A Fast Method Based on Multiple Clustering for Name Disambiguation in Bibliographic Citations

Yu Liu, Weijia Li, Zhen Huang, and Qiang Fang
School of Software, Dalian University of Technology, Economy and Technology Development Area, Dalian, 116620, China. E-mail: yuliu@dlut.edu.cn; lwjlaser@gmail.com; kobe_hz@163.com; fangqiang01@gmail.com

Name ambiguity in the context of bibliographic citation affects the quality of services in digital libraries. Previous methods are not widely applied in practice because of their high computational complexity and their strong dependency on excessive attributes, such as institutional affiliation, research area, address, etc., which are difficult to obtain in practice. To solve this problem, we propose a novel coarse-to-fine framework for name disambiguation which sequentially employs 3 common and easily accessible attributes (i.e., coauthor name, article title, and publication venue). Our proposed framework is based on multiple clustering and consists of 3 steps: (a) clustering articles by coauthorship and obtaining rough clusters, that is fragments; (b) clustering fragments obtained in step 1 by title information and getting bigger fragments; (c) and clustering fragments obtained in step 2 by the latent relations among venues. Experimental results on a Digital Bibliography and Library Project (DBLP) data set show that our method outperforms the existing state-of-the-art methods by 2.4% to 22.7% on the average pairwise F1 score and is 10 to 100 times faster in terms of execution time.

Introduction

With the development of the Internet and "cloud computing," the amount of data in online literature management systems shows a trend of explosive growth. One of the challenging problems in such systems is that articles written by different authors but with the same name are mixed together, that is the so called *name ambiguity*. The problem of name disambiguation is important, in the sense that author name search is one of the most used query methods in literature databases—name search constitutes 30% of the total number of searches as described in Deerwester, Dumais, Furnas, Landauer, and Harshman (1990)—whereas name ambiguity affects the purity of query results, for example, about 300 male names are used by 114 million Americans. That is 78.4% of the total male population. In the Digital Bibliography and Library Project (DBLP), 61 articles are written by 16 different authors all named "Michael Wagner" and there are over 200 "Wei Wang." Therefore, how to categorize ambiguous names has a significant impact on information retrieval, information integration, bibliometrics, etc.

A popular method to deal with name disambiguation in literature management is to put articles written by different authors with the same name into different clusters, so that each cluster belongs to one author, and each author's articles are gathered in one cluster.

Although related methods have yielded good results in experiments, few of them can be applied to real systems for the following two reasons: (a) requirements of extra attributes, such as institutional affiliation, research area, address, etc. (D'Angelo, Giuffrida, & Abramo, 2011), which are difficult to obtain in practical applications (Ferreira, Gonçalves, & Laender, 2012); (b) high computational complexity, which may be beyond the power of existing systems.

In view of the problems faced by the existing name disambiguation methods, we propose an algorithm called fast multiple clustering (FMC) for name disambiguation. Although each step is based on some well-known approaches, FMC is a novel coarse-to-fine multiple clustering framework on the whole, which can achieve better results with fewer attributes (only employing the three most common attributes of articles: article title, authorship, and publication venue) and lower cost of time. The characteristics of employing fewer attributes and lower computation cost improve its applicability in practice. In addition, instead of simply using venue information for solving name as in ambiguity in previous methods, we propose a novel model to mine and represent the latent relations among venues. The latent relations among venues can help to further improve the effectiveness of name disambiguation.

Received May 28, 2013; revised November 9, 2013; accepted November 11, 2013

© 2014 ASIS&T • Published online 5 June 2014 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/asi.23183

FMC consists of three steps: first, given the article list by authors with the same name, FMC groups articles into small clusters, also called fragments, by coauthorship where each cluster represents the articles of one author; second, FMC continues to cluster the fragments obtained from the previous step by correlating titles to reduce the number of fragments and increase the number of articles in the fragments; finally, the algorithm further tunes the clustering via the latent relations among venues, where the articles written by authors with the same name have been grouped together under the actual authors. Our experimental results show that FMC gets the best pairwise F1 score among four algorithms and reduces the runtime by 10 to 100 times as compared with the second-best algorithm Categorical Sampling Likelihood Ratio (CSLR) (Li, Cong, & Miao, 2012).

The remainder of this article is organized as follows. The second section introduces basic notations and formulates the problem of name disambiguation followed by the proposed FMC algorithm in the third section. The fourth section represents and analyzes the experimental results. The fifth section reviews related work. The final section presents our conclusion and directions for future work.

Problem Formulation

In literature management systems, each article contains three attributes: title, author, and venue. Given a set of persons with the same name α, denoted by A, the set of articles with α as an author is denoted by $S = \{p_1, p_2, p_3, \ldots p_n\}$, where $|S|$ is the number of articles in S. For a certain author $a_i \in A$, the set of articles published by a_i is denoted by S_i where $S_i \subseteq S$. We use A_p to denote the author list of article p. Then the coauthor list of article p is the set of authors in A_p excluding a_i, that is $A_p \backslash a_i$, and is represented by co(p). Hence the coauthor set of authors named a_i is $\mathrm{co}(S_i) = \bigcup_{i=1}^{n} \mathrm{co}(p_i)$. In addition, we also represent the set of titles for articles in S_i with $T(S_i)$ and the publication venues of articles in S_i with $V(S_i)$, respectively. The objective of name disambiguation is to find a partition $\{S_1, S_2, \ldots ,S_k\}$, such that S_i is composed of articles written by a single author a_i, where k is the number of authors sharing the same name α. The partition function is parameter by the coauthor set $co(S_i)$, the title set $T(S_i)$, and the venue set $V(S_i)$, that is $S_i = f(\mathrm{co}(S_i), T(S_i), V(S_i))$.

A Fast Method Based on Multiple Clustering for Name Disambiguation

Overview of Method

Literature management systems, such as DBLP, CiteSeerX, CiteULike, and Research Gate, record and manage the metadata of articles. The metadata consist of attributes such as: title, authors, abstract, venue, volume, number, pages, published time, etc. The number and type of attributes in the metadata are not fixed. The metadata in different systems vary with different business requirements.

To address the challenges, a practical name disambiguation method should contain the following features: (a) good disambiguation results; (b) wide application range, which means the method should have light dependence on attributes; (c) high efficiency, which means the method with less computation and lower run time is effective for systems with a large amount of data.

Previous name disambiguation methods have utilized different attributes, which bring challenge to name disambiguation methods. Tang, Fong, Wang, and Zhang's (2012) Arnetminer requires authors, title, venue, year, abstract, reference, and relations among citations. The method proposed by D'Angelo et al. (2011) uses not only publication year, authors, authors' addresses, and subject category of articles, but also all the researchers' names in Italy and their corresponding scientific disciplinary sectors and universities. These attributes are very hard to obtain for certain digital libraries, and even harder, or impossible, for literature systems like DBLP, CiteSeerX and the systems that are based on Web 2.0 such as CiteULike and Research Gate. Methods being designed for certain systems result in non-applicability for other applications. For attributes which are difficult to obtain, we intend to use three common attributes (title, authors, and venue) existing in almost all systems in the proposed FMC.

The importance of the three attributes is decided by their impact on the name ambiguity problem. First, as a scholar usually has several related research fields and collaborates with several comparatively stable researchers in each field, coauthors usually provides stronger evidence than the other attributes (Cota, Ferreira, Nascimento, Gonçalves, & Laender, 2010; Fan, Wang, Pu, Zhou, & Lv, 2011; Kang et al., 2009; Li et al., 2012; Tang et al., 2012). We extract coauthors from article set and cluster the articles by coauthorship in the first step. Second, as in any research field, there exist some term or words that are often used in titles. Scholars can have a habit of using some specific words in the title, and title information provides evidence to cluster the fragments. In addition, the more articles that are in fragments, the stronger the evidence of venue information, so it is better to increase the number of articles in fragments before using the venue information. Therefore, we merge clusters by title in the second step. Finally, the selected by researchers in different fields show different characteristics, such as venue titles, number of venues, distributions of venues, etc. In addition, each research field has several corresponding venues, and each venue only adopts certain articles in its related fields (e.g., researchers interested in the database usually submit their works to venues for data mining. Meanwhile, some venues of biology accept articles of data mining in the area of bioinformatics.). These latent relations among venues can help us to better solve the problem. Therefore, in the third step, we use venue sets extracted from fragments to guide merging.

DOI: 10.1002/asi

According to the given analysis and referring to the methods proposed by Cota et al. (2010) and Li et al. (2012), we propose a method based on multiple clustering to solve the problem described in Algorithm 1. The proposed algorithm consists of three steps. We use R_j to denote the clustering results obtained in Step j where Step 1 is merging by coauthorship; Step 2 is merging by title; Step 3 is merging by venue. In terms of characteristics of different attributes, each step uses a suitable model, similarity function, and clustering method. The details of the three steps are as follows.

ALGORITHM 1. Name disambiguation algorithm based on multiple clustering

```
Input: paper set $S$ and given name $\alpha$
Output: result $R_3$
extract co($S$) from $S$;
$R_1 \leftarrow$ ClusterByCoauthor ($S$, co($S$));
$R_2 \leftarrow$ ClusterByTitle ($R_1$);
$R_3 \leftarrow$ ClusterByVenue ($R_2$);
```

Merging by Coauthorship

Coauthorship is viewed as the most important attribute in resolving name disambiguation problems. In this article, we tackle the problems using an accurate and efficient method based on graph related algorithms. In this section, we will introduce this method with an example.

Figure 1 shows the article set $S = \{p_1, p_2, p_3, p_4, p_5\}$ of name α and its corresponding coauthor set co(α) = $\{a_1, a_2, a_3, a_4, a_5\}$. An article and an author connected by a line means that the author is a coauthor of the article. For example, co(p_1) = $\{a_1, a_2\}$ The relations between articles and coauthors in Figure 1 can be represented by a matrix as in Formula (1).

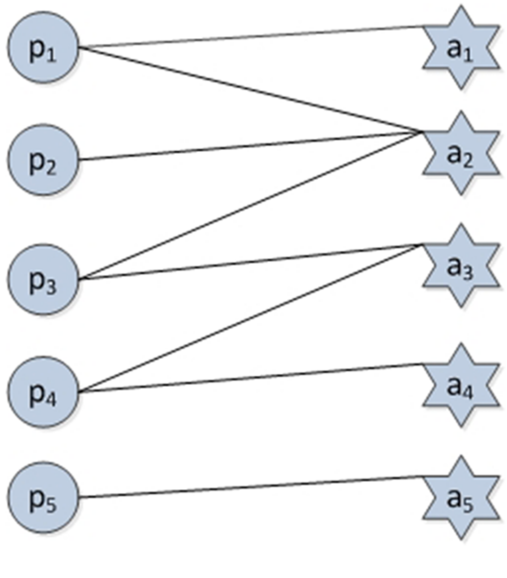

FIG. 1. An example of relations between articles and coauthors. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

$$M = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{ccccc} a_1 & a_2 & a_3 & a_4 & a_5 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \tag{1}$$

Where each row represents an article and each column represents an author. If article i has one coauthor j, then $M_{ij} = 1$, otherwise $M_{ij} = 0$.

Lemma 1. p_i and p_j are two articles written by α. If they shared at least one coauthor, such as in Formula (2), then they are written by the same author.

$$\text{co}(p_i) \cap \text{co}(p_j) \neq \varnothing \tag{2}$$

For example, in Figure 1, co(p_1) = $\{a_1, a_2\}$, co(p_2) = $\{a_2\}$. Therefore, they shared coauthor a_2. Combining Formula (1) and (2), we can get relations between articles, as in Formula (3).

$$MR = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{ccccc} p_1 & p_2 & p_3 & p_4 & p_5 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \tag{3}$$

Where each row or column represents an article and if $M_{ik} = M_{jk} = 1 (1 \le k \le 5)$, which means that co($p_i$) $\cap$ co(p_j) $\neq \varnothing$, we set $MR_{ij} = 1$. Otherwise, set $MR_{ij} = 0$. MR is a symmetric matrix which shows the coauthor relations among articles, and its size is $|S| \times |S|$, in this case, 5×5. The symmetry is obvious because if article i has a coauthor relation with article j then article j has the same relation with article i.

According to the Floyd-Warshall algorithm (Aini & Salehipour, 2012; Cormen, Leiserson, Rivest, & Stein, 2001) and MR, a reachability matrix, as in Formula (4), can be computed. In Formula (4), Rij = 1 means article i has a relation with article j through coauthorship. The procedure is as follows. First, we can get the distance between each pair of articles by using the Floyd-Warshall algorithm. If the distance between two articles is larger than 0, which means one article gets a relation to the other article, we set the corresponding value 1 in reachability matrix R. Then all the reachable articles in the matrix are written by the same author. Therefore, following Lemma 1 and Formula (4), we

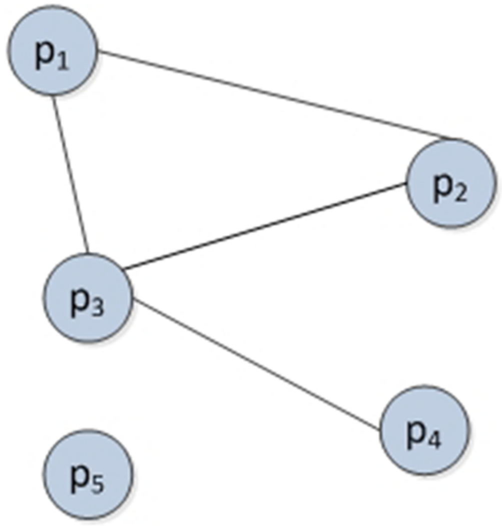

FIG. 2. The result of author merge. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

have the result shown in Figure 2 which shows the result $R_1 = \{\{p_1, p_2, p_3, p_4\}, \{p_5\}\}$.

$$R = \begin{array}{c|ccccc|} & p_1 & p_2 & p_3 & p_4 & p_5 \\ p_1 & 1 & 1 & 1 & 1 & 0 \\ p_2 & 1 & 1 & 1 & 1 & 0 \\ p_3 & 1 & 1 & 1 & 1 & 0 \\ p_4 & 1 & 1 & 1 & 1 & 0 \\ p_5 & 0 & 0 & 0 & 0 & 1 \end{array} \tag{4}$$

Merging by Title

In this step, we use title information to process fragments generated from the previous step. We choose agglomerative clustering (Jain, Murty, & Flynn, 1999) as the basic framework. We consider each input fragment as a cluster, and then during each processing cycle, we find the most similar (the similarity measures will be defined later) pairs of clusters, and merge them, until the maximal similarity between clusters falls below a certain threshold.

We choose the term frequency (TF) model (Jones, 1972; Salton & Buckley, 1988) used in text processing to present elements in R_1. Titles from each fragment are combined together as a document. The model is shown in Formula (5).

$$D_i = (w_1, w_2, \ldots w_n) \tag{5}$$

Where D_i is the i_{th} document formed by the i_{th} fragment. w_i is the weight of the i_{th} term in D_i. The weight is the frequency of the i_{th} term appears in D_i. Then each fragment can be presented by a feature vector. The similarity between two fragments in R_1 is computed by a cosine similarity function, as follows:

$$sim(D_i, D_j) = \frac{\sum_k D_k * D_{jk}}{|D_i| * |D_j|} \tag{6}$$

The detailed procedure of merging by author is described by Algorithm 2. The algorithm merges two fragments to form a new fragment when the similarity is greater than the threshold. During the process of merging, the algorithm should keep new fragments accurate. In order to meet the requirement, we take two measures as follows:

ALGORITHM 2

```
Input: The first step result $R_1$
Output: the merging result $R_2$ based on title information
fused ← true;
while fused do
  fused ← false;
  for each $r_1$ in $R_1$ do
    maxSim = 0; r' = null; r" = null;
    for each $r_2$ in $R_1$ do
      if $r_1 \neq r_2$ then
        $t_1$ ← getTitleTerms ($r_1$);
        $t_2$ ← getTitleTerms ($r_2$);
        if titleSimilarty($t_1$, $t_2$) > maxSim then
          maxSim = titleSimilarity ($t_1$, $t_2$); r' = $r_1$; r" = $r_2$;
        end if
        if maxSim > titleThreshold then
          $r_1$ ← fuse ($r_1$, $r_2$); remove ($R_1$, $r_2$); fused = false;
        end if
      end if
    end for
  end for
end while
$R_2$ = $R_1$;
```

1. A customized stopword list for literature titles. Literature titles need to be short and concise for data processing purposes. Obviously, since the length of a title is much shorter than a document, a stopword list used in text processing for a long document is not suitable. We collect all the terms which appeared in the titles of DBLP. The preposition, numeral, and quantifiers with an occurrence frequency of over 0.002% are added to the stopword list. After being filtered by the stopword list, the remaining words are nouns representing an author's research fields and interested problems.
2. A dynamic threshold. A large threshold value can ensure merging accuracy. However, with the increase of $|S|$, the number of authors and the similarity probability between fragments increase at the same time. Therefore, we use Formula (7) to adjust a threshold that increases with $|S|$.

$$\theta = \alpha + \min\left(\left\lfloor \frac{|S|}{\chi} \right\rfloor * 0.1, \beta\right) \tag{7}$$

Where θ is the threshold, and α, β, χ are three variables to adjust the threshold. By our statistics, most of $|S|$ is between 2 and 1,000. In addition, the impact on results is not significant when $|S|$ has small changes in certain ranges. For instance, one $|S|$ is 200 and another $|S|$ is 250. Although they may share the same threshold, the impact on the results may not be significant. Therefore, we round down to keep the threshold, when $|S|$ has small changes in certain ranges.

As the similarity is computed by a cosine similarity function, the value range of θ is between 0 and 1. To guarantee θ in [0, 1], we normalized it as $\left\lfloor \frac{|S|}{\chi} \right\rfloor$. Some values of $\left\lfloor \frac{|S|}{\chi} \right\rfloor * 0.1$ may be too small and some may be too large. The values that are very small bring a very small threshold that results in over merger and low precision. Whereas, the values that are very large lead to a very large threshold which results in almost no merger and low recall. Therefore, we set a lower-bounded α and an upper-bounded $\alpha + \beta$ for the threshold.

We choose three names (David Brown, Liping Wang, and Wen Gao) to adjust the parameters (α, β, χ). When α, β, and χ are respectively set to 0.1, 0.3, and 200, the results of experiments on these three names are relatively better. In addition, the results of our experiments on the data set in the following section verify the suitability of the values of the parameters.

Merging by Venue

In this step, we use the venue information of the previously obtained fragments to merge articles that belong to different fields but are written by the same author. Similar to the step of merging by title in the last step, agglomerative clustering is also the basic framework of this step.

As venue information is contained in almost all bibliographies, it has been used by several disambiguation algorithms. Cota et al. (2010) and Han, Zha, and Giles (2005) used the term frequency-inverse document frequency (tf-idf) model to deal with venue, and Tang et al. (2012) used covenue (the same published venue); Yin, Han, and Yu (2007) connected articles through the same venue. The given methods established relations among articles from the same venue but ignore the relations among different but related venues. In order to get better results, it is necessary to mine and utilize these latent relations. Li et al. (2012) mined and used these relations through probability distribution. Despite their success, their method used local data with much noise to predict the relations. In addition, probabilistic models have high computational costs.

In order to better exploit venue information, we propose a novel model using non-negative matrix factorization (NMF) (Lee & Seung, 1999) by reference to latent semantic analysis (LSA) (Deerwester et al., 1990) to represent the relations among venues. We use LSA as proposed by Landauer and Dumais (Cormen et al., 2001) to extract the relations among words in documents. LSA performs a low-rank approximation on a term-document matrix, which is generated by transforming textual data into a vector representation, thereby exhibiting the semantic connectedness among the documents of the corpus. Singular value decomposition (SVD) is the normal approximation method used for LSA. However, the results of SVD contains negative components, which are not natural for interpreting textual representation. NMF approximates the original matrix by two submatrices with the non-negative constraint. In contrast to poor interpretability caused by negative entries in matrix factors in SVD based factorizations, the non-negativity in NMF ensures that factors contain coherent parts of the original data (Peter, Shivapratap, Divya, & Soman, 2009). In addition, due to its non-negativity, NMF is helpful in document clustering based on topics (Xu, Liu, & Gong, 2003).

By reference to LSA, we establish a matrix representing the relation between author and venue. Assume $V = \{v_1, v_2, v_3 \ldots v_n\}$ is the set of venues and $A = \{a_1, a_2, a_3 \ldots a_m\}$ is the set of authors. We establish the author-venue matrix as follows:

$$R = \begin{array}{c} \\ a_1 \\ a_2 \\ \vdots \\ a_m \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & \cdots & v_n \end{array} \\ \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix} \end{array} \tag{8}$$

Where $r_{ij} \in R$ represents the number of articles, that authors a_i has published in venue v_j. If $r_{ij} = 0$, author a_i has not published any article in venue v_j.

The procedure of NMF is that: given a $m \times n$ matrix $R = (r_{ij})_{m \times n}$ and a positive integer $k \le \min(m, n)$, NMF is to find two new non-negative matrixes $W \ge 0$ and $H \ge 0$ such that:

$$R \approx WH \tag{9}$$

Where $W = (w_{i,j})_{m \times k}$ is a $m \times k$ basis matrix, and $H = (h_{i,j})_{k \times n}$ is a $k \times n$ coefficient matrix. Generally, k is chosen in accordance with $(m + n)k < mn$.

Here we process NMF based on Euclidean distance as Formula (10) and multiplicative Formula (11) and (12) update rules.

$$E(R \| WH) = \frac{1}{2} \| R - WH \|_F^2 \tag{10}$$

$$H_{a\mu} \leftarrow H_{a\mu} \frac{\left(W^T R\right)_{a\mu}}{\left(W^T WH\right)_{a\mu}} \tag{11}$$

$$W_{ia} \leftarrow W_{ia} \frac{\left(RH^T\right)_{ia}}{\left(WHH^T\right)_{a\mu}} \tag{12}$$

Then we get the factorized venue matrix $H(k \times n)$. Each venue has k features. The relation between venues can be represented by the similarity of features. So the relations can be computed by Formula (13).

$$Q = H^T H = \begin{array}{c c} & \begin{array}{cccc} v_1 & v_2 & \cdots & v_n \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ \vdots \\ v_n \end{array} & \left[\begin{array}{cccc} q_{11} & q_{12} & \cdots & q_{1n} \\ q_{21} & q_{22} & \cdots & q_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ q_{n1} & q_{n2} & \cdots & q_{nn} \end{array}\right] \end{array} \tag{13}$$

Where q_{ij} means the relational grade between venue v_i and venue $v_{j.}$ The higher the grade, the closer a relation the two venues have.

Because the venue relation model proposed is generated from authors and the corresponding venues, it's very suitable for the name disambiguation problem. The model has the following requirements:

1. The relation between author and venue that is used to establish the model should be relatively accurate.
2. The data scale should ensure the accuracy, objectivity, and stability of the results from LSA.
3. The model should try to balance the effects on name disambiguation problem among all related venues. So the venues used to establish the model should cover different venues as much as possible.

To meet the requirements, we collect data for the model based on statistical sampling. We choose 250,000 articles from DBLP and cooperation count for each name (here, cooperation count is defined as the total number of coauthors. For instance, name a has two articles in DBLP, one is coauthored with name b and c, and the other one is coauthored with name d, e, and f. So a's cooperation count is 5). Names with high cooperation count can be regarded as representatives in certain domains and the relations with their corresponding venues can also be represented relatively accurately so that accurate relations between venues can be obtained. It is obvious that full English names are hardly duplicated. Therefore, we choose full English names from the top 5% collaboration scores and their corresponding venues. Figure 3 shows the distribution of collaboration count (in descending order) for author names, and Table 1 show the top 10 names in terms of collaboration count. We then get a 2961×4772 matrix. The number of venues in DBLP is 5,885. Therefore, the matrix clearly meets the above three requirements.

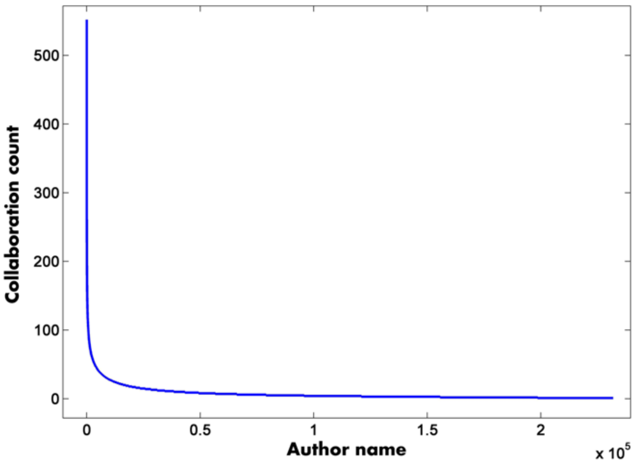

FIG. 3. Distribution of collaboration count. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

TABLE 1. Top 10 names in terms of cooperation.

Name	Collaboration score
Alberto L. Sangiovanni-Vincentelli	552
Luca Benini	471
Nicholas R. Jennings	438
Francky Catthoor	430
Mahmut T. Kandemir	422
Wayne Luk	381
Jurgen Teich	337
Hideharu Amano	331
Sushil Jajodia	306
Luc J. Van Gool	306
Wil M. P. van der Aalst	296

The venue similarity can be computed by the venue relation model Q. For the venue sets $V_1 = \{v_1, v_2, v_3, \ldots, v_m\}$ and $V_2 = \{v_1', v_2', v_3', \ldots, v_n'\}$ from two fragments, we obtain their similarity as follows:

$$\mathrm{Sim}_V(V_1, V_2) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} Q\left(v_i, v_j'\right)}{|V_1| \times |V_2|} \tag{14}$$

Similar to the previous step, we also set a dynamic threshold for venue merging in consideration of the increase in complexity with |S|.

TABLE 2. Statistics of the data set.

Name	#Pubs	# True author
Hui Fang	45	8
Ajay Gupta	25	8
Rakesh Kumar	104	8
Michael Wagner	61	16
Bing Liu	192	23
Jim Smith	54	5
Lei Wang	400	144
Wei Wang	833	216
Bin Yu	102	18

$$\alpha = \alpha + \min\left(\left\lfloor \frac{|S|}{\chi} \right\rfloor, \beta\right) \tag{15}$$

Formula (15) is similar to Formula (7). The difference between them is that the similarity of Formula (15) is computed by Formula (14) whose value range of the similarity is not in [0, 1] and may be larger than 1. Therefore, $\left\lfloor \frac{|S|}{\chi} \right\rfloor$ does not need to be reduced an order of magnitude.

As in the determination process of parameters in Formula (7), we choose the same three names to adjust the parameters and the results are better when α, β, χ are respectively set to 0.2, 5, and 200.

Experiments

Experimental Settings

Data set. We choose the data set used by Li et al. (2012). The data set is extracted from a January 2011 dump of DBLP. It contains 1.5 million items, which is very close to real life applications. Meanwhile, in order to compare with other methods, we run the algorithm under the same conditions. We use nine representative names to test and their statistics are shown in Table 2.

Comparison methods. We compared our method with three representative methods: "Distinct" (Yin et al., 2007), Arnetminer (Tang et al., 2012), and CSLR (Li et al., 2012). Detailed descriptions of these methods are given as follows. Distinct is the state-of-the-art method in supervised methods. Arnetminer has good performance in solving this problem and has been applied in a real world system (http://arnetminer.org). CSLR uses the latent relations among venues and yields good results in experiments. We reuse the results of these three contrastive methods from Li et al. (2012), since the settings are the same.

Evaluation metrics. To measure the prediction performance of our proposed method, we adopt pairwise precision, pairwise recall, and pairwise F1 scores, which are also used in Li et al. (2012), Tang et al. (2012), and Yin et al. (2007). The pairwise measures are adapted for evaluating disambiguation by considering the number of pairs of articles assigned with the same label. Here the label for a pair of articles denotes whether they are grouped in the same cluster. Any pair of articles that is annotated consistently with the ground truth is called a correct pair, and any pair of articles whose label is inconsistent with the ground truth is called a wrong pair. Note the counting is for pairs of articles with the same label (either predicted or labeled) only. Thereafter, we define the three scores:

$$PariwisePrecision = \frac{\#\,PairsCorrectlyPredicted}{\#\,TotalPairsPredicted} \tag{16}$$

$$PariwiseRecall = \frac{\#\,PairsCorrectlyPredicted}{\#\,TotalCorrectPairs} \tag{17}$$

$$PariwiseF1 = \frac{2 \times PairwisePrecision \times PairwiseRecall}{PairwisePrecision + PairwiseRecall} \tag{18}$$

Because bibliographic management systems require fast responses from the name disambiguation process, we also compare the efficiency of the competitive methods by measuring the run time.

Experimental results and discussion

Prediction evaluation. The prediction scores on the pairwise precision, pairwise recall and pairwise F1 for all the methods are shown in Table 3. For each name, we mark the best pairwise F1 score in bold. In addition, we underline the average pairwise F1 for each method in order to highlight them. In particular, the improvements of our FMC over all the other methods on all data sets are statistically significant, p-value < 0.05. For a better comparison, we also present the average pairwise F1 scores in Figure 4. It can be observed that FMC achieves the best average pairwise F1 score, and improves previous methods by 7.3%, 22.7%, and 2.4%, respectively.

From Table 3, we can observe that both FMC and CSLR outperform the other two methods. The failure of Distinct and Arnetminer may be explained by the missing of the venue information. As a result, the precision of Distinct is too low, which limited its practical applicability. Arnetminer also neglects the venue information, but requires excessive attributes and the relations among them, which are difficult to obtain. Given insufficient information (i.e., coauthor networks, titles, and venues), the performance of Arnetminer drops significantly in its real world application. For example, precision in recognizing the Chinese name "Wei Wang" in Arnetminer is extremely low: more than 900 articles are credited to professor Wei Wang at UCLA on the Arnetminer website at http://arnetminer.org/, whereas there are no more than 200 publications according to the author's webpage at http://www.cs.ucla.edu/~weiwang/chrolist.html.

TABLE 3. Experimental results (%).

	Distinct			Arnetminer			CLSR			Our (FMC)		
Name	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
Hui Fang	81.3	97.9	88	59.1	63.7	61.3	100.0	78.9	88.2	100.0	99.6	**99.8**
Ajay Gupta	65.3	87.9	**74.2**	60.0	65.4	62.6	96.0	39.6	56.1	70.6	61.5	65.8
Rakesh Kumar	89.9	96.0	92.5	98.4	89.3	93.7	99.9	97.8	**98.8**	92.6	88.5	90.5
Michael Wagner	67.4	98.2	**79.1**	55.6	36.7	44.2	88.1	64.6	74.6	81.7	62.8	71.0
Bing Liu	83.0	84.7	83.3	75.7	67.2	71.2	98.1	74.7	84.8	92.1	86.8	**89.4**
Jim Simth	94.8	87.8	90.0	88.6	45.1	59.7	100.0	48.8	65.6	100.0	89.8	**94.6**
Lei Wang	29.3	85.9	42.4	18.1	23.1	29.8	78.1	87.6	**82.6**	63.2	78.6	70.1
Wei Wang	25.8	84.2	38.9	9.7	88.2	17.5	81.0	71.8	**76.1**	36.9	88.2	52.0
Bin Yu	54.0	62.0	57.0	72.4	62.2	66.9	88.0	49.1	63.0	64.0	99.4	**77.9**
Average	65.6	97.2	71.7	59.7	60.1	56.3	92.1	68.1	76.6	77.9	83.9	**79.0**

Note. The best F1 measures are marked in bold. The average F1 values are underlined. In particular, the improvements of our FMC over all the other methods on all data sets are statistically significant (p-value < 0.05).

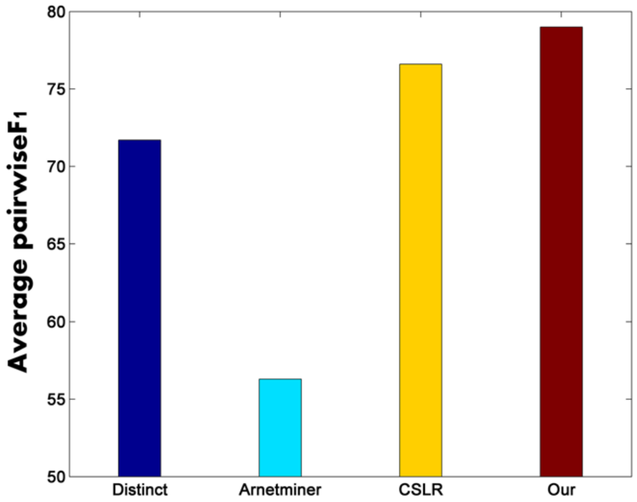

FIG. 4. Average pairwise F1 scores for four methods. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

When compared with CSLR, FMC performs slightly better than CSLR. It can be observed from Table 3 and Figure 5: FMC achieves the best average pairwise F1 score, and delivers 4 of the best results and behaves similarly to the best in the other five cases. For FMC, the most challenging cases are those of "Lei Wang" and "Wei Wang," where too many authors share these names, as confirmed by the statistics in Table 2. Nonetheless, FMC has outstanding performance when dealing with common cases, which are more frequently observed in real world applications.

In addition, when considering the balance between precision and recall, previous methods prefer precision over recall, resulting in articles from one author being grouped in different clusters. This increases the burden for system management and confuses system users when they search for author information. Methods with balanced performances on name disambiguation can reduce the fragments of profiles while keeping the content in the profiles relatively correct. Table 3 shows that our method achieves competitive recall scores while keeping precision at a high level. Therefore, FMC, with its highest pairwise F1 score, is a balanced method with high practicality.

Running time comparison. We show the running time of CSLR and FMC in Table 4, since these two methods achieve much better prediction performance than the other two methods, that is Distinct and Arnetminer.

As shown in Table 4, FMC is 35 times faster on average, while achieving significantly better prediction performance. The reasons are as follows: (a) when dealing with the title attribute, our customized stop word list filters out most of the unnecessary words. Meanwhile, TF is a simple but efficient model in modeling term importance; and (b) FMC only needs to get values from the established models in order to save the large amount of calculation in mining the latent relations in the stage of matrix factorization.

Discussion

In the effort to explain the results, we analyze the failure cases and find the reasons to be as follows: (a) some scholars' articles lie outside their main research fields; (b) some authors with the same name have similar research fields. For example, Professor "Bin Yu" at the University of California, Berkeley has several research fields including machine learning, empirical processes, signal processing, and information theory. These fields overlap with a different "Bin Yu" in Electroglas Inc. and Information Engineering University. A similar situation occurs in the case of "Lei Wang." There are at least eight different individuals who are interested in wireless related problems. These cases are not common but have a significant effect on evaluation results. And it will be difficult even for a human to make the correct decision based on these three attributes. Under the circumstances, more information may be provided for human judgment. Likewise, extra information may be needed for the algorithm to

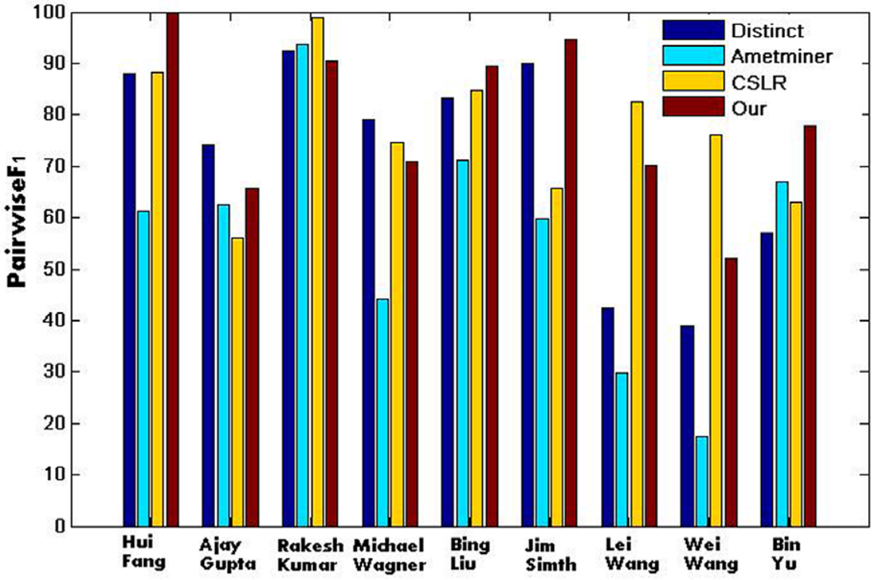

FIG. 5. Pairwise F1 scores for four methods for each name. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

TABLE 4. The run time comparison of CSLR and FMC.

Name	CSLR	FMC
Hui Fang	2s	0.053s
Ajay Gupta	5s	0.046s
Rakesh Kumar	5s	0.352s
Michael Wagner	6s	0.182s
Bing Liu	126s	6.757s
Jim Smith	3s	0.051s
Lei Wang	573s	20.152s
Wei Wang	3,727s	97.731s
Bin Yu	14s	0.726s
Average	495.667s	14.006s

handle the very ambiguous cases. And great deliberation may be needed in terms of how to effectively obtain and use the extra information.

Related Work

Numerous advanced methods from different research areas have been proposed to solve the name disambiguation problem. A recent review by Ferreira et al. (2012) lists 17 advanced methods. A comprehensive review of the problem and suggested solutions are given by Smalheiser and Torvik (2009). These methods can be divided into two categories according to their dependence on training sets: supervised methods (Ferreira, Veloso, Gonçalves, & Laender, 2010; Han, Giles, Zha, Li, & Tsioutsiouliklis, 2004; Veloso, Ferreira, Gonçalves, Laender, & Meira, 2012; Yin et al., 2007) and unsupervised methods (Bhattacharya & Getoor, 2007; Cota et al., 2010; D'Angelo et al., 2011; Fan et al., 2011; Getoor, 2006; Han et al., 2005; Huang, Ertekin, & Giles, 2006; Kanani, McCallum, & Pal, 2007; Kang et al., 2009; Li et al., 2012; Pereira et al., 2009; Tang et al., 2012; Torvik & Smalheiser, 2009; Torvik, Weeber, Swanson, & Smalheiser, 2005; Yang, Peng, Jiang, Lee, & Ho, 2008).

Supervised Methods

Supervised methods aim to solve the problem mainly through label information to train the model before disambiguating each author. They take as input a set of training examples consisting of pairs of articles which are labeled as either positive or negative for training a model. Then, they use the trained model to predict the author assignment of each article. For example, Han et al. (2004) propose two methods based on supervised learning techniques using Naïve Bayes and support vector machines (SVM) (Vapnik, 1999), respectively. Yin et al. (2007) propose a method called Distinct which also uses SVM and distinguishes object identities by fusing different types of linkages with differentiating weights. Veloso et al. (2012) propose "SLAND," a disambiguation method that infers the author of a reference by using a supervised rule-based associative classifier. SLAND is extended by Ferreira et al. (2010) to become self-trained but also uses a rule-based associative classifier. These methods are usually very effective when faced with a large number of examples of citations for each author and have successful applications in some cases. However, the acquisition of training examples usually requires skilled human intervention in training set labeling, and therefore requires additional labor costs. What is more, since examples for all possible authors may not be included in the training data and an author's research interest may evolve over time, new examples need to be inserted into the training data continuously and the methods need to be retrained periodically to maintain their effectiveness. Thus, these supervised methods cannot be used in applications because of their huge data and real-time updating requirements.

Unsupervised Methods

Because of the disadvantages of supervised methods, researchers have proposed unsupervised methods. Most of these unsupervised approaches formulate the author name disambiguation problem as a clustering task, where each cluster contains all the articles by the same author. These methods usually have three steps in common: using a model to represent the data, computing similarities, and clustering articles. But they usually differ in the selection of models, or similarity functions or clustering methods. Therefore, various models, similarity functions and clustering methods have been used or proposed to solve author name disambiguation problem. Concrete examples are as follows:

In Step 1, a model is used to represent the data to be differentiated. For example, Han et al. (2005) represent the data by a citation-term matrix and use tf-idf (Jones, 1972; Salton & Buckley, 1988) or normalized term frequency to calculate the weight of each element in the matrix. Huang et al. (2006) present a model in which a blocking method is first applied to create blocks of references to authors with similar names. Fan et al. (2011) propose a graph-based method called "GHOST" and it represents data as a graph, where each vertex represents a reference to be disambiguated and each undirected edge represents a coauthorship. Tang et al. (2012) incorporate all the relations among attributes of article (e.g., whether two articles are published in the same venue, or whether they share a same coauthor, or whether they cite each other, etc.) into hidden Markov random field (HMRF) to estimate the weight of the feature functions.

Next, the similarities are computed through similarity functions in step 2. For example, Han et al. (2005) and Cota et al. (2010) use a cosine similarity function in their methods. Fan et al. (2011) propose a graph-based similarity function and Li et al. (2012) propose a novel categorical set similarity measure for calculating the similarities. In addition, Huang et al. (2006) and Bhattacharya and Getoor (2007) use different functions for each attribute of data according to the characteristics of each attribute (such as the edit distance for e-mails and URLs, Jaccard similarity for addresses and affiliations, and soft-TFIDF [Cohen, Ravikumar, & Fienberg, 2003] for names, etc.).

Finally, a suitable clustering algorithm is chosen to complete the disambiguation. Various classical clustering algorithms are used in unsupervised methods. These papers (Bhattacharya & Getoor, 2007; Cota et al., 2010; Getoor, 2006; Kang et al., 2009; Li et al., 2012; Pereira et al., 2009; Torvik & Smalheiser, 2009) all use agglomerative clustering (Jain et al., 1999) to complete the disambiguation. Han et al. (2005) use K-way spectral clustering method (Zha, Ding, Gu, He, & Simon, 2001) and Huang et al. (2006) present a method using density-based spatial clustering of applications with a noise (DBSCAN) clustering algorithm (Ester, Kriegel, Sander, & Xu, 1996) for clustering articles. Fan et al.'s (2011) GHOST uses an affinity propagation clustering method (Frey & Dueck, 2007) to group the references to the same author in the last step.

Some unsupervised methods solve the name disambiguation problem with the help of data from external data sources, such as the Internet, search engine results, and statistical data. For example, in D'Angelo et al. (2011) and Torvik et al. (2005), in addition to the attributes of article title, venue title, coauthorship of records, other attributes, including language, affiliation, and research area are used for solving the author name ambiguity problem. The additional evidence from the web, such as topics, correlations, and publication pages of the authors or coauthors are gathered and also used for solving the problem (Kanani et al., 2007; Kang et al., 2009; Pereira et al., 2009; Yang et al., 2008). Although these external data sources can improve the effectiveness of author name disambiguation, such information is not always available, and can be difficult to obtain in practical applications. What's more, using these external data sources may lead to additional computation costs. Therefore, these methods are not suitable to be used in applications.

Unsupervised clustering methods have attracted the attention of scholars because of their good results, high flexibility (without using labeled training examples), and lower cost of time than supervised methods. However, the previous unsupervised clustering methods may not be suitable enough for practical applications for two reasons: (a) some of them require excessive attributes, which are difficult to obtain in practical applications; (b) some may also be quite complex and time consuming, which cannot be tolerated by existing system. Notably, our proposed method has the advantages of light requirements on article attributes, good disambiguation performance, and high efficiency, thus providing a practical solution to name disambiguation systems.

Conclusion and Future Work

We proposed a coarse-to-fine multiple clustering framework, called FMC, for name disambiguation in literature management. FMC solves the problem on the basis of multiple clustering. Each clustering step is based on different attributes. FMC completes the first clustering based on the reachability matrix formed by coauthor relations. We also analyzed the venue attribute of articles, and discovered the latent relations among venues based on the author-venues relationship, to further tune the clustering performance. Experimental results show that our proposed method not only greatly improves the prediction performance of previous methods, but also significantly saves the computation time. Another merit of the proposed method is that it only requires three common attributes, and is easily deployable in

DOI: 10.1002/asi

traditional digital libraries as well as other literature resource management systems, such as CiteULike and CiteSeerX, whose data sources are various and data quality is not stable.

In our future work, we will explore other matrix factorization algorithms to improve the venue relation model. This may further improve the effectiveness of name disambiguation. In addition, we will also try to develop a method to mine and use the relations among keywords in article titles for solving the name disambiguation problem, similar to the latent relations among venue.

Acknowledgments

We would like to thank the editor-in-chief and the anonymous reviewers of JASIST for their valuable comments and suggestions to improve the quality of the article.

References

Aini, A., & Salehipour, A. (2012). Speeding up the Floyd–Warshall algorithm for the cycled shortest path problem. Applied Mathematics Letters, 25(1), 1–5.

Bhattacharya, I., & Getoor, L. (2007). Collective entity resolution in relational data. ACM Transactions on Knowledge Discovery from Data (TKDD), 1(1), 5.

Cohen, W.W., Ravikumar, P.D., & Fienberg, S.E. (2003). A comparison of string DISTANCE metrics for name-matching tasks. Paper presented at the IIWeb.

Cormen, T.H., Leiserson, C.E., Rivest, R.L., & Stein, C. (2001). Introduction to algorithms. Cambridge: MIT press.

Cota, R.G., Ferreira, A.A., Nascimento, C., Gonçalves, M.A., & Laender, A.H.F. (2010). An unsupervised heuristic-based hierarchical method for name disambiguation in bibliographic citations. Journal of the American Society for Information Science and Technology, 61(9), 1853–1870.

D'Angelo, C.A., Giuffrida, C., & Abramo, G. (2011). A heuristic approach to author name disambiguation in bibliometrics databases for large-scale research assessments. Journal of the American Society for Information Science and Technology, 62(2), 257–269.

Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., & Harshman, R. (1990). Indexing by latent semantic analysis. Journal of the American Society for Information Science, 41, 391–407.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. KDD, 96, 226–231.

Fan, X., Wang, J., Pu, X., Zhou, L., & Lv, B. (2011). On graph-based name disambiguation. Journal of Data and Information Quality (JDIQ), 2(2), 10.

Ferreira, A.A., Gonçalves, M.A., & Laender, A.H.F. (2012). A brief survey of automatic methods for author name disambiguation. ACM SIGMOD Record, 41(2), 15–26.

Ferreira, A.A., Veloso, A., Gonçalves, M.A., & Laender, A.H.F. (2010). Effective self-training author name disambiguation in scholarly digital libraries. Paper presented at the Proceedings of the 10th Annual Joint Conference on Digital Libraries, ACM.

Frey, B.J., & Dueck, D. (2007). Clustering by passing messages between data points. Science, 315(5814), 972–976.

Getoor, I.B.L. (2006). A latent dirichlet model for unsupervised entity resolution. Paper presented at the Proceedings of the Sixth SIAM International Conference on Data Mining, SDM.

Han, H., Giles, L., Zha, H., Li, C., & Tsioutsiouliklis, K. (2004). Two supervised learning approaches for name disambiguation in author citations. Paper presented at the Digital Libraries, 2004. Proceedings of the 2004 Joint ACM/IEEE Conference on, IEEE.

Han, H., Zha, H., & Giles, C.L. (2005). Name disambiguation in author citations using a k-way spectral clustering method. Paper presented at the Digital Libraries, 2005. JCDL'05. Proceedings of the 5th ACM/IEEE-CS Joint Conference on, IEEE.

Huang, J., Ertekin, S., & Giles, C.L. (2006). Efficient name disambiguation for large-scale databases. In J Fürnkranz, T Scheffer, and M Spiliopoulou (Eds.), Knowledge discovery in databases: PKDD 2006 (pp. 536–544). Berlin Heidelberg: Springer.

Jain, A.K., Murty, M.N., & Flynn, P.J. (1999). Data clustering: A review. ACM Computing Surveys (CSUR), 31(3), 264–323.

Jones, K.S. (1972). A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation, 28(1), 11–21.

Kanani, P., McCallum, A., & Pal, C. (2007). Improving author coreference by resource-bounded information gathering from the web. Paper presented at the Proceedings of IJCAI, Hyderabad, India.

Kang, I.-S., Na, S.-H., Lee, S., Jung, H., Kim, P., Sung, W.-K., & Lee, J.-H. (2009). On co-authorship for author disambiguation. Information Processing & Management, 45(1), 84–97.

Lee, D.D., & Seung, H.S. (1999). Learning the parts of objects by non-negative matrix factorization. Nature, 401, 788–791.

Li, S., Cong, G., & Miao, C. (2012). Author name disambiguation using a new categorical distribution similarity. In P.A. Flach, T. De Bie, & N. Cristianini (Eds.), Machine learning and knowledge discovery in databases (pp. 569–584). Berlin Heidelberg: Springer.

Pereira, D.A., Ribeiro-Neto, B., Ziviani, N., Laender, A.H., Gonçalves, M.A., & Ferreira, A.A. (2009). Using web information for author name disambiguation. Paper presented at the Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries, ACM.

Peter, R., Shivapratap, G., Divya, G., & Soman, K.P. (2009). Evaluation of SVD and NMF methods for latent semantic analysis. International Journal of Recent Trends in Engineering, 1(3), 308–310.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information Processing & Management, 24(5), 513–523.

Smalheiser, N.R., & Torvik, V.I. (2009). Author name disambiguation. Annual Review of Information Science and Technology, 43(1), 1–43.

Tang, J., Fong, A.C.M., Wang, B., & Zhang, J. (2012). A unified probabilistic framework for name disambiguation in digital library. IEEE Transactions on Knowledge and Data Engineering, 24(6), 975–987.

Torvik, V.I., & Smalheiser, N.R. (2009). Author name disambiguation in MEDLINE. ACM Transactions on Knowledge Discovery from Data (TKDD), 3(3), 11.

Torvik, V.I., Weeber, M., Swanson, D.R., & Smalheiser, N.R. (2005). A probabilistic similarity metric for Medline records: A model for author name disambiguation. Journal of the American Society for Information Science and Technology, 56(2), 140–158.

Vapnik, V. (1999). The nature of statistical learning theory. New York: Springer.

Veloso, A., Ferreira, A.A., Gonçalves, M.A., Laender, A.H.F., & Meira, W., Jr. (2012). Cost-effective on-demand associative author name disambiguation. Information Processing & Management, 48(4), 680–697.

Xu, W., Liu, X., & Gong, Y. (2003). Document clustering based on non-negative matrix factorization. Paper presented at the Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, ACM.

Yang, K.-H., Peng, H.-T., Jiang, J.-Y., Lee, H.-M., & Ho, J.-M. (2008). Author name disambiguation for citations using topic and web correlation. In B. Christensen-Dalsgaard, D. Castelli, B.A. Jurik, & J. Lippincott (Eds.), Research and advanced technology for digital libraries (pp. 185–196). Berlin Heidelberg: Springer.

Yin, X., Han, J., & Yu, P.S. (2007). Object distinction: Distinguishing objects with identical names. Paper presented at the Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on, IEEE.

Zha, H., Ding, C., Gu, M., He, X., & Simon, H. (2001). Spectral relaxation for k-means clustering. Advances in Neural Information Processing Systems, 14, 1057–1064.

DOI: 10.1002/asi