TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 07/14 pp668-677
Volume 15, Number 6, December 2010

Disambiguating Authors by Pairwise Classification*

LIN Quan (林 泉)[1], WANG Bo (王 波)[2], DU Yuan (杜 圆)[3],
WANG Xuezhi (王雪至)[3], LI Yuhua (李玉华)[1,**], CHEN Songcan (陈松灿)[2]

1. Department of Computer Science, Huazhong University of Science and Technology, Wuhan 430074, China;
2. Department of Computer Science, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China;
3. Department of Computer Science, Tsinghua University, Beijing 100084, China;

Abstract: Name ambiguity is a critical problem in many applications, in particular in online bibliography systems, such as DBLP, ACM, and CiteSeerx. Despite the many studies, this problem is still not resolved and is becoming even more serious, especially with the increasing popularity of Web 2.0. This paper addresses the problem in the academic researcher social network ArnetMiner using a supervised method for exploiting all side information including co-author, organization, paper citation, title similarity, author's homepage, web constraint, and user feedback. The method automatically determines the person number *k*. Tests on the researcher social network with up to 100 different names show that the method significantly outperforms the baseline method using an unsupervised attribute-augmented graph clustering algorithm.

Key words: disambiguating; pairwise classification; arnetminer

Introduction

When people search for their own papers in different search engines, they sometimes cannot find their own papers because they are included in someone else's publication lists, or someone else's papers are in their publication list. In both circumstances, one possible cause is that the authors share the same names. Also, one person may have several different names, due to name abbreviations and misspellings.

Most search engines have not done well distinguishing between different authors who share the same name. However people need accurate searches for their own papers. Name disambiguation is needed to improve the search performance. For example, when searching for a name in DBLP it gives all the papers owned by "this name", and actually includes the papers owned by several different people. Examples show that only about 26 out of 100 papers are actually owned by one person.

This confusion is quite important when evaluating faculty publications or investigating an author's impact when accurate judgments are difficult because of the difficulty in finding the papers owned by one person. This also happens when establishing a social network or finding connections between people. Searches can connect two people who do not have any relationship. For example, Chun Chen has a paper coauthored with Bing Liu, while Robert L. Grossman also has a coauthor Bing Liu. However, if the two "Bing Liu" are not the same, then there is probably no relationship between Chun Chen and Robert L. Grossman.

The solution to this problem is referred to as "name

Received: 2010-09-16; revised: 2010-10-10

* Li Yuhua and Lin Quan were supported by the National Natural Science Foundation of China (Nos. 70771043, 60873225, and 60773191). Chen Songcan and Wang Bo were supported by the National Natural Science Foundation of China (No. 60773061) and the Natural Science Foundation of Jiangsu Province (No. BK2008381). Du Yuan and Wang Xuezhi were supported by the National High-Tech Research and Development (863) Program of China (No. 2009AA01Z138)

** To whom correspondence should be addressed.
E-mail: idcliyuhua@hust.edu.cn; Tel: 86-18971074695

disambiguation" where information from papers (such as coauthors, citations, PDF source information, and titles) is used to distinguish different people who share the same name. When there are many papers (from about 100 to 800) owned by the same name, information extracted from the Internet can be used to cluster the papers that are actually owned by one person and distinguish papers that are written by different people.

People intuitively solve this problem by thinking that if two papers have the same coauthor, then they must be written by the same person. The authors' organization can also provide very useful information. The coauthor information can be very helpful, but can only cluster a fraction of all papers. Also, coauthor information can be inaccurate, e.g., Jing Zhang can refer to many people. Also, the author's organization information is not always available (especially papers published long ago) with 48% of papers not having organization information[1].

Several approaches have been proposed for name disambiguation in different domains, such as disambiguation on Encyclopedic Knowledge or Wikipedia Data[2,3], Citations[4-7], Web Pages[8,9], Email data[10], and Internet Movie Database[11].

Cucerzan[3] presented a large-scale system for the recognition and semantic disambiguation of named entities based on information extracted from Wikipedia and Web search results. Bunescu and Pasco[2] trained a disambiguation support vector machine (SVM) kernel to exploit the high coverage and rich structure of the knowledge encoded in an online encyclopedia.

Han et al.[5] proposed an unsupervised learning approach using the K-way spectral clustering method. They studied two types of feature weights, the usual "TFIDF" and the normalized "TF" ("NTF"). Then, they calculated a Gram matrix for each name dataset and applied the K-way spectral clustering algorithm to the Gram matrix to get their result. Tan et al.[6] developed a search engine based on a clustering method. They represented the features of each citation as relevant URLs from the search engine and weighted each citation by its IHF. Then, they computed the pair wise similarity of two citations using cosine similarity with hierarchical agglomerative clustering to derive the final cluster.

Two supervised methods in Han et al.[4] used supervised learning approaches to disambiguate authors in citations. One approach used the Naive Bayes probability model which is a generative model that captures all the authors' writing patterns using only positive training citations while the other approaches used SVM and the vector space representation of the citations with a discriminative model learning from both positive and negative training citations to distinguish between different authors' citations. Therefore, both models will predict whether a new citation belongs to a certain author. However, one drawback of supervised methods is their limited scalability with such features. Also, the training of thousands of models for all individuals in a large digital library is impractical. Elmacioglu et al.[12] presented a similar method.

Zhang et al.[13] proposed a constraint-based probabilistic model for semi-supervised name disambiguation. They formalized the name disambiguation problem into a constraint based probabilistic framework using Hidden Markov Random Fields. Then, they defined six types of constraints and employed the EM algorithm to learn the different distance metrics for different people.

Other methods have utilized unsupervised clustering. Mann and Yarowsky[9] used an unsupervised clustering method over the feature space of biographic facts. Wang et al.[14] proposed an approach for finding atomic clusters to improve the performance of existing cluster-based methods, but this cannot be expanded to unknown names, because the number of atomic clusters is predefined based on the answer. If the annotated answer is not known, the improvement will be limited.

Tang et al.[15] proposed an approach to extract academic information from intranets and how to filter noise.

This paper describes a combination of different approaches, Co-Author, Co-Org, Citation, Digital-Lib, Title Similarity, PDF File, and Homepage. In addition, to make accurate final decisions in spite of some inaccurate information, these features are combined using perceptions to adjust the weights of different features, instead of simply merging them together.

The coauthor and organization information are most useful; although still inadequate in most cases. Tests show that citations are also very useful. The citations were divided into self-citations, where a person cites his/her own paper, which are clustered together and double citations, i.e., in paper 1 person A has coauthor

B and in paper 2 person A cites a paper which has author B, which are clustered together. This is useful because B may be the major professor of A, and A will probably cite B in the future, even after graduation. Google searches were also used as a feature since two papers by the same person will appear on the same web page (after filtering out most of the noise pages), which can be found using Google Engine.

Similar titles may also indicate a relationship. Some papers lacking all the other features can only be clustered using title similarity. This idea is based on the assumption that one person probably has only one field of study which will be very different from those with identical names. For example, two titles with the same words "Component-Oriented Phased Array Radars" have a large probability that they are owned by one person. However, this may also introduce some errors, like two people doing "software engineering" at the same time. A TFIDF model was used to calculate the similarity, which is quite effective.

PDF files are also useful because they contain much information that cannot be found on the Internet. The problem is, however, that PDF files are difficult to download and programs searching for organization information can only search for the known organization to see whether it appears in the file, which is not very accurate or effective. Regardless, this still helps to improve the accuracy.

Homepages are also useful features that are very helpful but hard to find. Publication pages are most useful but they have many forms, so they are not easily found. Publication list pages can cluster many papers that cannot be clustered by merely using the other features.

Other approaches such as journals or conferences names (jConf) are not accurate and are difficult to obtain. About 40% of the resumes can be found for a given set of names, and they are all in publication lists or homepages. Since the homepage is already difficult to find, resumes (especially in the PHD format) are not used here.

1 Name Disambiguation

1.1 Problem definition

First of all, some formal notation is introduced for name disambiguation.

Given an author name a, let P denote the set of papers owned by this name, that is, $P=\{p_1, p_2, \cdots, p_n\}$ where n is the total number. Each paper p_i $(1 \leqslant i \leqslant n)$ has the 6 attributes listed in Table 1.

Table 1 Attributes for each paper p_i

Attribute	Description
p_i.author	Author list for paper p_i, $\{a_i^{(0)}, a_i^{(1)}, \cdots, a_i^{(m)}\}$
p_i.reference	Reference list for paper p_i
p_i.title	Title of paper p_i
p_i.jconf	Published venue of paper p_i
p_i.year	Published year of paper p_i
p_i.content	Content of paper p_i

Among these six attributes, p_i.authors, p_i.title, p_i.jconf, and p_i.year are always available for all the papers, while p_i.references might be only partially available for each paper and p_i.content is available for only some papers. The first author, $a_i^{(0)}$, is used as the primary author, and the others as the secondary authors. Besides the paper attributes, the author's organization a.organization is also used.

The name disambiguation task is as follows: Given an author name, a, and the paper list $P=\{p_1, p_2, \cdots, p_n\}$, if there are m different persons $\{t_1, t_2, \cdots, t_m\}$ sharing this name, then the task is to assign each publication p_i to its "owner" $t_j (1 \leqslant j \leqslant m)$, who actually wrote this paper.

1.2 Feature definition

There is much information which can be utilized for name disambiguation. First, define a set of features to be exploited for each paper pair (p_i, p_j) for a given author name, a, as

$$R = \{r_1, r_2, \cdots, r_k\},$$

where each r_i denotes one feature capturing one relationship between papers p_i and p_j.

The seven different features as shown in Table 2 will be explained in the next subsection.

All the features are defined over all the papers sharing the same primary author. For each feature in Table 2, the feature value is binary, that is, if the description is true, then the value is 1; otherwise 0. If the feature value is 1, then that feature indicates that the two papers are probably written by the same author.

These features were chosen to simulate how humans think. A person having two papers will judge whether they are written by the same author using the same features.

Table 2 Feature definition for paper pair (p_i, p_j)

R Feature	Description
r_1 Co_Author	Exist $u,v>0$, $a_i^{(u)}=a_j^{(v)}$
r_2 Co-Org	a_i.organization= a_j.organization
r_3 Citation	p_i and p_j has citation links
r_4 Title-Similarity	a_i.title and a_j.title are similar
r_5 Homepage	p_i and p_j appear in someone's homepage
r_6 Digital-Lib	p_i and p_j appear in the same Apringer/ Citeseerx page
r_7 PDF File	a_i.organization appears on the PDF format file for p_j and vice-versa
r_8 Co-Conference	pi.jconf = pi.jconf
r_9 Resume	p_i and p_j appear together in someone's resume

1.3 Feature analysis

Co-Author The feature value is 1 if p_i and p_j share a secondary author, otherwise 0.

The reason to define such a feature is that people prefer to write papers with a certain group of coauthors with similar expertise or who work nearby while other people with the same name are unlikely to have the same coauthors. Thus, the system tries to disambiguate a name by the people around them.

The Co-Author feature first checks the secondary author sets of p_i and p_j to see whether they share the same author name. That is, the system checks the two sets $\{a_i^{(u)}\}$ and $\{a_i^{(v)}\}$ where $u,v>0$, and if $\exists\tilde{u},\tilde{v}$, $a_i^{\tilde{u}}=a_i^{\tilde{v}}$, then papers p_i and p_j are probably written by the same author.

As described in the introduction, this method may introduce some errors, since the co-authors $a_i^{\tilde{u}}$ and $a_i^{\tilde{v}}$ may be different people with the same name.

Co-Org The feature value is 1 if the primary authors of papers p_i and p_j are from the same organization; otherwise, it is 0.

The reasoning is that if the two primary authors of two papers have the same name and are from the same organization, then these two papers are probably written by the same author.

This should be quite easy to check whether $a_i^{(0)}$.organization and $a_j^{(0)}$.organization are the same. However, there are various ways to write the address for an organization. For example, those four addresses: (1) Computer Science Department, Brigham Young University, Provo, UT 84602-6576, (2) Computer Science Department, Brigham Young University, Provo, UT 84602-6576, USA, (3) Brigham Young Univ. Provo, Provo, and (4) Brigham Young University actually stand for the same organization even though they are different.

Furthermore, similar addresses may actually represent different organizations. For example, (1) Dept. of Computer Science, Tsinghua University, Beijing 100084, China, and (2) Dept. of Autom., Tsinghua Univ., Beijing 100084, China, represent two different departments of Tsinghua University. Therefore, an Address-Extraction method was developed to solve this problem. Usually an address consists of several phrases. The Address-Extraction method tries to divide an address into eight levels as shown in Table 3.

Table 3 Address phase level

Level	Level name	Example
1	Country	USA, China, Singapore
2	City	Beijing, Salt Lake City, New York
3	Company	American Can Company
4	University	City University of Hong Kong
5	School	School of EECS
6	Department	Department of Psychiatry
7	College	College of Mechanical
8	Laboratory	DoCoMo USA Labs

Then the address is represented as a vector: Org{level_1, $\text{level}_2,\cdots,\text{level}_3$}. For example, Computer Science Department, Brigham Young University, Provo, UT 84602-6576, USA, can be represented as {USA, UT, null, Brigham Young University, null, Computer Science Department, null, null}. Then two addresses O_1 and O_2 stand for the same organization if they satisfy:

(1)!$\exists i$ that $O_i.\text{level}_i \neq O_2.\text{level}_i$, when $O_i.\text{level}_i \neq$ null and $O_2.\text{level}_i \neq$ null;

(2)!$\exists i,j\in\{1,2,\cdots,8\}$ such that $O_i.\text{level}_i \neq$ null and $O_2.\text{level}_i \neq$ null.

Citation The feature value is 1 if there is a direct or indirect citation between the papers p_i and p_j; otherwise, it is 0.

The reasoning is that a researcher is more likely to cite his own papers or those written by his coauthors. For example, once a student has coauthored a paper with his professor, then he will probably cite this paper or other papers written by his professor in the future (even after he has graduated).

This Citation feature takes into consideration two

kinds of citations between two papers p_i and p_j.

(1) **Direct citations** Direct citations occur when one paper cites another paper, that is, p_i cites p_j. For this kind citation, the system merely searches p_i.references to check whether p_i cites p_j.

(2) **Indirect citations** The system looks for two step indirect citations where p_i cites a third paper p_q which has an author called b and b is one of the secondary authors of p_j. Then, there is an indirect citation between p_i and p_j. For instance, consider two papers written by Michael Gleicher. Let p_1 denote the paper "Constraint-based motion adaptation", and p_2 denote "Texture-consistent shadow removal". There is then a third paper p_3, "Multiple animated characters motion fusion", which cites p_1 and contains Feng Liu as an author who is also a secondary author of p_2. Therefore, papers p_1 and p_2 are probably written by the same author.

Title Similarity The feature value is 1 if the title similarity between two papers p_i and p_j is larger than a threshold Sim(p_i.title, p_j.title)>γ); otherwise, the value is 0.

The reasoning is that the topic (research interest) of one author will most likely be very different from other people with the same name. For example, two titles with the same words "Component-oriented phased array radars" have a high probability of being written by the same author.

The Title-Similarity feature uses the TF/IDF model to calculate the title similarity.

First, a stop list is used to filter out frequent words like "for", "and", and "or". Then all the words of all the titles are added to set $S=\{d_1, d_2,\cdots,d_n\}$ after filtering. Then

$$\mathrm{Sim}(p_i.\mathrm{title}, p_j.\mathrm{title})=\cos(\mathbf{vec}_i, \mathbf{vec}_j) \quad (1)$$

where $\mathbf{vec}_i=[f_{i1}, f_{i2},\cdots, f_{in,}]$ is the feature vector of paper i. Let N denote the total number of titles, N_k denote the number of titles having d_k appearances, and t_{ik} denote the times having d_k in p_i.title. Then define $f_{ik}=t_{ik}\cdot\log\left(\frac{N}{N_k}\right)$.

The title similarity is calculated in this way since when two titles share more words which occur less frequently in all titles, then these words are "unique" or "representative" and the two titles are similar.

However, this may also introduce some errors, as in the case when two people do research work on "Software Engineering" and "Software Engineering" appears only two times in all the titles.

The parameter γ is chosen using cross-validation. A small γ improves the recall but lowers the precision, because it clusters a certain amount of papers that cannot be clustered by other features. However, a small γ can increase errors as in the "Software Engineering" example.

Homepage The feature value is 1 if two papers appear on an author's homepage; otherwise, the value is 0.

The reasoning is that the author will list all his papers on his homepage, which will show whether these two papers are his. The Homepage feature first automatically searches for publication pages from the authors' homepages, then searches for the titles of the two papers in the publication lists. If the titles of the two papers appear on the same page then the two papers are probably written by the same author.

Preliminary statistics show that 64.9 percent of authors' homepages are found by the algorithm. Tests then show that this feature significantly improves the performance. For example, disambiguating the name "Thomas Wolf" with the help of the homepage of Thomas Wolf at Brock University significantly improves the recall from 10.31% to 64.40%.

Digital-Lib The feature value is 1 if two papers appear on the same Springer or CiteSeerx web page; otherwise, the value is 0.

The reasoning is that when two papers with the same primary author appear on the same digital library web page (only the Springer or CiteSeerx web pages were used.), that is, these two papers are cited by a third paper, then the two papers are probably about the same topic. Moreover, because they have the same primary author, these two papers are probably written by the same author.

The Digital-Lib feature uses Google to generate the feature using "p_i.title p_j.title Springer" or "p_i.title p_j.title CiteSeerx" as queries for Google searches. If one digital library web page is found containing these two titles, then the feature value is 1.

The biggest drawback of this feature is that it is very time-consuming to search for every pair of papers in Google. To reduce the cost, other features are used to first select the most disambiguated paper pairs before searching Google.

PDF File The feature value is 1 if $a_i^{(0)}$.organization appears in the PDF file for paper p_j; otherwise, the value is 0.

The reasoning is that most PDF files contain information about the authors such as organization name and address. Thus, when the organization information for an author cannot be found from other web pages, it may be found in the file properties.

However, this approach for getting the affiliation of an author is very difficult and there is yet no efficient method for this.

The PDF File feature gets the first l letters from the PDF format file of one publication and tests whether those letters contain the first author's organization from another publication. If so, the system clusters those two papers.

The system uses $l = 3000$, which is generally large enough to identify the author's organization from the PDF file.

The key difficulty in matching an organization's name in the first l letters is that this requires conversion of the PDF format into a TXT file, where the file may have some unrecognizable codes, which will confuse the matcher. Thus, the system must first remove these codes to achieve the correct transformation.

In addition, the result may match the organization of a secondary author in the first publication, rather than the intended author.

This method is very useful when other features are not effective, but it works only when the PDF file is available for a publication and the organization information is also available.

Co-Conference The feature value is 1 if papers p_i and p_j are published in the same journal or conference; otherwise, the value is 0. Here, if two papers are published at one conference with the same author, then these two papers probably belong to one author.

However, preliminary tests show that this feature is not as effective as expected. For example, for the name "Philip J. Smith", the Co-jConf precision is 85.7%, which is much less than 100% and its recall is only 3.6%, which means this feature does not contribute much to the algorithm. Similarly, the precision for "Feng Liu" using Co-jConf is only 46.9% and the recall is 6.51%.

Resume The feature value is 1 if the titles of the two papers are included in the same resume; otherwise, its value is 0. Thus, titles in the same resume are clustered together. The system converts the resume which is normally a PDF file into text file and then searches for two titles in this text. Resumes, like homepages, can be quite useful. Sometimes resumes are even more helpful since people tend to put more detailed information about their own publication into their resume, while homepages often have just a few representative publications.

An author's resume normally contains all his information such as his publication list and organization changes. Thus, this is very useful for disambiguating people by providing complementary information to the other features. For example, if two authors have the same name, the same research interests, but different organizations, then his work experience in his resume will list his organization changes track which can be used to further identify whether the names are the same person.

However, resumes cannot be easily found. Tests found about 40% of the desired resumes all in publication lists or homepages. Since the homepage is already difficult to locate, the resumes (especially in the PDF format) were not used as this algorithm.

Summary Since jconf is not accurate and resumes are hard to find, only 7 features, Co-Author, Co-Org, Citation, Digital-Lib, Title-Similarity, PDF-Content, and Homepage, r_1 to r_7 were used.

2 Algorithm

The algorithm first needs enough samples for training from various pairs of papers with all or some of the 7 features. Then, the features are transformed into number pairs, as the inputs to the perceptrons as shown in Fig. 1, where $\boldsymbol{w} = [w_1, w_2, \cdots, w_N]$ is weights of features.

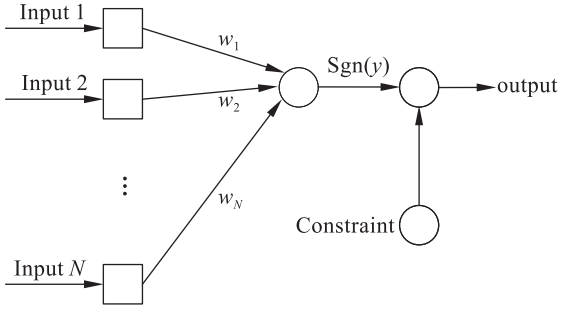

Fig. 1 Use of perceptron in name disambiguation

The number of features determines the number of inputs N, and the number of outputs is $M = 2$, either +1(represents the same cluster) or −1(represents

different clusters). Besides the features, constraints are also used to force the output to be +1, which means that the system is very confident about the results of these constraints.

The perceptron analyses the input $\boldsymbol{x}$ using $\hat{y} = \boldsymbol{\omega}^{\mathrm{T}} x + b$, where $\hat{y}$ is the output.

The objective function is defined as

$$L = \frac{1}{3} \| y^{\text{true}} - \hat{y} \|^2 \tag{2}$$

The weights are adjusted to minimize this objective function. Then, compute

$$\frac{\partial L(n)}{\partial w_i} = (y^{\text{true}} - \hat{y}(n))(-\hat{y}'(n)) \tag{3}$$

as the estimate of the gradient. Then,

$$\hat{y} = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x} + b = \sum_{i=1}^{n} x_i w_i + b \tag{4}$$

From Eq. (4) we can get $\hat{y}'(n) = x_i(n)$.

Defing η as the learning rate, then each time the output $\hat{y}$ is computed, the weights are updated as

$$Dw_i(n) = -\eta \frac{\partial L(n)}{\partial_{w_i}} \tag{5}$$

So after the n-th iteration,

$$w_i(n+1) = w_i(n) + \Delta w_i(n) = w_i(n) + \eta(y^{\text{true}} - \hat{y}(n)) x_i(n) \tag{6}$$

Or a momentum constant α can be added to memorize the old weights:

$$w_i(n+1) = w_i(n) + \eta(y^{\text{true}} - \hat{y}(n)) x_i(n) + \alpha w_i(n).$$

This corresponds to the objective function:

$$L = \frac{1}{2} \| y^{\text{true}} - \hat{y} \|^2 + \frac{1}{2} \lambda \bullet \| w \|^2$$

The stopping criteria is the maximum number of iterations on the change in the objective function which is sufficiently small in an epoch. The algorithm pseudo code is shown as follows.

Algorithm
Input: Samples with N features ;
Output: result +1 or −1;
Initialization: $\boldsymbol{w} \leftarrow 1$;
For each input sample, repeat:
    Compute the output $\hat{y}=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b$;
    According to the error, adjust the weight:
    $w_i(n+1) \leftarrow w_i(n)+\eta(y^{\text{true}}-\hat{y}(n))x_i(n)$
Until the stopping criteria is reached.
Compute the output of the perceptron.
Combine the constraints with the output:
if none of the constraints say the output should be +1
    then result = sgn(output);
else
    result = +1

The constraints which can force the output to be +1 are inputs, but without any weights. If at least one of the constraints say the output should be +1, regardless of the output of the perceptrons, the output is set to +1.

3 Test

The algorithm is very general and can effectively solve the author names disambiguation problem. This section describes an empirical study to test the algorithm's effectiveness.

3.1 Tests

Data sets The algorithm can be applied to various data sets. The initial tests need Arnetminer.org[15], which collects academic publication data from four different online digital library data sets: the DBLP, IEEE, ACM, and Springer databases.

- DBLP. 1 300 000 publications.
- IEEE. 450 223 publications from www.ieee.org.
- ACM. 1 343 442 publications, 3 687 675 citation relationships, with 585 826 publications added to the DBLP data set.
- Springer.

The data set used for the name disambiguation included 1 645 632 authors, 2 306 521 publications, and 1 343 442 citations. The tests focused on assigning papers to authors with identical names.

The algorithm was tested on 41 real names in the DBLP database. 20 human annotators were used to separate the papers with each paper labeled with a number indicating the actual author. The labeling was based on all the selection information including citation, coauthor, organization, and conference name. The remaining disagreements on the annotations were resolved using "majority voting". The 41 names are listed in Table 4, together with the number of authors and the number of papers and the authors having the most papers and the authors having least number of papers.

Evaluation measures Pair-wise measures were used to evaluate the name disambiguation results and for comparison with baseline methods. The pair-wise measures were based on traditional information retrieval measures adapted for evaluating disambiguation. The disambiguation result of paper pairs has two kinds that are written by same author and by different authors. Combined with two kinds of real status, here are

Table 4 Statistics of the data set

Name	Number of publications	Number of people
Michael Smith	38	24
Philip J. Smith	33	3
Yoshio Tanaka	43	3
Yang Yu	72	20
Hui Yu	32	22
Qiang Shen	70	3
Ping Zhou	36	18
Michael Lang	24	6
Manuel Silva	74	7
Charles Smith	7	4
Eric Martin	85	5
Kai Zhang	66	24
Fei Su	40	4
Robert Schreiber	59	2
Hiroshi Tanaka	43	8
Satoshi Kobayashi	38	6
Lei Jin	20	8
David Jensen	53	4
Thomas Wolf	36	9
Koichi Furukawa	77	3
Thomas Tran	16	2
Thomas Hermann	47	9
Yun Wang	57	22
Cheng Chang	27	5
Gang Luo	47	9
Bing Liu	215	25
R. Ramesh	46	9
David E. Goldberg	231	3
Feng Pan	73	15
Rakesh Kumar	96	12
Thomas D. Taylor	4	3
Richard Taylor	35	16
Jim Gray	200	9
Juan Carlos Lopez	36	1
Sanjay Jain	217	5
Ajay Gupta	36	9
Shu Lin	76	2
Michael Siegel	54	6
Daniel Massey	43	2
Michael Wagner	71	15
David C. Wilson	65	5

four statuses which are (1) true positive (tp): paper pairs are written by the same author and the disambiguation result is right; (2) false positive (fp): paper pairs are written by different people while the disambiguation thinks they are written by the same author; (3) true negative (tn): paper pairs are written by different people and disambiguation results also think so; and (4) false negative (fn): paper pairs are written by the same author while disambiguation thinks they are written by different author. The definitions are

$$\text{Precision} = \frac{\text{tp}}{\text{tp} + \text{fp}} \tag{9}$$

$$\text{Recall} = \frac{\text{tp}}{\text{tp} + \text{fn}} \tag{10}$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

Baseline methods The baselines used the bias classifier learned from each single feature and SA-Cluster which is a graph clustering method[16] with the coauthor relationship used as the edge and all the other relationships used as the attribute features.

3.2 Test results

The performance of the current algorithm for each author is listed in Table 5 with the performance for each baseline and the current algorithm shown in Section 2 in terms of the Recall, Precision, and F-measure. The results show that the current method significantly outperforms all the baselines.

Figure 2 also shows the contributions of each feature. Co-Author has the highest F-measure because the author names in each paper are complete, compared to the other features. Since two authors with the same name may coauthor with the same person and the coauthor name may also be ambiguous, the precision of this feature is not 100%.

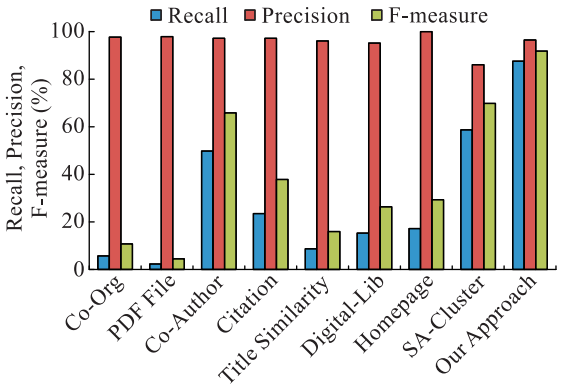

Fig. 2 Performances of all the baselines and our approach

Citation is very useful information because people tend to cite their own papers if they have published related papers. Since the Citation information is crawled

Table 5 Results for 41 names

Name	Recall (%)	Precision (%)	F-measure (%)
Robert Schreiber	72.81	100.00	84.27
Michael Smith	70.83	100.00	82.93
Hiroshi Tanaka	31.13	100.00	47.48
Satoshi Kobayashi	73.41	92.05	81.68
Philip J. Smith	83.83	100.00	91.21
David E. Goldberg	98.26	99.12	98.69
Yoshio Tanaka	85.48	94.64	89.83
Hui Yu	94.74	85.71	90.00
Feng Pan	56.55	94.37	70.72
Qiang Shen	90.37	100.00	94.94
Lei Jin	100.00	100.00	100.00
Yang Yu	89.53	99.25	94.13
Ping Zhou	82.00	100.00	90.11
Rakesh Kumar	97.49	100.00	98.73
David Jensen	87.43	95.56	91.31
Thomas Wolf	33.33	89.36	48.55
Michael Lang	53.66	100.00	69.84
Thomas D. Taylor	100.00	100.00	100.00
Manuel Silva	93.55	90.48	91.99
Charles Smith	100.00	100.00	100.00
Koichi Furukawa	72.72	96.93	83.10
Thomas Tran	56.04	100.00	71.83
Thomas Hermann	71.30	100.00	83.25
Richard Taylor	67.82	100.00	80.82
Jim Gray	85.72	93.76	89.24
Juan Carlos Lopez	89.05	100.00	94.21
Sanjay Jain	97.74	100.00	98.86
Ajay Gupta	63.03	100.00	77.32
Shu Lin	79.93	100.00	88.84
Michael Siegel	92.87	100.00	96.30
Eric Martin	98.31	100.00	99.15
Yun Wang	67.65	100.00	80.70
Kai Zhang	82.55	100.00	90.44
Cheng Chang	83.95	100.00	91.28
Daniel Massey	95.24	100.00	97.56
Fei Su	100.00	100.00	100.0
Michael Wagner	84.54	93.97	89.01
David C. Wilson	89.50	100.00	94.46
Gang Luo	100.00	98.41	99.20
Bing Liu	93.88	88.99	91.36
R. Ramesh	68.12	100.00	81.04

from the internet, it is not complete, and the probability of direct citations and indirect citations is quite low, so the recall is low. Also some errors exist because indirect citations use a coauthor name which may be an ambiguous name.

Homepage has the third highest F-measure and a precision is 100%. Since the homepage normally contains only the owner's papers. However owners usually only put their best papers on their homepages and not all author's homepages we can be found on the internet, so the recall is very low.

The Recall of Co-Org is also very low because the organization information is very incomplete. However the precision is very high (99.6%) because at the same organization two authors merely have the same name.

Digital-Lib supplements Citation data while PDF File supplements Co-Org. This information is difficult to fetch from the internet and some PDF files stand as images and they cannot be fetched, so the F-measure is low.

Title Similarity gives very good performance. Because the title information is complete and an author usually publishes a series of papers in one direction. These authors tend to name their papers in similar ways.

4 Conclusions

This paper presents an algorithm for Name Disambiguation using additional information. The algorithm combines 7 different features, Co-Author, Co-Org, Citation, Digital-Lib, Title Similarity, Homepage, and PDF File. Among these features, Co-Author and Title Similarity are the most effective and easy to use. Organization information and homepage information can be very useful, once they are found but they are difficult to find. The other features, Citations, Digital-Lib, and PDF File introduce some errors, as people may cite someone who has a same name so that double citations can be wrong in some cases. Digital-Lib can introduce errors when a noisy page is included and there is some inaccurate information in PDF files because the publisher information is sometimes also organization information.

Despite the inadequate and inaccurate information, use of all these features gives the best results. Test with a large database having more than one million authors and two million papers with result based on Recall, Precision, and F-measure show that the system can distinguish different people with identical names in real large databases.

Future work will exploit more useful features like

user feedback in a framework combining all the features in a more effective manner. This function can also be used in other applications for mining the advisor-advisee relationship, finding experts, and people searches.

References

[1] http://portal.acm.org, 2010.

[2] Bunescu C, Pasca M. Using encyclopedic knowledge for named entity disambiguation. In: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics. Trento, Italy, 2006: 9-16.

[3] Cucerzan S. Large-scale named entity disambiguation based on Wikipedia data. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Prague, Czech Republic, 2007: 708-716.

[4] Han H, Giles L, Zha H, et al. Two supervised learning approaches for name disambiguation in author citations. In: Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries. Tuscon, USA, 2004: 296-305.

[5] Han H, Zha H, Giles L. Name disambiguation in author citations using a k-way spectral clustering method. In: Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries. Denver, USA, 2005: 334-343.

[6] Tan F, Kan Y, Lee D. Search engine driven author disambiguation. In: Proceedings of the 6th ACM/IEEE-CS Joint Conference. Chapel Hill, USA, 2006: 314-315.

[7] Yin X, Han J, Yu S. Object distinction: Distinguishing objects with identical names. In: Proceedings of the 23rd International Conference on Data Engineering. Istanbul, Turkey, 2007: 1242-1246.

[8] Bekkerman R, McCallum A. Disambiguating web appearances of people in a social network. In: Proceedings of the 14th International Conference on World Wide Web. Chiba, Japan, 2005: 463- 470.

[9] Mann S, Yarowsky D. Unsupervised personal name disambiguation. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL. Edmonton, Canada, 2003: 33-40.

[10] Minkov E, Cohen W, Ng Y. Contextual search and name disambiguation in email using graphs. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Washington D. C., USA, 2006: 27-34.

[11] Malin B, Airoldi E, Carley M. A network analysis model for disambiguation of names in lists. *Comput. Math. Organ. Theory*, 2005, **11**(2): 119-139.

[12] Elmacioglu E, Fan Y, Su T, et al. Psnus: Web people name disambiguation by simple clustering with rich features. In: Proceedings of the 4th International Workshop on Semantic Evaluations. Prague, Czech Republic, 2007: 268-271.

[13] Zhang D, Tang J, Li J, et al. A constraint-based probabilistic framework for name disambiguation. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management. Lisbon, Portugal, 2007: 1019-1022.

[14] Wang F, Li J, Tang J, et al. Name disambiguation using atomic clusters. In: Proceedings of the Ninth International Conference on Web-Age information Management. Washington, D. C., USA, 2008: 357-364.

[15] Tang J, Zhang J, Yao L, et al. ArnetMiner. In: Proceeding of the 17th International Conference on World Wide Web. Beijing, China, 2008: 990-998.

[16] Zhou Y, Cheng H, Yu X. Graph clustering based on structural/attribute similarities. *VLDB Endow*, 2009, **2**(1): 718-729.