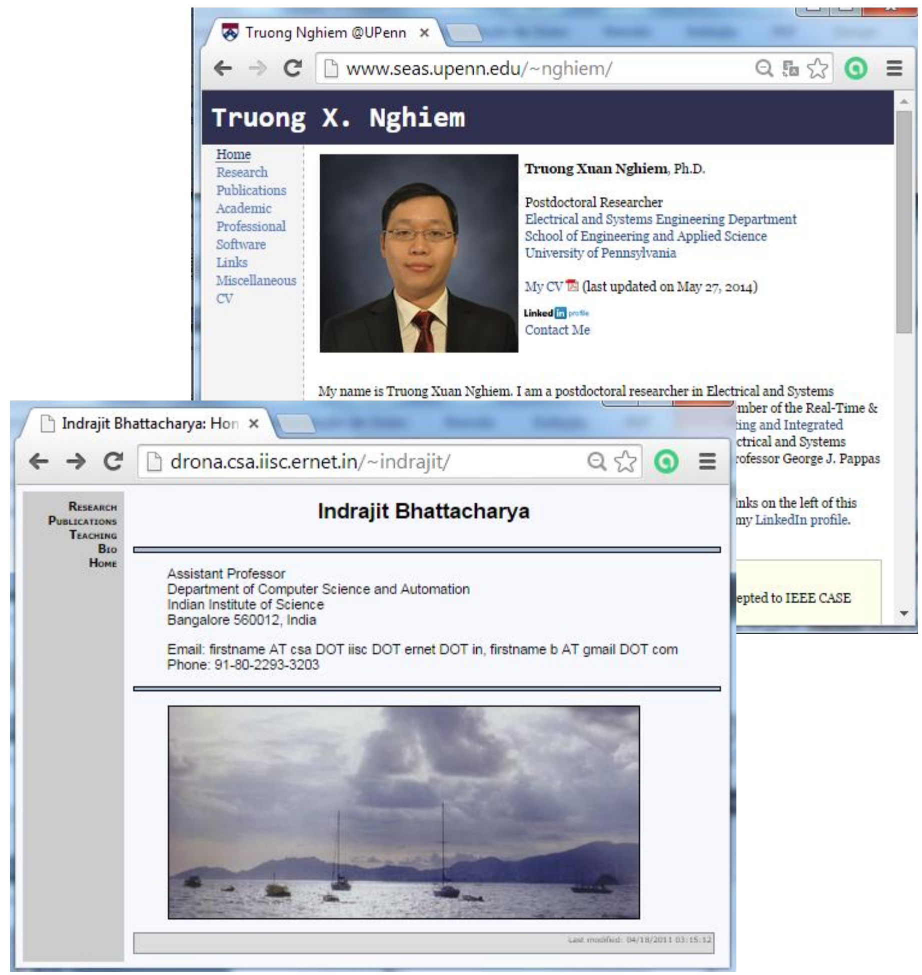
Citation Id	Citation
c_1	**(r_1) S. Godbole, (r_2) I. Bhattacharya, (r_3) A. Gupta, (r_4) A. Verma**. Building re-usable dictionary repositories for real-world text mining. CIKM, 2010.
c_2	**(r_5) Indrajit Bhattacharya, (r_6) Shantanu Godbole, (r_7) Ajay Gupta, (r_8) Ashish Verma, (r_9) Jeff Achtermann, (r_{10}) Kevin English**. Enabling analysts in managed services for CRM analytics. KDD, 2009.
c_3	**(r_{11}) T. Nghiem, (r_{12}) S. Sankaranarayanan, (r_{13}) G. E. Fainekos, (r_{14}) F. Ivancic, (r_{15}) A. Gupta, (r_{16}) G. J. Pappas**. Monte-carlo techniques for falsification of temporal properties of non-linear hybrid systems. HSCC, 2010.
c_4	**(r_{17}) William R. Harris, (r_{18}) Sriram Sankaranarayanan, (r_{19}) Franjo Ivancic, (r_{20}) Aarti Gupta**. Program analysis via satisfiability modulo path programs. POPL, 2010.

Author name disambiguation methods

Explored evidence

- Web information

r_1
r_2
r_3
r_4
r_5
…
r_n



Author name disambiguation methods

Implicit evidence

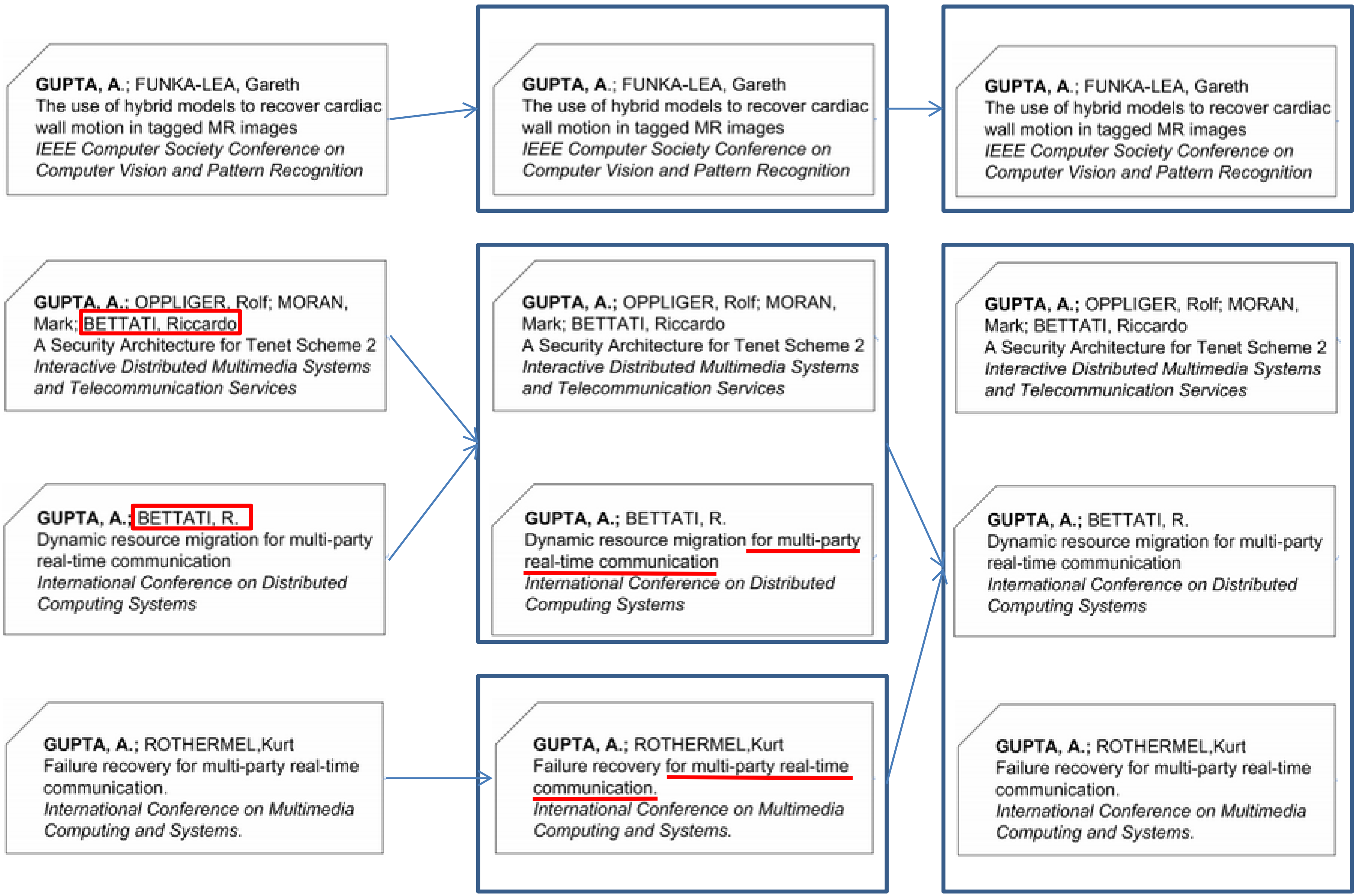
- Is inferred from visible elements of attributes.
- Several techniques have been implemented to find implicit evidence, such as the latent topics of a citation.
- One example is the Latent Direchlet Location (LDA) that estimates the topic distribution of a citation.
- This estimated distribution is used as new evidence (attribute) to calculate the similarity among references to authors.

HHC - Heuristic-based Hierarchical Clustering Method

HHC

- Deals at the same time with the homonym and synonym problems.
- Combines similarity functions with some heuristics:
  - Very rarely two authors with similar names that share a coauthor in common would be two different people in the real world.
  - The same authors publishes several works about the same subject.
- Attempts to resolve the name ambiguity problem in two main steps.

XXIX SBBD Simpósio Brasileiro de Banco de Dados



Algorithm 1. HHC.

Input: List R of citation records;
Output: List C of clusters of authorship records;

Let A be a list of authorship records;
Let C_1 and C_2 be lists of clusters;
Let G be a list of ambiguous groups;
Let R' be a list of citation records;
$R' \leftarrow \text{PreprocessCitationRecords}(R)$;
$A \leftarrow \text{CreateAuthorshipRecords}(R')$;
$G \leftarrow \text{CreateAmbiguousGroups}(A)$;
$C \leftarrow \emptyset$
for each ambiguous group g **in** G **do**
    $C_1 \leftarrow \text{FirstStep}(g)$;
    $C_2 \leftarrow \text{SecondStep}(C_1)$;
    $C \leftarrow C \cup C_2$;
end for

XXIX SBBD Simpósio Brasileiro de Banco de Dados

Algorithm 2. FirstStep.

Input: Ambiguous group g;
Output: List C of clusters of authorship records;

Let L and S be lists of authorship records;
Let C, C_1 and C_2 be lists of clusters;
$S \leftarrow \text{GetShortNameRecords}(g)$;
$L \leftarrow \text{GetLongNameRecords}(g)$;
$C_1 \leftarrow \emptyset$;
$C_2 \leftarrow \text{ProcessList}(L, C_1)$;
$C \leftarrow \text{ProcessList}(S, C_2)$;

Algorithm 4. SecondStep.

Input: List C_i of clusters of authorship records;
Output: List C_o of clusters of authorship records;

$C_o \leftarrow C_i$;
fused $\leftarrow$ **true**;
while *fused* **do**
    fused $\leftarrow$ **false**;
    for each c_1 **in** C_o **do**
        for each c_2 **in** C_o **do**
            if $c_1 \neq c_2$ **and the first author name from** c_1 **is similar to the first author name from** c_2 **then**
                $t_{t1} \leftarrow$ GetWorkTitleTerms(c_1);
                $t_{t2} \leftarrow$ GetWorkTitleTerms(c_2);
                $t_{v1} \leftarrow$ GetPublicationVenueTitleTerms(c_1);
                $t_{v2} \leftarrow$ GetPublicationVenueTitleTerms(c_2);
                if TitleSimilarity(t_{t1}, t_{t2}) $>$ *title-threshold* **or** VenueSimilarity(t_{v1}, t_{v2}) $>$ *venue-*threshold **then**
                    $c_1 \leftarrow$ Fuse(c_1, c_2);
                    remove(C_o, c_2);
                    fused $\leftarrow$ **true**;

HHC

- Similarity functions
  - For author and coauthor names
    - Fragment comparison
  - Work title and publication venue title
    - Cosine similarity function

HHC

Comparative evaluation

- Collections

BDBComp

Biblioteca Digital Brasileira de Computação

  - 363 citation records (1987 – 2007).
  - 184 distinct authors.

dblp

computer science bibliography

  - 4,287 records
  - 220 distinct authors

HHC
Comparative Evaluation

Evaluation Metrics

- ACP – *Average cluster purity*

$$\text{ACP} = \frac{1}{N}\sum_{i=1}^{e}\sum_{j=1}^{t}\frac{n_{ij}^2}{n_i}$$

- AAP – *Average author purity*

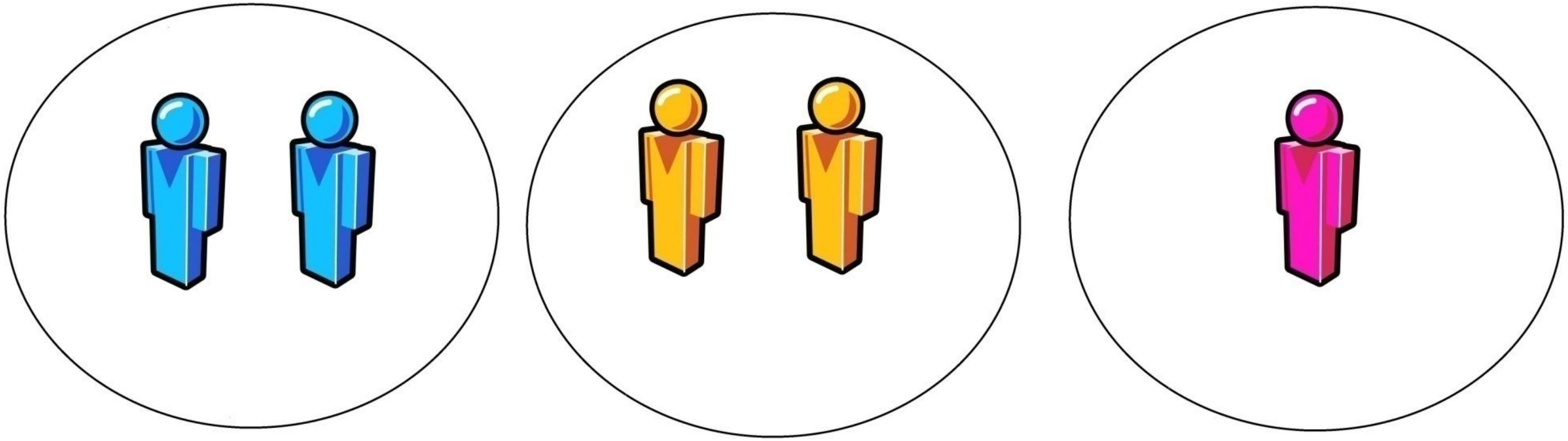
$$\text{AAP} = \frac{1}{N}\sum_{j=1}^{t}\sum_{i=1}^{e}\frac{n_{ij}^2}{n_j}$$

- K

$$K = \sqrt{\text{ACP} \times \text{AAP}}$$

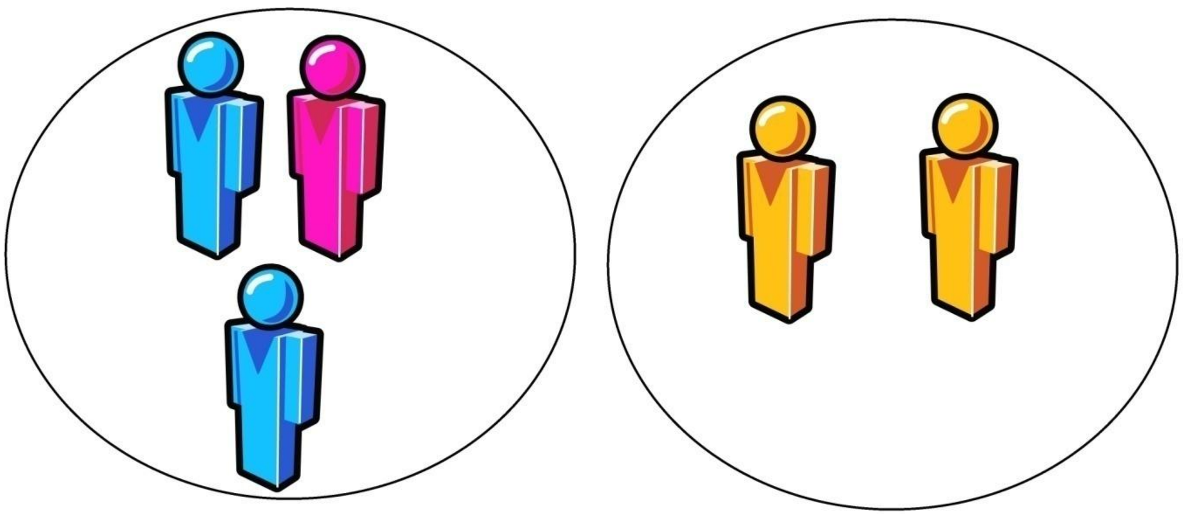
N = total number of references to authors.
t = number of theoretical clusters.
e = number of empirical clusters.
n_i = total number of references in the empirical cluster *i*.
n_{ij} = total number of references in the empirical cluster *i* that are also in the theoretical cluster *j*.

Experimental Evaluation



ACP = 1   clusters are pure.

Experimental Evaluation



AAP = 1 clusters are not fragmented.

HHC
Comparative Evaluation

	# of pairwise records in the generated clusters	# of pairwise records not in the generated clusters
# of pairwise records of same authors	a	b
# of pairwise records of different authors	c	d

Evaluation Metrics

- pP – *Pairwise precision*

$$pP = \frac{a}{a+c}$$

- pR – *Pairwise recall*

$$pR = \frac{a}{a+b}$$

- pF1

$$pF1 = \frac{2 \cdot pP \cdot pR}{pP + pR}$$

HHC
Comparative Evaluation

- Baselines
  - Supervised methods
    - SVM
    - Naïve Bayes
  - Unsupervised methods
    - K-way spectral clustering
    - SVM-DBSCAN

HHC

Comparative evaluation

- DBLP

Method	ACP	AAP	K	pP	pR	pF1
HHC	**0.86 ± 0.010**	0.68 ± 0.011	0.77 ± 0.008	**0.84 ± 0.014**	0.65 ± 0.017	**0.73 ± 0.013**
SVM	0.75 ± 0.010	**0.85 ± 0.006**	**0.80 ± 0.008**	0.61 ± 0.012	**0.91 ± 0.007**	**0.72 ± 0.010**
NaiveBayes	0.67 ± 0.011	0.80 ± 0.009	0.73 ± 0.009	0.53 ± 0.011	0.85 ± 0.009	0.64 ± 0.010
K-way	0.75 ± 0.011	0.47 ± 0.009	0.59 ± 0.009	0.66 ± 0.017	0.30 ± 0.008	0.40 ± 0.010
SVM-DBSCAN	0.24 ± 0.039	**0.83 ± 0.082**	0.43 ± 0.013	0.17 ± 0.007	0.78 ± 0.092	0.27 ± 0.010

HHC

Comparative evaluation

- BDBComp

Method	ACP	AAP	K	pP	pR	$pF1$
HHC	**0.88 ± 0.021**	**0.99 ± 0.010**	**0.93 ± 0.015**	**0.58 ± 0.085**	**0.83 ± 0.119**	**0.65 ± 0.089**
SVM	0.26 ± 0.028	0.95 ± 0.018	0.48 ± 0.024	0.10 ± 0.025	**0.70 ± 0.136**	0.16 ± 0.032
Naive Bayes	0.20 ± 0.008	**0.97 ± 0.020**	0.42 ± 0.009	0.10 ± 0.016	**0.80 ± 0.131**	0.16 ± 0.019
K-way	**0.89 ± 0.017**	0.97 ± 0.016	**0.93 ± 0.015**	**0.67 ± 0.122**	**0.79 ± 0.140**	**0.71 ± 0.129**
SVM-DBSCAN	0.36 ± 0.117	0.80 ± 0.066	0.48 ± 0.069	0.04 ± 0.023	0.31 ± 0.215	0.05 ± 0.028

HHC

- Discussion
  - HHC uses specific heuristics to solve the author name ambiguity problem.
  - HHC deals with both the synonym and homonym citation problems.
  - HHC does not need any training examples.
  - HHC does not make use of any privileged information such as the number of correct groups to be generated.

SAND: Self-training Author Name Disambiguator

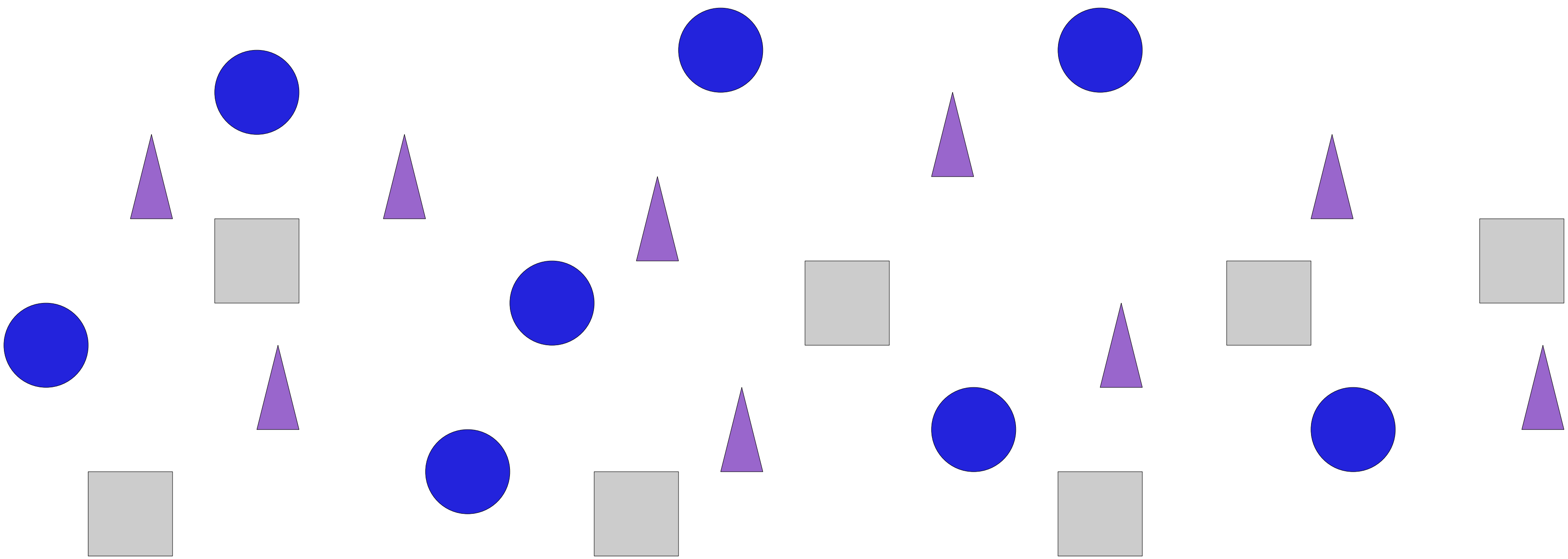
SAND: Self-training Author Name Disambiguator

- SAND exploits the strengths of both author grouping and author assignment methods.
- SAND works in three steps.
  - Author grouping – recurring patterns in the coauthorship graph are exploited in order to produce very pure clusters of references.
  - Cluster selection – a subset of the clusters produced in the previous step is selected as training data for the next step.
  - Author assignment, a learned function is derived to disambiguate the references in the clusters that were not selected in the previous step.

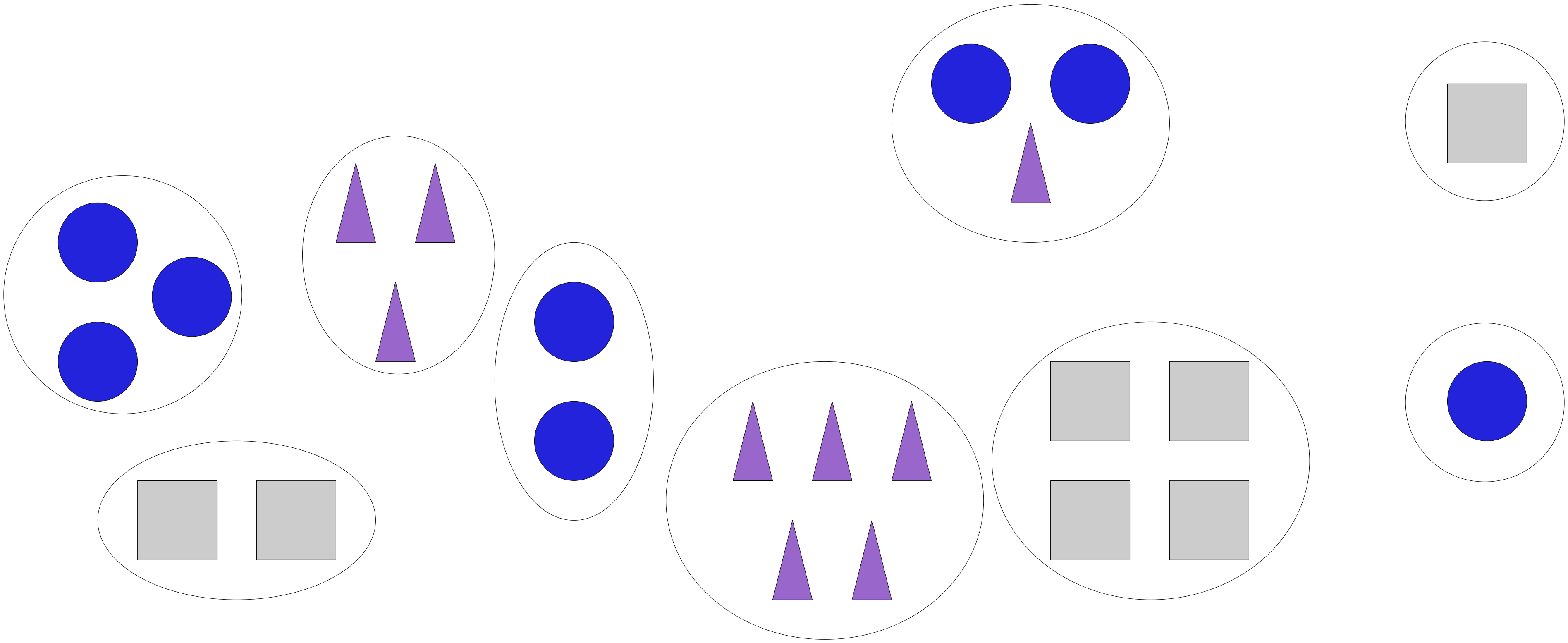
SAND Design
The Author Grouping Step

- The goal of this step is to automatically create pure clusters of references.
- The approach we adopt is to organize references within each ambiguous group into individual clusters.
- The key intuition is that some of these clusters can be associated with a unique author label.
- Pure clusters are extracted by exploiting highly discriminative attributes, so that references associated with different authors are unlikely to be grouped together into the same cluster.

SAND Design
The Author Grouping Step



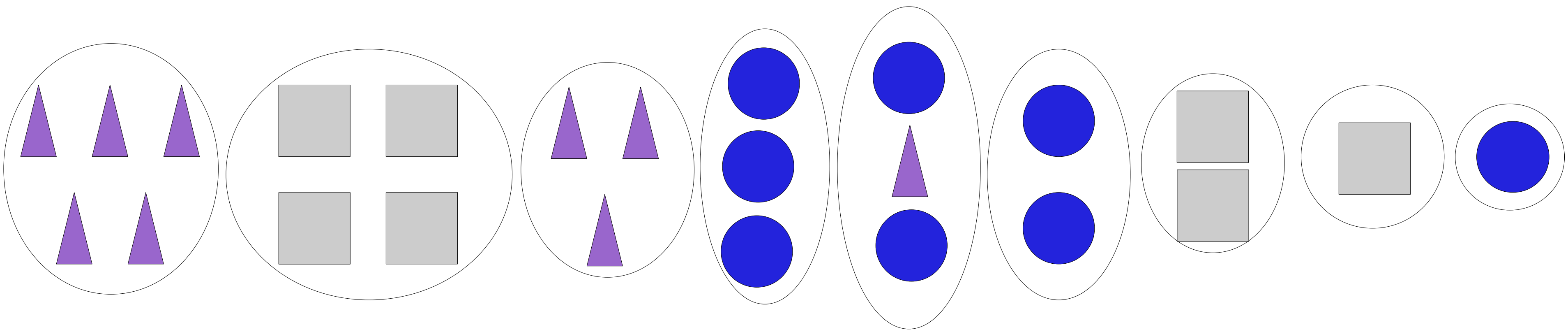
SAND Design
The Author Grouping Step



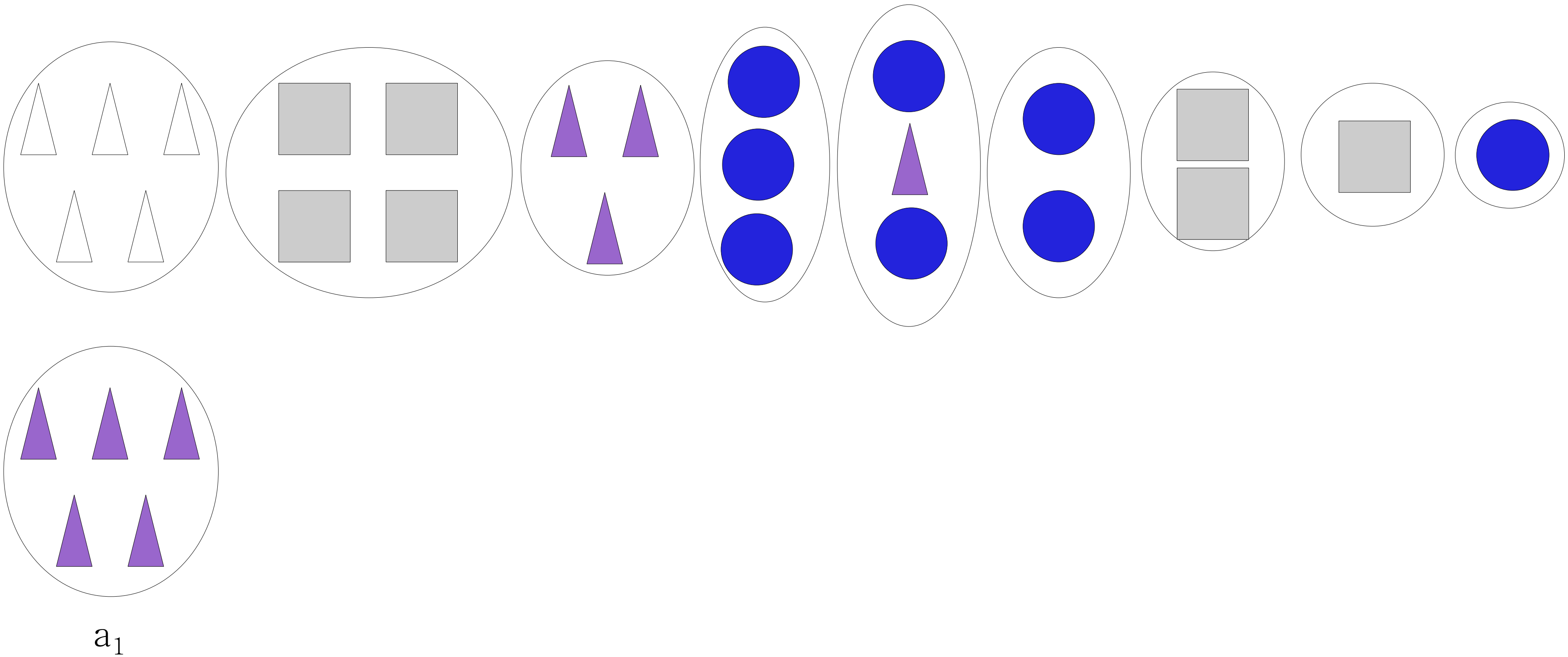
SAND Design
The Cluster Selection Step

- Aims to generate the initial training examples
- We associate the clusters in the training data to different authors
- Thus, we must select only the clusters belonging to different real authors to compose the training data
- We select the most dissimilar clusters to compose the training data

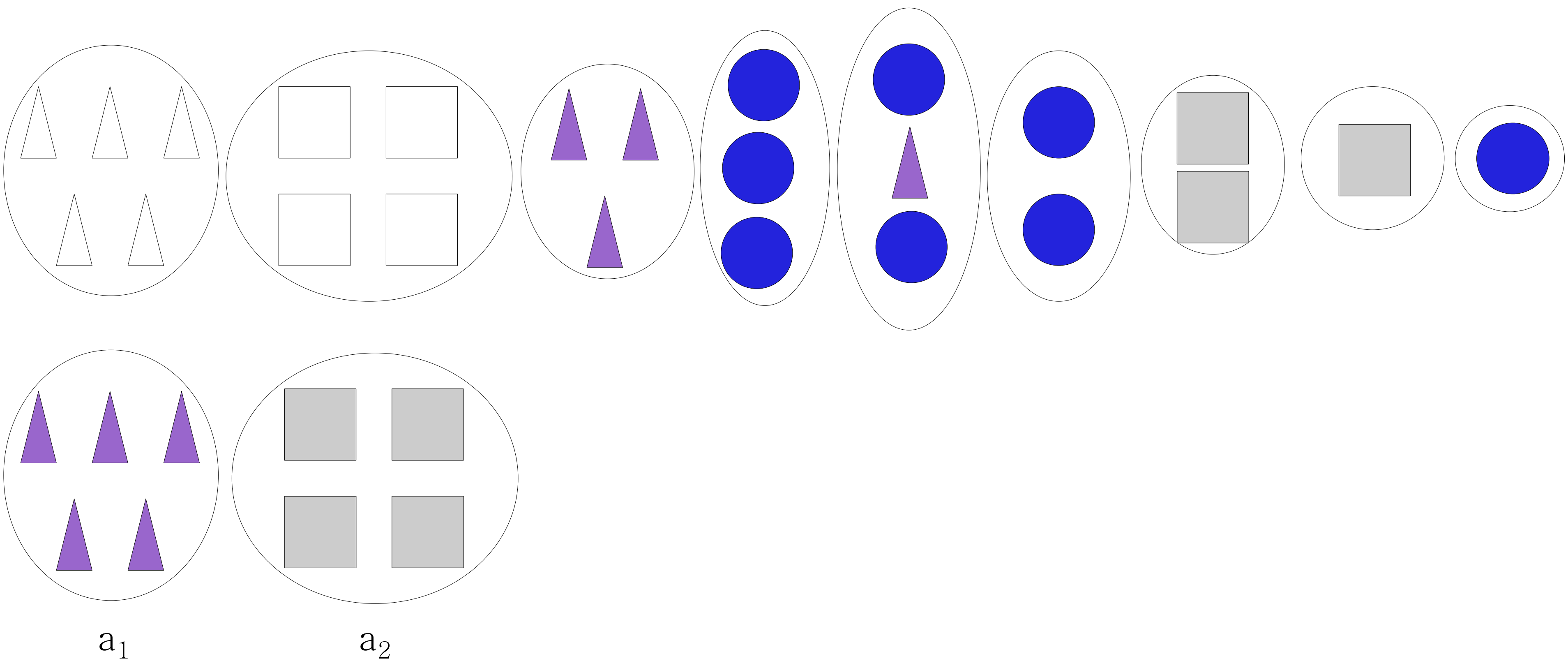
SAND Design
The Cluster Selection Step



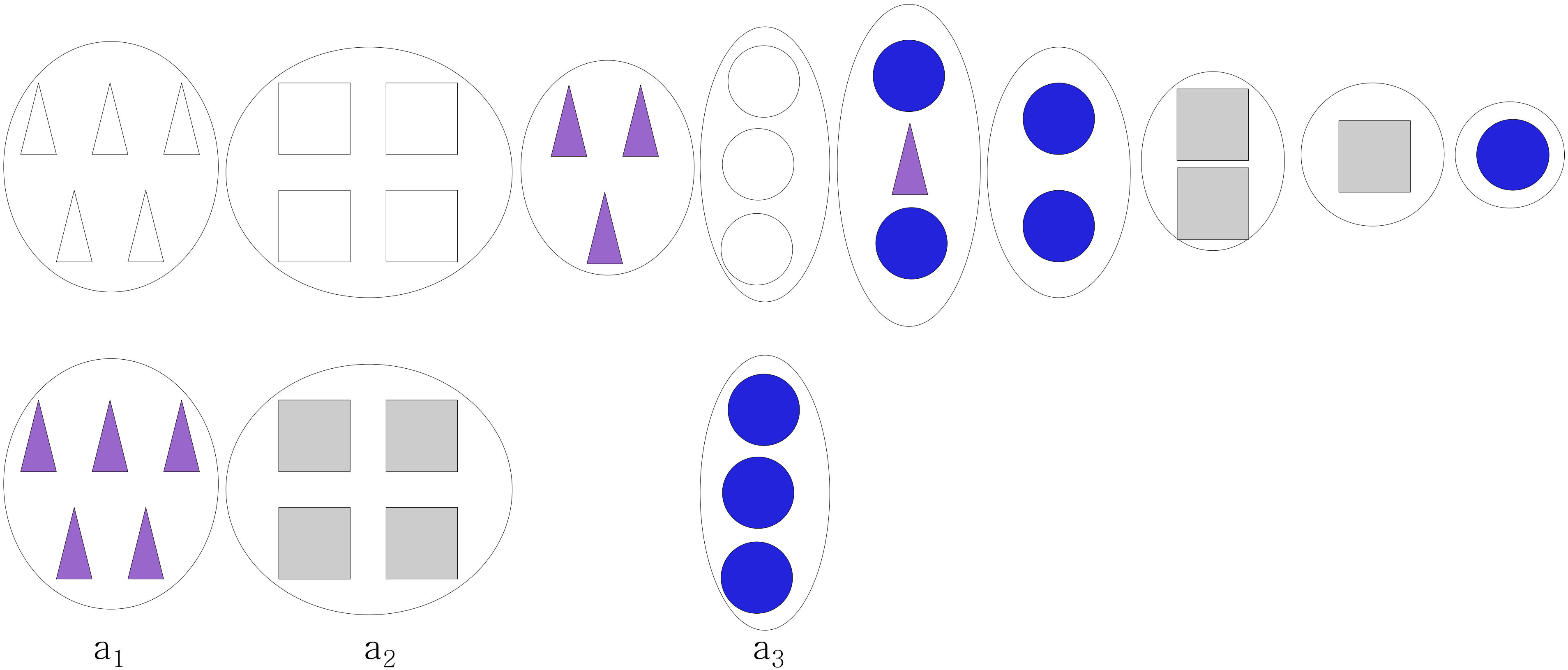
SAND Design
The Cluster Selection Step



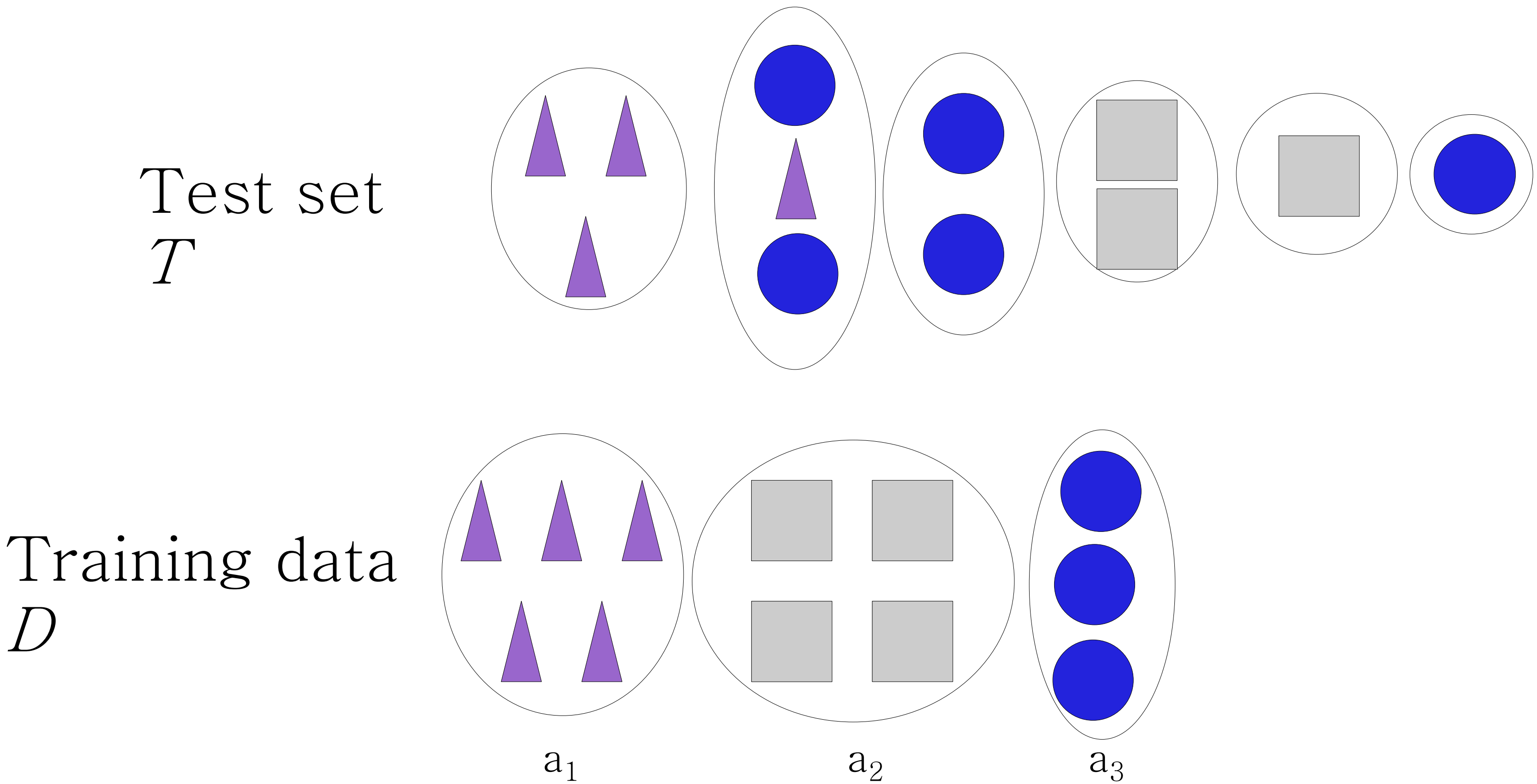
SAND Design
The Cluster Selection Step



SAND Design
The Cluster Selection Step



SAND Design
The Cluster Selection Step



SAND Design

The Cluster Selection Step

- We evaluate three strategies to measure the similarity/dissimilarity among clusters:
  - Strategy 1. We compare two clusters c_i and c_j using the attributes of the references in these clusters.
  - Strategy 2. We compare two clusters c_i and c_j using only the author name assigned to them.
  - Strategy 3. This strategy combines both previous strategies.

SAND Design

The Author Assignment Step

- The set of examples, *D*, is used to produce a disambiguation function from $\{f_1, f_2, . . . , f_m\}$ to $\{a_1, a_2, . . . , a_n\}$ that is used to predict the correct author of the references in the test set *T* .
- It is based on a lazy associative classifier to produce disambiguation functions from *D*.

SAND Design
The Author Assignment Step

- Associative Name Disambiguation
  - The proposed technique exploits the fact that:
    - There are strong associations between features $\{f_1, f_2, \ldots, f_m\}$ and specific authors $\{a_1, a_2, \ldots, a_n\}$.
  - The proposed technique uncovers such associations from *D*, and then produces a disambiguation function $\{f_1, f_2, \ldots, f_m\} \rightarrow \{a_1, a_2, \ldots, a_n\}$.
  - Demand-Driven Rule Extraction
    - It projects/filters the training data according to the features in reference $x \in T$
    - It extracts rules from this projected training data

SAND Design
The Author Assignment Step

- Predicting the Author of the each Reference

$$s(a_i, x) = \frac{\sum_{j=1}^{|\mathcal{R}_{a_i}^{x}|} \theta(r_j)}{|\mathcal{R}_{a_i}^{x}|}$$

$$\hat{p}(a_i|x) = \frac{s(a_i, x)}{\sum_{j=1}^{n} s(a_j, x)}$$

SAND Design

The Author Assignment Step

- Exploiting Reliable Predictions
  - Additional examples may be obtained from the predictions performed using the disambiguation function.
  - Given an arbitrary reference $x \in T$, and the two most likely authors for x, a_i and a_j, we denote as $\Delta(x)$ the reliability of predicting a_i.

$$\Delta(x) = \frac{\widehat{p}(a_i \mid x)}{\widehat{p}(a_j \mid x)}$$

  - The idea is to only predict a_i if $\Delta(x) \geq \Delta_{min}$.
- Temporary Abstention – it abstains from such doubtful predictions.

SAND Design
The Author Assignment Step

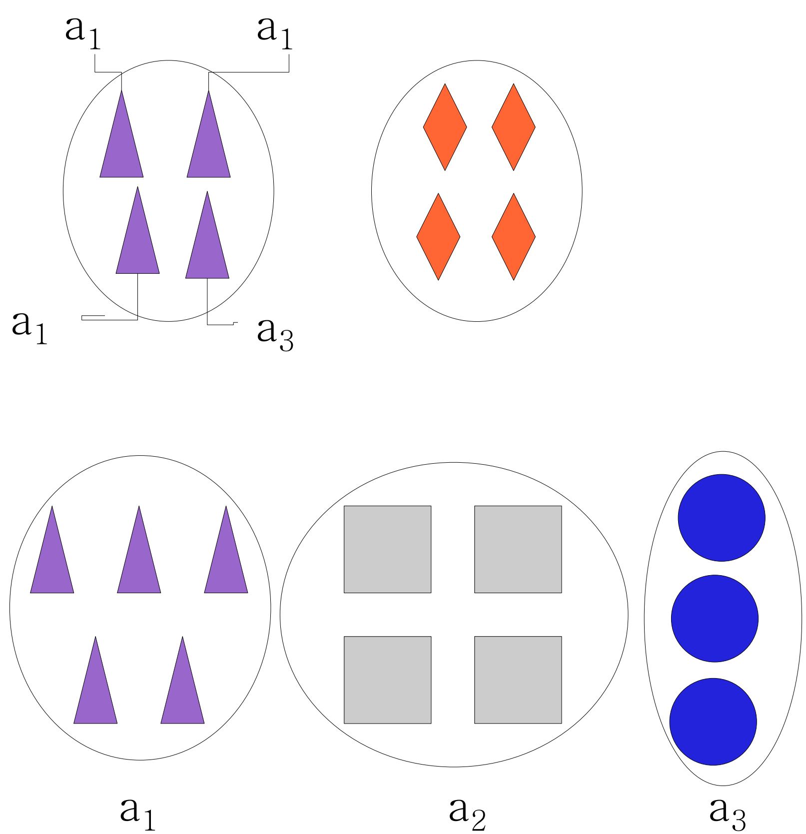
- We propose to use the lack of rules supporting any already seen author as evidence indicating the appearance of an unseen author.
- The number of rules that is necessary to consider an author as an already seen one is controlled by a parameter, γ_{min}.
- For an reference $x \in T$, if the number of rules extracted from D^x ($\gamma(x)$) is smaller than γ_{min}, then the author of *x* is considered as a new/unseen author and a new label a_k is created to identify such author.
  - This prediction is considered as a new example and included into *D*.
- An appropriate value for γ_{min} can be obtained by performing cross-validation in *D*.

The Author Assignment Step

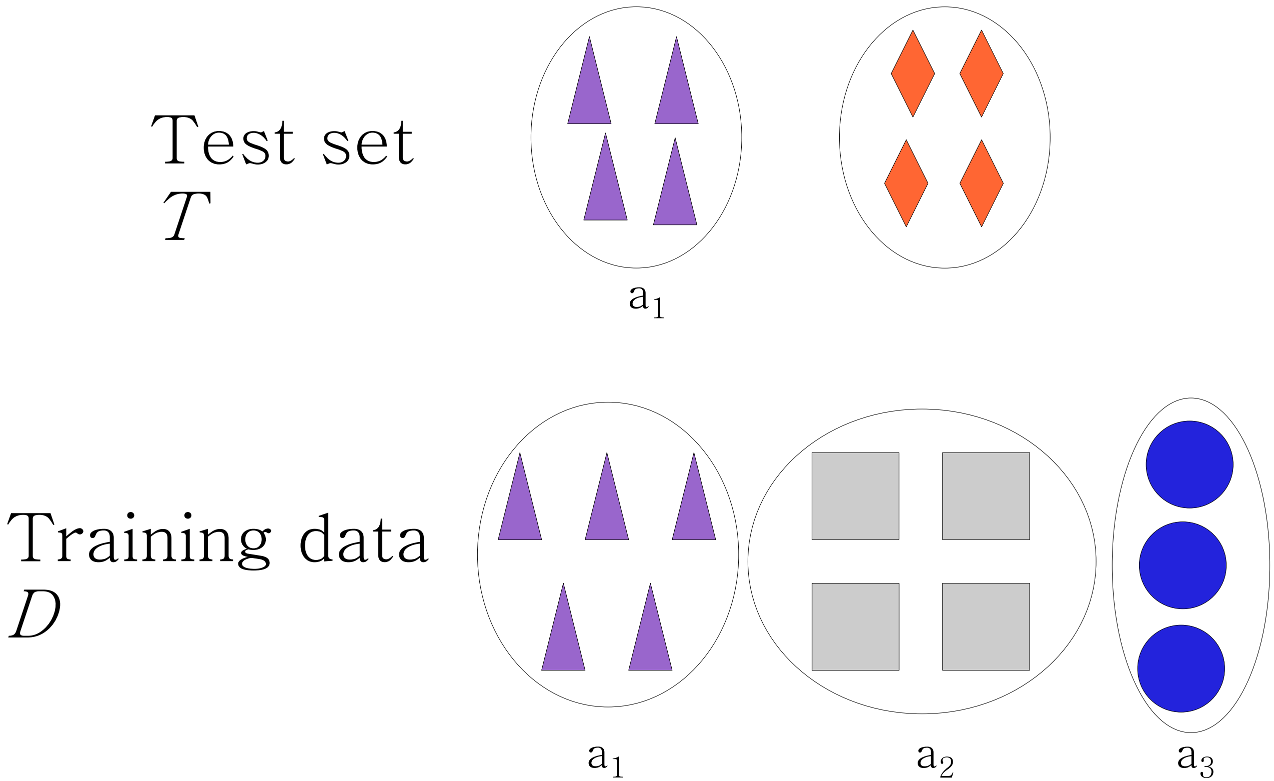
Predicting the Author of Each Cluster

Test set T

Training data D



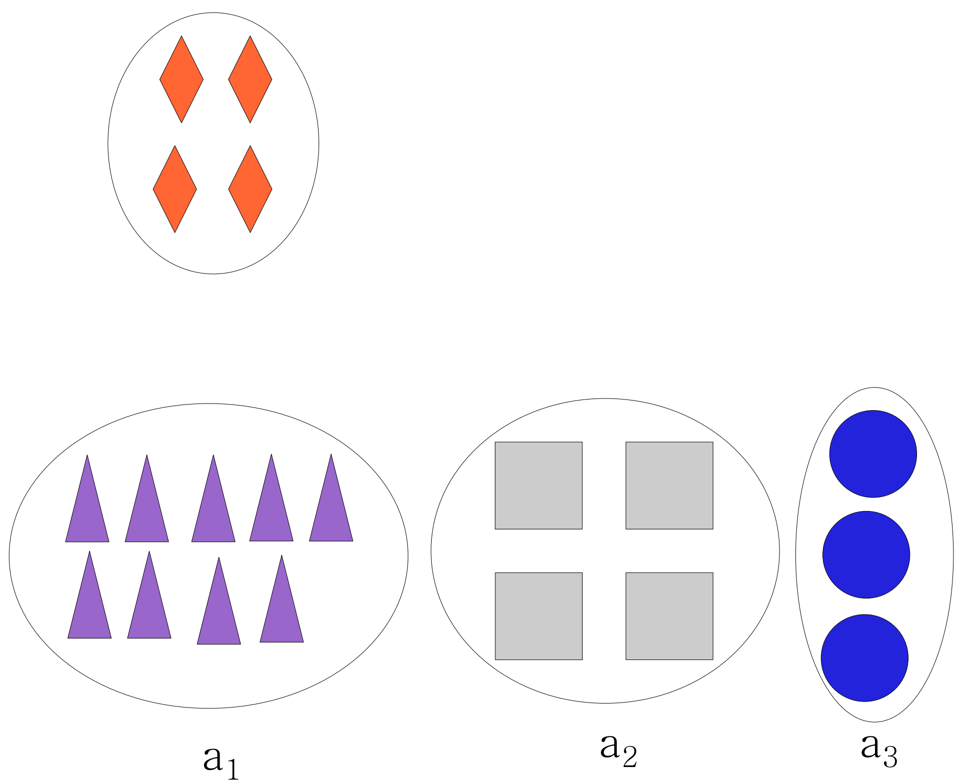
The Author Assignment Step
Predicting the Author of Each Cluster



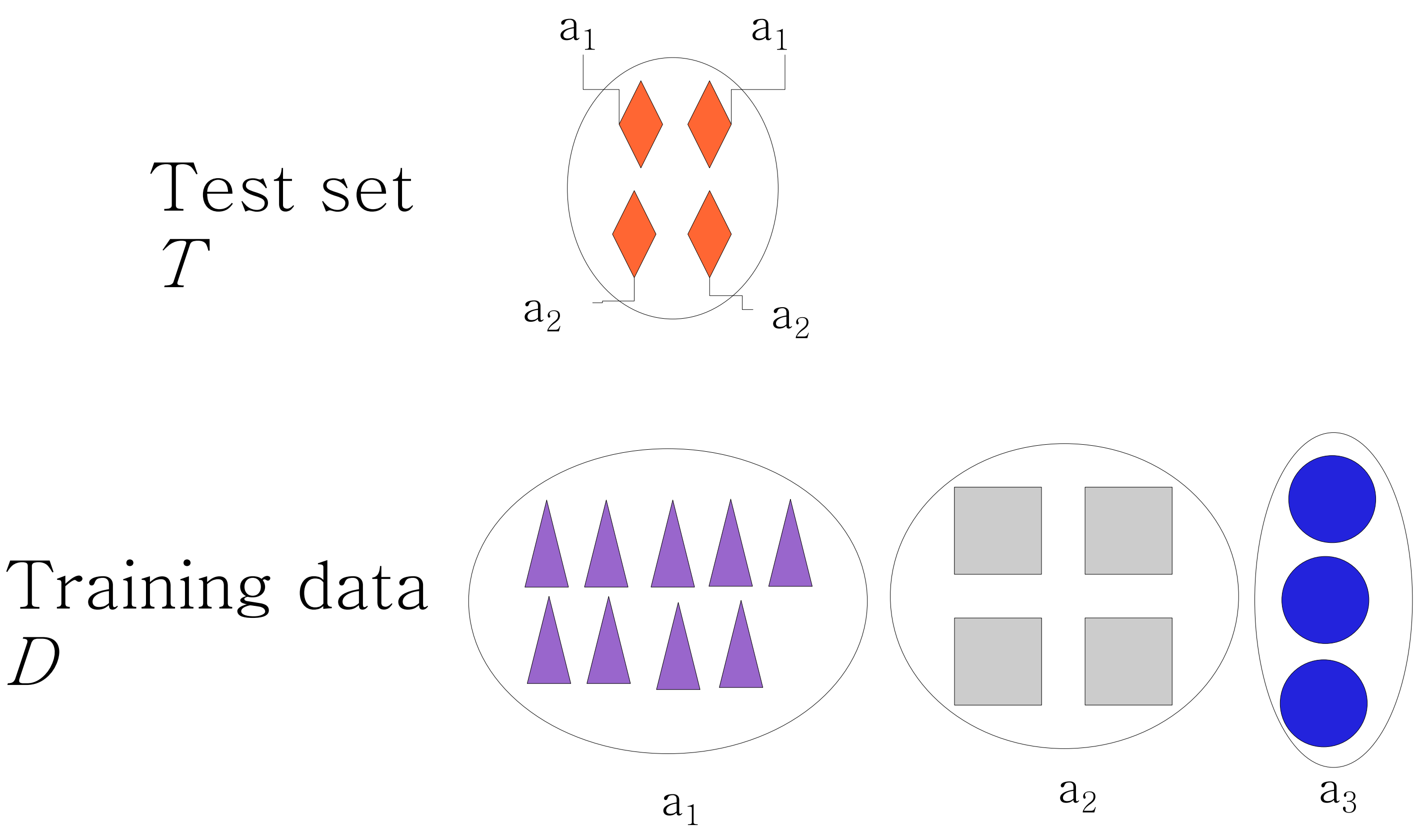
The Author Assignment Step
Predicting the Author of Each Cluster

Test set
T

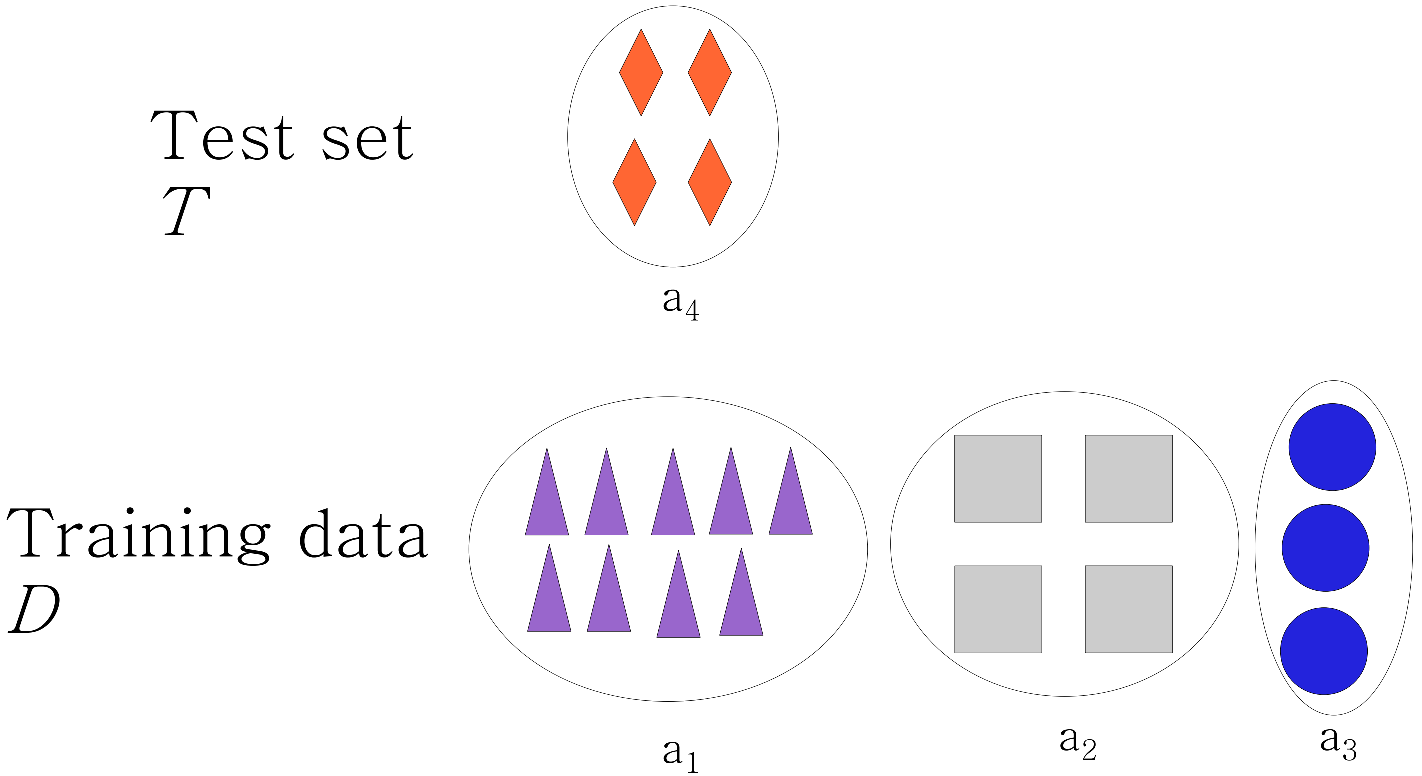
Training data
D



The Author Assignment Step
Predicting the Author of Each Cluster



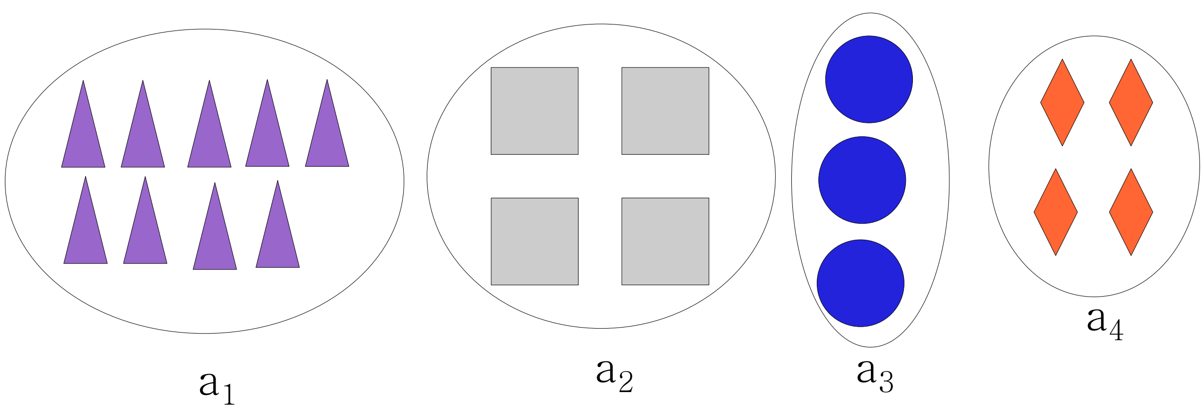
The Author Assignment Step
Predicting the Author of Each Cluster



The Author Assignment Step
Predicting the Author of Each Cluster

Test set
T

Training data
D



Experimental evaluation

- Collections
  - DBLP, BDBComp and synthetic data produced with SyGAR.
- Evaluation metrics
  - The K and pairwise F1 metrics.
- We compare the effectiveness of SAND against six baselines:
  - SVM
  - NB
  - SLAND
  - KWAY
  - LASVM-DBSCAN
  - HHC

Experimental Evaluation
Evaluating the Author Grouping Step

Table : Results obtained by the author grouping step in the DBLP collection, without using the popular last names.

Ambiguous Group	ACP	AAP	K	pP	pR	pF1
A Gupta	0.990 ± 0.002	0.416 ± 0.033	0.641 ± 0.025	0.994 ± 0.001	0.398 ± 0.056	0.567 ± 0.058
A Kumar	0.995 ± 0.003	0.242 ± 0.011	0.490 ± 0.011	0.995 ± 0.003	0.098 ± 0.006	0.178 ± 0.010
C Chen	0.953 ± 0.003	0.202 ± 0.003	0.439 ± 0.003	0.906 ± 0.008	0.050 ± 0.001	0.095 ± 0.002
D Johnson	1.000 ± 0.000	0.301 ± 0.008	0.548 ± 0.008	1.000 ± 0.000	0.295 ± 0.016	0.455 ± 0.019
J Martin	0.987 ± 0.007	0.500 ± 0.007	0.702 ± 0.007	0.957 ± 0.023	0.322 ± 0.005	0.482 ± 0.008
J Robinson	1.000 ± 0.000	0.355 ± 0.007	0.596 ± 0.005	1.000 ± 0.000	0.285 ± 0.010	0.443 ± 0.011
J Smith	0.971 ± 0.007	0.263 ± 0.031	0.504 ± 0.032	0.982 ± 0.018	0.279 ± 0.054	0.432 ± 0.067
K Tanaka	1.000 ± 0.000	0.380 ± 0.008	0.616 ± 0.006	1.000 ± 0.000	0.231 ± 0.008	0.375 ± 0.011
M Brown	1.000 ± 0.000	0.395 ± 0.007	0.629 ± 0.006	1.000 ± 0.000	0.340 ± 0.013	0.507 ± 0.015
M Jones	1.000 ± 0.000	0.281 ± 0.015	0.530 ± 0.014	1.000 ± 0.000	0.251 ± 0.021	0.400 ± 0.026
M Miller	0.991 ± 0.005	0.603 ± 0.026	0.773 ± 0.017	0.988 ± 0.009	0.586 ± 0.034	0.735 ± 0.026

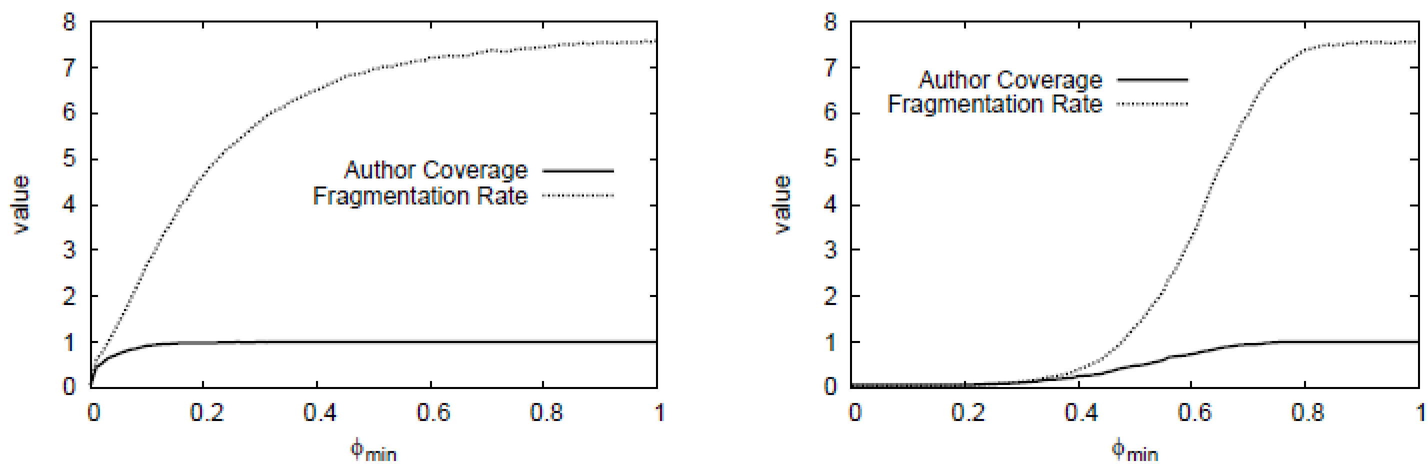
Experimental Evaluation
Evaluating the Author Grouping Step

Table : Results obtained by the author grouping stepin the DBLP andcollections, using the popular last names.

Ambiguous Group	ACP	AAP	K	pP	pR	pF1
A Gupta	0.990 ± 0.002	0.429 ± 0.030	0.651 ± 0.023	0.994 ± 0.001	0.427 ± 0.051	0.596 ± 0.053
A Kumar	1.000 ± 0.000	0.241 ± 0.013	0.491 ± 0.013	1.000 ± 0.000	0.097 ± 0.007	0.176 ± 0.011
C Chen	0.950 ± 0.004	0.260 ± 0.004	0.497 ± 0.005	0.843 ± 0.031	0.087 ± 0.003	0.158 ± 0.005
D Johnson	1.000 ± 0.000	0.274 ± 0.033	0.523 ± 0.032	1.000 ± 0.000	0.253 ± 0.059	0.401 ± 0.078
J Martin	1.000 ± 0.000	0.508 ± 0.004	0.713 ± 0.003	1.000 ± 0.000	0.320 ± 0.002	0.485 ± 0.002
J Robinson	1.000 ± 0.000	0.347 ± 0.016	0.589 ± 0.014	1.000 ± 0.000	0.279 ± 0.020	0.435 ± 0.025
J Smith	0.987 ± 0.004	0.200 ± 0.030	0.443 ± 0.033	0.993 ± 0.005	0.186 ± 0.042	0.312 ± 0.059
K Tanaka	1.000 ± 0.000	0.378 ± 0.017	0.615 ± 0.014	1.000 ± 0.000	0.231 ± 0.013	0.374 ± 0.017
M Brown	1.000 ± 0.000	0.368 ± 0.000	0.607 ± 0.000	1.000 ± 0.000	0.301 ± 0.000	0.463 ± 0.000
M Jones	1.000 ± 0.000	0.266 ± 0.017	0.516 ± 0.017	1.000 ± 0.000	0.238 ± 0.023	0.383 ± 0.031
M Miller	0.993 ± 0.004	0.589 ± 0.015	0.765 ± 0.010	0.989 ± 0.008	0.575 ± 0.022	0.727 ± 0.019

Experimental Evaluation

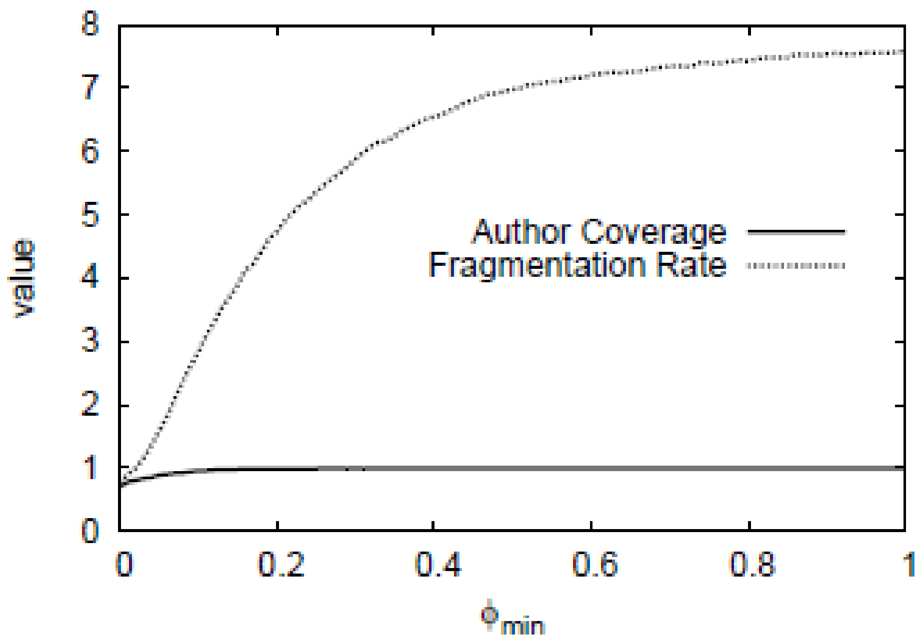
Evaluating the Clustering Selection Step – DBLP



(a) Centroid – cosine (b) Centroid – euclidian distance

Experimental Evaluation

Evaluating the Clustering Selection Step using Dissimilar Author Names

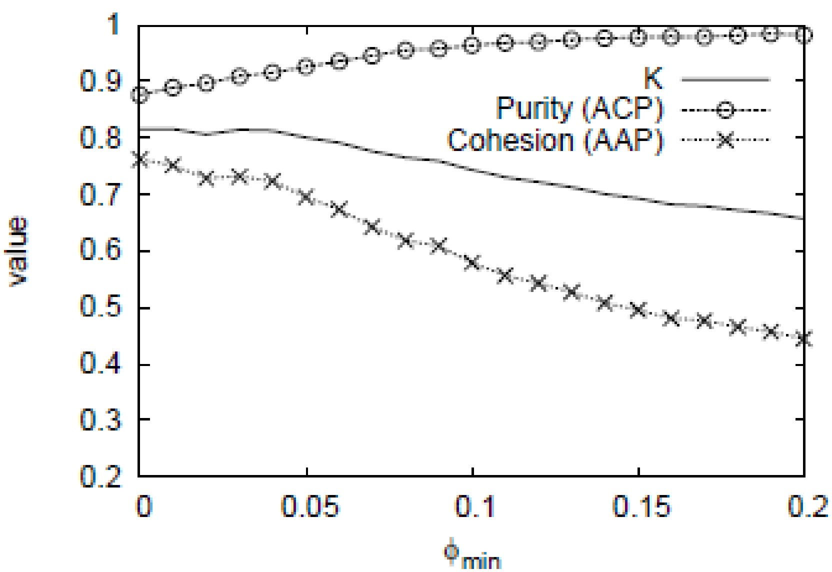


DBLP

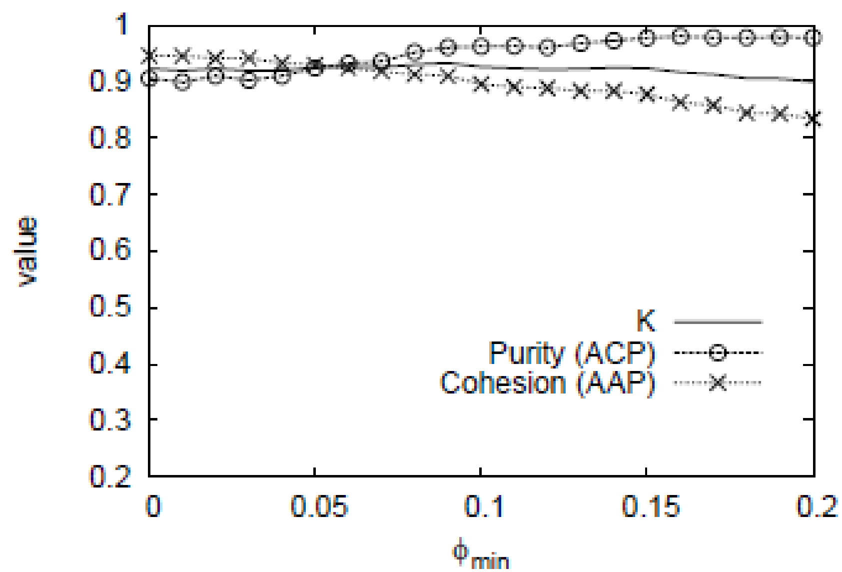
Experimental Evaluation

Evaluating SAND in the DBLP and BDBComp collections

DBLP



BDBComp



Experimental Evaluation

Comparison with the Author Grouping Baselines

	DBLP		BDBComp	
Method	K	pF1	K	pF1
SAND	**0.815**	**0.796**	**0.924**	**0.752**
HHC	0.773	0.751	**0.913**	**0.756**
KWAY	0.560	0.402	0.805	0.436
LASVM-DBSCAN	0.551	0.406	0.757	0.211

Experimental Evaluation

Comparison with the Supervised Author Assignment Methods

	DBLP		BDBComp	
Method	K	pF1	K	pF1
SAND	0.775±0.010	0.720±0.018	**0.940**±0.014	**0.462**±0.040
SLAND	**0.877**±0.007	**0.867**±0.008	0.900±0.016	**0.456**±0.028
SVM	0.799±0.008	0.721±0.010	0.481±0.024	0.160±0.032
NB	0.736±0.009	0.647±0.012	0.420±0.009	0.160±0.019

Experimental Evaluation

Comparison with Other Supervised Methods for the Author Assignment Step

	DBLP		BDBComp	
Method	K	pF1	K	pF1
SAND	**0.815**±0.010	**0.796**±0.020	**0.924**±0.004	**0.752**±0.015
S-SVM	0.666±0.009	0.489±0.018	**0.917**±0.006	0.412±0.020
S-NB	0.640±0.014	0.466±0.026	0.883±0.013	0.286±0.037

SAND

- Discussion
  - SAND is particularly suitable to operate in scenarios with scarce information
  - SAND outperformed unsupervised methods by more than 27% in the K metric and more than 36% under the pF1 metric.
  - SAND also demonstrated to be very competitive, sometimes even superior, to several supervised author assignment methods, with K values up to 0.94.

SAND

- Future work
  - Find out situations in which only the first step is sufficient to disambiguate an ambiguous group
  - Generalize SAND to disambiguate other applications, e.g., ambiguous place names;
  - Investigate other manners to identify when a reference belongs to an author who does not have any citation record in the digital library
  - Exploit situations in which labeling a small amount of informative instances may be useful using techniques such as active learning and user relevance feedback in doubtful cases.

INDi - Incremental Unsupervised Name Disambiguation

INDi - Incremental Unsupervised Name Disambiguation

- Identifies the correct authors of the new citation records to be inserted in a digital library.
  - Identifies whether the new records belong to authors already in the digital library or not.
- Based on heuristics.
  - Very rarely two authors with similar names that share a coauthor in common would be two different people in the real world.
  - Authors tend to publish on the same subjects and venues for some portion of their careers.

INDi

- 3 Steps.
- General Idea:
  - Similar author name AND
  - At least one coauthor in common AND
  - Similar work title OR publication venue title
- Functions similarity
  - For author and coauthor names.
    - Fragment Comparison algorithm [Oliveira 2005, UFMG].
  - For work and publication venue titles.
    - Cosine similarity metric.

INDi

Step 1

Jano M. Souza

Coauthors	**Work Title**	**Publication Venue Title**
G. Zimbrao, V. Almeida	approximate spatial query processing using raster signatures	xvi simposio brasileiro de banco de dados

Jano M. Souza

Coauthors	Work Title	Publication Venue Title
G. Zimbrao, V. Almeida	approximate spatial query processing using raster signatures	xvi simposio brasileiro de banco de dados

Cluster Jano Moreira de Souza			
Author	**Coauthor**	**Work Title**	**Publication Venue Title**
	G. Zimbrao, R. Monteiro, I. Azevedo	A multi-user key and data exchange protocol to manage a secure database	xv simposio brasileiro de banco de dados
	R. Miranda, M. Estolano, F Neto	A raster approximation for processing of polyline joins	xviii simposio brasileiro de banco de dados

Jano M. Souza

Coauthor	Work Title	Publication Venue Title
G. Zimbrao V. Almeida	approximate spatial query processing using raster signatures	xvi simposio brasileiro de banco de dados

Cluster Jano Moreira de Souza			
Author	**Coauthor**	**Work Title**	**Publication Venue Title**
	G. Zimbrao R. Monteiro, I. Azevedo	A multi-user key and data exchange protocol to manage a secure database	xv simposio brasileiro de banco de dados
	R. Miranda, M. Estolano, F Neto	A raster approximation for processing of polyline joins	xviii simposio brasileiro de banco de dados

Jano M. Souza

Coauthor	Work Title	Publication Venue Title
G. Zimbrao, V. Almeida	approximate spatial query processing using raster signatures	xvi simposio brasileiro de banco de dados

Cluster Jano Moreira de Souza			
Author	**Coauthor**	**Work Title**	**Publication Venue Title**
	G. Zimbrao, R. Monteiro, I. Azevedo	A multi-user key and data exchange protocol to manage a secure database	xv simposio brasileiro de banco de dados
	R. Miranda, M. Estolano, F Neto	A raster approximation for processing of polyline joins	xviii simposio brasileiro de banco de dados

Jano M. Souza

Coauthor	Work Title	Publication Venue Title
G. Zimbrao, V. Almeida	approximate spatial query processing using raster signatures	xvi simposio brasileiro de banco de dados

Cluster Jano Moreira de Souza			
Author	**Coauthor**	**Work Title**	**Publication Venue Title**
	G. Zimbrao, R. Monteiro, I. Azevedo	A multi-user key and data exchange protocol to manage a secure database	xv simposio brasileiro de banco de dados
	R. Miranda, M. Estolano, F Neto	A raster approximation for processing of polyline joins	xviii simposio brasileiro de banco de dados

INDi

Step 2

A. Gupta

Coauthors	Work Title	Publication Venue Title
-	Steiner points in tree metrics don't (really) help.	SODA 2001

A. Gupta

Coauthors	Work Title	Publication Venue Title
-	Steiner points in tree metrics don't (really) help.	SODA 2001

Cluster Anupam Gupta			
	Coauthor	Work Title	Publication Venue Title
	Chandra Chekuri, Ilan Newman, Yuri Rabinovich, Alistair Sinclair	Embedding k-outerplanar graphs into l1	SODA 2003

A. Gupta

Coauthors	Work Title	Publication Venue Title
-	Steiner points in tree metrics don't (really) help.	SODA 2001

Cluster Anupam Gupta			
	Coauthor	**Work Title**	**Publication Venue Title**
	Chandra Chekuri, Ilan Newman, Yuri Rabinovich, Alistair Sinclair	Embedding k-outerplanar graphs into l1	SODA 2003

A. Gupta

Coauthors	Work Title	Publication Venue Title
-	Steiner points in tree metrics don't (really) help.	SODA 2001

Cluster Anupam Gupta			
	Coauthor	Work Title	Publication Venue Title
	Chandra Chekuri, Ilan Newman, Yuri Rabinovich, Alistair Sinclair	Embedding k-outerplanar graphs into l1	SODA 2003

INDi

Step 3

A. Gupta

Coauthors	Work Title	Publication Venue Title
Chandra Chekuri, Ilan Newman, Yuri Rabinovich, Alistair Sinclair	Embedding k-outerplanar graphs into l1	SODA 2003

Cluster Anupam Gupta			
	Coauthor	**Work Title**	**Publication Venue Title**
	-	Steiner points in tree metrics don't (really) help.	SODA 2001

INDi

- If all the tests in Steps 1-3 fail, we include the new reference as belonging to a new author.

Experimental Evaluation

Collections

BDBComp
Biblioteca Digital Brasileira de Computação

- 363 citations (1987 – 2007).
- 184 distinct authors.

Synthetic Collections

- SyGAR [Ferreira et al. 2009, ECDL] – tool for generating synthetic collections of citation records.
- *Synthetic* 5 e *Synthetic* 10: 5 datasets.
  10 loads(years)/dataset.
  ± 7000 citations/load.

Experimental Evaluation

Baseline

- HHC - Heurist-based Hierarchical Clustering [Cota et al. 2010, JASIS-T].

Evaluation metric

- ACP, AAP e K

Experimental Evaluation

The parameter values used by INDi and HHC in each dataset.

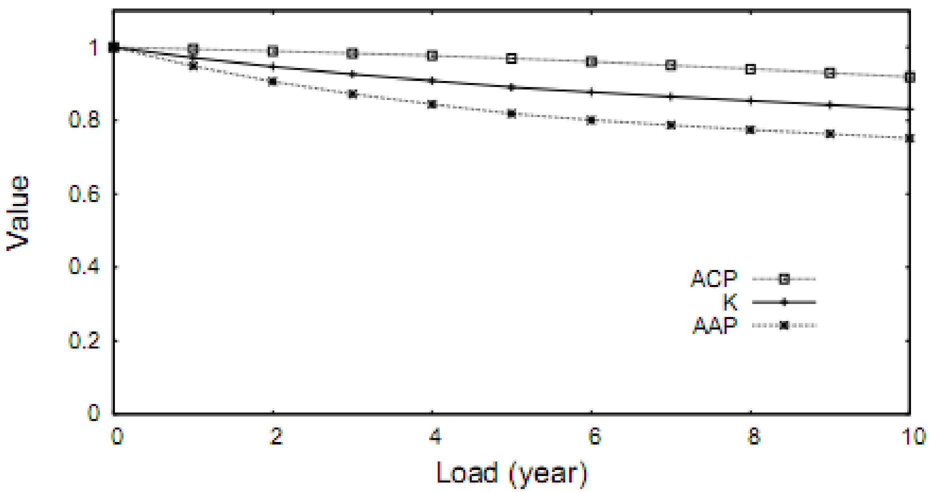
Dataset	INDi			HHC	
	α_{Title}	α_{Venue}	δ	title threshold	venue threshold
BDBComp	0.0	0.2	0.2	0.4	0.4
Synthetic 5	0.1	1.0	0.2	0.3	1.0
Synthetic 10	0.1	0.9	0.2	0.2	0.6

Experimental Evaluation

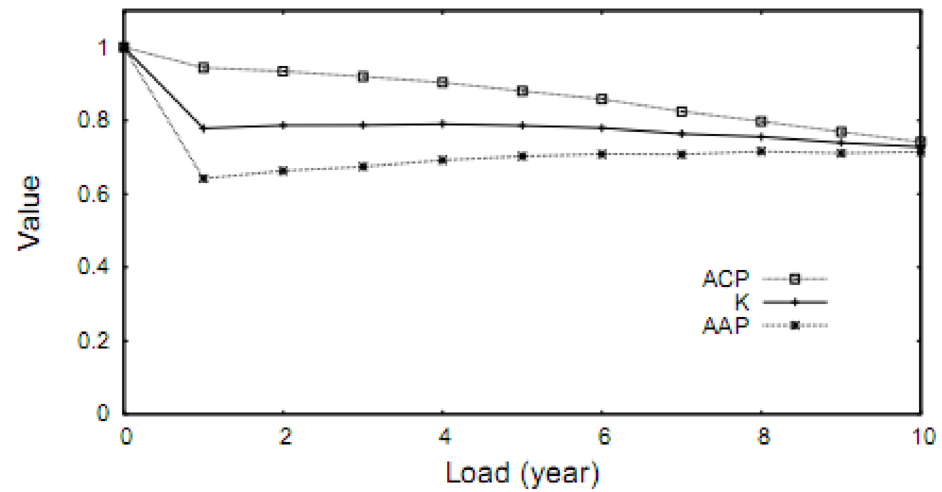
- Results
  - The results in the synthetic datasets correspond to the average results using the 5 datasets.
  - The initial state corresponds to a disambiguated digital library.
  - At each year, a new load of records is inserted into the digital library.
  - HHC reprocesses the whole digital library each time a new load is added to the digital library.

Experimental Evaluation

Performance obtained by INDi and HHC on entire DL.
Collection: Synthetic 5



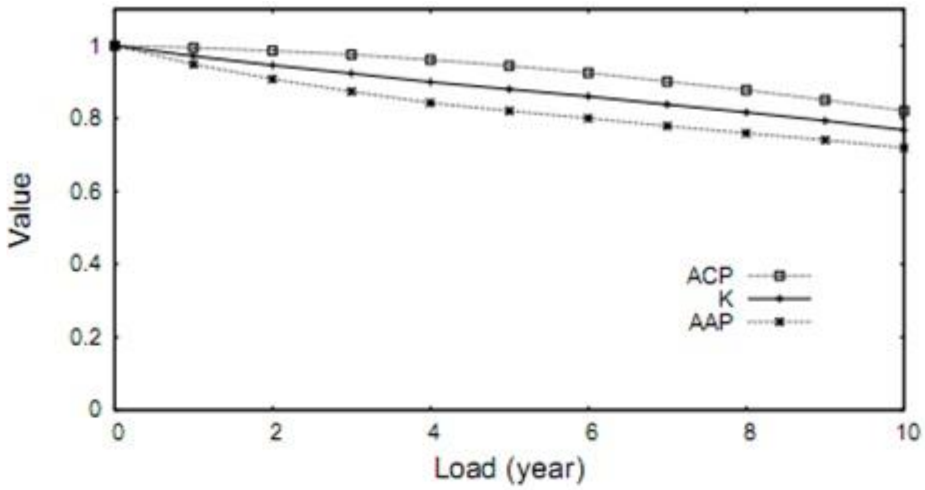
(a) INDi-Synthetic 5



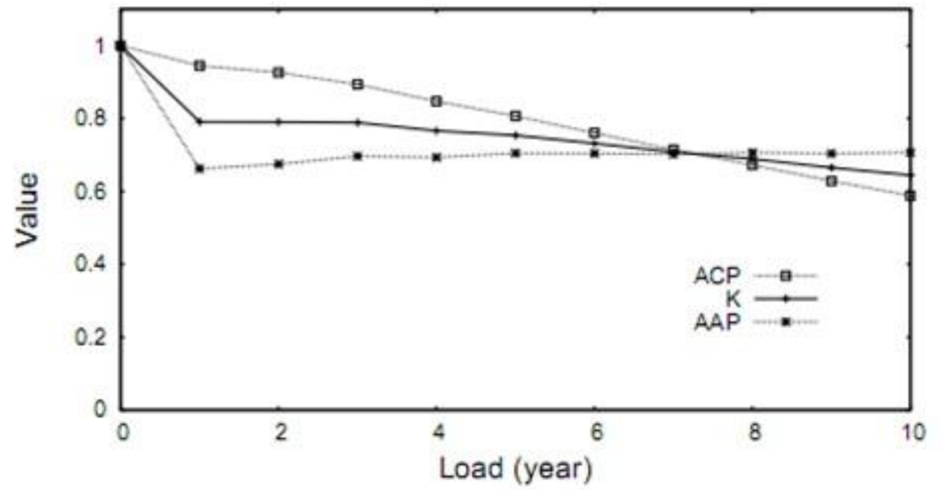
(b) HHC-Synthetic 5

Experimental Evaluation

Performance obtained by INDi and HHC on entire DL Collection: Synthetic 10



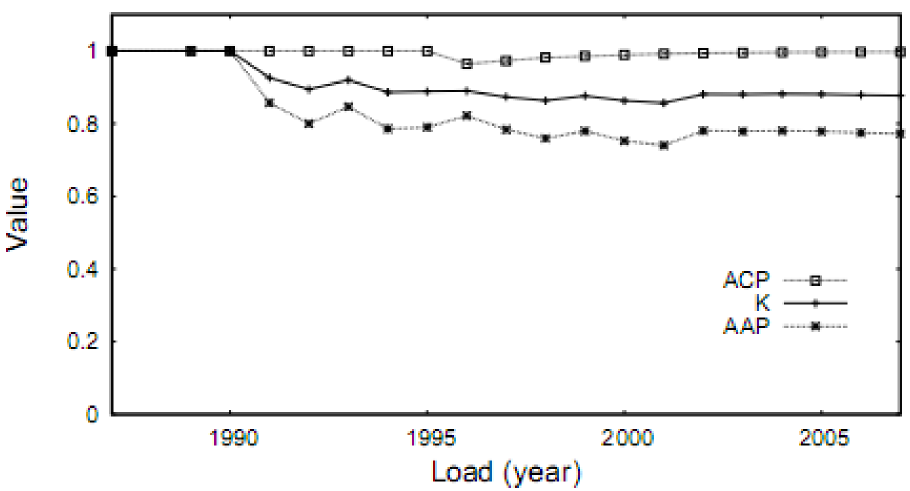
(c) INDi-Synthetic 10



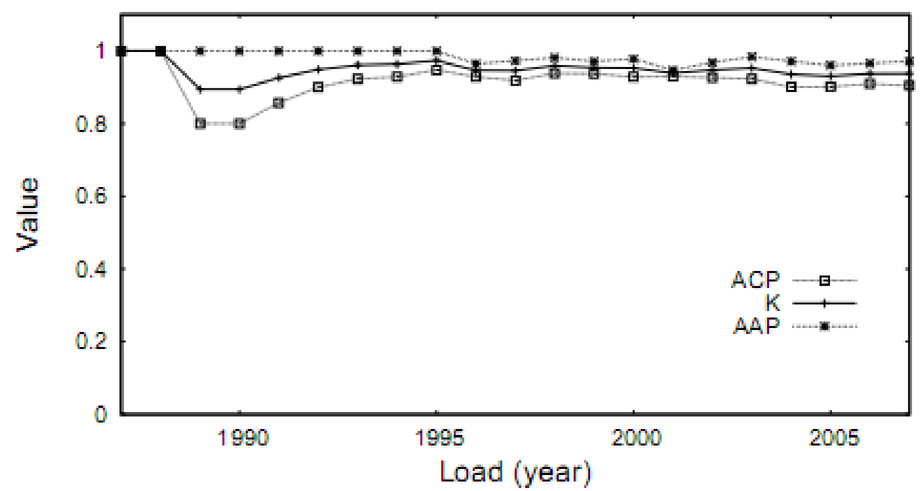
(d) HHC-Synthetic 10

Experimental Evaluation

Performance obtained by INDi and HHC on entire DL Collection: BDBComp



(e) INDi-BDBComp



(f) HHC-BDBComp

Experimental Evaluation

Results obtained by INDi and HHC at the last year with 95% of confidence interval.

Dataset	INDi			HHC		
	K	ACP	AAP	K	ACP	AAP
Synthetic 5	0.831±0.007	0.919±0.010	0.752±0.007	0.728±0.009	0.742±0.011	0.715±0.013
Synthetic 10	0.768±0.009	0.821±0.013	0.719±0.009	0.644±0.018	0.588±0.025	0.707±0.023
BDBComp	0.877	0.997	0.772	0.937	0.905	0.972

Synthetic 5 : INDi is 14% superior to HHC.
Synthetic 10: INDi is 19% superior to HHC.
BDBComp: HHC is 6% superior to INDi.

Experimental Evaluation

Running time (seconds) of INDi and HHC disambiguating Synthetic 10 dataset with 95% of confidence interval.

Method	Load (with average number of references)									
	1(4920)	2(5612)	3(6348)	4(7090)	5(7915)	6(8804)	7(9682)	8(10598)	9(11604)	10(12663)
INDi	0.81 ± 0.16	1.17 ± 0.10	1.51 ±.0.15	1.92 ±0.21	2.91 ±0.24	3.7 ±0.39	4.56 ±0.40	5.64 ±0.47	7.10 ±0.80	8.40 ±0.68
HHC	11.50 ±1.06	14.89 ±0.81	17.49 ±1.79	21.51 ±1.11	25.79 ±1.96	31.48 ±1.98	33.83 ±1.91	45.93 ±2.30	59.05 ±5.01	73.07 ±5.30

Both methods were implemented in Java.

INDi

Analysis of Cases of Failure

Percentage of new and existing authors correctly and incorrectly identified by INDi.

Dataset	New Authors		Existing Authors	
	Correct	Incorrect	Correct	Incorrect
Sinthetic 5	41.803	58.197	87.289	12.711
Sinthetic 10	31.235	68.765	88.723	11.277
BDBComp	99.507	0.493	62.821	37.179

INDi

- Discussion
  - Using datasets generated by a synthetic data generator, INDi shows gains of up of to 19% when compared to a state-of-the-art method.
    - without the cost of having to disambiguate the whole DL at each new load.
    - without the need of any training.

INDi

- Discussion
  - Using data extracted from BDBComp, INDi presents small loses when compared to the same baseline.
    - INDi produces fewer cases of mixed citations, which is a problem that is much harder to manually fix afterwards.
  - INDi does not undo manual corrections.

INDi

Future work

- To investigate and propose alternatives to properly address the cases of failure generated by our method.
- To design strategies to automatically discover the best thresholds for a given dataset.

SyGAR – Synthetic Generator of Authorship Records

SyGAR

- Motivation
  - A solid analysis of existing methods should consider various scenarios that occur in real digital libraries.
  - In addition to dynamic patterns, the analysis should also address the robustness of existing methods under data errors
    - Typographical errors
    - Optical character recognition
    - Speech recognition errors
- The construction of a real, previously disambiguated, temporal collection capturing different relevant dynamic scenarios and including various data errors is quite costly.
- An alternative is to build realistic *synthetic collections* that capture all scenarios of interest, under controlled conditions.

SyGAR

- A generator of realistic synthetic collections, designed for the specific problem of name ambiguity, should be able to:
  - Generate data whose disambiguation is non-trivial, following patterns similar to those found in real collections;
  - Generate successive loads of data containing new publications of the same set of authors;
  - Generate data for new authors that were not originally included in the collection;
  - Generate data reflecting changes in the authors' publication profiles (e.g., changes in the topics in which the authors publish), simulating changes of research interests over time;
  - Introduce controlled errors on generated data, simulating errors caused by typos, misspelling, or OCR.

SyGAR Design

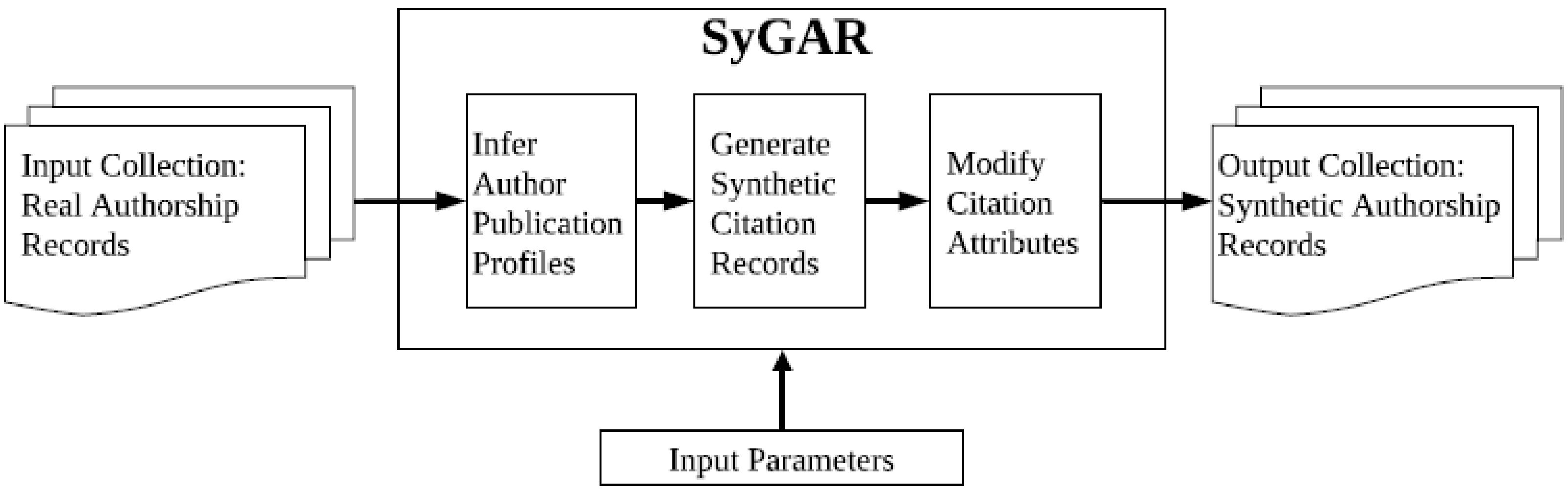


Figure : SyGAR Main Components.

SyGAR Design

- Inferring Publication Profiles from the Input Collection
  - The profile of author a is extracted from the input collection by summarizing her list of citation records into four probability distributions, namely:
    1. a's distribution of number of coauthors per record - $P^a_{nCoauthors}$;
    2. a's coauthor popularity distribution - $P^a_{Coauthor}$;
    3. a's distribution of number of terms in a work title - P^a_{nTerms};
    4. a's topic popularity distribution - P^a_{Topic}.

SyGAR Design

- Each topic *t* is further characterized by two probability distributions:
  1. *t*'s term popularity distribution - P^t_{Term};
  2. *t*'s venue popularity distribution - P^t_{Venue}.

SyGAR Design

- Latent Dirichlet Allocation

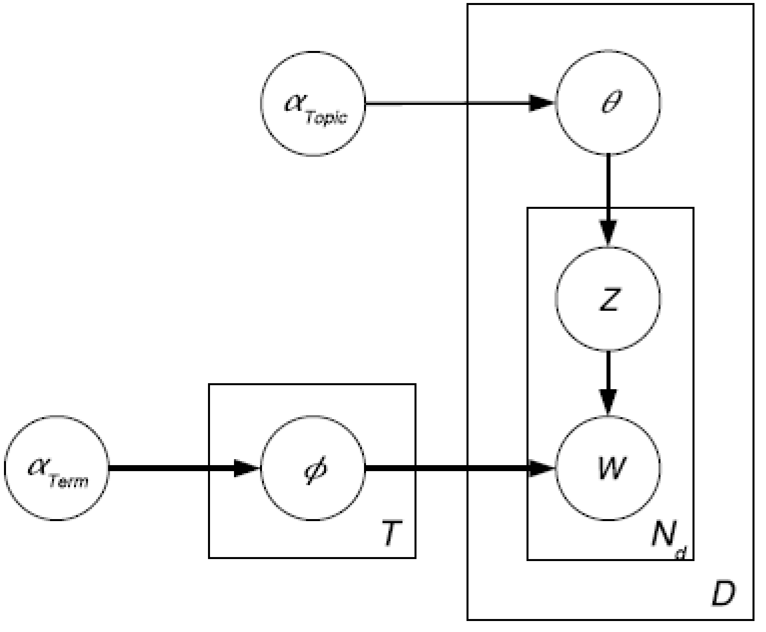


Figure : A plate representation of the LDA.

SyGAR Design

- Inferring Topics distributions
- Topic distribution P^a_{Topic} of each author a
  – SyGAR combines the weights of the topics of all citation records in which *a* is an author
  – Only topics with weights greater than or equal to β_{Topic} (input parameter) are selected from each citation record of *a*.
- The venue popularity distribution of each topic *t*, P^t_{Venue}
  – SyGAR combines the weights of *t* associated with citation records containing the same publication venue

SyGAR Design

- Generating Records for Existing Authors
  - Each synthetic record for existing authors is created as follows:
    1. Select one of the authors of the collection according to the desired distribution of number of records per author. Let it be a.
    2. Select the number of coauthors according to $P^{a}_{nCoauthors}$. Let it be a_c.
    3. Repeat a_c times:
       - with probability $1 - \alpha_{NewCoauthor}$, select one coauthor according to $P^{a}_{Coauthor}$;
       - otherwise, uniformly select a *new coauthor* among remaining coauthors in the input collection.
    4. Combine the topic distributions of a and each of the selected coauthors. Let it be P^{all}_{Topic}.

SyGAR Design

- Generating Records for Existing Authors
  5. Select the number of terms in the title according to $P^a{}_{nTerms}$. Let it be a_t.
  6. Repeat a_t times: select one topic t according to $P^{all}{}_{Topic}$ and select one term for the work title according to $P^t{}_{Term}$.
  7. Select the publication venue:
     - With probability $1 - \alpha_{NewVenue}$, select a venue according to $P^t{}_{Venue}$, where t is the topic that was selected most often in Step 6;
     - Otherwise, randomly select a new venue among remaining venues in the input collection.

SyGAR Design

- Adding New Authors
  - We adopt a strategy that exploits the publication profiles from author and co-authors, extracted from the input collection.
  - A new author *a* is created by first selecting one of its coauthors. Let say it is c_a.
  - The new author inherits c_a's profile, but the inherited topic and coauthor distributions are changed as follows:
    - The new author inherits only a percentage $\%_{InheritedTopics}$ of the topics associated with c_a
    - We set *a*'s coauthor list equal to c_a plus all coauthors of c_a that have at least one of the topics in l_{Topic} associated with them.
  - The name of the new author is generated with the initial of the first name and the full last name of an existing author using the distribution of the number of records per ambiguous group.

SyGAR Design

- Changing an Author's Profile

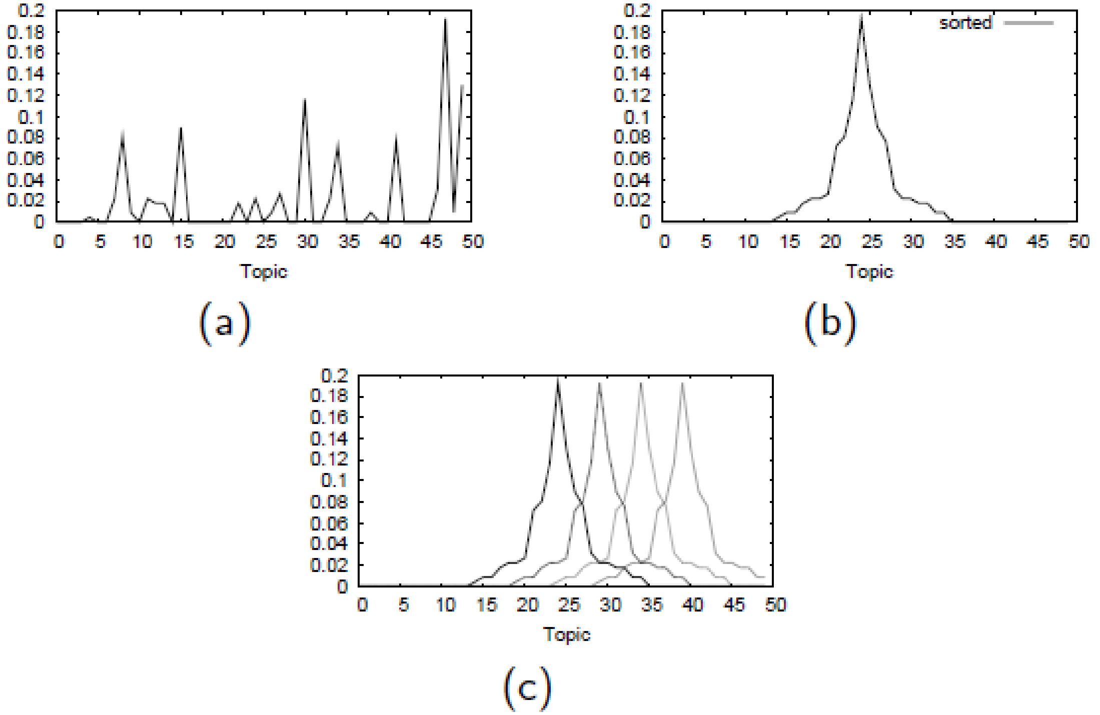


Figure : Changing Author *a*'s Profile by Altering her Topic Distribution.

SyGAR

- Validation
  - We here select three methods, each one representative of a different technique:
    - The SVM-based name disambiguation method (SVM)
    - The Unsupervised Heuristic-based Hierarchical Clustering method (HHC)
    - The K-way Spectral Clustering-based method (KWAY)
  - We validate SyGAR by comparing the performance of the selected name disambiguation methods on real and synthetically generated collections.

SyGAR

- Validation

Table : SyGAR Validation – Average K Results and 95% Confidence Intervals for Real and Synthetically Generated Collections ($N_{Topics} = 300$). Statistical ties are in bold.

Collection	KWAY	SVM	HHC
Real	0.530±0.009	0.764±0.005	**0.770**±0.006
Synthetic 1	0.478±0.005	0.698±0.008	**0.753**±0.013
Synthetic 2	0.484±0.007	0.706±0.005	0.750±0.011
Synthetic 3	0.478±0.008	0.701±0.006	0.752±0.005
Synthetic 4	0.480±0.006	0.708±0.007	0.755±0.006
Synthetic 5	0.477±0.009	0.702±0.006	0.751±0.011

SyGAR

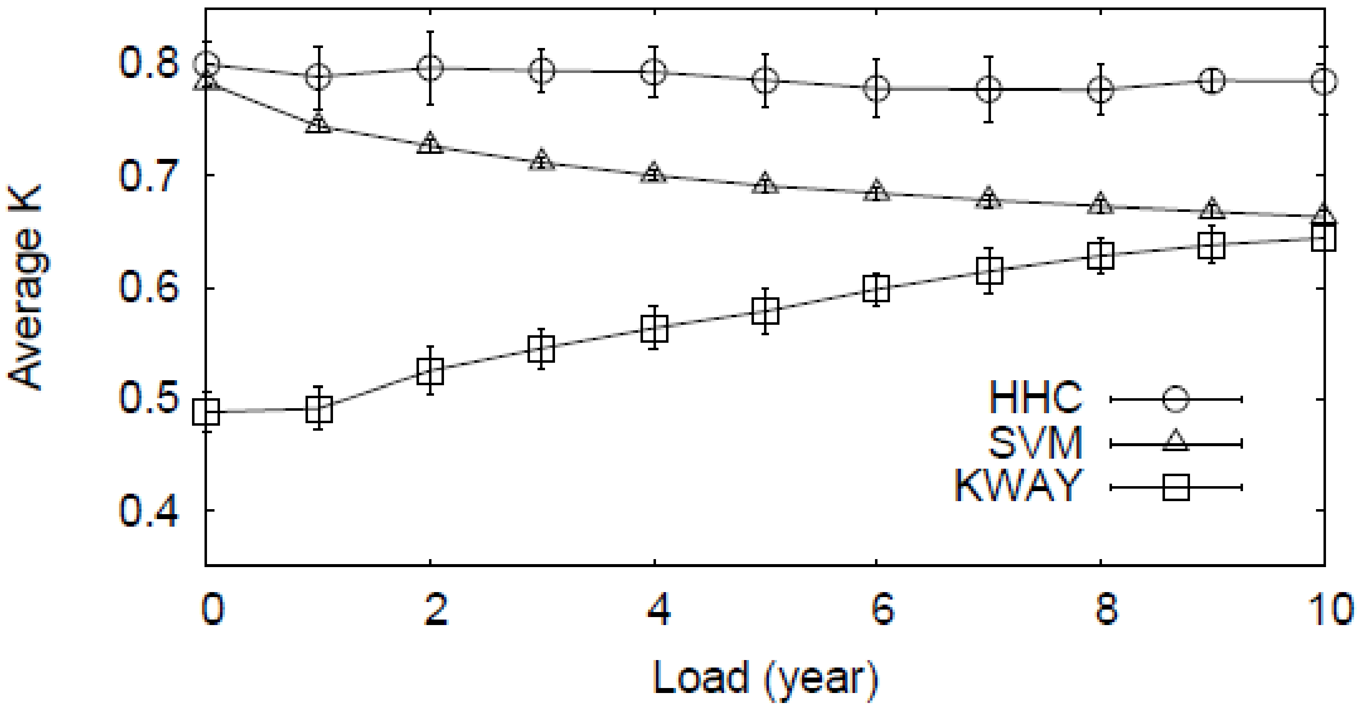
- Validation

Table : SyGAR Validation: Average K Results and 95% Confidence Intervals for Real and 5 Synthetically Generated Collections ($N_{Topics} = 600$).

Collection	KWAY	SVM	HHC
Real	0.530±0.009	0.764±0.005	0.770±0.006
Synthetic 1	0.499±0.008	0.746±0.007	0.793±0.008
Synthetic 2	0.489±0.006	0.743±0.007	0.790±0.009
Synthetic 3	0.493±0.006	0.742±0.007	0.799±0.012
Synthetic 4	0.491±0.006	0.750±0.006	0.796±0.006
Synthetic 5	0.497±0.010	0.743±0.010	0.801±0.008

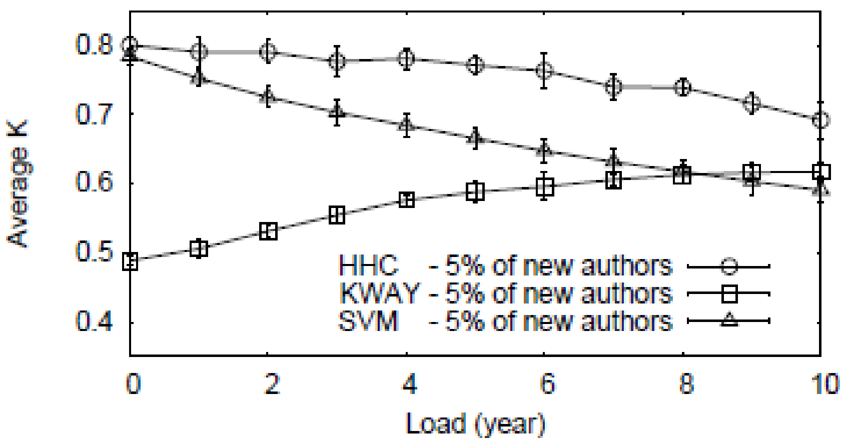
Evaluating Disambiguation Methods

Scenario 1 – Evolving DL with Static Author Population and Publication Profiles

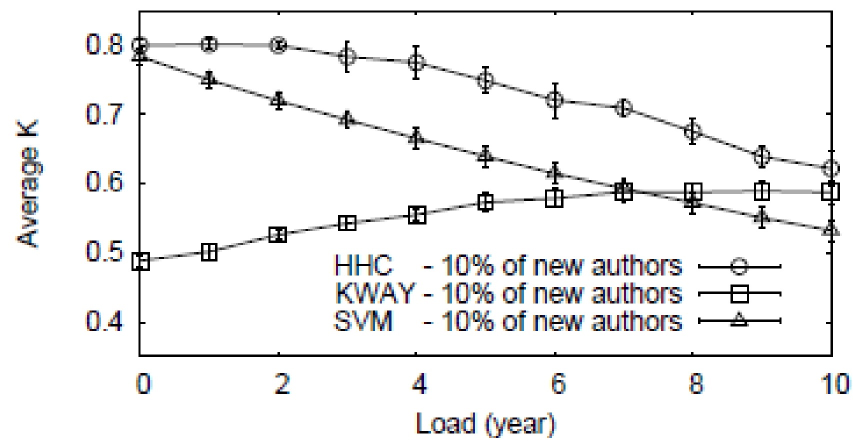


Evaluating Disambiguation Methods

Scenario 2 – Evolving DL and Addition of New Authors ($\%_{InheritedTopics}$=80%)



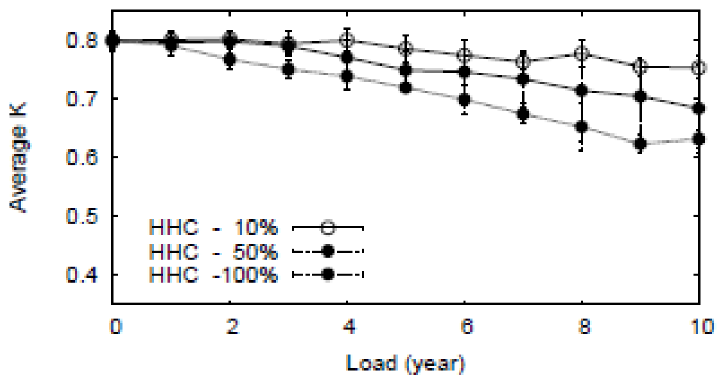
(a) $\%_{NewAuthors}$=5%



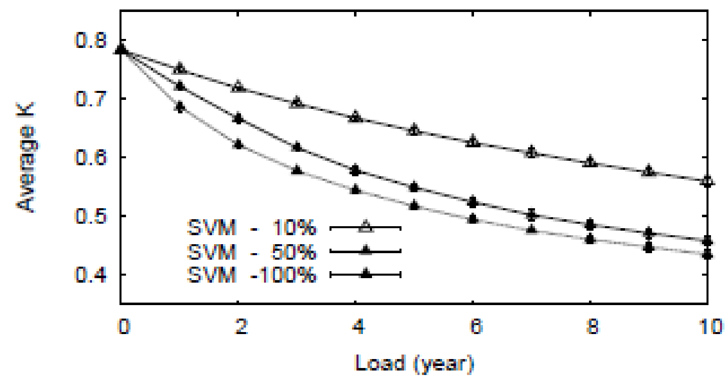
(b) $\%_{NewAuthors}$=10%

Evaluating Disambiguation Methods

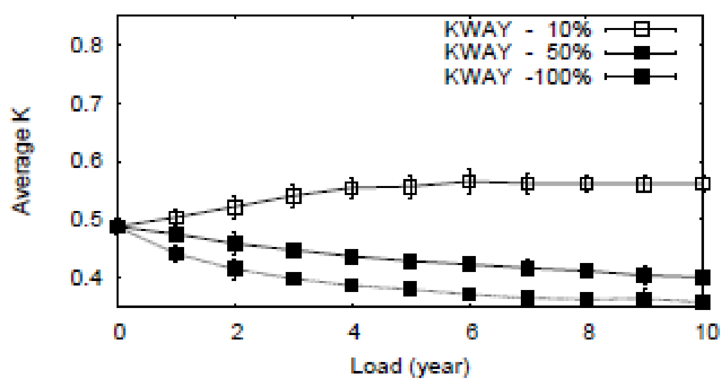
Scenario 3 – Dynamic Author Profiles ($\delta = 5$ and $\%_{ProfileChanges} = 10\%, 50\%$ and 100%)



(a) HHC



(b) SVM



(c) KWAY

Open challenges

- Very Little Data in the Citations.
  - In most cases we have only the basic information about the citations available. Furthermore, in some cases author names contain only the initial and the last surname and the publication venue title is abbreviated.
- Very Ambiguous Cases.
  - Several methods exploit coauthor-based heuristics, by explicitly assuming the hypotheses that: (i) very rarely ambiguous references will have coauthors in common who have also ambiguous names; or (ii) it is rare that two authors with very similar names work in the same research area.

Open challenges

- Citations with Errors.
  - Errors occur in citation data which are sometimes impossible to detect. The methods need to be tolerant to such errors.
- Efficiency.
  - With the high amount of articles being published nowadays in the different knowledge areas, the solutions need to deal with the problem efficiently.

Open challenges

- Different Knowledge Areas.
  - As we have seen, most of the collections used to evaluate the methods are related to Computer Science. However, other knowledge areas (e.g., Humanities, Medicine) may have different publication patterns.
- Incremental Disambiguation.
  - Ideally disambiguation should be performed incrementally as new citations are incorporated into the DL.

Open challenges

- Author Profile Changes.
  - It is common that the research interests of an author change over time. These changes cause modifications in the model representing the author profile causing difficulties for the methods.
- New Authors.
  - The methods should be capable of identifying references to new ambiguous authors who do not have citations in the DL yet.

References

- Bhattacharya, I. and Getoor, L. (2007). Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data*, 1(1).
- Carvalho, A. P., Ferreira, A. A., Laender, A. H. F., and Gonçalves, M. A. (2011). Incremental unsupervised name disambiguation in cleaned digital libraries. *Journal of Information and Data Management*, 2(3):289-304.
- Cota, R. G., Ferreira, A. A., Gonçalves, M. A., Laender, A. H. F., and Nascimento, C. (2010). An unsupervised heuristic-based hierarchical method for name disambiguation in bibliographic citations. *Journal of the American Society for Information Science and Technology*, 61(9):1853-1870.
- Fan, X., Wang, J., Pu, X., Zhou, L., and Lv, B. (2011). On graph-based name disambiguation. *ACM Journal of Data and Information Quality*, 2:10:1-10:23.
- Ferreira, A. A., Gonçalves, M. A., and Laender, A. H. F. (2014). Disambiguating Author Names using Minimum Bibliographic Information. *World Digital Library, 7(1):71-84.*
- Ferreira, A. A., Gonçalves, M. A., Veloso, A., and Laender, A. H. F. (2014). Self-training author name disambiguation for information scarce scenarios. *Journal of the American Society for Information Science and Technology, 65(6):1257-1278.*

References

- Ferreira, A. A., Gonçalves, M. A., Almeida, J. M., Laender, A. H. F., and Veloso, A. (2012). A tool for generating synthetic authorship records for evaluating author name disambiguation methods. *Information Sciences*, 206:42-62.
- Ferreira, A. A., Goncalves, M. A., and Laender, A. H. F. (2012). A brief survey of automatic methods for author name disambiguation. *SIGMOD Record*, 41(2):15-26.
- Ferreira, A. A., Veloso, A., Goncalves, M. A., and Laender, A. H. F. (2010). Effective self-training author name disambiguation in scholarly digital libraries. In *Proceedings of the 2010 ACM/IEEE Joint Conference on Digital Libraries*, pages 39-48, Gold Coast, Queensland, Australia.
- Han, H., Giles, C. L., Zha, H., Li, C., and Tsioutsiouliklis, K. (2004). Two supervised learning approaches for name disambiguation in author citations. In *Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 296-305, Tuscon, USA.
- Han, H., Xu, W., Zha, H., and Giles, C. L. (2005a). A hierarchical naive Bayes mixture model for name disambiguation in author citations. In *Proceedings of the 2005 ACM Symposium on Applied Computing*, pages 1065-1069, Santa Fe, New Mexico, USA.
- Han, H., Zha, H., and Giles, C. L. (2005b). Name disambiguation in author citations using a k-way spectral clustering method. In *Proceedings of the 5th ACM/IEEE Joint Conference on Digital Libraries*, pages 334-343, Denver, CO, USA.
- Huang, J., Ertekin, S., and Giles, C. L. (2006). Efficient name disambiguation for large scale databases. In *Proceedings of the European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 536-544, Berlin, Germany.

XXIX SBDD Simpósio Brasileiro de Banco de Dados

References

- Laender, A. H. F., Goncalves, M. A., Cota, R. G., Ferreira, A. A., Santos, R. L. T., and Silva, A. J. C. (2008). Keeping a digital library clean: new solutions to old problems. In *Proceedings of the 2008 ACM Symposium on Document Engineering*, Sao Paulo, Brazil, September 16-19, 2008, pages 257-262. ACM.
- Levin, F. H. and Heuser, C. A. (2010). Evaluating the use of social networks in author name disambiguation in digital libraries. *Journal of Information and Data Management*, 1(2):183-197.
- Levin, M., Krawzyk, S., Bethard, S., and Jurafsky, D. (2012). Citation-based bootstrapping for large-scale author disambiguation. *Journal of the American Society for Information Science and Technology*, 63(5):1030-1047.
- Pereira, D. A., Ribeiro-Neto, B. A., Ziviani, N., Laender, A. H. F., Goncalves, M. A., and Ferreira, A. A. (2009). Using web information for author name disambiguation. In *Proceedings of the 2009 ACM/IEEE Joint Conference on Digital Libraries*, pages 49-58, Austin, TX, USA.
- Tang, J., Fong, A. C. M.,Wang, B., and Zhang, J. (2012). A unied probabilistic framework for name disambiguation in digital library. *IEEE Transactions on Knowledge and Data Engineering*, 24(6):975-987.
- Torvik, V. I. and Smalheiser, N. R. (2009). Author name disambiguation in medline. *ACM Transactions on Knowledge Discovery from Data*, 3(3):1-29.
- Treeratpituk, P. and Giles, C. L. (2009). Disambiguating authors in academic publications using random forests. In *Proceedings of the 2009 ACM/IEEE Joint Conference on Digital Libraries*, pages 39-48, Austin, TX, USA.
- Veloso, A., Ferreira, A. A., Goncalves, M. A., Laender, A. H., and Meira Jr., W. (2012). Cost-effective on-demand associative author name disambiguation. Information Processing & Management, 48(4):680-697.

Acknowledgements

XXIX SBBD
Simpósio Brasileiro de Banco de Dados

CNPq
Conselho Nacional de Desenvolvimento Científico e Tecnológico

C A P E S

FAPEMIG

UFOP decom
departamento de computação

DCC
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

UFMG

LBD
UFMG

inweb
Instituto Nacional de Ciência e Tecnologia para a Web

Thanks

- Anderson Almeida Ferreira
  - Departamento de Computação
  - Universidade Federal de Ouro Preto
  - ferreira@iceb.ufop.br
- Marcos André Gonçalves
  - Departamento de Ciência da Computação
  - Universidade Federal de Minas Gerais
  - mgoncalv@dcc.ufmg.br
- Alberto H. F. Laender
  - Departamento de Ciência da Computação
  - Universidade Federal de Minas Gerais
  - laender@dcc.ufmg.br