A Hierarchical Naive Bayes Mixture Model for Name Disambiguation in Author Citations

Hui Han[1,2]
[1]Yahoo Inc.
701 First Avenue
Sunnyvale, CA, 95129
huihan@yahoo-inc.com

Wei Xu[3]
[3]NEC Laboratories America, Inc.
10080 North Wolfe Road, Suite SW3-350
Cupertino, CA 95014
xw@sv.nec-labs.com

Hongyuan Zha[2]
[2]Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA, 16802
zha@cse.psu.edu

C. Lee Giles[2,4]
[4]School of Information Sciences and Technology
The Pennsylvania State University
University Park, PA, 16802
giles@ist.psu.edu

ABSTRACT

Because of name variations, an author may have multiple names and multiple authors may share the same name. Such name ambiguity affects the performance of document retrieval, web search, database integration, and may cause improper attribution to authors. This paper presents a hierarchical naive Bayes mixture model, an unsupervised learning approach, for name disambiguation in author citations. This method partitions a collection of citations [1] into clusters, with each cluster containing only citations authored by the same author, thus disambiguating authorship in citations to induce author name identities. Three types of citation features are used: co-author names, paper title words, and journal or proceeding title words. The approach is illustrated with 16 name datasets that are constructed based on the publication lists collected from author homepages and DBLP computer science bibliography.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval

Keywords

Name Disambiguation, Feature Selection, Unsupervised Learning

1. INTRODUCTION

Due to identical names, name misspellings, inconsistent inclusion of initials, pseudonyms, and marriage, we observe two types of name ambiguities in research papers or bibliographies (citations). The first type is that an author has multiple name labels. For example, the author "Michelle Q Wang" has different names after marriage: "Michelle Q Wang-Baldonado" or "Michelle QW Baldonado". The second type is that multiple authors may share the same name label. For example, "D. Johnson" may refer to "David B. Johnson" from Rice University, "David S. Johnson" from AT&T research lab, or "David E. Johnson" from Utah University (assuming the authors still have these affiliations).

Name ambiguity can affect the quality of scientific data gathering, can decrease the performance of information retrieval and web search, and can cause the incorrect identification of and credit attribution to authors. For example, the author page of "Jia Li" in the DBLP refers to the "Jia Li" from the Department of Statistics at the Pennsylvania State University. However, the "Home Page" link in her author page directs to the professor with the identical name in the Department of Mathematical Sciences at the University of Alabama in Huntsville. Another example is from CiteSeer's statistics in May 2003 [2], which shows that "D. Johnson" is the most cited author in computer science. However, the citation number that "D. Johnson" obtained in CiteSeer's statistics is actually the sum of several different authors such as "David B. Johnson", "David S. Johnson", and even "Joel T. Johnson".

Given a set of citations that have an ambiguous (e.g. identical) name label, how do we disambiguate authors if the name label refers to a single author, or different authors with ambiguous names? Such problem can be addressed by either supervised or unsupervised learning methods. Supervised learning methods consider each canonical author name as a class, and identify the correct author class for each citation. However, supervised learning methods need authors' previous citations to train classifiers, which are not necessarily available. With unsupervised learning methods, we do not need labeled data for training. The name disambiguation problem can be formulated as partitioning collections of citations into clusters, with each cluster containing only citations authored by the same author, thus disambiguating authorship in citations to induce author name identities. This paper introduces an unsupervised learning approach based on a hierarchical naive Bayes mixture model to disambiguate names in author citations. Three types of features are used: coauthor names, paper title words, and publi-

[1]http://www.library.umass.edu/reference/glossary.html#cite

[2]http://citeseer.ist.psu.edu/mostcited.html

SAC'05 March 13-17, 2005, Santa Fe, New Mexico, USA

cation venue title words. "Publication venue title" refers to the title of any the publication sources, such as proceedings or journals.

2. RELATED WORK

Name ambiguity is a special case of the general problem of *identity uncertainty*, where objects are not labeled with unique identifiers [18]. Previous research has addressed the identity uncertainty problem using different methods, such as record linkage [9], duplicate record detection and elimination [4, 14, 17], merge/purge [13], data association [2], database hardening [6], word sense disambiguation [20], citation matching [16], name matching [3, 19, 5], and name authority control in library cataloging practice [7].

Name authority control and name matching are the work most similar to ours. Name authority control aims to find the authoritative form of names, i.e., the unambiguous reference to an individual [7]. Name authority control usually provides a set of rules and standardized terms for consistent name representation (e.g., the form of the name to be used). Much work in name authority control relies on manual analysis [11]. Recent research [7, 12] considers supervised learning systems, and relies much on a priori knowledge of ambiguous name entities or name lists.

Name matching usually identifies a name entity with different name labels from duplicate records of different syntactic formats. Name matching does not focus on the case of different name entities that have identical name labels. Our method disambiguates names from different records (citations) authored by the same name entity, and addresses both types of name ambiguities previously mentioned. Our method works in conjunction with name matching that usually uses string-based comparison to induce the correct name entities from names with misspellings and abbreviations.

3. EXPERIMENTAL DATASETS

We have two types of citation data, and a citation of either type contains three attributes: coauthor name(s), paper title and publication venue title. The first type of citations are publication lists collected from the web, mostly from researchers' homepages. This type of data contains two datasets, one contains 15 different "J Anderson"s who have 229 citations in total; the other contains 11 different "J Smith"s who have 339 citations in total. Citation attributes are parsed by using regular expressions.

The second type of citations are mainly downloaded from the DBLP Computer Science Bibliography which contains more than 400,000 citation records with parsed citation attributes in the XML format. We concatenate the three attributes in each citation as a string, and then cluster citations with author names of the same first name initial and the same last name. We sort the formed citation clusters by the number of name variations contained, and select 14 large sets of ambiguous names from the DBLP bibliography for experiments, as shown in Table 1. Each name dataset has more than 10 canonical authors in consideration. Moreover, we enrich the datasets with publication lists downloaded from author homepages that are found when we label the canonical author names (as next subsection describes). The goal is to provide each canonical author name with the maximal amount of available citation information.

The DBLP datasets seem to be more challenging than the web collected datasets. Because most authors in the DBLP datasets come from the computer science community, different researchers are likely to have overlapping research interests, and publish papers in the same research area. Common paper or publication venue title keywords shared by different authors are in fact "ambiguous" information, which makes disambiguation harder.

3.1 Data Processing

3.1.1 Labeling

For evaluation purpose, we manually label the canonical name entities and associated citations from both the web collected datasets and the DBLP datasets. Citations listed in an author's publication home page are considered as being written by the same author. Authors with the same name and same affiliation, or same email address are considered to be the same. Authors of the same name that also have the same co-author names (in a complete name format) are very likely the same author. Citations that have the same name label, and are about the same topic are likely to be written by the same author. We also sent emails to some authors to confirm their authorship of citations. The citations for which we had insufficient information to be judged were eliminated. Moreover, we populate the datasets with publication lists downloaded from the available home page URLs of authors in the datasets.

3.1.2 Data Preprocessing

All the author names in the citations are simplified to first name initial and last name. For example, "Yong-Jik Kim" is simplified to "Y Kim". A reason for the simplification is that the first name initial and last name format is popular in bibliographic records. Since more name information usually helps name entity disambiguation, insufficient name information from simplified name format would be good for evaluating our algorithms. Moreover, the simplified name format may avoid some cases of name misspellings. Third, the simplified name format helps to construct the ambiguous name datasets, because there are usually more canonical names that share the identical first name initial and last name than the canonical names that share the complete name. Words of paper titles and publication venue titles are stemmed, and stop words are removed. Conference or publication venue title abbreviations are replaced by their available full names [3].

3.2 Evaluation Method

We evaluate experimental results based on the confusion matrix, where $A[i, j]$ represents the number of "Author i" predicted as "Author j" in matrix A. $A[i, i]$ represents the number of correctly predicted names for "Author j". We define the disambiguation accuracy as the sum of diagonal elements divided by the total number of elements in the matrix.

3.3 The Hierarchical Naive Bayes Mixture Model

3.3.1 The Mixture Model

We assume that a citation C_m is generated by a mixture of K components (canonical authors). Equation (1) shows that the probability of citation C_m is equal to the weighted sum of C_m's probability for each canonical author X_i alone. $P(X_i)$ is the weight, or prior probability for each canonical author X_i.

$$P(C_m) = \sum_{i=1}^{K} (P(X_i) * P(C_m|X_i)) \quad (1)$$

Each of the K canonical authors is modeled by a hierarchical naive Bayes model as described in next section. We use the Expectation-Maximization (EM) algorithm to estimate the mixture model parameters, as described in the next section, with the target function of maximizing the likelihood of the citation dataset, i.e.,

$$\max(\sum_{m} (P(C_m))) \quad (2)$$

After the model parameters are estimated, we assign each citation C_m to the canonical author that maximizes $P(X_i|C_m)$. According to Bayes rule, each citation C_m is assigned to the canonical

[3] http://www.informatik.uni-trier.de/∼ley/db/conf/indexa.html and http://www.informatik.uni-trier.de/∼ley/db/journal/index.html

Name	A Gupta	A Kumar	C Chen	D Johnson	J Lee	J Martin	J Robinson	J Smith	K Tanaka	M Brown	M Jones	M Miller	S Lee	Y Chen
N	11	5	20	6	38	4	6	12	5	5	6	5	36	22
C	507	210	630	335	1187	66	148	848	258	115	221	389	1290	1051

Table 1: The 14 DBLP name datasets. Column "N": the number of canonical authors in each dataset; Column "C": the number of citations in each dataset. E.g., Dataset "J. Lee" has 38 different "J. Lee" and 1187 citations.

author that has the maximal probability of producing C_m, that is, $max\ (P(C_m|X_i) * P(X_i))$.

3.3.2 Model Hierarchy

We assume that coauthors, paper titles, and publication venue titles are independent citation attributes. Therefore, we decompose $P(C_m|X_i)$ as

$$P(C_m|X_i) = \prod_{j=1}^{3} P(A_j|X_i) = P(A_1|X_i)P(A_2|X_i)P(A_3|X_i) \quad (3)$$

, where A_j denotes the different type of attribute; that is, A_1 - coauthor names; A_2 - paper title; A_3 - publication venue title. We also assume that different elements (an coauthor, or a title word of the paper or the publication venue) in an attribute type are conditionally independent from each other. Alghough such independent assumptions may not hold for real-world data, (e.g., multiple coauthors always appear together), empirical evidence shows that naive Bayes often performs well in spite of such violation [10, 8].

We build a hierarchical naive Bayes model to estimate $P(A_j|X_i)$, as shown by Figure 1. We expect this hierarchical model to capture the coauthoring history and patterns of X_i, and to help disambiguate the omitted author from the rest of a citation C_m. We estimate the conditional probabilities $P(A_1|X_i)$ that an author writes a paper with coauthors, $P(A_2|X_i)$ that an author writes a paper title, and $P(A_3|X_i)$ that an author publishes in a particular publication venue. This model has the hypothesis that (1) Different authors X_i have different probabilities of writing papers alone, or writing papers with previously seen or unseen coauthors; (2) Each author X_i has his/her own list of previously seen coauthors, and a unique probability distribution on these previously seen coauthors to write papers with; (3) Author keyword usage patterns are similar to coauthor patterns. We expect author-specific probabilities to capture information such as the research field, keywords in the research direction, and the preference of title word usage from past citations of X_i.

While an author may write papers alone or write papers with coauthors (as shown by Equations (4) and (5)), a paper title or a publication venue title must contain keywords (as Equation (6) shows).

$$P(A_1|X_i) = \begin{cases} P(A_1|Co_1=0,X_i) * P(Co_1=0|X_i) \\ \text{(if } A_1 \text{ writes paper alone)} \\ P(A_1|Co_1=1,X_i) * P(Co_1=1|X_i) \\ \text{(if } A_1 \text{ writes paper with coauthors)} \end{cases} \quad (4)$$

$$P(A_1|Co_1=0,X_i) = \begin{cases} 1 & \text{if } A_1 \text{ is empty} \\ 0 & \text{if } A_1 \text{ is not empty} \end{cases} \quad (5)$$

$$P(A_j|X_i) = P(A_j|Co_1=1,X_i)$$
$$P(Co_j=1|X_i) = 1$$
$$P(A_j,Co_j=0|X_i) = 0, (j=2,3) \quad (6)$$

3.3.3 Model Parameters Estimation

This subsection describes estimation of the conditional probabilities that are decomposed from $P(A_1|X_i)$ from the training citations. The probability estimation is the maximum likelihood estimation for parameters of multinomial distributions. The pseudo count 1 is added in parameter estimation to avoid zero probability

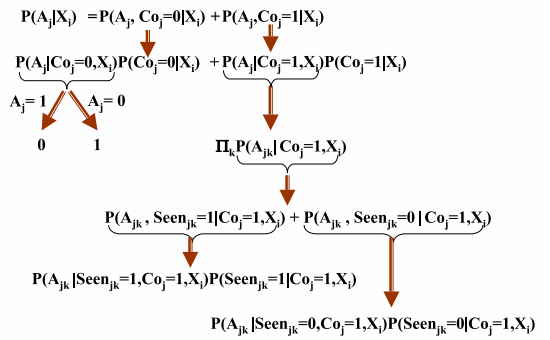

Figure 1: A hierarchical naive Bayes model of estimating $P(A_j|X_i)$.

in the estimation results. Parameter estimations for $P(A_2|X_i)$ and $P(A_3|X_i)$ are similar to the estimation of $P(A_1|X_i)$.

- $P(Co_1 = 0|X_i)$ - the probability of X_i writing a future paper alone conditioned on the event of X_i, estimated as the proportion of the papers that X_i authors alone among all the papers of X_i.
- $P(Co_1 = 1|X_i)$ - the probability of X_i writing a future paper with coauthors conditioned on the event of X_i. $P(Co_1 = 1|X_i) = 1 - P(Co_1 = 0|X_i)$.
- $P(Seen_{1k} = 1|Co_1 = 1, X_i)$ - the probability of X_i writing a future paper with previously seen coauthors conditioned on the event that X_i writes a future paper with coauthors. We regard the authors coauthoring a paper with X_i at least twice in the training citations as the **"seen coauthors"**; the other coauthors coauthoring a paper with X_i only once in the training citations is considered as the **"unseen coauthors"**. Therefore, we estimate $P(Seen_{1k} = 1|Co_1 = 1, X_i)$ as the proportion of the number of times that X_i coauthors with "seen coauthors" among the total number of times that X_i coauthors with any coauthor. Note that if X_i has n coauthors in a training citation C_m, we count that X_i coauthors n times in citation C_m.
- $P(Seen_{1k} = 0|Co_1 = 1, X_i)$ - the probability of X_i writing a future paper with "unseen coauthors" conditioned on the event that X_i writes a paper with coauthors. This probability and $P(Seen_{1k} = 1|Co_1 = 1, X_i)$ do not depend on k. $P(Seen_{1k} = 0|Co_1 = 1, X_i) = 1 - P(Seen_{1k} = 1|Co_1 = 1, X_i)$
- $P(A_{1k}|Seen_{1k} = 1, Co_1 = 1, X_i)$ - the probability of X_i writing a future paper with a particular coauthor A_{1k} conditioned on the event that X_i writes a paper with previously seen coauthors. We estimate it as the proportion of the number of times that X_i coauthors with A_{1k} among the total number of times X_i coauthors with any coauthor.
- $P(A_{1k}|Seen_{1k} = 0, Co_1 = 1, X_i)$ - the probability of X_i writing a future paper with a particular coauthor A_{1k}

conditioned on the event that X_i writes a paper with unseen coauthors. Considering all the names in the training citations as the population and assuming that X_i has equal probability to coauthor with an unseen author, we estimate $P(A_{1k}|Seen_{1k}=0, Co_1=1, X_i)$ as 1 divided by the total number of author (or coauthor) names in the training citations minus the number of coauthors of X_i. However, the small citation size may underestimate the population of new coauthors that X_i will coauthor with in the real-world. This may in turn underestimates the probability of an author coauthoring with previously seen coauthors. In this case a larger population size is needed.

3.3.4 The Expectation-Maximization Algorithm

	Average			Best		
	K means	Mixture model		K means	Mixture model	
		Original	Word cluster		Original	Word cluster
A. Gupta	29.7%	47.6%	46.8%	37.1%	56.2%	54.0%
A. Kumar	43.0%	46.3%	48.1%	55.6%	60.0%	56.7%
C. Chen	24.6%	38.2%	41.2%	34.9%	45.4%	45.2%
D. Johnson	41.2%	44.5%	45.6%	56.8%	55.8%	60.9%
J. Lee	19.0%	49.5%	45.3%	24.9%	53.3%	48.6%
J. Martin	39.2%	65.6%	66.4%	46.7%	80.3%	77.3%
J. Robinson	28.7%	57.1%	57.6%	38.7%	72.3%	66.2%
J. Smith	34.6%	61.7%	62.7%	48.4%	68.2%	71.5%
K. Tanaka	50.1%	59.5%	61.6%	70.8%	66.3%	73.3%
M. Brown	40.3%	66.0%	65.5%	53.6%	78.3%	80.0%
M. Jones	36.8%	65.7%	62.9%	46.1%	76.5%	70.1%
M. Miller	50.6%	59.8%	59.5%	66.7%	68.9%	63.2%
S. Lee	20.4%	46.4%	39.5%	23.8%	51.7%	42.1%
Y. Chen	28.8%	49.0%	49.2%	46.1%	51.8%	52.3%
Avg	**34.8%**	**54.1%**	**53.7%**	**46.4%**	**63.2%**	**61.5%**
StdDev	**10.0%**	**9.2%**	**9.5%**	**13.9%**	**11.2%**	**12.1%**
P Value	**5.41E-06**			**0.00047**		
P Value		**0.62**			**0.21**	

Table 2: The name disambiguation accuracies(%) on 14 DBLP name datasets achieved by both methods. "Mixture model" : our hierarchical naive Bayes mixture model; "Original" : the citations that contain original words as downloaded; "Word cluster": citations that have the original words replaced by their cluster labels; "Avg": average results; "StdDev": standard deviation; "P value": two tail value from T-test.

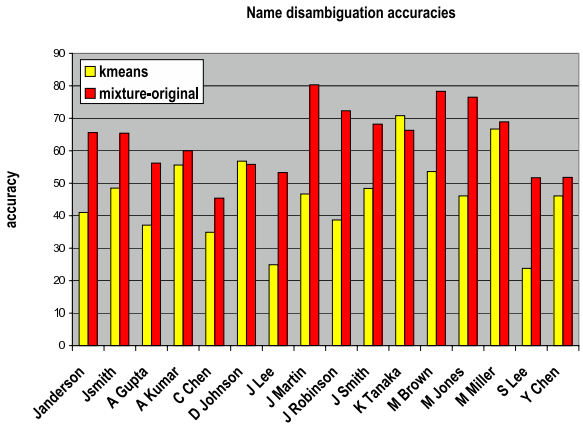

Figure 2: The best name disambiguation accuracies of 10 times experiments on 16 name datasets by both methods.

Step1. Initialization. Randomize and equally assign N citations (N is the total number of citations in the dataset) into K clusters. Estimate the following probabilities: the prior probability of each of the K components, $P(k)$ ($k \in \{1, \cdots, K\}$); the hierarchical conditional probabilities as shown in Figure 1 (e.g., $P(Co_j = 1|k)$, $P(Co_j = 0|k)$, $P(Seen_{jk} = 1|Co_j = 1, k)$, $P(Seen_{jk} = 0|Co_j = 1, k)$, $P(A_{jk}|Seen_{jk} = 1, Co_j = 1, k)$, $P(A_{jk}|Seen_{jk} = 0, Co_j = 1, k)$); and $P(C_m|k)$.

$$P(k) = \frac{1}{K} \tag{7}$$

Step2. E-step. Reassign all citations to each cluster according to the posterior probability of each cluster producing the citation C_m.

$$P(k|C_m) = \frac{P(C_m|k) * P(k)}{\sum_k (P(C_m|k) * P(k))} \tag{8}$$

Step3. M-step. Compute $P(k)$, hierarchical conditional probabilities (e.g., $P(Co_j = 1|k)$, $P(Co_j = 0|k)$, $P(Seen_{jk} = 1|Co_j = 1, k)$, $P(Seen_{jk} = 0|Co_j = 1, k)$, $P(A_{jk}|Seen_{jk} = 1, Co_j = 1, k)$, $P(A_{jk}|Seen_{jk} = 0, Co_j = 1, k)$), and $P(C_m|k)$. N is the total number of citations in the dataset.

$$P(k) = \frac{\sum_m (P(k|C_m))}{N} \tag{9}$$

Step4. If the algorithm converges ($|| \sum_m (P(C_m)) - \sum'_m (P(C_m)) || < 0.1$), classify each citation C_m to the component(cluster) k that maximizes $P(k|C_m)$. Otherwise, continue step2 and step3.

$$P(C_m) = \sum_k P(C_m|k) * P(k) \tag{10}$$

3.4 The K means Algorithm

To study the performance of our algorithms on name disambiguation, we choose the K means algorithm for comparison. In K means algorithm, each citation is represented by a feature vector, with each coauthor name and each keyword of the paper title and the publication venue title as a feature of the vector. The weight of each feature is the "tf.idf" value of the feature. Euclidean distance is used to assign citation feature vectors to clusters.

3.5 Cluster Semantically Similar Words

Because the paper and publication venue title words are sparse, and an author may not reuse a certain group of words with high probabilities, it is reasonable to cluster the semantically similar words and model the probability that an author uses the similar words for his/her paper title. In our experiments, we cluster the paper title words and publication venue title words using Pantel and Lin [15] 's CBC (Clustering By Committee) clustering algorithm. We then replace each title word by its cluster label, which we call "feature transformation", before applying the hierarchical-naive-Bayes-model-based name disambiguation approach.

3.6 Experiments

We apply both K means algorithm and the hierarchical naive Bayes mixture model to both types of datasets. The number of clusters is set as the number of canonical names an ambiguous name label corresponds to in the labeled dataset. Tables 2, 3 show the average and the best results of 10 times experiments on both types of datasets respectively. Figure 2 shows the histogram of the best results achieved from 10 times experiments by both methods. The hierarchical naive Bayes mixture model is shown to outperform the K means algorithm on all datasets. Although both algorithms are prone to local minima, the mixture model appears to be a better fit for the problem of name disambiguation in author citations than the K means algorithm. The main reason is that our hierarchical naive Bayes model captures the author patterns that are not easily incorporated into feature vector space model that is used by K-means algorithm. These author patterns are the prior probability of an author, the probability that an author writes papers alone, the probabilities that an author writes a future paper with previously unseen coauthors, and the probabilities that an author writes a future paper using previously unused keywords.

We applied the CBC word clustering algorithm to clustering paper title words and publication venue title words. We then made "feature transformation" to titles by replacing each title word of a citation by its cluster label. We applied the hierarchical-naive-Bayes-mixture-model-based method to these citations that are "fea-

	Average		Best	
	K means	**Mixture model**	**K means**	**Mixture model**
J. Anderson	30.0%	57.6%	41.0%	65.6%
J. Smith	31.2%	59.8%	48.5%	65.4%
Avg	**30.6%**	**58.7%**	**44.8%**	**65.5%**
StdDev	**0.85%**	**1.56%**	**5.30%**	**0.14%**
P Value	**0.011**		**0.117**	

Table 3: Name disambiguation accuracies(%) on two web datasets.

ture transformed". Tables 2 and 3 list the name disambiguation results on the DBLP datasets before and after the feature transformation. The word clustering algorithm is shown to improve the name disambiguation results on datasets of "A Kumar", "D Johnson", "J Smith", "K Tanaka", and "Y Chen". However, word clustering does not improve the name disambiguation results on all the datasets. A possible reason is that the word clustering gathers information and also loses information. Choice of the size of a cluster affects the balance of information gain and information lose. Large size cluster seems to gather more information than smaller size cluster. However, large size cluster can lose more information than the smaller size cluster.

3.7 Conclusions and Discussion

This paper proposes an unsupervised learning method based a hierarchical naive Bayes mixture model for name disambiguation in author citations. This method outperforms K-means algorithm in 16 datasets, which is statistically significant. The main reason is that our hierarchical naive Bayes model captures the author patterns that are not easily incorporated into feature vector space model that is used by K-means algorithm. These author patterns are the prior probability of an author, the probability that an author writes papers alone, the probabilities that an author writes a future paper with previously unseen coauthors, and the probabilities that an author writes a future paper using previously unused keywords.

By clustering paper and publication venue title words and using word clusters as features, we increased accuracies of name disambiguation on some datasets. This shows the promise of applying word clustering, a feature representation and transformation technique, to text clustering, which agrees to previous research [1]. Further work needs automated thresholding in search for the optimal size of formed word clusters.

In our hand-labeling of the datasets, we used extra information such as affiliations, email addresses, resumes, home pages, and some human judgment. Therefore, in order to improve the name disambiguation performance, we most likely need more features as those that are used in our hand-labeling than the three citation attributes that we currently use. We would also like to address the issue of automatically choosing the number of name clusters.

Acknowledgments

We wish to thank Mark Stefik and Cheng Li for their valuable comments on our name disambiguation work. We would like to acknowledge partial support from NSF Grant 0121679 and CCF 0305879. We appreciate the citation data provided by DBLP computer science bibliography.

4. REFERENCES

[1] L. D. Baker and A. K. McCallum. Distributional clustering of words for text classification. In W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *Proceedings of the 21st ACM International Conference on Research and Development in Information Retrieval*, pages 96–103, 1998.

[2] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.

[3] M. Bilenko, R. Mooney, W. Cohen, P. Ravikumar, and S. Fienberg. Adaptive name matching in information integration. *IEEE Intelligent Systems*, 18(5):16–23, 2003.

[4] D. Bitton and D. J. DeWitt. Duplicate record elimination in large data files. *ACM Transactions on Database Systems*, 8(2):255–265, 1983.

[5] L. K. Branting. Name-matching algorithms for legal case-management systems. *Journal of Information, Law and Technology (JILT)*, 1, 2002.

[6] W. W. Cohen, H. A. Kautz, and D. A. McAllester. Hardening soft information sources. In *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining*, pages 255–259, 2000.

[7] T. DiLauro, G. S. Choudhury, M. Patton, J. W. Warner, and E. W. Brown. Automated name authority control and enhanced searching in the levy collection. *D-Lib Magazine*, 7(4), 2001.

[8] P. Domingos and M. J. Pazzani. Beyond independence: Conditions for the optimality of the simple bayesian classifier. In *International Conference on Machine Learning*, pages 105–112, 1996.

[9] I. P. Fellegi and A. B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64:1183–1210, 1969.

[10] J. Friedman. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Journal of Data Mining and Knowledge Discovery*, 1, 1997.

[11] P. Gillman. National name authority file: Report to the national council on archives. Technical Report British Library Research and Innovation Report 91, The British Library Board, 1998.

[12] H. Han, C. L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis. Two supervised learning approaches for name disambiguation in author citations. In *Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital libraries*, 2004.

[13] M. A. Hernandez and S. J. Stolfo. Real-world data is dirty: Data cleansing and the merge/purge problem. *Data Mining and Knowledge Discovery*, 2(1):9–37, 1998.

[14] M.-L. Lee, T. W. Ling, and W. L. Low. Intelliclean: a knowledge-based intelligent data cleaner. In *In 6th International Conference on Knowledge Discovery and Data Mining*, pages 290–294, 2000.

[15] D. Lin and P. Pantel. Concept discovery from text. In *Proceedings of Conference on Computational Linguistics*, pages 577–583, 2002.

[16] A. McCallum, K. Nigam, and L. H. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Knowledge Discovery and Data Mining*, pages 169–178, 2000.

[17] A. E. Monge and C. Elkan. An efficient domain-independent algorithm for detecting approximately duplicate database records. In *Research Issues on Data Mining and Knowledge Discovery*, pages 23–29, 1997.

[18] H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser. Identity uncertainty and citation matching. In *Proceedings of Neural Information Processing Systems: Natural and Synthetic 15*, 2002.

[19] S. Tejada, C. Knoblock, and S. Minton. Learning domain-independent string transformation weights for high accuracy object identification. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 350–359, 2002.

[20] H. R. Turtle and W. B. Croft. Uncertainty in information retrieval systems. *Uncertainty Management in Information Systems*, pages 189–224, 1996.