An Efficient Approach for Finding the Essential Experts In Digital Library

A.Geethika (M.Tech), Chadalawada Ramanamma Engineering College,
Mr.J.Nagamuneiah, Associate Professor, Chadalawada Ramanamma Engineering College

Abstract— Name ambiguity is a special case of identity uncertainty where one person can be referenced by multiple name variations in different situations or even share the same name with other people. In this paper, we focus on Name Disambiguation problem. When non-unique values are used as the identifier of Entities, due to their homonym, confusion can occur. In particular, when (part of) "names" of entities are used as their identifier, the problem is often referred to as the name disambiguation problem, where goal is to sort out the erroneous entities due to name homonyms (e.g., if only last name is used as the identifier, one cannot distinguish "Vannevar Bush" from "George Bush"). We formalize the problem in a unified probabilistic framework and propose a algorithm for parameter estimation. We use a dynamic approach for estimating the number of people *K* and for finding the experts in digital library by counting the number of accesses of the paper.

Index Terms — Information search and retrieval, Clustering Algorithms.

1. INTRODUCTION

Name ambiguity is a real-world problem that one name possibly refers to more than one actual persons. It is a critical problem in many applications, such as: expert finding, people connection finding, and information integration.

Specifically, in scientific bibliography systems, the name disambiguation problem can be formalized as: given a list of publications with all sharing an identical author name but might actually referring to different persons, the task then is to assign the publications to some different clusters, each of which contains publications written by the same person.

By viewing all publications as vectors of multiple dimensions and each publication as a point in the multiple-dimensional space, we can obtain a straightforward solution by using clustering methods to deal with this problem.

Say you are looking for information about a particular person. A search engine returns many pages for that person's name but which pages are about the person you care about, and which are about other people who happen to have the same name? To solve this problem we have formalized the problems in a unified framework and proposed a generalized probabilistic model to the problem and have proposed a two-step parameter estimation algorithm. We also investigate a dynamic approach for estimating the number of people *K*. We also propose an algorithm for counting the number of times accessing of a particular paper.

BACKGROUND

With the emergence of major search engines like Google and Yahoo! that automate the process of gathering web pages to facilitate searching, it has become increasingly common for Internet users to search for their desired results to specific queries through search engines, with name queries making up approximately 5-10% of all searchers.

Name queries are usually treated by search engines as normal keyword searches without attention to the ambiguity of particular names.

For example, searching Google for "Yang Song" results in more than 11,000,000 pages with the same person's name, of which even the first page shows five different people's home pages. Table 1 lists the first four results which correspond to four different people. Due to this

heterogeneous nature of data on the Internet crawled by search engines, the issue of identity uncertainty or *name ambiguity* has attracted significant research attention. Beyond the problem of sharing the same name among different people, name misspelling, name abbreviations and other reference variations compound the challenge of name disambiguation.

The same issue also exists in most Digital Libraries (DL), hampering the performance and quality of information retrieval and credit attribution. In DL such as DBLP1 and CiteSeer, textual information is stored in metadata records to speed up field searching, including titles, venues, author names and other data. However, the existence of synonyms and *polysems* as well as typographical errors makes the problem of disambiguating author names in bibliographies (citations) non-trivial.

In the case of *synonyms*, an author may have multiple name variations/abbreviations in citations across publications, e.g., the author name "C. Lee Giles" is sometimes written as "C. L. Giles" in citations.

For *polysems*, different authors may share the same name label in multiple citations, e.g., both "Guangyu Chen" and "Guilin Chen" are used as "G. Chen" in their citations. In addition to the issue of citations, authors may be inclined to use different name variations even in the title pages of their publications due to a variety of reasons (such as the change of their maiden names).

Yang Song Homepage of **Yang Song**, PhD candidate of Penn State Department of Computer Sciences and Engineering. http://www.cse.psu.edu/~yasong/
Yang Song Home page of **Yang Song**, CALTECH, Department of Electrical Engineering... http://www.vision.caltech.edu/yangs/
Yang Song's Homepage **SONG, Yang**, Department of Statistics, UW-Madison Medical Science Center... http://www.cs.wisc.edu/~yangsong/
Song Yang the Cartoonist **Song Yang** is certainly the most successful cartoonist on the Mainland... http://japanese.china.org.cn/english/NM-e/155786.htm

Table: First 4 search results of the query "Yang Song" from Google that refer to 4 different people.

MOTIVATION

Figure 1 shows a simplified example for name disambiguation. In Figure 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 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 the eleven papers should be assigned to three different authors.

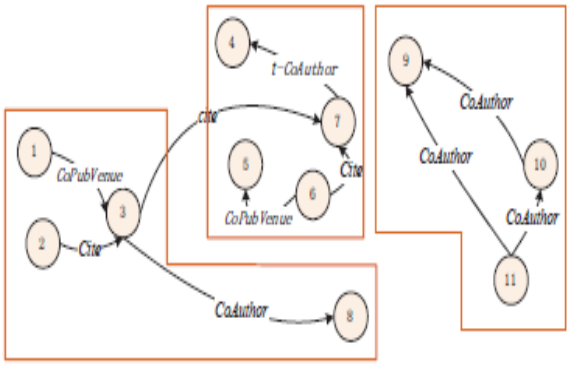


Figure 1: An example of name disambiguation

An immediate observation from Figure 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, an immediate challenging issue is how to propose an algorithm for the name disambiguation problem by considering both attribute information of the node and the relationships between nodes.

PRIOR WORK

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., (Han et al., 2004)) 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., (Han et al., 2005;Shu et al., 2009; Song et al., 2007; Yin et al., 2007)), clustering algorithms or topic models are employed to find paper 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 towards better data.

Although much progress has been made, existing methods do not achieve satisfactory disambiguation results due to their limitations:

1. The performance of all the aforementioned methods depends on accurately estimating *K*. Although several clustering algorithm such as *X*-means (Pelleg & Moore, 2000) 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.
2. In exiting methods, the data usually only contains 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.

Solution

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;

(3)An empirical verification of the effectiveness of the proposed framework and

(4) Finding the experts based on clicking the particular paper.

2. PROBLEM FORMALIZATION

Here we assign six attributes to each paper *pi* as shown in Table 1.

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

TABLE 1: ATTRIBUTES OF EACH PUBLICATION

Definition 1:

Principle Author and Secondary Author:

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

Relationships:

We described 5 different types of undirected relationships between papers. They are:

1) CoPubVenue ($r1$): It represents two papers published at the same venue.

2) Coauthor ($r2$): It represents that two papers $p1$ and $p2$ have a secondary author with the same name.

3) Citation ($r3$): It represents one paper citing another paper.

4) Constraint ($r4$): It denotes constraints supplied via user feedback.

5) τ-Coauthor ($r5$): It represents τ–extension Coauthor relationship. We use an example to explain this relationship. Suppose paper *pi* has authors "David Mitchell" and "Andrew Mark", and *pj* 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 *pi* and *pj* have a 2-CoAuthor relationship.

Clustering:

Clustering algorithms maps the data items into clusters, where clusters are natural grouping of data items based on similarity methods. Unlike classification & prediction which analyzes class label data objects, clustering analyzes data objects without class-labels and tries to generate such labels.

Definition 2:

Cluster Atom: A Cluster atom is a cluster in which papers are closely connected.

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.

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.

Definition 3:

Publication Informative Graph:

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

3. FRAMEWORK

3.1 Basic Idea

There are 2 observations for the Name Disambiguation problem:

(1) Papers with similar content tend to have the same label that is 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 for 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 (MRFs) (Hamersley & Clifford, 1971; Kindermann & Snell, 1980). 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.

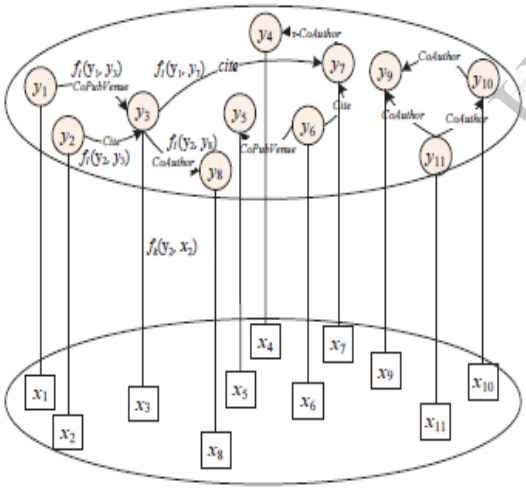


Figure 2: Graphical representation of HMRF model.

3.2 Hidden Markov Random Fields:

A Markov Random Field (MRF) is a conditional probability distribution of labels (hidden variables) that obeys the Markov property (Hamersley, 1971). A **Markov Property** states that for a stochastic process, if the conditional probability distribution of future states of the process depends only on the present state not on the sequence of events that preceded it.

A Hidden Markov Random Fields (HMRF) is a member of the family of MRFs and its concept is derived from Hidden Markov Models (HMM). **Hidden Markov Model** is a statistical markov model in which the system being modeled is assumed to be a markov process with hidden states.

A HMRF is mainly composed of three components:

(1) An observable set of random variables $X=\{xi\}n\ i=1$,

(2) A hidden field of random variables $Y=\{yi\}\ n_i=1$, and

(3) 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 yi takes a value from the set $\{1, K\}$ which are the indexes of the clusters. The observation variables X correspond to papers, where every random variable xi is generated from a conditional probability distribution $P(xi|yi)$ determined by the corresponding hidden variable yi. Further, the random variables X are assumed to be generated conditionally independently from the hidden variables Y, i.e.,

$$P(X\backslash Y) = \pi_{xi=X} P(x_i\backslash y_i)$$

Figure 2 shows the graphical structure of the HMRF for the example in Figure 1. We see that dependent edges are provided between the hidden variables corresponding to the relationships in Figure 1. The value of each hidden variable (e.g., $y1=1$) 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 *yi* for the observation variable *xi* depends only on the cluster labels of observations that have relations with *xi* (Kindermann & Snell, 1980).
By the fundamental theorem of random fields (Hamersley & Clifford, 1971), the probability distribution of the label configuration *Y* has the form:

$$P(y)= \frac{1}{z_1}\exp\left(\sum_{(y_i,y_j)\in E,k} \lambda_k f_k(y_i, y_j)\right),$$

$$z1 = \sum_{(y_i,y_j)} \sum_{(y_i,y_j)\in E,k} \lambda_k f_k(y_i, y_j)$$

where $f_k(y_i, y_j)$ is a non-negative potential function (also called the feature function) defined on edge (*yi*, *yj*) and *E* represents all edges in the graph; *fl*(*yi*, *xi*) is a potential function defined on node *xi*; λk and αl are weights of the edge feature function and the node feature function respectively; *Z*1 and Z2 are normalization factors. To facilitate further discussion, we hereafter use *X* to denote the publication set *P* and use *xi* to denote the vector *V* (*pi*) of the paper *pi*.

3.3 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 sub-clusters. Next, for each sub-cluster,
we again use the measurement to determine whether to split. The operation repeats until some condition is satisfied (e.g., no sub-cluster can be split). In the process, we call *Mh* the model corresponding to the solution with the person number *h*. We therefore have a family of alternative models {*Mh*}, where *h* ranges from 1 to *n*, inclusively. Now our task is to choose the best model from {*Mh*}.

Bayesian Information criterion:

Many measurements can be used for model selection, such as Silhouette Coefficient (Kaufman & Rousseau, 1990), Minimum Description Length (MDL) (Rissanen, 1983), Akaike Information Criterion (AIC) (Akaike, 1974), and posterior probability estimation (Kass & Wasserman, 1995). 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 (Kass & Wasserman, 1995) as the criterion:

$$BIC^{v}(M_h)= \log(P(M_h / P)) - \frac{|\lambda|}{2} \log(n)$$

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 non-zero parameters in the model *Mh* 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 *Mh* fits the whole data set. We use this criterion for the model selection because it can be easily extended to different situations.

4. PARAMETER ESTIMATION

At a high level, the learning algorithm for parameter estimation primarily consists of two iterative steps: *Assignment* of papers, and *Update* of parameters. 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 groups are used as our initial assignment. If γ<*K*, we randomly

Choose another (*K*-γ) papers as the cluster centroids.If γ>*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.

Algorithm 1: Estimation of the parameter

Input: P= {p_1,p_2,…..,pn}

Output: model parameters Θ and Y={y_1,y_2,…,y_n},where y_i Ɛ [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 Ɛ [1,k];
   - 1.3 Calculate each paper cluster centroid $\mu_{(i)}$;
   - 1.4 For each pair xi and each relationship (x_i.x_j),calculate $f_i(y_i,x_i)$ and fi(yi,yj);
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.

Estimation of *K and paper counter:*

Our strategy for estimating *K* 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.

Algorithm 3: Estimation of K and Paper counter

Input: P= {p_1,p_2,..,p_n}
Output: K,Y={y_1,y_2,…,y_n}, where y_iƐ[1,K]

1. *I=0;k=1,that is to view P as one cluster: $C^{(i)}$={C_1};*
2. *do {*
3. *foreach cluster C in $C^{(i)}${*
4. *find a best two sub cluster model M_2 for C;*
5. *if(BIC(M_2)>BIC(M_1))*
6. *split cluster C into two sub clusters $C^{(i+1)}$={C_1,C_2};*
7. *calculate BIC score for obtained the new model;*
8. *}while(existing split);*
9. *Choose the model as output with highest BIC score;*
10. *foreach paper p_i in P{*
11. *initialize counter I for each paper p_i*
12. *if paper p_i is selected then*
13. *increment counter(i)*
14. *sort the counter$_i$ values for paper$_i$*
15. *Display papers with maximum counter value.*Type equation here.

In each iteration, we try to split every cluster *C* into two sub-clusters. We calculate a local BIC score of the new sub-model *M*2. If BIC (M_2) > 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. One difficulty in the algorithm might be how to find the best two sub-cluster models for the cluster *C* (Line 4).

With different initialization, the resulting sub-clusters 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 centroid and thus our algorithm tends to result in stable split results.

EXPERIMENTAL RESULTS

5.1 Experimental Setting

Data Sets We created a dataset, which includes 240 real author names and 1646 papers.

Table: DATA SET

Abbr.Name	#Publications	#Actual Persons
Subramanyam	12	3
Nagraj	286	4
Satheesh	54	25
Robert	109	40
John	42	21
Williams	33	5
Rajesh	66	12
Ajay	21	2
Rahim	105	4
Dhanush	44	12
Vijay	110	2
Kiran	61	5
Madhan	130	11
Prakash	27	4
Ram	306	90

]In these names, some names are only associated with a few persons, for example "Subramanyam" is the name of three persons and "Nagraj" four; while some names seem to be popular. For example, there are 25 persons with the name "Satheesh" and 40 persons named "Robert". 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, email 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. We also found that the disambiguation results are extremely unbalanced.

For example, there are 286 papers authored by "Nagraj" with 282 of them authored by Prof. Nagraj from the Institute of Computing at Chinese Academy of Science and only four papers are authored by the other three persons named "Nagraj". 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.

5.2 Experimental Design:

We use Pair wise Precision, Pair wise Recall, and Pair wise F1 score, to evaluate our method and to compare with previous methods. The pair wise 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 dataset, we call it a mistakenly predicted pair.

We further compared our method with two existing methods for name disambiguation: DISTINCT (Yin et al., 2007), a combination method based on two similarity measures: set resemble of neighbor tuples and random walk probability; CONSTRAINT (Zhang et al., 2007), 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.

(3)We calculated the number of access of each paper.

5.3 Experimental Results

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 SA Cluster considers the relationship between nodes, it incorporates the relationship information into a 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.

Below table 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% by F1 score). This confirms that a model which cannot capture dependencies between papers would not result in good performance.

Table: Results of the approach with different settings

Method	Precision	Recall	F1-Measure
Auto K	83.07	79.80	80.09
w/o auto K	90.93	88.76	88.98
w/o relation	67.55	50.79	56.01

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 "john" with 2. The reason might be that *X*-means cannot make use of the relationships between papers. We compared our approach with DISTINCT (Yin et al., 2007). We used person names that were used both in (Yin et al., 2007) and our experiments for comparisons. We conducted the experiments on our data set, which is a newer version of data used in (Yin et al., 2007). For example, we have 109 papers for "Robert" and 33 papers for "Williams", while in (Yin et al., 2007) the numbers are 55 and 19. In addition, we do not consider the Proceeding Editor relation. Moreover, our approach has the advantage that it can automatically find the number *K*, where as 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 Coauthors and CoPubVenue relations, although the two relations can be derived from the paper-conference and author-paper relations.

5.4 Distribution analysis

We use a dimension reduction method for performing a distribution analysis. 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. Name disambiguation on this kind of data can be solved pretty well by all approach; (2) publications are mixed together but with a dominant author who writes most of

the papers. Our approach can achieve a F1 score of 88.36% and the discovered number *K* is close to the actual number; and (3) publications of different authors are mixed. Our method can obtain a performance of 92.25%. However, it would be difficult to accurately find the number *K*.

5.5 Application experiments

We applied the name disambiguation to help expert finding, which is to identify persons with some given expertise or experience.

Precision, Recall & F-measure:

Precision: Precision determines the fraction of records that actually turns out to be positive in the group the classifier has declared as a positive class.

Precision = TP / (TP + FP)

Recall : Recall is computed as the fraction of correct instances among all instances that actually belong to the relevant subset i.e (Recall = Actual True Positive rate) .

Recall = TP / (TP + FN)

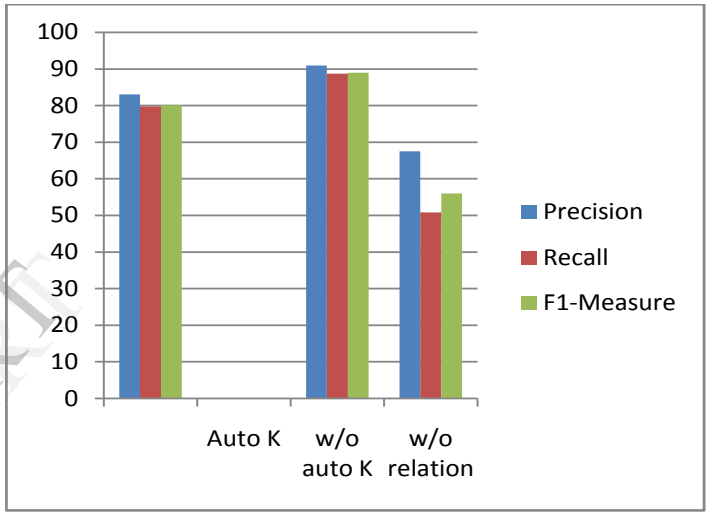
F-measure: A measure that combines precision and recall is the harmonic mean of precision and recall, the traditional F-measure.

F-measure = 2 * ((Precision.Recall) / Precision + Recall)) or = 2*TP / (2*TP) + FP + FN

There are four ways of being right or wrong:

1. TN / True Negative: case was negative and predicted negative
2. TP / True Positive: case was positive and predicted positive
3. FN / False Negative: case was positive but predicted negative
4. FP / False Positive: case was negative but predicted positive

Precision, recall, and the F measure are set-based measures. They are computed using unordered sets of documents. We need to extend these measures (or to define new measures) if we are to evaluate the ranked retrieval results that are now standard with search engines. In a ranked retrieval context, appropriate sets of retrieved documents are naturally given by the top k retrieved documents.



We see that clear improvements can be obtained by using the proposed name disambiguation approach.

6. 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. and have proposed a two-step parameter estimation algorithm. We have also explored a dynamic approach for estimating the number of people *K* and for calculating the number of access for a particular paper. Experimental results indicate that the proposed method significantly outperforms the baseline

methods. When applied to expert finding, clear improvement (+2%) can be obtained.

REFERENCES

1. " A Unified Probabilistic Framework for Name Disambiguation in Digital Library",Jie Tang,A.C.M.Fong,Bo Wang and Jing Zhang, Member IEEE.
2. H. Han, H. Zha, and C.L. Giles, "Name disambiguation in author citations using a K-way spectral clustering method", *Proc.of JCDL'05*, pp. 334 – 343, 2005.
3. L. Kaufman and P. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*, New York: Wiley, 1990.
4. X. Li, P. Morie, D. Roth, "Identification and tracking of ambiguous names: discriminative and generative approaches", *Proc.of AAAI'04*, pp. 419-424, 2004.
5. D. Pelleg and A. Moore, "X-means: extending K-means with efficient estimation of the number of clusters", *ICML'00*, 200
6. Y. Song, J. Huang, I. G. Councill, J. Li, and C. L. Giles, "Efficient Topic-based Unsupervised Name Disambiguation", *Proc. Of JCDL'07*, pp. 342-351,2007.
7. Y. Zhou, H. Cheng, and J. X. Yu, "Graph clustering based on structural/attribute similarities", *Proc. of VLDB'09/PVLDB*,vol.2 (1), pp. 718-729, 2009.
8. B. On and D. Lee, "Scalable Name Disambiguation using Multilevel Graph Partition", *Proc. of SDM'07*, 2007.
9. R. Bekkerman and A. McCallum, "Disambiguating web appearances of people in a social network", *Proc. of WWW'05*, pp.463-470, ACM Press, 2005.