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

Name: **Mohammed Zaki**

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, ...
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

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

List of publications from the DBLP Bibliography Server - FAQ

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

Mohammed Javeed Zaki from the Department of Computer Science, Rensselaer Polytechnic Institute, USA
## The author name disambiguation task

### An illustrative example

<table>
<thead>
<tr>
<th>Citation Id</th>
<th>Citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_1$</td>
<td>(r₁) S. Godbole, (r₂) I. Bhattacharya, (r₃) A. Gupta, (r₄) A. Verma. Building re-usable dictionary repositories for real-world text mining. CIKM, 2010.</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Citation Id</th>
<th>Citation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>c\textsubscript{1}</strong></td>
<td>((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.</td>
</tr>
<tr>
<td><strong>c\textsubscript{2}</strong></td>
<td>((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.</td>
</tr>
<tr>
<td><strong>c\textsubscript{3}</strong></td>
<td>((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.</td>
</tr>
<tr>
<td><strong>c\textsubscript{4}</strong></td>
<td>((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.</td>
</tr>
</tbody>
</table>
# The author name disambiguation task

## Preprocessing – stop-word removal

<table>
<thead>
<tr>
<th>Citation Id</th>
<th>Citation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>c&lt;sub&gt;1&lt;/sub&gt;</strong></td>
<td>(r&lt;sub&gt;1&lt;/sub&gt;) S. Godbole, (r&lt;sub&gt;2&lt;/sub&gt;) I. Bhattacharya, (r&lt;sub&gt;3&lt;/sub&gt;) A. Gupta, (r&lt;sub&gt;4&lt;/sub&gt;) A. Verma. building usable dictionary repositories real world text mining. CIKM, 2010.</td>
</tr>
<tr>
<td><strong>c&lt;sub&gt;3&lt;/sub&gt;</strong></td>
<td>(r&lt;sub&gt;11&lt;/sub&gt;) T. Nghiem, (r&lt;sub&gt;12&lt;/sub&gt;) S. Sankaranarayanan, (r&lt;sub&gt;13&lt;/sub&gt;) G. E. Fainekos, (r&lt;sub&gt;14&lt;/sub&gt;) F. Ivancic, (r&lt;sub&gt;15&lt;/sub&gt;) A. Gupta, (r&lt;sub&gt;16&lt;/sub&gt;) G. J. Pappas. monte carlo techniques falsification temporal properties linear hybrid systems. HSCC, 2010.</td>
</tr>
<tr>
<td><strong>c&lt;sub&gt;4&lt;/sub&gt;</strong></td>
<td>(r&lt;sub&gt;17&lt;/sub&gt;) William R. Harris, (r&lt;sub&gt;18&lt;/sub&gt;) Sriram Sankaranarayanan, (r&lt;sub&gt;19&lt;/sub&gt;) Franjo Ivancic, (r&lt;sub&gt;20&lt;/sub&gt;) Aarti Gupta. program analysis satisfiability modulo path programs. POPL, 2010.</td>
</tr>
</tbody>
</table>
### The author name disambiguation task

#### Preprocessing - stemming

<table>
<thead>
<tr>
<th>Citation Id</th>
<th>Citation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>c₁</strong></td>
<td>(r₁) S. Godbole, (r₂) I. Bhattacharya, (r₃) A. Gupta, (r₄) A. Verma. build usable dictionary repository real world text mine. CIKM, 2010.</td>
</tr>
</tbody>
</table>
The author name disambiguation task

- \( a_1 = \{(r_1), (r_6)\} \) - 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

- Type of approach
  - Author grouping
  - Author assignment
- Explored evidence
  - Citation information
  - Web information
  - Implicit evidence
- Classification
- Clustering
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 \( \{f_1, f_2, ..., f_q\} \).
  – 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
  – 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
      – $s^{(i,j)} = [\text{sim}_{1}(t^{(i)}_{u,1}, t^{(j)}_{v,1}), \ldots, \text{sim}_{m}(t^{(i)}_{u,m}, t^{(j)}_{v,m})]$
      – Edit distance $\rightarrow$ emails and URLs
      – Jaccard similarity $\rightarrow$ addresses and affiliations
      – Soft-TFIDF $\rightarrow$ 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 $T$.
    • $F: \{f_1, f_2, ..., f_m\} \rightarrow \{a_1, a_2, ..., a_n\}$
  – The disambiguator essentially divides the records in $T$ into $n$ sets $\{a_1, a_2, ..., a_n\}$, 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

- A set of references whose authors are known (Training set)
- Classifier
- A set of references whose authors are unknown
- Disambiguation Function $F$
- The set of references with their corresponding author
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 an 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


  – 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

<table>
<thead>
<tr>
<th>Citation Id</th>
<th>Citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_1$</td>
<td>$(r_1)$ 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.</td>
</tr>
<tr>
<td>$c_4$</td>
<td>$(r_{17})$ William R. Harris, $(r_{18})$ Sriram Sankaranarayanan, $(r_{19})$ Franjo Ivancic, $(r_{20})$ Aarti Gupta. Program analysis via satisfiability modulo path programs. POPL, 2010.</td>
</tr>
</tbody>
</table>
Author name disambiguation methods

Explored evidence

• Web information

\[ r_1 \]
\[ r_2 \]
\[ r_3 \]
\[ r_4 \]
\[ r_5 \]
\[ \ldots \]
\[ r_n \]
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.
Algorithm 1. HHC.

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

1. Let $A$ be a list of authorship records;
2. Let $C_1$ and $C_2$ be lists of clusters;
3. Let $G$ be a list of ambiguous groups;
4. Let $R'$ be a list of citation records;
5. $R' \leftarrow$ PreprocessCitationRecords($R$);
6. $A \leftarrow$ CreateAuthorshipRecords($R'$);
7. $G \leftarrow$ CreateAmbiguousGroups($A$);
8. $C \leftarrow \emptyset$
9. for each ambiguous group $g$ in $G$ do
   10. $C_1 \leftarrow$ FirstStep($g$);
   11. $C_2 \leftarrow$ SecondStep($C_1$);
   12. $C \leftarrow C \cup C_2$;
13. end for
Algorithm 2. FirstStep.

**Input:** Ambiguous group $g$;

**Output:** List $C$ of clusters of authorship records;

1. Let $L$ and $S$ be lists of authorship records;
2. Let $C$, $C_1$ and $C_2$ be lists of clusters;
3. $S \leftarrow \text{GetShortNameRecords}(g)$;
4. $L \leftarrow \text{GetLongNameRecords}(g)$;
5. $C_1 \leftarrow \emptyset$;
6. $C_2 \leftarrow \text{ProcessList}(L,C_1)$;
7. $C \leftarrow \text{ProcessList}(S,C_2)$;

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

1. $C_o \leftarrow C_i$;
2. $fused \leftarrow true$;
3. while $fused$ do
   4. $fused \leftarrow false$;
   5. for each $c_1$ in $C_o$ do
      6. for each $c_2$ in $C_o$ do
         7. 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
            8. $t_{t1} \leftarrow \text{GetWorkTitleTerms}(c_1)$;
            9. $t_{t2} \leftarrow \text{GetWorkTitleTerms}(c_2)$;
           10. $t_{v1} \leftarrow \text{GetPublicationVenueTitleTerms}(c_1)$;
           11. $t_{v2} \leftarrow \text{GetPublicationVenueTitleTerms}(c_2)$;
            12. if $\text{TitleSimilarity}(t_{t1}, t_{t2}) > \text{title-threshold}$ or $\text{VenueSimilarity}(t_{v1}, t_{v2}) > \text{venue-threshold}$ then
               13. $c_1 \leftarrow \text{Fuse}(c_1, c_2)$;
               14. remove($C_o, c_2$);
               15. $fused \leftarrow true$;
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

– 184 distinct authors.

dblp
computer science bibliography

– 4,287 records
– 220 distinct authors
HHC
Comparative Evaluation

Evaluation Metrics

• ACP – *Average cluster purity*

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

• AAP – *Average author purity*

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

• $K$

\[
K = \sqrt{ACP \times AAP}
\]

$N$ = 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

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

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

- BDBComp

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

\[ a_1 \quad a_2 \quad a_3 \]
SAND Design

The Cluster Selection Step

Test set $T$

Training data $D$

$a_1$ $a_2$ $a_3$
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 $\{f_1, f_2, \ldots, f_m\}$ to $\{a_1, a_2, \ldots, a_n\}$ 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\} \to \{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

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

\[
s(a_i, x) = \frac{|R_{a_i}^x|}{\sum_{j=1}^{|R_{a_i}^x|} \theta(r_j)}
\]
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 $a_i$.

$$\Delta(x) = \frac{\hat{p}(a_i | x)}{\hat{p}(a_j | x)}$$

  – The idea is to only predict $a_i$ if $\Delta(x) \geq \Delta_{\text{min}}$.

• 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, $\gamma_{min}$.

• For an reference $x \in T$, if the number of rules extracted from $D^x (\gamma(x))$ is smaller than $\gamma_{min}$, 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 $\gamma_{min}$ 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

Test set $T$

Training data $D$

$T$

$D$

$T$

$D$

$D$

$D$
The Author Assignment Step
Predicting the Author of Each Cluster

Test set
\( T \)

Training data
\( D \)

\( a_1 \)

\( a_2 \)

\( a_3 \)
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

Test set
\( T \)

Training data
\( D \)
The Author Assignment Step
Predicting the Author of Each Cluster

Test set
$T$

Training data
$D$

$a_1$

$a_2$

$a_3$

$a_4$
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.

<table>
<thead>
<tr>
<th>Ambiguous Group</th>
<th>ACP</th>
<th>AAP</th>
<th>K</th>
<th>pP</th>
<th>pR</th>
<th>pF1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Gupta</td>
<td>0.990 ± 0.002</td>
<td>0.416 ± 0.033</td>
<td>0.641 ± 0.025</td>
<td>0.994 ± 0.001</td>
<td>0.398 ± 0.056</td>
<td>0.567 ± 0.058</td>
</tr>
<tr>
<td>A Kumar</td>
<td>0.985 ± 0.003</td>
<td>0.242 ± 0.011</td>
<td>0.490 ± 0.011</td>
<td>0.995 ± 0.003</td>
<td>0.098 ± 0.006</td>
<td>0.178 ± 0.010</td>
</tr>
<tr>
<td>C Chen</td>
<td>0.953 ± 0.003</td>
<td>0.202 ± 0.003</td>
<td>0.439 ± 0.003</td>
<td>0.906 ± 0.008</td>
<td>0.050 ± 0.001</td>
<td>0.095 ± 0.002</td>
</tr>
<tr>
<td>D Johnson</td>
<td>1.000 ± 0.000</td>
<td>0.301 ± 0.008</td>
<td>0.548 ± 0.008</td>
<td>1.000 ± 0.000</td>
<td>0.295 ± 0.016</td>
<td>0.455 ± 0.019</td>
</tr>
<tr>
<td>J Martin</td>
<td>0.987 ± 0.007</td>
<td>0.500 ± 0.007</td>
<td>0.702 ± 0.007</td>
<td>0.957 ± 0.023</td>
<td>0.322 ± 0.005</td>
<td>0.482 ± 0.008</td>
</tr>
<tr>
<td>J Robinson</td>
<td>1.000 ± 0.000</td>
<td>0.355 ± 0.007</td>
<td>0.596 ± 0.005</td>
<td>1.000 ± 0.000</td>
<td>0.285 ± 0.010</td>
<td>0.443 ± 0.011</td>
</tr>
<tr>
<td>J Smith</td>
<td>0.971 ± 0.007</td>
<td>0.263 ± 0.031</td>
<td>0.504 ± 0.032</td>
<td>0.982 ± 0.018</td>
<td>0.279 ± 0.054</td>
<td>0.432 ± 0.067</td>
</tr>
<tr>
<td>K Tanaka</td>
<td>1.000 ± 0.000</td>
<td>0.380 ± 0.008</td>
<td>0.616 ± 0.006</td>
<td>1.000 ± 0.000</td>
<td>0.231 ± 0.008</td>
<td>0.375 ± 0.011</td>
</tr>
<tr>
<td>M Brown</td>
<td>1.000 ± 0.000</td>
<td>0.395 ± 0.007</td>
<td>0.629 ± 0.006</td>
<td>1.000 ± 0.000</td>
<td>0.340 ± 0.013</td>
<td>0.507 ± 0.015</td>
</tr>
<tr>
<td>M Jones</td>
<td>1.000 ± 0.000</td>
<td>0.281 ± 0.015</td>
<td>0.530 ± 0.014</td>
<td>1.000 ± 0.000</td>
<td>0.251 ± 0.021</td>
<td>0.400 ± 0.026</td>
</tr>
<tr>
<td>M Miller</td>
<td>0.991 ± 0.005</td>
<td>0.603 ± 0.026</td>
<td>0.773 ± 0.017</td>
<td>0.988 ± 0.009</td>
<td>0.586 ± 0.034</td>
<td>0.735 ± 0.026</td>
</tr>
</tbody>
</table>
Experimental Evaluation
Evaluating the Author Grouping Step

Table: Results obtained by the author grouping step in the DBLP and collections, using the popular last names.

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

DBLP

BDBComp

K

Purity (ACP)

Cohesion (AAP)

value

value

$\phi_{\min}$

$\phi_{\min}$

0

0.2

0

0.2

0.4

0.6

0.8

1

0

0.2

0.4

0.6

0.8

1

0.2

0.4

0.6

0.8

1

0.2

0.4

0.6

0.8

1
Experimental Evaluation
Comparison with the Author Grouping Baselines

<table>
<thead>
<tr>
<th>Method</th>
<th align="right">DBLP K</th>
<th align="right">DBLP pF1</th>
<th align="right">BDBComp K</th>
<th align="right">BDBComp pF1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAND</td>
<td align="right">0.815</td>
<td align="right">0.796</td>
<td align="right">0.924</td>
<td align="right">0.752</td>
</tr>
<tr>
<td>HHC</td>
<td align="right">0.773</td>
<td align="right">0.751</td>
<td align="right">0.913</td>
<td align="right">0.756</td>
</tr>
<tr>
<td>KWAY</td>
<td align="right">0.560</td>
<td align="right">0.402</td>
<td align="right">0.805</td>
<td align="right">0.436</td>
</tr>
<tr>
<td>LASVM-DBSCAN</td>
<td align="right">0.551</td>
<td align="right">0.406</td>
<td align="right">0.757</td>
<td align="right">0.211</td>
</tr>
</tbody>
</table>
Experimental Evaluation

Comparison with the Supervised Author Assignment Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>DBLP K</th>
<th>DBLP pF1</th>
<th>BDBComp K</th>
<th>BDBComp pF1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAND</td>
<td>0.775±0.010</td>
<td>0.720±0.018</td>
<td>0.940±0.014</td>
<td>0.462±0.040</td>
</tr>
<tr>
<td>SLAND</td>
<td>0.877±0.007</td>
<td>0.867±0.008</td>
<td>0.900±0.016</td>
<td>0.456±0.028</td>
</tr>
<tr>
<td>SVM</td>
<td>0.799±0.008</td>
<td>0.721±0.010</td>
<td>0.481±0.024</td>
<td>0.160±0.032</td>
</tr>
<tr>
<td>NB</td>
<td>0.736±0.009</td>
<td>0.647±0.012</td>
<td>0.420±0.009</td>
<td>0.160±0.019</td>
</tr>
</tbody>
</table>
# Experimental Evaluation

Comparison with Other Supervised Methods for the Author Assignment Step

<table>
<thead>
<tr>
<th>Method</th>
<th>DBLP</th>
<th>BDBComp</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>K</td>
<td>pF1</td>
</tr>
<tr>
<td>SAND</td>
<td>0.815±0.010</td>
<td>0.796±0.020</td>
</tr>
<tr>
<td>S-SVM</td>
<td>0.666±0.009</td>
<td>0.489±0.018</td>
</tr>
<tr>
<td>S-NB</td>
<td>0.640±0.014</td>
<td>0.466±0.026</td>
</tr>
</tbody>
</table>
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
<table>
<thead>
<tr>
<th>Coauthors</th>
<th>Work Title</th>
<th>Publication Venue Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>G. Zimbrao, V. Almeida</td>
<td>approximate spatial query processing using raster signatures</td>
<td>xvi simposio brasileiro de banco de dados</td>
</tr>
</tbody>
</table>
### Cluster Jano Moreira de Souza

<table>
<thead>
<tr>
<th>Author</th>
<th>Coauthor</th>
<th>Work Title</th>
<th>Publication Venue Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>G. Zimbrao, R. Monteiro, I. Azevedo</td>
<td>A multi-user key and data exchange protocol to manage a secure database</td>
<td>xv simposio brasileiro de banco de dados</td>
<td></td>
</tr>
<tr>
<td>R. Miranda, M. Estolano, F Neto</td>
<td>A raster approximation for processing of polyline joins</td>
<td>xviii simposio brasileiro de banco de dados</td>
<td></td>
</tr>
<tr>
<td>Author</td>
<td>Coauthor</td>
<td>Work Title</td>
<td>Publication Venue Title</td>
</tr>
<tr>
<td>--------</td>
<td>----------</td>
<td>------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>G. Zimbrao</td>
<td>V. Almeida</td>
<td>approximate spatial query processing using raster signatures</td>
<td>xvi simposio brasileiro de banco de dados</td>
</tr>
</tbody>
</table>

Cluster Jano Moreira de Souza

<table>
<thead>
<tr>
<th>Author</th>
<th>Coauthor</th>
<th>Work Title</th>
<th>Publication Venue Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>G. Zimbrao</td>
<td>R. Monteiro, I. Azevedo</td>
<td>A multi-user key and data exchange protocol to manage a secure database</td>
<td>xv simposio brasileiro de banco de dados</td>
</tr>
<tr>
<td></td>
<td>R. Miranda, M. Estolano, F Neto</td>
<td>A raster approximation for processing of polyline joins</td>
<td>xviii simposio brasileiro de banco de dados</td>
</tr>
<tr>
<td>Author</td>
<td>Coauthor</td>
<td>Work Title</td>
<td>Publication Venue Title</td>
</tr>
<tr>
<td>--------</td>
<td>----------</td>
<td>------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>G. Zimbrao, V. Monteiro, I. Azevedo</td>
<td>A multi-user key and data exchange protocol to manage a secure database</td>
<td>xv simposio brasileiro de banco de dados</td>
<td></td>
</tr>
<tr>
<td>G. Zimbrao, R. Miranda, M. Estolano, F Neto</td>
<td>A raster approximation for processing of polyline joins</td>
<td>xviii simposio brasileiro de banco de dados</td>
<td></td>
</tr>
<tr>
<td>Coauthor</td>
<td>Work Title</td>
<td>Publication Venue Title</td>
<td></td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------------------------------------------------</td>
<td>--------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>G. Zimbrao, V. Almeida</td>
<td>approximate spatial query processing using raster signatures</td>
<td>xv simposio brasileiro de banco de dados</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Author</th>
<th>Coauthor</th>
<th>Work Title</th>
<th>Publication Venue Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jano M. Souza</td>
<td>G. Zimbrao, R. Monteiro, I. Azevedo</td>
<td>A multi-user key and data exchange protocol to manage a secure database</td>
<td>xv simposio brasileiro de banco de dados</td>
</tr>
<tr>
<td></td>
<td>R. Miranda, M. Estolano, F Neto</td>
<td>A raster approximation for processing of polyline joins</td>
<td>xviii simposio brasileiro de banco de dados</td>
</tr>
</tbody>
</table>
INDi

Step 2
<table>
<thead>
<tr>
<th>Coauthors</th>
<th>Work Title</th>
<th>Publication Venue Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Steiner points in tree metrics don't (really) help.</td>
<td>SODA 2001</td>
</tr>
<tr>
<td>Coauthors</td>
<td>Work Title</td>
<td>Publication Venue Title</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>A. Gupta</td>
<td>-</td>
<td>SODA 2001</td>
</tr>
</tbody>
</table>

**Cluster Anupam Gupta**

<table>
<thead>
<tr>
<th>Coauthor</th>
<th>Work Title</th>
<th>Publication Venue Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chandra Chekuri, Ilan Newman, Yuri Rabinovich, Alistair Sinclair</td>
<td>Embedding k-outerplanar graphs into l1</td>
<td>SODA 2003</td>
</tr>
<tr>
<td>Coauthors</td>
<td>Work Title</td>
<td>Publication Venue Title</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>-</td>
<td>Steiner points in tree metrics don't (really) help.</td>
<td><strong>SODA 2001</strong></td>
</tr>
</tbody>
</table>

**Cluster Anupam Gupta**

<table>
<thead>
<tr>
<th>Coauthor</th>
<th>Work Title</th>
<th>Publication Venue Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chandra Chekuri, Ilan Newman, Yuri Rabinovich, Alistair Sinclair</td>
<td>Embedding k-outerplanar graphs into l1</td>
<td><strong>SODA 2003</strong></td>
</tr>
<tr>
<td>Coauthors</td>
<td>Work Title</td>
<td>Publication Venue Title</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>-</td>
<td>Steiner points in tree metrics don't (really) help.</td>
<td>SODA 2001</td>
</tr>
</tbody>
</table>

Cluster Anupam Gupta

<table>
<thead>
<tr>
<th>Coauthor</th>
<th>Work Title</th>
<th>Publication Venue Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chandra Chekuri, Ilan Newman, Yuri Rabinovich, Alistair Sinclair</td>
<td>Embedding k-outerplanar graphs into l1</td>
<td>SODA 2003</td>
</tr>
</tbody>
</table>
INDi

Step 3
<table>
<thead>
<tr>
<th>Coauthors</th>
<th>Work Title</th>
<th>Publication Venue Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chandra Chekuri, Ilan Newman,</td>
<td>Embedding k-outerplanar graphs into l1</td>
<td>SODA 2003</td>
</tr>
<tr>
<td>Yuri Rabinovich, Alistair</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sinclair</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Coauthor</th>
<th>Work Title</th>
<th>Publication Venue Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>Steiner points in tree metrics don't (really) help.</td>
<td>SODA 2001</td>
</tr>
</tbody>
</table>
INDi

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

Collections

184 distinct authors.

Synthetic Collections

• 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.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>INDi</th>
<th>HHC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\alpha_{\text{Title}}$</td>
<td>$\alpha_{\text{Venue}}$</td>
</tr>
<tr>
<td>BDBComp</td>
<td>0.0</td>
<td>0.2</td>
</tr>
<tr>
<td>Synthetic 5</td>
<td>0.1</td>
<td>1.0</td>
</tr>
<tr>
<td>Synthetic 10</td>
<td>0.1</td>
<td>0.9</td>
</tr>
</tbody>
</table>
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
Experimental Evaluation

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

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

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

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

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Load (with average number of references)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1(4920)</td>
</tr>
<tr>
<td>INDi</td>
<td>0.81 ± 0.16</td>
</tr>
<tr>
<td>HHC</td>
<td>11.50 ± 1.06</td>
</tr>
</tbody>
</table>

Both methods were implemented in Java.
INDi
Analysis of Cases of Failure

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>New Authors</th>
<th>Existing Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Correct</td>
<td>Incorrect</td>
</tr>
<tr>
<td>Sinthetic 5</td>
<td>41.803</td>
<td>58.197</td>
</tr>
<tr>
<td>Sinthetic 10</td>
<td>31.235</td>
<td>68.765</td>
</tr>
<tr>
<td>BDBComp</td>
<td><strong>99.507</strong></td>
<td><strong>0.493</strong></td>
</tr>
</tbody>
</table>
INDi

- Discussion
  - Using datasets generated by a synthetic data generator, INDi shows gains of up 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 $P_{a \text{Topic}}$ 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 $\beta_{\text{Topic}}$ (input parameter) are selected from each citation record of $a$.

• The venue popularity distribution of each topic $t$, $P_{t \text{Venue}}$
  – SyGAR combines the weights of $t$ associated with citation records containing the same publication venue
• 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 $P_{nCoauthors}^a$. Let it be $a_c$.
    3. Repeat $a_c$ times:
      – with probability $1 - \alpha_{NewCoauthor}$, select one coauthor according to $P_{Coauthor}^a$;
      – 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 $P_{all\ Topic}^a$. 
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 $P_{allTopic}$ 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 $\%_{\text{InheritedTopics}}$ 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 $l_{\text{Topic}}$ 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 ($N_{\text{Topics}} = 300$). Statistical ties are in bold.

<table>
<thead>
<tr>
<th>Collection</th>
<th>KWAY</th>
<th>SVM</th>
<th>HHC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>0.530±0.009</td>
<td>0.764±0.005</td>
<td><strong>0.770±0.006</strong></td>
</tr>
<tr>
<td>Synthetic 1</td>
<td>0.478±0.005</td>
<td>0.698±0.008</td>
<td><strong>0.753±0.013</strong></td>
</tr>
<tr>
<td>Synthetic 2</td>
<td>0.484±0.007</td>
<td>0.706±0.005</td>
<td>0.750±0.011</td>
</tr>
<tr>
<td>Synthetic 3</td>
<td>0.478±0.008</td>
<td>0.701±0.006</td>
<td>0.752±0.005</td>
</tr>
<tr>
<td>Synthetic 4</td>
<td>0.480±0.006</td>
<td>0.708±0.007</td>
<td>0.755±0.006</td>
</tr>
<tr>
<td>Synthetic 5</td>
<td>0.477±0.009</td>
<td>0.702±0.006</td>
<td>0.751±0.011</td>
</tr>
</tbody>
</table>
SyGAR

• Validation

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

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

($\%_{\text{InheritedTopics}} = 80\%$)

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

(b) $\%_{\text{NewAuthors}} = 10\%$
Evaluating Disambiguation Methods

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


References


References

Acknowledgements
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