A Unified Probabilistic Framework for Name Disambiguation in Digital Library

Jie Tang, A.C.M. Fong, Bo Wang, and Jing Zhang

Abstract—Despite years of research, the name ambiguity problem remains largely unresolved. Outstanding issues include how to capture all information for name disambiguation in a unified approach, and how to determine the number of people K in the disambiguation process. In this paper, we formalize the problem in a unified probabilistic framework, which incorporates both attributes and relationships. Specifically, we define a disambiguation objective function for the problem and propose a two-step parameter estimation algorithm. We also investigate a dynamic approach for estimating the number of people K. Experiments show that our proposed framework significantly outperforms four baseline methods of using clustering algorithms and two other previous methods. Experiments also indicate that the number K automatically found by our method is close to the actual number.

Index Terms—Digital libraries, information search and retrieval, database applications, heterogeneous databases.

1 Introduction

Different people may share identical names in the real world. It is estimated that the 300 most common male names are used by more than 114 million people (taking about 78.74 percent) in the United States (http://names.mongabay.com/male_names.htm). In many applications such as scientific literature management and information integration, the people names are used as the identifier to retrieve the information. Name ambiguity will greatly hurt the quality of the retrieved information.

To underline the seriousness of the problem, we have examined 100 person names in the publication data and found, for example, there are 54 papers authored by 25 different "Jing Zhang" in the DBLP database. Also, three students named "Yi Li" have graduated from the first author's lab.

1.1 Motivation

We begin by illustrating the problem with an example drawn from a real-world system (http://arnetminer.org) [40]. In this system, we try to extract researcher profiles from the web and integrate the publication data from online databases such as DBLP, ACM Digital Library, CiteSeer, and SCI. In the integration, we inevitably have the name ambiguity problem. Fig. 1 shows a simplified example. In Fig. 1, each node denotes a paper (with title omitted). Each directed edge denotes a relationship between two papers with a label representing the type of the relationship (cf. Section 2.1 for definitions of the relationship types). The distance between two nodes denotes the similarity of the two papers in terms of some content-based similarity measurement (e.g., cosine similarity). The solid polygon outlines the ideal disambiguation results, which indicate that 11 papers should be assigned to three different authors. An immediate observation from Fig. 1 is that a method based on only content similarity (the distance) would be difficult to achieve satisfactory performance, and that different types of relationships can be helpful, but with different degrees of contribution. For example, there is a CoAuthor relationship between nodes #3 and #8. Although the similarity between the two nodes is not high, benefiting from the CoAuthor relationship, we can still assign the two nodes (papers) to the same author. On the contrary, although there is a Citation relationship between nodes #3 and #7, the two papers are assigned to two different authors. Thus, one challenge here is how to design an algorithm for the name disambiguation problem by considering both attribute information of the node and the relationships between nodes.

1.2 Prior Work

The problem has been independently investigated in different domains, and is also known as entity resolution [4], [5], [7], web appearance disambiguation [3], [20], name identification [26], and Object distinction [49]. Despite many approaches proposed, the name ambiguity problem remains largely unresolved.

In general, existing methods for name disambiguation mainly fall into three categories: *supervised based*, *unsupervised based*, and *constraint based*. The supervised-based approach (e.g., [17]) tries to learn a specific classification model for each author name from the human-labeled training data. Then, the learned model is used to predict the author assignment of each paper. In the unsupervised-based approach (e.g., [18], [36], [37], [49]), clustering algorithms or topic models are employed to find paper

- *J. Tang and J. Zhang are with the Department of Computer Science and Technology, Tsinghua University, Rm 1-308, FIT Building, Beijing 100084, China. E-mail: jietang@tsinghua.edu.cn, zhangjing0544@gmail.com.*
- *A.C.M. Fong is with the School of Computing and Mathematical Sciences, Auckland University of Technology, AUT Tower Level 1, 2-14 Wakefield Street, Auckland 1142, New Zealand. E-mail: afong@aut.ac.nz.*
- *B. Wang is with the Department of Computer Science, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China. E-mail: bowang@nuaa.edu.cn.*

Manuscript received 1 July 2008; revised 5 Apr. 2010; accepted 16 Nov. 2010; published online 27 Dec. 2010.
Recommended for acceptance by B.C. Ooi.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2008-07-0335.
Digital Object Identifier no. 10.1109/TKDE.2011.13.

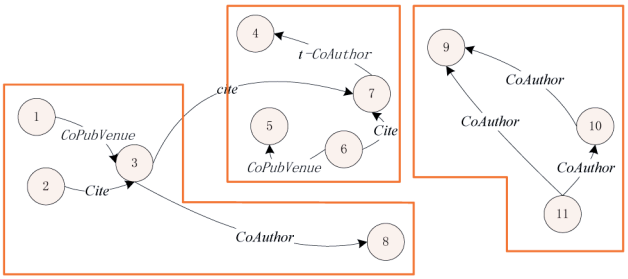


Fig. 1. An example of name disambiguation.

partitions, and papers in different partitions are assigned to different authors. The constraint-based approach also utilizes the clustering algorithms. The difference is that user-provided constraints are used to guide the clustering algorithm toward better data partitioning (e.g., [2], [51]).

Furthermore, several other approaches based on rules, citation/author graphs, and combinations of the different approaches have been studied. For example, Whang et al. [47] introduce a negative rules-based approach to remove the inconsistencies in the databases and develop two algorithms to identify important properties to create the rules. Davis et al. [11] have developed an interactive system which permits a user to locate the occurrences of named entities within a given text. The system is to identify references to a single art object (e.g., a particular building) in text related to images of that object in a digital collection. McRae-Spencer and Shadbolt [28] present a graph-based approach to author disambiguation on large-scale citation networks by using self-citation, coauthor relationships. The approach can achieve a high precision but a relatively low recall. Yu et al. [50] have developed supervised approaches to identify the full forms of ambiguous abbreviations within the context they appear. More recently, Chen et al. [8] study how to combine the different disambiguation approaches and propose an entity resolution ensemble framework, which combines the results of multiple base-level entity resolution systems into a single solution to improve the accuracy of entity resolution. Whang et al. [46] propose an iterative blocking framework where the resolution results of blocks are reflected to subsequently processed blocks. On and Lee [32] study the scalability issue of the name disambiguation problem. Although much progress has been made, existing methods do not achieve satisfactory disambiguation results due to their limitations:

1. Some existing graph clustering methods (e.g., [31], [35], [48]) focus on partitioning the data graph based on the topological structure; some other methods (e.g., [18], [42]) aim to cluster the data graph according to node similarity. A few researchers (e.g., [38], [52]) try to combine the two pieces of information. For example, Zhou et al. attempt to combine information based on both vertex attributes (i.e., node similarity) and graph topological structure by first constructing an attribute augmented graph through explicit assignments of ⟨attribute, value⟩ pairs to vertices, and subsequently estimating the pairwise vertices' closeness using a neighborhood random walk model. The pairwise comparisons mean that they subsequently discard topological information. Although the authors were able to demonstrate that attribute similarity increases the closeness of pairwise vertices in their distance measure, how to optimally balance the contributions of the different information is still an open problem They are only able to conclude that adding attribute similarity information to the clustering objective will not degrade the intracluster closeness. Further, in [52], the experimental data sets contain very few attributes. The first data set (political blogs) only has one (binary) attribute and the second data set of DBLP bibliographical data only has two attributes. We argue that much richer node attribute information is required for tackling the name disambiguation problem effectively.
2. The performance of all the aforementioned methods depends on accurately estimating K. Although several clustering algorithm such as X-means [33] can automatically find the number K based on some splitting criterion, it is unclear whether such a method can be directly applied to the name disambiguation problem.
3. In exiting methods, the data usually only contain homogeneous nodes and relationships; while in our problem setting, there may be multiple different relationships (e.g., CoAuthor and Citation) between nodes. The types of different relationships may have different importance for the name disambiguation problem. How to automatically model the degree of contributions of different relationships is still a challenging problem.

1.3 Our Solution

Having conducted a thorough investigation, we propose a unified probabilistic framework to address the above challenges. Specifically, we formalize the disambiguation problem using a Markov Random Fields (MRF) [16], [24], in which the data are cohesive on both local attributes and relationships. We explore a dynamic approach for estimating the number of people K and a two-step algorithm for parameter estimation. The proposed approach can achieve better performance in name disambiguation than existing methods because the approach takes advantage of interdependencies between paper assignments. To the best of our knowledge, our work is the first to formalize all the problems for name disambiguation in a unified framework and tackle the problems together.

The proposed framework is quite general. One can incorporate any relational features or local features into the framework, e.g., a feature based on the web search engine used. The framework can be also extended to deal with many other problems such as entity resolution in a relational database [4].

Our contributions in this paper include: 1) formalization of the name disambiguation problem in a unified probabilistic framework; 2) proposal of an algorithm to solve the parameter estimation in the framework; and 3) an empirical verification of the effectiveness of the proposed framework.

TABLE 1
Attributes of Each Publication p_i

Attribute	Description
$p_i.title$	title of p_i
$p_i.pubvenue$	published conference/journal of p_i
$p_i.year$	published year of p_i
$p_i.abstract$	abstract of p_i
$p_i.authors$	authors name set of p_i $\{a_i^{(0)},a_i^{(1)},\ldots a_i^{(u)}\}$
$p_i.references$	references of p_i

2 Problem Formalization

2.1 Definitions

In the discussion that follows, we assign six attributes to each paper p_i as shown in Table 1. Such publication data can be extracted from sources such as DBLP, Libra.msra.cn, Arnetminer.org, and Citeseer.ist.psu.edu.

Definition 1 (Principle Author and Secondary Author). *Each paper p_i has one or more authors $\mathrm{A_{pi}} = \{\mathrm{a_i^{(0)}}, \mathrm{a_i^{(1)}}, \ldots \mathrm{a_i^{(u)}}\}$. We describe the author name that we are going to disambiguate as the principle author $\mathrm{a_i^{(0)}}$ and the rest (if any) as secondary authors.*

We define five types of undirected relationships between papers (Table 2). Specifically,

- CoPubVenue (r_1) represents two papers published at the same venue. For example, if both papers are published at "KDD," we create an undirected CoPubVenue relationship between the two papers. Intuitively, two researchers with the same name may work in different research fields, thus would publish papers at different venues.
- CoAuthor (r_2) represents that two papers p_1 and p_2 have a secondary author with the same name, i.e., $A'_{p1} \cap A'_{p2} \neq \emptyset$, where A'_{p1} denotes the set of authors of paper p_1 excluding the principle author $a_i^{(0)}$, i.e., $A'_{p1} = A_{p1} \setminus a_i^{(0)}$. Typically, two papers that have many common coauthors would belong to the same person.
- Citation (r_3) represents one paper citing another paper. It is likely that an author cites his own previous work. Further, we incorporate latent citation information as follows: If paper p_1 cites papers $p_2, p_3, \ldots, p_{\mathrm{n}}$, then we establish undirected pairwise relationships among all cited papers, in addition to directed pairwise relationships between p_1 and the cited papers.
- Constraint (r_4) denotes constraints supplied via user feedback. For instance, the user can specify that two papers should be disambiguated to the same person or should belong to different persons.
- τ-CoAuthor (r_5) represents τ-extension CoAuthor relationship. We use an example to explain this relationship. Suppose paper p_i has authors "David Mitchell" and "Andrew Mark," and p_j has authors "David Mitchell" and "Fernando Mulford." We are going to disambiguate "David Mitchell." And if "Andrew Mark" and "Fernando Mulford" also coauthor another paper, then we say p_i and p_j have a 2-CoAuthor relationship.

TABLE 2
Relationships between Papers

R	W	Relation Name	Description
r_1	w_1	CoPubVenue	$p_i.pubvenue = p_j.pubvenue$
r_2	w_2	CoAuthor	$\exists r, s>0$, $a_i^{(r)}=a_j^{(s)}$
r_3	w_3	Citation	p_i cites p_j or p_j cites p_i
r_4	w_4	Constraint	feedback supplied by users
r_5	w_5	τ-CoAuthor	τ-extension co-authorship (τ>1)

To make it clear, we explain further about how to determine whether two papers have a τ-CoAuthor relationship. From the entire paper data set, we can construct a coauthor network, where each node denotes an author name and each edge denotes a coauthor relationship. For any two papers p_1 and p_2, we can obtain their corresponding sets A'_{p1} and A'_{p2} by their coauthors. If and only if $A'_{p1} \cap A'_{p2} \neq \emptyset$, we say the two papers have a CoAuthor relationship. For determining a 2-extension CoAuthor relationship, we construct two coauthor sets A^2_{p1} and A^2_{p2} according to the coauthor network. Specifically, A^2_{p1} is the set of authors by extending A'_{p1} with all neighbors of the authors in A'_{p1}, i.e., $A^2_{p1} = A'_{p1} \cup \{NB(a)\}_{a \in A'_{p1}}$, where $NB(a)$ is the set of neighbors of node a. Then, we say the two papers p_1 and p_2 have a 2-CoAuthor relationship, if and only if $A^2_{p1} \cap A^2_{p2} \neq \emptyset$. For determining whether two papers have a 3-extension CoAuthor relationship, we further extend A^2_{p1} to find an author set A^3_p for each paper and if the two sets have an intersection, we say the two papers have a 3- CoAuthor relationship. The weight of each type of relationship r_i is denoted by w_i. Estimation of the value of different weights will be described in Section 4.

In the name disambiguation problem, some papers may easily be clustered together or may be assigned together by the user. These papers will not be partitioned in the disambiguation algorithm. We describe such group of papers as *cluster atom*.

Definition 2 (Cluster Atom). *A cluster atom is a cluster in which papers are closely connected (e.g., the similarity $K(x_i, x_j) > threshold$). Papers with similarity less than the threshold will be assigned to disjoint cluster atoms.*

Finding cluster atoms would be greatly helpful to name disambiguation. For example, we can take the cluster atoms as the initialization of the disambiguation algorithm. For finding the cluster atoms, one can use a constrained-based clustering algorithm or simply use some constraints. In addition, we define the concept of *cluster centroid*. Derived from the clustering analysis, there are typically two methods to find the centroid of a cluster, the data point that is nearest to the center of the cluster or the centroid that is calculated as the arithmetic mean of all data points assigned to the cluster.

2.2 Name Disambiguation

Given a person name a, we denote publications containing the author name a as $P = \{p_1, p_2, \ldots, p_n\}$. The publication data with relationships can be modeled by networks comprising nodes and edges. We use an adaptive version

of the so-called informative graph [13] to represent the publication data. Publications and relationships are transformed into an undirected graph, in which each node represents a paper and each edge a relationship. Attributes of a paper are attached to the corresponding node as a feature vector. For the vector, we use words (after stop words filtering and stemming) in the attributes of a paper as features and use the number of their occurrences as the values. Formally, we can define the publication informative graph as follows:

Definition 3 (Publication Informative Graph). *Given a set of papers $P = \{p_1, p_2, \ldots, p_n\}$, let $r_k(p_i, p_j)$ be a relationship r_k between p_i and p_j. A publication informative graph is a graph $G = (P, R, V_P, W_R)$, where each $v(p_i) \in V_P$ corresponds to the feature vector of paper p_i and $w_k \in W_R$ denotes the weight of relationship r_k. Let $r_k(p_i, p_j) = 1$ iff there is a relationship r_k between p_i and p_j; otherwise, $r_k(p_i, p_j) = 0$.*

Suppose there are K persons $\{y_1, \ldots, y_K\}$ with the name a, our task is to disambiguate the n publications to their real researcher $y_i, i \in [1, K]$. More specifically, the major tasks of name disambiguation can be defined as:

1. Formalizing the disambiguation problem. The formalization needs to consider both local attribute features associated with each paper and relationships between papers.
2. Solving the problem in a principled approach. Based on the formalization, propose a principled approach and solve it in an efficient way.
3. Determining the number of people K. Given a disambiguation task (without any prior information), determine the actual K.

It is nontrivial to perform these tasks. First, it is not immediately clear how to formalize the entire disambiguation problem in a unified framework. Second, some graph models, e.g., Markov Random Field [16], are usually applied to model relational data. However, in the publication informative graph, the papers might be arbitrarily connected by different types of relationships. It is unclear how to perform inference (or parameter estimation) in such a graph with arbitrary structure. In addition, estimating the number of people K is also a challenging task.

3 Our Framework

3.1 Basic Idea

We have two basic observations for the name disambiguation problem: 1) papers with similar content tend to have the same label (belonging to the same author); and 2) papers having strong relationship tend to have the same labels, for example, two papers with coauthors who also author many other papers. An ideal solution is to disambiguate the papers by leveraging both content similarity and paper relationships. This is a nontrivial problem, because most existing clustering methods cannot well balance the two pieces of information.

In this paper, we propose a unified framework based on Markov Random Fields [16], [24]. More accurately, we formalize both content-based information and structure-based information into a Hidden Markov Random Field (HMRF) model as feature functions. The contribution degrees of the two types of information are formalized as weights of the feature functions. The importance of different types of relationships is also modeled as weights of corresponding feature functions. Solving the HMRF model includes both estimating the weights of feature functions and assigning papers to different persons. Such a framework also offers two additional advantages: first, it supports unsupervised learning, supervised learning, and semi-supervised learning. In this paper, we will focus on unsupervised learning for name disambiguation, but it is easy to incorporate some prior/supervised information into the model. Second, it is natural to do model selection in the HMRF model. The objective function in the HMRF model is a posterior probability distribution of hidden variables given observations, which is a criterion for model selection as well.

3.2 Hidden Markov Random Fields

A Markov Random Field is a conditional probability distribution of labels (hidden variables) that obeys the Markov property [16]. Many special cases of MRF can be developed. A Hidden Markov Random Fields is a member of the family of MRFs and its concept is derived from Hidden Markov Models (HMM) [15]. A HMRF is mainly composed of three components: an observable set of random variables $X = \{x_i\}_{i=1}^n$, a hidden field of random variables $Y = \{y_i\}_{i=1}^n$, and neighborhoods between each pair of variables in the hidden field.

We formalize the disambiguation problem as that of grouping relational papers into different clusters. Let the hidden variables Y be the cluster labels on the papers. Every hidden variable y_i takes a value from the set $\{1, \ldots, K\}$, which are the indexes of the clusters. The observation variables X correspond to papers, where every random variable x_i is generated from a conditional probability distribution $P(x_i|y_i)$ determined by the corresponding hidden variable y_i. Further, the random variables X are assumed to be generated conditionally independently from the hidden variables Y, i.e.,

$$P(X|Y) = \prod_{x_i \in X} P(x_i|y_i). \quad (1)$$

Fig. 2 shows the graphical structure of the HMRF for the example in Fig. 1. We see that dependent edges are provided between the hidden variables corresponding to the relationships in Fig. 1. The value of each hidden variable (e.g., $y_1 = 1$) denotes the assignment result. We do not model the indirect relationships between neighbors, but the model can propagate the dependencies along with the relationship.

As HMRF is a special case of MRF, the probability distribution of the hidden variables obeys the Markov property. Thus, the probability distribution of the value of y_i for the observation variable x_i depends only on the cluster labels of observations that have relations with x_i [24]. By the fundamental theorem of random fields [16], the probability distribution of the label configuration Y has the form

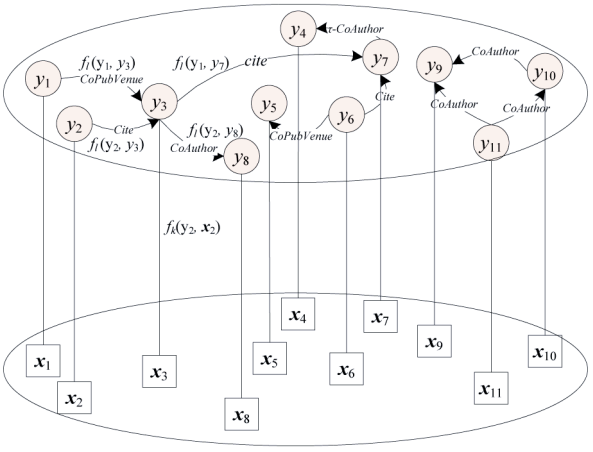


Fig. 2. Graphical representation of the HMRF model. $f(y_i, y_j)$ and $f(y_i, \mathbf{x}_i)$ are edge feature and node feature, respectively, and will be described in the next section.

$$P(Y) = \frac{1}{Z_1}\exp\left(\sum_{(y_i,y_j)\in E,k} \boldsymbol{\lambda}_k f_k(y_i, y_j)\right),$$
$$Z_1 = \sum_{y_i,y_j}\sum_{(y_i,y_j)\in E,k} \boldsymbol{\lambda}_k f_k(y_i, y_j) \tag{2}$$

and by further restricting the publication data being generated under the spherical Gaussian distribution, we have

$$P(X|Y) = \frac{1}{Z_2}\exp\left(\sum_{x_i\in X,l} \boldsymbol{\alpha}_l f_l(y_i, x_i)\right),$$
$$Z_2 = \sum_{y_i}\sum_{x_i\in X,l} \boldsymbol{\alpha}_l f_l(y_i, x_i), \tag{3}$$

where $f_k(y_i, y_j)$ is a nonnegative potential function (also called the feature function) defined on edge (y_i, y_j) and E represents all edges in the graph; $f_l(y_i, x_i)$ is a potential function defined on node x_i; λ_k and α_l are weights of the edge feature function and the node feature function, respectively; Z_1 and Z_2 are normalization factors.

To facilitate further discussion, we hereafter use X to denote the publication set P and use x_i to denote the vector $v(p_i)$ of the paper p_i.

3.3 Disambiguation Objective Function

We define an objective function as the Maximum a-Posteriori configuration of the HMRF, i.e., by maximizing $P(Y|X)$. $P(X)$ is usually taken as constant. Therefore, according to the Bayes rule $P(Y|X) \propto P(Y)P(X|Y)$, our objective function can be defined as

$$L_{\max} = \log(P(Y|X)) = \log(P(Y)P(X|Y)). \tag{4}$$

By substituting (2) and (3) into (4), we obtain

$$L_{\max} = \log\left(\frac{1}{Z_1 Z_2}\exp\left(\sum_{(y_i,y_j)\in E,k} \boldsymbol{\lambda}_k f_k(y_i, y_j) + \sum_{x_i\in X,l} \boldsymbol{\alpha}_l f_l(y_i, x_i)\right)\right). \tag{5}$$

Essentially, in the above objective function, we use the two kinds of feature functions, node feature function $f_l(y_i, x_i)$ and edge feature function $f_k(y_i, y_j)$, to represent the attribute information associated with each paper and the relationship information between papers, respectively.

The edge feature function $f_k(y_i, y_j)$ is used to characterize the relationship between two papers. Intuitively, if two papers have a strong relationship and also are similar to each other, then it is very likely that the two papers will be assigned to the same cluster. Specifically, the edge feature function must capture the paper relationships such as CoPubVenue and CoAuthor (as shown in Table 2) and a measure of similarity. Thus, we define the edge feature function as

$$f_k(y_i, y_j) = K(x_i, x_j)\sum_{r_m\in R_{ij}} [w_m r_m(x_i, x_j)]. \tag{6}$$

Here, $K(x_i, x_j)$ is a similarity function between papers x_i and x_j; w_m is the weight of relationship r_m; R_{ij} denotes the set of relationships between x_i and x_j; and $r(x_i, x_j)$ denotes a function of the relationship between x_i and x_j. The simplest way to define the relation function $r(x_i, x_j)$ is to define it with binary values as described in Definition 3. Here, we further consider a definition which combines the time information, i.e., $r_1(x_i, x_j) = exp\{-|x_i.year - x_j.year|\}$. This definition is derived from an observation on the name ambiguity problem: the CoAuthor and CoPubVenue relations are often time-dependent, e.g., authors tend to publish at several focused conferences/journal intensively in a specific period and coauthors also tend to collaborate with each other in a specific period.

The node feature function $f_l(y_i, x_i)$ mainly captures the attribute information associated with paper x_i. The basic idea here is if the paper is similar to all the other papers in a cluster, then it is very likely that the paper will be assigned to the cluster. For this cluster assignment operation, we define the node feature function as

$$f_l(y_i, x_i) = K(y_i, x_i) = K(\boldsymbol{\mu}_{(i)}, x_i), \tag{7}$$

where $\mu_{(i)}$ is the cluster centroid that the paper x_i is assigned to. Notation $K(x_i, \mu_{(i)})$ represents the similarity between paper x_i and its assigned cluster center $\mu_{(i)}$.

Then, putting (6) and (7) into (5), we obtain

$$L_{\max} = \sum_{(x_i,x_j)\in E,k} \lambda_k K(x_i, x_j) r_k(x_i, x_j) + \sum_{x_i\in X,l} \boldsymbol{\alpha}_l K(x_i, \boldsymbol{\mu}_{(i)}) - \log Z, \tag{8}$$

where $Z = Z_1 Z_2$. Without any loss of generality, we combine the weight of edge feature function λ_k and the weight of the relationship w_m, and write as λ for simplicity.

3.4 Criteria for Model Selection

We use Bayesian Information Criterion (BIC) as the criterion to estimate the number of people K. We define an objective function for the disambiguation task. Our goal is to optimize a parameter setting that maximizes the local objective function with some given K and find a number K that maximizes the global objective function.

Specifically, we first consider $K = 1$, that is, there is only one person with the given name a. Then, we use a measurement to determine whether the paper cluster should be split into two subclusters. Next, for each subcluster, we again use the measurement to determine whether to split. The operation repeats until some condition is satisfied (e.g., no subcluster can be split). In the process, we call M_h the model corresponding to the solution with the person number h. We therefore have a family of alternative models $\{M_h\}$, where h ranges from 1 to n, inclusively.

Now, our task is to choose the best model from $\{M_h\}$. Many measurements can be used for model selection, such as Silhouette Coefficient [23], Minimum Description Length (MDL) [34], Akaike Information Criterion (AIC) [1], and posterior probability estimation [22]. We chose BIC as the criterion, because BIC criterion is fundamentally similar to other criteria such as MDL and has a stronger penalty than the other criteria such as AIC, which is desirable in our problem. Based on these considerations, we use a variant of the BIC measurement [22] as the criterion

$$BIC^v(M_h) = \log(P(M_h|P)) - \frac{|\lambda|}{2} \cdot \log(n), \tag{9}$$

where $P(M_h|P)$ is the posterior probability of model M_h given the observations P.$|\lambda|$ is the number of parameters in M_h (which can be defined in different ways, e.g., the number of nonzero parameters in the model M_h or the sum of the probabilities of $P(Y)$). n is the paper number. The second part is a penalty to model complexity.

In essence, a BIC score approximates how appropriately the model M_h fits the whole data set. We use this criterion for the model selection because it can be easily extended to different situations. For example, conventional clustering algorithms like K-means [27] or X-means [33] regard the data as independent and thus the posterior probability $P(M_h|P)$ can be simplified to $P(P|M_h)$ according to the Bayesian rule $P(M_h|P) \propto P(P|M_h)P(M_h)$ by taking the prior $P(M_h)$ as uniform. However, we intend to take advantage of dependencies between the clustering results. Thus, viewing $P(M_h)$ as uniform is inappropriate. Our definition in (2) considers the dependencies using a Markov field.

4 Parameter Estimation

4.1 Algorithm

The parameter estimation problem is to determine the values of the parameters $\Theta = \{\lambda_1, \lambda_2, \ldots; \alpha_1, \alpha_2, \ldots\}$ and to determine assignments of all papers. More accurately, we optimize the log-likelihood objective function (8) with respect to a conditional model $P(Y|X, \Theta)$.

At a high level, the learning algorithm (cf. Algorithm 1) for parameter estimation primarily consists of two iterative steps: *Assignment* of papers, and *Update* of parameters Θ. The basic idea is that we first randomly choose a parameter setting Θ and select a centroid for each cluster. Next, we assign each paper to its closest cluster and then calculate the centroid of each paper-cluster based on the assignments. After that, we update the weight of each feature function by maximizing the objective function.

Algorithm 1. Parameter estimation

Input: $P=\{p_1, p_2, \ldots, p_n\}$

Output: model parameters Θ and $Y=\{y_1, y_2, \ldots, y_n\}$, where $y_i \in [1, K]$

1. Initialization

1.1 randomly initialize parameters Θ;

1.2 for each paper x_i, choose an initial value y_i, with $y_i \in [1, K]$;

1.3 calculate each paper cluster centroid $\mu_{(i)}$;

1.4 for each paper x_i and each relationship (x_i, x_j), calculate $f_i(y_i, x_i)$ and $f_k(y_i, y_j)$.

2. Assignment

2.1 assign each paper to its closest cluster centroid;

3. Update

3.1 update of each cluster centroid;

3.2 update of the weight for each feature function.

For initialization, we randomly assign the value of each parameter (λ and α). For initialization of the cluster centroid, we first use a graph clustering method to identify the cluster atoms. Basically, papers with similarity less than a threshold will be assigned to disjoint cluster atoms. We greedily assign papers in the described fashion by always choosing the paper that has the highest similarity to the cluster centroid u. In this way, we get γ cluster atoms. If γ is equal to the number of people K, then these γ groups are used as our initial assignment. If $\gamma < K$, we randomly choose another (K-γ) papers as the cluster centroids. If $\gamma > K$, we group the nearest cluster atoms until there are only K groups left. We now introduce in detail the two steps in our parameter estimation algorithm.

Assignments. In *Assignments*, each paper x_i is assigned to $\mu_{(h)}$ to maximize $\log P(y_i|x_i)$

$$\begin{aligned} \log P(y_i|x_i) &\propto L_{x_i}(\mu_{(h)}, x_i) \\ &= \sum_{(x_i,x_j)\in E_i, R_i, k} \lambda_k K(x_i, x_j) r_k(x_i, x_j) \\ &\quad + \sum_l \alpha_l K(x_i, \mu_{(h)}) - \log Z, \end{aligned} \tag{10}$$

here Z degrades to a normalization factor on x_i only and can be removed as we only care about the relative score for different assignment (for different y_i), E_i denotes all relationships related to x_i, and $K(x_i, x_j)$ denotes the similarity function defined using a cosine similarity measurement

$$K(x_i, x_j) = \frac{x_i^T x_j}{\|x_i\|\|x_j\|}, \text{ where } \|x_i\| = \sqrt{x_i^T x_i}. \tag{11}$$

The similarity function can be easily extended by using any kernel function (e.g., the radius kernel function), benefiting from the fact that there are only pairs of papers (x_i, x_j) and pairs of paper-cluster centroid (x_i, y_i) in our objective function.

With a kernel function, each paper is actually mapped into another new space, which may help disambiguate the papers in some special applications. We tried a few kernel

functions, e.g., sigmoid kernel and radius kernel. However, they are not very helpful in our current task.

Now, the task is to calculate all parametric terms in (10). The first two terms in (10) are a polynomial combination of the similarity function $K(x_i, \mu_{(_h)})$ and the relational similarity function $K(x_i, x_j)$, which can be calculated. However, it is intractable to obtain an exact solution of the partition function, i.e., $\log(Z)$, because the marginalization would take place within the logarithm ($Z = Z_1 Z_2$). A few algorithms have been proposed for approximate inference, e.g., belief propagation [30] and contrastive divergence (CD) [19]. We now examine how to approximate the partition function via contrastive divergence in our disambiguation objective function.

Based on Jensen's inequality [21], we can obtain an upper bound of the negative log-likelihood—$\log(L)$ with a Kullback-Liebler (KL) divergence

$$\begin{aligned} L^{KL} &= KL(q\|P) \\ &= \sum_{y_i} q(y_i|x_i)\log(q(y_i|x_i)) - \sum_{y_i} q(y_i|x_i)\log(P(y_i|x_i)) \\ &= -H(q) - \langle \log(P(y_i|x_i))\rangle_{q(y_i)}, \end{aligned} \tag{12}$$

where $q(y_i|x_i)$ is an approximation of the distribution $P(y_i|x_i)$. $\langle . \rangle_q$ is the expectation under the distribution q.

Maximizing the log-likelihood of the data (5) is equivalent to minimizing the KL divergence (12) between the data distribution q^0 and the equilibrium distribution over the visible variables, q^∞, where the first term can be calculated by the observations with their currently assigned labels and the second term the probability when we use model distribution with all possible labels. Again, the obvious difficulty in the above equation is how to derive the second term. A Markov chain Monte Carlo (MCMC) method can be used to estimate the approximation distribution $q^\infty(y_i|x_i)$ with the start point of MCMC being viewed as $q^0(y_i|x_i)$. To make the process more efficient, we can use the contrastive divergence algorithm [19], which approximates the distribution by several Gibbs sampling steps (or just one step). Thus, the objective function becomes

$$\begin{aligned} L^{KL} &= KL(q^0\|P) \approx KL(q^0\|P) - KL(q^l\|P) \\ &= \langle \log(P(y_i|x_i))\rangle_{q^0(y_i)} - \langle \log(q^l(y_i|x_i))\rangle_{q^l(y_i)}. \end{aligned} \tag{13}$$

In contrastive divergence learning, instead of minimizing $KL(q^0\|q^\infty)$, we minimize the difference between $KL(q^0\|q^l)$ and $KL(q^l\|q^\infty)$, where q^l is the distribution over the "l-step" reconstruction of the data vectors (i.e., observations) that are generated after l-step Gibbs sampling. As indicated in [19], the step l can be simply set as 1 in most cases. (That is, we can simply consider one Gibbs sampling iteration to minimize the $KL(q^0\|q^1)$). The procedure of reconstructing the data vector (i.e., q^1) from the distribution q^0 is described in Algorithm 2.

Algorithm 2: One-step sampling
Input: current observation x^0 and labels y^0
Output: sampling results of y^1 and x^1

1: Draw an observation x, from the distribution of $q^0(x_i)$ ($q(x)$ can be obtained by summing over all possible labels);
2: Compute $P(y_i|x)$, the posterior probability distribution over the label variable given the observation x;
3: Compute $P(y_i|y_{\neg i})$, the probability distribution over the label variable given labels of its neighboring observations;
4: Draw a new label y^1_i for each observation from the probability distribution $P(y_i|x)P(y_i|y_{\neg i})$;
5: Given the chosen label, compute the conditional distribution of $P(x_i|y_i)$;
6: Draw each feature of the new observation x^1_i from the conditional distribution $P(x_i|y_i)$.

Finally, based on the reconstructed data vector, we can calculate (13). The stochastic sampling sometimes is time demanding. To make it more efficient, one can use the deterministic mean field algorithm [44] to replace the sampling procedure.

After solving the third term in (10), we can compute the solution for the whole objective function. Finally, a greedy algorithm is used to sequentially update the assignment of each paper. An assignment of a paper is performed while keeping the other papers fixed. The process is repeated until no paper changes its assignment between two successive iterations.

Update. In Update, each cluster centroid is first updated by the arithmetic mean of the papers contained in it

$$\mu_{(h)} = \frac{\sum_{i:y_i=h} x_i}{\|\sum_{i:y_i=h} x_i\|_{\mathbf{A}}}. \tag{14}$$

Then, by differentiating the objective function with respect to each parameter λ_k, we have

$$\frac{\partial L}{\partial \lambda_k} = -\sum_{(x_i,x_j)\in E} K(x_i,x_j) r(x_i,x_j) - \frac{\partial \log Z}{\partial \lambda_k}. \tag{15}$$

We see that the second term is intractable, because calculation of Z needs to sum up all possibilities of assignments of each paper. Again, we start from the KL divergence objective function (13) and use the CD algorithm to calculate the derivatives of L^{KL} with respect to λ_k

$$\begin{aligned} \frac{\partial L^{KL}}{\partial \lambda_k} &= \left\langle \frac{\partial \log(P(y_i|x_i))}{\partial \lambda_k} \right\rangle_{q^0(y_i)} - \left\langle \frac{\partial \log(q(y_i|x_i))}{\partial \lambda_k} \right\rangle_{q^1(y_i)} \\ &= -\sum_{(x_i,x_j)\in E} K(x_i,x_j) r(x_i,x_j) - \left\langle \frac{\partial \log(q(y_i|x_i))}{\partial \boldsymbol{\lambda}_k} \right\rangle_{q^1(y_i)}. \end{aligned} \tag{16}$$

The first term is the polynomial combination of the similarity function and the second term can be calculated after the 1-step sampling (Algorithm 2).

Finally, each parameter is updated by

$$\lambda_k^{new} = \lambda_k^{old} + \Delta \frac{\partial L}{\partial \lambda_k}, \tag{17}$$

where Δ is the learning rate. We do the same for α.

4.2 Estimation of K

Our strategy for estimating K (see Algorithm 2) is to start by setting it as 1 and we then use the BIC score to measure whether to split the current cluster. The algorithm runs iteratively. In each iteration, we try to split every cluster C into two subclusters. We calculate a local BIC score of the new submodel M_2. If $\mathrm{BIC}(M_2) > \mathrm{BIC}(M_1)$, then we split the cluster. We calculate a global BIC score for the new model. The process continues by determining if it is possible to split further. Finally, the model with the highest global BIC score is chosen.

Algorithm 3. Estimation of K

Input: $P=\{p_1, p_2, \ldots, p_n\}$

Output: K, $Y=\{y_1, y_2, \ldots, y_n\}$, where $y_i \in [1, K]$

1: i=0, K=1, that is to view P as one cluster: $\mathbf{C}^{(i)}=\{C_1\}$;
2: do{
3: foreach cluster C in $\mathbf{C}^{(i)}$ {
4: find a best two sub-clusters model M_2 for C;
5: if(BIC(M_2)>BIC(M_1))
6: split cluster C into two sub clusters $\mathbf{C}^{(i+1)}=\{C_1, C_2\}$;
7: calculate BIC score for the obtained new model;
8: }while(existing split);
9: choose the model as output with the highest BIC score;

One difficulty in the algorithm might be how to find the best two subcluster models for the cluster C (Line 4). With different initialization, the resulting subclusters might be different. Fortunately, this problem is alleviated in our framework, benefiting from the cluster atoms identification. In disambiguation, a cluster can consist of several cluster atoms. To split further, we use the cluster atoms as initializing centroids and thus our algorithm tends to result in stable split results.

For the parameter $|\lambda|$ in (9), we simply define it as the sum of the K cluster probabilities, parameters, and cluster centroids, i.e.,

$$\sum_{i=1}^{K} (P(y_i) + \mu_{(i)}) + \sum_{\lambda \in \Theta} \lambda. \tag{18}$$

5 Experimental Results

5.1 Experimental Setting

Data Sets. We evaluated the proposed method in the context of ArnetMiner.org [40]. We created a data set, which includes 32 real author names and 2,074 papers. In these names, some names are only associated with a few persons, for example "Cheng Chang" is the name of three persons and "Wen Gao" four; while some names seem to be popular. For example, there are 25 persons with the name "Jing Zhang" and 40 persons named "Lei Wang." Statistics of this data set are shown in Table 3. Five PhD students from CS conducted manual disambiguation on all papers of the 32 author names. A spec was created to guide the annotation process. Each paper was labeled with a number indicating the actual person. The labeling work was carried out based on the publication lists on the authors' homepages and based on the affiliations, e-mail addresses in the web databases (e.g., ACM Digital Library). We calculated the Kappa coefficient for the annotated data. The average Kappa score is 0.82, which indicates a good agreement between the annotators. For disagreements in the annotation, we applied "majority voting." The data set will be online available.[1]

TABLE 3
Data Sets

Abbr. Name	#Publications	#Actual Person	Abbr. Name	#Publications	#Actual Person
Cheng Chang	12	3	Gang Wu	40	16
Wen Gao	286	4	Jing Zhang	54	25
Yi Li	42	21	Kuo Zhang	6	2
Jie Tang	21	2	Hui Fang	15	3
Bin Yu	66	12	Lei Wang	109	40
Rakesh Kumar	61	5	Michael Wagner	44	12
Bing Liu	130	11	Jim Smith	33	5
Ajay Gupta	27	4	Wei Wang	306	90
Dimitry Pavlov	16	2	David Jensen	43	3
Charles Smith	7	4	David Brown	53	7
David C. Wilson	52	5	George Miller	17	2
James H. Anderson	112	2	James Johnson	17	3
John Miller	74	2	Joseph Miller	10	2
Paul Jones	13	3	Richard Taylor	93	10
Robert Fisher	105	4	Robert Moore	92	3
Robert Williams	8	2	William Cohen	110	2

We also found that the disambiguation results are extremely unbalanced. For example, there are 286 papers authored by "Wen Gao" with 282 of them authored by Prof. Wen Gao from the Institute of Computing at Chinese Academy of Science and only four papers are authored by the other three persons named "Wen Gao."

We generated relationships between papers by string matching. For example, if both papers are published at SIGKDD, we created a CoPubVenue relationship between them. The conference full name (e.g., International Conference on Knowledge Discovery and Data Mining) and its acronym (e.g., SIGKDD) are considered as the same.

Experimental Design. We use PairewisePrecision, PairwiseRecall, and PairwiseF$_1$ score, to evaluate our method and to compare with previous methods. The pairwise measures are adapted for evaluating disambiguation by considering the number of pairs of papers assigned with the same label. Specifically, for any two papers annotated with the same label by the human annotator, we call it a correct pair. For two papers with the same label predicted by an approach, but do not have the same label in the human annotated data set, we call it a mistakenly predicted pair. Thus, we can define the measures as follows:

$$PairwisePrecision = \frac{\#PairsCorrectlyPredictedToSameAuthor}{\#TotalPairsPredictedToSameAuthor}$$

$$PairwiseRecall = \frac{\#PairsCorrectlyPredictedToSameAuthor}{\#TotalPairsToSameAuthor}$$

$$PairwiseF_1 = \frac{2 \times PairwisePrecision \times PairwiseRecall}{PairwisePrecision + PairwiseRecall}.$$

We considered several baseline methods based on K-means [27], SOM [43], and X-means [33]. The latter was used to find the number of people K. In these methods, we try to combine all the features defined in our method. Specifically, for title, we partition it into a bag of words and generate a

1. http://arnetminer.org/disambiguation.

TABLE 4
Results of Name Disambiguation (Percent)

Person Name	K-means			HAC			SOM			SACluster			CONSTRAINT			Our Approach (Fixed K)		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Cheng Chang	89.47	68.00	77.27	100.0	100.0	100.0	76.30	65.42	70.44	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Wen Gao	96.25	49.78	65.62	96.60	62.64	76.00	98.12	47.14	63.68	73.52	98.27	84.11	99.29	98.59	98.94	99.29	98.59	98.94
Yi Li	13.91	39.02	20.51	86.64	95.12	90.68	43.67	32.72	37.41	77.42	84.21	80.67	70.91	97.50	82.11	70.91	97.50	82.11
Jie Tang	95.38	72.09	82.12	100.0	100.0	100.0	84.92	70.65	77.13	90.14	82.04	85.90	100.0	100.0	100.0	100.0	100.0	100.0
Gang Wu	28.41	20.49	23.81	97.54	97.54	97.54	24.79	31.28	27.66	43.66	87.32	58.22	71.86	98.36	83.05	81.62	98.36	89.21
Jing Zhang	7.88	26.03	12.10	85.00	69.86	76.69	38.76	64.23	48.35	72.00	86.75	78.69	83.91	100.0	91.25	83.91	100.0	91.25
Kuo Zhang	60.00	60.00	60.00	100.0	100.0	100.0	82.50	70.20	75.85	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Hui Fang	60.87	90.32	72.73	100.0	100.0	100.0	40.60	80.60	54.00	92.21	54.20	68.27	100.0	100.0	100.0	100.0	100.0	100.0
Bin Yu	21.23	35.50	26.57	67.22	50.25	57.51	18.30	27.50	21.98	39.26	55.19	45.88	92.31	66.67	77.42	89.32	84.53	86.86
Lei Wang	11.98	21.87	15.48	68.45	41.12	51.38	21.52	57.34	31.29	44.40	75.59	55.94	91.58	92.59	92.08	88.64	89.06	88.85
Rakesh Kumar	68.82	91.28	78.47	63.36	92.41	75.18	62.83	90.17	74.06	80.98	82.43	81.70	92.37	99.18	95.65	99.14	96.91	98.01
Michael Wagner	57.66	52.32	54.86	18.35	60.26	28.13	52.18	46.39	49.11	42.20	64.04	50.87	26.25	77.78	39.25	85.19	76.16	80.42
Bing Liu	53.10	31.73	39.72	84.88	43.16	57.22	76.80	72.60	74.64	30.21	63.05	40.85	83.72	98.63	90.57	88.25	86.49	87.36
Jim Smith	62.59	44.16	51.78	92.43	86.80	89.53	43.10	40.50	41.76	83.14	80.87	81.99	70.91	97.50	82.11	95.81	93.56	94.67
Wei Wang	11.97	10.30	11.07	8.70	100.0	16.01	10.50	10.50	10.50	12.00	66.73	20.35	33.67	84.26	48.11	83.67	84.26	83.96
Ajay Gupta	67.33	58.62	62.67	41.88	100.0	59.04	61.82	43.59	51.13	51.16	77.65	61.68	90.67	96.55	93.52	97.67	96.55	97.11
Dimitry Pavlov	85.71	85.71	85.71	85.71	85.71	85.71	87.40	83.20	85.25	100.0	100.0	100.0	88.70	89.23	88.96	86.67	100.0	92.86
David Jensen	82.57	41.51	55.25	85.85	94.88	90.14	80.52	40.13	53.56	81.13	85.26	83.14	82.51	65.23	72.86	83.83	68.46	75.37
David Brown	63.84	78.64	70.47	35.89	100.0	52.82	59.21	36.34	45.04	42.29	86.39	56.78	50.23	75.23	60.24	89.32	91.45	90.37
David C. Wilson	65.50	21.58	32.46	85.54	99.79	92.12	49.53	23.12	31.52	100.0	100.0	100.0	75.12	60.45	66.99	94.33	67.30	78.55
George Miller	85.19	65.71	74.19	85.87	75.24	80.20	68.90	67.85	68.37	50.97	79.94	62.25	72.37	74.56	73.45	85.87	75.24	80.20
James H. Anderson	80.23	96.05	87.43	89.15	99.27	93.94	76.50	76.50	76.50	98.08	51.52	67.55	85.99	80.12	82.95	88.51	85.80	87.13
James Johnson	69.23	81.82	75.00	73.77	100.0	84.91	81.76	53.82	64.91	88.11	69.52	77.72	78.32	75.67	76.97	100.0	100.0	100.0
John Miller	69.99	96.81	81.24	69.35	90.75	78.62	72.83	68.51	70.60	77.36	63.08	69.49	72.65	79.07	75.72	83.38	97.73	89.99
Joseph Miller	57.14	72.73	64.00	54.55	54.55	54.55	49.32	67.18	56.88	61.29	44.19	51.35	55.21	59.34	57.20	86.55	74.55	80.10
Paul Jones	51.61	64.00	57.14	36.36	80.00	50.00	48.19	59.31	53.17	16.79	63.49	26.56	38.64	63.45	48.03	84.00	84.00	84.00
Richard Taylor	68.85	19.91	30.89	80.17	99.93	88.97	72.31	34.56	46.77	53.80	94.69	68.62	68.23	64.54	66.33	94.33	79.72	86.41
Robert Fisher	92.87	61.17	73.76	96.14	100.0	98.03	73.16	48.57	58.38	81.02	86.57	83.70	85.21	74.54	79.52	92.82	79.13	85.43
Robert Moore	92.10	66.01	76.90	86.90	93.10	89.89	80.60	48.33	60.43	100.0	100.0	100.0	89.91	78.54	83.84	84.04	75.66	79.63
Robert Williams	63.64	46.67	53.85	66.67	66.67	66.67	57.83	33.96	42.79	73.90	90.69	81.44	65.12	58.23	61.48	86.67	60.00	70.91
William Cohen	82.25	90.12	86.01	81.53	97.98	89.00	80.45	52.60	63.61	100.0	100.0	100.0	86.01	85.23	61.48	80.37	83.34	81.83
Charles Smith	50.00	33.00	39.76	30.00	100.0	46.15	57.92	62.15	59.96	44.42	74.46	55.65	45.27	67.89	85.62	100.0	100.0	100.0
Avg.	61.49	56.03	56.21	73.58	85.53	75.52	60.41	53.34	54.59	68.80	79.63	71.23	76.47	83.09	78.62	**90.13**	**88.26**	**88.80**

feature for each word; for conference, we define it as one feature and the value is the conference name; for author list, we treat them in the similar way as the title, that is, partition the author list into authors and define a feature for each author and the value is binary (indicating existence or not); while for citations, we also define multiple features and the value is set as the index of the cited paper. In addition, we considered two other baseline methods. The first one is based on hierarchical agglomerative clustering (HAC) on a list of citations and utilizes a search engine to help the disambiguation task [39], with the same feature definition as defined above. The other is based on SAClutering [52], which tries to partition the nodes in a graph into K clusters by using both structural and attributes information associated to each node. For fair comparison, in SACluster, we inputted the same attribute features defined in our approach and the same relationship information. The only difference is that SACluster does not differentiate the types of different relationships; thus, we simply consider all relationships as the same link in SACluster [52].

We further compared our method with two existing methods for name disambiguation: DISTINCT [49], a combination method based on two similarity measures: set resemble of neighbor tuples and random walk probability; CONSTRAINT [51], a constraint-based clustering algorithm for name disambiguation. For fair comparisons, 1) in all baseline methods and the compared methods, the number K for each author name is set as the actual person number; thus, the performance is the upper bound for the methods; and 2) we do not use user feedback (relationship r_4) in our experiments (as the baselines cannot use the user feedback).

5.2 Experimental Results

5.2.1 Results

We conducted disambiguation experiments for papers related to each of the author names in the data set. Table 4 shows the results. It can be seen that our method clearly outperforms the baseline methods for name disambiguation ($+32.77\%$ over K-Means, $+13.28\%$ over HAC, $+33.21\%$ over SOM, $+17.57$ over SACluster, and $+10.18\%$ over CONSTRAINT by average F_1 score).

The baseline methods suffer from two disadvantages: 1) they cannot take advantage of relationships between papers and 2) they rely on a fixed distance measure. Although SACluster considers the relationship between nodes, it incorporates the relationship information into a

TABLE 5
Results of Our Approach with Different Settings

Method	Precision	Recall	F1-Measure
Our Approach (Auto K)	83.01	79.54	80.05
Our Approach (w/o auto K)	90.13	88.26	88.80
Our Approach (w/o relation)	67.05	50.59	55.95

fixed distance function, thus cannot explicitly describe the correlation between the paper assignments. Our framework directly models the correlation as the dependencies between assignment results, and utilizes an unsupervised algorithm to learn the similarity function between papers. We conducted sign tests on the results. The p values are much smaller than 0.01, indicating that the improvements by our approach are statistically significant.

Table 6 lists the results of automatic estimation of the number K (the number in the round brackets is the actual number). We see that the estimated numbers by our approach are close to the actual numbers. Table 5 further lists the average results of our approach with different settings, where "w/o auto K" represents the result of our approach with a predefined cluster number K and "w/o relation" represents the result of our approach without using relationships (i.e., we set all edge feature function $f_k(y_i, y_j)$ to be zero). We see that the relationship is very important in our approach. Without the relationships, the performance of our approach drops sharply (-23.08 percent by F_1 score). This confirms that a model which cannot capture dependencies between papers would not result in good performance.

We applied X-means to find the number of people K. We assigned the minimum number as 1 and maximum number as n, the same setting as in our algorithm. We found that X-means fails to find the actual number. It always outputs only one cluster except "Yi Li" with 2. The reason might be that X-means cannot make use of the relationships between papers.

TABLE 6
Result of Automatically Discovered Person Number

Person Name	Actual Number	Auto Number	Person Name	Actual Number	Auto Number
Cheng Chang	3	3	Dimitry Pavlov	2	1
Wen Gao	4	5	David Jensen	3	6
Yi Li	21	13	David Brown	7	9
Jie Tang	2	2	David C. Wilson	5	5
Gang Wu	16	12	George Miller	2	6
Jing Zhang	25	16	James H. Anderson	2	7
Kuo Zhang	2	2	James Johnson	3	3
Hui Fang	3	3	John Miller	2	5
Bin Yu	12	10	Joseph Miller	2	3
Lei Wang	40	22	Paul Jones	3	5
Rakesh Kumar	5	5	Richard Taylor	10	14
Michael Wagner	10	11	Robert Fisher	4	7
Bing Liu	11	12	Robert Moore	3	6
Jim Smith	5	5	Robert Williams	2	5
Wei Wang	90	22	William Cohen	2	9
Ajay Gupta	4	6	Charles Smith	4	4

TABLE 7
Comparison with DISTINCT

Person Name	DISTINCT			Our Approach		
	Prec.	Rec.	F1	Prec.	Rec.	F1
Cheng Chang	55.07	44.19	49.03	100.00	100.00	100.00
Wen Gao	92.07	98.68	95.26	99.29	98.59	98.94
Jie Tang	79.36	93.37	85.80	100.00	100.00	100.00
Jing Zhang	100.00	75.56	86.08	83.91	100.00	91.25
Kuo Zhang	78.57	84.78	81.56	100.00	100.00	100.00
David Jensen	85.69	100.00	92.29	83.83	68.46	75.37
David Brown	69.77	74.99	72.29	89.32	91.45	90.37
David C. Wilson	87.10	90.00	88.53	94.33	67.30	78.55
Richard Taylor	68.35	63.11	65.63	94.33	79.72	86.41
Charles Smith	78.42	76.67	77.54	100.00	100.00	100.00
Hui Fang	88.60	95.00	91.69	100.00	100.00	100.00
Rakesh Kumar	92.90	96.80	94.81	99.14	96.91	98.01
Michael Wagner	72.30	75.40	73.82	85.69	82.31	83.97
Bing Liu	78.30	95.70	86.13	88.25	86.49	87.36
Jim Smith	86.30	90.40	88.30	96.37	93.80	95.07
Lei Wang	80.80	89.60	84.97	89.17	88.94	89.05
Bin Yu	68.90	77.80	73.08	95.27	72.63	82.42
Wei Wang	78.60	78.30	78.45	85.19	83.12	84.14
Ajay Gupta	98.70	92.30	95.39	97.67	96.55	97.11
Avg.	81.04	83.82	82.14	93.78	89.80	91.48

We compared our approach with DISTINCT [49]. We used person names that were used both in [49] and our experiments for comparisons. We conducted the experiments on our data set, which is a newer version of data used in [49]. For example, we have 109 papers for "Lei Wang" and 33 papers for "Jim Smith," while in [49] the numbers are 55 and 19. In addition, we do not consider the Proceeding Editor relation. Table 7 shows the comparison results. We see that on average our method clearly outperforms DISTINCT ($+8.34\%$ by F_1). Moreover, our approach has the advantage that it can automatically find the number K, whereas in DISTINCT the number needs to be supplied by the user. The relations used in DISTINCT and our approach are different. DISTINCT mainly considers the author-paper and paper-conference relations, and does not directly consider the CoAuthor and CoPubVenue relations, although the two relations can be derived from the paper-conference and author-paper relations.

5.2.2 Efficiency Performance

We evaluated the efficiency performance of our approach for the 32 author names on a desktop computer with Intel Core Duo processor (1.6 GHz). Table 8 lists the CPU time required for assigning the papers to different authors. We only list six authors who publish more than 100 papers and the average time for 100 random names. For most author names, all the algorithms use less than 1 second. The total running time of all algorithms is similar with each other.

TABLE 8
Comparison of Efficiency Performance (Seconds)

Person Name	*K*-means	X-Means	HAC	SACluster	DISTINCT	Our Approach
Wen Gao	4.8	5.1	12.9	30.4	56.0	20.3
Lei Wang	3.7	2.4	6.8	4.1	12.1	4.6
Bing Liu	1.6	1.9	4.2	5.4	1.1	5.8
Wei Wang	28.7	5.1	73.1	46.9	83.3	100.2
Robert Fisher	2.8	1.3	5.6	0.2	0.2	0.8
William Cohen	0.8	1.2	3.0	0.06	0.6	0.9
Average over 100	0.52	0.26	1.14	0.96	0.87	1.42

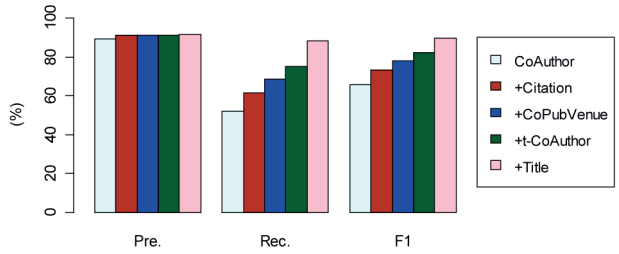


Fig. 3. Contribution of relationships.

5.2.3 Feature Contribution Analysis

We investigated the contribution of the defined features (including edge and node features) for name disambiguation. Specifically, we first rank the individual features by their performance, then add those features one by one in the order of their disambiguating power. In particular, we first use CoAuthor, followed by adding Citation, and then CoPubVenue, Paper Title. In each step, we evaluate the performance of our method. Fig. 3 shows the average Precision, average Recall, and average F1-score of our method with different feature combinations. At each step, we observed improvements. We can also see that most of the features (except CoAuthor) mainly contribute to the improvement of recall, while the improvement over precision is limited.

5.2.4 Distribution Analysis

We also perform a distribution analysis using a dimension reduction method [10]. We found that the feature distributions for all names can be typically categorized into the following scenarios: 1) publications of different persons are clearly separated ("Hui Fang"). Name disambiguation on this kind of data can be solved pretty well by our approach and the number K can also be found accurately; 2) publications are mixed together but with a dominant author who writes most of the papers (e.g., "Bing Liu"); our approach can achieve a F_1 score of 87.36 percent and the discovered number K is close to the actual number; and 3) publications of different authors are mixed (e.g., "Jing Zhang"). Our method can obtain a performance of 91.25 percent. However, it would be difficult to accurately find the number K. For example, the number found by our approach for "Jing Zhang" is 14, but the correct number should be 25. For a detailed analysis, please refer to [41].

5.2.5 Application Experiments

We applied the name disambiguation to help expert finding, which is to identify persons with some given expertise or experience. In particular, we evaluated expert finding with and without name disambiguation. Specifically, we selected 12 most frequent queries from the query log of ArnetMiner, and used a pooled relevance judgment [6] together with human judgments to create a data set for evaluation. Interested readers are referred to [51], [40] for details of the experimental setting. We conducted evaluation in terms of P@5, P@10, P@20, P@30, R-prec, mean average precision (*MAP*), *bpref*, and mean reciprocal rank (*MRR*). Fig. 4 shows the results of expert finding. In Fig. 4, EF represents expert finding using name disambiguation by our method and EF-NA represents expert finding without name disambiguation. We see that clear improvements can be obtained by using the proposed name disambiguation approach.

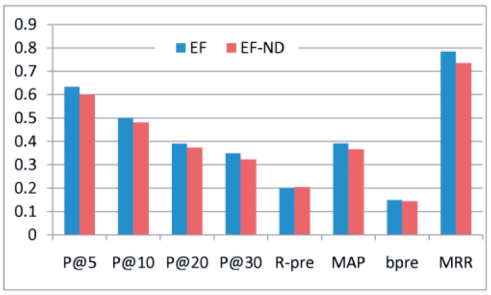


Fig. 4. Performances of expert finding.

5.3 Online System

To further demonstrate the effectiveness of the proposed approach, we have applied the disambiguation method in the Arnetminer system. Fig. 5 shows a snapshot of the disambiguation result. The user searches for "Jie Tang" and the system returns three different persons on the top of the page and below shows the detailed profile information of each person. The method runs in an offline mode and so far the system already generates the disambiguation results for more than 10,000 person names. Please note that this is an ongoing project. Visitors should expect the system to change.

6 Discussion

6.1 Connections with Previous Work

We analyze the connection of our framework with several previous works on disambiguation/clustering.

Connection with K-means: Our framework can describe relationships between data points whereas K-means [27] cannot. In essence, our framework uses edge potential functions to model the relationships. By removing the edge potential functions from (8), we have

$$L_{\max} = \sum_{x_i \in X, l} \alpha_l K(x_i, \mu_i) - \log Z. \tag{19}$$

By further removing the weight α_l for each similarity function, we obtain a naïve K-means clustering algorithm.

Connection with X-means: X-means [33] is proposed to dynamically find the clustering number K. It also employs BIC for model selection. However, as our model differs in nature from X-means, the selection process and the clustering algorithm are also different. The model selection

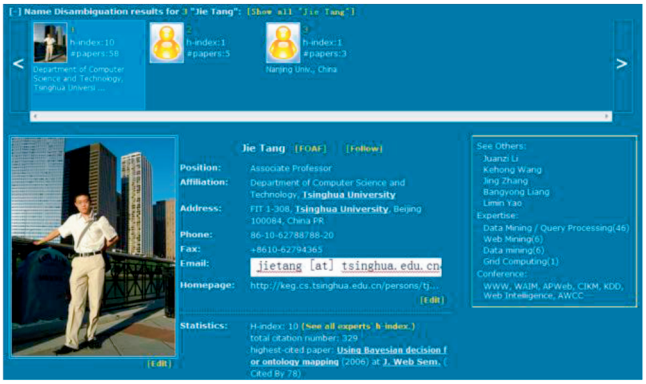


Fig. 5. Name disambiguation system (http://arnetminer.org).

method in our framework is similar to that in X-means if we consider the prior probability $P(Y)$ uniform, i.e., ignoring dependencies between data points. Except for model selection, X-means itself is similar to K-means.

Connection with the constraint-based disambiguation method: In constraint-based clustering, e.g., [2], the user can supply constraints to guide the clustering process. In our prior work, we have applied it to name disambiguation and obtained improvements [51], [41]. The usual constraints include must-link and cannot-link. Must-link means that two data points must be grouped into one cluster and cannot-link means two data points must be grouped into different clusters. We can adapt our framework as constraint-based clustering by redefining the edge potential function.

Connection with disambiguation using spectral graph clustering: Spectral graph clustering [12] aims at finding subgraphs with minimum cut of relationships between data points. K-way spectral graph clustering algorithm has been employed for name disambiguation [18]. We can give a penalty to the linked data pair if they were assigned to different clusters (i.e., $I(i \neq j)$) in the objective function. Then, our framework can adapt to this context by removing the second part from (8)

$$L_{\min} = -\sum_{(x_i,x_j)\in E,R,k} K(x_i,x_j) r_k(x_i,x_j) + \log Z. \qquad (20)$$

In essence, this new objective function means that we ignore the generative probabilities in the HMRFs and only focus on the dependencies between papers.

Comparing with the previous work, our framework offers several advantages: 1) In traditional methods, assignments of papers are independent, thus cannot take advantage of relationships between papers. 2) The proposed framework can be easily extended to semi-supervised learning by supplying user feedback. 3) Our framework can be viewed as a general framework of several unsupervised methods.

7 Conclusion and Future Work

In this paper, we have investigated the problem of name disambiguation. We have formalized the problems in a unified framework and proposed a generalized probabilistic model to the problem. We have defined a disambiguation objective function for the problem and have proposed a two-step parameter estimation algorithm. We have also explored a dynamic approach for estimating the number of people K. Experimental results indicate that the proposed method significantly outperforms the baseline methods. When applied to expert finding, clear improvement ($+2\%$) can be obtained.

As the next step, it would be interesting to investigate how to make use of the time information for name disambiguation, as the ambiguity problem evolves with the time. Moreover, it is also interesting to study how topic models like LDA can improve name disambiguation.

Acknowledgments

The authors would like to thank Hong Cheng for providing the source code of SACluster and Xiaoxin Yin for providing the source code of DISTINCT for the comparison experiments. They also thank Prof. Philip Yu for his valuable suggestions on this work. Jie Tang is supported by the Natural Science Foundation of China (No. 61073073), the Chinese National Key Foundation Research (No. 60933013, No.61035004), and a Special Fund for FSSP.

References

[1] H. Akaike, "A New Look at the Statistical Model Identification," *IEEE Trans. Automatic Control,* vol. AC-19, no. 6, pp. 716-723, Dec. 1974.

[2] S. Basu, M. Bilenko, and R.J. Mooney, "A Probabilistic Framework for Semi-Supervised Clustering," *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (SIGKDD '04),* pp. 59-68, 2004.

[3] R. Bekkerman and A. McCallum, "Disambiguating Web Appearances of People in a Social Network," *Proc. Int'l Conf. World Wide Web (WWW '05),* pp. 463-470, 2005.

[4] O. Benjelloun, H. Garcia-Molina, D. Menestrina, Q. Su, S.E. Whang, and J. Widom, "Swoosh: A Generic Approach to Entity Resolution," *The VLDB J.,* vol. 18, pp. 255-276, 2008.

[5] I. Bhattacharya and L. Getoor, "Collective Entity Resolution in Relational Data," *ACM Trans. Knowledge Discovery from Data,* vol. 1, article 5, 2007.

[6] C. Buckley and E.M. Voorhees, "Retrieval Evaluation with Incomplete Information," *Proc. Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '04),* pp. 25-32, 2004.

[7] Z. Chen, D.V. Kalashnikov, and S. Mehrotra, "Adaptive Graphical Approach to Entity Resolution," *Proc. Seventh ACM/IEEE-CS Joint Conf. Digital Libraries (JCDL '07),* pp. 204-213, 2007.

[8] Z. Chen, D.V. Kalashnikov, and S. Mehrotra, "Exploiting Context Analysis for Combining Multiple Entity Resolution Systems," *Proc. SIGMOD Int'l Conf. Management of Data (SIGMOD '09),* pp. 207-218, 2009.

[9] D. Cohn, R. Caruana, and A. McCallum, "Semi-supervised Clustering with User Feedback," Technical Report TR2003-1892, Cornell Univ., 2003.

[10] D. Cai, X. He, and J. Han, "Spectral Regression for Dimensionality Reduction," technical report, 2856, UIUC 2004.

[11] P.T. Davis, D.K. Elson, and J.L. Klavans, "Methods for Precise Named Entity Matching in Digital Collections," *Proc. ACM/IEEE-CS Joint Conf. Digital Libraries (JCDL '03),* p. 125, 2003.

[12] C. Ding, "A Tutorial on Spectral Clustering," *Proc. Int'l Conf. Machine Learning (ICML '04),* 2004.

[13] M. Ester, R. Ge, B.J. Gao, Z. Hu, and B. Ben-Moshe, "Joint Cluster Analysis of Attribute Data and Relationship Data: The Connected K-Center Problem," *Proc. SIAM Conf. Data Mining (SDM '06),* 2006.

[14] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. PAMI-6, no. 6, pp. 721-742, Nov. 1984.

[15] Z. Ghahramani and M.I. Jordan, "Factorial Hidden Markov Models," *Machine Learning,* vol. 29, pp. 245-273, 1997.

[16] J. Hammersley and P. Clifford, "Markov Fields on Finite Graphs and Lattices," Unpublished manuscript, 1971.

[17] H. Han, L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis, "Two Supervised Learning Approaches for Name Disambiguation in Author Citations," *Proc. ACM/IEEE Joint Conf. Digital Libraries (JCDL '04),* pp. 296-305, 2004.

[18] H. Han, H. Zha, and C.L. Giles, "Name Disambiguation in Author Citations Using a K-Way Spectral Clustering Method," *Proc. ACM/IEEE Joint Conf. Digital Libraries (JCDL '05),* pp. 334-343, 2005.

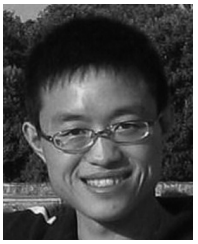
[19] G.E. Hinton, "Training Products of Experts by Minimizing Contrastive Divergence," *J. Neural Computation,* vol. 14, pp. 1771-1800, 2002.

[20] L. Jiang, J. Wang, N. An, S. Wang, J. Zhan, and L. Li., "GRAPE: A Graph-Based Framework for Disambiguating People Appearances in Web Search," *Proc. Int'l Conf. Data Mining (ICDM '09),* pp. 199-208, 2009.

[21] M.I. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul, "An Introduction to Variational Methods for Graphical Models," *Learning in Graphical Models,* vol. 37, pp. 105-161, 1999.

[22] R. Kass and L. Wasserman, "A Reference Bayesian Test for Nested Hypotheses and Its Relationship to the Schwarz Criterion," *J. Am. Statistical Assoc.,* vol. 90, pp. 773-795, 1995.

[23] L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis.* Wiley, 1990.
[24] R. Kindermann and J.L. Snell, *Markov Random Fields and Their Applications.* Amer Math. Soc., 1980.
[25] H. Kunsch, S. Geman, and A. Kehagias, "Hidden Markov Random Fields," *J. Annals of Applied Probability,* vol. 5, no. 3, pp. 577-602, 1995.
[26] X. Li, P. Morie, D. Roth, "Identification and Tracing of Ambiguous Names: Discriminative and Generative Approaches," *Proc. 19th Nat'l Conf. Artificial Intelligence (AAAI '04),* pp. 419-424, 2004.
[27] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," *Proc. Fifth Berkeley Symp. Math. Statistics and Probability,* 1967.
[28] D.M. McRae-Spencer and N.R. Shadbolt, "Also by the Same Author: AKTiveAuthor, a Citation Graph Approach to Name Disambiguation," *Proc. ACM/IEEE Joint Conf. Digital Libraries (JCDL '06),* pp. 53-54, 2006.
[29] E. Minkov, W.W. Cohen, and A.Y. Ng, "Contextual Search and Name Disambiguation in Email Using Graphs," *Proc. 29th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '06),* pp. 27-34, 2006.
[30] K.P. Murphy, Y. Weiss, and M.I. Jordan, "Loopy Belief Propagation for Approximate Inference: An Empirical Study," *Proc. Conf. Uncertainty in Artificial Intelligence (UAI '99),* pp. 467-475, 1999.
[31] M.E.J. Newman and M. Girvan, "Finding and Evaluating Community Structure in Networks," *Physical Rev. E,* vol. 69, p. 026113, 2004.
[32] B. On and D. Lee, "Scalable Name Disambiguation Using Multi-Level Graph Partition," *Proc. SIAM Int'l Conf. Data Mining (SDM '07),* 2007.
[33] D. Pelleg and A. Moore, "X-Means: Extending K-Means with Efficient Estimation of the Number of Clusters," *Proc. Int'l Conf. Machine Learning (ICML '00),* 2000.
[34] J. Rissanen, "A Universal Prior for Integers and Estimation by Minimum Description Length," *J. Annals of Statistics,* vol. 11, no. 2, pp. 416-431, 1983.
[35] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 888-905, Aug. 2000.
[36] L. Shu, B. Long, and W. Meng, "A Latent Topic Model for Complete Entity Resolution," *Proc. IEEE Int'l Conf. Data Eng. (ICDE '09),* pp. 880-891, 2009.
[37] Y. Song, J. Huang, I.G. Councill, J. Li, and C.L. Giles, "Efficient Topic-Based Unsupervised Name Disambiguation," *Proc. ACM/IEEE Joint Conf. Digital Libraries (JCDL '07),* pp. 342-351, 2007.
[38] Y. Sun, Y. Yu, and J. Han, "Ranking-Based Clustering of Heterogeneous Information Networks with Star Network Schema," *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (SIGKDD '09),* 2009.
[39] Y.F. Tan, M. Kan, and D. Lee, "Search Engine Driven Author Disambiguation," *Proc. ACM/IEEE Joint Conf. Digital Libraries (JCDL '06),* pp. 314-315, 2006.
[40] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "ArnetMiner: Extraction and Mining of Academic Social Networks," *Proc. 14th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (SIGKDD '08),* 2008.
[41] J. Tang, L. Yao, D. Zhang, and J. Zhang, "A Combination Approach to Web User Profiling," *ACM Trans. Knowledge Discovery from Data,* vol. 5, article 2, Dec. 2010.
[42] Y. Tian, R.A. Hankins, and J.M. Patel, "Efficient Aggregation for Graph Summarization," *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '08),* pp. 567-580, 2008.
[43] J. Vesanto and E. Alhoniemi, "Clustering of the Self-Organizing Map," *IEEE Trans. Neural Network,* vol. 11, no. 3, pp. 586-600, May 2000.
[44] M. Welling and G.E. Hinton, "A New Learning Algorithm for Mean Field Boltzmann Machines," *Proc. Int'l Conf. Artificial Neural Networks (ICANN '01),* pp. 351-357, 2001.
[45] M. Welling and K. Kurihara, "Bayesian K-Means as a "Maximization-Expectation" Algorithm," *Proc. SIAM Int'l Conf. Data Mining (SDM '06),* pp. 472-476, 2006.
[46] S.E. Whang, D. Menestrina, G. Koutrika, M. Theobald, and H. Garcia-Molina, "Entity Resolution with Iterative Blocking," *Proc. SIGMOD Int'l Conf. Management of Data (SIGMOD '09),* pp. 219-232, 2009.
[47] S.E. Whang, O. Benjelloun, and H. Garcia-Molina, "Generic Entity Resolution with Negative Rules," *The VLDB J.,* vol. 18, no. 6, pp. 1261-1277, 2009.
[48] X. Xu, N. Yuruk, Z. Feng, and T.A.J. Schweiger, "Scan: A Structural Clustering Algorithm for Networks," *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (SIGKDD '07),* pp. 824-833, 2007.
[49] X. Yin, J. Han, and P.S. Yu, "Object Distinction: Distinguishing Objects with Identical Names," *Proc. Int'l Conf. Data Eng. (ICDE '07),* pp. 1242-1246, 2007.
[50] H. Yu, W. Kim, V. Hatzivassiloglou, and J. Wilbur, "A Large Scale, Corpus-Based Approach for Automatically Disambiguating Biomedical Abbreviations," *ACM Trans. Information Systems,* vol. 24, no. 3, pp. 380-404, 2006.
[51] D. Zhang, J. Tang, J. Li, and K. Wang, "A Constraint-Based Probabilistic Framework for Name Disambiguation," *Proc. ACM Conf. Information and Knowledge Management (CIKM '07),* pp. 1019-1022, 2007.
[52] Y. Zhou, H. Cheng, and J.X. Yu, "Graph Clustering Based on Structural/Attribute Similarities," *Proc. VLDB Endowment,* vol. 2, no. 1, pp. 718-729, 2009.



Jie Tang is an associate professor at Tsinghua University. His research interests are social network analysis, data mining, and semantic web.

A.C.M. Fong is a professor in the School of Computing and Mathematical Sciences, Auckland University of Technology. He has published widely in the areas of data mining and communications.

Bo Wang is currently working toward the PhD degree from Nanjing University of Aeronautics and Astronautics. His research interests include transfer learning and information network analysis.

Jing Zhang received the MS degree from Tsinghua University in 2008. Her research interests include information retrieval and text mining.